

JAVA BİLGİSAYAR DİLİYLE PROGRAMLAMA



Java 8

M. Turhan ÇOBAN

**EGE Üniversitesi, Mühendislik Fakültesi,
Makine Mühendisliği bölümü**

turhan.coban@turhancoban.com

JAVA BİLGİSAYAR DİLİYLE PROGRAMLAMA

İÇİNDEKİLER

1.0 PROGRAMLAMA DİLİNE GİRİŞ

- 1.1 NİÇİN JAVA PROGRAMLAMA DİLİ
- 1.2 BİLGİSAYARDA JAVA PROGRAMLARINI KURMA
- 1.3 JAVA PROGRAMINI YAZMA VE ÇALIŞTIRMA
- 1.4 TAMSAYI VE GERÇEK SAYI DEĞİŞKENLER VE MATEMATİK İŞLEMLER
- 1.5 STRING, CHAR VE BOOLEAN TİPİ DEĞİŞKENLER VE İŞLEMLERİ
- 1.6 JAVA NESNESİ OLARAK TANIMLANMIŞ TEMEL DEĞİŞKENLER
- 1.7 FİNAL TERİMİ VE SABİTLER
- 1.8 JAVA DİLİNDE BİT KOMUTLARI VE İŞLEMLERİ
- 1.9 JAVA DİLİNİN TEMEL KOMUTLARI : IF KOMUT YAPISI
- 1.10 JAVA DİLİNİN TEMEL KOMUTLARI : WHILE, DO-WHILE, FOR, SWITCH-CASE
TEKRARLAMA KOMUT YAPILARI
- 1.11 ARİTMETİK İŞLEMLERDE DEĞİŞKEN TÜRÜ DEĞİŞTİRME (CASTING) OPERASYONU
- 1.12 SAYI DEĞİL VE SONSUZ SONUÇLARI
- 1.13 EKERSİZLER
- 1.14 PROBLEMLER

2.0 SINIF KAVRAMINA GİRİŞ VE METODLAR

- 2.1 JAVA API KÜTÜPHANESİ
- 2.2 METODLAR (FONKSİYONLAR-SUBROUTİNLER)
- 2.3 NESNE (OBJECT) TANIMI VE METOTLARDA KULLANIMI
- 2.4 METOTLARIN KENDİ KENDİNİ ÇAĞIRMASI (RECURSION)
- 2.5 AYNI ADLI METOTLARIN BİR ARADA KULLANILMASI (OVERLOADİNG)
- 2.6 METOT (METHOD) VE SINIF (CLASS) DEĞİŞKENLERİ
- 2.7 ALIŞTIRMALAR
- 2.8 PROBLEMLER

3.0 SINIF (CLASS) YAPILARINA GİRİŞ

- 3.0 STANDART KAVRAMLARIN TÜRKÇE KARŞILIKLARI
- 3.1 SINIF YAPISININ GENEL TANIMI
- 3.2 KURUCU (CONSTRUCTOR) METOT
- 3.3 DİĞER METOTLAR
- 3.4 THIS DEYİMİNİN KULLANIMI
- 3.5 SINIF DEĞİŞKENLERİNİN DIŞ DÜNYADAN GİZLENMESİ
- 3.6 ALIŞTIRMALAR
- 3.7 PROBLEMLER
- 3.8 HAFTA 6 ÖDEVLERİ

4.0 SINIFLARI BİR ARADA KULLANMA, SINIF NESNELERİNİN DİĞER SINIFLARDA SINIF DEĞİŞKENİ OLARAK KULLANILMASI (COMPOSITION), KALITIM (INHERITANCE)

- 4.1 SINIF NESNELERİNİN DİĞER BİR SINIFTA KULLANILMASI (COMPOSITION)
- 4.2 SINIF NESNELERİNİN BİRBİRİNDEN KALITIM YOLUYLA TÜRETİLMESİ (INHERITANCE)
- 4.3 ALIŞTIRMALAR
- 4.4 PROBLEMLER

5.0 BOYUTLU DEĞİŞKENLER

- 5.0 BOYUTLU DEĞİŞKENLERİN GENEL TANIMI
- 5.1 BİR BOYUTLU DEĞİŞKENLER VE NESNELER
- 5.2 GELİŞTİRİLMİŞ FOR DÖNGÜ YAPISI

- 5.3 NESNELERİN BOYUTLU YAPILARI
- 5.4 NESNE VE DEĞİŞKENLERİN ÇOK BOYUTLU KULLANIMI
- 5.5 BOYUTLU DEĞİŞKENLERDE BOYUT DEĞİŞTİRME
- 5.6 ALIŞTIRMALAR
- 5.7 PROBLEMLER

6.0 ABSTRACT SINIFLAR, INTERFACE, İNDİREK REFERANS KULLANIMI, İÇ SINIFLAR VE LAMDA DEYİMİ

- 6.1 ABSTRACT SINIFLAR
- 6.2 INTERFACE
- 6.3 İÇ SINIF TANIMI VE LAMBDA DEYİMLERİ
- 6.4 ALIŞTIRMALAR
- 6.5 PROBLEMLER

7.0 GRAFİK ÇİZİMİ

- 7.1 JAVA SWING GRAFİK ORTAMI JFRAME VE JAPPLET SINIFLARI
- 7.2 JAPPLET KULLANARAK GRAFİK PENCERESİ OLUŞTURMA VE KULLANMA
- 7.3 ALIŞTIRMALAR
- 7.4 PROBLEMLER

8.0 GRAFİK KONTROL VE PROGRAMLAMA-GRAFİK KULLANICI ARAYÜZÜ (GRAPHIC USER INTERPHASE-GUI)

- 8.1 JBUTTON
- 8.2 JLABEL, JTEXTFIELD VE JTEXTAREA GRAFİK KONTROL SINIFLARI
- 8.3 MOUSE KONTROLU
- 8.4 KONTROL KUTUSU (JCHECKBOX)
- 8.5 RADIO KONTROL DÜĞMESİ
- 8.6 LIST KONTROLU JSCROLLPANE FORMAT DEYİMİ
- 8.7 MENU KONTROLU
- 8.8 JSLIDER GRAPHIC KONTROLU
- 8.9 JTABLE TABLO OLUŞTURMA
- 8.10 GRAFİK SAYFASINI FORMATLAMA
- 8.11 ALIŞTIRMALAR
- 8.12 PROBLEMLER

9.0 HATA KONTROLU

- 9.1 GENEL TANIMLAR
- 9.2 ALIŞTIRMALAR
- 9.3 PROBLEMLER

10.0 İLERİ VERİ YAPILARI

- 10.1 GENERIC(GENEL OLARAK TANIMLANMIŞ) METHODLAR
- 10.2 GENERIC(GENEL OLARAK TANIMLANMIŞ) SINIFLAR
- 10.3 LAMDA DEYİMLERİNİ GENEL SINIF TANIMLARIYLA KULLANMA
- 10.4 STRINGTOKENIZER SINIFI
- 10.5 STRINGBUFFER SINIFI
- 10.6 DİNAMİK BOYUTLU DEĞİŞKEN KULLANIMI : LIST, ARRAYLIST, VECTOR, LINKEDLIST,QUE SINIFLARI
- 10.7 DİNAMİK HAFIZAYLA TANIMLANAN BOYUTLU SINIFLAR : TREE (AĞAÇ YAPILARI)
- 10.8 DİNAMİK HAFIZAYLA TANIMLANAN BOYUTLU SINIFLAR :: HASHTABLE YAPISI
- 10.9 DİNAMİK HAFIZAYLA TANIMLANAN BOYUTLU SINIFLAR: NAVIGABLE SET

- 10.10 DİNAMİK HAFIZAYLA TANIMLANAN BOYUTLU SINIFLAR: VERİ DEPOLAMA İÇİN
ARRAYLIST YAPISINI KULLANAN BİR PROGRAM
- 10.11 ALIŞTIRMALAR
- 10.12 PROBLEMLER

11.0 DOSYALAR, DOSYALARDAN VERİ OKUMA VE YAZMA

- 11.1 FILE SINIFI
- 11.2 READING SEQUENTIAL FILES THROUGH CHANNELS
 - 11.2.1 BUFFERED READER SINIFI
 - 11.2.2 SCANNER SINIFI
- 11.3 WRITING INTO SEQUENTIAL FILES THROUGH CHANNELS
- 11.4 ZIP, UNZIP SINIFLARI
- 11.5 ALIŞTIRMALAR
- 11.6 PROBLEMLER

12.0 VERİTABANI PROGRAMLAMA

- 12.1 SQLITE VERİTABANI
- 12.2 SQLITE VERİTABANI İLE SQL KOMUTLARININ KULLANIMI
- 12.3 SQL VE SQLITE VERİ TABANININ JAVA ORTAMINDA KULLANILMASI
- 12.4 HAFTA 16 EKERSİZLERİ
- 12.5 HAFTA 16 ÖDEVLERİ

13.0 HESAP ÇİZELGESİ PROGRAMLARINDAN VERİ OKUMA VE YAZMA

- 17.1 GENEL TANIM
- 17.2 HESAP ÇİZELGESİ PROGRAMLARINDAN VERİ OKUMA
- 17.3 HESAP ÇİZELGESİ PROGRAMLARINI FORMATLAMA
- 17.4 HESAP ÇİZELGESİ PROGRAMLARINA VERİ YAZMA
- 17.4 ALIŞTIRMALAR
- 17.5 PROBLEMLER

14.0 GERÇEK ZAMAN PROGRAMLAMA

- 18.1 PROGRAMLAMADA THREAD KULLANIMI
- 18.2 SİNKRONİZASYON
- 18.4 ALIŞTIRMALAR
- 18.5 PROBLEMLER

15.0 JAVA PROGRAMLARININ İÇİNDE BAŞKA JAVA PROGRAMLARINI YAZDIRMA, DERLEME VE ÇALIŞTIRMA (GERÇEK ZAMAN PROGRAMLAMA)

- 19.1 JAVA PROGRAMLARINI GERÇEK ZAMANDA DERLEME
- 19.2 PROGRAM İÇİNDE PROGRAM YAZMA VE DERLEME
- 19.3 ALIŞTIRMALAR
- 19.4 PROBLEMLER

16.0 NETWORK PROGRAMLAMAYA GİRİŞ

- 16.1 UNIFORM RESOURCE LOCATORS (URL) YAPISI
- 16.2 SOCKET PROGRAMLAMA (TCP/IP)
- 16.3 DATAGRAM
- 16.4 ALIŞTIRMALAR
- 16.5 PROBLEMLER

17.0 JAVA ANADİL PROGRAMLAMASINA GİRİŞ

- 17.1 TEMEL KAVRAMLAR, C VE C++ PROGRAMLAMA DİLİ
- 17.2 DEĞİŞKENLERİN VE JAVA NESNELERİNİN AKTARILMASI

17.3 ALIŖTIRMALAR

17.4 PROBLEMLER

19. REFERANS LİSTESİ

20.0 EKLER

Java deyimleri özeti

Konsol (dos) girdi Scanner input = new Scanner(System.in); int intValue = input.nextInt(); long longValue = input.nextLong(); double doubleValue = input.nextDouble(); float floatValue = input.nextFloat(); String string = input.next(); Konsol (dos) çıktı System.out.println(herhangibirdeğer);	Kullanıcı arayüzü(GUI) grafik girdi String string = JOptionPane.showInputDialog("Enter input"); int intValue = Integer.parseInt(string); double doubleValue = Double.parseDouble(string); Kullanıcı arayüzü(GUI) grafik çıktı JOptionPane.showMessageDialog(null , "girdiyi yaz");
--	---

Temel veri türleri byte 8 bits (-128 den 127 e) short 16 bits (-32768 den 32767 e) int 32 bits (-2147483648 den 2147483647 e) long 64 bits (-9223372036854775808 den 9223372036854775808 e) float 32 bits (-3.40292347e+38 den 3.40292347e+38 e) double 64 bits (1.7976931348623157e+308 den 1.7976931348623157e+308 e) char 16 bits (Unicode) boolean 1 bit (true/false)	Aritmetik işlemler + toplama - çıkarma * çarpma / bölme % kalan ++ var bir artma (önceden) -- var bir azalma(önceden) var ++ bir artma(sonradan) var --bir azalma(sonradan)	işlemler = eşitleme, atama += toplama -= çıkarma *= çarpma /= bölme %= kalan işlemleri
Karşılaştırma işlevçileri < küçüktür <= küçüktür veya eşittir > büyüktür >= büyüktür veya eşittir == eşittir != eşit değildir	Mantık işlevçileri && VE VEYA ! DEĞİL ^ exclusive VEYA	if (Eğer) deyimi if (şart1-boolean) {deyimler;} else if (şart2-boolean) {deyimler;} else if (şart3-boolean) {deyimler;} else {deyimler;}
switch deyimi switch (integer seçici) { case değer1: deyimler; break ; ... case değeri: deyimler; break ; default : deyimler; }	While ve do-while döngü deyimleri while (şart-boolean) { deyimler; } do { deyimler; } while (şart-boolean);	For döngü deyimi for (başlangıç değerleri, deyimleri; şart-boolean;arttırma deyimi) { deyimler; }

Sıkça kullanılan statik metodlar ve sabitler Math.PI Math.exp() Math.random() Math.pow(a, b) System.currentTimeMillis() System.out.println(anyValue) JOptionPane.showMessageDialog(null , message) JOptionPane.showInputDialog(prompt-message) Integer.parseInt(string) Double.parseDouble(string) Arrays.sort(type[] list) Arrays.binarySearch(type[] list, type key)	Statik tek boyutlu değişkenler int [] list = new int [10]; list.length; int [] list = {1, 2, 3, 4}; statik çok boyutlu değişkenler int [][] list = new int [10][10]; list.length; list[0].length; int [][] list = {{1, 2}, {3, 4}};
---	---

1.0 PROGRAMLAMA DİLİNE GİRİŞ

1.1 NİÇİN JAVA PROGRAMLAMA DİLİ

Java Programlama dili şu anda dünyadaki en popüler programlama dillerinden biri haline gelmiştir. Java SUN bilgisayar şirketince orijinal olarak elektrikli ev araçlarının (mikrodalga fırınları, buzdolapları , televizyonlar, uzaktan kumanda cihazları vs.) birbiriyle haberleşmesini sağlamayı amaçlayan bir proje içerisinde 1991 yılında geliştirilmeye başlandı. Orijinal adı bu dilin yaratıcıları James Gosling, Patrick Naughton, Chis Wartdh, Ed Frank ve Mike Sheridan tarafından **Oak** olarak konulan programlama dili daha sonra bu isimde başka bir programlama dili olduğu keşfedilince o anda bir kahvehanede kahve içen programlama gurubu tarafından kahve markasından esinlenerek **Java** olarak değiştirildi. Akıllı elektronik ev araçları pazarı SUN gurubunun tahminlerinden çok daha yavaş bir gelişme gösteriyordu. Bu yüzden Java dili projesi ticari bir geliştirme projesi olarak büyük olasılıkla iptal edilecekti. 1993 Yılında "World Wide Web" büyük bir atılım göstererek bütün dünyaya yayılmaya başladı. Javanın Dinamik Web sayfaları hazırlamadaki büyük potansiyelini gören SUN şirketi projeyi bu tarafa yönlendirdi ve bu javaya yeni bir canlılık ve yaşama umudu sağladı.

Mayıs 1995 de SUN javayı büyük bir konferansta tanıttı. Program iş dünyası tarafından derhal büyük bir ilgiyle karşılandı. Java Modern bilgisayar dünyasının ses, grafik işlem, haberleşme gibi ihtiyaçlarına cevap verebilen ve Ticari gayeler için hazırlanan bir Program dili olarak daha önceki bilgisayar dillerinin hiç birinin kapsayamadığı özellikleri içermekteydi. Bunun yanı sıra dil komut yapısı olarak C++ diline çok yakın olması da öğrenilmesini kolaylaştırıyordu. SUN Javayı "World Wide Web" de kullanmak isteyen herkese ücretsiz olarak sundu. Java internette yayınlanmasının ardından çok büyük bir patlama yaşadı. 1997 ye gelindiğinde dünyadaki bütün bilgisayar okullarında temel bilgisayar dili olarak gösterilmeye başlandı. Dünyada şu anda hala en çok kullanılan bilgisayar dili olan C++ dilinin yapılan hataları tam olarak denetlenmesi programın çalışma hızını artırma yönünden iyi bir özellik olsa da profesyonel programcılar dışında kullanılmasını sınırlandırıcı bir etki yapıyordu. Java ise bütün hataları bildiren yapısı ve modern bilgisayarın bütün fonksiyonlarına ulaşabilen kütüphaneleriyle programcıların çok daha kolaylıkla öğrenebileceği bir dildir. Burada hemen şunu da belirtelim. C dili hızlı çalışma amacı birinci planda tutularak yaratılmış bir dildir. Javada ise emniyet ilk planda yer almıştır. Hız açısından düşünüldüğünde java C (ve C++) diliyle rekabet edemez. zaten program derleyicisi de C++ dilinde yazılmıştır.

Javanın diğer önemli bir temel özelliği Nesne kökenli (object oriented) bir dil olmasıdır. Nesne kökenli diller, nesnelere gerçek dünyadakine daha benzer bir yapıda tanımlayarak anlaşılmasını kolaylaştırırlar. Nesnelere gerçek dünyadaki gibi masa, sandalye, bilgisayar, gerçek gaz, ısı eşanjörü gibi tanımlayarak programlamak insan beyninin anlaması açısından çok daha kolaydır. Bundan önceki tam nesnel kökenli programlama dillerinden hiçbiri çok yaygınlık kazanmamıştı. Bu tür dillere Smalltalk,ü Örnek verebiliriz. C++ nesnel kökenli programlama yapabilen bir dildir. Fakat yapısal (structural) birprogramlama dili olan C dilinin bir uzantısı olarak geliştirildiğinden tam anlamıyla nesnel kökenli bir dil olduğu söylenemez.

Java dilini geleceğin dili yapan diğer bir özelliği de çok kullanımlı (multi-tasking) ve paralel kullanımlı(multi-treading) bir dil olmasıdır. Çok kullanımlılık birden fazla işlemin aynı anda yapılabilmesinin tanımıdır. Paralel kullanımlılık ise birden fazla programın aynı anda hafızayı beraber kullanarak kullanılabilmesidir. Örnek olarak World ve Excel programlarının Windows NT ortamında aynı anda kullanılmasıdır. ADA gibi bazı eski Program dillerinde çok kullanımlılık programlanabiliyordu. Paralel kullanım olanaklarını sunan ilk bilgisayar dili ise javadır. Paralel kullanım paralel programlama kavramından ayrıdır ve karıştırılmamalıdır. Paralel programlanmada birden fazla Bilgi işlem Ünitesine (CPU) ayrı programlar veya bir programın ayrı parçaları gönderilir. Paralel kullanımda ise bir CPU nun kullanım zamanı küçük parçalara ayrılarak değişik Program veya Program parçacıkları bu zaman paketçiklerini paylaşarak kullanırlar.

Javayı önemli bir Program dili haline getiren en önemli özelliği ise kullanılan bilgisayardan bağımsız olmasıdır. Javada yazılan bir Program Unix, Machintosh, Windows 95 veya Windows NT veya herhangi bir 32 bit makinada hiç değiştirilmeden kullanılabilir. Java programlarının grafikleri "Wold Wide Web" sayfalarının programlama dili olan html (hypertext markup language) ile aktarılır. Bu yüzden html ve java programlarını

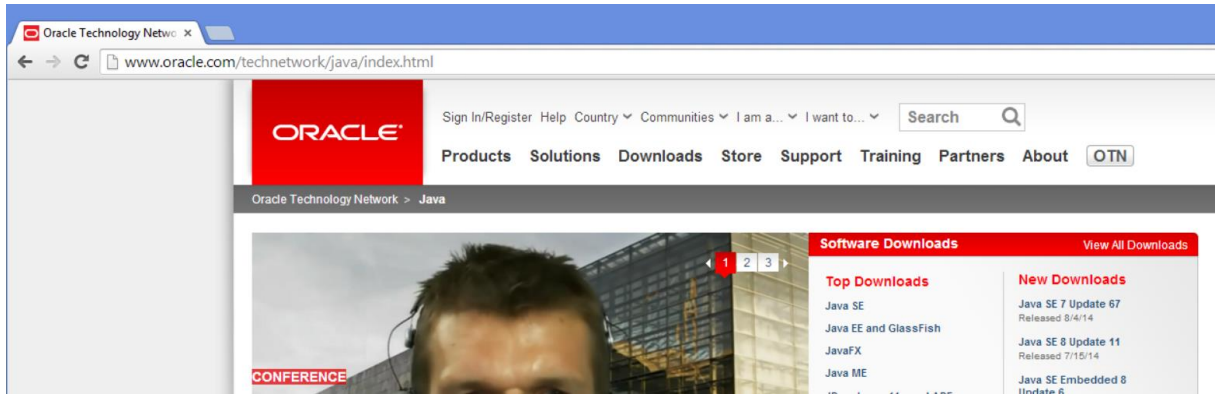
birlikte kullanmak ve java programlarını gerçek zamanda www sayfalarında göstermek mümkündür. Kendi web sayfanızı veya webde yazılmış kitabınızı bütün dünyaya aktarırken yaptığımız analizleri de bu kitabın dinamik bir parçası olarak sunabilirsiniz. Bu Türkiye gibi ülkelerin dünyaya kapılarını daha kolay açmaları açısından oldukça önemli bir özelliktir.

Javanın HTML diliyle kullanılabilmesi, bazı kişilerde htmlin bir parçası olduğu gibi bir kavram gelişmesine yol açmıştır. Programlamaya yeni başlayanlar sık sık html ile javayı birbirine karıştırırlar. veya javanın HTML'nin bir uzantısı olduğunu düşünürler. Bu temelde oldukça yanlış bir varsayımdır. HTML internet belgelerini birbirine bağlamak amacıyla geliştirilmiş bir belge işlem sistemidir ve bir programlama dili değildir. HTML ile javanın tek gerçek ilgisi HTML deki applet komutudur. bu komut yardımıyla java dilinde yazılmış programların sonuçları HTML ortamına dinamik olarak aktarılabilir.

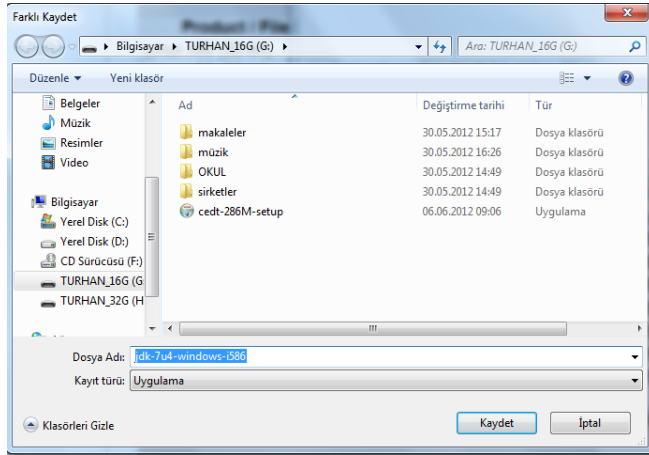
Android giderek yayılan bir işletim sistemi olarak önce telefon gibi sistemlerin temel işletim sistemi olarak ortaya çıktı ve giderek bilgisayar alanlarında yayılmakta. Linux tabanlı bu işletim sisteminde java programlama dili önemli bir yer tutmaktadır. Android uygulamaları java programlama dili kullandığından bu dilin gelişmesinde önemli bir rol oynamaya aday görünmektedir.

1.2 BİLGİSAYARDA JAVA PROGRAMLARINI KURMA

Java programlama diliyle bilgisayar programı yazmaya başlayabilmek için elinizde Java kurma seti (Java Development Kit- JDK) bulunmalıdır. Bu paket Oracle bilgisayar sitesinin adresinden ücretsiz olarak indirilebilmektedir. <http://www.oracle.com/technetwork/java/index.html>



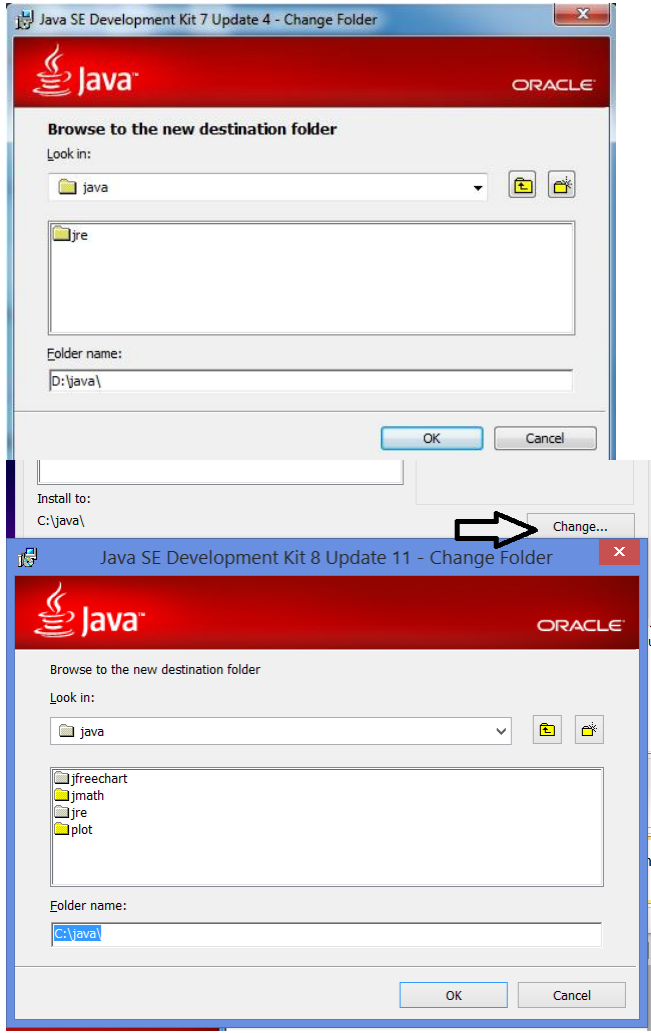
Siteye girdiğinizde önce JAVA SE program paketini seçiniz. Bundan sonra JDJ7.4 sürümünü veya bulunan en yeni sürümü seçebilirsiniz. “Download”(yükle) düğmesine basarak programı yüklemeye başlayabilirsiniz



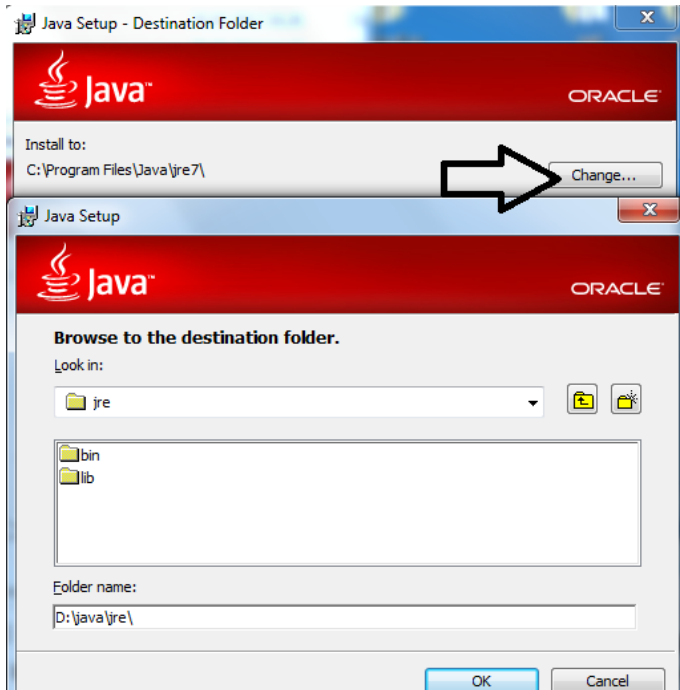
Programı internetten yükledikten sonra çalıştırarak bilgisayarımıza yükleyebiliriz.

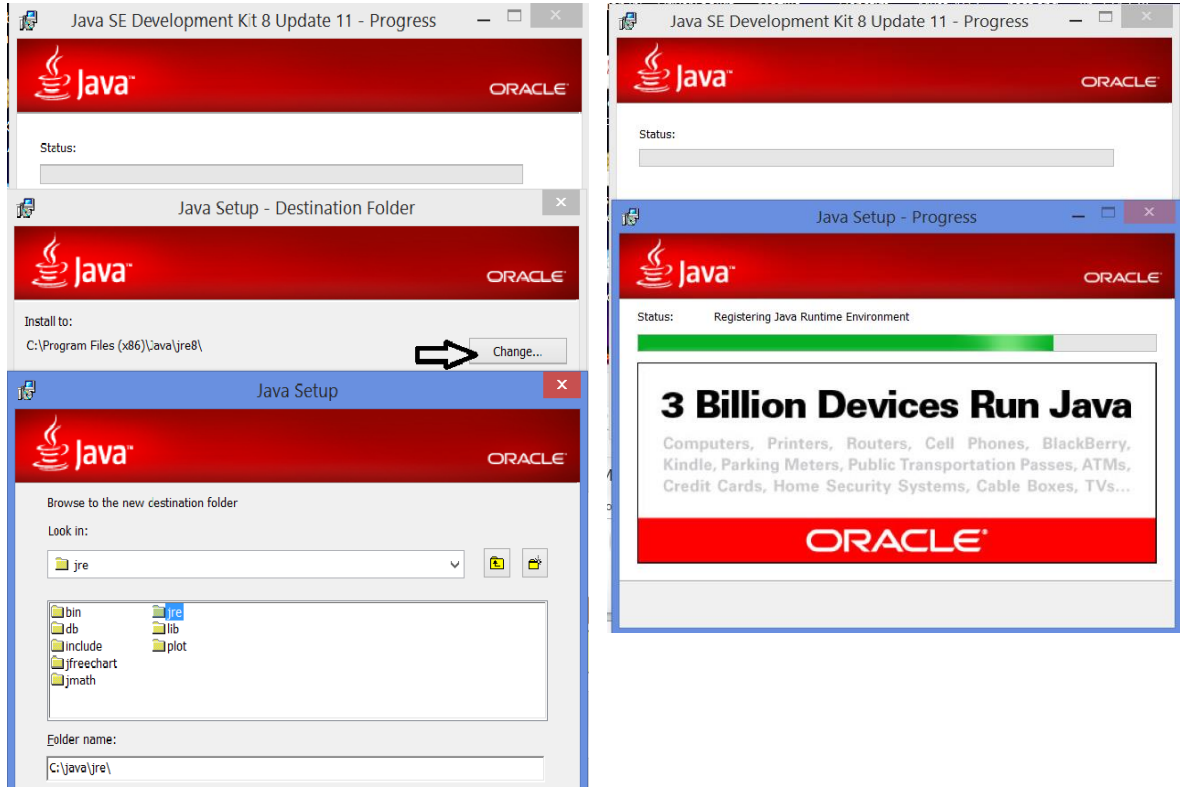


Yükleme esnasında program size yüklenecek dosya sistemini soracaktır. c:\java veya d:\java gibi basit bir dosya sistemi kullanırsanız, daha sonra erişim gerektiğinde erişim daha kolay olabilir.



Daha sonra sistem java çalıştırma paketi (Java runtime environment JRE) yükleme dosya sistemi soracaktır. Bunun için JDK paketini yüklediğiniz yerin altındaki jre dosya sistemini seçebilirsiniz (örneğin d:\java\jre)



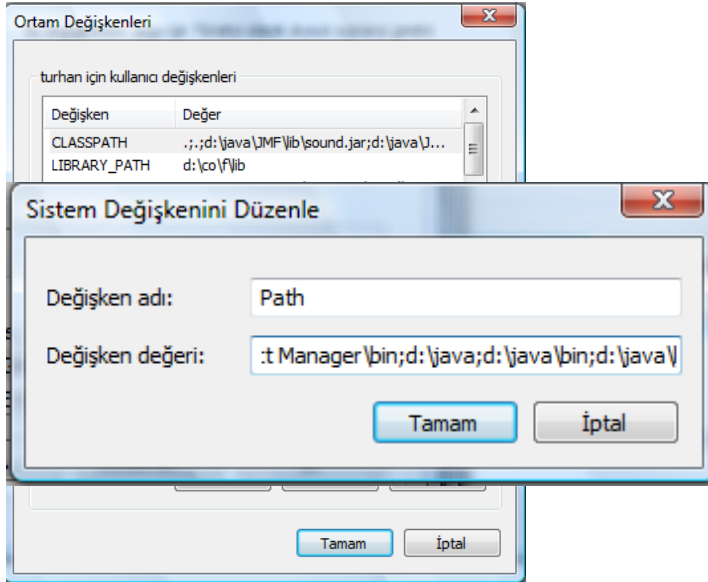
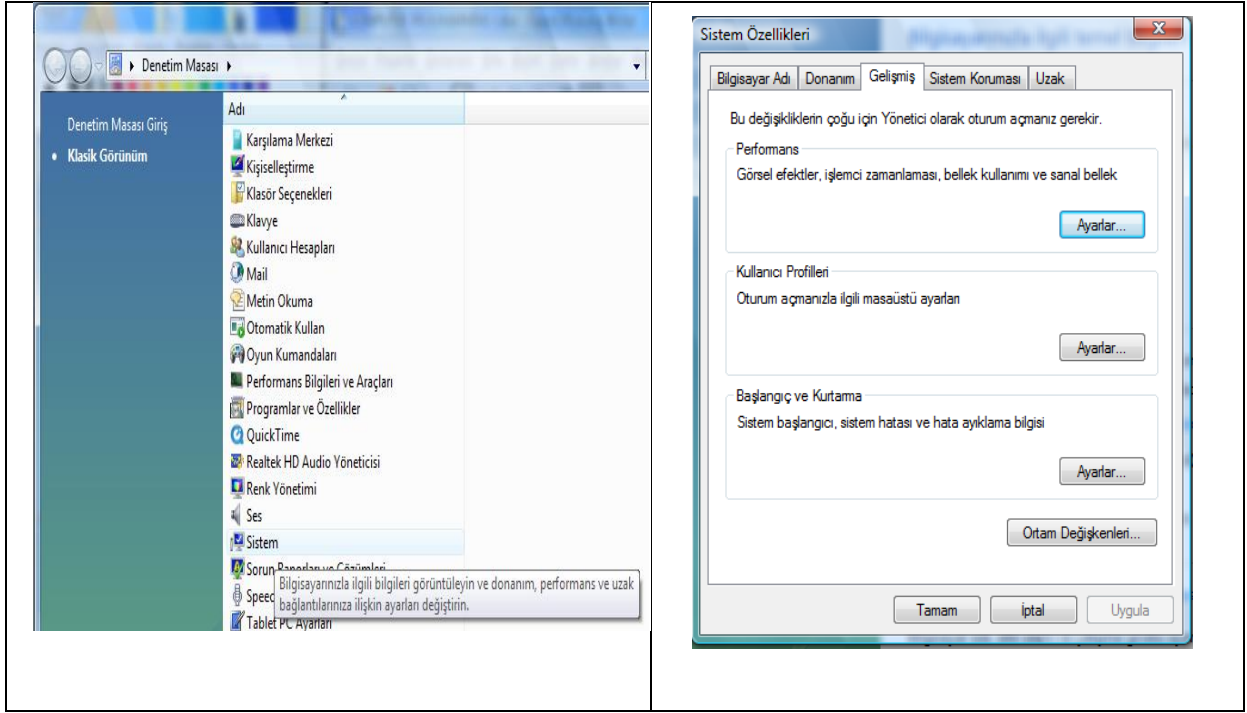


Böylece java programlama paketini bilgisayarınıza yüklemiş oldunuz. Java programlama paketini doğru yükleyip yüklediğinizi kontrol etmek isterseniz

<http://java.com/tr/download/help/testvm.xml>

adresine girerek kontrol edebilirsiniz. Bir sonraki adım yüklediğiniz java paketinin yerini işletim sistemine tanıtmaktır. Bunun için eğer Windows sistemi kullanıyorsanız sistem komutlarına JAVA_HOME deyimine

javayı yüklediğiniz dosya sistemine tanıttıktan sonra java dosyasını ve bin ve lib alt dosyalarını path deyimini ekleyebilirsiniz. Windows sisteminde bu deyimleri eklemek için Denetim masasından sistem (system) dosyasını seçmeli , sistem özellikleri, gelişmiş ve sonra ortam değişkenleri düğmelerine basarak açılan pencereden path deyimine yeni dosya sisteminlerini ekleyebilirsiniz ve yeni JAVA_HOME değişkeni atayabilirsiniz



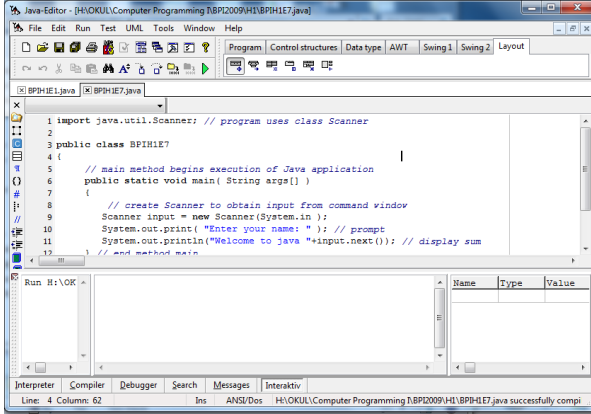
Listeden path deyimini bulunuz ve listenin sonuna örneğin

....;d:\java;d:\java\lib;d:\java\bin;

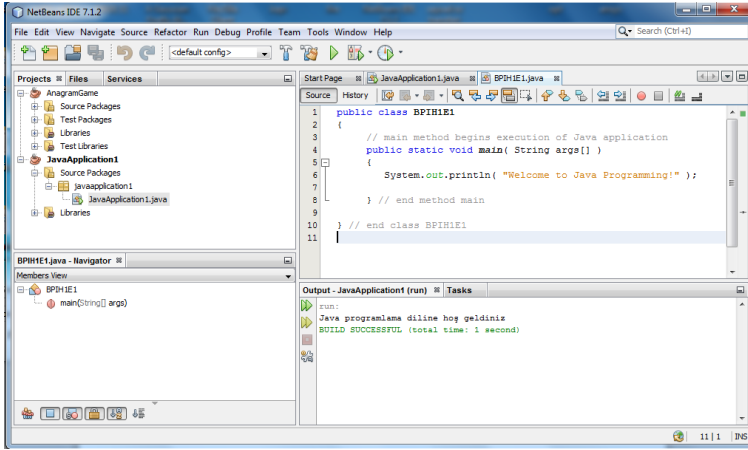
Deyimini ekleyiniz. Bundan sonra bir editör programı kurmanız tavsiye edilir. Java programlama dilini etkin olarak kullanabilen çeşitli ücretsiz editörler mevcuttur. Bunların birkaçını burada tanımlayalım.

Everyone java Editor(<http://sourceforge.net/projects/eje/>) Basit bir java editörü

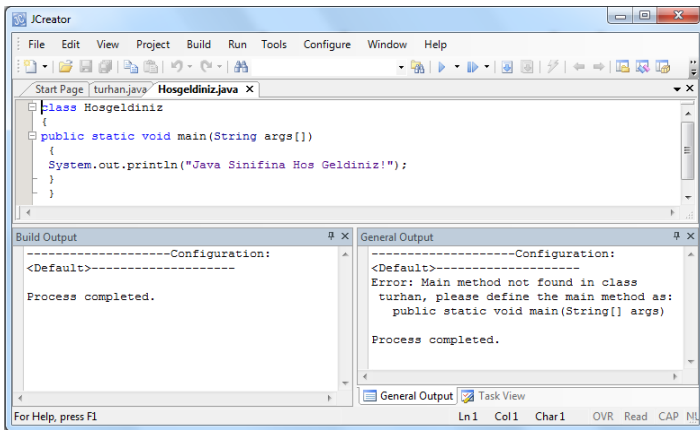
Java-editor (<http://javaeditor.org/index.php/Download>) Direk olarak java dilinde programlama yapmak için oluşturulmuş basit ve kullanışlı bir editör



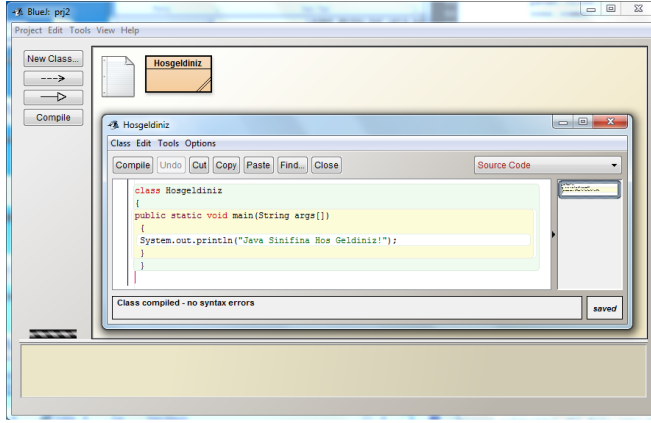
Netbeans : Oldukça gelişmiş bir java geliştirme sistemidir (<http://netbeans.org/downloads/index.html>)



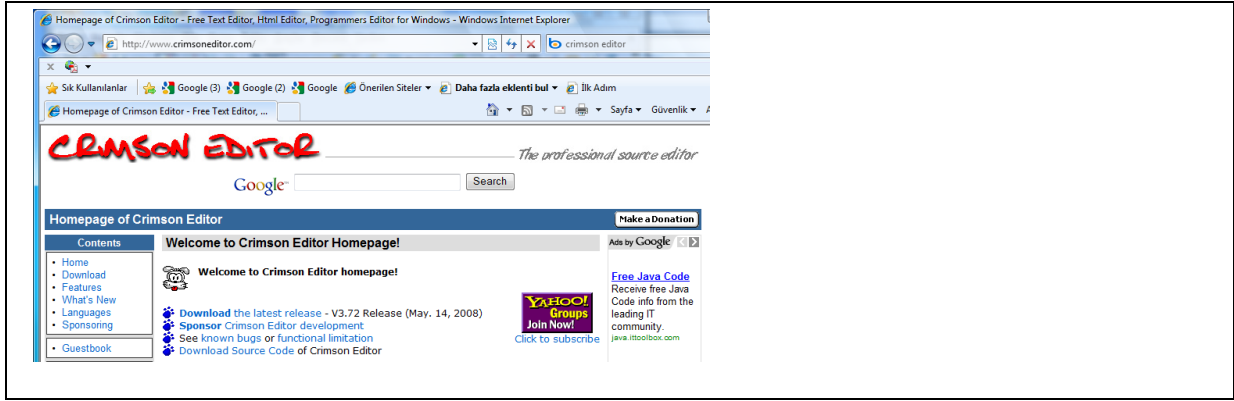
Diğer güzel bir Java editörü de JCreator editörüdür. Bu editör java dilinde program yazmak için dizayn edilmiş, göreceli olarak kullanımı basit bir editördür. Kopyesi <http://jcreator.en.softonic.com/> adresinden alınabilir.



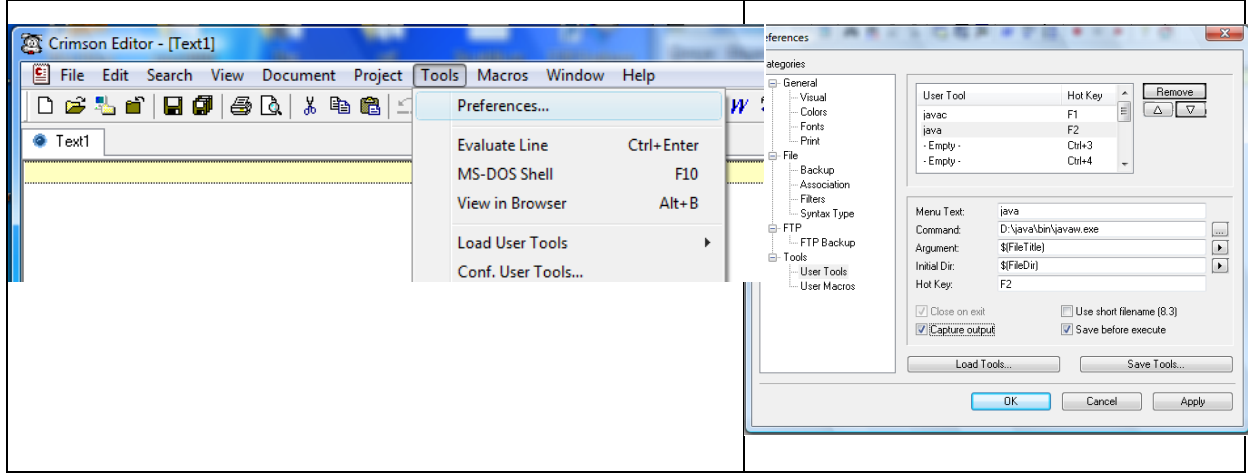
Son günlerde kendinden çok söz ettiren göreceli kullanımı basit, fakat kompleks işlemleri rahatlıkla yapma olasılığı veren diğer bir editörde BlueJ editörüdür. Bu editöre <http://www.bluej.org/> adresinden ulaşılabilir.



Benim burada örneklerde kullanacağım editör basit bir dilden bağımsız (her dili çalıştırabilen) editör olan Crimson Editör olacaktır. Bu editörün bir kopyesini <http://www.crimsoneditor.com/> adresinden sağlayabilirsiniz. İnternette şu anda benim bulduğum en son versiyonu 3.72. yüklenecek dosyanın adı : cedt-241-setup.exe. Bu dosyaya bastığımızda bilgisayarınıza yüklenecektir. Yükleme yaptıktan sonra özellikle java dosyalarını çalıştırmak için bazı küçük ayarlamalar gerekmektedir. Programdan Tools penceresini açınız. Buradan preferences ve user tools düğmelerini seçerek konsol ortamında java dosyalarını çalıştırma yolu olan javac ve java komutlarını aşağıdaki pencerelerde gösterildiği gibi girebiliriz. (burada java d:\java dosya



sistemine yüklenmiştir.) Bunlara ek olarak internet ortamında kullanıma açık olarak bulunan çeşitli java paketleri sistemimize eklenebilir. Bunlara birkaç örnek verelim. Boyutlu grafik uygulamaları için programlarımda visAD isimli bir paket kullanıyorum. Bu paketin internet adresi : <http://www.java2s.com/Code/Java/Advanced-Graphics/GraphLibraryVisad.htm>, kullanırken aynı zamanda java 3D paketine de ihtiyaç gösteriyor. Java 3D paketini <http://java3d.java.net/binary-builds.html>, adresinden edinebilirsiniz. Java3D paketini çalıştırdıktan sonra yüklediğiniz visAD paketindeki visad.jar dosyasını ../jre/lib/ext adresine kopyalayın. Paket çalışmaya hazır hale gelmiş olacaktır.



Excel programlama ortamına yazma ve okuma isterseniz <http://jexcelapi.sourceforge.net/> adresinden jxl.jar dosyasını alarak ../jre/lib/ext dosya sistemine kopyelemeniz, bu programı bizim kodlarımız, veya kendi yazdığımız kodlarla çalıştırmak için kafi gelecektir.

Eğer SQLite veri tabanı kullanmak isterseniz

<http://www.xerial.org/maven/repository/artifact/org/xerial/sqlite-jdbc/3.7.2/>

sitesinde bulunan [sqlite-jdbc-3.7.2.jar](#) dosyasını yine ../jre/lib/ext dosya sistemine kopyelememiz gerekir.

1.3 JAVA PROGRAMINI YAZMA VE ÇALIŞTIRMA

Java programı **isim.java** şeklinde editörle yazılır. Java programınız yazıldıktan sonra ilk step java derleyici (compiler) programıyla **isim.class** ara bilgisayar kodunu oluşturmaktır.

Java programını yazarken dikkat etmeniz gereken diğer bir özellik de javanın nesne sınıfının ismiyle dosyadaki isminin aynı olmasıdır. Bilgisayar konsol ekranında (DOS veya UNIX Console)

javac isim.java

Komutunu bilgisayara gönderdiğinizde programınızda hata yoksa ve bilgisayarınız java komutunun yerini biliyorsa (path deyimi) java derleyicisi **isim.class** dosyasını yaratır. Eğer programınızda hata varsa hataların listesini verir. Programlamanın dili ingilizce olduğundan bu hata açıklamaları da ingilizcedir. Editör programına geri dönüp hataları düzeltmeniz, programı tekrar kaydetmeniz ve sonra bu komutu tekrar denemeniz gerekecektir. Eğer derlerken sınıf isimleri ve değişken isimlerini Türkçe olarak kullanabilmek istiyorsak üstteki deyimi

javac -encoding UTF-8 isim.java

şeklinde değiştirebiliriz. Bu durumda Türkçe program ve değişken isimlerini rahatça kullanabiliriz. Java programları iki ortam için yaratılabilir. Birincisi Konsol ortamıdır (DOS, Unix console gibi). Konsol ortamını kullanırken java dilinin konsol komutu java kullanılarak isim.class dosyası yerel bilgisayarın makine diline program çalışırken dönüştürülür. Bu temelde bir tercüme (interpretation) işlemidir. Sonuç olarak java dilinde iki kademeli bir proses kullanılır, birinci işlemde java dosyası class dosyasına çevrilir (derlenir), ikinci işlemde ise class dosyası yerel bilgisayarın makine diline dönüştürülür. konsol ortamında çalıştırılan program sonuçları yine konsol ortamında alınacağı gibi pop-up pencere grafik ortamında da alınabilir. Veya yine grafik ortamı olarak çıktısı internet browser ortamında direk olarak gösterilebilecek şekilde de yazılabilir. Bu tür grafik java programlarına applet adını veriyoruz. Eğer Program konsol ortamında yazılmışsa

java isim

komutu kullanılarak sonuçlar ekranda veya açılan bir grafik penceresinde görülebilir. Eğer Program applet ortamında yazılmışsa java programına ilave olarak html dilinde **isim.html** dosyası açılır. En basit şekilde **html** dosyası şu şekilde tanımlanabilir :

```
<html >
<applet code="isim.class" width=200 height=50 >
</applet >
</html>
```

Buradaki width pencerenin enini pixel olarak, height'da yüksekliğini yine pixel(görüntü noktası) olarak vermektedir. daha sonra bu dosya konsoldan

appletviewer isim.html

komutuyla çalıştırılabilir. Buradaki appletviewer komutu java program paketindeki basitleştirilmiş bir browser programıdır. Bu komut yerine isim.html dosyası herhangi bir browser programına yüklenerek görülebilir. Programcılıkta adeta adet haline gelen ilk uygulama daima o dilde ekrana Hosgeldiniz çıktisi almaktır. Bizde bu kuralın dışına çıkmayalım. Aşağıdaki listede ilk konsol programımız **Hosgeldiniz.java** görülmektedir.

Program 1.1 [Hosgeldiniz.java](#) konsol programı

```
class Hosgeldiniz
{
public static void main(String args[])
{
System.out.println("Java Sinifina Hos Geldiniz!");
}
}
```

bu programı çalıştırmak için

javac Hosgeldiniz.java
java Hosgeldiniz

komutlarını konsolda vermemiz yeterlidir
konsol çıktısı olarak:

Java Sinifina Hos Geldiniz!

Yazısını alırız. Veya programımızı crimson editörde yazmış isek ve F1 kısa tuşunu javac komutunu yüklemek için F2 kısa tuşunu da java komutunu yüklemek için kullanmış isek ve capture output tuşunu işaretlemişsek: F1 ve F2 tuşlarına basarak program çıktımızı şekildeki gibi elde ederiz.

Şimdi bu program yapısına biraz göz göz atalım.

// işaretiyle başlayan yazılar bilgisayar tarafından kullanılamaz sadece programı okuyanı bilgilendirmek amacı taşır /* */ işaretleriyle de verilebilir. Örneğin :

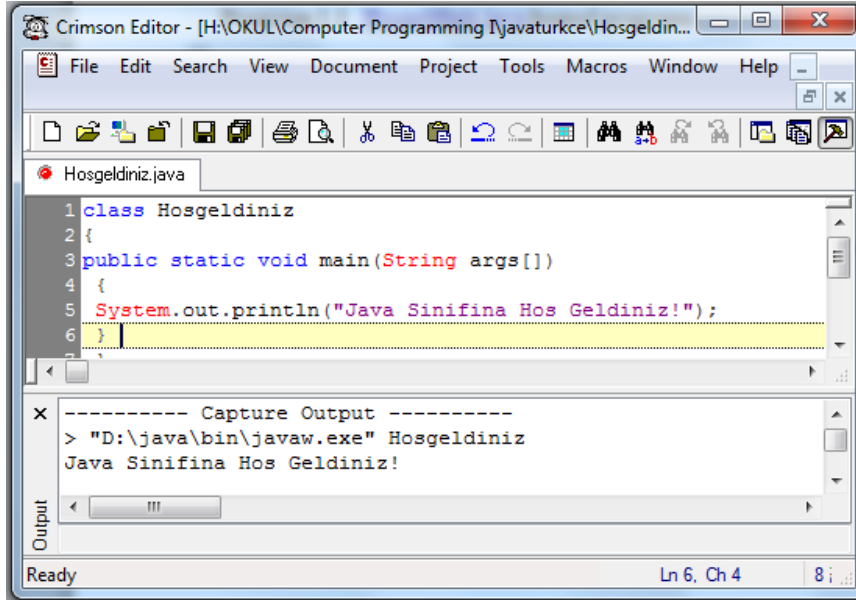
```
/* Bu kısımda yazılanlarla bilgisayar ilgilenmez
Sadece insanların programı anlaması içindir
*/
```

class Hosgeldiniz Class (sınıf) nesne kökenli programlama yapısının temel taşıdır. Her Class'ın kendine özgü bir ismi olması gerekir. Daha önce yazılmış Class'lar **import** deyiimiyle programa çağırılabilirler. Yeniden

yazılmaları gerekmez. Class yapısı ve tanımlamaları üzerinde daha sonraki bölümlerde çok daha detaylı bilgi verilecektir.

{ } başlatma ve bitirme : Sınıf tanımlaması da dahil olmak üzere Tüm Program parçacıkları { işaretiyle başlar ve } işaretiyle son bulur. {} işaretleri arasında yazılan program parçacıklarını normal metin yazılmasındaki paragraf kavramına benzetebiliriz. Ayrıca program koduna baktığımızda ";" işaretine sık sık rastlıyoruz. Bu işaret java komutunun bittiğini belirtir. Cümle yağısında kullanılan nokta "." işareti gibi işlev görür. Java programlarını istersek ";" ve "{}" ayraçlarını kullanarak tek bir satır olarak yazabiliriz. Fakat bu programın bizim tarafımızdan okunurluğunu zorlaştırır. Program kodunu yazarken boşluk ve satır aralıkları bırakarak insanlar tarafından programın okumasını kolaylaştırabiliriz. Modern programcılıktaki en büyük sorunlardan biri bilgisayar kodlarının bakım ve onarımı sorunudur. **Ticari olarak yazılan her programın belli bir ömrü olduğunu ve daha sonra başka bir programcı tarafından üzerinde tekrar çalışılacağını sürekli olarak hatırlamak zorundayız.**

Şekil 1.1 Hosgeldiniz.java programının ve consol çıktısının Crimson editör ortamında görünümü



public static void main(String args[]) diğer dillerde fonksiyon (function), subroutine, procedure gibi adlarla anılan Program alt parçacığının başlığı. Bu Program parçacıkları javada Metot (Method) adıyla anılır. Metotların girdileri ve çıktıkları olabilir. Bu metotun adı **main** (ana Program) dır. String değişken türü args[] değişkeni girdi olarak girebilir. Bu metotta **void** çıktı türü kullanılmıştır. Void kelimesi bu metotdan hiçbir çıktı alınmayacağını belirtir. **public** metota dışarıdan ulaşılabilme izni verir.

System.out.println("Java Sinifina Hos Geldiniz!"); Bu satır System.out sınıfından println metotunu çalıştırır. Bu metot method girişinde gönderilen String tipi değişkeni konsol çıktısına aktarır. Fortran dilindeki write(*,*) veya print*, c dilindeki printf, c++ dilindeki cout<< terimlerinin javadaki tam karşılığıdır.

"Java Sinifina Hos Geldiniz!" stringi aynen aktarılırken içinde c dilinde kullanılan temel kontrol terimleri yer alabilir ve aynen c dilindeki anlamlarıyla kullanılabilir. Bu kontrol terimlerinden en yaygın olarak kullanılan bazıları:

\n alt satıra geç
\t bir sonraki tab'a atla
\r gir tuşu bas
\ \ işaretini yaz
\” “ işaretini yaz

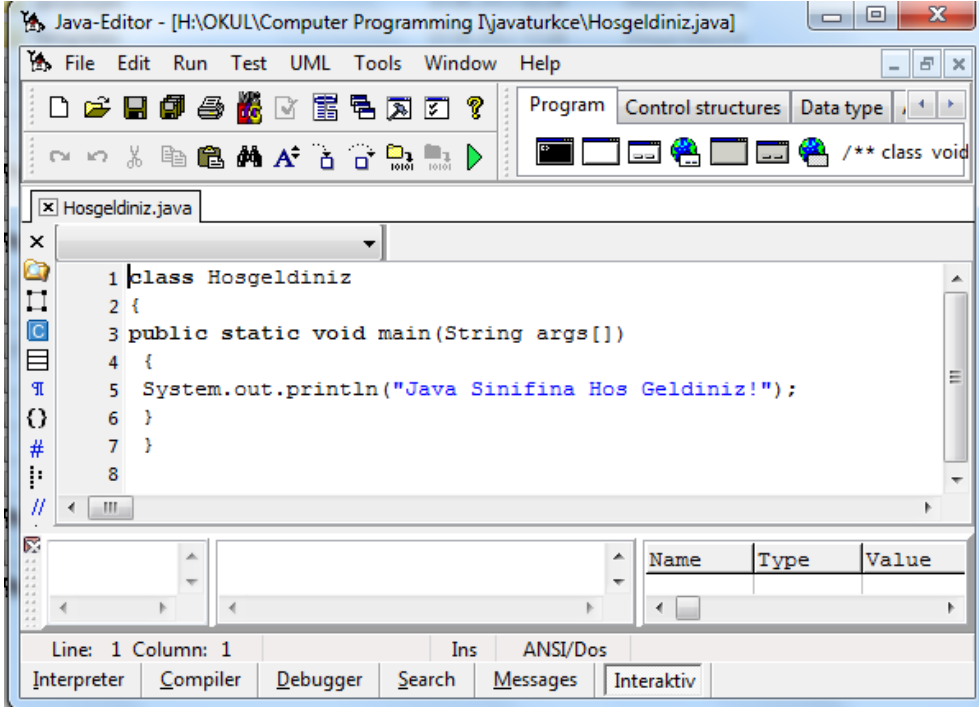
şeklinde gösterilebilir.

Bu ilk programımızı göstermek amacıyla yukarıda gadi geçen diğerk iki editör programını ile de çalıştırılm

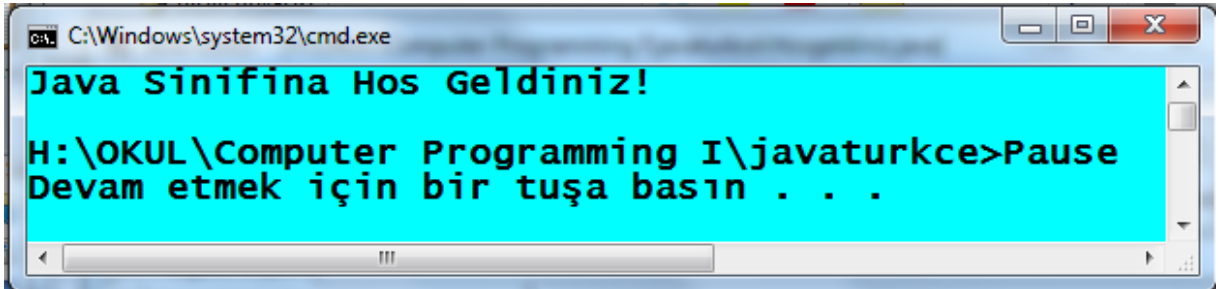
İkinci konsol programımız HosgeldinizSW.java Program 1.2 de verilmiştir. Bu program SWING kütüphanesinin alt kütüphanesi olan JOptionPane kütüphanesi kullanılarak yaratılmıştır. Çıktı yazısı ekranda grafik pencere olarak belirir ve Java sınıfına Hos geldiniz! mesajını verir. Tüm bu kütüphanelerin detayları ilerideki bölümlerde verilecektir. Şimdilik nasıl kullanıldıklarını öğrenmekle ve programlarımızı örnek programlara benzeterek yazmaya çalışacağız. Uzman programcılar isterlerse ilgili bölümlerden komut detaylarına derhal girebilirler.

Programı java editör ortamında çalıştırmak istersem, önce dosyayı editörde yazar (vaya yazılmış dosyayı yükler) ve çalıştır düğmesine basarız. Sonuç konsol ortamında verilir.

Şekil 1.2 Hosgeldiniz.java programının java-editor ortamında görünümü

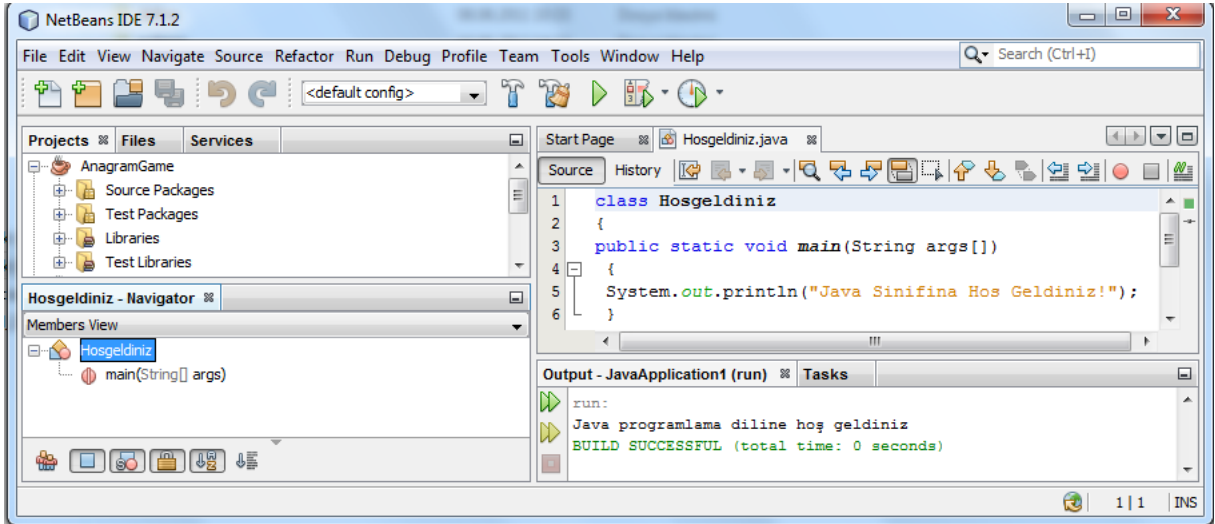


Şekil 1.3 Hosgeldiniz.java programının çıktısının (java-editor) konsol ortamında görünümü

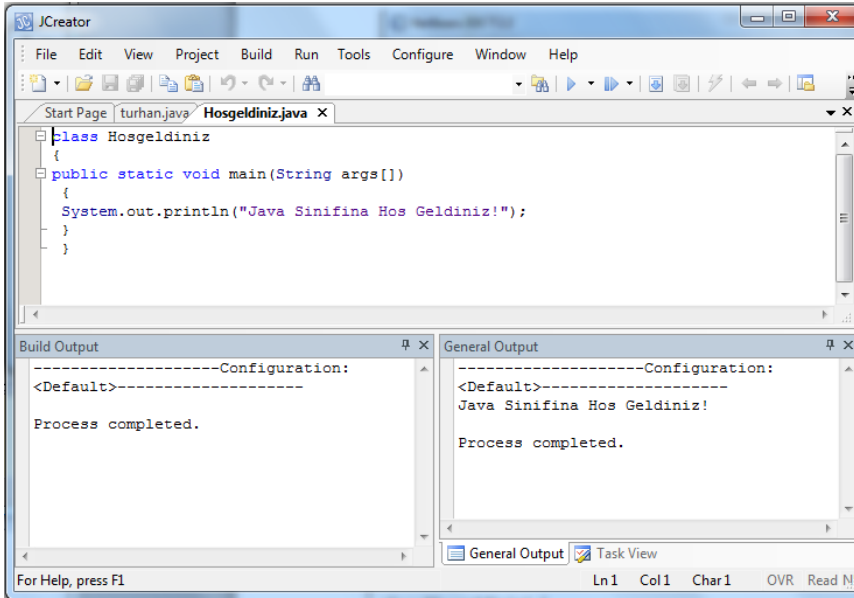


Şimdi aynı programı netbeans ortamında çalıştırılm

Şekil 1.4 Hosgeldiniz.java programının ve consol çıktısınınNetBeans editör ortamında görünümü



Aynı programı JCreator da çalıştırsak



Bu ön bilgidenden sonra java ortamındaki girdi çıktı değişken ve ortamlarını tanımlamak amacıyla değişik Hoşgeldiniz programlarını tanımlayalım:

Program 1.2 Hosgeldiniz1.java programı

```
import javax.swing.JOptionPane;

class Hosgeldiniz1 {
    public static void main (String args[])
    {JOptionPane.showMessageDialog(null,"Java sınıfına hoş geldiniz!");}
}
```

Bu programı

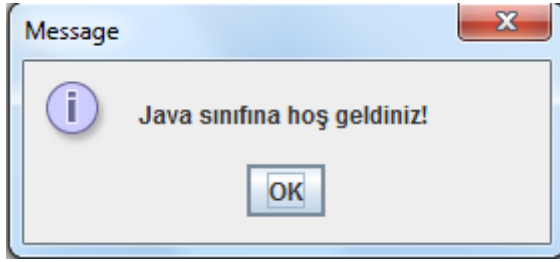
```
javac HosgeldinizSW.java
```

ve

```
java HosgeldinizSW
```

komutlarını kullanarak veya editörde çalıştırdığımızda

Şekil 1.5 Hosgeldiniz1 dosyasının konsol SWING pop-up grafik çıktısı kullanarak görülmesi



Çıktısını alırız. Bu çıktıyı alabilmek için java SWING kütüphanesi içerisinde yer alan **JOptionPane.showMessageDialog(...)** metodunu kullandık. Bu metodu programımızda çağırabilmek için de **import javax.swing.JOptionPane;** yapısını kullandık. Buradaki import kütüphanedeki javax program paketini çağırır. Bu metodun görevi System.out.println metoduyla aynıdır. Farkı ise grafik ortamda çıktı vermesidir. Programdan da da görüleceği gibi sınıfımızın (class) adı Hosgeldiniz1 dir. Bir önceki programdaki gibi bu programda kullandığımız komutlar **public static void main(String args[])** komutuyla tanımlanan ana metodun içinde yazılmıştır. Şimdi de JApplet grafik ortamında bir çıktı çağırılım:

Program 1.3 Hosgeldiniz2.java programı

```
import javax.swing.*;
public class Hosgeldiniz2 extends JApplet
{
public void paint(Graphics g)
{ g.drawString("Java sınıfına hoş geldiniz!",10,10);}
}
```

Görüldüğü gibi bu programın içinde main metodu bulunmamaktadır. Çıktı paint metodunun içinde verilmiştir. Bu programı internet ortamına html dosyası ile aktarıyoruz.

Program 1.4 Hosgeldiniz2.html programı

```
<html >
<applet code="Hosgeldiniz2.class" width=200 height=50 >
</applet >
</html>
```

Burada yazıyı temel olarak bir grafik metodu olan

```
g.drawString("Java sınıfına hoş geldiniz!",10,10);
```

metodunu kullanarak pencere ortamına aktardık. bu metotta kullanılan 10,10 deyimini yazının grafik ortamındaki x ve y koordinatlarını göstermektedir. y koordinatı yukardan aşağı doğru tanımlanmıştır.

html kodunda

```
<applet code="HosgeldinizApplet.class" width=200 height=50 >
```

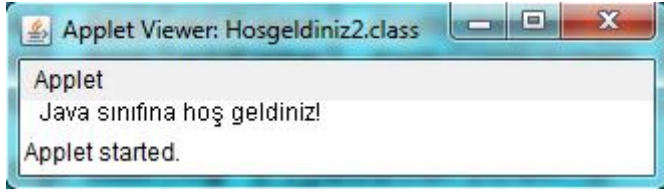
komutu html'e compile edilmiş olan **HosgeldinizApplet.class** dosyasının gösterilmesini istemektedir. Şüphesiz aynı komut çok daha kompleks bir html dosyasının içinde de aynı görevi görmek amacıyla kullanılabilir.

Programı çalıştırmak için

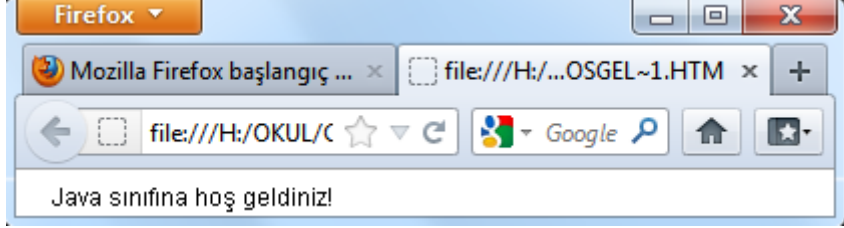
```
javac Hosgeldiniz2t.java
appletviewer Hosgeldiniz2.html
```

komutlarını konsolda (veya editör ortamında tanımlayarak) vermemiz yeterlidir. Elbette Hosgeldiniz2.html dosyasının yukarıda belirtildiği gibi hazırlanmış olması da gerekir. Fakat Konsol programlarından farklı olarak sonucu konsol yerine Applette (browser programında) görürüz. Aynı sonucu **appletviewer** deyimini kullanmadan herhangi bir browser programı çağırarak da elde edebiliriz. Browser programında sonucu görebilmek için önce programı açmamız sonra da HosgeldinizApplet.html dosyasını bularak açmamız gerekir. Çıktı görüntüsü aşağıdaki gibi olacaktır.

Şekil 1.6 Hosgeldiniz1.html dosyasının Appletviewer programıyla gösterilmesi



Şekil 1.7 Hosgeldiniz1.html dosyasının Firefox browser programıyla gösterilmesi



```
import javax.swing.*;
```

```
import java.awt.*;
```

deyimleri java kütüphanesindeki JApplet ve pencere(swing sınıfı) sınıf grafik programlarını programımıza aktarır.

public class Hosgeldiniz2 extends JApplet deyimi Hosgeldiniz2 isimli sınıfı (class) yaratır. Bu sınıf tüm Applet sınıfının bir devamı olarak tanımlanmaktadır (**extends JApplet**) diğer bir deyimle JApplet sınıfının bütün metotları extends deyimleriyle HosgeldinizApplet sınıfımızın içine kopyalanmıştır.

public void paint(Graphics g) deyimi appletin grafik olarak çizilmesini sağlar. Bu Metot **Graphics** sınıfı **g** nesnesini girdi olarak kullanır. Metotun çıktısı yoktur.

g.drawString("Java sınıfına hoş geldiniz!",10,10); Bu terim Appletin x =10, y=10 noktasına **Java sınıfına hoş geldiniz!** Yazısını çizer. DrawString Grafik sınıfı **g** nesnesine ait bir metottur.

Bu programda temel grafik kütüphanesi olarak java swing kütüphanesi kullanılmıştır. ayrıca konsol programlarından farklı olarak bu programda main metodunun kullanılmadığına da dikkat ediniz. bir program hem applet hem de konsol olarak da oluşturulabilir, yani hem main metodu olabilir hem de applet (html) ortamında çalışabilir.

Aynı işi yapmak için bir çok değişik program yazdık. Bazılarımız buna ne gerek vardı, tek bir şekilde bu işi halledemezmi idik diyebilirler. Haklıdırlarda. Fakat modern bilgisayar dillerinde kullanıcıları da göz önüne almak durumundayız. Ve daha burada belirtmediğimiz pek çok farklı şekillerde de yapabiliriz. Aynı işi daha renkli ve çeşitli şekillerde yapmamız temel olarak program kullanıcılarına daha çeşitli alternatifler sunabilir. Aynı zamanda diğer programlama alternatiflerine göre daha üstün bir çalışma ortamı yaratabilir.

Şimdi de programımıza bir basamak daha ekleyelim ve program kullanıcıları isimlerini sorsun, yani ekrandan bir veri girelim. Bunu yapmak için ilk tanımlayacağımız program konsol ortamında çalışan **Hosgeldiniz3.java** programı olacaktır. Bu programın çıktısı dos konsol ortamında şekil 1.6 da görülmektedir. Program ekranda isminizi giriniz yazacak, ve ekranda isim girildikten sonra

```
java sınıfına hoş geldiniz! isim
```

çıkıtısını verecektir.

Program 1.9 [Hosgeldiniz3.java](#) programı

```

import java.io.*;

class Hosgeldiniz3 {
    public static void main (String args[]) throws IOException
    {
        BufferedReader cin=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("isminizi giriniz : ");
        String isim=new String(cin.readLine());
        System.out.println("Java Sinifina Hos Geldiniz! "+isim);
        cin.close();
    }
}

```

Şekil 1.8 Hosgeldiniz3.java dosyasının sonuçlarının dos çıktısı kullanarak görülmesi

```

=====
C:\co\java\prog>java Hosgeldiniz2
isminizi giriniz :
Turhan
Java Sinifina Hos Geldiniz! Turhan
C:\co\java\prog>
=====

```

Bu programda okuma işlemini yapma için önce

```

    BufferedReader cin=new BufferedReader(new InputStreamReader(System.in));
Deyimini kullanarak bir okuma kanalı açtık, sonra
    System.out.println("isminizi giriniz : ");
    String isim=new String(cin.readLine());

```

Deyimindeki **cin.readLine()** komutuyla ismi cin kanalından okuyarak isim değişkenine aktardık.

Şimdi de swing kütüphanesindeki **JOptionPane.showInputDialog()** kullanarak girdi penceresi açan ve çıktı penceresine **JOptionPane.showMessageDialog()** kullanarak mesajı yazan **Hosgeldiniz1SW.java** programını göreceğiz.

Program 1.10 [Hosgeldiniz1SW.java](#) programı

```

import javax.swing.JOptionPane;
class Hosgeldiniz4
{
    public static void main (String args[])
    {
        String isim=JOptionPane.showInputDialog("Lutfen isminizi giriniz :");
        JOptionPane.showMessageDialog(null,"Java Sinifina Hos Geldiniz! "+isim,
        "Hosgeldiniz1SW programi",JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}

```

Bu programı çalıştırmak içinde

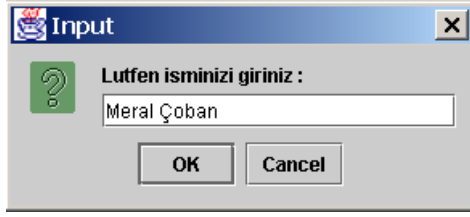
```
javac Hosgeldiniz4.java
```

komutuyla bu programı derlememiz ve

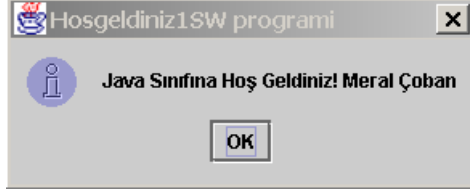
```
java Hosgeldiniz4
```

komutuyla çalıştırmamız gerekecektir. Program girdi ve çıktı pencereleri Şekil 1.9ve Şekil 1.10 da verilmiştir.

Şekil 1.9 Hosgeldiniz4.java dosyasının pencere popup girdi penceresi



Şekil 1.9 Hosgeldiniz4.java dosyasının çıktı penceresi



Şekil 1.8 Hosgeldiniz1SW.java dosyasının pencere popup çıktı penceresi

JOptionPane.INFORMATION_MESSAGE komutu ise bilgi (information) mesajı penceresi açılacağını belirtmiştir. (grafikteki i harfi – information – bilgi kelimesinin baş harfidir). Bu alanda kullanabileceğimiz diğer alternatifler şunlardır :

- JOptionPane.ERROR_MESSAGE : hata mesajı
- JOptionPane.WARNING_MESSAGE : uyarı mesajı
- JOptionPane.QUESTION_MESSAGE : soru mesajı
- JOptionPane.PLAIN_MESSAGE : sade (ikonsuz) mesaj

1.4 TAMSAYI VE GERÇEK SAYI DEĞİŞKENLER VE MATEMATİK İŞLEMLER

Programlama dillerinde rakamlar bilgisayar belleğinin temel depolama birimlerine yazılırlar. Temel bilgisayar bellek birimi bit olarak adlandırılır. Bir bilgisayar belleğindeki tek bir transistörden oluşmuştur. Bu transistörden akım geçiyorsa transistörün veya bitin bellek değeri 1 (bir) veya true (doğru) olarak alınır. Eğer akım geçmiyor veya düşük düzeyde bir akım geçiyorsa transistörün bellek değeri 0 (sıfır) veya false(yanlış) olarak alınır. Bilgisayar bit birimleri bir araya gelerek bilgisayar temel değişken türlerini yazabileceğimiz bir sistem oluştururlar. Değişken türü bilgisayar tarafından bilinmelidir, çünkü aynı bit topluluğu bir harfi simgeleyebileceği gibi bir rakamı da simgeleyebilir. Java dilinde Tablo 1.1 de görülen temel değişken türleri mevcuttur.

Tablo 1.1 Java Temel Değişken türleri

Değişken Türü	Türkçe karşılığı	Bit büyüklüğü	Sınır Değerleri
boolean	Mantık değişkeni	1	true(doğru) , false(yanlış)
char	harf değişkeni	16	'\u0000' den '\uFFFF'
byte	tam sayı değişkeni	8	-128 den 127 e
short	tam sayı değişkeni	16	-32768 den 32767 e
int	tam sayı değişkeni	32	-2157483648 den 2147483647 e
long	tam sayı değişkeni	64	-9223372036854775808 den 9223372036854775808 e
float	Gerçek sayı değişkeni	32	-3.40292347e+38 den 3.40292347e+38 e
double	Gerçek sayı değişkeni	64	-1.7976931348623157e+308 den 1.7976931348623157e+308 e

Şimdi bu temel değişken türlerini daha detaylı inceleyelim:

Tam sayı deęişken türleri (byte, short,int,long)

Tam sayı deęişkenler hafızada işgal ettikleri yere göre byte(8 bit), short(16 bit), int(32 bit) ve long(64 bit) adını alırlar. Bir bitlik hafızaya sadece iki rakamın (0 veya 1) yazılabileceęi göz önüne alınırsa örneğın sekiz bitlik byte türü tamsayı deęişkenine 256 sayı (ikili sayı eşiti 11111111) yazılabileceęi ortaya çıkar. Bitlerden biri + veya - işareti için kullanıldığından byte deęişkeninin sınır deęerleri -128den 127 e kadardır (toplam 256 sayı). Tamsayı deęişken türleri içinde en fazla kullanılan int türüdür. Aşağıdaki küçük program parçacığında int türü **ilktamsayi** deęişkenine 32 rakamı yüklenecektir.

```
int ilktamsayi;  
ilktamsayi=32;
```

bu iki satırlık program parçacığı tek bir satır olarak

```
int ilktamsayi=32;
```

şeklinde de yazılabildi. Javadaki tamsayı deęişken türleri + ve – deęeri alabilen türlerdir. Örneğın byte deęişken türü –128 den +127 e kadar toplam 256 sayı kullanabilir. Burada C ve C++ gibi dillerde mevcut bulunan unsigned deyiminin (byte deęişkenini 0 256 arasında tanımlayabilir) javada mevcut bulunmadığını belirtelim. 15 sayısını bilgisayar ortamında byte türü deęişken olarak oluşturmak istersek her bir bitin bilgisayardaki konumu tablodaki gibi oluşur :

+/-	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
	64	32	16	8	4	2	1
0	0	0	0	1	1	1	1

Gerçek sayı deęişken türleri (float, double,long double)

Gerçek sayı sistemleri de 0 ve 1 bitleri kullanılarak oluşturulabilir. Gerçek sayıların tamsayıdan küçük kısımları 2 li tabanda eksi üstler kabul edilerek oluşturulur. Örnek olarak 16 bitlik gerçek bir sayıyı ikili sistemdeki bilgisayar belleğinde şu şekilde gösterebiliriz :

Tablo 1.4.1 gerçek sayı 22.625e-17 yazılmasında kullanılan onlu ve ikili sayıların bellekte gerçek sayı organizasyonu örneęi

+/-	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	+/-	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
+	16	8	4	2	1	0.5	0.25	0.125	0.0625	-	16	8	4	2	1
0	1	0	1	1	0	1	0	1	0	1	1	0	0	0	1

Yukarıda 16 bitle temsil edilen sayı : +16+4+2+0.5+0.125)*10⁻⁽¹⁶⁺¹⁾ = 22.625e-17 rakamıdır. Gerçek sayı deęişkeninde yeterli hassasiyeti sağlayabilmek için genelde 64 bit uzunluğundaki **double** deęişken türü kullanılır. Javadaki Matematik kütüphaneleri de double deęişken türü için tanımlanmıştır. Aşağıdaki küçük program parçacığında **double** türü **ilkgerçekdeęisken** deęişkenine 22.625e-7 sayısını yüklüyoruz.

```
double ilkgerçekdeęisken;  
ilkgerceldeęisken=22.625e-17;
```

eđer 64 bit double sayı hassaslığı yetmezse **long double** kullanılabilir. Bu deęişken türü 128 bit boyutundadır.

Javadaki tamsayı deęişkenler için tanımlanan aritmetik işlemler şunlardır:

Javadaki temel aritmetik işlemler şunlardır :

- + toplama
 - çıkarma
 - * çarpma
 - / bölme
 - % tamsayılar için bölmede kalan (modül)
- Örneğın bir toplama işlemi:

```
int degisken1=3;
int degisken2=6;
int toplam=degisken1+degisken2;
```

Şeklinde yapılabilir. Örnek programımızda iki integer tam sayıyı grafik ortamından okunduktan sonra toplanarak sonuçlar yazılacaktır. Grafik ortamında rakam okumak için **JOptionPane** sınıfı kullanılmıştır. Okuma eylemi **showInputDialog** deyiimiyle yapılmaktadır.

Program 1.4.1 BPIH1E5.java programı

```
import javax.swing.*;

public class BPIH1E5
{
    // main metodu
    public static void main( String args[] )
    {
        int number1; // birinci integer
        int number2; // ikinci integer
        int toplam; // toplam rakam
        number1 = Integer.parseInt(JOptionPane.showInputDialog("ilk tamsayısı gir:"));
        number2 = Integer.parseInt(JOptionPane.showInputDialog("ikinci tamsayısı gir:"));
        toplam = number1 + number2; // sayıları topla
        JOptionPane.showMessageDialog(null,"Toplam =" +toplam ); // toplamı göster
    } // main metodu sonu
} // BPIH1E5 sınıfı sonu
```

Gerçek sayı değişkenlerin aritmetik işlemleri tam sayı değişkenlerin aritmetik işlemlerine benzer. Tek farkı kalan işleminin bulunmamasıdır. Üstte tamsayı olarak yaptığımız toplama işlemini gerçek sayılar için yinelersek :

Program 1.4.2 BPIH1E5A.java program

```
import javax.swing.*;

public class BPIH1E5A
{
    // main metodu
    public static void main( String args[] )
    {
        double number1; // birinci gerçek sayı
        double number2; // ikinci gerçek sayı
        double toplam; // toplam rakam
        number1 = Double.parseDouble(JOptionPane.showInputDialog("ilk gerçek sayıyı gir:"));
        number2 = Double.parseDouble(JOptionPane.showInputDialog("ikinci gerçek sayıyı gir:"));
        toplam = number1 + number2; // sayıları topla
        JOptionPane.showMessageDialog(null,"Toplam =" +toplam ); // toplamı göster
    } // main metodu sonu
} // BPIH1E5A sınıfı sonu
```

Şimdi de bir fonksiyon hesabı yapalım $y=f(x)=x^2+2x-2$ verilmiş olsun. Ekrandan girilecek bir x rakamı için y değerini hesaplayalım

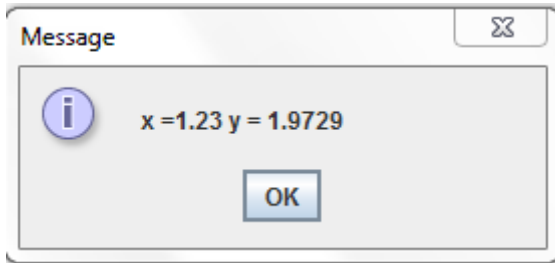
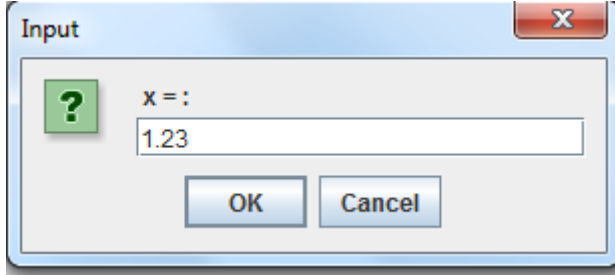
Program 1.4.3 BPIH1E5A.java program

```
import javax.swing.*;
```

```

public class BPIH1E6
{
    // main metodu
    public static void main( String args[] )
    {
        double x = Double.parseDouble(JOptionPane.showInputDialog(" x = :"));
        double y=x*x+2.0*x-2;
        JOptionPane.showMessageDialog(null," x =" +x+" y = "+y ); // sonucu göster
    } // main metodu sonu
}

```



Java ve C dillerinde temel aritmetik işlemlerinin yanı sıra bu işlemlerin kısaltılmış sembolleri de oldukça sık kullanılır. Bu işlemler :

++ bir ilave et
 -- bir çıkar
 += sağ taraftaki rakamı ilave et
 -= sağ taraftaki rakamı çıkar
 *= sağ taraftaki rakamla çarp
 /= sağ taraftaki rakama böl
 %= sağ taraftaki rakama bölüp kalanını al işlemleridir.

++ ve -- işlemleri değişkenden önce veya sonra gelebilir. Bu iki şeklin değişken tek başlarına kullanıldığında bir farklılıkları yoktur, fakat ikinci bir değişkenle birlikte tek bir deyimde kullanıldıklarında ikinci değişkenin değeri değişir. Aşağıdaki iki program parçasına göz atalım.

++ işlemcisinin değişkenden sonra kullanılması

```

int a=2;
int c;
c=a++; // a nın değeri 3 c nin değeri 2

```

++ işlemcisinin değişkenden önce kullanılması

```
int a=2;  
int c;  
c=++a; // a nın değeri 3 c nin değeri 3
```

a değişkeninin değeri başlangıçta 2 ise görüldüğü gibi hem ++a işleminde hem de a++ işleminde değeri 3 e çıkmaktadır. Ancak c değişkeninin değeri c=++a işleminde 2 olarak kalırken (önce değeri yükle sonra arttır), c=++a işleminde 3 olmaktadır(önce arttır, sonra değeri yükle). ++ ve -- işlemleri genelde tam sayı değişkenler için kullanılır.

Program 1.4.5 da aritmetik işlemleri gösteren bir program verilmiştir. Bu programın sonuçları da altta verilmektedir. Programı basamak basamak takip ederek sonuçların doğruluğunu elde hesaplayarak kontrol edebilirsiniz.

Program 1.4.5 : [Aritmetik1.java](#) aritmetik işlem örneği

```
import java.io.*;  
public class Aritmetik1  
{ public static void main(String[] args)  
{  
int sayi1=1;  
int sayi2=2;  
sayi1+=sayi2;  
System.out.println("sayi1 += : "+ sayi1);  
sayi1-=sayi2;  
System.out.println("sayi1 -= : "+ sayi1);  
sayi1*=sayi2;  
System.out.println("sayi1 *= : "+ sayi1);  
sayi1/=sayi2;  
System.out.println("sayi1 /= : "+ sayi1);  
}  
}
```

Aritmetik1 sınıfının sonuçları

```
sayi1 += : 3  
sayi1 -= : 1  
sayi1 *= : 2  
sayi1 /= : 1
```

1.5 STRING, CHAR VE BOOLEAN TİPİ DEĞİŞKENLER VE İŞLEMLERİ

String nesne tipi değişkeni

String değişkeni yazı yazdırma işleri için kullanılır. Nesne türü değişkendir. Java dilinde char değişken türü kullanılarak tanımlanmış bir nesne tipi değişkendir. Aşağıdaki küçük program parçacığında bu değişken türünün kullanılmasını görüyoruz.

```
String a="ali";  
String c="veli";  
String d;  
d=a+b; // d nin değeri "ali veli"  
System.out.println(d); // bu satır ali veli çıktısı verir
```

String türü sabitler her zaman " işaretleri arasına yazılırlar ve + işaretiyle bir araya getirilebilirler.

String türü değişkenleri tanımlamak için aynı zamanda

```
String a=new String("ali");  
String c=new String("veli");
```



```
String d=new String();
d=a+b; // d nin değeri "ali veli"
System.out.println(d); // bu satır ali veli çıktısı verir
Formunu da kullanabiliriz. İkinci verdiğimiz form işlem olarak ilk verilenin tamamen aynıdır.
```

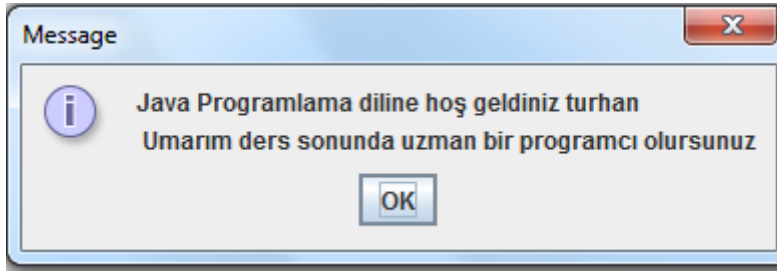
String türü bir değişkenin toplam boyutu length() at metoduyla belirlenebilir. Örneğin

```
String a=new String("ali");
int x=a.length();
x değişkenine 3 değerini yükleyecektir.
```

Program 1.5.1 : [BPIH1E7.java](#) String tipi değişken işlem örneği

```
import javax.swing.*;

public class BPIH1E7
{
    // main metodu
    public static void main( String args[] )
    {
        String s1 = "Java Programlama diline hoş geldiniz ";
        String s2=JOptionPane.showInputDialog(" isminizi giriniz :");
        String s3="\n Umarım ders sonunda uzman bir programcı olursunuz";
        String s4=s1+s2+s3;
        JOptionPane.showMessageDialog(null,s4); // sonucu göster
    } // main metodu sonu
} // BPIH1E7 sınıfı sonu
```



String tipi değişkenlerde de ekleme yapmak için + deyimini yerine += deyiminin kullanılması mümkündür.

Program 1.5.2 : [BPIH1E8.java](#) String tipi değişken işlem örneği

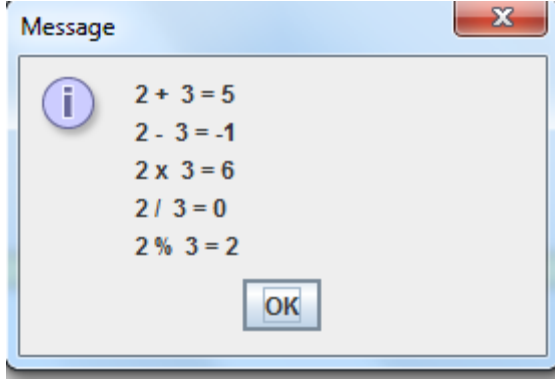
```
import javax.swing.*; // program uses class JOptionPane in swing library

class BPIH2E8
{
    // main metod
    public static void main(String args[])
    {
        int sayi1,sayi2;
        int toplam,fark,carpim,bolum,kalan;
        sayi1=2;
        sayi2=3;
        toplam = sayi1+sayi2;
        fark = sayi1-sayi2;
        carpim = sayi1*sayi2;
        bolum=sayi1/sayi2;
        kalan=sayi1%sayi2;
        String s=sayi1+" + "+sayi2+" = "+toplam+"\n";
        s+=sayi1+" - "+sayi2+" = "+fark+"\n";
        s+=sayi1+" x "+sayi2+" = "+carpim+"\n";
    }
}
```

```

s+=sayi1+" / "+sayi2+" = "+bolum+"\n";
s+=sayi1+" % "+sayi2+" = "+kalan+"\n";
OptionPane.showMessageDialog(null,s);
} // main metodu sonu
}

```



Char (Harf) tipi deęişkenler

Harf (char) tipi deęişkenler bilgisayarda harfleri tanımlayan temel deęişkenlerdir. Üstte incelediğimiz String tipi deęişkenlerde bu tür deęişkenlerin birçoğunun bir araya gelmesiyle oluşur. Java char tipi deęişkenlerin tanımlanmasında uluslararası harf tanımlama standartı olan Unicode standartını kullanır. Unicode harf standart sistemi 4 hegzagonal (16 sayılı) sayıdan oluşur. 16lı sayı sistemlerinin onlu ve ikili sayı sistemleriyle ilişkisi Tablo 1.5.1 de verilmiştir.

Tablo 1.5.1 Heksagonal(16 tabanlı), 10 tabanlı ve binary(iki tabanlı sayı sistemlerinin eşdeęer deęerleri

Heksagonal	10tabanlı	2 tabanlı (binary)
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

ISO Unicode standartlarında '\u0041' kodu 'A' harfi anlamına gelir. Veya '\u03E1' kodu 'α' harfini tanımlar. ISO Unicode karakter setinin ilk iki rakamı sıfır olduğunda ASCII karakter kodunu tanımlar. Unicode hakkında daha ayrıntılı bilgi için

<http://unicode.org>

internet adresini ziyaret edebilirsiniz. Aşağıdaki küçük program parçacığı **char** tipi A1, A2 ve alpha1,alpha2 deęişkenlerine A ve α harflerini yüklemektedir.

```

char A1,A2;
char alpha1,alpha2;
A1='\u0041';

```

```
A2='A';  
alpha1='\u03E1';  
alpha2='α';
```

Burada şunu kaydedelim alpha1 değişkenini her ortamda doğru olarak görebiliriz, fakat alpha2 değişkeni bazı ortamlarda doğru olarak görülmeyebilir. Tablo 1.5.2 de sık kullanabileceğimiz bazı Unicode kodları listelenmiştir. Karakter kodları kullanılarak yazmada çeşitli kontrol işlemleri de örneğin bir harf geri gitme, satır başı yapma, tab pozisyonuna gitme gibi. Harf değişkenleri String değişkenleriyle toplama işlemi yoluyla rahatça birleştirilebilir. Bu arada bazı karakterlerin konsol ortamında görülme zorluğu olabileceğini de belirtelim, ancak grafik ortamında her zaman doğru olarak gösterileceklerdir. Bu yüzden değişik harfler gösterirken grafik ortam çıktısını tercih etmeniz tavsiye edilir. İlk örnek programımızda bazı Türkçe harflerin toplanarak birleştirilerek yazılmasını görmekteyiz. İkinci program yunan karakterlerini listelemektedir. Üçüncü programda Türkçe harfler yer almakta, son örnek programda ise matematik denklemler yazılmaktadır.

Program 1.5.3 characterA.java program

```
import java.io.*;  
import javax.swing.*;  
  
class harfA  
{  
    public static void main(String args[])  
    {  
        char b1,b2,b3,b4,b5,b6;  
        b1='ü';  
        b2='ğ';  
        b3='ç';  
        b4='a';  
        b5='b';  
        b6='c';  
        String s="" +b1+b2+b3+b4+b5+b6;  
        JOptionPane.showMessageDialog(null,s,"harf işlem",JOptionPane.PLAIN_MESSAGE);  
    }  
}
```

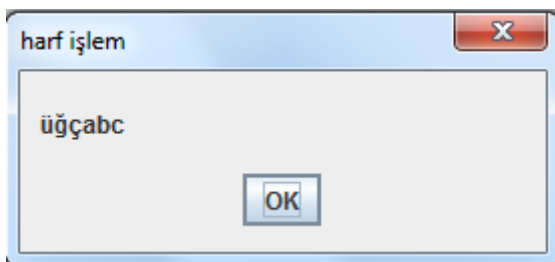


Table 1.5.2 Some unicode character tables

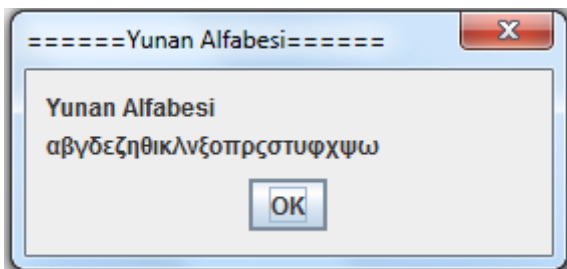
	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	\$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074
5	ENQ 0005	NAK 0015	% 0025	5 0035	E 0045	U 0055	e 0065	u 0075
6	ACK 0006	SYN 0016	& 0026	6 0036	F 0046	V 0056	f 0066	v 0076
7	BEL 0007	ETB 0017	' 0027	7 0037	G 0047	W 0057	g 0067	w 0077
8	BS 0008	CAN 0018	(0028	8 0038	H 0048	X 0058	h 0068	x 0078
9	HT 0009	EM 0019) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079
A	LF 000A	SUB 001A	* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A
B	VT 000B	ESC 001B	+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F

	008	009	00A	00B	00C	00D	00E	00F
0	XXX 0080	DC5 0090	NB SP 00A0	° 00B0	À 00C0	Đ 00D0	à 00E0	đ 00F0
1	XXX 0081	PU1 0091	¡ 00A1	± 00B1	Á 00C1	Ñ 00D1	á 00E1	ñ 00F1
2	BPH 0082	PU2 0092	¢ 00A2	² 00B2	Â 00C2	Ò 00D2	â 00E2	ò 00F2
3	NBH 0083	STS 0093	£ 00A3	³ 00B3	Ã 00C3	Ó 00D3	ã 00E3	ó 00F3
4	IND 0084	CCH 0094	¤ 00A4	´ 00B4	Ä 00C4	Ô 00D4	ä 00E4	ô 00F4
5	NEL 0085	MW 0095	¥ 00A5	µ 00B5	Å 00C5	Õ 00D5	å 00E5	õ 00F5
6	SSA 0086	SPA 0096	¦ 00A6	¶ 00B6	Æ 00C6	Ö 00D6	æ 00E6	ö 00F6
7	ESA 0087	EPA 0097	§ 00A7	· 00B7	Ç 00C7	× 00D7	ç 00E7	÷ 00F7
8	HTS 0088	SOS 0098	¨ 00A8	¸ 00B8	È 00C8	Ø 00D8	è 00E8	ø 00F8
9	HTJ 0089	XXX 0099	© 00A9	¹ 00B9	É 00C9	Ù 00D9	é 00E9	ù 00F9
A	VTS 008A	SCI 009A	ª 00AA	º 00BA	Ê 00CA	Ú 00DA	ê 00EA	ú 00FA
B	PLD 008B	CSI 009B	« 00AB	» 00BB	Ë 00CB	Û 00DB	ë 00EB	û 00FB
C	PLU 008C	ST 009C	¬ 00AC	¼ 00BC	Ì 00CC	Ü 00DC	ì 00EC	ü 00FC
D	RI 008D	OSC 009D	SHY 00AD	½ 00BD	Í 00CD	Ý 00DD	í 00ED	ý 00FD
E	SS2 008E	PM 009E	® 00AE	¾ 00BE	Î 00CE	Þ 00DE	î 00EE	þ 00FE
F	SS3 008F	APC 009F	¯ 00AF	¿ 00BF	Ï 00CF	ß 00DF	ï 00EF	ÿ 00FF

	220	221	222	223	224	225	226	227	228	229	22A	22B	22C	22D	22E	22F
0	∇ 2200	\sqcup 2210	\angle 2220	\mathbb{F} 2230	\wr 2240	\doteq 2250	\neq 2260	\nlessdot 2270	\nlessgtr 2280	\square 2290	\boxtimes 22A0	\wp 22B0	\wedge 22C0	\complement 22D0	\nlessdot 22E0	$\cdot\cdot$ 22F0
1	\complement 2201	Σ 2211	\triangleleft 2221	\mathcal{F} 2231	\nlessdot 2241	\doteq 2251	\equiv 2261	\nlessdot 2271	\nlessgtr 2281	\sqsubset 2291	\square 22A1	\wp 22B1	\vee 22C1	\supset 22D1	\nlessdot 22E1	$\cdot\cdot$ 22F1
2	∂ 2202	$-$ 2212	\nlessdot 2222	\mathcal{F} 2232	\wr 2242	\doteq 2252	\neq 2262	\nlessdot 2272	\subset 2282	\sqsubset 2292	\top 22A2	\triangleleft 22B2	\cap 22C2	\supset 22D2	\nlessdot 22E2	\in 22F2
3	\exists 2203	\mp 2213	$ $ 2223	\mathcal{F} 2233	\wr 2243	\doteq 2253	\equiv 2263	\nlessdot 2273	\supset 2283	\sqsubset 2293	\top 22A3	\triangleleft 22B3	\cup 22C3	\supset 22D3	\nlessdot 22E3	\in 22F3
4	\nlessdot 2204	\dagger 2214	\dagger 2224	$\cdot\cdot$ 2234	\neq 2244	\doteq 2254	\nlessdot 2264	\nlessdot 2274	\nlessdot 2284	\sqsubset 2294	\top 22A4	\nlessdot 22B4	\diamond 22C4	\supset 22D4	\sqsubset 22E4	\in 22F4
5	\emptyset 2205	$/$ 2215	\parallel 2225	$\cdot\cdot$ 2235	\equiv 2245	\doteq 2255	\nlessdot 2265	\nlessdot 2275	\nlessdot 2285	\oplus 2295	\top 22A5	\nlessdot 22B5	\cdot 22C5	$\#$ 22D5	\sqsubset 22E5	\in 22F5
6	\triangle 2206	\backslash 2216	\nlessdot 2226	\cdot 2236	\nlessdot 2246	\mathbb{H} 2256	\equiv 2266	\nlessdot 2276	\supset 2286	\ominus 2296	\top 22A6	\nlessdot 22B6	\star 22C6	\nlessdot 22D6	\nlessdot 22E6	\in 22F6
7	∇ 2207	\ast 2217	\wedge 2227	$\cdot\cdot$ 2237	\nlessdot 2247	\doteq 2257	\equiv 2267	\nlessdot 2277	\supset 2287	\otimes 2297	\top 22A7	\nlessdot 22B7	\ast 22C7	\nlessdot 22D7	\nlessdot 22E7	\in 22F7
8	\in 2208	\circ 2218	\vee 2228	\cdot 2238	\approx 2248	\equiv 2258	\nlessdot 2268	\nlessdot 2278	\nlessdot 2288	\ominus 2298	\top 22A8	\nlessdot 22B8	\times 22C8	\nlessdot 22D8	\nlessdot 22E8	\in 22F8
9	\nlessdot 2209	\cdot 2219	\cap 2229	$\cdot\cdot$ 2239	\nlessdot 2249	\nlessdot 2259	\nlessdot 2269	\nlessdot 2279	\nlessdot 2289	\odot 2299	\top 22A9	\nlessdot 22B9	\times 22C9	\nlessdot 22D9	\nlessdot 22E9	\in 22F9
A	\in 220A	$\sqrt{\quad}$ 221A	\cup 222A	\doteq 223A	\equiv 224A	\nlessdot 225A	\nlessdot 226A	\nlessdot 227A	\nlessdot 228A	\odot 229A	\top 22AA	\top 22BA	\times 22CA	\nlessdot 22DA	\nlessdot 22EA	\in 22FA
B	\ni 220B	$\sqrt[3]{\quad}$ 221B	\int 222B	\nlessdot 223B	\equiv 224B	\nlessdot 225B	\nlessdot 226B	\nlessdot 227B	\nlessdot 228B	\otimes 229B	\top 22AB	\nlessdot 22BB	\nlessdot 22CB	\nlessdot 22DB	\nlessdot 22EB	\in 22FB
C	\nlessdot 220C	$\sqrt[4]{\quad}$ 221C	\mathbb{F} 222C	\sim 223C	\equiv 224C	\nlessdot 225C	\nlessdot 226C	\nlessdot 227C	\nlessdot 228C	\ominus 229C	\top 22AC	\nlessdot 22BC	\nlessdot 22CC	\nlessdot 22DC	\nlessdot 22EC	\in 22FC
D	\ni 220D	∞ 221D	\mathbb{F} 222D	\sim 223D	\nlessdot 224D	\mathbb{H} 225D	\nlessdot 226D	\nlessdot 227D	\nlessdot 228D	\ominus 229D	\nlessdot 22AD	\nlessdot 22BD	\nlessdot 22CD	\nlessdot 22DD	\nlessdot 22ED	\in 22FD
E	\blacksquare 220E	∞ 221E	\mathcal{F} 222E	\sim 223E	\nlessdot 224E	\mathbb{H} 225E	\nlessdot 226E	\nlessdot 227E	\nlessdot 228E	\oplus 229E	\top 22AE	\nlessdot 22BE	\nlessdot 22CE	\nlessdot 22DE	$\cdot\cdot$ 22EE	\in 22FE
F	\prod 220F	\perp 221F	\mathbb{F} 222F	\sim 223F	\nlessdot 224F	\nlessdot 225F	\nlessdot 226F	\nlessdot 227F	\nlessdot 228F	\square 229F	\top 22AF	\nlessdot 22BF	\nlessdot 22CF	\nlessdot 22DF	$\cdot\cdot$ 22EF	\in 22FF

Program 1.5.4 harfB.java programı

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;
class harfB
{
    public static void main(String args[])
    { //Yunan Harfleri
        char b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14;
        char b15,b16,b17,b18,b19,b20,b21,b22,b23,b24;
        String s="Yunan Alfabeti \n";
        b1="\u03B1"; // alfa
        b2="\u03B2"; // beta
        b3="\u03B3"; // gamma
        b4="\u03B4"; // delta
        b5="\u03B5"; // epsilon
        b6="\u03B6"; // fi
        b7="\u03B7"; // eta
        b8="\u03B8"; // teta
        b9="\u03B9"; // ypsilon
        b10="\u03BA"; // kappa
        b11="\u03BB"; // lambda
        b12="\u03BC"; // mü
        b13="\u03BD"; // nü
        b14="\u03BE"; // ksi
        b15="\u03BF"; // o
        b16="\u03C0"; // pi
        b17="\u03C1"; // ro
        b18="\u03C2"; //
        b19="\u03C3"; //
        b20="\u03C4"; //
        b21="\u03C5"; //
        b22="\u03C6"; //
        b23="\u03C7"; //
        b24="\u03C8"; //
        s+=""+b1+b2+b3+b4+b5+b6+b7+b8+b9+b10+b11+b12+b13+b14+b15+b16+b17+b18+b19+b20+b21+b22+b23+b24;
        JOptionPane.showMessageDialog(null,s,"=====Yunan Alfabeti=====",JOptionPane.PLAIN_MESSAGE);
    }
}
```



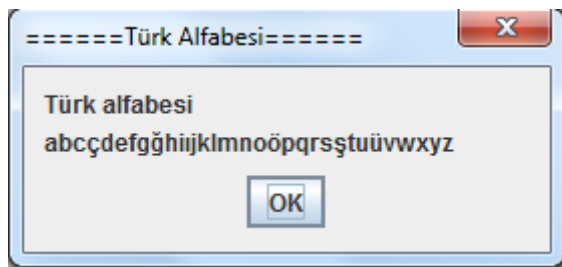
Program 1.5.5 harfC.java programı

```
import java.io.*;
import javax.swing.*;
```

```

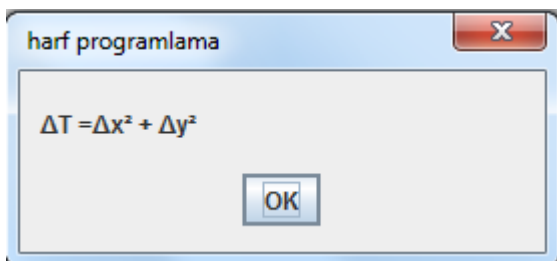
class harfC
{
    public static void main(String args[])
    {
        char b1,b2,b3,b3a,b4,b5,b6,b7,b7a,b8,b9,b9a,b10,b11,b12,b13,b14,b15,b15a,b16,b17,b18,b19,b20;
        char b21,b22,b23,b24,b25,b25a,b26,b27,b27a,b28,b29,b30,b31,b32,b33;
        String s="Türk alfabeti\n";
        b1='\u0061'; //a
        b2='\u0062'; //b
        b3='\u0063'; //c
        b3a='\u00E7'; //ç
        b4='\u0064'; //d
        b5='\u0065'; //e
        b6='\u0066'; //f
        b7='\u0067'; //g
        b7a='\u011F'; //ğ
        b8='\u0068'; //h
        b9a='\u0131'; //ı
        b9='\u0069'; //i
        b10='\u006A'; //j
        b11='\u006B'; //k
        b12='\u006C'; //l
        b13='\u006D'; //m
        b14='\u006E'; //n
        b15='\u006F'; //o
        b15a='\u00F6'; //ö
        b22='\u0070'; //p
        b23='\u0071'; //q
        b24='\u0072'; //r
        b25='\u0073'; //s
        b25a='\u015F'; //ş
        b26='\u0074'; //t
        b27='\u0075'; //u
        b27a='\u00FC'; //ü
        b28='\u0076'; //v
        b29='\u0077'; //w
        b30='\u0078'; //x
        b31='\u0079'; //y
        b33='\u007A'; //z
        s+=""+b1+b2+b3+b3a+b4+b5+b6+b7+b7a+b8+b9+b9a+b10+b11+b12+b13+b14+b15+b15a+b22+b23+b24+b25+b25a+b26+b27+b27a+b28+b29+b30+b31+b32+b33;
        JOptionPane.showMessageDialog(null,s,"====Türk Alfabeti====",JOptionPane.PLAIN_MESSAGE);
    }
}

```



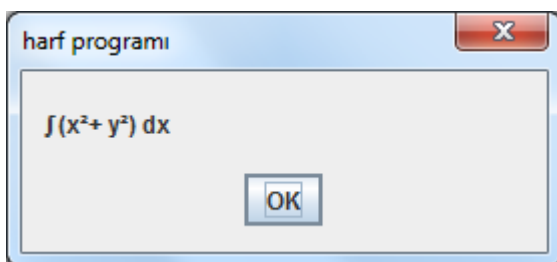
Program 1.5.6 harfD.java program

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;
class harfD
{
    public static void main(String args[])
    {
        char b1,b2;
        b1='\u0394'; //büyük harf delta
        b2='\u00B2'; //kare işareti
        String s="" + b1 + "T =" + b1 + "x" + b2 + " + " + b1 + "y" + b2;
        JOptionPane.showMessageDialog(null,s," harf programlama",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Program 1.5.7 characterE.java program

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;
class harfE
{
    public static void main(String args[])
    {
        char b1,b2;
        b1='\u222B'; //integral
        b2='\u00B2';
        char b4='\u221E';
        String s="" + b1 + "(x" + b2 + " + y" + b2 + ") dx";
        JOptionPane.showMessageDialog(null,s,"harf programı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



boolean deęişken türü

boolean deęişken türü mantık işlemlerinde kullanılır. Sadece true veya false deęerleri alır. true doğru false yanlış anlamı taşır. Mantık deęişkenlerine doğrudan true veya false deęerleri yüklenebileceęi gibi dięer deęişkenleri mantık işlemlerini kullanarak karşılaştırarakta deęerleri programların içinde hesaplanabilir. Aşağıdaki küçük program parçası mantık deęişkeni ilk mantık deęişkenini tanımlamakta ve doğru deęerini bu deęişkene yüklemektedir.

boolean ilkmantıkdeęiskeni;

ilkmantıkdeęiskeni=true;

Tam sayı ve gerçek deęişkenlerin aritmetik işlemlerde kullanılabildięi gibi boolean deęişkenler de mantıksal işlemlerde kullanılabılırler. Mantıksal işlem işlemcileri şunlardır :

Tablo 1.5.3 Java dilindeki mantıksal işlemler ve anlamları

Mantıksal İşlemci	Anlamı
&&	Ve
	Veya
>	Büyüktür
<	Küçüktür
==	Eşittir
>=	büyük ve eşittir
<=	küçük ve eşittir
!	Deęildir
!=	eşit deęildir

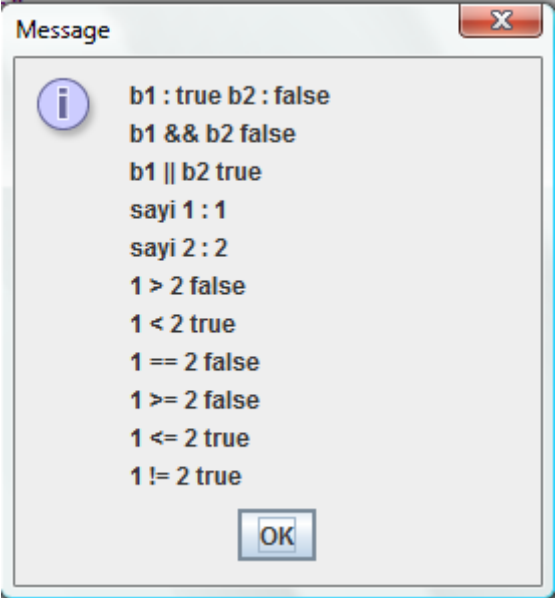
Mantıksal işlemleri pekiştirmek amacıyla Program 1.18 de bir örnek verilmiştir.

Program 1.5.8 : Java mantıksal işlemleri örnek problemi [booleanvar.java](#)

```
import javax.swing.*;
public class booleanvar
{
    public static void main(String[] args)
    {
        boolean b1,b2;
        b1=true;
        b2=false;
        String s="b1 : "+ b1+" b2 : "+b2+"\n";
        s+="b1 && b2 "+ (b1&& b2)+"\n";
        s+="b1 || b2 "+ (b1||b2)+"\n";
        int sayi1,sayi2;
        sayi1=1;
        sayi2=2;
        s+="sayi 1 : "+sayi1+"\n";
        s+="sayi 2 : "+sayi2+"\n";
        b1=(sayi1 > sayi2);
        s+="sayi1 > "+sayi2+" "+ b1+"\n";
        b1=(sayi1 < sayi2);
        s+="sayi1 < "+sayi2+" "+ b1+"\n";
        b1=(sayi1 == sayi2);
        s+="sayi1 == "+sayi2+" "+ b1+"\n";
        b1=(sayi1 >= sayi2);
        s+="sayi1 >= "+sayi2+" "+ b1+"\n";
        b1=(sayi1 <= sayi2);
        s+="sayi1 <= "+sayi2+" "+ b1+"\n";
        b1=(sayi1 != sayi2);
```

```
s+=sayi1+" != "+sayi2+" "+ b1;  
OptionPane.showMessageDialog(null,s);  
}  
}
```

Aşağıda bu örnek problemin (Program 1.5.8) sonucunu verilmektedir. Satır satır programı takip ederek sonuçları kontrol ediniz.



Stringin alt metodu equals() karşılaştırma yapar. Örneğin

```
String a=new String("ali");  
boolean y=a.equals("ali");  
boolean z=a.equals("veli");
```

y boolean değişkenine true değerini, z boolean değişkenine false değerini yükleyecektir.

1.6 JAVA NESNESİ OLARAK TANIMLANMIŞ TEMEL DEĞİŞKENLER

Java nesne temelli bir dildir. Javadaki tüm değişken türlerinin nesne kökenli eşdeğerleri mevcuttur. Bunun yanında temel değişken türleri arasında yer almayan üst bölümde yer verdiğimiz String gibi sadece nesne kökenli olarak tanımlanmış değişken türleri de mevcuttur. Şimdi bunların bir kısmını biraz daha detaylı inceleyelim.

Integer nesne tipi değişkeni
Integer değişkeni, string değişkeni gibi

```
Integer i;  
i=new Integer(3);  
veya  
Integer i=new Integer(3);
```

Şeklinde tanımlanabilir. Integer tanımı String değişkeni üzerinden de tanımlanabilir. Örneğin

```
String s="15";  
Integer i=new Integer(s);
```

Tanımlı 15 değerini yükler. Integer değerini int değerine (temel değişken) çevirebiliriz.

```
int x;  
Integer y=new Integer(3);
```

```
x=y.Integer.IntegerValue();
```

String değerini int değerine direkt olarak çevirmek içinse

```
String s="15";  
int x=Integer.parseInt(s);
```

kullanılır.

int tipi (temel) değişkeni String değişkenine değiştirmek için

```
int x=3;  
String s=Integer.toString(x);
```

kullanılır.

Double nesne tipi değişkeni

Double değişkeni kullanım olarak Integer değişkeninden bir farkı yoktur. Aynı tanımlamalar Integer yerine Double kullanılarak yapılabilir. Değişkenler

```
Double x;  
X=new Double(3.66e5);  
veya  
Double x=new Double(3.66e5);
```

Şeklinde tanımlanabilir. Double tanımı String değişkeni üzerinden de tanımlanabilir. Örneğin

```
String s="15.66e-3";  
Double x=new Double(s);
```

Tanımı $15.66 \cdot 10^{-3}$ değerini yükler. Double değerini double değerine (temel değişken) çevirebiliriz.

```
double x;  
Double y=new Double(15.66e-3);  
x=y.Double.doubleValue();
```

```
int x;  
Double y=new Double(15.66e3);  
x=y.Double.intValue();
```

String değerini double (temel değişken) değerine direkt olarak çevirmek içinse

```
String s="15.25";  
double x=Double.parseDouble(s);
```

İşlemi kullanılabilir.

double tipi (temel) değişkeni String değişkenine değiştirmek için

```
double x=3.75;  
String s=Double.toString(x);
```

kullanılır.

diğer nesne temelli değişken türleri

Object, Long, Float, Boolean, Character, Vector gibi diğer nesne tipi değişken türleri de mevcuttur. Object nesne tipi tüm bu değişken nesne tiplerini içinde barındıran genel bir tiptir. Tüm nesne tipi değişkenlerin bizim için temel avantajı alt metodlarını kullanarak işlemler gerçekleştirime olasılıklarıdır. Fakat bu tür değişkenler temel değişken türlerine göre daha çok hafıza yeri işgal ettiklerinden mecbur kalınmadıkça da kullanılmamalıdır.

Program 1.4.2 yi biraz deęiřtirerek yazarsak

Program 1.6.1 BPIH1E5A.java programı

```
import javax.swing.*;

public class BPIH1E5B
{
    // main metodu
    public static void main( String args[] )
    {
        double number1; // birinci gerek sayı
        double number2; // ikinci gerek sayı
        double toplam; // toplam rakam
        String s1=JOptionPane.showInputDialog("ilk gerek sayıyı gir:");
        number1 = Double.parseDouble(s1);
        String s2=JOptionPane.showInputDialog("ikinci gerek sayıyı gir:");
        number2 = Double.parseDouble(s2);
        toplam = number1 + number2; // sayıları topla
        String s3="Toplam =" +toplam;
        JOptionPane.showMessageDialog(null,s3 ); // toplamı gster
    } // main metodu sonu

} // BPIH1E5B sınıfı sonu
```

Veya String deęiřkenini nce Double deęiřkenine sonra da double deęiřkenine dnřtrebiliriz.

Program 1.6.2 BPIH1E5A.java programı

```
import javax.swing.*;

public class BPIH1E5C
{
    // main metodu
    public static void main( String args[] )
    {
        double number1; // birinci gerek sayı
        double number2; // ikinci gerek sayı
        double toplam; // toplam rakam
        String s1=JOptionPane.showInputDialog("ilk gerek sayıyı gir:");
        Double n1=new Double(s1);
        number1 = n1.doubleValue();
        String s2=JOptionPane.showInputDialog("ikinci gerek sayıyı gir:");
        Double n2=new Double(s2);
        number2 = n2.doubleValue();
        toplam = number1 + number2; // sayıları topla
        String s3="Toplam =" +toplam;
        JOptionPane.showMessageDialog(null,s3 ); // toplamı gster
    } // main metodu sonu

} // BPIH1E5B sınıfı sonu
```

1.7 FİNAL TERİMİ VE SABİTLER

Java dilinde deęiřken yerine sabit kullanmak istersek tanımın bařına **final** szcęn gettiririz. Final olarak tanımlanan sabitlerin bir kere deęerleri verildikten sonra deęiřtirilemez. rneęin

Program 1.8.1 : Java bit işlemleri örnek problemi [bitislemi.java](#)

```
class bitislemi{
    public static void main (String args[])
    {
        int x=7;
        System.out.println("x = "+x);
        x= x<<1;
        System.out.println("x = "+x);
        x= x<<1;
        System.out.println("x = "+x);
        x= x>>2;
        System.out.println("x = "+x);
        x= ~x;
        System.out.println("x = "+x);
    }
}
```

bu programın sonucu

```
x = 7
x = 14
x = 28
x = 7
x = -8
```

şeklinde olacaktır. Çıkan sonuçların bit eşdeğerlerini kontrol ediniz. Bit işlemleri aslında bilgisayarın ve digital elektronik devrelerin temel işlemleridir. Bu işlemler assembler dillerinin de temel işlemlerini teşkil ederler.

1.9 JAVA DİLİNİN TEMEL KOMUTLARI : IF KOMUT YAPISI

Bilgisayar dillerinde değişkenlerden ve işlemlerden sonraki en temel yapı temel dil komutlarıdır. Şimdi bu komut yapılarına tek tek göz atalım. İlk yapımız **if - elseif - else mantıksal karşılaştırma yapısı**'dır. Bu yapı programda karar mekanizmalarını oluşturmak için kullanılır. Temel yapısı :

```
if( boolean değişken yada sabit)
    { boolean değişken veya sabitin değeri true olduğunda yapılan işlemler}
else if( boolean değişken yada sabit)
    { boolean değişken veya sabitin değeri true olduğunda yapılan işlemler}
else
    { yukardaki boolean değişken veda sabitlerin hiçbiri true değerine eşit değilse yapılacak olan işlemler}
```

else if yapısı probleme göre istenilen sayıda olabilir, if ve else yapısı sadece bir tane olarak kullanılır. if yapısı java ve C dillerinde kısaltılmış (kompozit) bir şekilde de yazılabilir bu form

değişken adı = if(boolean değişken yada sabit ? boolean true ise değişken değeri : boolean false ise değişken değeri);

şeklinde dir. Küçük bir örnek programla if-elseif-else yapısının çalışmasını inceleyebiliriz.

Program 1.9.1 : if yapısını gösteren [ifyapisi.java](#) programı

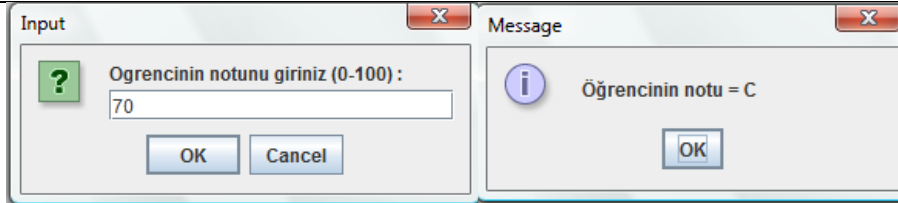
```
import javax.swing.JOptionPane;

class ifyapisi
{
    public static void main (String args[])
    {
        String s1=JOptionPane.showInputDialog("Ogrencinin notunu giriniz (0-100) : ");
        int not=Integer.parseInt(s1);
    }
}
```

```

String s;
if( not >= 90) { s="A";}
else if(not >=75) { s="B";}
else if(not >=60) { s="C";}
else if(not >=50) { s="D";}
else if(not >=40) { s="E";}
else { s="F";}
s=" Öğrencinin notu = "+s;
JOptionPane.showMessageDialog(null,s);
}
}

```



If yapısına boolean değişkenleri direk olarak yazarsak

Program 1.9.2 : if yapısını gösteren [ifyapisi1.java](#) programı

```

import javax.swing.JOptionPane;

class ifyapisi1
{
    public static void main (String args[])
    {
        String s1=JOptionPane.showInputDialog("Oğrencinin notunu giriniz (0-100) : ");
        int not=Integer.parseInt(s1);
        String s;
        Boolean b1,b2,b3,b4,b5;
        b1=not >= 90;
        b2=not >=75;
        b3=not >=60;
        b4=not >=50;
        b5=not >=40;
        if(b1) { s="A";}
        else if(b2) { s="B";}
        else if(b3) { s="C";}
        else if(b4) { s="D";}
        else if(b5) { s="E";}
        else { s="F";}
        JOptionPane.showMessageDialog(null,s,"Oğrenci notu",JOptionPane.PLAIN_MESSAGE);
    }
}

```

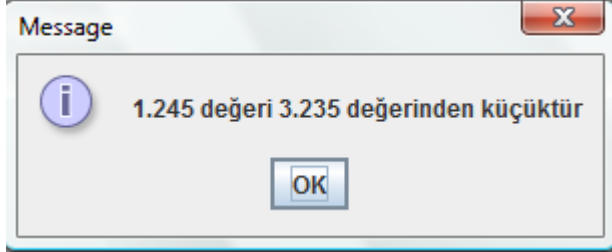
Bu program birinci programla aynıdır.

Program 1.9.3 : if yapısını gösteren [if2.java](#) programı

```

import javax.swing.JOptionPane;
public class if2
{
    public static void main (String args[])
    {
        double x=1.245;
        double y=3.235;
        String s1="";
        if(x<y)

```

```
{ s1=x+" değeri "+y+" değerinden küçüktür";}  
else if(x>y)  
{ s1=x+" değeri "+y+" değerinden büyüktür";}  
else if(x==y)  
{ s1=x+" değeri "+y+" değerine eşittir";}  
JOptionPane.showMessageDialog(null,s1);  
}}
```

Aşağıdaki program parçasında a değişkeninin değeri : true yazısını yazdıracaktır.

```
boolean a=true;  
if(a)  
{System.out.println("a değişkeninin değeri : true");}  
else  
{System.out.println("a değişkeninin değeri : false");}
```

if - elseif - else yapıları iç içe girmiş yapılar veya basit mantıkların && (ve), || (veya) işlemleriyle bağlanmasıyla daha kompleks mantıksal şartlar oluşturulabilir.

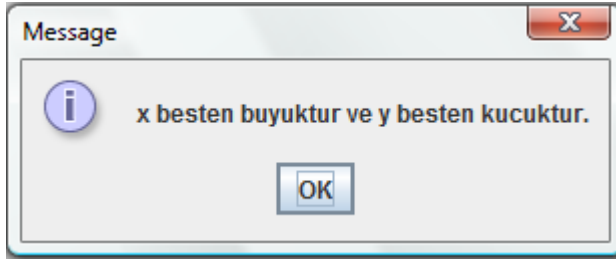
İç içe geçmiş if yapısı kullanılarak mantıksal kontrol prosesi alttaki programda açıklanmıştır :

İç içe geçmiş if yapısı :

Program 1.9.3 : iç içe geçmiş if yapısını gösteren [if3.java](#) programı

```
import javax.swing.JOptionPane;  
public class if3  
{  
public static void main (String args[])  
{  
String s1="";  
double x = 7.0;  
double y = 3.0;  
if(x > 5)  
{ if(y > 5) { s1="x ve y besten buyuktur.";}  
else if(y==5) { s1="x besten buyuktur ve y bese esittir.";}  
else { s1="x besten buyuktur ve y besten kucuktur.";}  
}  
else  
{ if(y > 5) { s1="x besten kucuktur veya esittir ve y 5 den buyuktur.";}  
else if(y==5) { s1="x besten kucuktur veya esittir ve y bese esittir.";}  
else { s1="x besten kucuktur veya esittir ve y besten kucuktur.";}  
}  
JOptionPane.showMessageDialog(null,s1);  
}
```

```
}
```



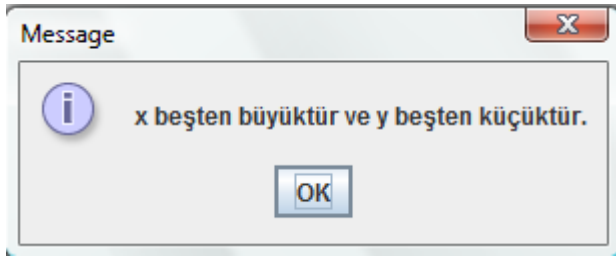
Burada ise && terimiyle bağlanmış bir program parçacığı görüyorsunuz.

&& (ve) deyimiyle bağlanmış if yapısı

Program 1.9.4 : iç içe geçmiş if yapısını gösteren [if4.java](#) programı

```
import javax.swing.JOptionPane;
public class if4
{
    public static void main (String args[])
    {
        String s1="";
        double x = 7.0;
        double y = 3.0;
        if(x>5 && y>5) { s1="x ve y beşten büyüktür.";}
        else if(x>5 && y==5) { s1="x beşten büyüktür ve y ve beşe eşittir.";}
        else if(x>5 && y<5) { s1="x beşten büyüktür ve y beşten küçüktür.";}
        else if(x<5 && y>5) { s1="x beşten küçüktür ve y 5 ten büyüktür.";}
        else if(x < 5 && y==5) { s1="x beşten küçüktür ve y beşe eşittir.";}

        JOptionPane.showMessageDialog(null,s1);
    }
}
```

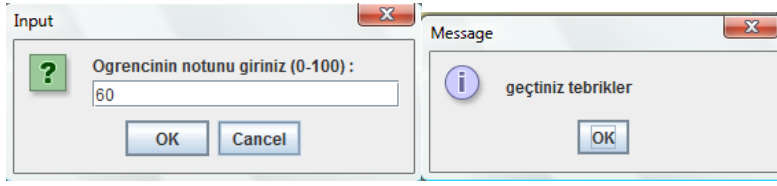


Yukarıdaki program parçacıklarından da görebildiğimiz gibi her türlü kompleks mantığı bilgisayar programlarına aktarmamız mümkündür. If - else if - else yapıları programlamada en çok kullanılan mantık aktarımı yapı taşlarıdır. Bu bölümü bitirmeden önce komposit if yapısına da bir örnek verelim

Program 1.9.4 : kompozit if yapısını gösteren [if5.java](#) programı

```
import javax.swing.JOptionPane;

class if5
{
    public static void main (String args[])
    {
        int not=Integer.parseInt(JOptionPane.showInputDialog("Ogrencinin notunu giriniz (0-100) : "));
        String s=(not >= 50 ? "geçtiniz tebrikler" : "malesef sınıfta kaldınız" );
        JOptionPane.showMessageDialog(null,s);
    }
}
```

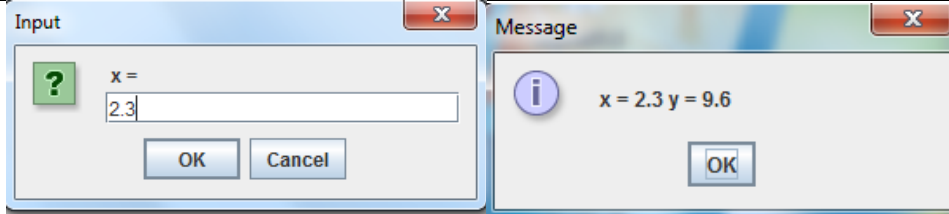


Bir sonraki örnekte kısmi devamlı bir fonksiyonu kompozit if yapısıyla tanımlayalım

```
import javax.swing.JOptionPane;
```

```
class if6
```

```
{
    public static void main (String args[])
    {
        double x=Double.parseDouble(JOptionPane.showInputDialog(" x = "));
        double y=(x <= 5 ? 2.0*x+5.0 : 3.0*x*x );
        JOptionPane.showMessageDialog(null," x = "+x+" y = "+y);
    }
}
```



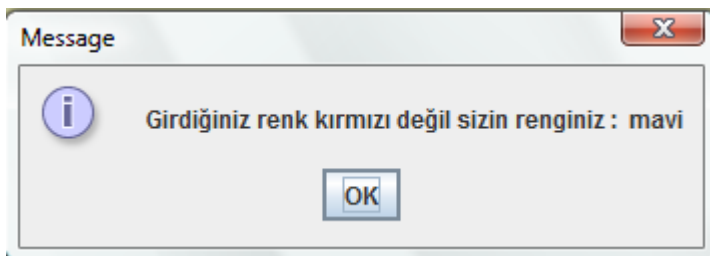
Bir sonraki alıřtırmada String türü deęiřkenler karşılařtırılmaktadır.

Program 1.9.5 : String tipi deęiřkenlerde karşılařtırmayı gösteren [if String.java programı](#)

```
import java.io.*; //java girdi cikti sinifini cagır
import javax.swing.JOptionPane;
```

```
class if_String
```

```
{
    public static void main(String args[])
    {
        // rengi tahmin et
        String s1;
        String s=JOptionPane.showInputDialog(" Bir renk giriniz = ");
        boolean b1=s.equals("kırmızı");
        if(b1) {s1=" Doğru renk kırmızıydı";}
        else {s1=" Girdiđiniz renk kırmızı deęil sizin renginiz : "+s;}
        JOptionPane.showMessageDialog(null,s1);
    }
}
```



Aynı programın kompozit if ile yazılmış versiyonu:

Program 1.9.6: String tipi değişkenlerde kompozit if yapısı ile karşılaştırmayı gösteren kompozit_if_String.java programı

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.JOptionPane;

class kompozit_if_String
{
public static void main(String args[])
{
// rengi tahmin et
String s=JOptionPane.showInputDialog(" Bir renk giriniz = ");
boolean b1=s.equals("kırmızı");
String s1=(b1 ? "Doğru renk kırmızıydı" : " Girdiğiniz renk kırmızı değil sizin renginiz : "+s);
JOptionPane.showMessageDialog(null,s1);
}
}
```

1.10 JAVA DİLİNİN TEMEL KOMUTLARI : WHILE, DO-WHILE, FOR, SWITCH-CASE TEKRARLAMA KOMUT YAPILARI

WHILE VE DO-WHILE TEKRARLAMA YAPISI

While en çok kullanılan tekrarlar yapılarından biridir. () içindeki boolean terim true (doğru) olduğu sürece yeniden işlemi döndürür. Eğer programda boolean işlemi sonuçlandırarak bir ifade yoksa sonsuza dek veya program başka bir yöntemle durdurulana dek devam eder.

```
while( boolean değişken yada sabit)
{ boolean değişken veya sabitin değeri true olduğunda yapılan işlemler }
```

örneğin :

```
while(true)
{ System.out.println("Bu yazı sürekli olarak tekrar edecektir"); }
```

yapısı sonsuz bir döngü oluşturur. Aşağıdaki program yapısı ise sayı 1000 den büyük veya 1000 e eşit olduğunda durur.

```
int sayı=2;
while(sayı <= 1000)
{ sayı *= 2; }
```

While tekrarlar yapısı bazen biraz daha farklı bir yapı içinde verilebilir. Aşağıda gösterilen bu yeni yapı, bir önceki while yapısına göre daha ender kullanılır. **do..while** yapısının while yapısından temel farkı, döngünün içine en az bir kere girilme zorunluğunun olmasıdır.

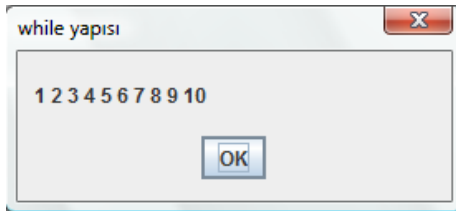
```
do
{
boolean değişken veya sabitin değeri true olduğunda yapılan işlemler.
Bu paranteze boolean değerinin ne olduğuna bakılmaksızın en az bir kere girilir
```

}
while(boolean deęişken yada sabit)

Program 1.10.1 While tekrarlamaya yapısı

```
import javax.swing.*;

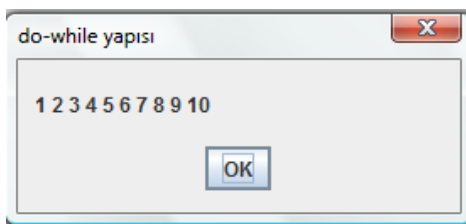
public class while1
{
    public static void main(String args[])
    {
        int saydirici=1;
        String s="";
        while(saydirici <= 10)
        { s=s+saydirici+" ";
          saydirici++;
        }
        JOptionPane.showMessageDialog(null,s,
        "while yapısı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Program 1.10.2 Do-while tekrarlamaya yapısı

```
import javax.swing.*;

public class dowhile1
{
    public static void main(String args[])
    {
        int saydirici=1;
        String s="";
        do
        { s=s+saydirici+" ";
          saydirici++;
        } while(saydirici <= 10);
        JOptionPane.showMessageDialog(null,s,
        "do-while yapısı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Bir sonraki örnekte ekponensiyel fonksiyon seri ile hesaplanacaktır.

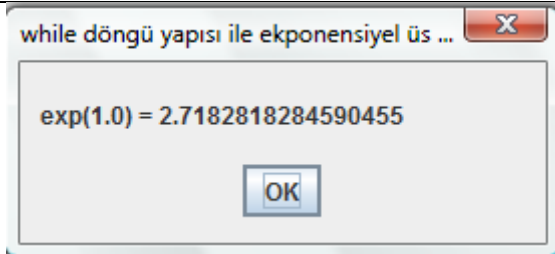
$$e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

e=2.7182818...

Program 1.10.3 BPIH4E8B.java

```
import javax.swing.*; // program poweres class Scanner

public class BPIH4E8C
{
    public static void main( String args[] )
    {
        // sayı e = 2.718...
        // exp(x) = 1 + x / 1! + x2 / 2! + x3 / 3! + x4 / 4! + ..
        String s;
        double x=Double.parseDouble(JOptionPane.showInputDialog("x = "));
        double power=1;
        double exponent=1;
        double factorial=1;
        double sayi=1;
        while(sayi<=200)
        {
            factorial*=sayi;
            power*=x;
            exponent+=power/factorial;
            sayi++;
        }
        s="exp("+x+") = "+exponent;
        JOptionPane.showMessageDialog(null,s,"while döngü yapısı ile ekponensiyel üs
hesabı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



FOR TEKRARLAMA YAPISI

For tekrarlama yapısı genellikle sayıları belli bir düzen içinde arttırmak için kullanılır. Genel yapısı aşağıda gösterilmiştir.programlamadaki sayı saydırmaya dönük uygulamalarda en fazla kullanılan yapıdır.

for tekrarlama yapısı

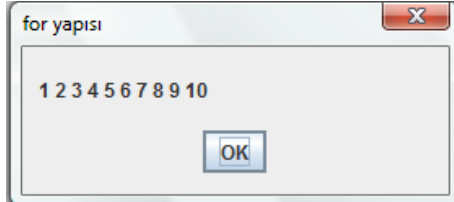
```
for(baslangıç değerleri ; boolean değişken veya sabiti ; tekrarlama değişkeni artışı)
{ boolean değişken veya sabiti true (doğru) olduğunda yapılan işlemler }
```

Program 1.10.4 : birden 100 e kadar sayıları yazdıran [for1.java](#) import java.io.*;

```
import javax.swing.*;

public class for1
{
    public static void main(String args[])
    {
```

```
String s="";
for(int saydirici=1;saydirici <= 10; saydirici++)
{ s=s+saydirici+" "; }
JOptionPane.showMessageDialog(null,s,
"for yapısı",JOptionPane.PLAIN_MESSAGE);
}
}
```

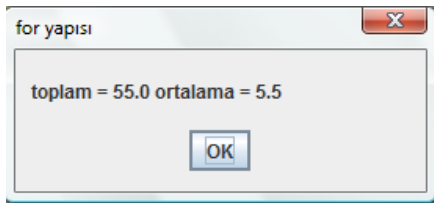


Şimdi de 1 den 10 a kadar sayıların toplamını ve ortalamasını hesaplayalım :

Program 1.10.5 Do-while tekrarlar yapısı

```
import javax.swing.*;

public class for2
{
    public static void main(String args[])
    {
        double toplam=0,ortalama;
        int saydirici;
        for(saydirici=1;saydirici <= 10; saydirici++)
        { toplam+=saydirici; }
        ortalama=toplam/10;
        String s="toplam = "+toplam+" ortalama = "+ortalama;
        JOptionPane.showMessageDialog(null,s,
"for yapısı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Bir sonraki eksersizde grafik ekranı ve * kullanarak for yapısıyla ilkel bir grafik sistemi yaratacağız.

Program 1.10.6 BPIH4E5B.java : an example of for loop

```
import javax.swing.*; // program uses class Scanner

public class BPIH4E5B
{
    public static void main( String args[] )
    {
        int x1;
        int x2;
        int y=0;
        x1=Integer.parseInt(JOptionPane.showInputDialog("ilk sayıyı giriniz: ")); // read number
        x2=Integer.parseInt(JOptionPane.showInputDialog("ikinci sayıyı giriniz: ")); // read number
        String yıldiz="*";
        String bosluk=" ";
    }
}
```

```

String toplam=yıldız;
String s="";
for(int x=x1;x<=x2;x++)
{
    y=x*x+2*x+1;
    for(int i=0;i<y;i++)
    {
        toplam = bosluk+toplam;}
    s+=toplam+"\n";
    toplam=yıldız;
}
OptionPane.showMessageDialog(null,s,"yıldız çizimi",OptionPane.PLAIN_MESSAGE);
}
}

```



Pi sayısının hesaplanmasında seri çözümü kullanılabilir $\pi=3.141592653589793$, Aşağıda verilen seriyi kullanarak pi sayısını hesaplayınız.

Program 1.10.7 BPIH4E6.java : an example of for loop

```

import javax.swing.*; // program uses class Scanner

public class BPIH4E6B
{
    // main method begins execution of Java application
    public static void main( String args[] )
    {
        //pi = 4 - 4/3 + 4/5 -4/7+4/9-4/11+...+
        String s;
        int i;
        double pi=0;
        int artieksi=-1;
        for(i=1;i < 100000;i++)
        {
            artieksi*=-1;
            pi+=4/(2.0*i-1.0)*artieksi;
        }
    }
}

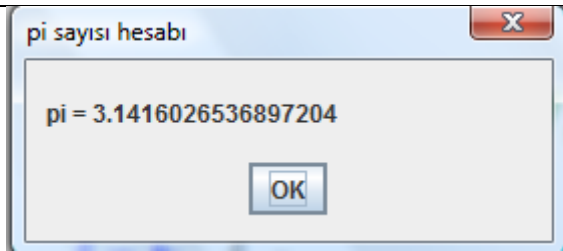
```



```

s="pi = "+pi;
JOptionPane.showMessageDialog(null,s,"pi sayısı hesabı",JOptionPane.PLAIN_MESSAGE);
} // end method main
}

```



Bir sonraki örnekte faktöriyel hesabı yapılacaktır. Faktöriyel formülü
 $n! = n * (n-1) * (n-2) * \dots * 4 * 3 * 2 * 1$ $4! = 4 * 3 * 2 * 1$ $3! = 3 * 2 * 1$

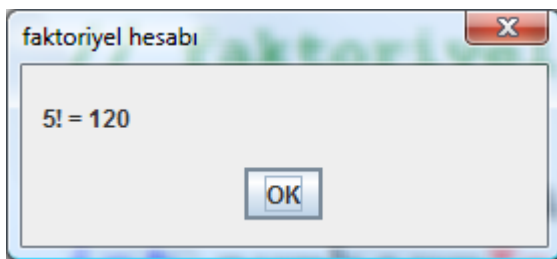
Program 1.10.8 BPIH4E7B.java : an example of for loop

```

import javax.swing.*;

public class BPIH4E7B
{
    public static void main( String args[] )
    {
        // faktöriyel hesabı
        String s;
        long factorial=1;
        int number=Integer.parseInt(JOptionPane.showInputDialog("Enter a number: ")); // read number
        for(int i=1;i <=number;i++)
        {
            factorial*=i;
        }
        s="" + number + "! = " + factorial;
        JOptionPane.showMessageDialog(null,s,"faktöriyel hesabı",JOptionPane.PLAIN_MESSAGE);
    }
}

```



Bir sonraki örnekte eksponensiyel fonksiyon seri ile hesaplanacaktır.

$$e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$e = 2.7182818\dots$

Program 1.10.9 BPIH4E8B.java

```

import javax.swing.*; // program poweres class Scanner

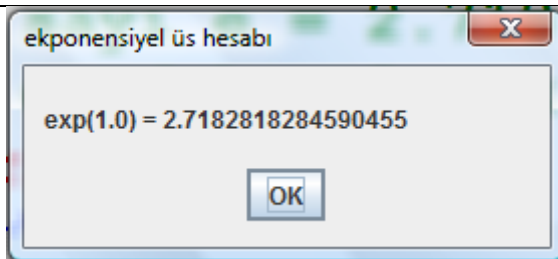
public class BPIH4E8B

```

```

{
public static void main( String args[] )
{
// sayı e = 2.718...
// exp(x) = 1 + x / 1! + x2 / 2! + x3 / 3! + x4 / 4! + ..
String s;
double x=Double.parseDouble(JOptionPane.showInputDialog("x = "));
double power=1;
double exponent=1;
double factorial=1;
for(double sayi=1;sayi<=200;sayi++)
{
factorial*=sayi;
power*=x;
exponent+=power/factorial;
}
s="exp("+x+") = "+exponent;
JOptionPane.showMessageDialog(null,s,"ekponensiyel üs hesabı",JOptionPane.PLAIN_MESSAGE);
}
}

```



Başka bir seri hesabı daha yapalım:

$$\ln(x) = \sum_{n=0}^{\infty} \frac{(-1)^{n+1} (x-1)^n}{n} = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots$$

Program 1.10.10 BPIH4E13B.java

```

import javax.swing.*;

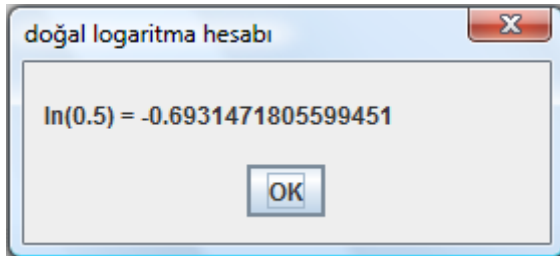
public class BPIH4E13B
{
public static void main( String args[] )
{
// e = 2.718...
// ln(x) = (x-1) - (x-1)^2/2+(x-1)^3/3+(x-1)^4/4..
// x<2 bölgesinde geçerlidir;
String s;
double x=Double.parseDouble(JOptionPane.showInputDialog(" x = "));
double power=1;
double ln=0;
double xm1=x-1;;
int plusminus=-1;
for(double n=1;n<=10000000;n++)
{
plusminus*=-1;
power*=xm1;
}
}
}

```

```

ln+=plusminus*power /n;
}
s="ln("+x+") = "+ln;
JOptionPane.showMessageDialog(null,s,"doğal logaritma hesabı ",JOptionPane.PLAIN_MESSAGE);
}
}

```



Bir sonraki örnekte bir fonksiyon çeşitli x değerleri için hesaplanacaktır.

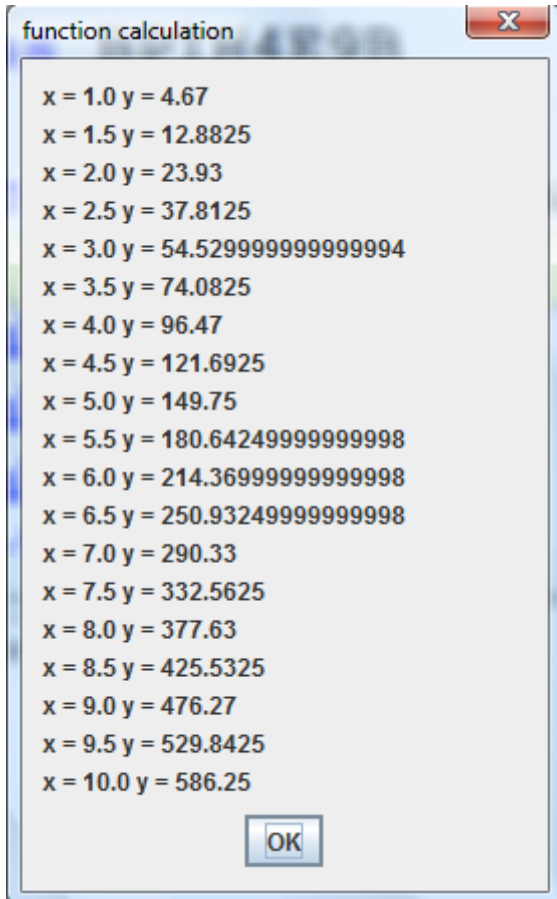
Program 1.10.11 BPIH4E9B.java

```

import javax.swing.*; // program uses class Scanner

public class BPIH4E9B
{
    public static void main( String args[] )
    {
        double x1;
        double x2;
        double dx;
        double y;
        x1=Double.parseDouble(JOptionPane.showInputDialog("Enter first double: ")); // read number
        x2=Double.parseDouble(JOptionPane.showInputDialog("Enter last double: ")); // read number
        dx=Double.parseDouble(JOptionPane.showInputDialog("Enter increment step: ")); // read number
        String s="";
        for( double x=x1;x<=x2; x+=dx)
        { y=5.67*x*x+ 2.25*x-3.25;
          s+="x = "+x+" y = "+y+"\n";
        }
        JOptionPane.showMessageDialog(null,s,"function calculation",JOptionPane.PLAIN_MESSAGE); //display
x and y
    }
}

```

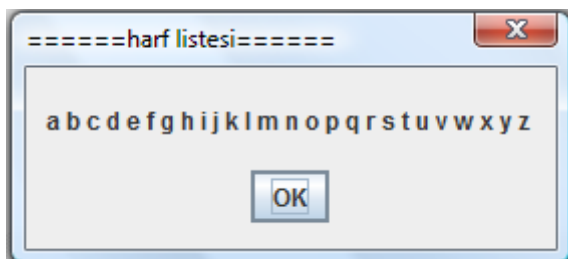


Loop yapıları kullanılarak char değişkeni de seçiştirilebilir

Program 1.10.12 BPIH4E12B.java : an example of for loop

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;

class BPIH4E12B
{
    public static void main(String args[])
    {
        String s="";
        for(char b='\u0061';b<='\u007A';b++)
        {s+=b+" ";}
        JOptionPane.showMessageDialog(null,s,"====harf listesi====",JOptionPane.PLAIN_MESSAGE);
    }
}
```



SWITCH - CASE YAPISI

Switch case yapısı çeşitli alternatifler içeren bir tablodan seçme yapma türü işlerde kullanılır. Temel yapı:

switch - case yapısı:

```
switch( degişken ismi)
{
  case degişkenin alacağı deger :
  bu deger oluştuğunda yapılacak işlemler
  break; // bu deyim döngünün bitmesini sağlar.
  case degişkenin alacağı başka bir deger :
  bu deger oluştuğunda yapılacak işlemler
  break; // bu deyim döngünün bitmesini sağlar.
  .....
  default :
  eğer yukarıdaki case kutularından hiçbirine girilemediyse
  yapılacak işlemler
  break;
}
```

şeklinde. Switch yapısından sonra degişken ismi gelir. Her case deyiminden sonra ise degişkenin alacağı yeni bir deger alternatifi verilir. Eğer switch yapısında verilen alternatif hiçbir case bloğunda tanımlanmamışsa default bloğuna gider ve buradaki işlemler gerçekleştirilir. Case ve default blokları break deyimini ile sona erer. break deyimini herhangi bir döngü yapısından döngü bitmeden çıkma amacıyla da kullanılabilir. break yapısıyla döngünün dışına çıkma amacıyla diğer döngü yapılarında da kullanılabilir (if, for while gibi), fakat bu tür break kullanılarak döngü dışına çıkılması yapısal programlama teknikleri tarafından tavsiye edilmez. Bilgisayar listelerini yapısalıktan uzaklaştırıp, spagetti programlama dediğimiz karmakarışık programlamaya yol açabilir. Döngü yapılarından standart kontrolleri yoluyla çıkılması tavsiye edilir.

```
int i=0;
while(true)
{
  i++;
  if(i==3) break;
  System.out.println("Bu yazı uc kere tekrar edecektir");}
```

Program 1.27 de verilen switchApplet.java programı switch deyiminin nasıl kullanıldığını göstermektedir. Programcılara göre kullanılma sıklığı degişebildiği varsayılsa bile, switch yapısı şu ana kadar bahsedilen yapılar içinde en az kullanılanıdır.

Program 1.10.13 switch-case tekrarlama yapısı

```
import javax.swing.JOptionPane;

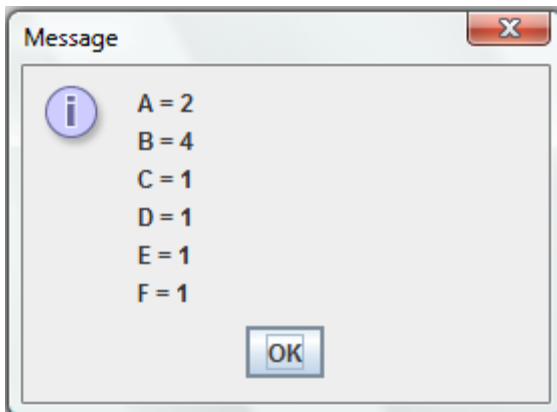
class switchcaseyapisi
{
  public static void main (String args[])
  { int Asayisi=0,Bsayisi=0,Csayisi=0,Dsayisi=0,Esayisi=0,Fsayisi=0;
    for(int i=0;i<10;i++)
    {
      String not=JOptionPane.showInputDialog("Öğrencinin notunu giriniz (a,b,c,d,e,f) : ");
      switch(not)
      { case "A": case "a": {Asayisi++;break;}

```

```

    case "B": case "b": {Bsayisi++;break;}
    case "C": case "c": {Csayisi++;break;}
    case "D": case "d": {Dsayisi++;break;}
    case "E": case "e": {Esayisi++;break;}
    case "F": case "f": {Fsayisi++;break;}
}
}
String s="A = "+Asayisi+"\n";
s+= "B = "+Bsayisi+"\n";
s+= "C = "+Csayisi+"\n";
s+= "D = "+Dsayisi+"\n";
s+= "E = "+Esayisi+"\n";
s+= "F = "+Fsayisi+"\n";
JOptionPane.showMessageDialog(null,s);
}
}

```



1.11 ARİTMETİK İŞLEMLERDE DEĞİŞKEN TÜRÜ DEĞİŞTİRME (CASTİNG) OPERASYONU

C veya C++ dilinde bir değişken türünden değişkeni diğer değişken türünden değişkene direk olarak atama mümkündür. örneğin

```

char a='x';
int b=a;

```

işlemi c++ da geçerli bir işlemdir. Javada ise bu işlem hata verir. fakat aynı işlem

```

char a='x';
int b=(int)a;

```

şeklinde gerçekleştirilebilir (**int**) deyimi orijinal olarak char olarak tanımlanan a değişkeninin değerini int türüne dönüştürmüş sonra da int türünden b değişkenine yüklemiştir. Bu dönüşüm işlemlerini tüm değişken türleri için kullanabiliriz, yalnız burada programcının çok dikkatli olması ve değişken türlerini, ve sınırlarını iyi tanması gerekmektedir. Değişken türlerini değiştirirken bazı bilgiler kaybolacağı gibi hiç istemediğimiz bir sonuca ulaşma olasılığımız da mevcuttur.

Program 1.11.1cast işlemi

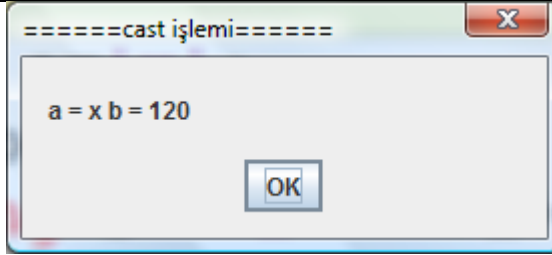
```

import javax.swing.*;

class cast
{
public static void main(String arg[])
{
char a='x';

```

```
int b=(int)a;
String s=" a = "+a+" b = "+b;
JOptionPane.showMessageDialog(null,s,"=====cast işlemi=====","JOptionPane.PLAIN_MESSAGE");
}
}
```



1.12 SAYI DEĞİL VE SONSUZ SONUÇLARI

Java gerçek sayı değişken türleri sayı değil (NaN) ve artı sonsuz (Pozitif infinity) ve eksi sonsuz(negative infinity) sonuçları verebilirler. sayı değil (NaN) sonucu sıfır bölü sıfır, sıfır çarpı sonsuz gibi işlemlerden çıkar. sonsuz işlemi ise sayı bölü sıfır gibi işlemlerden çıkar. Bu işlemler sadece float ve double değişkenleri için tanımlıdır. Tamsayı değişkenler sıfıra bölünemezler.

1.13 EKŞERSİZLER

EX 1. Konsol alanında “Merhaba benim ismim” ve sizin isminizi yazacak bir bilgisayar programı geliştiriniz.

```
public class BPIH1CEX2
{
public static void main( String args[] )
{
System.out.println( "Merhaba benim ismim Ali!" );
}
}
```

EX 2. Grafik ortamında Merhaba benim ismim sonra isminizi yazacak bir program yazınız.

```
import javax.swing.*; // program uses class JOptionPane in swing library
public class BPIH1CEX3
{
// main method begins execution of Java application
public static void main( String args[] )
{
JOptionPane.showMessageDialog(null, "Merhaba benim ismim Ali!");
} // end method main
} // end class BPIH1E3
```

EX 3. Grafik ortamında önce isminizi giriniz sonra “Merhaba benim ismim” ve sonra isminizi yazacak programı yazınız

```
import javax.swing.*; // program uses class JOptionPane in swing library

public class BPIH1CEX3
{
public static void main( String args[] )
{
String isim=JOptionPane.showInputDialog("isminizi giriniz: ");
JOptionPane.showMessageDialog(null,"Merhaba benim ismim "+isim+"!");
}
```

```
}}
```

EX 4. Ekrandan(konsol) iki deęişken girecek ve bunları bölecek bir program yazınız.

```
import java.util.Scanner;

public class BPIH1CEX4
{
public static void main( String args[] )
{
Scanner input = new Scanner( System.in );
int number1;
int number2;
int division;
System.out.print( "ilk tamsayıyı giriniz: " );
number1 = input.nextInt(); // read first number from user
System.out.print( "ikinci tamsayıyı giriniz: " );
number2 = input.nextInt(); // read second number from user
division = number1/number2;
System.out.println( "Bölme deęeri = "+ division );
}
}
```

EX 5. Grafik ortamda ikisayı giriniz ve bunların bölümünü çıktı olarak

```
import java.util.Scanner; // program uses class Scanner
import java.util.Locale;

public class BPIH1CEX5
{
// main method begins execution of Java application
public static void main( String args[] )
{
// create Scanner to obtain input from command window
Scanner input = new Scanner( System.in );
Locale us=new Locale("US");
input.useLocale(us);
double number1; double number2;
double division;
System.out.print( "ilk sayıyı giriniz: " );
number1 = input.nextDouble();
System.out.print( "ikinci sayıyı giriniz: " );
number2 = input.nextDouble();
division = number1/number2;
System.out.println("bölme = "+ division );
}
}
```

EX 6. Aşağıdaki çıktıyı grafik ortamında oluşturan bir program yazınız

$$\int (\alpha^2 + \beta^2) d\alpha$$

EX 7. Aşağıdaki programı yazınız, derleyiniz ve çalıştırınız

```
public class doubleA1
```



```

{   public static void main(String[] args)
    {
        double A=1;
        A++;
        double A1=A;
        A1+=3;
        double A2=A1;
        A2--;
        double A3=A2;
        A3*=5;
        System.out.println("A="+A+"A1="+A1+"A2="+A2+"A3="+A3);
    }
}

```

EX 8. Aşağıdaki kodu çalıştırın ve nerede hata olduğunu açıklayınız.

```

import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;

class finalC
{   public static void main(String args[])
    {
        final double pi = 3.141592653589793;
        double b=4*pi;
        pi=2*pi;
        String s="b = "+b+" pi = "+pi;
        JOptionPane.showMessageDialog(null,s,"final deyimi",JOptionPane.PLAIN_MESSAGE);
    }
}

```

EX 9. Boolean tipi değişken kullanarak AND(&&), ve OR(||) tabloları oluşturunuz.

true && true = true
true && false = false
false && true = false
false && false = false

true || true = true
true || false = true
false || true = true
false || false = false

çıktıları grafik ortamında alınız. (JOptionPane.showMessegeDialog)

EX10. Bir renk seçini(String tipi değişken) . Eğer renk siyah, kahverengi, gri veya beyazsa soğuk bir renk seçtiniz, eğer renk sarı, yeşil, kırmızı ise sıcak bir renk seçtiniz yazdırınız.

EX 11. 4 sayı giriniz, bu sayıların ortalamasını hesaplayınız. Eğer her sayı ortalama sayıdan küçükse sınıfta kaldın, büyükse geçtin yazdırınız.

EX 12) 0 dan 100 e kadar tek sayıların **(1,3,5,7,9....99)** while döngüsü kullanarak ortalamasını alan bir program yazınız

EX13) $A = \sum_{n=1}^{10000} \frac{1}{n^2}$ serisini hesaplayan (for döngüsü kullanınız) bir program yazınız A sayısının

$\frac{2}{6} = 1.6449$ değerine eşit olduğu iddia edilmektedir, kanıtlayınız.

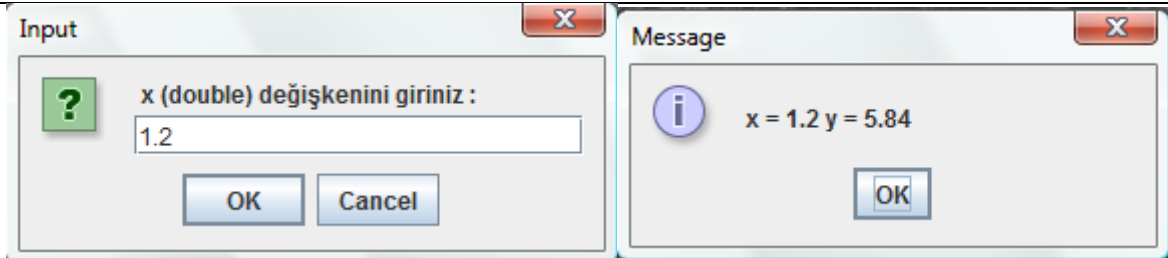
EX14) $y=f(x)=\frac{1}{x^2}$, fonksiyonunu x'in 1 den 20 'e kadar değerlerini do-while döngüsü ile hesaplayacak bir program yazınız.

PROBLEMLER

HW 1 double tipi x değişkenini grafik ekranından giren ve $y=x*x+2*x+2$ fonksiyonunu hesaplayan, x ve y nin değerini ekrana grafik ortamında yazdıran bir program yazınız

```
import javax.swing.*; // program uses class JOptionPane in swing library

public class BPIH1HW1
{
    // main metodu
    public static void main( String args[] )
    {
        double x;
        double y;
        x=Double.parseDouble(JOptionPane.showInputDialog(" x (double) değışkenini giriniz : "));
        y=x*x+2*x+2;
        JOptionPane.showMessageDialog(null,"x = "+x+" y = "+y ); // x ve y değerlerini göster
    } // metodunun sonu
} // BPIH1HW1 sınıfının sonu
```



HW 2 int tipi x değışkenini konsol ekranından giren ve $y=x*x+2/x+2$ fonksiyonunu hesaplayan, x ve y nin değerini ekrana konsol ortamında yazdıran bir program yazınız

```
import java.util.Scanner;

public class BPIH1HW2
{
    // main mmetodu
    public static void main( String args[] )
    {
        // Scanner okuma kanalı aç
        Scanner input = new Scanner( System.in );
        int x;
        int y;
        System.out.print( "x (integer) = " ); // ekrana yaz
        x = input.nextInt(); // x değışkenini ekrandan oku
        y=x*x*x+2/x+1;
        System.out.println( "x = "+x+" y = "+y); // x ve y değışkenini göster
    }
}
```

```
} // end method main  
  
} // BPIH1HW2 sınıfı sonu
```

```
----- Capture Output -----  
> "C:\turhan\java\bin\javaw.exe" BPIH1HW2  
x (integer) = 2  
x = 2 y = 10  
  
> Terminated with exit code 0.
```

HW3 Konsol ortamında satırbaşı da yaparak(\n) aşağıdaki metni yazdırınız

“Merhaba benim ismim (Your name). İlk defa bilgisayar dersi alıyorum”

“bilgisayar dili, özellikle Java dili, öğrenmek çok güzel bir şey”

HW4 Grafik çıktı ortamında ortamında satırbaşı da yaparak(\n) aşağıdaki metni yazdırınız

“Merhaba benim ismim (Your name). İlk defa bilgisayar dersi alıyorum”

“bilgisayar dili, özellikle Java dili, öğrenmek çok güzel bir şey”

HW 5 ISO Unicode karakter kodları (char) kullanarak aşağıdaki metni grafik ortamında yazdırınız

$$\iint (\alpha^2 + \beta^2) \partial\alpha\partial\beta$$

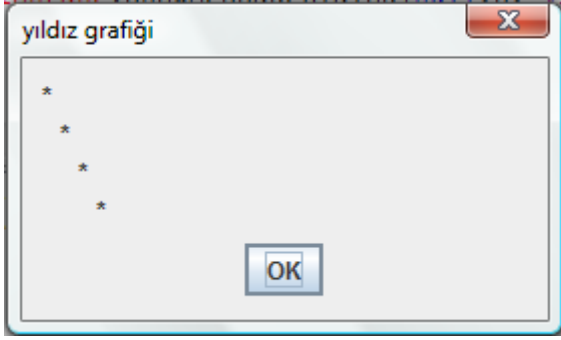
JOptionPane.showMessageDialog metodu kullanılacaktır.

The name of the program and class will be **BPIH2HW1.java**

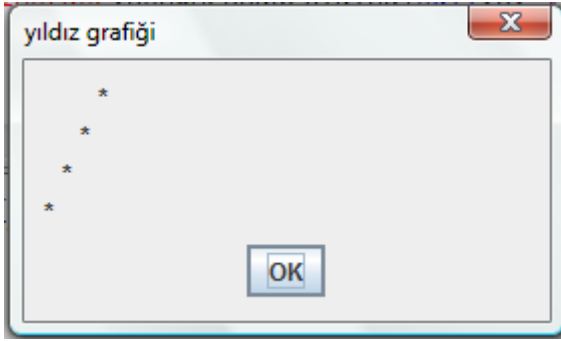
HW6 Aşağıdaki programı inceleyiniz

```
import java.io.*; //java girdi cikti sinifini cagir  
import javax.swing.*;  
  
class star  
{  
    public static void main(String args[])  
    {  
        final String yildiz = "*";  
        String s1=yildiz+"\n";  
        String s2=" "+s1;  
        String s3=" "+s2;  
        String s4=" "+s3;  
        String s=s1+s2+s3+s4;  
        JOptionPane.showMessageDialog(null,s,"yıldız grafiği",JOptionPane.PLAIN_MESSAGE);  
    }  
}
```

Program çıktısı aşağıdaki gibi olmalıdır :



Programı yazarak java ortamında çalıştırınız. Şimdi programı aşağıdaki girdiyi verecek şekilde değiştiriniz



HW7. Aşağıdaki programı yazınız, derleyiniz ve çalıştırınız

```
import javax.swing.JOptionPane;

public class intA1
{
    public static void main(String[] args)
    {
        int A=1;
        A++;
        int A1=A;
        A1+=3;
        int A2=A1;
        A2--;
        int A3=A2;
        A3*=5;
        int A4=A3;
        A4/=5;
        String s="A = "+A;
        s+=" A1 = "+A1;
        s+=" A2 = "+A2;
        s+=" A3 = "+A3;
        s+=" A4 = "+A4;
        String s1="Tamsayı ve String String toplama";
        JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
    }
}
```

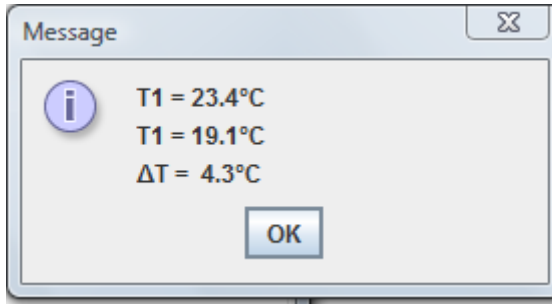
HW8. Aşağıdaki programı yazınız, derleyiniz ve çalıştırınız

```
import javax.swing.JOptionPane;
import java.util.Locale;
```

```

public class BPIH2HW4A
{
    // main method begins execution of Java application
    public static void main( String args[] )
    {
        Locale us=new Locale("US");
        double T1;
        double T2; //
        double dT; // sum of number1 and number2
        String a1="İlk sıcaklığı giriniz "+'\u000B'+ "C : ";
        T1 = Double.parseDouble(JOptionPane.showInputDialog(a1));
        String a2="İkinci sıcaklığı giriniz "+'\u000B'+ "C : ";
        T2 = Double.parseDouble(JOptionPane.showInputDialog(a2));
        dT = T1 - T2; // sıcaklık farkı
        String s1="T1 = %4.1f'+'\u00B0'+ "C \n";
        String s2="T1 = %4.1f'+'\u00B0'+ "C \n";
        String s3='\u0394'+ "T = %4.1f'+'\u00B0'+ "C \n";
        String s=s1+s2+s3;
        String s4=String.format(us,s,T1,T2,dT);
        JOptionPane.showMessageDialog(null,s4); // sonucu göster
    } // metodun sonu
} // BPIH2HW4A sınıfının sonu

```



HW9) Ekrandan iki gerçek sayı giriniz. Sayıları karşılaştırınız ve sonuçlarını ekrana yazdırınız. Örneğin eğer girilen sayılar 1.23 ve 2.34 ise

1.23 < 2.34 true

1.23 > 2.34 false

1.23 == 2.34 false

Sonucunu elde edin

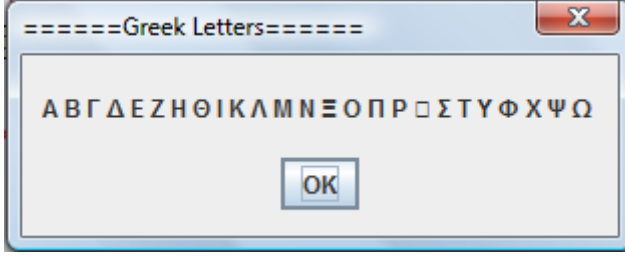
HW10) Bir öğrencinin ismini, ödev notunu, vize notunu ve final notunu öğrenin ve geçme notunu hesaplayarak yazdırın

Geçme notu = 0.25*ödev notu + 0.3*vize notu+0.4*final notu

Eğer geçme notu 60 ın altında ise kaldı, üstünde ise geçti yazdırın.

HW11) Bir isim girin, eğer isim Mehmet, Ahmet, Hüseyin, Ali veya veli ise, Tebrikler özel çekilişimizde birmilyon TL kazandınız yazdırınız. Tüm diğer isimler için üzgünüz çekilişte kaybettiniz yazdırınız.

HW12) While döngüsü kullanarak çıktıda gösterilen karakter dizini yazdırınız.



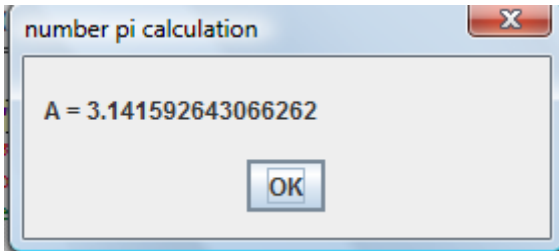
HW13) for döngüsü kullanarak verilen A dizini hesaplayınız

$$A = 2 \cdot \sum_{k=1}^{\infty} \frac{1}{(2k)^2}$$

Note : $b=(2.0*k)*(2.0*k)$; değişken dönüşümünü yaparak formülün yazılımını kolaylaştırabilirsiniz.

```
import javax.swing.*;

public class BPIH4HW2
{
    public static void main( String args[] )
    {
        String s;
        double n;
        double A=2.0;
        double b=1.0;
        for(double k=1.0;k < 100000000.0;k++)
        {
            b=(2.0*k)*(2.0*k);
            A*=b/(b-1);
        }
        s="A = "+A;
        System.out.println(s);
        JOptionPane.showMessageDialog(null,s,"pi hesabı",JOptionPane.PLAIN_MESSAGE);
    }
}
```



HW14) ln(x) hesaplamak için verilen seri formüllerinden birisi

$$\ln(x) = \sum_{n=1}^{\infty} \frac{1}{n} \left(\frac{x-1}{x+1} \right)^n$$

$y = \frac{x-1}{x+1}$ formundadır. do-döngü yapısı kullanarak seriyi hesaplayınız. Not: $z=y*y$ dönüşümü uygulanırsa seri hesabı kolaylaşır.

$$\ln(x) = \sum_{n=1}^{\infty} \frac{1}{n} z^n$$

```
import javax.swing.*;

public class BPIH4HW3
```

```
{
public static void main( String args[] )
{
//  $\ln(x) = 1+y^2/3+y^4/5+y^6/7+\dots$   $y=(x-1)/(x+1)$ 
String s;
double x=Double.parseDouble(JOptionPane.showInputDialog(" x = ")); // sayıyı oku
double us=1;
double ln=1;
double y=(x-1)/(x+1);
int plusminus=-1;
double n=1;
do
{
us*=y*y;
ln+=us /(2.0*n+1.0);
n++;
} while(n<=10000000);
ln*=2*y;
s="ln("+x+") = "+ln;
JOptionPane.showMessageDialog(null,s,"doğal logaritma hesabı",JOptionPane.PLAIN_MESSAGE);
}
}
```

BÖLÜM 2 : SINIF KAVRAMINA GİRİŞ VE METODLAR

2.1 JAVA API KÜTÜPHANESİ

Gerçek problemlerin boyutları genelde birinci bölümde gördüğümüz problemlere göre çok daha büyüktür. Bu yüzden genelde programlar küçük parçalara bölünerek daha kolaylıkla anlaşılır şekle getirilmeye çalışılır. Buna *böl parçala ve yönet tekniği* de diyebiliriz. Bu küçük parçalara javada metot adı verilir. Diğer bilgisayar dillerinde subroutine, fonksiyon gibi adlara da raslıyoruz. Java programları yeni metotları içeren java sınıflarıyla(class) java sınıf kütüphanelerinde yer alan metotların birleşmesinden oluşur. Java kütüphanesinde matematik işlemlerinden grafik, ses, giriş çıkış işlemleri, yanlış kontrolü gibi birçok konularda yazılmış metotlar mevcuttur. Java metotlar koleksiyonu **Java API (Application Programming Interphase-Uygulama Program Arayüzü)** adını alır. Java sınıf kütüphanesi adı da verilir. Şüphesiz bunun dışında java metotlar koleksiyonlarının olduğu çeşitli diğer kütüphaneler de mevcuttur.

Java API kütüphanelerinin bazıları şunlardır :

java.applet : (java applet paketi) Java applet paketi applet grafik ortamını sağlar (grafik programlarının internet browser programları aracılığıyla görüntülenebilmesinin temellerini oluşturur.). Bu sınıfa paralel olarak Java 1.2 de grafik applet arayüz birimi **javax.swing.JApplet** geliştirilmiştir.

java.awt : (java window araçları paketi) Bu sınıf grafik arayüzleri tanımlamaya yarar. Java 1.2 de paralel sınıf **javax.swing** geliştirilmiştir.

java.awt.color: renk tanımlar

java.awt.datatransfer: bilgisayarın geçic hafızasıyla java programı arasında bilgi akışını sağlar.

java.awt.dnd : ekle ve çalıştır türü programlar arasında geçiş yapabilen paketlerden oluşur.

java.awt.event : grafik programlama ortamıyla kullanıcı arasındaki ilişkiyi sağlar. Örneğin grafik penceresinde return tuşuna basıldığında veya farenin düğmesine basıldığında ne yapılması gerektiğini belirler. **Java.awt** ve **javax.swing** sınıflarının ikisiyle de kullanılır.

java.awt.font : yazı fontları ve bu fontların değiştirilmesiyle ilgili programları içerir.

java.awt.geom : iki boyutlu geometrik şekilleri tanımlar.

java.awt.im : java programlarına çince, japonca ve kore dilinde girdi girebilmek için tanımlanmış bir arabirimdir.

java.awt.image : java görüntü işleme paketi

java.awt.image.renderable : bir program içindeki resimleri depolama ve değiştirme için gerekli olan programları içerir.

java.awt.print : java programlarından yazıcılara yazım için bilgi aktaran paketleri içerir.

java.beans : java fasulyeleri paketi, bu paketin detayları için ilgili bölümü inceleyiniz.

java.beans.beancontext : tekrar kullanılabilen ve otomatik olarak birleştirilebilen program parçacıkları tanımlar.

java.io : dosya ve ekrandan girdi ve çıktı yapmak için gerekli program paketleri tanımlanır.

java.lang : bu paket birçok temel program içerir Bunlardan biri olan Math paketi aşağıda açıklanacaktır. Lang paketi java programlarınca otomatik olarak kullanılır. İport deyimisiyle iave edilmesi gerekmez.

java.lang.ref : bu paket java programıyla garbage collector (çöp toplayıcısı) denilen otomatik kullanılmayan hafıza temizleme programıyla ilişkiyi sağlar.

java.lang.reflect : bu paketdeki programlar java programının çağrıldığında içinde bulunan değişken ve metotları dinamik olarak tespit etmesini ve kullanma izni almasını sağlar.

java.math : bu sınıf değişebilir hassasiyette aritmetik işlemler yapmak için gereken tanımları ve sınıfları içerir.

java.net : ağ işlemlerinde kullanılan çeşitli paketleri tanımlar

java.rmi , java.rmi.activation, java.rmi.dgc, java.rmi.registry, java.rmi.server: programların ve metodların uzaktan çalışabilmeleri için gerekli tanımlamaları içerir.

java.security, java.security.acl, java.security.cert, java.security.interfaces, java.security.spec : güvenlikle ilgili programlar demetini içerir.

java.sql : database programıyla java programlarını bağlantısını sağlar.

java.text : sayı karakter ve string tarih gibi işlemlerle ilgili programlar demeti içerir. Örneğin çok dilli programlar yazmayı sağlayan **internalisation** paketi bu paket içinde yer alır.

java.util : bu sınıf tarih ve zamanla ilgili fonksiyonlar, tesadüfi sayı üreticileri, **StringTokenizer** gibi programları barındırır.

java.util.jar bu paket java programlarını paketlemeye yarayan jar yapılarını tanımlamada gerekli olan programları tanımlar.

java.util.zip : bu paket programları sıkıştırmaya yarayan programları tanımlar.

Java.accessibility : bu paket özürülülerin kullanabileceği ara üniteler tanımlar. Ekran okuyucular ve ekran büyüteçleri gibi.

javax.swing : grafik kullanıcı arabirimi tanımlar. Bu sınıftaki tanımlar aracılığıyla ekrandan grafik ortamında veri transferi kolaylıkla yapılabilir.

javax.swing.borders : grafik arabirimi sınır yaratımında çeşitli sınır şekilleri yaratmakta kullanılır.

javax.swing.colorchooser : renk seçimi için tanımlanmış grafik kullanıcı arabirimini tanımlar.

Java API dosyalarındaki metotların sadece isimlerinin listesi yaklaşık 200 sayfa boyutunda bir liste tuttuğundan burada sadece yeri geldiğine programları kullanırken örneklerde gerekli API isimlerini göreceğiz. Şu ana kadar kullandığımız java programlarında java.applet, java.awt, java.io javax.swing, javax.swing.JOptionPane gibi bazı API paketlerini kullandık. Java swing (javax.swing) grafik pencere(window) ortamında programlama için gerekli girdi çıktı grafik ortamlarını içerir. java.io paketi javaya dışarıdan(dosya,ekran v.b) bilgi giriş çıkışını ayarlar. Java metodlarının tamamının listesini (İngilizce olarak) Java doküman kütüphanesindeki api alt gurubunda bulabilirsiniz. Java doküman kütühanesi

<http://docs.oracle.com/javase/6/docs/api/>

adresinden detaylı olarak incelenebilir veya kendi bilgisayarınıza aktarılabilir. Günümüzde kütüphanenin tüm detaylarına internet ortamında erişilebildiğinden bilmediğiniz bir kütüphanenin detaylı bilgilerine arama motorlarından sorarak ta anında ulaşabilirsiniz.

Java dilinde en çok kullanılan API sınıflarından birisi Math sınıfıdır. Bu sınıf(class) java.lang paketinde yer alır. Java.lang paketi java programı açılırken otomatik olarak çağırılır. Bütün diğer paketler **import** deyimi kullanılarak programa ilave edilirler. Math sınıfında tanımlanan metotların bazıları Şekil 2.1 de listelenmiştir. Tablodaki x veya y değişkenleri **double** değişken türündendir. Math sınıfında(class) iki tane de sabit

tanımlanmıştır. Bu sabitlerden birisi **Math.PI** 3.14159265358979323846 (pi) sayısına eşittir. Diğeri **Math.E** 2.7182818284590452354 (e) sayısına eşittir.

Tablo 2.1 Math sınıfında(class) çalışan metotlardan bazıları

Metot	Tanım	Örnek
abs(x)	x değişkeninin mutlak değeri türkçesi : Mutlak . x değeri double, float ve int olarak tanımlanmıştır	Math.abs(9.2) = 9.2 Math.abs(-9.2) = 9.2
ceil(x)	x değişkenini bir üst tamsayıya dönüştürür (türkçesi : tavan)	Math.ceil(9.2) = 10 Math.ceil(-9.8) = -9
floor(x)	x değişkenini bir alt tamsayıya dönüştürür. (türkçesi : taban)	Math.floor(9.2) = 9 Math.floor(-9.8) = -10
cos(x)	x in trigonometrik cosünüsü (x radyan)	Math.cos(1.0)=0.54030230568
sin(x)	x in trigonometrik sinüsü (x radyan cinsinden tanımlanmıştır.)	Math.sin(1.0)=0.8414709840709
tan(x)	x in trigonometrik tanjantı (x radyan)	Math.tan(1.0)=1.557040724655
acos(x)	x in trigonometrik cosünüsünün tersi (sonuç radyan)	Math.acos(0.54030230568)=1.0
asin(x)	x in trigonometrik sinüsü (sonuç radyan)	Math.asin(0.8414709840709)=1.0
atan(x) atan2(x,y)	x in trigonometrik tanjantı (sonuç radyan)	4.0*Math.atan(1.0)=3.14159265359 4.0*Math.atan2(1.0,1.0)=3.14159265359
exp(x)	e^x e(2.7182818284590452354) nin x inci kuvveti	Math.exp(1.0) =2.718281828459 Math.exp(2.0) =7.389056098931
expm1(x)	$e^x - 1$	Math.expm1(1.0) =1.718281828459 Math.expm1(2.0) =6.389056098931
log(x)	e tabanına göre logaritma	Math.log(2.718281828459)=1.0
log10(x)	10 tabanına göre logaritma	Math.log(10.0)=1.0
log1p(x)	(x+1) in e tabanına göre logaritması	Math.log(1.718281828459)=1.0
pow(x,y)	x in y inci kuvveti	Math.pow(3,2)=9.0
max(x,y)	x ve y nin büyük olanı. x değeri double, float ve int ve long olarak tanımlanmıştır	Math.max(2.3,12.7)=12.7
min(x,y)	x ve y nin küçük olanı. x değeri double, float ve int ve long olarak tanımlanmıştır	Math.min(2.3,12.7)=2.3
random()	Raslantısal sayı (0 ile 1 arasında)	Math.random() = 0 ila bir arasında her sayı çıkabilir
scalb(x,n)	$x*2^n$ değerini hesaplayarak döner burada x double veya float olabilir n int türü değişkendir	Math.scalb(3.0,2)=12.0
signum(x)	Double veya float tipi x değişkeninin işaretini döner eğer x 0 ise 0, 0 dan büyükse 1, 0 dan küçükse -1 değerini verir	Math.signum(-2)=-1, Math.signum(0)=0.0
sinh(x)	x in hiperbolik sinüsü	Math.sinh(1.0)= 1.175201
tanh(x)	x in hiperbolik tanjantı	Math.tanh(1.0)= 0.761594
acosh(x)	x in hiperbolik cosünüsünün tersi	Math.acosh(1.543081)=1.0
asinh(x)	x in in hiperbolik sinüsünün tersi	Math.asinh(1.175201)=1.0

Program 2.1 de Math sınıfındaki metotları kullanan [Matematik1](#) sınıfını görüyoruz.

Program 2.1 Math sınıfında(class) çalışan metotlardan bazıları gösteren [matematik1.java](#) programı

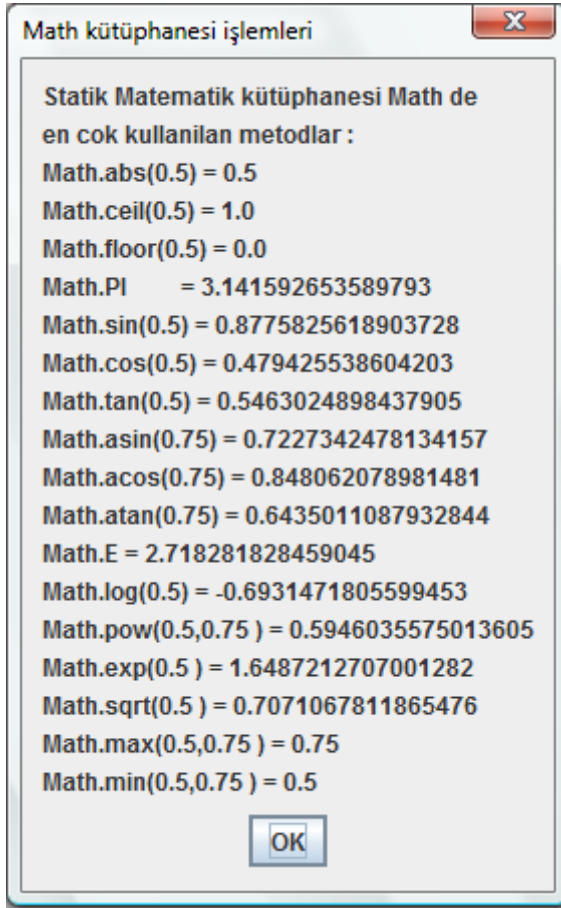
```
import javax.swing.JOptionPane; // giriş çıkı
```

```

class matematik1
{
    public static void main (String args[])
    {
        double x,y;
        String s="";
        x=Double.parseDouble(JOptionPane.showInputDialog
("Bir gerç k sayı giriniz : "));
        y=Double.parseDouble(JOptionPane.showInputDialog
("İkinci bir gerç k sayı giriniz : "));

        s += "Statik Matematik k t phanesi Math de \n";
        s += "en cok kullanılan metodlar :\n" ;
        // Math.abs(3.2)=3.2  Math.abs(-3.2)=3.2
        s += "Math.abs("+x+") = "+Math.abs(x)+"\n";
        // Math.ceil(9.2)=10  Math.ceil(-9.8)=-9
        // Math.ceil(x) x den buyuk en k c kk tam sayıya yuvarlar
        s += "Math.ceil("+x+") = "+Math.ceil(x)+"\n";
        // Math.floor(9.2)=9  Math.floor(-9.8)=-10
        // Math.floor(x) x den k c k en buyuk tam sayıya yuvarlar
        s += "Math.floor("+x+") = "+Math.floor(x)+"\n";
        // Trigonometrik fonksiyonlar sin(x) cos(x) tan(x)  x:radyan
        // Math.PI pi sayisi
        s += "Math.PI = "+Math.PI+"\n";
        s += "Math.sin("+x+") = "+Math.sin(x)+"\n";
        s += "Math.cos("+x+") = "+Math.cos(x)+"\n";
        s += "Math.tan("+x+") = "+Math.tan(x)+"\n";
        //Trigonometrik fonksiyonlar asin(y) acos(y) atan(y)
        // asin,acos,atan : radyan
        s += "Math.asin("+y+") = "+Math.asin(y)+"\n";
        s += "Math.acos("+y+") = "+Math.acos(y)+"\n";
        s += "Math.atan("+y+") = "+Math.atan(y)+"\n";
        // Math.log(x) dogal logaritma (e) tabanınaa gore
        // Math.E e sayisi = 2.718281828...
        s += "Math.E = "+Math.E+"\n";
        s += "Math.log("+x+") = "+Math.log(x)+"\n";
        // Math.pow(x,y) x in y inci kuvveti
        s += "Math.pow("+x+", "+y+") = "+Math.pow(x,y)+"\n";
        // Math.exp(x) Math.E=e=2.718281828.. in x inci kuvveti
        s += "Math.exp("+x+") = "+Math.exp(x)+"\n";
        // Math.sqrt(x) x in kare koku
        s += "Math.sqrt("+x+") = "+Math.sqrt(x)+"\n";
        // Maximum - minimum functions
        // Math.max(x,y)  Math.min(x,y)
        s += "Math.max("+x+", "+y+") = "+Math.max(x,y)+"\n";
        s += "Math.min("+x+", "+y+") = "+Math.min(x,y)+"\n";
        JOptionPane.showMessageDialog(null,s,
"Math k t phanesi iřlemleri",JOptionPane.PLAIN_MESSAGE);
    }
}

```



Şekil 2.1 matematik1SW.java programının çıktısı

2.2 METOTLAR

Metotlar java programlarının ana parçalarıdır. Metotlar sınıfların(class) içinde yer alan küçük program parçacıklarıdır. Metotların çoğunda değişken parametreler metotlar ve sınıflar arasında iletişimi sağlarlar. Ayrıca her metotun kendine özgü değişkenleri de vardır. Metot yapısının ana sebebi programları modüler hale getirmektir. Aynı zamanda aynı program parçacığının tekrarlanması önlemeyi de sağlar. Her metot çağrıldığı program parçacığına belli bir değişkeni aktarabilir. Metotların tanımlarında aktardıkları değişken türü tanımlanır. Eğer metot hiçbir değişken aktarmıyorsa **void** sözcüğüyle tanımlanır. Metotların genel tanımı aşağıdaki gibidir. Parantez içindeki terimler kullanılmıyabilir.

Genel Metot tanımı

(public) (static) sınıf değişken türü sınıf ismi(sınıf değişken girdi listesi)

```
{  
sınıf içinde geçerli değişken tanımları  
Metotun ana gövdesi  
return metot çıktı değişkeni  
}
```

Örnek olarak aşağıdaki metodu verebiliriz :

```
Public static double metotorneği(int x,double y)  
{  
double z = 3.5;  
double f ;  
f = z*x*x+y;
```

```
return f;
}
```

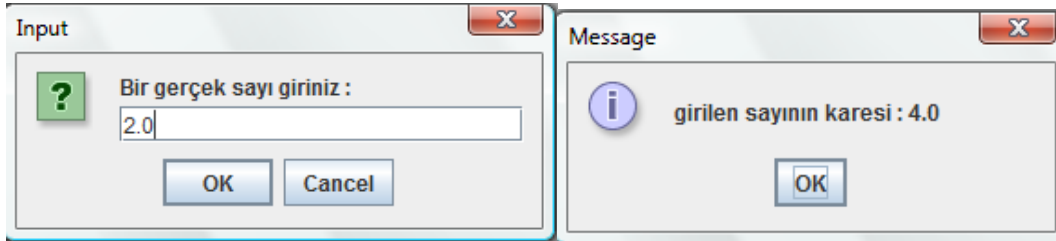
Bu metoddaki x ve y gerçek (double) değişkenleri metodun girdi değişkenleridir. z ve f değişkenleri metodun yerel değişkenleridir ve bu metod dışında tanımları yoktur. Metod f değişkeninin değerini çıktı olarak metodun dışına aktarmaktadır. Eğer yukarıdaki metod aynı sınıfın içindeki başka bir metotta veya main metotunda aşağıdaki gibi bir örnekte kullanılırsa,

```
double z = 2.5;
double r ;
int i = 3;
r = metotornegi(i,z);
System.out.println("r = "+r);
```

r = 34 sonucuna ulaşılır. Şimdi metodların statik metod olarak kullanılmasını görelim.

Program 2.2.1 : [metotornegil.java](#) programında kare statik metodunun kullanılması

```
import javax.swing.*;
public class metotornegil
{
    public static double kare(double x)
    { return x*x; }
    public static void main(String[] args)
    { double sayi=Double.parseDouble(JOptionPane.showInputDialog("Bir gerçek sayı giriniz : "));
      JOptionPane.showMessageDialog(null,"girilen sayının karesi : "+kare(sayi));
    }
}
```



Şekil 2.2.1 [metotornegil.java](#) programında kare metodunun kullanılması

Programda tanımlanan **public static double kare(double x)** metodu, **public static void main(String[] args)** metodundaki

```
JOptionPane.showMessageDialog(null,"girilen sayının karesi : "+kare(sayi));
```

işleminde çağırılmıştır. Bilgisayar **kare(sayi)** ifadesini gördüğünde metodun içine gider, orada tanımlanan tüm işlemleri yaptıktan sonra **return** kelimesinin karşısındaki işlemleri çağırıldığı yere geri iletir. Metotorneğiil java programında çağırılan kare(sayi) metodunda sadece sayının kendisiyle çarpımı return kelimesinin karşısına yazıldığından, **kare(sayi)** ifadesi **sayı*sayı** ifadesiyle eşdeğerdir. Burada kare metodundaki değişken adının **sayı** değil **x** olarak verildiğini de burada not edelim. Sayı değişkeni metodun girişinde x değişkenine yüklenir. Gerekli işlemler metotta yapıldıktan sonra sadece metodun sonuçları return deyimiyle metodun değişken türü olarak (bu metod için double) geri döner.

Bu işlemi iki ayrı java programında (sınıfında) da gerçekleştirebilirdik. Bir sonraki örneğimizde birinci programımızda kare metodu tanımlanacak, ikinci programımızda ise çağrılacaktır.

Program 2.2.2 : [metotornegilA.java](#) programında kare statik metodunun tanımlanması

```
import javax.swing.*;
public class metotornegi1A
{
public static double kare(double x)
{ return x*x; }
}
```

Program 2.2.3 : [metotornegi1B.java](#) programında kare statik metodunun kullanılması

```
import javax.swing.*;
public class metotornegi1B
{
public static void main(String[] args)
{
double sayi=Double.parseDouble(JOptionPane.showInputDialog("Bir gercek sayi giriniz : "));
JOptionPane.showMessageDialog(null,"girilen sayinin karesi : "+metotornegi1A.kare(sayi));
}
}
```

Görüldüğü gibi **metotornegi1A** sınıfında tanımlanan **public static double kare(double x)** metodunu, **metotornegi1B** sınıfında çağırırken **metotornegi1A.kare(sayi)** ifadesi kullanılmıştır. Burada önce sınıfın ismi sonra da nokta(.) işaretiyle ayrılmış olarak metodun ismi kullanılarak çağırılmıştır. Bu çağrı şekli sadece static olarak çağrılan metodlar için geçerlidir.

2.3 NESNE (OBJECT) TANIMI VE METOTLARDA KULLANIMI

Birinci bölümde temel değişken türlerini (double,int,boolean v.b.) nasıl tanımladığımızı görmüştük. Nesne tipi programlamanın en önemli özelliği kendi değişken türlerimizi yaratabilmemizdir. İlerideki bölümlerde kendi değişken türlerimizi sınıf (class) yapısını kullanarak nasıl yaratacağımızı daha detaylı olarak inceleyeceğiz. Bu bölümde nesnelerin program içindeki tanımlanmasına göz atalım. Daha önce tanımladığımız bazı programlarda nesne tanımları zaten geçmişti. Örneğin daha önceki programlarımızda bilgi okutmak için kullandığımız

```
Scanner input = new Scanner( System.in );
```

Terimi **Scanner** sınıfından(class) **input** nesnesini tanımlar. Nesne(object) tanımı yaparken aynı metotlarda olduğu gibi değişken veya nesnelere girdi olarak tanımlamamız mümkündür.

Diğer bir örnek olarak yine daha önceki örnek problemde kullandığımız

```
Locale us=new Locale("US");
```

Deyimini verebiliriz. Bu deyim Locale sınıfından us nesnesini tanımlamaktadır. Bu sınıf formatlama işleminde kullanılacak standartları belirler (örneğin gerçek sayıyı . veya , ile ayırma gibi)

Bu örneklerde de görüldüğü gibi Nesne tanımı şu şekilde yapılmaktadır:

```
Sınıf (class)_ismi nesne(object)_ismi;
```

```
nesne(object)_ismi = new Sınıf (class)_ismi(değişken veya nesne girdi listesi)
```

veya

Sınıf(class)_ismi nesne(object_ismi= new Sınıf (class) ismi(değişken veya nesne girdi listesi)

Nesne tanımını daha iyi anlamak için metotorneği1A sınıfını yazdığımız programı biraz değiştirerek metotorneği2 ve metotorneği2A sınıflarını oluşturalım.

Program 2.3.1 : [metotorneği2.java](#) programında kare statik metotunun kullanılması

```
import javax.swing.*;
public class metotorneği2
{
    public double kare(double x)
    {
        return x*x;
    }
}
```

Görüldüğü gibi bu programda kare metodunun tanımında static sözcüğü kullanılmamıştır. Şimdi bu metodu ikinci bir programda çağıralım.

Program 2.4 : [metotorneği2A.java](#) programında kare metotunun kullanılması

```
import javax.swing.*;
public class metotorneği2A
{
    public static void main(String[] args)
    {
        double sayı=Double.parseDouble(JOptionPane.showInputDialog("Bir gerçek sayı giriniz : "));
        metotorneği2 nesne1=new metotorneği2();
        JOptionPane.showMessageDialog(null,"girilen sayının karesi : "+nesne1.kare(sayı));
    }
}
```

Metotorneği2.java programında bir sınıf(class) yaratılmıştır. Metotörneği2A sınıfında kare metodu kullanılacaktır. Bunun için önce metotörneği2 sınıfından nesne1 nesnesi yaratılmış ve kare metodu bu nesne1 nesnesi referans alınarak **nesne1.kare(sayı)** şeklinde çağırılmıştır. Kare metotunu çağırırken bu yerin adresini kullanır. Metotörneği1.java programında kare(sayı) doğrudan kullanılmıştı. Çünkü kare metodu aynı sınıfın(class) içinde tanımlanmıştı. Metotörneği1B.java ve metotörneği2A.java programında ise kare metodu ayrı bir sınıf(class) ta tanımlandığından yerinin tanımlanması gerekir. Bu bizim kendi odamızın yatak odasını tanımlarken sadece yatak odası dememizle başka bir evin yatak odasını tanımlarken Ahmet beyin evinin yatak odası dememiz gibidir. Metotlar statik olarak da tanımlanabilirler. Statik olarak tanımlanan metotlar nesne kullanılmadan direk olarak sınıf(class) adları kullanılarak çağırılabilirler. **metotörneği1A.kare(sayı)** örneğinde olduğu gibi. Dinamik olarak tanımlanan metotlarda ise önce bir nesne yaratmamız ve metodu bu nesne üzerinden tanımlamamız gerekir.

metotörneği2 nesne1=new metotörneği2();

double y=**nesne1.kare(sayı)**

2.4 METOTLARIN KENDİ KENDİNİ ÇAĞIRMASI (RECURSION)

Bazı problemlerin çözümünde bir metotun kendi kendini çağırması yararlı olabilir. Java bir metotun kendi kendini çağırmasına izin verir. Burada dikkatli olunması gereken nokta bu kendi kendine çağırılma döngüsünün sonsuza kadar sürmesinin bir kontrol yapısı kullanılarak engellenmesidir. Kendi kendini çağırın programlarda mutlaka bir döngü çıkış şartı tanımlanmalıdır.

Program 2.4.1 de daha önce de gördüğümüz faktoriyeli ana metotumuz olan main içinde hesaplayan faktoriyel programını görmekteyiz. Program 2.4.2 de faktoriyel hesabı for döngüsü kullanılarak faktoriyel metotunda gerçekleştirilmektedir.

Program 2.4.1 : Metot kullanılmadan (main metotunun içinde) faktoriyel hesaplayan faktoriyel.java programı

```
import javax.swing.JOptionPane;

class faktoriyel
{
    public static void main(String args[])
    {
        int faktoriyel=1;
        int faktoriyelsayi;
        faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayı giriniz : "));
        for(int sayi=1;sayi<=faktoriyelsayi;sayi++)
        { faktoriyel*=sayi;}
        String s=faktoriyelsayi+" faktoriyel: "+faktoriyel;
        JOptionPane.showMessageDialog(null,s, "Metod örneği faktoriyelSW.java ",
        JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Program 2.4.2 : Metot kullanarak (faktoriyel metotunun içinde) faktoriyel hesaplayan faktoriyel1.java programı

```
import javax.swing.JOptionPane;

class faktoriyel1
{
    public static long faktoriyel(int x)
    { long faktoriyel=1;
        for(int sayi=1;sayi<=x;sayi++)
        { faktoriyel*=sayi;}
        return faktoriyel;
    }
    public static void main(String args[])
    { int faktoriyelsayi;
        faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayı giriniz : "));
        String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);
        JOptionPane.showMessageDialog(null,s,"Metod örneği faktoriyel1.java ",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Program 2.13 : kendi kendini çağır (recursive) faktoriyel metotunu kullanarak faktoriyel hesaplayan Faktoriyel2.java programı

```
import javax.swing.JOptionPane;

class faktoriyel2
{
```



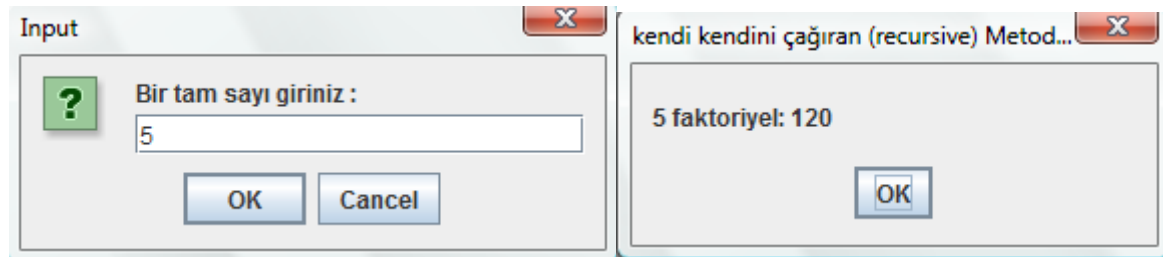
```

public static long faktoriyel(int x)
{
    if( x <= 1 )
        return 1;
    else
        return x * faktoriyel( x - 1);
}

public static void main(String args[])
{
    int faktoriyelsayi;
    faktoriyelsayi=Integer.parseInt(JOptionPane.showInputDialog("Bir tam sayı giriniz : "));
    String s=faktoriyelsayi+" faktoriyel: "+faktoriyel(faktoriyelsayi);
    JOptionPane.showMessageDialog(null,s, "kendi kendini çağıran (recursive) Metod örneği faktoriyel2SW.java
",
    JOptionPane.PLAIN_MESSAGE);
}
}

```

Verilen 3 programda da çıktılar aynıdır.



$$5! = 5 * 4! = 4 * 3! = 3 * 2! = 2 * 1! = 1$$

faktoriyel metodu 1 e ulaştığında tekrar kendisini çağırmıyacağından geriye doğru yaptığı hesapları göndermeye başlayacaktır.

$$1 = (2*1!(=1)) = 2 = (3 * 2!(=2)) = 6 = (4*3!(=6))=24 = (5*4!(24))=120 = 5!(=120)$$

sonuç 120 olarak ana programa gönderilecektir. Kendi kendini çağıran metodlar mutlak bir gereksinim olmadıkça tercih edilmez. Bu tür programlamada hem sonsuz döngülerin görülmesi daha güç olabilir, hem de for, while gibi döngüler kullanılarak programlamaya göre programlar daha yavaş çalışır. Gerçek uygulamalarda kendi kendini çağıran programlar genellikle bir döngü yapısı kurmadan çok tek kerelik çağrımları gerçekleştirmek için kullanılırlar.

2.5 AYNI ADLI METOTLARIN BİR ARADA KULLANILMASI (OVERLOADİNG)

Java aynı adlı metotların aynı sınıf içerisinde kullanılmasına izin verir. Aynı sınıfta(class) kullanılan metotların girdi değişkenlerinin değişken türlerinin veya değişken sayılarının farklı olması gerekir. Java aynı isimli ve aynı değişken türlü iki metodu birbirinden ayıramaz. Örneğin

Public double Bmetodu(double Adeğişkeni)

Ve

Public double Bmetodu(double Bdeğişkeni)

Veya

Public int Bmetodu(double Cdeğişkeni)

Java tarafından ayırt edilemez , Fakat

Public double Cmetodu(double Adeğişkeni)

Ve

Public double Cmetotu(int Bdeğişkeni)

veya

Public double Cmetotu(double Adeğişkeni, double Bdeğişkeni)

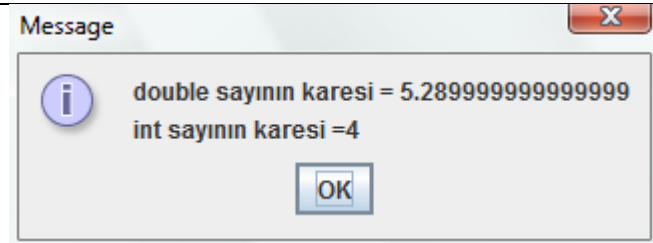
Java tarafından ayırt edilir ve birlikte aynı sınıfın (class) içinde yer alabilir. Eğer birbirinin tam olarak eşiti metotlar değişik sınıflarda yer alıyorsa bunun herhangi bir sakıncası yoktur. Program 2.5.1'de metotorneği3.java programı görülmektedir. Bu programda daha önceki programlarda da kullandığımız kare metotunu int ve double girdi değişkenleriyle iki kere aynı sınıfın içerisinde tanımlayacağız.

Program 2.5.1 : [metotorneği3.java](#) programı

```
import javax.swing.*;
public class metotorneği3
{
public double kare(double x)
{ return x*x; }
public int kare(int x)
{ return x*x; }
}
```

Program 2.5.2 : [metotorneği3A.java](#) programı

```
import javax.swing.*;
public class metotorneği3A
{
public static void main(String[] args)
{
double sayi=2.3;
int sayi1=2;
metotorneği3 y=new metotorneği3();
String s="double sayının karesi = "+y.kare(sayi)+"\n";
s+="int sayının karesi = "+y.kare(sayi1)+"\n";
JOptionPane.showMessageDialog(null,s);
}
}
```



İkinci bir örnek daha verelim. Bu örneğimizde exponent fonksiyonu (e sayısının üs fonksiyonu) double ve integer olarak aynı programda tanımlanmaktadır.

Program 2.5.3 : [metotorneği4.java](#) programı

```
import javax.swing.*;
public class metotorneği4
{
public double exp(double x)
```

```

{
    double faktoriyel=1;
    double us=1;
    double exponent=1;
    for(double i=1;i<=300;i++)
    {
        faktoriyel*=i;
        us*=x;
        exponent+=us/faktoriyel;
    }
    return exponent;
}

// int girişli exp metodu
public double exp(int x)
{
    double exponent=1.0;
    for(int i=1;i<=x;i++)
    {
        exponent*=Math.E;
    }
    return exponent;
}
}

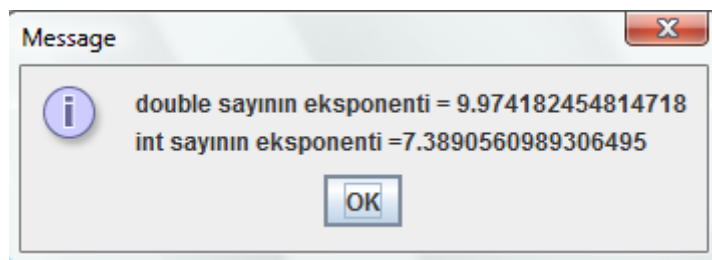
```

Program 2.5.4 : [metotornegi4A.java](#) programı

```

import javax.swing.*;
public class metotornegi4A
{
    public static void main(String[] args)
    {
        double sayi=2.3;
        int sayi1=2;
        metotornegi4 y=new metotornegi4();
        String s="double sayının eksponenti = "+y.exp(sayi)+"\n";
        s+="int sayının eksponenti ="+y.exp(sayi1)+"\n";
        JOptionPane.showMessageDialog(null,s);
    }
}

```



Aynı adlı metodların birlikte kullanılmasına son bir örnek, alttaki programımızda `JOptionPane.showMessageDialog` kütüphane çıktı metodu iki ayrı print metoduna dönüştürülmüştür.

```

import java.io.*; //java girdi çıktı sınıfını çağır
import javax.swing.*;
class BPIE2_1
{
    public static String charstring(char x1,char x2)
    {String s="";
        int i=0;
        if(x2>x1)

```

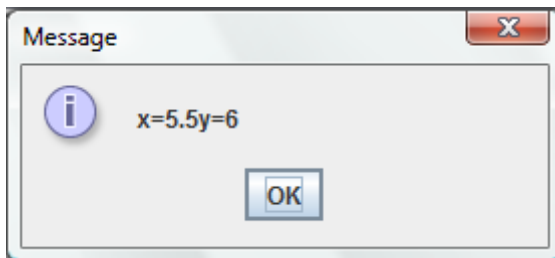


```
int y;
public void metot1()
{ x=5.5;
  y=6;
}

public void metot2()
{ JOptionPane.showMessageDialog(null,"x="+x+"y="+y);}
}
```

```
import javax.swing.*;

public class metotvesinifdegiskeni1
{
public static void main(String s[])
{ metotvesinifdegiskeni z=new metotvesinifdegiskeni();
  z.metot1();
  z.metot2();
}
}
```



Metodun içinde kullanılan yerel değişkenler diğer smetodlara aktarılmazlar, aktarılmaya çalışıldığında hata verirler.

```
import javax.swing.*;

public class metotvesinifdegiskeniA
{
//Buradaki değişkenler tüm sınıfa aittir

public void metot1()
{ double x;
  int y;
  x=5.5;
  y=6;
}

public void metot2()
{ JOptionPane.showMessageDialog(null,"x="+x+"y="+y);}
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javac.exe" metotvesinifdegiskeniA.java
```

```
metotvesinifdegiskeniA.java:15: error: cannot find symbol
{OptionPane.showMessageDialog(null,"x="+x+"y="+y);}
      ^
```

symbol: variable x

location: class metotvesinifdegiskeniA

```
metotvesinifdegiskeniA.java:15: error: cannot find symbol
{OptionPane.showMessageDialog(null,"x="+x+"y="+y);}
      ^
```

symbol: variable y

location: class metotvesinifdegiskeniA

2 errors

> Terminated with exit code 1.

Bir metodun içinde sınıf değişkeniyle aynı adda bir değişken kullanılıyorsa metod değişkeni değer alır, sınıf değişkeni değişmez.

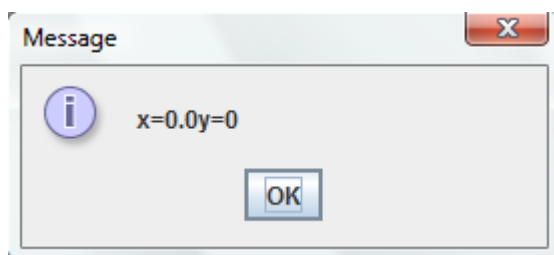
```
import javax.swing.*;

public class metotvesinifdegiskeni2
{
//Buradaki değişkenler tüm sınıfa aittir
double x;
int y;
public void metot1()
{double x=5.5;
double y=6;
}

public void metot2()
{OptionPane.showMessageDialog(null,"x="+x+"y="+y);}
}
```

```
import javax.swing.*;

public class metotvesinifdegiskeni2A
{
public static void main(String s[])
{metotvesinifdegiskeni2 z=new metotvesinifdegiskeni2();
z.metot1();
z.metot2();
}
}
```



2.7 ALIŞTIRMALAR

1. Programda verilen metodların sonuçlarını Math kütüphanesindeki eşdeğerleriyle karşılaştırarak hata analizi yapınız. (bunun için gerekli çıktı programını yazınız)

Program 2.7.1 metod.java çeşitli matematiksel dizi örnekleri

```
import javax.swing.*;
public class metod
{

    public static double exp(double x)
    {
        double factorial=1;
        double power=1;
        double exp=0;
        for(int k=1;k<200;k++)
        {exp+=power/factorial;
        power*=x;
        factorial*=k;
        }
        return exp;
    }

    public static double ln(double x)
    {
        double X=(x-1.0)/(x+1.0);
        double ln=X;
        double power=X;
        double K=1;
        for(int k=2;k<100;k++)
        { power*=X*X;
        K=2.0*k-1.0;
        ln+=power/K;
        }
        ln*=2.0;
        return ln;
    }

    public static double sin(double x)
    {int isaret=1;
    double pow=x;
    double fact=1;
    int n=1;
    double sin=0;
    for(int k=1;k<100;k++)
    {
        sin+=pow/fact*isaret;
        pow*=x*x;
        n=2*k+1;
        fact*=n*(n-1);
        isaret*=-1;
    }
    return sin;
    }

    public static double sinh(double x)
    {
        double pow=x;
```

```

double fact=1;
int n=1;
double sinh=0;
for(int k=1;k<100;k++)
{
sinh+=pow/fact;
pow*=x*x;
n=2*k+1;
fact*=n*(n-1);
}
return sinh;
}

public static double cos(double x)
{int isaret=1;
double pow=1;
double fact=1;
int n=1;
double cos=1;
for(int k=1;k<100;k++)
{
pow*=x*x;
n=2*k;
fact*=n*(n-1);
isaret*=-1;
cos+=pow/fact*isaret;
}
return cos;
}
public static double cosh(double x)
{
double pow=1;
double fact=1;
int n=1;
double cosh=1;
for(int k=1;k<100;k++)
{
pow*=x*x;
n=2*k;
fact*=n*(n-1);
cosh+=pow/fact;
}
return cosh;
}

public static double sqrt(double x)
{return exp(0.5*ln(x));}

public static double pow(double a,double x)
{return exp(x*ln(a));}

public static double quadpolyroot1(double a,double b,double c)
{ //y=f(x)=a*x*x+b*x+c=0
double delta=b*b-4*a*c;
double x1=0;
if(delta<0) System.out.println("Roots are complex");
else { x1=(-b+sqrt(delta))/(2*a);}
return x1;
}

```



```

}

public static double quadpolyroot2(double a,double b,double c)
{ //y=f(x)=a*x*x+b*x+c=0
double delta=b*b-4*a*c;
double x2=0;
if(delta<0) System.out.println("Roots are complex");
else { x2=(-b-sqrt(delta))/(2*a);}
return x2;
}

public static void main(String arg[])
{
//f(x)=exp(x)
double x=Double.parseDouble(JOptionPane.showInputDialog("x="));

System.out.println("exp("+x+")="+exp(x));
System.out.println("ln("+x+")="+ln(x));
System.out.println("sqrt("+x+")="+sqrt(x));
System.out.println("cosh("+0+")="+cosh(Math.PI/2.0));
System.out.println("cosh("+0+")="+Math.cosh(Math.PI/2.0));
}
}

```

2. Math kütüphanesindeki Math.random() metotunu kullanarak **static zar** metotunu yazınız. Bu metodu [H4A1a](#) sınıfında(class) tanımlayınız.

Not : 1-den 6 ya kadar tesadüfi (random) sayıları $1+(\text{int})(\text{Math.random()}*6)$ formülü ile hesaplayabiliriz.

Program 2.7.2 H4A1a.java ve static int zar metodu

```

public class H4A1a
{
public static int zar()
{
return 1+(int)(Math.random()*6);
}
}

```

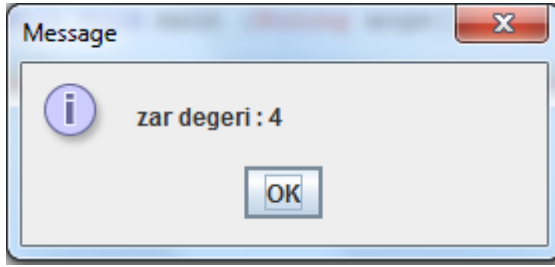
[H4A1b](#) sınıfında(class) java konsol ana metodu (public void main) yazarak zar degerini yazdiriniz.

Program 2.7.3 H4A1b.java

```

import javax.swing.*;
class H4A1b
{
public static void main (String args[])
{
JOptionPane.showMessageDialog(null,"zar degeri : "+H4A1a.zar());
}
}

```

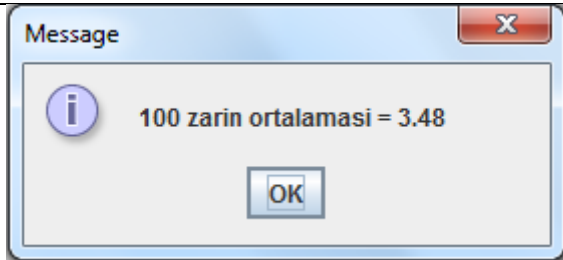


[H4A1c](#) sinifinda java konsol ana metotunda (public void main) metotunda yüz(100) kere zar atılarak ortalamasını hesaplayınız.

Not : for veya while döngüsü kullanabilirsiniz.

Program 2.7.4 H4A1c.java

```
import javax.swing.*;
class H4A1c
{
//yuz zarin ortalamasi
public static void main (String args[])
{
double ortalama;
double toplam=0;
for(int sayi=0;sayi<100;sayi++)
{
toplam+=H4A1a.zar();
}
ortalama=toplam/100.0;
JOptionPane.showMessageDialog(null,"100 zarin ortalamasi = "+ortalama);
}
}
```



[H4A1d](#) sinifinda java konsol ana metotunda (public void main) bin(1000) kere zar atılarak her rakamın kac kere geldigini bulunuz ve yazdırınız.

Not : switch – case deyimi bu tür programlar için idealdir

Program 2.7.5 H4A1d.java

```
public class H4A1d
{
//bin zar atiminda her zar kac kere gelir
public static void main (String args[])
{
int zardegeri;
int zar1,zar2,zar3,zar4,zar5,zar6;
int zaratimsayisi=0;
zar1=0;
```

```
zar2=0;
zar3=0;
zar4=0;
zar5=0;
zar6=0;
while(zaratimsayisi<1000)
{
zardegeri=H4A1a.zar();
switch(zardegeri)
{
case 1:
zar1++;
break;
case 2:
zar2++;
break;
case 3:
zar3++;
break;
case 4:
zar4++;
break;
case 5:
zar5++;
break;
case 6:
zar6++;
break;
}
zaratimsayisi++;
}
System.out.println("toplum zar 1 sayisi : "+zar1);
System.out.println("toplum zar 2 sayisi : "+zar2);
System.out.println("toplum zar 3 sayisi : "+zar3);
System.out.println("toplum zar 4 sayisi : "+zar4);
System.out.println("toplum zar 5 sayisi : "+zar5);
System.out.println("toplum zar 6 sayisi : "+zar6);
}
}
```

```
toplum zar 1 sayisi : 169
toplum zar 2 sayisi : 165
toplum zar 3 sayisi : 165
toplum zar 4 sayisi : 146
toplum zar 5 sayisi : 178
toplum zar 6 sayisi : 177
```

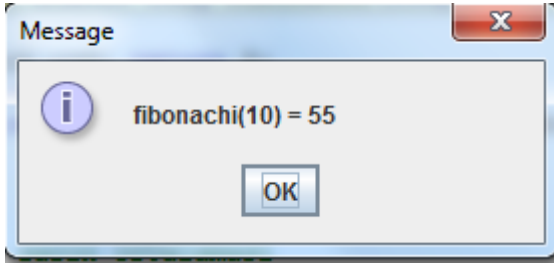
3. **fibonachiltesti.java** programını inceleyiniz. Programın kendi kendini çağırarak (recursive) versiyonu yerine döngü kullanan versiyonunu geliştiriniz. Fibonacci serisi $F_n = F_{n-1} + F_{n-2}$ başlangıç değerleri $F_0 = 0$ ve $F_1 = 1$. Formülüyle tanımlanır. Seri Lakabı Fibonacci (*filius Bonaccio* , Bonaccionun oğlu teriminden gelmektedir) Pisalı Leonardo tarafından ilk defa oluşturulmuştur. Fibonacci 102 yılında yazdığı Liber

Abaci isimli kitabında bu seriyi vermiştir. Seri daha önce de Hintliler tarafından bilinmekte ve kullanılmaktaydı.

Program 2.7.6 fibonachi testi

```
import javax.swing.*;
class fibonachitesti
{
    public static long fibonachi(long sayi)
    {
        if(sayi <=0) return 0;
        else if(sayi ==1) return 1;
        else return fibonachi(sayi - 1)+fibonachi(sayi - 2);
    }

    //yuz zarin ortalamasi
    public static void main (String args[])
    {
        int n=Integer.parseInt(JOptionPane.showInputDialog("n = "));
        JOptionPane.showMessageDialog(null,"fibonachi("+n+") = "+fibonachi(n));
    }
}
```



4. Doğal logaritma seri açılımı olarak :



$$y = \frac{x-1}{x+1}$$

Formülü ile verilmiştir:

Program 2.7.7 BPIE2_4.java : doğal logaritmayı statik metod olarak hesaplayan bir program.

```
import javax.swing.*; // program poweres class Scanner

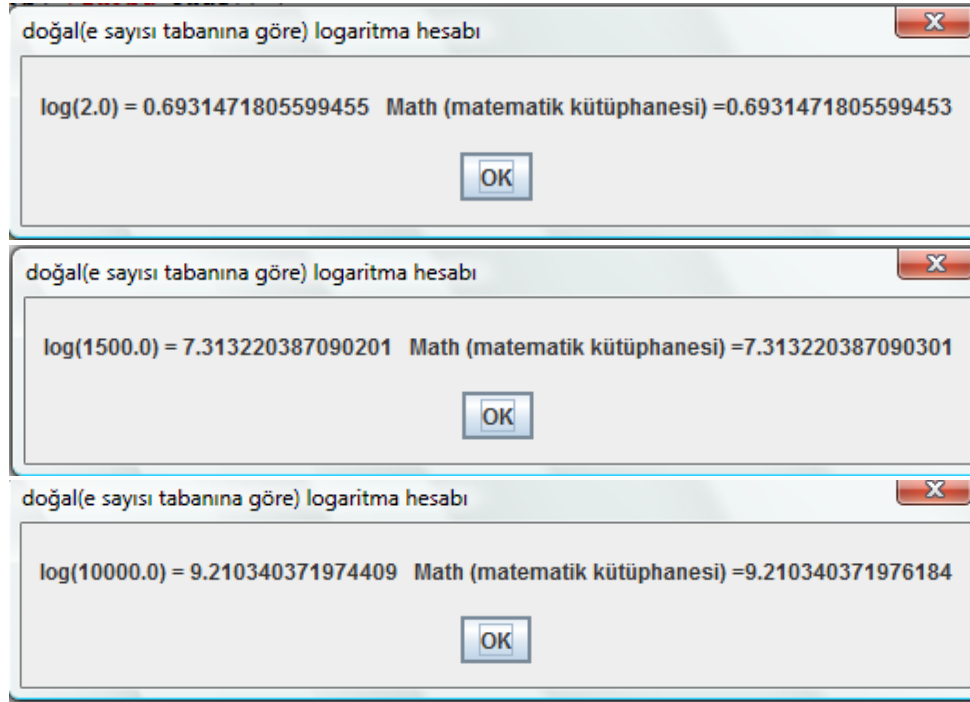
public class BPIE2_4
{
    public static double log(double x)
    {
        // ln(x) = 1+y^2/3+y^4/5+y^6/7+... y=(x-1)/(x+1)
        double power=1;
        double ln=1;
```

```

double y=(x-1)/(x+1);
double n=1;
do
{ power*=y*y;
ln+=power/(2.0*n+1.0);
n++;
} while(n<=100000);
ln*=2*y;
return ln;
}
// main metodunun başlangıcı
public static void main( String args[] )
{
String s;
double x=Double.parseDouble(JOptionPane.showInputDialog("gerçek bir sayı giriniz : ")); // read number
s="log("+x+") = "+log(x)+" "+Math (matematik kütüphanesi)="+Math.log(x);
System.out.println(s);
JOptionPane.showMessageDialog(null,s,"doğal(e sayısı tabanına göre) logaritma hesabı",JOptionPane.PLAIN_MESSAGE);
} // main metodunun sonu
} // BPIE2_4 sınıfının sonu

```

Sonuçlar hesapladığımız serinin doğruluğunu görebilmek amacıyla Math kütüphanesiyle de karşılaştırıldı.



Hesaplamadaki hataları azaltmak için büyük sayıları küçük sayılara dönüştürerek hesaplayabiliriz.

Örneğin:

$$\ln(123.456)=\ln(1.23456*10^2)=\ln(1.23456)+\ln(10^2)=\ln(1.23456)+2*\ln(10)=$$

$$\ln(1.23456)+2*2.302585092994046$$

$$\ln(1234.56)=\ln(1.23456)+3*2.302585092994046$$

$$\ln(12345.6)=\ln(1.23456)+4*2.302585092994046$$

Bu özelliği kullanan ikinci bir program yazalım:

Program 2.7.8 BPIE2_4A.java : doğal logaritmayı statik metod olarak rakam düzelmesiyle birlikte hesaplayan bir program.

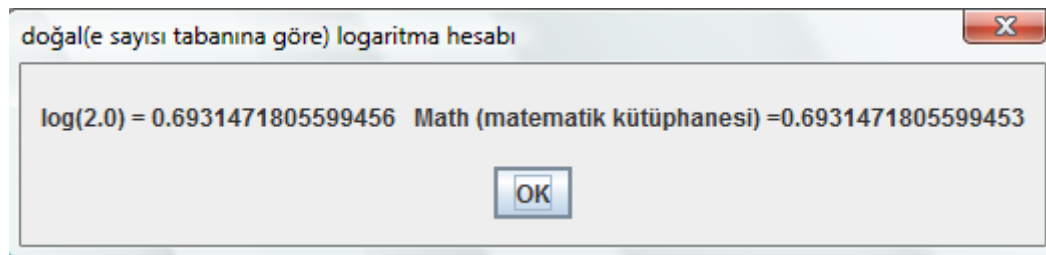
```

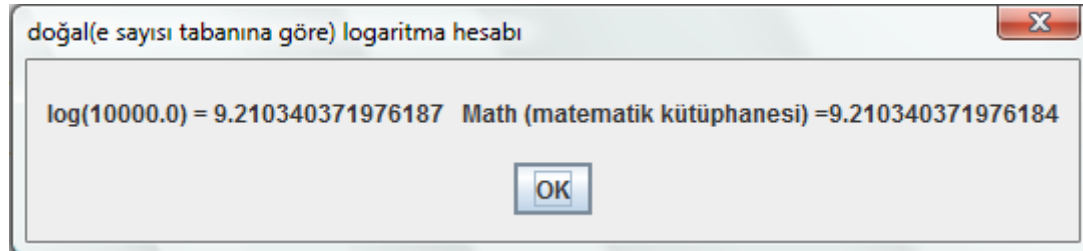
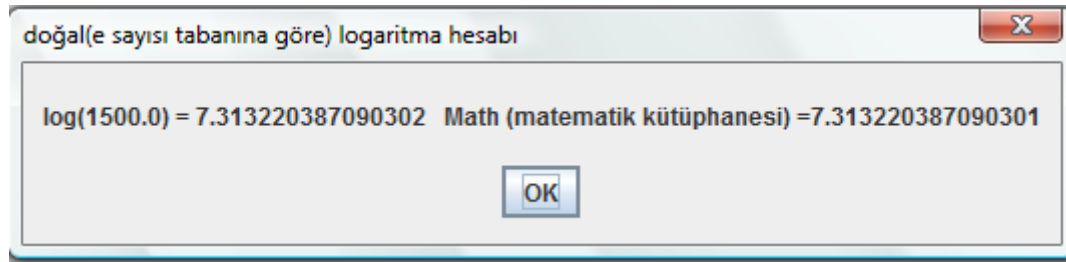
import javax.swing.*; // program poweres class Scanner

public class BPIH5E14
{
public static double log(double x)
{
// ln(x) = 1+y^2/3+y^4/5+y^6/7+...  y=(x-1)/(x+1)
//
double ln10=2.302585092994046;
int m=0;
int j=1;
double z=x;
j=(int)z;
while(j>0)
{
if(j>0) {x=z;m++;}
z=x/10;
j=(int)z;
}
x=z;

double power=1;
double ln=1;
double y=(x-1)/(x+1);
double n=1;
do
{
power*=y*y;
ln+=power/(2.0*n+1.0);
n++;
} while(n<=100000);
ln*=2*y;
ln+=m*ln10;
return ln;
}
// main method begins execution of Java application
public static void main( String args[] )
{
String s;
double x=Double.parseDouble(JOptionPane.showInputDialog("Enter a real number: ")); // read number
s="log("+x+") = "+log(x)+"  "+"Math library =" +Math.log(x);
JOptionPane.showMessageDialog(null,s,"natural logarithm calculation",JOptionPane.PLAIN_MESSAGE);
} // end method main
} // end class BPIH4E1

```





5. Burada vereceğimiz örnek program String tipi değişken kullanarak bir fonksiyon çizmektedir.

```
import javax.swing.*;

class BPIE2_5
{
    public static String boskarakter(int bosluk,char x1)
    {
        String s="";
        for(int i=0;i<bosluk;i++)
        {
            s+=" ";
        }
        s+=x1+"\n";
        return s;
    }

    public static int y(int x)
    {
        return x*x;
    }

    public static void plot(int x1,int x2,char c1)
    {
        String s="";
        for(int x=x1;x<=x2;x++)
        {
            s+=boskarakter(y(x),c1);
        }
        print(s);
    }

    public static void print(String s)
    {
        JOptionPane.showMessageDialog(null,s," ",JOptionPane.PLAIN_MESSAGE);
    }

    public static void main(String args[])
    {
        plot(-10,10,'*');
    }
}
```



6. $A=6\sum_{n=1}^{10000}\frac{1}{n^2}$ serisini ve bu seriyi kullanarak ~~TAM SIKI~~ sayısını hesaplayan bir statik metod yazınız.

7. **Tamsayı** x1 den x2 ye kadar olan sayıların ortalamasını alacak bir statik metod yazınız ve sonuçları çıktı olarak alınız.

8. **Gerçek sayı fonksiyonu** $y=f(x)=\frac{1}{x^2}$ i xin 1 den 20 ye kadar değerleri için hesaplayınız. Bu hesabı yaparken y yi statik bir fonksiyon olarak tanımlayınız.

2.8 PROBLEMLER

1. Bir kürenin hacmini hesaplayacak bir program yazınız. Kürenin hacmi küreHacmi isimli metot tarafından hesaplınsın. Program kürenin yarıçapını sorsun ve yarıçap girildiğinde yeni küre hacmi ekranda gösterilsin.

2. Bir dik üçgenin hipotenüsünü hesaplayan bir program yazınız. Program dik üçgenin iki dik girsin ve hipotenüs metoduyla hesaplınsın ve sonucu yazdırsın.

Not : Dik üçgenin hipotenüsü :

Hipotenüs = Math.sqrt(dikkenar1*dikkenar1+dikkenar2*dikkenar2);

3. iki kenar ve aralarındaki açı (derece cinsinden) verildiğinde üçgenin üçüncü kenarını hesaplayacak

public static double ucgenkarşikenar(double birinciyan, double ikinciyan, double aciderece)

metodunu yazınız.

not:karsikenar=Math.sqrt(birinciyan*birinciyan+ikinciyan*ikinciyan-
2*birinciyan*ikinciyan*Mathd.cosd(aradakiaci derece));

bu metodu H3OD5_2000 sınıfı **JoptionPane** konsol programı main metodunda kullanınız (iki yan kenar ve aradaki açı derece cinsinden verilecek, üçüncü kenar hesaplanarak yazılacak)

4. Maths sınıfında **static double exp(double x)** metodunu yazılmıştı. Bu metodu başka bir sınıf içinde çağırarak ekrandan girilen bir x değeri için sonucu hesaplatır.

5 . Math kütüphanesinde aşağıdaki metotlar tanımlanmıştır.

public static native double sin(double a)

a radyan cinsinden

public static native double cos(double a)

a radyan cinsinden

public static native double tan(double a)

a radyan cinsinden

public static native double asin(double a)

sonuç değeri : -pi/2 den pi/2 ye kadar.

public static native double acos(double a)

sonuç değeri 0 dan pi' ye kadar

public static native double atan(double a)

sonuç değeri : -pi/2 den pi/2 ye kadar.

public static native double atan2(double a,
double b)

sonuç değeri : -pi den pi ye kadar

bu metotların hepsi radyan cinsinden tanımlıdır. Bu metotların derece eşitlerini tanımlamak istiyoruz. [Mathd](#)

sınıfı yaratarak, bu sınıfta

public static double sind(double a)

a derece cinsinden

public static double cosd(double a)

a derece cinsinden

public static double tand(double a)

a derece cinsinden

public static double asind(double a)

sonuç değeri : -90 dan 90 dereceye kadar.

public static double acosd(double a)

sonuç değeri 0 dan 180 dereceye kadar

public static double atand(double a)

sonuç değeri : -90 dan 90 a kadar.

public static double atan2d(double a,
double b)

sonuç değeri : -180 den 180 dereceye kadar

statik metotlarını oluşturunuz. Bu programa derece cinsinden bir girdi giriniz ve hesaplanmış sin,cos,tan değerlerini yazdırınız.

not :

derece=radyan/Math.PI*180;

Bir konsol programı oluşturunuz. Bu programa 0-1 arasında bir girdi giriniz. Mathd sınıfından atand,acosd ve asind statik metotlarını kullanarak bu metotların sonuçlarını konsol ortamında çıktıya yazdırınız.

6. kendi kendini çağıran (recursive), tabanın kuvvetinci kuvvetini hesaplayacak (taban^kuvvet, 2^3=8 gibi)

public static pow(double taban, double kuvvet)

metotunu, konsol programı olarak yazınız. main metotunda dışarıdan bir sayı okuyarak bu metot yardımıyla sonucu hesaplayınız.

7. döngü yapısıyla hesap yapan, tabanın kuvvetinci kuvvetini hesaplayacak (taban^kuvvet, 2^3=8 gibi) public pow(double taban, double kuvvet) H3OD4a sınıfında yazınız. H3OD4 sınıfında bir program oluşturarak H3OD4a sınıfındaki pow metodu yardımıyla verilen bir sayının verilen bir kuvvetini hesaplayınız. ve çıktı olarak yazdırınız.

```
class H3OD4a
{
public static double kuvvet(double a,double b)
{
double k=1.0;
if(b<0)
{
for(int i=0;i<(int)(-b);i++)
{k*=a;}
k=1.0/k;
}
else if(b==0)
{
k=1.0;
}
else if(b>0)
{
for(int i=1;i<=(int)b;i++)
{k*=a;}
}
return k;
}
}
```

8. iki kenar ve aralarındaki açı (derece cinsinden) verildiğinde üçgenin üçüncü kenarını hesaplayacak **public static double ucgenkarsikenar(double birinciyan, double ikinciyan, double aciderece)** metotunu yazınız.
not : **karsikenar=Math.sqrt(birinciyan*birinciyan+ikinciyan*ikinciyan - 2*birinciyan*ikinciyan*Mathd.cosd(aradakiaci derece));**
bu metodu H3OD5 sınıfı konsol programı main metotunda kullanınız (iki yan kenar ve aradaki açı derece cinsinden verilecek, üçüncü kenar hesaplanarak yazılacak)

9. Hata fonksiyonu erf(x) aşağıdaki formülle tanımlanabilir.

$$erf(x) = \frac{1}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{n!(2n+1)}$$

Bu formülü hesaplayacak olan statik erf metodunu yazınız ve bir örnekte deneyiniz.

10. Bir konsol programı yazınız. Text sınıfı kullanarak dışarıdan x değerini, **celcius**, fahrenheit cinsinden girilen değerin celcius eşitini hesaplasın. Bunun için

$$C=5/9*(F-32)$$

formülünü kullanabilirsiniz.

İkinci metot, **fahrenheit**, derece celcius cinsinden girilen değerin fahrenheit eşdeğerini hesaplasın.

11. Bir program yazınız. Bu programın içindeki kubunhacmi metodu kübün hacmini hesaplasın. Ana metotta (main) kullanıcıya kübün kenar uzunluğunu sorsun ve hacmini versin.

12. **mesafe** metodunu yazınız. Bu metod iki noktanın (x1,y1) ve (x2,y2) arasındaki mesafeyi hesaplasın. Bu metodu bir konsol test programında deneyin.

$$mesafe(x1, y1, x2, y2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

13. Bir tam sayı eğer kendisinden başka bir tam sayıyla bölünemiyorsa, asal sayı adını alır. Örneğin 2,3,5,7,11.. Asal sayıları bulan bir metod yazınız. Bu metodu bir applette kullanarak 1 den 1000 e kadar olan sayılar arasındaki asal sayıları belirleyiniz.

14. Üç (gerçek veya tamsayı) sayının arasından minimum olanı bulan **minimum3** metodunu yazınız ve bir konsol programında test ediniz

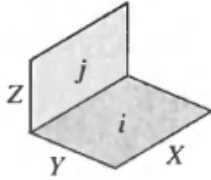
15. Üç sayının (gerçek veya tamsayı) arasından maksimum olanı bulan **maksimum3** metodunu yazınız ve bir konsol programında test ediniz

17. **tekmi** isimli boolean bir metod yazınız. bu metod eğer girilen sayı tekse true değeri, sayı çiftse false değeri versin. Bu metodu bir programda test ediniz.

18. kürenin hacmini hesaplayan hacim(double x) metodunu içeren küre programını oluşturunuz. Bu program kürenin yarıçapını sorsun, ve sonuç olarak hacim metodunda hesaplanan kürenin hacmini versin.

19. Komşu kenarlı birbirine dik iki yüzeyin ışınım şekil faktörünü ana metod içinde hesaplayan bir versiyonu aşağıda verilmiştir. Burada verilen hesaplamayı bir metoda dönüştürünüz.

KOMŞU KENARLI BİRBİRİNE DİK İKİ YÜZEYİN İŞINIM ŞEKİL FAKTÖRÜ $H = Z/X, W = Y/X$



$$F_{ij} = \frac{1}{\pi W} \left(W \tan^{-1} \frac{1}{W} + H \tan^{-1} \frac{1}{H} - (H^2 + W^2)^{1/2} \tan^{-1} \frac{1}{(H^2 + W^2)^{1/2}} + \frac{1}{4} \ln \left\{ \frac{(1 + W^2)(1 + H^2)}{1 + W^2 + H^2} \left[\frac{W^2(1 + W^2 + H^2)}{(1 + W^2)(W^2 + H^2)} \right]^{W^2} \times \left[\frac{H^2(1 + H^2 + W^2)}{(1 + H^2)(H^2 + W^2)} \right]^{H^2} \right\} \right)$$

```
import java.io.*; //java girdi cikti sinifini cagir
import javax.swing.*;
```

```
class isinim
{
    public static void main(String args[])
    {
        double X=0.5;
        double Y=1.0;
        double Z=1.0;
        double PI=Math.PI;
        double H=Z/X;
        double W=Y/X;
```

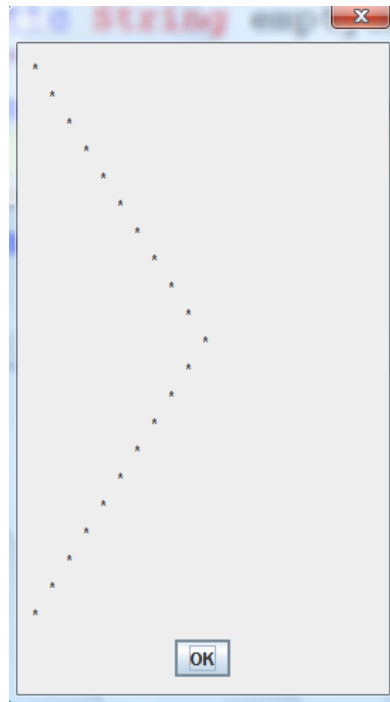
```

double Fij=1/(PI*W)*(W*Math.atan(1/W)+H*Math.atan(1/H)-
Math.sqrt(H*H+W*W)*Math.atan(1/Math.sqrt(H*H+W*W))+
0.25*Math.log((1+W*W)*(1+H*H)/(1+W*W+H*H))*Math.pow((W*W*(1+W*W+H*H)/(1+W*W)/(W*W+H
*H)),(W*W))*Math.pow((H*H*(1+W*W+H*H)/(1+H*H)/(W*W+H*H)),(H*H)));
String s="Fij="+Fij;
JOptionPane.showMessageDialog(null,s,"ışınım şekil faktörü",JOptionPane.PLAIN_MESSAGE);
}
}

```

20. İki zar atıldığında gelen zarlar iki ile onikiarasında olacaktır. Her bir rakamın gelme olasılığını hesaplayınız ve olasılık dağılımını inceleyiniz. Bunun için **yukarıda verdiğimiz zar metodunu kullanan bir statik metod yazınız**

21. **Program 5.1.9** Alıştırma 5 de string değişkenini kullanarak satır grafiği çizme örneği görmüştük. Aynı yöntemi kullanarak alttaki şekli oluşturan bir metod yazınız ve çalıştırınız.



BÖLÜM 3 . SINIF (CLASS) YAPILARINA GİRİŞ

3.0 STANDART KAVRAMLARIN TÜRKÇE KARŞILIKLARI

Class : sınıf ana nesne kökenli programlama elemanı.

Object : nesne sınıf (Class) yapıları bilgisayar belleğinde bir işlem için kullanıldığında aldıkları (özel) isim.

new : sınıfların kurucu metotunu çağırıp nesne tanımını ve bilgisayar adreslerini oluşturan deyim

Method: metot sınıfların içindeki işlevsel program parçacıkları

Constructor : Kurucu metot nesne ilk defa oluşturulurken hangi sınıf değişkenlerinin hangi değerleri alacağını belirleyen metot

3.1 SINIF YAPISININ GENEL TANIMI

Şu na kadarki bütün programlarımızda sınıf(class) yapısını kullandık. Fakat tam olarak sınıfların ne olduğunu ve neler yapabileceklerini anlatmadık. Sınıf yapısının en önemli özelliği yeni bir değişken türü tanımlıyor olmasıdır. Şu ana kadar programlarımızı tanımlamak amacıyla çeşitli sınıflar tanımladık. Bu bölümde sınıfların değişken türü olarak tanımlanmasını ve kullanmasını göreceğiz. Sınıf (Class) deyince herhangi bir nesnenin genel tanımlanmasını anlıyoruz. Bu sınıflardan değişkenler tanımlandıklarında nesnelere oluştururlar. Örneğin kalem dediğimizde genel bir sınıfı anlatıyoruz. Alinin cebindeki kırmızı kalem ise belirli bir nesnedir. Sınıfın genel tanımı şu şekilde yapılabilir :

```
class sınıfın_ismi
{
//sınıfa ait değişken listesi
değişken_türü değişken_ismi;
değişken_türü değişken_ismi;
.....
//sınıfa ait metot listesi
değişken türü metotadı(metot girdi değişken listesi)
{
//metotun içindeki hesaplanan işlemler
return çıktı_değişkeni;
}
.....
}
```

Sınıf (class) tanımında metotların (veya değişkenlerin) bulunması şart değildir. Buldukları takdirde bunlar o sınıfa ait veya sınıfın üyesi metotlar olarak düşünülür. İlk sınıf örneği olarak içinde hiçbir metotun yer almadığı bir kutu sınıfı tanımlıyalım. Bir kutuyu üç geometrik boyutu (eni,boyu,yüksekliği) cinsinden tanımlayabiliriz.

```
class Kutu
{
double en;
double boy;
double yukseklik;
}
```

Üstte bahsettiğimiz gibi sınıf yeni bir değişken türü yaratır. Yukardaki sınıfın yarattığı değişken türünün adı Kutu'dur. Kutu değişkenini başka bir programda tanımlarken bu değişken türünden bir nesne yaratılır. Örneğin

```
Kutu kutu1= new Kutu();
```

kutu1, kutu türünde tanımlanmış bir nesnedir. Bu nesnenin kendisinin de en boy ve yükseklik olmak üzere üç alt değişkeni mevcuttur. Eğer bir programda kutu1 değişkeni tanımlanmışsa, bu değişkenin alt değişkenlerine şu şekilde ulaşılabilir :

kutu1.en=1.0;

Şimdi Kutu sınıfından nesneyi kullanacak ufak bir kututesti sınıfı oluşturalım ve Kutu sınıfını bu sınıfta kullanalım.

Program 3.1 : sınıf örneği, Kutu sınıfı tanımlayan [kututesti.java](#) programı

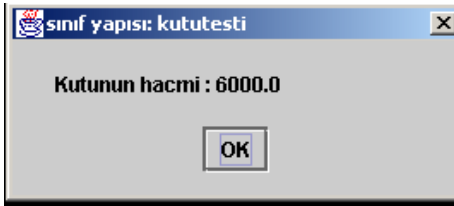
```
import javax.swing.JOptionPane; // giriş çıkış

class Kutu
{
double en;
double boy;
double yukseklik;
}

class kututesti
{
public static void main(String args[])
{
double hacim;
String s="";
Kutu kutu1=new Kutu();
kutu1.en=10.0;
kutu1.boy=20.0;
kutu1.yukseklik=30.0;
hacim=kutu1.en*kutu1.boy*kutu1.yukseklik;
s=s+"Kutunun hacmi : "+hacim+"\n";
JOptionPane.showMessageDialog(null,s,
"sınıf yapısı: kututesti1",JOptionPane.PLAIN_MESSAGE);
System.exit(0);}
}
```

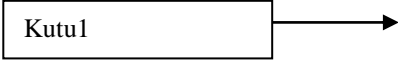
Bu programın adının [kututesti.java](#) olması zorunludur. Çünkü ana metod(main) kututesti sınıfında tanımlanmıştır. Fakat programı javac terimiyle bilgisayar diline çevirdiğimizde iki tane sınıf(class) dosyasının oluştuğunu görürüz. Birisi **Kutu.class** diğerde **kututesti.class**. Ana program kututesti olduğu için bu programın işletimi **java kututesti**

deyimini kullanarak yapılır. Program işletildiğinde



Sonucunu verir. kutu1 nesnesi yaratıldığında bilgisayar belleğinde kutu1 için bir adres tanımlanır. New Kutu() deyimi verildiğinde ise bilgisayarın belleğinde en,boy ve yükseklik değişkenleri için yer ayrılır ve bu yerlerin adresleriyle kutu1 nesnesinin adresi arasında bağlantı sağlanır. Bunu grafik olarak anlatalım :

Tablo 3.1 Nesne adreslerinin bilgisayar belleğindeki yerleri ve yerleşmesi

Deyim	Bilgisayar ana nesne belleği	Nesne alt hafıza belleği
Kutu kutu1;	null	Mevcut değil
kutu1=new Kutu();	 Kutu1 adresi (örneğin: 23436A3F)	en Boy Yükseklik

Not: null terimi bilgisayar belleğinin boş olduğunu gösterir. Yukardaki grafikte ilk olarak

Kutu kutu1;

tanımı yapıldığında bilgisayar belleğinde 1 byte boyutunda bir yer tanımlar. Ve bu yere boş anlamına gelen null değerini yükler. İkinci durumda ise (**kutu1=new Kutu()** tanımında) kutu1 nesne adresindeki kutuda en,boy,yükseklik değişkenlerinin adresleri kaydedilmiştir. Bu adres aracılığıyla en boy ve yükseklik değişkenlerine kutu1 değişkeni üzerinden ulaşılabilir. Diğer deyimle en,boy,yükseklik kutu1 nesnesinin alt değişkenleridir.

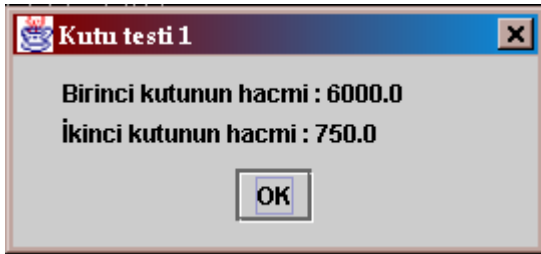
3.2 KURUCU (CONSTRUCTOR) METOT

İkinci bir programda Kutu sınıfı iki nesne tanımlama gereği duyduğumuzu varsayalım. Buna örnek olarak Program 3.2 de kututesti1.java programı tanımlanmıştır.

Program 3.2 : Kurucu metot örneği, Kutu sınıfı iki nesne tanımlayan [kututesti1.java](#) programı

```
class kututesti1
{
    public static void main(String args[])
    {
        double hacim;
        Kutu kutu1=new Kutu();
        String s="";
        kutu1.en=10.0;
        kutu1.boy=20.0;
        kutu1.yukseklk=30.0;
        hacim=kutu1.en*kutu1.boy*kutu1.yukseklk;
        s+="Birinci kutunun hacmi : "+hacim+"\n";
        Kutu kutu2=new Kutu();
        kutu2.en=5.0;
        kutu2.boy=10.0;
        kutu2.yukseklk=15.0;
        hacim=kutu2.en*kutu2.boy*kutu2.yukseklk;
        s+="İkinci kutunun hacmi : "+hacim+"\n";
        JOptionPane.showMessageDialog(null,s,
            "Kutu testi 1",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Bu programın çıktısı :



Bu programda Kutu sınıfından **kutu1** ve **kutu2** değişkenleri tanımlanmıştır. Bu basit programda bile en,boy ve yükseklik değişkenlerini iki defa çağırmak program listesinin uzamasına sebep olmuştur. Gerçek uygulamalarda ise çok daha yüksek sayıda değişken içeren sınıflar oluşturulabilir ve bu sınıflarda oluşturulan toplam nesne sayısı da çok fazla olabilir. Bu yüzden nesneyi ilk tanımlarken aynı zamanda değişkenlerinin değerlerini de tanımlayabilirsek, çok daha okunaklı ve kısa bir bilgisayar listesiyle aynı işi yapabiliriz. Bu iş için kurucu (constructor) adı verilen metotlar kullanılır. Kurucu metotları diğer metotlardan biraz farklıdır. Önce metot

önüne metot değişken türü gelmez ve metot dışına hiçbir değişken göndermezler. İsimleri de her zaman sınıf(class) ismiyle aynı olur. Program 3.1 deki programı kurucu (constructor) metodu kullanarak tekrar yazalım.

Program 3.3 : Kurucu metot örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan [kututesti2.java](#) programı

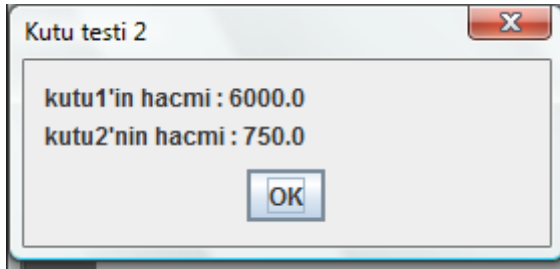
```
import javax.swing.JOptionPane; // giriş çıkış

class Kutu
{
double en;
double boy;
double yukseklik;

//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklk=y;
}

}

class kututesti2
{
public static void main(String args[])
{
double hacim;
String s="";
Kutu kutu1=new Kutu(10.0,20.0,30.0);
hacim=kutu1.en*kutu1.boy*kutu1.yukseklk;
s+="kutu1'in hacmi : "+hacim+"\n";
Kutu kutu2=new Kutu(5.0,10.0,15.0);
hacim=kutu2.en*kutu2.boy*kutu2.yukseklk;
s+="kutu2'nin hacmi : "+hacim;
JOptionPane.showMessageDialog(null,s,
"Kutu testi 2",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}
```



Bu program bir öncekinin aynı sonucu verir. Fakat burada bir önceki programda yapılan

Kutu kutu1=new Kutu();

Tanımlı yerini

Kutu kutu1=new Kutu(10.0,20.0,30.0);

Tanımlama bırakmış ve kututesti2 sınıfı yazılırken en ,boy yükseklik değişkenlerinin yeniden tanımlanması ihtiyacını ortadan kaldırmıştır.

3.3 DİĞER METOTLAR

Tekrar yukarıda incelediğimiz kutu sınıfını göz önüne alalım. Hacim bir kutunun özellikleri arasında yer alır, ve her zaman kutunun hacminin hesaplanması gerekebilir. Bu yüzden hacim işlemi Kutu sınıfının içinde tanımlamak daha ideal olacaktır. Program 3.4 deki kutuorneği3.java programında hacim hesaplarını yapan hacim metodu Kutu sınıfının içinde tanımlanmıştır.

Program 3.4 : Kurucu metod örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan ve hacmi hacim() metoduyla hesaplayan [kututesti3.java](#) programı

```
import javax.swing.JOptionPane; // giriş çıkış

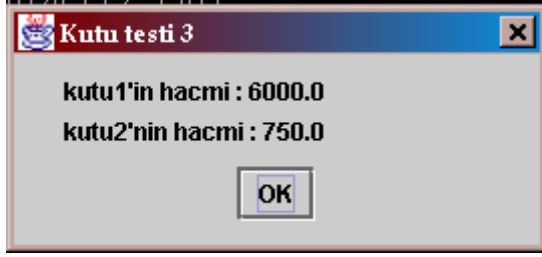
class Kutu
{
double en;
double boy;
double yukseklik;

//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklk=y;
}

//bu metod kutunun hacmini hesaplar
double hacim()
{
return en*boy*yukseklk;
}
}

class kututesti3
{
public static void main(String args[])
{
double hacim;
String s="";
Kutu kutu1=new Kutu(10.0,20.0,30.0);
s+="kutu1\in hacmi : "+kutu1.hacim()+"\n";
Kutu kutu2=new Kutu(5.0,10.0,15.0);
s+="kutu2\in hacmi : "+kutu2.hacim();
JOptionPane.showMessageDialog(null,s,
"Kutu testi 3",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}
```

Program çıktısı yine aynı olacaktır.



Bir önceki bölümde metod yazımı konusunda anlattıklarımızın hepsini buraya ilave edebiliriz. Sınıfın içinde her türlü metodu yazarak çalıştırma hakkına sahibiz. Ek olarak sınıf tanımını tam olarak yapıldığı için artık sınıf değişkenleri ile interaktif olarak çalışan dinamik metodları da daha iyi anlama olasılığı oluştu.

Program 3.5 : çeşitli metodların sınıf içinde kullanıldığı metod örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan ve hacmi hacim() metoduyla hesaplayan çıktı ve yaz metodlarıyla çıktı alan [kututesti4.java](#) programı

```
import javax.swing.JOptionPane; // giriş çıkış

class Kutu
{
double en;
double boy;
double yukseklik;

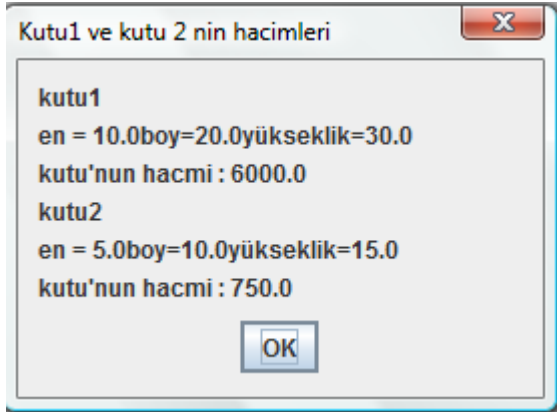
//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklk=y;
}

//bu metod kutunun hacmini hesaplar
double hacim()
{return en*boy*yukseklk;}

//bu metod kutu sınıfı için çıktı hazırlar
public String cikti(String s1)
{String s=s1+"\n"+"en = "+en+"boy="+boy+"yükseklk="+yukseklk+"\n";
s+="kutu\nnun hacmi : "+hacim()+"\n";
return s;
}
public static void yaz(String s,String s2)
{JOptionPane.showMessageDialog(null,s,s2,JOptionPane.PLAIN_MESSAGE);}
}

class kututesti4
{
public static void main(String args[])
{
double hacim;
Kutu kutu1=new Kutu(10.0,20.0,30.0);
```

```
Kutu kutu2=new Kutu(5.0,10.0,15.0);
String s=kutu1.cikti("kutu1")+kutu2.cikti("kutu2");
Kutu.yaz(s,"Kutu1 ve kutu 2 nin hacimleri");
}
}
```



Son programımızda yaz metodunu static cikti metodunu dinamik olarak tanımladık.

3.4 THIS DEYİMİNİN KULLANIMI

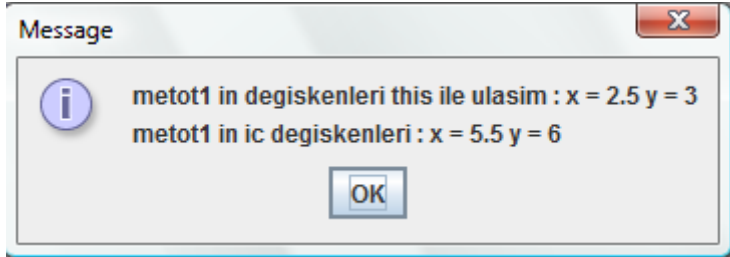
Bazen bir metodun içinde o metodun ait olduğu sınıftan yaratılacak nesneyi veya o nesnenin bir alt değişkenini tanımlamak gerekir. Nesne daha tanımlanmadığından direk olarak nesne ismini kullanamayız. Bunun yerine java **this** deyimini kullanır. This deyimini bilhassa sınıfa ait değişken isimlerinin aynıısı metotda kullanılmışsa da işe yarar. Bu durumda tüm sınıfa ait değişkenler **this.değişken_ismi** komutuyla çağırılabilir

Program 3.4.1 : [metotvesinifdeg3.java](#) programıyla ve this deyimini yardımıyla metodun içinde yerel ve sınıfa ait değişkenlere ulaşım.

```
import javax.swing.*;

public class metodvesinifdeg3
{
//buradaki degiskenler tum sinifa aittir
double x ;
int y ;
void metot1() //metota hiç bir değişken girmiyor
{
double x;
int y;
x=5.5;
y=6;
//tum sinifa at degiskenler this kelimesi ile birlikte kullanilabilirler
String s="metot1 in degiskenleri this ile ulasim : x = "+this.x+" y = "+this.y+"\n";
s+="metot1 in ic degiskenleri : x = "+x+" y = "+y;
JOptionPane.showMessageDialog(null,s);
}
public void metot2()
{
x = 2.5;
y = 3;
metot1();
}
```

```
public static void main(String arg[])
{
    metotvesinifdeg3 w=new metotvesinifdeg3();
    w.metod2();
}
}
```



Sınıfa ait her değişkeni başına **this.** getirerek kullanabiliriz. Fakat this kullanımı mutlaka gerekli değilse kullanılmaması tavsiye edilir.

Örneğin yukarıdaki kurucu metot kutu

```
//kurucu metot kutu
kutu(double en,double boy,double yukseklik)
{
    this.en=en;
    this.boy=boy;
    this.yukseklik=yukseklik;
}
```

şeklinde this değişkenini kullanarak da yazılabilirdi. Burada girdi değişkeni ile sınıf değişkeninin adı aynı verilmiştir. This sol taraftaki değişkenin sınıf değişkeni olduğunu bilgisayara bildirmektedir.

Bir sonraki örnekte bir robot sınıfı tanımlanmıştır. Çeşitli kurucu metodlarla robotun ilk pozisyonu ve adı tanımlanmıştır. Robotu belli yönlere döndüren ve ilerleten ek metodlar mevcuttur. Dördüncü kurucu metodda girdi değişkeni olarak robot sınıfından r1 değişkenini kullandığımızı not edelim.

Program 3.4.2 robot.java : a simple class application

```
public class robot
{
    public String name;
    public double R;
    public double theta;
    public robot(String is) {R=0;theta=0;name=is;}
    public robot(String is,double Ri) {name=is;R=Ri;theta=0;}
    public robot(String is,double Ri,double theta_angle) {name=is;R=Ri;theta=theta_angle*Math.PI/180.0;}
    public robot(robot r1) {name=r1.name;R=r1.R;theta=r1.theta;}
    public void kuzeye_don() {theta=Math.PI/2.0;}
    public void guneye_don() {theta=3.0*Math.PI/2.0;}
    public void batiya_don() {theta=Math.PI;}
    public void doguya_don() {theta=Math.PI;}
    public void don(double angle) {theta+=angle*Math.PI/180.0;}
    public void ilerle() {R+=1;}
    public void ilerle(double Ri) {R+=Ri;}
    public void geri_git() {R-=1;}
    public void geri_git(double Ri) {R-=Ri;}
    public String cikti()
}
```

```

{ String s="-----"+name+"-----\n";
  s+="karteziyen koordinatlar x = "+R*Math.sin(theta)+" y = "+R*Math.cos(theta)+" \n";
  s+="polar koordinatlar R = "+R+" teta açısı = "+theta*180/Math.PI+" \n";return s; }
}

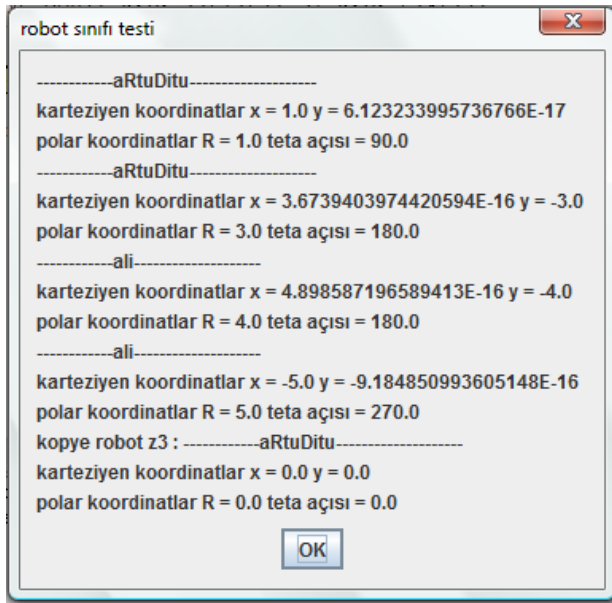
```

Program 3.4.3 robottest.java : a test program for class robot : a simple class application

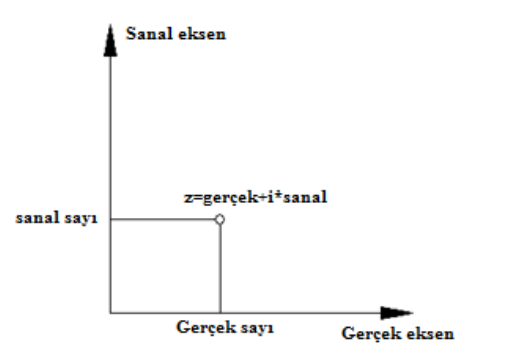
```

import javax.swing.*;public
class robottest
{ public static void main(String args[])
{ robot R2D2=new robot("aRtuDitu");
  robot z2=new robot("ali",1.0,90.0);
  robot z3=new robot(R2D2);
  String s="";
  R2D2.kuzeye_don();R2D2.ilerle();s+=R2D2.cikti();
  R2D2.batiya_don();R2D2.ilerle(2.0);s+=R2D2.cikti();
  z2.doguya_don();z2.ilerle(3.0);s+=z2.cikti();
  z2.guneye_don();z2.ilerle();s+=z2.cikti();
  s+="kopye robot z3 : "+z3.cikti();
  JOptionPane.showMessageDialog(null,s, "robot sınıfı testi",JOptionPane.PLAIN_MESSAGE);
}
}

```



Bir sonraki sınıf örneğimizde kompleks sayılar tanımlanmaktadır. Kompleks sayılar iki alt bileşeni olan sayılardır $z = Gerçek_sayı + \sqrt{-1} * sanal_sayı = Gerçek_sayı + i * sanal_sayı$, burada $i = \sqrt{-1}$ değinin kısaltılmış yazılışdır.



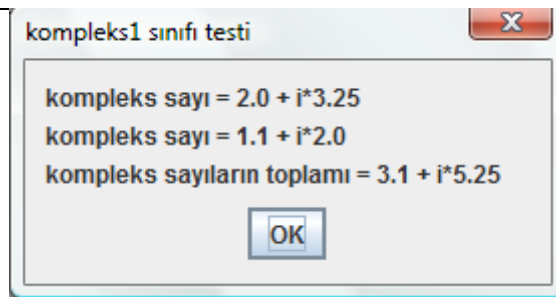
İlk tanımlanacak kompleks1 sınıfında sadece gerçek ve sanal sayılar tanımlandı

Program 3.4.4 kompleks1.java : a simple class application

```
public class kompleks1
{ public double gercek;
  public double sanal;
}
```

Program 3.4.5 kompleks1test.java : a test program for class complex1 : a simple class application

```
import javax.swing.*;
public class kompleks1test
{ public static void main(String args[])
  { kompleks1 z1=new kompleks1();
    z1.gercek=2.0;
    z1.sanal=3.25;
    String s="kompleks sayı = "+z1.gercek+" + i*"+z1.sanal+" \n";
    kompleks1 z2=new kompleks1();
    z2.gercek=1.1;
    z2.sanal=2.0;
    s+="kompleks sayı = "+z2.gercek+" + i*"+z2.sanal+" \n";
    kompleks1 z3=new kompleks1();
    z3.gercek=z1.gercek+z2.gercek;
    z3.sanal=z1.sanal+z2.sanal;
    s+="kompleks sayıların toplamı = "+z3.gercek+" + i*"+z3.sanal+" \n";
    JOptionPane.showMessageDialog(null,s, "kompleks1 sınıfı testi",JOptionPane.PLAIN_MESSAGE);
  }
}
```



Kompleks1 sınıfı kullanılırken yapılan işlemlerin büyük bir kısmı yine main metodunda yapıldı. Sınıf kavramına geçmemizin temel nedeni bir sınıfın kendine ait tüm işlemleri içerisinde barındırdığı değişkenler ve metodlar üzerinden yapmasıdır. Bu yüzden toplama işlemini sınıfın içine taşımamız daha doğru bir programlama oluşturacaktır.

Program 3.4.6 complex3.java : a simple class application

```
public class kompleks3
{ public double gercek;
  public double sanal;
  public kompleks3(double g,double s)
  { gercek=g;
    sanal=s;
  }
  public kompleks3(kompleks3 s1)
  { gercek=s1.gercek;
    sanal=s1.sanal;
  }
}
```

```

public void topla(kompleks3 s1)
{ gercek+=s1.gercek;
  sanal+=s1.sanal;
}
public static kompleks3 topla(kompleks3 s1,kompleks3 s2)
{komples3 z3=new kompleks3((s1.gercek+s2.gercek),(s1.sanal+s2.sanal));
  return z3;
}
public String toString()
{ String s="+gercek+" + i*"+sanal+" \n";
  return s;
}
}

```

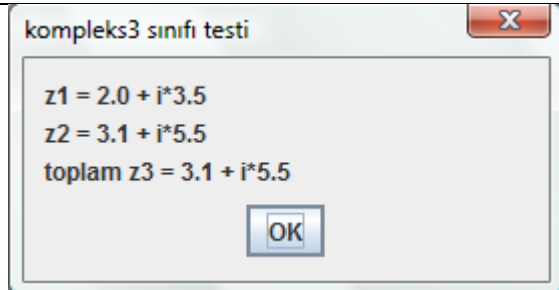
İki değişik test programı tanımlanmıştır. Birinci program kompleks3test'te dinamik topla metodu kullanılarak toplama yapılmıştır.

Program 3.4.7 kompleks3test.java : a simple class application

```

import javax.swing.*;
public class kompleks3test
{ public static void main(String args[])
  { kompleks3 z1=new kompleks3(2.0,3.5);
    kompleks3 z2=new kompleks3(1.1,2.0);
    kompleks3 z3=new kompleks3(z2);
    z3.topla(z1);
    String s="z1 = "+z1.toString()+"z2 = "+z2.toString()+"toplama z3 = "+z3.toString();
    JOptionPane.showMessageDialog(null,s, "kompleks3 sınıfı testi",JOptionPane.PLAIN_MESSAGE);
  }
}

```



İkinci test programı kompleks3test1 de ise statik topla metodu kullanılmıştır. Burada z1 z2 ve z3 olarak 3 tane kompleks değişken tanımlanmış, statik olarak z1 ve z2 değerleri toplanarak z3 değerine atanmıştır.

Program 3.4.8 kompleks3test1.java : sınıf uygulama örneği

```

import javax.swing.*;
public class kompleks3test1
{ public static void main(String args[])
  { kompleks3 z1=new kompleks3(2.0,3.5);
    kompleks3 z2=new kompleks3(1.1,2.0);
    kompleks3 z3=komples3.topla(z1,z2);
    String s="z1 = "+z1.toString()+"z2 = "+z2.toString()+"total z3 = "+z3.toString();
    JOptionPane.showMessageDialog(null,s, "kompleks3 sınıfı testi",JOptionPane.PLAIN_MESSAGE);
  }
}

```

Son örneğimiz bir şirket programı olacak. Bu program şirketin hesabını incelemektedir (çok basit olarak)

Program 3.4.9 sirket.java : sınıf uygulama örneği

```
public class sirket
{ public String isim;
  public double sermaye,satis_tutari,kar;
  public int islem;
  //bu metod bir sirket muhasebesi tanımlar

  //kurucu metod
  public sirket(String is,double Ri)
  { isim=is;
    sermaye=Ri;
    satis_tutari=Ri;
    kar=0;
    islem=0;
  }

  public void satin_al(double x)
  { satis_tutari-=x;
    islem++;
    if(satis_tutari<=0)
      {System.out.println("sirket is broken "+islem);}
  }

  public void sat(double x)
  { satis_tutari+=x;
    kar=satis_tutari-sermaye;
    islem++;
  }

  public String hesap()
  { String s="-----"+isim+"-----\n";
    if(satis_tutari<=0)
      s+="Şirket iflas etti !!!! \n";
    else
      s+=" sermaye = "+sermaye+" TL\n";
      s+=" satış tutarı = "+satis_tutari+" TL\n";
      s+=" kar = "+kar+" TL\n";
      s+=" işlem sayısı = "+islem+" \n";
    return s;
  }
}
```

Program 3.4.10 sirketest.java : sınıf uygulama örneği

```
import javax.swing.*;

public class sirketest
{ public static void main(String args[])
  {
    sirket T=new sirket("Defne A.Ş.",1.0e6);
    for(int i=0;i<100;i++) {T.satin_al(100);T.sat(400);}
  }
}
```



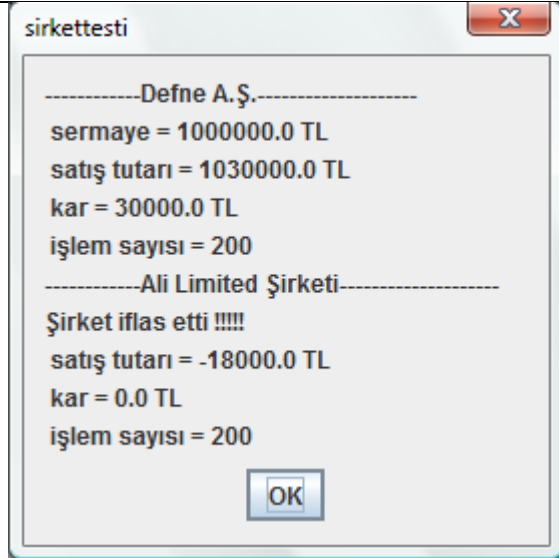
```

String s=T.hesap();

sirket A=new sirket("Ali Limited Şirketi",1000.0);
for(int i=0;i<100;i++) {A.satin_al(100);A.satin_al(90);}
s+=A.hesap();

JOptionPane.showMessageDialog(null,s, "sirkettesti",JOptionPane.PLAIN_MESSAGE);
}
}

```



3.5 SINIF DEĞİŞKENLERİNİN DIŞ DÜNYADAN GİZLENMESİ

Yukarda incelediğimiz kutu metodu. Program 3.5 de daha detaylı yazılmıştır. Burada en boy ve yükseklik değişkenleri private olarak tanımlanmıştır. Private olarak tanımlanan değişkenlere dışarıdan direk olarak

```

Kutu x=new Kutu(3.0,5.0,7.0);
x.en=8.0;

```

gibi direk ulaşım mevcut değildir. Bu değişkenlere ulaşım (değerlerini değiştirme veya çağırma) ancak ulaşım metodları adını verdiğimiz metodlar aracılığıyla yapılabilir.

Program 3.5.1 : Kurucu metod örneği, Kutu sınıfı iki nesneyi kurucu(constructor) metodu yardımıyla tanımlayan ve hacmi hacim() metoduyla hesaplayan ve diğer metod tanımlarının da bulunduğu kututesti4.java programı.

```

import javax.swing.JOptionPane; // giriş çıkış

class Kutu
{
private double en;
private double boy;
private double yukseklik;

//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;

```

```

yukseklık=y;
}

//sınıf deęişkenlerini deęiştirme set metodu
public void setKutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklık=y;
}

//sınıf deęişkenlerine ulaşma (get) metotları
public double getEn()
{return en;}

public double getBoy()
{return boy;}

public double getTukseklık()
{return yukseklık;}

//bu metot kutunun hacmini hesaplar (işlem- hesap) metodu
public double hacim()
{
return en*boy*yukseklık;
}

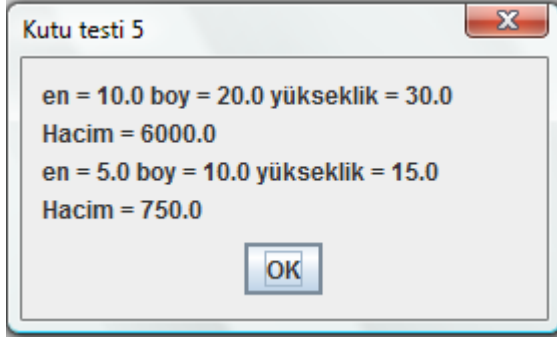
//string çıktı metodu
public String toString()
{
return("en = "+en+" boy = "+boy+" yukseklık = "+yukseklık+"\n"+
"Hacim = "+hacim()+"\n");
}
}

class kututesti5
{
public static void main(String args[])
{
double hacim;
String s="";
Kutu kutu1=new Kutu(10.0,20.0,30.0);
s+=kutu1.toString();
Kutu kutu2=new Kutu(5.0,10.0,15.0);
s+=kutu2.toString();
JOptionPane.showMessageDialog(null,s,
"Kutu testi 5",JOptionPane.PLAIN_MESSAGE);
}
}

```

Bu programın çıktısı :

03004.JPG



Şekil 3.4 kututesti4.java

İkinci bir örnek olarak dörtişlemhesapmakinesi sınıfına bakalım. Burada da sınıf değişkenleri protected olarak tanımlanmıştır. Aşağıda dört işlem (toplama çıkarma carpma bolme yapan dortislemhesapmakinesi sınıfını görüyorsunuz.

Program 3.5.2 : dörtislemhesapmakinesi.java programı ve sınıf değişkenlerinin değerlerinin metot kullanılarak değiştirilmesi

```
public class dortislemhesapmakinesi
{
    //sınıf değişkenleri x sonuc hafıza
    protected double x,sonuc,hafıza;
    // kurucu metot dortislemhesapmakinesi
    public dortislemhesapmakinesi(double y)
    {
        //constructor
        sonuc=y;
        hafıza=0;
    }
    // ikinci bir kurucu metot dortislemhesapmakinesi
    public dortislemhesapmakinesi()
    { sonuc=0; hafıza=0; }
    public double topla(double x)
    { sonuc+=x; return sonuc; }
    public double cikar(double x)
    { sonuc-=x; return sonuc; }
    public double carp(double x)
    { sonuc*=x; return sonuc; }
    public double bol(double x)
    { sonuc/=x; return sonuc; }
    //bu metotla sonuc değeri değiştirilir
    public void gir(double x)
    { sonuc=x; }
    public double oku()
    {
        return sonuc;
    } public void C()
    { sonuc=0; }
    public void M(double x)
    { //hafızaya al;
        hafıza=x;
    }
    public double MR()
    { //hafızayı çağır
        return hafıza;
    }
}
```

```

}
public void MTopla(double x)
{
hafıza+=x;
}

public void Mcikar(double x)
{
hafıza-=x;
}
public double isaretdegistir()
{
sonuc=-sonuc;
return sonuc;
}

public String cikti()
{
return ""+sonuc;
}
public String MtoString()
{
return ""+hafıza;
}
}

```

Bu sınıfta (class) sınıf değişkeni **sonuc** değerlerinin değiştirilmesi için **gir** metodu yazılmıştır. Bunun sebebi programlarda sınıfa ait değişkenlere dışarıdan direk ulaşımın önlenmesini sağlamaktır. İyi programlama şartlarından birisi sınıfın değişkenlerine sadece o sınıfın metotları üzerinden ulaşmaktır. Herkese serbest ulaşım verilmez. Hatta bunu sağlamak için değişkenlerin başına sadece bu sınıfa aittir dışarıdan kullanılamaz anlamındaki **private** sözcüğü getirilebilir. **private** sözcüğü yerine ancak dışarda bu sınıftan yaratılana yeni sınıflar (akraba sınıflar) kullanabilir, yabancı sınıflar kullanamaz anlamındaki **protected** sözcüğü de sık olarak kullanılır. Protected sözcüğü aynı directorydeki diğer dosyaları da (public) gibi kabula eder, fakat başka direktorylerdeki dosyalar için private olarak davranır.

dortislemhesapmakinesi.java programının x, sonuc ve hafıza değişkenleri için **protected** sözcüğü kullandığımız için

```

dortislemhesapmakinesi h = new dortislemhesapmakinesi();
h.sonuc = 2.0;

```

şeklinde ulaşılabilir. Aynı işlemi yapmak için aşağıdaki üç yoldan birini kullanabiliriz.

```

dortislemhesapmakinesi h = new dortislemhesapmakinesi();
h.gir(2.0);

```

veya

```

dortislemhesapmakinesi h = new dortislemhesapmakinesi(2.0);

```

şeklinde sonuç değişkenine 2.0 değeri atanır.

Hafıza değişkeninin değerinin değiştirilmesi için ise :

```

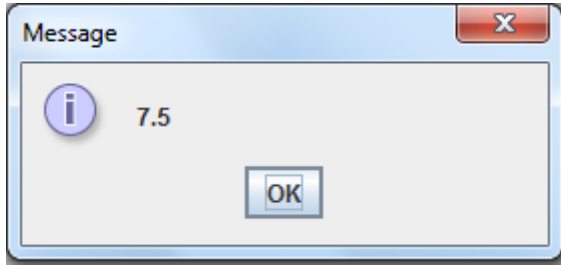
h.M(3.0);

```

metodu çağırılır.

Program 3.5.3: dörtislemhesapmakinası programı'nı kullanan test programı

```
import javax.swing.*;
public class dortislemhesapmakinasitest
{
public static void main(String arg[])
{
dortislemhesapmakinası d=new dortislemhesapmakinası(2.0);
d.topla(3.0);
d.carp(1.5);
JOptionPane.showMessageDialog(null,d.cikti());
}
}
```



Sınıf değişkenlerinin dış dünyadan gizlenmesi niçin gereklidir? Bunun en önemli sebebi program güvenliğinin sağlanmasıdır. Değişkenlere dışarıdan direk müdahale edilebilirse, değerleri de değiştirilebilir. Örneğin değişkeni bir banka hesabı veya şirketin özel hesabı olarak düşünebiliriz. Dış dünyadan gizlenmemiş bir değişkenin değeri değiştirilebilir. Private veya protected değişkenler ise ancak metotlar üzerinden değiştirilebilir, burada yeterli güvenlik önlemlerini almak mümkündür. Bir banka hesabı programı verelim

Program 3.5.3 : bankahesabi.java programı ve sınıf değişkenlerinin değerlerinin metot kullanılarak değiştirilmesi

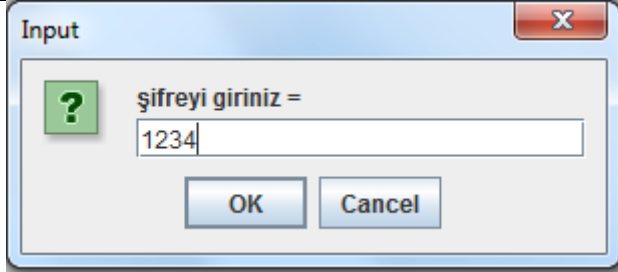
```
import javax.swing.*;
public class bankahesabi
{
private String bankahesapno;
private int sifre;
private int sayi;

public bankahesabi(String an,int c)
{
bankahesapno=an;
sifre=c;
sayi=0;
}
public String output()
{ while(sayi<3)
{ sayi++;
int c1=Integer.parseInt(JOptionPane.showInputDialog("şifreyi giriniz = "));
{if(c1==sifre) return bankahesapno;
}
}
return "yanlış şifre girildi";
}
}
```

Program 3.6 : bankahesabitesti.java programı ve sınıf değişkenlerinin değerlerinin metot kullanılarak değiştirilmesi

```
public class bankahesabitesti
{
```

```
public static void main(String arg[])
{
bankahesabi isim=new bankahesabi("123A4B5",12345);
System.out.println(isim.output());
}
}
```



```
----- Capture Output -----
> "D:\java\bin\javaw.exe" bankahesabitest1
123A4B5
> Terminated with exit code 0.
```

Görüldüğü gibi böyle bir programda hesap numarasını açık olarak veremezdik, Değişken değerini ancak şifreyi bilen gerçek müşteriye açabiliriz.

3.6 ALIŞTIRMALAR

1. Program 3.4.6 kompleks3.java da kompleks sayılar tanımlanmıştı. İki kompleks sayının çarpımı $z1=Re1+i*Im1$ ve $z2=Re2+i*Im2$ kompleks sayılar olsun:

$z1*z2=(Re1*Re2-Im1*Im2) + i*(Re1*Im2+Re2*Im1)$ olarak yazılabilir. Toplama işlemine paralel olarak kompleks3.java sınıfına çarpma işlemi ekleyiniz.

public void carp(kompleks3 s1)

public static kompleks3 carp(kompleks3 s1,kompleks3 s2)

ve metodlarınızı bir test programında test ediniz.

2. İki boyutlu karteziyen koordinat sisteminde nokta bu iki boyutu üzerinden ifade edilebilir. Örneğin $P1(x1,y1)$ x ekseninde $x1$ y ekseninde $y1$ boyutuna sahip noktayı belirlemektedir. Bir nokta sınıfı yazarak kurucu metodlar ve çıktı metodlarıyla nokta noktayı belirleyebilelim. Bir test metodu yazıp noktamızı sınıfa girip çıktı metoduyla yazdıralım.

3. Program 3.5.1 de verilen kutu sınıfına private olarak String tipi renk değişkenini ekleyin ve renk değişkenini okuyacak ve değer atayacak renngioku ve rengigir metodlarını oluşturunuz.

public String renngioku() {.....}

public void write_color(String col) {.....}

4. Yukardaki banka hesabı programını biraz daha geliştirerek gerçek bir banka hesabında olan şifre değiştirme ve hesap balansı alma kısımlarını verdiğiniz programı inceleyiniz

```
import javax.swing.*;

public class bankahesabi1
{
private String isim;
private int bankahesapno;
private String sifre;
private int sifre_deneme;
private static int maximum_sifre_deneme=3;
private double hesap_para_miktari;
```

```

public bankahesabi1(String iisim,int i_bankahesapno,String i_sifre,double pgir)
{
    isim=iisim;
    bankahesapno=i_bankahesapno;
    sifre=i_sifre;
    hesap_para_miktari=pgir;
    sifre_deneme=0;
}

public String hesabi_goster()
{
    String s1="null";
    String sifre_oku;
    int bankahesapno_oku=Integer.parseInt(JOptionPane.showInputDialog("hesap numarası = "));
    while(sifre_deneme<maximum_sifre_deneme)
    {
        sifre_oku=JOptionPane.showInputDialog("sifre = ");
        if(sifre_oku.equals(sifre)){sifre_deneme=0;return yaz();}
        else{sifre_deneme++;JOptionPane.showMessageDialog(null,"sifreniz hatalı lütfen tekrar deneyiniz","Hatalı sifre",JOptionPane.PLAIN_MESSAGE);}
    }
    return s1;
}

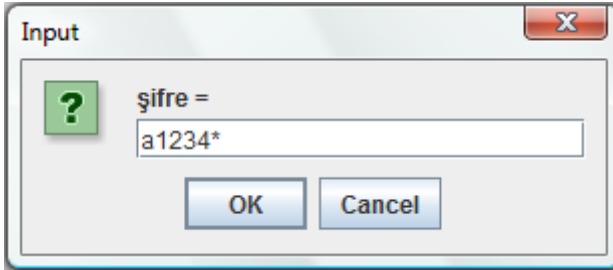
public String sifreyi_degistir()
{
    String s1="şifreyi değiştir : ";
    String sifre_oku;
    while(sifre_deneme<maximum_sifre_deneme)
    {
        sifre_oku=JOptionPane.showInputDialog("şifre = ");
        if(sifre_oku.equals(sifre) )
        {
            String newsifre1=JOptionPane.showInputDialog("yeni şifrenizi giriniz = ");
            String newsifre2=JOptionPane.showInputDialog("yeni şifrenizi tekrar giriniz = ");
            if(newsifre1.equals(newsifre2))
            {sifre=newsifre2;s1="şifreniz başarıyla değiştirildi";break;}
            else
            {JOptionPane.showMessageDialog(null,"girdiğiniz 2 şifre aynı değil ","yanlış yeni şifre girişi",JOptionPane.PLAIN_MESSAGE);s1="sifre change failuare";}
            sifre_deneme=0;
        }
        else
        {JOptionPane.showMessageDialog(null,"şifreniz hatalı lütfen yeniden deneyiniz","yanlış şifre girişi",JOptionPane.PLAIN_MESSAGE);
        sifre_deneme++;s1="şifre hatası";
        }
    }
    return s1;
}

private String yaz()
{
    String s="Hesap ismi = "+isim+"\n";
    s+="Banka hesap numarası = "+bankahesapno+"\n";
    s+="Hesaptaki para miktarı = "+hesap_para_miktari+" T.L.\n";
    return s;
}
}

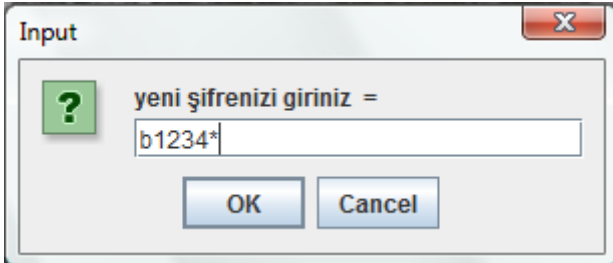
```

```
import javax.swing.*;

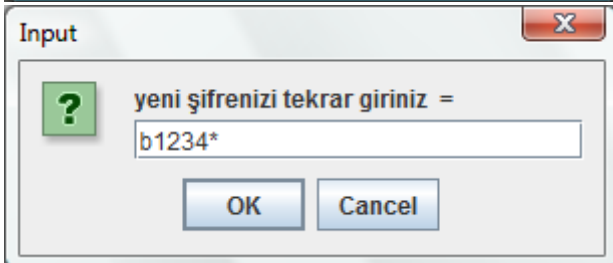
public class bankaheسابtesti1
{
public static void main(String arg[])
{
bankaheسابi h1=new bankaheسابi("ali veli",12345,"a1234*",100000.0);
JOptionPane.showMessageDialog(null,h1.sifreyi_degistir(),"Banka hesap numarası : ",JOptionPane.PLAIN_MESSAGE);
JOptionPane.showMessageDialog(null,h1.heسابi_goster(),"Bank hesap bilgileri : ",JOptionPane.PLAIN_MESSAGE);
}}
}
```



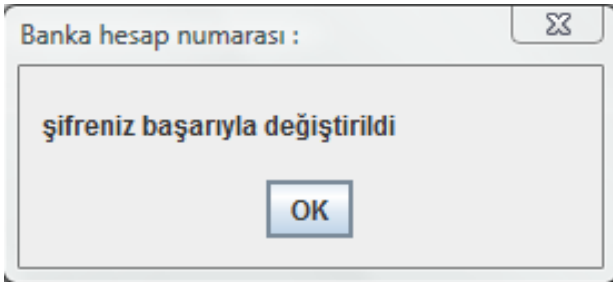
Input dialog box titled "Input" with a close button (X). The dialog contains a green question mark icon, the text "şifre =", and a text input field containing "a1234*". Below the input field are "OK" and "Cancel" buttons.



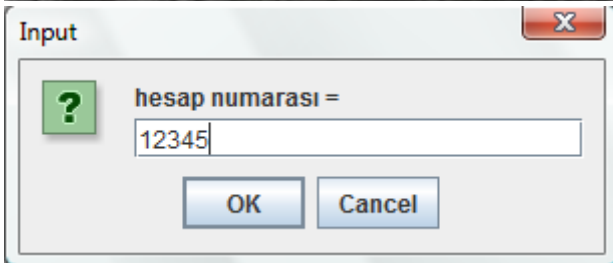
Input dialog box titled "Input" with a close button (X). The dialog contains a green question mark icon, the text "yeni şifrenizi giriniz =", and a text input field containing "b1234*". Below the input field are "OK" and "Cancel" buttons.



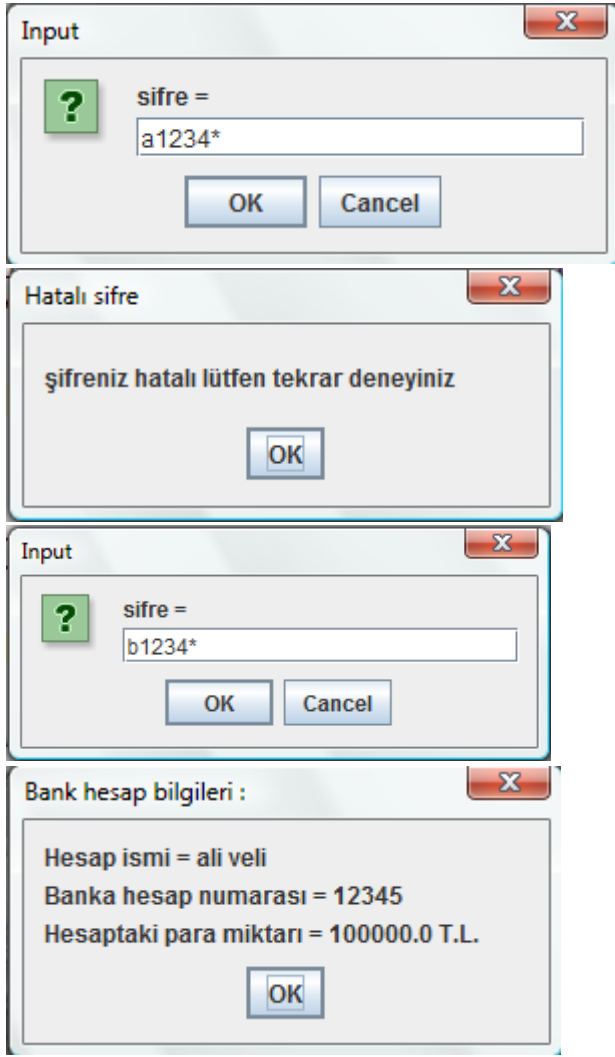
Input dialog box titled "Input" with a close button (X). The dialog contains a green question mark icon, the text "yeni şifrenizi tekrar giriniz =", and a text input field containing "b1234*". Below the input field are "OK" and "Cancel" buttons.



Banka hesap numarası : dialog box with a close button (X). The dialog contains the text "şifreniz başarıyla değiştirildi" and an "OK" button.



Input dialog box titled "Input" with a close button (X). The dialog contains a green question mark icon, the text "hesap numarası =", and a text input field containing "12345". Below the input field are "OK" and "Cancel" buttons.



3.7 PROGRAMLAR

1) **nokta.java** ve **ucgen.java** sınıfları verilmiş. Uchen sınıfı üçgenin alanını ve çevresini hesaplamak için yazılmış bir sınıftır. Ancak çevresinin hesap formülünün girilmesi unutulmuş. Doğru formülü girerek sınıfı tamamlayınız ve bir test programında çalıştırınız. Nokta sınıfı ve üçgen sınıfı düzlem koordinat sisteminde tanımlanmıştır.

```
public class nokta
{
public double xi,xj;

public nokta()
{
xi=0;
xj=0;
}

public nokta(double yi,double yj)
{
xi=yi;
xj=yj;
}
```

```

public nokta(nokta y)
{
xi=y.xi;
xj=y.xj;
}

public void noktaGir(double yi,double yj)
{
xi=yi;
xj=yj;
}

public void noktaGir(nokta y)
{
xi=y.xi;
xj=y.xj;
}

public void topla(nokta y)
{
xi+=y.xi;
xj+=y.xj;
}

public static nokta topla(nokta y1,nokta y2)
{
nokta x=new nokta(y1);
x.topla(y2);
return x;
}

public void fark(nokta y)
{
xi-=y.xi;
xj-=y.xj;
}

public static nokta fark(nokta y1,nokta y2)
{
nokta x=new nokta(y1);
x.fark(y2);
return x;
}

public nokta noktaOku()
{
return this;
}

public double R(nokta y)
{
//iki nokta arasındaki mesafe
return Math.sqrt((xi-y.xi)*(xi-y.xi)+(xj-y.xj)*(xj-y.xj));
}

public double R()
{

```

```

public class ucgen
{
nokta p1,p2,p3;
public ucgen(double p1i,double p1j,double p2i,double p2j,double p3i,double p3j)
{
p1=new nokta(p1i,p1j);
p2=new nokta(p2i,p2j);
p3=new nokta(p3i,p3j);
}
public ucgen(nokta p1i,nokta p2i,nokta p3i)
{
p1=p1i;
p2=p2i;
p3=p3i;
}

public double alan()
{
double A=0.5*((p1.xi*p2.xj-p2.xi*p1.xj)+(p3.xi*p1.xj-p1.xi*p3.xj)+(p2.xi*p3.xj-p3.xi*p2.xj));
return A;
}
public double cevre()
{
double C=0;
return C;
}
}

```

2) nokta3D ve kuvvet sınıfları kuvvet ve moment dönüşümlerini yapmak için oluşturulmuştur. Kuvvey sınıfı kuvvetin 3 boyutlu etki noktası, üç boyutlu kuvvet ve moment komponentlerini kapsamaktadır. Bu sınıf yardımıyla bir kuvvet veya kuvvet çiftini yeni bir noktaya taşıyabiliriz. Programlar ve bir örnek program verilmiştir.

$n_1(1,1,1)$ noktasında $F_1(1,1,1)$ ve $M_1(0,0,0)$ ve

$n_2(1,2,3)$ noktasında $F_2(1,0,0)$ ve $M_2(0,0,0)$ kuvvet-moment çifti verilmişse, bu çiftin $n_3(0,0,0)$ noktasındaki toplam etkisini hesaplayınız.

```

public class nokta3D
{
public double xi,xj,xk;

public nokta3D()
{
xi=0;
xj=0;
xk=0;
}
}

```

```
public nokta3D(double yi,double yj,double yk)
{
xi=yi;
xj=yj;
xk=yk;
}

public nokta3D(nokta3D y)
{
xi=y.xi;
xj=y.xj;
xk=y.xk;
}

public void noktaGir(double yi,double yj,double yk)
{
xi=yi;
xj=yj;
xk=yk;
}

public void noktaGir(nokta3D y)
{
xi=y.xi;
xj=y.xj;
xk=y.xk;
}

public void toplama(nokta3D y)
{
xi+=y.xi;
xj+=y.xj;
xk+=y.xk;
}

public static nokta3D toplama(nokta3D y1,nokta3D y2)
{
nokta3D x=new nokta3D(y1);
x.toplama(y2);
return x;
}

public void fark(nokta3D y)
{
xi-=y.xi;
xj-=y.xj;
xk-=y.xk;
}

public static nokta3D fark(nokta3D y1,nokta3D y2)
{
nokta3D x=new nokta3D(y1);
x.fark(y2);
return x;
}

public nokta3D noktaOku()
```

```

{
return this;
}

public double R(nokta3D y)
{
//iki nokta arasındaki mesafe
return Math.sqrt((xi-y.xi)*(xi-y.xi)+(xj-y.xj)*(xj-y.xj)+(xk-y.xk)*(xk-y.xk));
}

public double R()
{
//koordinat merkezi ile nokta arasındaki mesafe
return Math.sqrt(xi*xi+xj*xj+xk*xk);
}

public double cosai()
{
//noktanın i eksenine yaptığı açının cosinüsü
return xi/R();
}

public double cosaj()
{
//noktanın j eksenine yaptığı açının cosinüsü
return xj/R();
}

public double cosak()
{
//noktanın k eksenine yaptığı açının cosinüsü
return xk/R();
}

public boolean esittir(nokta3D v)
{
boolean b=((xi==v.xi)&&(xj==v.xj)&&(xk==v.xk));
return b;
}

public boolean buyuktur(nokta3D v)
{
return (this.R())>v.R();
}

public boolean kucuktur(nokta3D v)
{
return (this.R())<v.R();
}

public String cikti()
{
String s="";
s+=xi+" i ";
if(xj>=0)
s+=" "+xj+" j";
else if(xj<0)
s+="- "+Math.abs(xj)+" j";
}

```

```
if(xk>=0 )
    s+=" "+xk+" k";
else if(xk<0)
    s+=" "+Math.abs(xk)+" k";
return s;
}
}
```

```
public class kuvvet
{
    nokta3D n; // etki noktası
    double Fi,Fj,Fk; // Kuvvet komponentleri
    double Mi,Mj,Mk; // Moment komponentleri

    public kuvvet()
    {
        n=new nokta3D();
        Fi=0;
        Fj=0;
        Fk=0;
        Mi=0;
        Mj=0;
        Mk=0;
    }

    public kuvvet(double yi,double yj,double yk,
        double Ffi,double Ffj,double Ffk,
        double Mfi,double Mfj,double Mfk)
    {
        n=new nokta3D(yi,yj,yk);
        Fi=Ffi;
        Fj=Ffj;
        Fk=Ffk;
        Mi=Mfi;
        Mj=Mfj;
        Mk=Mfk;
    }

    public kuvvet(nokta3D n1,double Ffi,double Ffj,double Ffk,
        double Mfi,double Mfj,double Mfk)
    {
        n=new nokta3D(n1);
        Fi=Ffi;
        Fj=Ffj;
        Fk=Ffk;
        Mi=Mfi;
        Mj=Mfj;
        Mk=Mfk;
    }

    public kuvvet(kuvvet F)
    {
        n=new nokta3D(F.n);
        Fi=F.Fi;
        Fj=F.Fj;
        Fk=F.Fk;
        Mi=F.Mi;
        Mj=F.Mj;
        Mk=F.Mk;
    }
}
```

```

}

public void tasi(nokta3D n1)
{ //kuvveti uygulama noktasını deęiřtirir
  //momentler deęiřir kuvvetler aynı kalır
  double di=n1.xi-n.xi;
  double dj=n1.xj-n.xj;
  double dk=n1.xk-n.xk;
  Mi=Mi+(dj*Fk-dk*Fj);
  Mj=Mj-(di*Fk-dk*Fi);
  Mk=Mk+(di*Fj-dj*Fi);
  n=n1;
}

public static kuvvet tasi(nokta3D n1,kuvvet F1)
{ //kuvveti uygulama noktasını deęiřtirir
  //momentler deęiřir kuvvetler aynı kalır
  kuvvet F=new kuvvet(F1);
  double di=n1.xi-F.n.xi;
  double dj=n1.xj-F.n.xj;
  double dk=n1.xk-F.n.xk;
  F.Mi=F.Mi+(dj*F.Fk-dk*F.Fj);
  F.Mj=F.Mj-(di*F.Fk-dk*F.Fi);
  F.Mk=F.Mk+(di*F.Fj-dj*F.Fi);
  F.n=n1;
  return F;
}

public void topla(kuvvet F)
{ // iki kuvvet toplanıyor, uygulama noktası il kuvvetin uygulama noktası
  // momentler de tařınıyor
  nokta3D n=new nokta3D(F.n);
  double di=F.n.xi-n.xi;
  double dj=F.n.xj-n.xj;
  double dk=F.n.xk-n.xk;
  Mi=Mi+(dj*Fk-dk*Fj);
  Mj=Mj-(di*Fk-dk*Fi);
  Mk=Mk+(di*Fj-dj*Fi);
  Fi+=F.Fi;
  Fj+=F.Fj;
  Fk+=F.Fk;
}

public static kuvvet topla(kuvvet F1,kuvvet F2)
{ // iki kuvvet toplanıyor, uygulama noktası ilk kuvvetin uygulama noktası
  // momentler de tařınıyor
  kuvvet F3=new kuvvet(F1);
  double di=F3.n.xi-F2.n.xi;
  double dj=F3.n.xj-F2.n.xj;
  double dk=F3.n.xk-F2.n.xk;
  F3.Mi+=-(dj*F2.Fk-dk*F2.Fj);
  F3.Mj=-((di*F2.Fk-dk*F2.Fi));
  F3.Mk+=((di*F2.Fj-dj*F2.Fi));
  F3.Fi+=F2.Fi;
  F3.Fj+=F2.Fj;
  F3.Fk+=F2.Fk;
  return F3;
}

```

```

public double F()
{
// Toplam kuvvet komponenti
return Math.sqrt(Fi*Fi+Fj*Fj+Fk*Fk);
}
public double M()
{
// Toplam moment komponenti
return Math.sqrt(Mi*Mi+Mj*Mj+Mk*Mk);
}

public String Fcikti()
{
String s="";
s+=Fi+" i ";
if(Fj>=0 )
s+=" "+Fj+" j";
else if(Fj<0)
s+="- "+Math.abs(Fj)+" j";
if(Fk>=0 )
s+=" "+Fk+" k";
else if(Fk<0)
s+="- "+Math.abs(Fk)+" k";
return s;
}
public String Mcikti()
{
String s="";
s+=Mi+" i ";
if(Mj>=0 )
s+=" "+Mj+" j";
else if(Mj<0)
s+="- "+Math.abs(Mj)+" j";
if(Mk>=0 )
s+=" "+Mk+" k";
else if(Mk<0)
s+="- "+Math.abs(Mk)+" k";
return s;
}

public String Pcikti()
{ return n.cikti();}

public String cikti()
{ return "P = "+Pcikti()+" F = "+Fcikti()+" M = "+Mcikti();}

}

```

```

public class kuvvettest1
{
public static void main(String arg[])
{
nokta3D n1=new nokta3D(0.0,0.0,0.0);
nokta3D n2=new nokta3D(2.0,0.0,0.0);
nokta3D n3=new nokta3D(4.0,0.0,0.0);
kuvvet F1=new kuvvet(n2,0.0,0.0,1.0,0.0,0.0,0.0);
}
}

```



```
kuvvet F2=new kuvvet(n3,0.0,0.0,-1.0,0.0,0.0,0.0);
F1.tasi(n1);
F1.topla(F2);
System.out.println(F1.cikti());
}
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" kuvvetttest1
P = 0.0 i + 0.0 j+ 0.0 k F = 0.0 i + 0.0 j+ 0.0 k M = 0.0 i + 2.0 j+ 0.0 k
> Terminated with exit code 0.
```

```
public class kuvvetttest
{
public static void main(String arg[])
{
nokta3D n1=new nokta3D(0.0,0.0,0.0);
nokta3D n2=new nokta3D(2.0,0.0,0.0);
kuvvet F=new kuvvet(n2,0.0,0.0,1.0,0.0,0.0,0.0);
F.tasi(n1);
System.out.println(F.cikti());
}}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" kuvvetttest
P = 0.0 i + 0.0 j+ 0.0 k F = 0.0 i + 0.0 j+ 1.0 k M = 0.0 i + 2.0 j+ 0.0 k
> Terminated with exit code 0.
```

3) Program 3.4.2 robot.java de çeşitli yönlere hareket edebilen bir robot tasarlamıştık şimdi roborumuzu daha esnek hale getirebilmek için kuzeybatı, kuzeydoğu, göeybatı, güneydoğu yönlerine de dönebilmesini sağlamak istiyoruz. Bunu sağlayacak ek metodları geliştiriniz. Ve bir test metodunda deneyiniz.

4) bayagikesir sınıfı tanımlanmıştır ve kullanımını gösteren küçük bir bayagikesirtest proogramı verilmiştir. Yeni bir test programı oluşturarak aşağıdaki bayagikesir işlemini hesaplatırınız.

$$A = \frac{\frac{1}{2} + 2\frac{3}{4}}{\frac{1}{4} + \frac{2}{3}}$$

```
//=====
// bayagikesir sinifi tanimi
// Dr. Turhan Coban
//=====
import java.io.*;
// sinif bayagikesir
// bu sinif bayagikesir sayilarin matematik
// islemlerini tanimlar
//
class bayagikesir {
// sinif degiskenleri
int pay;
```

```

int payda;
// kurucu metotlar

public bayagikesir()
{
pay=0;
payda=1;
}

public bayagikesir(int npay,int npayda)
{
pay=npay;
payda=npayda;
}

public bayagikesir(int tamsayi,int npay,int npayda)
{
payda=npayda;
pay=npay+tamsayi*npayda;
}

public bayagikesir(int numer)
{
pay=numer;
payda=1;
}
public bayagikesir(bayagikesir c )
{
pay=c.Pay();
payda=c.Payda();
}
// giriş- çıkış metodları

public int enbuyukortakbolen()
{
int n=pay;
int m=payda;
// iki tam sayinin en buyuk ortak bolenini hesaplar
if(n==0)
return m;
if(m==0)
return n;
while(m != n)
{
if(n>m)
n=n-m;
else
m=m-n;
}
return n;
}

public void sadelestir()
{
//sadelestir
int isaret=1;
if(pay<0)
{
isaret=-isaret;
pay=-pay;
}
}

```

```

}
if(payda<0)
{
isaret=-isaret;
payda=-payda;
}
if(payda==0) {pay=0;payda=1;}
int ebob=enbuyukortakbolen();
ebob=Math.abs(ebob);
pay=isaret*pay/ebob;
payda=payda/ebob;
}

public int Pay()
{
return pay;
}

public int Payda()
{
return payda;
}

public void payGir(int r)
{
pay=r;
}

public void paydaGir(int i)
{
payda=i;
}
public void bayagikesirGir(bayagikesir sag)
{
pay=sag.Pay();
payda=sag.Payda();
}

public void bayagikesirGir(int nr,int ni)
{
pay=nr;
payda=ni;
}

public void bayagikesirGir(int nr)
{
pay=nr;
payda=1;
}

public void bayagikesirGir(double d)
{
// tam sayinin bayagikesir esitini yukle
// eh birazcik yaklasim var tabi
bayagikesirGir((int)d*10000,10000);
sadelestir();
}

public double toDouble()
{

```

```

//bayagikesir sayinin gercek sayi esidi
return ((double)pay/(double)payda);
}

public static bayagikesir BayagikesireCevir(double d)
{
// tam sayinin bayagikesir esiti
// eh birazcik yaklasim var tabi
bayagikesir b=new bayagikesir();
b.bayagikesirGir((int)d*10000,10000);
b.sadelestir();
return b;
}

public void topla(bayagikesir sag)
{
//ikinci bir bayagikesirle topla
pay = pay*sag.Payda() + sag.Pay()*payda;
payda = payda*sag.Payda();
sadelestir();
}

public void topla(int sag)
{
//bir gercek sayıyla topla
pay = pay + sag*payda;
sadelestir();
}

public void cikar(bayagikesir sag)
{
//ikinci bir bayagikesirle topla
pay = pay*sag.Payda() - sag.Pay()*payda;
payda = payda*sag.Payda();
sadelestir();
}

public void cikar(int sag)
{
//bir gercek sayıyla topla
pay = pay - sag*payda;
sadelestir();
}

public void carp(bayagikesir sag )
{
//bir bayagikesir sayiyla carp
pay = pay*sag.Pay();
payda = payda*sag.Payda();
sadelestir();
}

public void carp(int sag )
{
//bir gercek sayiyla carp
pay = pay*sag;
sadelestir();
}

public void bol(bayagikesir sag )

```

```

{
//bir bayagikesir sayıyla böl
pay = pay*sag.Payda();
payda = payda*sag.Pay();
sadelestir();
}

public void bol(int sag )
{
//bir pay sayıyla böl
payda = payda*sag;
sadelestir();
}

public static bayagikesir topla(bayagikesir sol, bayagikesir sag)
{
// iki bayagikesir sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1= sol.Pay() * sag.Payda() + sag.Pay() * sol.Payda();
int i1= sol.Payda()* sag.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir topla(bayagikesir sol, int sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sol.Pay() + sag*sol.Payda();
int i1=sol.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir topla(int sol, bayagikesir sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sag.Pay() + sol*sag.Payda();
int i1=sag.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
return sonuc;
}
//=====

public static bayagikesir cikar(bayagikesir sol, bayagikesir sag)
{
// iki bayagikesir sayının farkını return deyimiyle bayagikesir olarak
// aktarır
int r1= sol.Pay() * sag.Payda() - sag.Pay() * sol.Payda();
int i1= sol.Payda()* sag.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
}

```

```

return sonuc;
}

public static bayagikesir cikar(bayagikesir sol, int sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sol.Pay() - sag*sol.Payda();
int i1=sol.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir cikar(int sol, bayagikesir sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sag.Pay() - sol*sag.Payda();
int i1=sag.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir(r1,i1);
sonuc.sadelestir();
return sonuc;
}

//=====
public static bayagikesir carp(bayagikesir sol, bayagikesir sag)
{ // iki bayagikesir sayının carpimini aktar
bayagikesir sonuc;
sonuc=new bayagikesir(sol.Pay()*sag.Pay(),sol.Payda()*sag.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir carp(bayagikesir sol, int sag)
{ // bir bayagikesir ve bir double sayinin carpimini aktar
bayagikesir sonuc;
sonuc=new bayagikesir(sol.Pay()*sag,sol.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir carp(int sol, bayagikesir sag)
{ // bir bayagikesir ve bir double sayinin carpimini aktar
bayagikesir sonuc;
sonuc=new bayagikesir(sag.Pay()*sol,sag.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir bol(bayagikesir sol, bayagikesir sag)
{ // iki bayagikesir sayinin bölümlerini aktar
double a=sag.Pay()*sag.Pay()+sag.Payda()*sag.Payda();
bayagikesir sonuc;
sonuc=new bayagikesir((int)(sol.Pay()*sag.Payda()),(int)(sol.Payda()*sag.Pay()));
sonuc.sadelestir();
return sonuc;
}

```

```

public static bayagikesir bol(bayagikesir sol, int sag)
{ // bayagikesir sayıyı double sayıya bol
bayagikesir sonuc;
sonuc=new bayagikesir(sol.Pay(),(sol.Payda()*sag));
sonuc.sadelestir();
return sonuc;
}

public boolean kucuktur(bayagikesir sol,bayagikesir sag)
{
// less then karşılaştırması of two bayagikesir numbers
return (sol.toDouble() < sag.toDouble());
}

public boolean kucuktur_esittir(bayagikesir sol,bayagikesir sag)
{
// less then and esittir karşılaştırması of two bayagikesir numbers
return (sol.toDouble() <= sag.toDouble());
}

public boolean buyuktur(bayagikesir sol,bayagikesir sag)
{
// buyuktur then karşılaştırması of two bayagikesir numbers
return sol.toDouble()>sag.toDouble();
}

public boolean buyuktur_esittir(bayagikesir sol,bayagikesir sag)
{
// buyuktur then and esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() >= sag.toDouble();
}

public boolean esittir(bayagikesir sol,bayagikesir sag)
{
// esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() == sag.toDouble();
}

public boolean esit_degildir(bayagikesir sol,bayagikesir sag)
{
// not esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() != sag.toDouble();
}

public static String cikti(bayagikesir value)
{
String b="";
if(Math.abs(value.Payda())!=1)
{
b=b+"( "+value.Pay()+ " / "+value.Payda()+ " )";
}
else
{
b=b+value.Pay()+" ";
}
return b;
}

public String cikti()

```

```

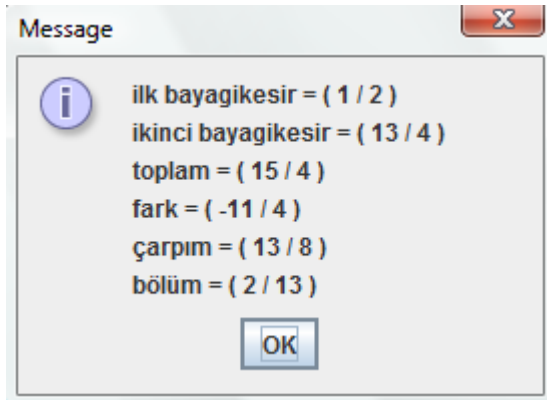
{
// yazima hazır bayagikesir formda String deęişkeni iletir.
String b="";
if(Math.abs(Payda())!=1)
{
b=b+"( "+pay+" / "+Payda()+" )";
}
else
{
b=b+Pay()+" ";
}
return b;
}
}

```

```

import javax.swing.*;
public class bayagikesirtest
{
public static void main(String arg[])
{
bayagikesir b1=new bayagikesir(1,2);
bayagikesir b2=new bayagikesir(3,1,4);
String s1="ilk bayagikesir = "+b1.cikti()+"\n";
s1+="ikinci bayagikesir = "+b2.cikti()+"\n";
s1+="toplam = "+bayagikesir.topla(b1,b2).cikti()+"\n";
s1+="fark = "+bayagikesir.cikar(b1,b2).cikti()+"\n";
s1+="arpım = "+bayagikesir.carp(b1,b2).cikti()+"\n";
s1+="bölüm = "+bayagikesir.bol(b1,b2).cikti()+"\n";
JOptionPane.showMessageDialog(null,s1);
}
}

```



5) Nokta sınıfını kullanarak dikdörtgen sınıfını yaratınız. Dikdörtgen iki noktayla tanımlanır. Dikdörtgen sınıfının kurucu metotunu, çevresini hesaplayan çevre metotunu, Alanını hesaplayan alan metotunu, dikdörtgeni tanımlayan iki noktayı String deęişkeni olarak tanımlayan cikti metotunu ve gerekli olan dięer metotları tanımla. dikdortgentesti sınıfı oluşturarak bu sınıfın içinde **cokguzelbirdikdortgen** isimli bir nesne oluştur ve bu nesnenin başlangıç ve bitiş noktalarıyla boyunu yazdır. Girdi deęerlerini program içinde sabit olarak tanımla.

6) Nokta sınıfını kullanarak doğru sınıfını yaratınız. Doğru iki noktayla tanımlanır. . Doğru sınıfının kurucu metotunu, uzunluęunu hesaplayan uzunluk metotunu, doğruyu tanımlayan iki noktayı String deęişkeni olarak tanımlayan cikti metotunu ve gerekli olan dięer metotları tanımla. dogrutesti sınıfı oluşturarak bu sınıfın içinde **cokguzelbirdikdogru** isimli bir nesne oluştur ve bu nesnenin başlangıç ve bitiş noktalarıyla boyunu yazdır. Girdi deęerlerini program içinde sabit olarak tanımla.

7. Alıştırma 4 te verilen bankahesabi1 programına para yatırma ve para çekme metodu ekleyiniz.

BÖLÜM 4 : SINIFLARI BİR ARADA KULLANMA, SINIF NESNELERİNİN DİĞER SINIFLARDA SINIF DEĞİŞKENİ OLARAK KULLANILMASI (COMPOSITION), KALITIM(INHERITANCE)

4.1 SINIF NESNELERİNİN DİĞER BİR SINIFTA KULLANILMASI (COMPOSITION)

Bir önceki bölümümüzde sınıf nesnelерinin diğеr sınıflarda sınıf değışkeni olarak kullanılmasını görmüştük. Örneğın nokta sınıfını üçgen sınıfının sınıf değışkeni olarak kullanarak üçgen sınıfını yaratmıştık. Bu örnekte üçgen sınıfı 3 tane nokta sınıfı değışkenini köşe noktaları olarak tanımlamış ve üçgeni yaratmıştık. Yeni bir örnek problem olarak yine aynı nokta temel programını kullanarak bir daire ve sonra bir silindir tanımlayacağız. Daire merkez noktası ve çapı ile, silindir bir daire (merkez noktası, çapı) ve yüksekliğı ile tanımlanacaktır. Burada noktayla ilgili matematiksel tanımlar yapmadığımız için tanımı biraz daha basit tuttuk, ama daha önceki nokta sınıfını hiç değıştirmeden kullanabilirdik.

Program 4.1.1 Nokta1 sınıfı

```
public class nokta1
{
public double xi,xj;

public nokta1()
{
xi=0;
xj=0;
}

public nokta1(double yi,double yj)
{
xi=yi;
xj=yj;
}

public nokta1(nokta y)
{
xi=y.xi;
xj=y.xj;
}

public double alan()
{return 0.0;}

public double hacim(){return 0.0;}

public String cikti()
{
String s="";
s+=xi+" i ";
if(xj>=0 )
s+=" "+xj+" j";
else if(xj<0)
s+="- "+Math.abs(xj)+" j ";
return s;
}
public String isim(){return "nokta";}
}
```

Program 4.1.2 daire sınıfı

```
public class daire
{
public double yaricap;
public nokta1 p;
```

```

public daire()
{ p=new nokta1(0.0,0.0);
  yaricapgir(0.0);}

public daire(double r, double a, double b)
{ p=new nokta1(a,b);
  yaricapgir(r);
}

public daire(double r, nokta1 n)
{ p=new nokta1(n.xi,n.xj);
  yaricapgir(r);
}

public daire(daire c)
{ p=new nokta1(c.p.xi,c.p.xj);
  yaricapgir(c.yaricap);
}

public void yaricapgir(double r)
{ if(r >= 0.0)
  yaricap=r;
  else
  yaricap=0.0;
}

public double alan()
{return Math.PI*yaricap*yaricap;}
public double hacim(){return 0.0;}
public String cikti()
{ return "Merkez = "+p.cikti()+" yarıçap = "+yaricap;}

public String isim(){return "daire";}
}

```

Program 4.1.3 silindir sınıfı

```

public class silindir
{
protected double yukseklik;
protected daire c;

public silindir()
{c=new daire(0.0,0.0,0.0);
  yukseklikgir(0);}

public silindir( double h,daire d)
{c=new daire(d);
  yukseklik=h;
}

public silindir( double h, double r, double a, double b)
{c=new daire(r,a,b);
  yukseklik=h;
}

public void yukseklikgir (double h )
{ if(h>=0) yukseklik=h;
  else yukseklik=0;
}
}

```

```

public double alan()
{return 2*c.alan()+2*Math.PI*c.yaricap*yukseklk;}

public double hacim(){return c.alan()*yukseklk;}

public String cikti()
{return c.cikti()+" yukseklk = "+yukseklk;}

public String isim(){return "silindir";}
}

```

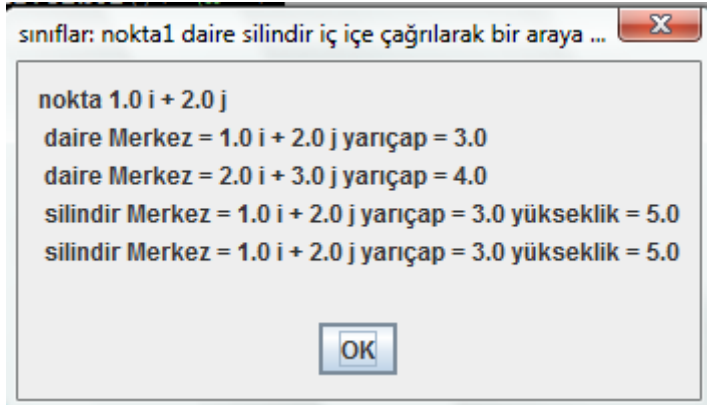
Program 4.1.3 nokta1, daire, silindir sınıflarını bir arada kullanan test programı

```

import javax.swing.*;

class nokta1dairecilindirtest1
{
    public static void main(String args[])
    {
        nokta1 n1=new nokta1(1.0,2.0);
        daire d1=new daire(3.0,n1);
        daire d2=new daire(4.0,2.0,3.0);
        silindir s1=new silindir(5.0,d1);
        silindir s2=new silindir(5.0,3.0,1.0,2.0);
        String s=n1.isim()+" "+n1.cikti()+"\n "+
        d1.isim()+" "+d1.cikti()+"\n "+
        d2.isim()+" "+d2.cikti()+"\n "+
        s1.isim()+" "+s1.cikti()+"\n "+
        s2.isim()+" "+s2.cikti()+"\n ";
        JOptionPane.showMessageDialog(null,s,
        "sınıflar: nokta1 daire silindir iç içe çağrılarak bir araya getirildi",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);}
}

```



Burada üç sınıfımız nokta1, daire ve silindir iç içe birbirinden türetilerek tanımlanmıştır. Bunu oluşturmak için her sınıfı diğer sınıfın içinde sınıf değişkeni olarak tanımladık.

Sınıfların birlikte kullanılmasına bir örnek daha verelim. İlk sınıfımız olarak bir araç tanımlama sınıfı yaratalım.

Program 4.1.4 arac sınıfı

```

import javax.swing.*;

public class arac
{
    //geçerli araç tipleri bus,truck,minitruck,minivan,auto
    public String aractipi;
}

```

```

public String brandName;
public double power; //KW olarak
public String renk;
public double agirlik; //yüksüz net ağırlık
//kurucu metodlar

public arac()
{aractipi="tanımlanmadı";
brandName="tanımlanmadı";
power=0;
renk="tanımlanmadı";
agirlik=0;
}

public arac(String i_aractipi,String i_brandName,double i_motor,String i_renk,double i_agirlik)
{//arac tipleri "otobüs","kamyon","kamyonet","minibüs","otomobil","tır","motosiklet","bisiklet","yol dışı araç"
//if not define as "tanımlanmadı"
if(i_aractipi.equals("kamyon") ||
i_aractipi.equals("kamyonet") ||
i_aractipi.equals("otobüs")||
i_aractipi.equals("minibüs")||
i_aractipi.equals("otomobil") ||
i_aractipi.equals("motosiklet") ||
i_aractipi.equals("bisiklet") ||
i_aractipi.equals("tır") ||
i_aractipi.equals("yol dışı araç")
)
{aractipi=i_aractipi;}
else
{aractipi="tanımlanmadı";}
brandName=i_brandName;
power=i_motor;
renk=i_renk;
agirlik=i_agirlik;
}

public arac(arac A)
{//gerekli tanımları yap
aractipi=A.aractipi;
brandName=A.brandName;
power=A.power;
renk=A.renk;
agirlik=A.agirlik;
}

public void define_arac(String i_aractipi,String i_brandName,double i_motor,String i_renk,double i_agirlik)
{//arac type "otobüs","kamyon","kamyonet","minibüs","otomobil","tır"
//if not define as "tanımlanmadı"
if(i_aractipi.equals("kamyon") ||
i_aractipi.equals("kamyonet") ||
i_aractipi.equals("otobüs")||
i_aractipi.equals("minibüs")||
i_aractipi.equals("otomobil") ||
i_aractipi.equals("motosiklet") ||
i_aractipi.equals("bisiklet") ||
i_aractipi.equals("tır") ||
i_aractipi.equals("yol dışı araç")
)
{aractipi=i_aractipi;}
else

```

```

{aractipi="tanımlanmadı";}
brandName=i_brandName;
power=i_motor;
renk=i_renk;
agirlik=i_agirlik;
}

public String cikti()
{String s1="";
s1+="aracın tipi : "+aractipi+"\n";
s1+="Aracın markası : "+brandName+"\n";
s1+="Motor gücü : "+power+" KW\n";
s1+="renk : "+renk+"\n";
s1+="ağırlık : "+agirlik+" kg\n";
return s1;//gerekli tanımları yap
}

public void ciktipenceresi()
{
String s1="arac sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,cikti(),s1, JOptionPane.PLAIN_MESSAGE);
}
}

```

Bu sınıfı değişik araçları tanımlamak için kullanabiliriz.

Program 4.1.5 aractest1 sınıfı

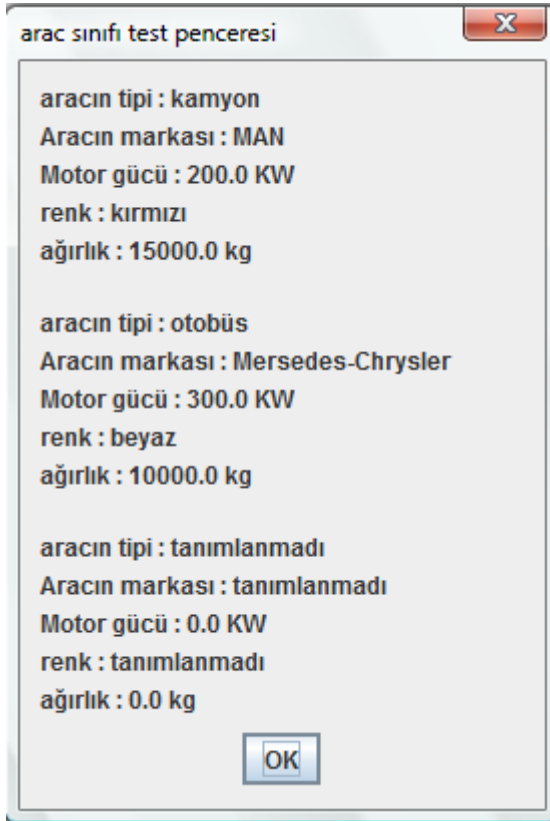
```

import javax.swing.*;
class aractest1
{

public static void ciktipenceresi(String s)
{
String s1="arac sınıfı test penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
arac k1=new arac("kamyon","MAN",200.0,"kırmızı",15000);
arac o1=new arac("otobüs","Mercedes-Chrysler",300.0,"beyaz",10000);
arac a=new arac();
String s=k1.cikti()+"\n"+o1.cikti()+"\n"+a.cikti();
ciktipenceresi(s);
}
}

```



Araç sınıfındaki değişik araç tipleri alt sınıf değişkenleri oluşturularak tanımlanabilir.

Program 4.1.6 minibüs sınıfı sınıfı

```
public class minibüs
{
    arac v;

    public minibüs(String i_marka,double i_motor,String i_renk,double i_agirlik)
    {v=new arac("minibüs",i_marka,i_motor,i_renk,i_agirlik);}

    public minibüs()
    { v=new arac();
      v.aractipi="minibüs";
    }
    public minibüs(minibüs A)
    {v=new arac(A.v);
      v.aractipi="minibüs";
    }
}
```

Program 4.1.7 otobüs sınıfı sınıfı

```
public class otobüs
{
    arac v;

    public otobüs(String i_marka,double i_motor,String i_renk,double i_agirlik)
    {v=new arac("otobüs",i_marka,i_motor,i_renk,i_agirlik);}

    public otobüs()
    { v=new arac();
```

```
v.aractipi="otobüs";
}
public otobus(otobus A)
{ v=new arac(A.v);
  v.aractipi="otobüs";
}
}
```

Program 4.1.8 otobus sınıfı sınıfı

```
public class kamyon
{
  arac v;

  public kamyon(String i_marka,double i_motor,String i_renk,double i_agirlik)
  { v=new arac("kamyon",i_marka,i_motor,i_renk,i_agirlik);}

  public kamyon()
  { v=new arac();
    v.aractipi="kamyon";
  }
  public kamyon(kamyon A)
  { v=new arac(A.v);
    v.aractipi="kamyon";
  }
}
```

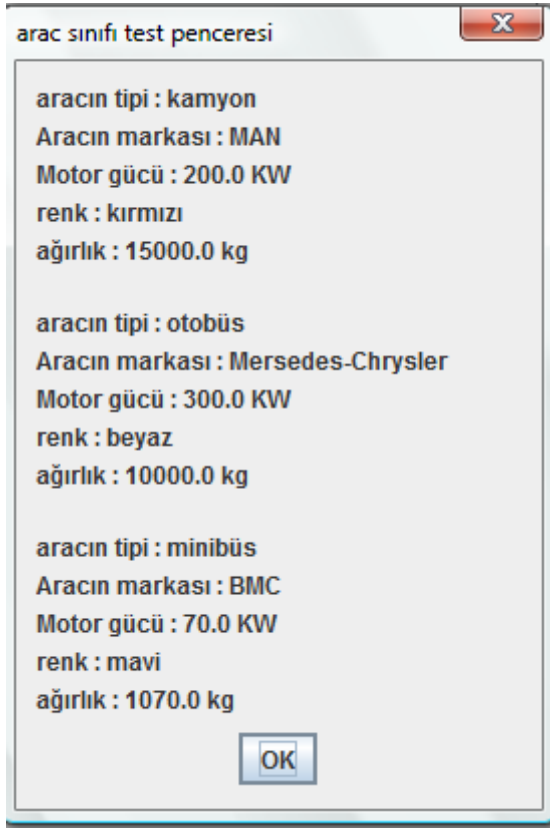
Araç sınıfından türettiğimiz bu sınıf bir test programında kullanacak olursak

Program 4.1.9 aractesti2 sınıfı

```
import javax.swing.*;
class aractesti2
{

  public static void ciktipenceresi(String s)
  {
    String s1="arac sınıfı test penceresi";
    JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
  }

  public static void main(String args[])
  {
    kamyon k1=new kamyon("MAN",200.0,"kırmızı",15000);
    otobus o1=new otobus("Mercedes-Chrysler",300.0,"beyaz",10000);
    minibus mb1=new minibus("BMC",70,"mavi",1070);
    String s=k1.v.cikti()+"\n"+o1.v.cikti()+"\n"+mb1.v.cikti();
    ciktipenceresi(s);
  }
}
```



Görüldüğü gibi alt sınıfları üst sınıfları sınıf değişkeni olarak tanımlama ile oluşturabiliriz.

4.2 SINIF NESNELERİNİN BİRBİRİNDEN KALITIM YOLUYLA TÜRETİLMESİ (INHERITANCE)

Mevcut sınıflardan yeni sınıflar üretmenin tek yolu sınıf değişkeni olarak ekleyeceğimiz sınıftan nesne kullanmak değildir. Diğer önemli bir metod da sınıfları birbirinden türetmektir. Bir sınıfı diğer sınıftan türetmek için **extends** kelimesini kullanıyoruz. Örneğin `public class nokta2 extends nokta1` gibi `extends` sözcüğü sınıf içindeki tüm metod ve sınıf değişkenlerini yeni metoda kopyeler.

Program 4.2.1 nokta1 sınıfından extends (kalıtım) yoluyla türetilmiş nokta2 sınıfı

```
public class nokta2 extends nokta1
{
    public nokta2()
    {
        super();
    }

    public nokta2(double yi,double yj)
    {
        super(yi,yj);
    }

    public nokta2(nokta y)
    {
        super(y);
    }
}
```

Program 4.2.2 nokta2 sınıfı test programı nokte2test

```
import javax.swing.*;

class nokta2test
{
    public static void main(String args[])
    {
        nokta2 n1=new nokta2(1.0,2.0);
    }
}
```



```
String s=n1.isim()+" "+n1.cikti()+"\n ";
OptionPane.showMessageDialog(null,s,
    "nokta1 den extends yoluyla türetilmiş nokta2 sınıfı testi",OptionPane.PLAIN_MESSAGE);
System.exit(0);
}
```

Programdan da görüldüğü gibi nokta2 de direk olarak yazılmamış cikti() metodu nokta1 sınıfından geldiği için direk olarak kullanılabilmiştir. Kurucu metodlar tekrar yazılmış, ancak tam tanım yerine sadece süper metodunun kullanımıyla yetinilmiştir. Süper metodu extends yoluyla türetilen metoddur (burada nokta1 metodu)

Şimdi de daha önce sınıf değişkenleri (nesne) yoluyla aktarım yaparak oluşturduğumuz daire ve silindir sınıflarının extends ile oluşturulmuş benzerleri daire1 ve silindir1 sınıflarını inceleyelim.

Program 4.2.3 nokta1 sınıfından extends ile türetilen daire1 sınıfı

```
public class daire1 extends nokta1
{
    public double yaricap;
    // bu sınıfta xi ve xj olmak üzere
    // iki tane daha sınıf değişkeni var
    public daire1()
    { super(0.0,0.0);
      yaricapgir(0.0);}

    public daire1(double r, double a, double b)
    { super(a,b);
      yaricapgir(r);
    }
    public daire1(daire1 d)
    { super(d.xi,d.xj);
      yaricapgir(d.yaricap);
    }
    public daire1(double r, nokta1 n)
    { super(n);
      yaricapgir(r);
    }

    public void yaricapgir(double r)
    { if(r >= 0.0)
      yaricap=r;
      else
      yaricap=0.0;
    }

    public double alan()
    {return Math.PI*yaricap*yaricap;}
    public double hacim(){return 0.0;}
    public String cikti()
    { return "Merkez = "+super.cikti()+" yarıçap = "+yaricap;}

    public String isim(){return "daire";}
}
```

Program 4.2.4 daire1 sınıfından extends ile türetilen silindir1 sınıfı

```
public class silindir1 extends daire1
{
    protected double yukseklik;

    public silindir1()
    {super(0.0,0.0,0.0);}
```

```

yuksekligir(0);}

public silindir1( double h,daire1 d)
{super(d);
yukseklig=h;
}

public silindir1( double h, double r, double a, double b)
{super(r,a,b);
yukseklig=h;
}

public void yuksekligir (double h )
{ if(h>=0) yukseklig=h;
else yukseklig=0;
}

public double alan()
{return 2*super.alan()+2*Math.PI*yaricap*yukseklig;}

public double hacim(){return super.alan()*yukseklig;}

public String cikti()
{return super.cikti()+" yükseklik = "+yukseklig;}

public String isim(){return "silindir";}
}

```

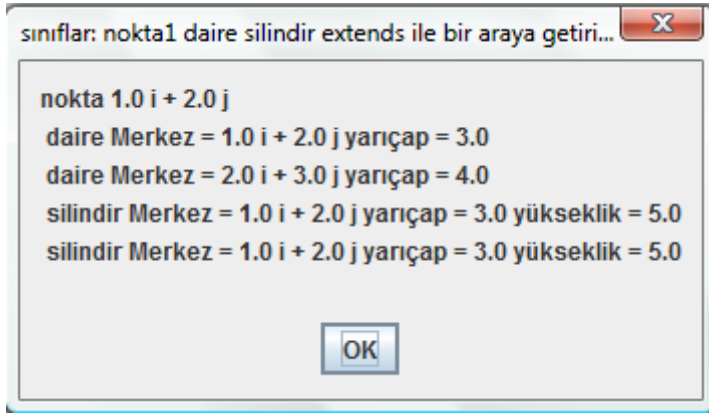
Program 4.2.5 nokta1,daire1 ve silindir1 sınıfları test programı

```

import javax.swing.*;

class nokta1daire1silindir1test2
{
public static void main(String args[])
{
nokta1 n1=new nokta1(1.0,2.0);
daire1 d1=new daire1(3.0,n1);
daire1 d2=new daire1(4.0,2.0,3.0);
silindir1 s1=new silindir1(5.0,d1);
silindir1 s2=new silindir1(5.0,3.0,1.0,2.0);
String s=n1.isim()+" "+n1.cikti()+"\n "+
d1.isim()+" "+d1.cikti()+"\n "+
d2.isim()+" "+d2.cikti()+"\n "+
s1.isim()+" "+s1.cikti()+"\n "+
s2.isim()+" "+s2.cikti()+"\n ";
JOptionPane.showMessageDialog(null,s,
"sınıflar: nokta1 daire silindir extends ile bir araya getirildi",JOptionPane.PLAIN_MESSAGE);
System.exit(0);}
}

```



Temel sonuç bir önceki nokta1-daire-silindir sınıfları uygulaması ile aynıdır. Extends yoluyla elde edilen sınıflarda nesne yoluyla elde edilen metodlara göre bir yeni özellik daha mevcuttur, bu da indirek referans kullanabilme özelliğidir. Extends yoluyla bağlanan sınıflardaki değişkenler birbirine eşitlenerek indirek referans üzerinden kullanılabilirler.

Program 4.2.6 nokta1,daire1 ve silindir1 sınıfları test programı (indirek referans kullanımı ile)

```
import javax.swing.*;

class nokta1 daire1 silindir1 test3
{
    public static void main(String args[])
    {

        nokta1 n1=new nokta1(1.0,2.0);
        daire1 d1=new daire1(3.0,n1);
        daire1 d2=new daire1(4.0,2.0,3.0);
        silindir1 s1=new silindir1(5.0,d1);
        silindir1 s2=new silindir1(5.0,3.0,1.0,2.0);
        nokta1 n2,n3,n4,n5;
        n2=d1;
        n3=d2;
        n4=s1;
        n5=s2;
        String s=n1.isim()+" "+n1.cikti()+"\n "+
        n2.isim()+" "+n2.cikti()+"\n "+
        n3.isim()+" "+n3.cikti()+"\n "+
        n4.isim()+" "+n4.cikti()+"\n "+
        n5.isim()+" "+n5.cikti()+"\n ";
        JOptionPane.showMessageDialog(null,s,
        "indirek referans kullanımı",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Sonuç bir önceki programla aynıdır. Bu programda daire1 ve silindir1 sınıfı nesnelere nokta1 sınıfına yüklenerek (indirek referans üzerinden) kullanılmıştır. Bu özellik bize daha yazılmamış sınıflar üzerinden programlama olanağı sağlar.

Bir önceki bölümdeki ikinci örneğimizde araç sınıfından kamyon, kamyonet ve minibüs sınıflarını türetmiştik. Şimdi benzer sınıfları extends yoluyla türetilim.

Program 4.2.7 extends yoluyla araç sınıfından türetilmiş kamyon1 sınıfı

```
public class kamyon1 extends arac
{

    public kamyon1(String i_marka,double i_motor,String i_renk,double i_agirlik)
    {super("kamyon",i_marka,i_motor,i_renk,i_agirlik);}

    public kamyon1()
```

```

{ super();
  aractipi="kamyon";
}
public kamyon1(kamyon1 A)
{ super(A);
}
}

```

Program 4.2.8 extends yoluyla araç sınıfından türetilmiş otobus1 sınıfı

```

public class otobus1 extends arac
{

public otobus1(String i_marka,double i_motor,String i_renk,double i_agirlik)
{super("otobüs",i_marka,i_motor,i_renk,i_agirlik);}

public otobus1()
{ super();
  aractipi="otobüs";
}
public otobus1(otobus1 A)
{super(A);
}
}

```

Program 4.2.9 extends yoluyla araç sınıfından türetilmiş minibus1 sınıfı

```

public class minibus1 extends arac
{
public minibus1(String i_marka,double i_motor,String i_renk,double i_agirlik)
{super("minibüs",i_marka,i_motor,i_renk,i_agirlik);}

public minibus1()
{ super();
  aractipi="minibüs";
}
public minibus1(minibus1 A)
{super(A);
}
}

```

Program 4.2.10 extends yoluyla araç sınıfından türetilmiş sınıfların testi

```

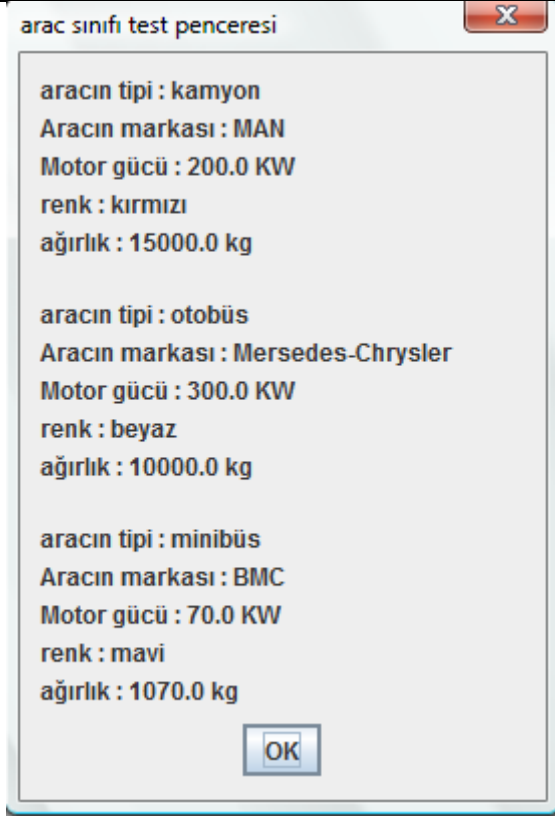
import javax.swing.*;
class aractesti3
{

public static void ciktipenceresi(String s)
{
String s1="arac sınıfı test penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
kamyon1 k1=new kamyon1("MAN",200.0,"kırmızı",15000);
otobus1 o1=new otobus1("Mercedes-Chrysler",300.0,"beyaz",10000);
minibus1 mb1=new minibus1("BMC",70,"mavi",1070);
String s=k1.cikti()+"\n"+o1.cikti()+"\n"+mb1.cikti();
ciktipenceresi(s);
}
}

```

```
}  
}
```



Burada da indirek referanslama mümkündür.

Program 4.2.10 extends yoluyla araç sınıfından türetilmiş sınıfların ve indirek referanslamamın testi

```
import javax.swing.*;  
class aractesti4  
{  
  
public static void ciktipenceresi(String s)  
{  
String s1="arac sınıfı test penceresi";  
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);  
}  
  
public static void main(String args[])  
{  
kamyon1 k1=new kamyon1("MAN",200.0,"kırmızı",15000);  
otobus1 o1=new otobus1("Mercedes-Chrysler",300.0,"beyaz",10000);  
minibus1 mb1=new minibus1("BMC",70,"mavi",1070);  
arac a1,a2,a3;  
a1=k1;  
a2=o1;  
a3=mb1;  
String s=a1.cikti()+"\n"+a1.cikti()+"\n"+a3.cikti();  
ciktipenceresi(s);  
}  
}
```

Sonuç bir önceki programın aynısıdır. Son bir örnek olarak bir kitap örneği verelim. Önce genel bir kitap sınıfı oluşturalım.

Program 4.2.11 kitap sınıfı

```
public class kitap
{
String isim;
String yazar;
String kitap_turu;
String dil;

public kitap(String kisim,String kyazar,String kkitap_turu,String kdil)
{isim=kisim;
yazar=kyazar;
kitap_turu=kkitap_turu;
dil=kdil;
}

public kitap(kitap k)
{isim=k.isim;
yazar=k.yazar;
kitap_turu=k.kitap_turu;
dil=k.dil;
}

public String cikti()
{
String s="isim = "+isim+"\n";
s+="yazar = "+yazar+"\n";
s+="kitap türü = "+kitap_turu+"\n";
s+="kitap dili = "+dil+"\n";
return s;
}}
```

Program 4.2.12roman sınıfı

```
public class roman extends kitap
{
public roman(String kisim,String kyazar,String klanguage)
{ super(kisim,kyazar,"roman",klanguage);}
}
```

Program 4.2.13 siir sınıfı

```
public class siir extends kitap
{
public siir(String kisim,String kyazar,String klanguage)
{ super(kisim,kyazar,"siir",klanguage);}
}
```

Program 4.2.14 ansiklopedi sınıfı

```
public class ansiklopedi extends kitap
{
public ansiklopedi(String kisim,String kyazar,String klanguage)
{ super(kisim,kyazar,"ansiklopedi",klanguage);}
}
```

Program 4.2.15 kitaptest1 sınıfı

```
import javax.swing.*;
class kitaptest1
{
public static void ciktipenceresi(String s)
```

```

{
String s1="kitap sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
kitap dergi=new kitap("Bilim ve Teknik","TÜBİTAK","dergi","Turkish");
roman r1=new roman("It is all quiet in the westen front ","Erich Maria Reamarque","İngilizce");
kitap r2=new kitap("The God Delusion","Richard Dawkins","roman","İngilizce");
siir s1=new siir("Human lanscapes from my country","Nazım Hikmet","İngilizce ve Türkçe");
ansiklopedi a1=new ansiklopedi("Encyclopedia Britannica","İngilizce");
String s=dergi.cikti()+"\n"+r1.cikti()+"\n"+r2.cikti()+"\n"+s1.cikti()+"\n"+a1.cikti();
ciktipenceresi(s);
}
}

```



Buradan da görülebileceği gibi genel kitap türünü veya ondan türetilmiş sınıfları kitap bilgisine erişmek için kullanabiliyoruz. Bura da da indirek referanslama yapılabilir.

```

import javax.swing.*;
class kitaptest2
{
public static void ciktipenceresi(String s)
{

```

```

String s1="kitap sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
kitap dergi=new kitap("Bilim ve Teknik","TÜBİTAK","dergi","Turkish");
roman r1=new roman("It is all quiet in the westen front ","Erich Maria Reamarque","İngilizce");
kitap r2=new kitap("The God Delusion","Richard Dawkins","roman","İngilizce");
siir s1=new siir("Human lanscapes from my country","Nazım Hikmet","İngilizce ve Türkçe");
ansiklopedi a1=new ansiklopedi("Encyclopedia Britannica","İngilizce");
kitap k1,k2,k3,k4,k5;
k1=dergi;
k2=r1;
k3=r2;
k4=s1;
k5=a1;

String s=k1.cikti()+"\n"+k2.cikti()+"\n"+k3.cikti()+"\n"+k4.cikti()+"\n"+k5.cikti();
ciktipenceresi(s);
}
}

```

4.3 ALIŞTIRMALAR

1. Kutu1 sınıfı verilmiş, bu sınıfı kullanarak demiryolu vagonu sınıfı yazılacaktır. Bu sınıfı kutu sınıfını sınıf değişkeni olarak kullanarak yaratınız. Ve test ediniz.

```

public class kutu1
{
public String isim;
public double en,boy,yukseklk;
public String renk;
public double dara;
public double netagirlik;
public double toplamagirlik;
public kutu1(String isimi,double eni,double boyi,double yuksekliki,String renki,double
darai,double netagirliki)
{isim=isimi;
en=en;
boy=boyi;
yukseklk=yukseklki;
renk=renki;
dara=darai;
netagirlik=netagirliki;
toplamagirlik=dara+netagirlik;
}
public double hacim()
{ return en*boy*yukseklk;}
}

```

2. Kutu1 sınıfı verilmiş, bu sınıfı kullanarak demiryolu vagonu sınıfı yazılacaktır. Bu sınıfı kutu sınıfını kalıtım yoluyla (extends) kullanarak yaratınız ve test ediniz.
3. nokta1 sınıfından yola çıkarak sınıf değişkeni tanımlama yoluyla tanımlanan daire sınıfından yine sınıf değişkeni tanımlama yoluyla koni sınıfını tanımlayınız ve test ediniz.
4. nokta1 sınıfından yola çıkarak kalıtım (extends) yoluyla tanımlanan daire1 sınıfından yine sınıf değişkeni tanımlama yoluyla koni1 sınıfını tanımlayınız ve test ediniz.
5. Üç boyutlu uzayda iki tane nokta3D noktası ve yarıçap tanımlayarak 3 boyutlu uzayda bir silindir tanımlayınız.
6. Uzayda dört tane nokta3D sınıfı nokta bir üçgen prizma tanımlar. Bu prizmayı tanımlayanve hacmini hesaplayan bir sınıfı sınıf değişkeni kullanarak tanımlayınız ve hacmini hesaplayınız. Bir test programı yazarak test ediniz.

Not: üçgen prizmanın hacmi:

$$V = \frac{1}{6} \det \begin{vmatrix} 1 & x_{i1} & x_{j1} & x_{k1} \\ 1 & x_{i2} & x_{j2} & x_{k2} \\ 1 & x_{i3} & x_{j3} & x_{k3} \\ 1 & x_{i4} & x_{j4} & x_{k4} \end{vmatrix}$$

Det determinant anlamındadır.

7. Uzayda dört tane nokta3D sınıfı nokta bir üçgen prizma tanımlar. Bu prizmayı tanımlayanve hacmini hesaplayan bir sınıfı kalıtım (extends) kullanarak tanımlayınız ve hacmini hesaplayınız. Bir test programı yazarak test ediniz.

5 BOYUTLU DEĞİŞKENLER

5.1 BOYUTLU DEĞİŞKENLERİN GENEL TANIMI

Boyutlu değişkenler birden fazla aynı tür değişkeni (veya nesneyi) bir gurup olarak bir arada bulunduran değişken türleridir. Gruptaki her değişkene bir indeks değişkeni üzerinden ulaşılabilir.

Genel olarak boyutlu değişken java ortamında şu şekilde tanımlanır:

Değişken_türü değişken_adı[];

veya

Değişken_türü[] değişken_adı;

Tanımlamaya örnek verecek olursak:

int ayin_gunleri[]; veya

int[] ayin_gunleri;

Bu tanım sadece değişkenin boyutlu değişken olduğunu gösterir, toplam boyutunun ne olduğunu içermez. Boyut verisini tanımlamak için :

Değişken_adi=new Değişken_türü[değişken boyutu];

Tanımlamayı yapmamız gerekir. Örneğin :

ayin_gunleri =new int[7];

Bu tanımdan sonra alt değişkenlere indeks üzerinden ulaşabiliriz.

ayin_gunleri [0]=31;

ayin_gunleri [1]=28;

Burada indeks değişkenini daima 0 dan başladığını özellikle not edelim. Bu kavramı küçük bir program örneğinin içinde sunalım.

Program 5.1.1 boyutlu değişken örneği boyut1a.java

```
import java.io.*;
class boyut1a
{
public static void main(String args[] )
{
int ayin_gunleri[];
ayin_gunleri=new int[12];
ayin_gunleri[0]=31;
ayin_gunleri[1]=28;
ayin_gunleri[2]=31;
ayin_gunleri[3]=30;
ayin_gunleri[4]=31;
ayin_gunleri[5]=30;
ayin_gunleri[6]=31;
ayin_gunleri[7]=31;
ayin_gunleri[8]=30;
ayin_gunleri[9]=31;
ayin_gunleri[10]=30;
ayin_gunleri[11]=31;
System.out.println("Nisan ayı "+ayin_gunleri[3]+" gün çeker");
}
}
```

Program çıktısı

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" boyut1a
Nisan ayı 30 gün çeker
```

> Terminated with exit code 0.
> Terminated with exit code 0.

Boyutlu deęişkeni ilk defa tanımlarken boyut ve deęerler bir arada büyük parantez {} kullanılarak verilebilir.

Deęişken_türü deęişken_adı[]={deęer_0,deęer_1,...,deęer_n};

Burada deęişken boyutunun ayrıca tanımlanmadığına dikkati çekelim. Program büyük parantez içindeki virgülle ayrılmış deęişkenleri sayarak boyutlu deęişkenin boyutunu kendisi tayin eder.

Bir örnek verecek olursak

int ayin_gunleri[]={31,28,31,30,31,30,31,31,30,31,30,31};

üstteki programın bu şekilde getirilerek verilmiş eşdeęeri alttadır. :

Program 5.1.2 boyutlu deęişken örneęi boyut1b.java

```
import java.io.*;
class boyut1b
{
public static void main(String args[] )
{
int ayin_gunleri[]={ 31,28,31,30,31,30,31,31,30,31,30,31 };
System.out.println("Nisan ayı "+ayin_gunleri[3]+" gün çeker");
}
}
```

Programın çıktısı bir önceki programla aynı olacaktır. Boyutlu deęişkenlere indeksler üzerinden ulaştığımızdan döngü yalarıyla birlikte kullanılmak için ideal bir form oluştururlar. Bir sonraki program verilen sayıların ortalamasını hesaplamaktadır.

5.1.3 mat1.java for döngüsü kullanarak ortalama hesaplayan bir örnek

```
class mat1
{
public static void main(String args[])
{
int sayi[]={ 10,5,7,9,11,13,14,18 };
double toplam=0;
int i;
for(i=0;i<sayi.length;i++)
{
toplam+=sayi[i];
}
System.out.println("Ortalama = "+toplam/sayi.length);
}
}
```

Çıktı şu şekilde görünecektir :

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" mat1
Ortalama = 10.875
> Terminated with exit code 0.
```

Elbette aynı sonuca for döngüsü yerine while döngüsü kullanarak ta ulaşabilirdik.

5.1.4 math2.java while döngüsü kullanarak ortalama hesaplayan bir örnek

```
class mat2
{
public static void main(String args[])
{
int sayi[]={ 10,5,7,9,11,13,14,18};
double toplam=0;
int i=0;
while(i<sayi.length)
{
toplam+=sayi[i];
i++;
}
System.out.println("Ortalama = "+toplam/sayi.length);
}
}
```

Sonuçlar bir önceki programla aynıdır. Şimdi de iki boyutlu iki değişkeni toplayalım:

5.1.5 math3.java iki iki boyutlu değişkeni toplayan bir örnek

```
class mat3
{
public static void main(String args[])
{
int sayi1[]={ 10,5,7,9,11,13,14,18};
int sayi2[]={ 11,3,8,5,9,12,7,20};
int sayi3[]=new int[sayi1.length];
double toplam=0;
int i=0;
while(i<sayi1.length)
{
sayi3[i]=sayi1[i]+sayi2[i];
System.out.println("sayi["+i+"] = "+sayi3[i]);
i++;
}}}
```

Bu işlemi yaparken çıktı değişkenlerinin daha düzenli olması için şekil verme işlemi olan format işlemi kullanabiliriz.

5.1.6 math4.java iki iki boyutlu değişkeni toplayan ve formatlayarak yazdıran bir örnek

```
import javax.swing.*;
import java.util.Locale;
class mat4
{
public static String output(int a[])
{
Locale tr=new Locale("TR");
int x=25;
String s1="";
for(int i=0;i<a.length;i++)
{ s1+=String.format(tr," %10d ",a[i]);}
s1+="\n";
}
```

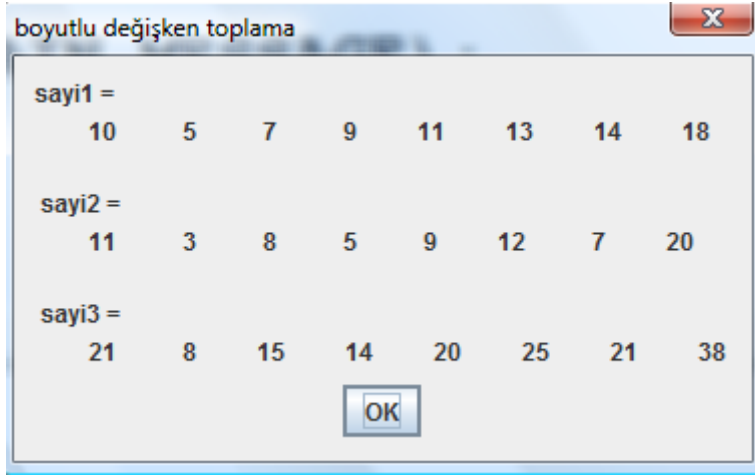
```

    return s1;
}

public static void main(String args[])
{
int sayi1[]={ 10,5,7,9,11,13,14,18};
int sayi2[]={ 11,3,8,5,9,12,7,20};
int sayi3[]=new int[sayi1.length];
double toplam=0;
int i=0;

while(i<sayi1.length)
{
sayi3[i]=sayi1[i]+sayi2[i];
i++;
}
String s1="sayi1 =\n"+output(sayi1)+"\n sayi2 =\n"+output(sayi2)+"\n sayi3 =\n"+output(sayi3);
String s="boyutlu deęişken toplama";
JOptionPane.showMessageDialog(null,s1,s,JOptionPane.PLAIN_MESSAGE);
}
}

```



Tablo 8.1.1 Format yapıları

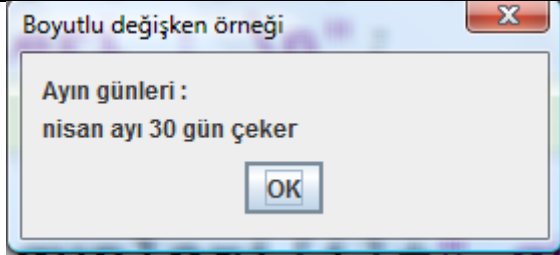
'b' veya 'B' genel
'h' veya 'H' genel
's' veya 'S' String
'c' veya 'C' harf deęişkeni
'd' on basamak (desimal) tam sayı deęişkeni (integer)
'o' sekiz basamak (octal) tam sayı deęişkeni (integer)
'x' 16 basamak integer (hexadecimal) tam sayı deęişkeni (integer)
'e' veya 'E' bilimsel geręek sayı (eksponent ile birlikte verilir)
'f' veya 'F' geręek sayı
'g' veya 'G' geręek sayı sayının büyüklüęüne göre f veya e gibi davranır
'a' veya 'A' geręek sayı

't' veya 'T' zaman

Şüphesiz boyutlu değişkenleri diğer değişken türleri için de aynı şekilde kullanabiliriz

5.1.7 ayingunleri.java döngü yapıları kullanarak boyutlu değişkenlere ulaşılması

```
import java.io.*;
import javax.swing.*;
class ayingunleri {
public static int ay(String s)
{
String ay[]={"ocak","şubat","mart","nisan","mayıs","haziran","temmuz",
"agustos","eylül","ekim","kasım","aralık"};
int i;
int j=-1;
for(i=0;i<12;i++)
{if(s.equals(ay[i])) {j=i;break;}}
return j;
}
public static void main(String args[] )
{ int ayin_gunleri[]={31,28,31,30,31,30,31,31,30,31,30,31};
String s="Ayın günleri : \n";
String s2="nisan";
int i=ay(s2);
s+=s2+" ayı "+ayin_gunleri[i]+" gün çeker \n";
String s1="Boyutlu değişken örneği";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}
```



5.3 GELİŞMİŞ FOR DÖNGÜ YAPISI

Java'nın son versiyonlarında gelişmiş ve kullanılması daha kolay bir for yapısı verilmiştir. Kişisel olarak ben eski yapıyı kullanmaya devam edecek olsam da bu yeni yapıyı da size öğretmemiz gerekiyor. Bu for yapısı özellikle boyutlu değişkenlerle birlikte kullanılma gayesiyle oluşturulmuştur.

5.3.1 ForTest.java standart for döngüsünün boyutlu değişkenlerle kullanımı

```
public class ForTest
{
public static void main( String args[] )
{
int boyut[] = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
int toplam = 0;
for ( int i=0;i<boyut.length;i++)
{toplam += boyut[i];}
}
}
```

```
System.out.println( "toplam = "+toplam );
} // main metodunun sonu
}
```

Gelişmiş for yapısını kullanarak aynı döngüyü kurabiliriz.

5.3.2 ForTest1.java gelişmiş for döngüsünün boyutlu değişkenlerle kullanımı

```
public class ForTest1
{
public static void main( String args[] )
{
int boyut[] = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
int toplam = 0;
for ( int sayi:boyut)
{toplam += sayi;}

System.out.println( "toplam = "+toplam );
} // end main
}
```

Örnekte görüldüğü gibi gelişmiş for döngüsü

```
for ( int sayi:boyut)
{toplam += sayi;}
```

Standart döngü ise

```
for ( int i=0;i<boyut.length;i++)
{toplam += boyut[i];}
```

Formundaydı. Bu yeni yapıyı kullanmanın elbetteki artı ve eksileri mevcuttur. Standart yapıyı tüm programcılar rahatlıkla anlayabilirken, gelişmiş for yapısını anlayamayabilirler. Bilgisayar programlama dilleri de insan dilleri gibi konuşucuların dili anlamasını gerektirir.

5.4 NESNELERİN BOYUTLU YAPILARI

Boyut kavramı basit değişkenlere uygulandığı gibi nesne tipi değişkenlere de uygulanabilir. Sınıf nesnelere boyutlu değişken olarak kullanılmasına örnek olarak Kutu sınıfını verelim.

5.4.1 Kutu.java sınıfı

```
public class Kutu
{
private double en;
private double boy;
private double yukseklik;

//kurucu metod kutu
Kutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklik=y;
}
```

```

//sınıf değişkenlerini değiştirme set metodu
public void setKutu(double e,double b,double y)
{
en=e;
boy=b;
yukseklık=y;
}

//sınıf değişkenlerine ulaşma (get) metotları
public double getEn()
{return en;}

public double getBoy()
{return boy;}

public double getTukseklık()
{return yukseklık;}

//bu metot kutunun hacmini hesaplar (işlem- hesap) metodu
public double hacim()
{
return en*boy*yukseklık;
}

//string çıktı metodu
public String toString()
{
return("en = "+en+" boy = "+boy+" yukseklık = "+yukseklık+"\n"+
"Hacim = "+hacim()+"\n");
}
}

```

5.4.2 Kutu sınıfını boyutlu değişken olarak kullanan Kututest1 sınıfı

```

import java.io.*;
import javax.swing.*;
class Kututest1
{
public static void main(String args[] )
{
Kutu x[]=new Kutu[4];
x[0]=new Kutu(1.1,2.2,3.3);
x[1]=new Kutu(1.23,2.11,3.12);
x[2]=new Kutu(2.5,2.11,3.0);
x[3]=new Kutu(2.5,2.11,3.1);
String s1="boyutlu değişken örneği ";

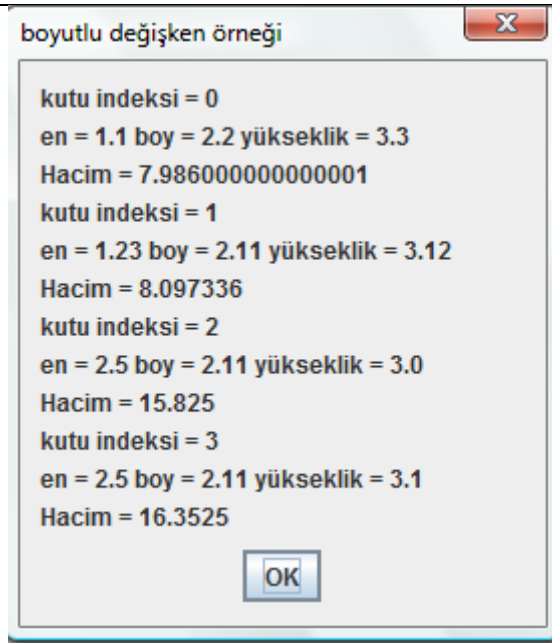
```



```

String s="";
for(int i=0;i<x.length;i++)
    {s+="kutu indeksi = "+i+"\n"+x[i].toString();}
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}

```



Nesne türü boyutlu deęişkenlerin basit boyutlu deęişkenlerden temel farkı nesne tipi deęişkenlerin boyutları tanımlandıktan sonra her bir alt boyutun kurucu metodlar kullanılarak tanımlanması gerekmesidir. İkinci örneęimizde boyutlu gerçek sayı deęişkeni vektör sınıfının tanımında kullanacaęız.

5.4.3 vektör sınıfı

```

import java.util.Locale;
public class vektor
{
    public double V[];
    //kurucu metodlar
    public vektor(double Vi[])
    { V=new double[Vi.length];
      input_vektor(Vi);
    }

    public vektor(vektor Vi)
    { V=new double[Vi.V.length];
      input_vektor(Vi.V);
    }

    public vektor(int n)
    { V=new double[n];
      for(int i=0;i<n;i++)
          { V[i]=0.0;}
    }

    public void input_vektor(double Vi[])

```

```

{ for(int i=0;i<V.length;i++)
    {V[i]=Vi[i];}
}

public static vektor toplava(vektor V1,vektor V2)
{
//
int n=V1.V.length;
vektor V3=new vektor(n);
for(int i=0;i<n;i++)
    {V3.V[i]=V1.V[i]+V2.V[i];}
return V3;
}

public static vektor toplava(double[] V1,double[] V2)
{
//
int n=V1.length;
vektor M=new vektor(n);
for(int i=0;i<n;i++)
    {M.V[i]=V1[i]+V2[i];}
return M;
}

public void toplava(vektor V1)
{
//
int n=V.length;
for(int i=0;i<n;i++)
    {V[i]+=V1.V[i];}
}

public void toplava(double [] B)
{
//
int n=V.length;
for(int i=0;i<n;i++)
    {V[i]+=B[i];}
}

public static String cikti(vektor M)
{ int n=M.V.length;
  String s="";
  for(int i=0;i<n;i++)
    {s+=M.V[i]+" ";}
s+="\n";
return s;
}

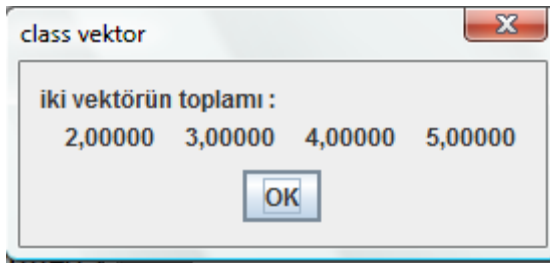
public String cikti()
{ Locale tr=new Locale("TR");
  int n=V.length;
  String s1="";
  for(int i=0;i<n;i++)
    { s1+=String.format(tr, " %10.5f ",V[i]);}
  s1+="\n";
return s1;
} //

```

```
}//
```

5.4.4 vektor sınıfını kullanan test programı vektortest

```
import javax.swing.*;
class vektortest {
public static void main(String args[] )
{ double A[]={ 1.0,2.0,3.0,4.0};
  double B[]={ 1.0,1.0,1.0,1.0};
  vektor M1=new vektor(A);
  vektor M2=new vektor(B);
  M1.topla(M2);
  String s="iki vektörün toplamı : \n"+M1.cikti();
  String s1="class vektor";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
} }
```



Bu arada çıktının Türkçe gerçek değişken ayraçları olan virgülle ayrıldığını not edelim, çünkü double değişkenler vektör sınıfında Türkçe formatında tanımlandı. Üçüncü örnek olarak daha önce tanımladığımız kitap sınıfını kullanarak kütüphane sınıfını yaratalım: Kütüphane doğal olarak bir çok kitabın bir arada olduğu bir yerdir

5.4.5 kutuphane sınıfı

```
public class kutuphane
{
String isim;
int kitap_sayisi;
kitap katalog[];

public kutuphane(String kismi,kitap kkatalog[])
{ isim=kismi;
  kitap_sayisi=kkatalog.length;
  katalog=new kitap[kitap_sayisi];
  for(int kitap_number=0;kitap_number<kitap_sayisi;kitap_number++)
  { katalog[kitap_number]=kkatalog[kitap_number];}
}

public String cikti()
{ String s="kutuphane ismi : "+isim+"\n";
  s+="Toplam kitap sayısı : "+kitap_sayisi+"\n\n";
  for(int kitap_number=0;kitap_number<kitap_sayisi;kitap_number++)
  { s+=katalog[kitap_number].cikti()+"\n";}
  return s;
}
}
```

5.4.6 Kutuphane sınıfını kullanan test programı kutuphanetest

```
import javax.swing.*;
class kutuphanetest1
{
public static void ciktipenceresi(String s)
{
String s1="kutuphane sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{

kitap b[]=new kitap[5];
b[0]=new kitap("Bilim ve Teknik","TÜBİTAK","magazine","Turkish");
b[1]=new roman("It is all quiet in the westen front ","Erich Maria Reamarque","İngilizce");
b[2]=new kitap("The God Delusion","Richard Dawkins","fiction","İngilizce");
b[3]=new siir("Human lanscapes from my country","Nazım Hikmet","İngilizce and Turkish");
b[4]=new ansiklopedi("Encyclopedia Britannica","İngilizce");
kutuphane l=new kutuphane("izmir milli kütüphanesi",b);
String s=l.cikti();
ciktipenceresi(s);
}}
```

Bu son örnekte kitapları tanımlarken roman siir ve ansiklopedi sınıflarını da kullandığımızı not edelim. Bunun temel sebebi extend ile türetildiklerinden kitap sınıfı üzerinden referansla i(indirek referansla) kullanılabilmesidir. Yine benzer bir örnek geçen bölümümüzde verilen araç sınıfına da uyarlanabilir.

5.4.7 vehicle_register class that use vehicle class as an array object variable

```
public class arac_kayit
{
String isim;
int arac_sayisi;
arac liste[];

public arac_kayit(String kisim,arac kliste[])
{ isim=kisim;
arac_sayisi=kliste.length;
liste=new arac[arac_sayisi];
for(int i=0;i<arac_sayisi;i++)
{liste[i]=kliste[i];}
}

public String cikti()
{ String s="araç kayıt istasyonu adı : "+isim+"\n";
s+="Toplam arac sayısı : "+arac_sayisi+"\n\n";
for(int i=0;i<arac_sayisi;i++)
{s+=liste[i].cikti()+"\n";}
return s;
}
```

}



5.4.8 An example program utilises vehicle class and vehicle_registration class

```
import javax.swing.*;
class aractest1
{
public static void ciktipenceresi(String s)
{
String s1="araç sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
arac v[]=new arac[3];
v[0]=new kamyon1("MAN",200.0,"kırmızı",15000);
v[1]=new otobus1("Mercedes-Chrysler",300.0,"beyaz",10000);
```

```
v[2]=new minibus1("BMC",70,"mavi",1070);
arac_kayit vr=new arac_kayit("izmir araç istasyonu",v);
String s=vr.cikti()+"\n";
ciktipenceresi(s);
}
}
```



5.5 NESNE VE DEĞİŞKENLERİN ÇOK BOYUTLU KULLANIMI

Çok boyutlu değişkenler birden fazla boyutlu olarak düşünülebilir. Bu durumda örneğin iki boyutlu sistemi vektörün vektörü gibi düşünebiliriz.

5.5.1 boyut2D sınıfı

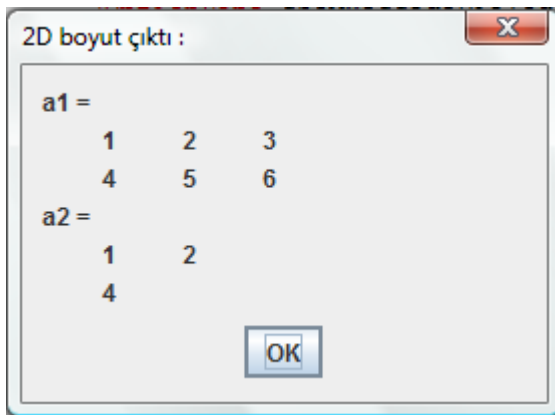
```
import javax.swing.*;
import java.util.Locale;

public class boyut2D
{
    public static void main(String[] args)
    {
        int a1[][]={{1,2,3},{4,5,6}};
        int a2[][]={{1,2},{4}};
    }
}
```

```

String s="2D boyut çıktı : ";
String s1="a1 =\n"+cikti(a1);
s1+="a2 = \n"+cikti(a2);
JOptionPane.showMessageDialog(null,s1,s,JOptionPane.PLAIN_MESSAGE);
}
public static String cikti(int a[][])
{
Locale tr=new Locale("TR");
int x=25;
String s1="";
for(int i=0;i<a.length;i++)
{ for(int j=0;j<a[i].length;j++)
{ s1+=String.format(tr," %10d ",a[i][j]);}
s1+="\n";
}
return s1;
}}

```



Aynı kavram sınıflara da uygulanabilir. İki boyutlu matrisleri içeren matris sınıfı tanımlanmıştır

5.4.2 matris sınıfı

```

import java.util.Locale;
public class matris
{
public double A[][];
//constructor methods
public matris(double Ai[][])
{ A=new double[Ai.length][Ai[0].length];
  matris_gir(Ai);
}

public matris(matris M)
{ A=new double[M.A.length][M.A[0].length];
  matris_gir(M.A);
}

public matris(int n,int m)
{ A=new double[n][m];
  for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)

```

```

        {A[i][j]=0.0;}
    }
}

public void matris_gir(double Ai[][])
{ for(int i=0;i<A.length;i++)
    { for(int j=0;j<A[0].length;j++)
        {A[i][j]=Ai[i][j];}
    }
}

public static matris toplama(matris M1,matris M2)
{
//
int n=M1.A.length;
int m=M1.A[0].length;
matris M3=new matris(n,m);
for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)
        {M3.A[i][j]=M1.A[i][j]+M2.A[i][j];}
    }
return M3;
}

public static matris toplama(double[][] A1,double[][] A2)
{
//
int n=A1.length;
int m=A1[0].length;
matris M=new matris(n,m);
for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)
        {M.A[i][j]=A1[i][j]+A2[i][j];}
    }
return M;
}

public void toplama(matris M1)
{
//
int n=A.length;
int m=A[0].length;
for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)
        {A[i][j]+=M1.A[i][j];}
    }
}

public void toplama(double [][] B)
{
//
int n=A.length;

```



```

int m=A[0].length;
for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)
      { A[i][j]+=B[i][j];}
    }
}

public static String cikti(matris M)
{ int n=M.A.length;
  int m=M.A[0].length;
  String s="";
  for(int i=0;i<n;i++)
    { for(int j=0;j<m;j++)
      { s+=M.A[i][j]+" ";}
      s+="\n";
    }
  return s;
}

public String cikti()
{ Locale tr=new Locale("TR");
  int n=A.length;
  int m=A[0].length;
  String s1="";
  for(int i=0;i<n;i++)
  { for(int j=0;j<m;j++)
    { s1+=String.format(tr," %10.5f ",A[i][j]);}
    s1+="\n";
  }
  return s1;
} //end of method cikti

} //matris sınıfının sonu

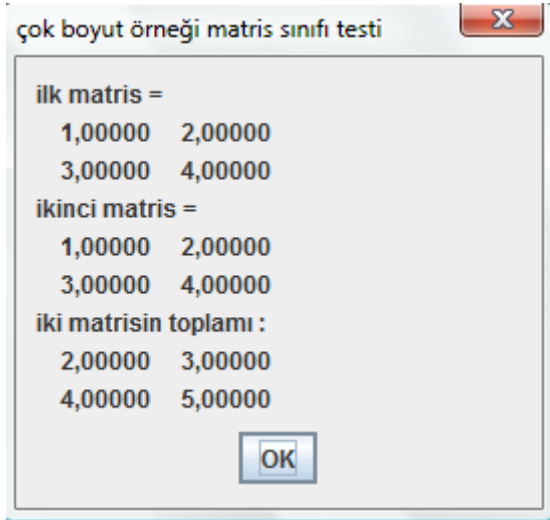
```

5.5.3 iki matrisi toplayan örnek program

```

import javax.swing.*;
class matristest {
public static void main(String args[] )
{ double A[][]={{ 1.0,2.0},{3.0,4.0}};
double B[][]={{ 1.0,1.0},{1.0,1.0}};
matris M1=new matris(A);
String s="ilk matris = \n"+M1.cikti();
matris M2=new matris(B);
s+="ikinci matris = \n"+M1.cikti();
M1.topla(M2);
s+="iki matrisin toplamı : \n"+M1.cikti();
String s1="çok boyut örneği matris sınıfı testi";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
} }

```



Bir sonraki örnek programda çok boyutlu double tip değişkenler önce matris sınıfına yüklenip sonra toplanmaktadır.

5.5.4 İki matrisi toplayan örnek

```
import javax.swing.*;
class matristest1 {
public static void main(String args[] )
{ double A[][]={{1.0,2.0},{3.0,4.0}};
double B[][]={{1.0,1.0},{1.0,1.0}};
matris M[]=new matris[3];
M[0]=new matris(A);
M[1]=new matris(B);
M[2]=matris.topla(M[0],M[1]);
String s="ilk matris = \n"+M[0].cikti();
s+="ikinci matris = \n"+M[1].cikti();
s+="iki matrisin toplamı : \n"+M[2].cikti();
String s1="çok boyut örneği matris sınıfı testi";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
} }
}
```

Sonuç bir önceki örneğin aynıdır.

5.6 BOYUTLU DEĞİŞKENLERDE BOYUT DEĞİŞTİRME

Boyutlu değişkenlerin boyutları ilk tanımlamalarında belirtildiğinden normal olarak değiştirilemez. Ancak indirek yollarla boyutun değiştirilmesi mümkündür. Bu yol önce yeni boyutta bir boyutlu değişken tanımlamak, sonra boyutlu değişkenin içerisindeki değerleri bu yeni değişkene aktarmak ve sonra orijinal boyutlu değişkenin adresini yeni oluşturulan boyutlu değişken adresiyle değiştirmek şeklinde yapılır.

```
double a[]={3.0,5.0,7.0}
double b[]=new double[4];
for(int i=0;i<a.length;i++)
{b[i]=a[i];}
a=b;
```

program parçacığında a değişkeninin boyutu üçten 4 e değiştirilmiştir. Daha detaylı bir örnekle boyut değiştirmeyi inceleyelim.

Program 5.6.1 : boyutlu değiştirme örneği : doubleBoyut.java

```
public class doubleBoyut
{
```

```
//sınıf değişkenleri
public double a[];
public int length;

public doubleBoyut(double x[] )
{
length=x.length;
a=new double[length];
for(int i=0;i<length;i++)
a[i]=x[i];
}

public doubleBoyut(int n)
{
a=new double[n];
length=a.length;
}

public void boyutEkle(int n)
{
if(n>0)
{
int z=a.length+n;
double[] b=new double[z];
for(int i=0;i<a.length;i++)
{b[i]=a[i];}
a=b;
length=a.length;
}
}

public void boyutEkle()
{
boyutEkle(1);
}

public void boyutAzalt(int n)
{
int z=a.length-n;
if(z>1)
{
double[] b=new double[a.length-n];
for(int i=0;i<z;i++)
b[i]=a[i];
a=b;
}
else
{
double[] b=new double[1];
b[0]=a[0];
a=b;
}
}
```

```

    length=a.length;
}

public void boyutAzalt()
{
    boyutAzalt(1);
}

public double getValue(int i)
{
    return a[i];
}

public double[] getValue()
{
    return a;
}

public void setValue(double x,int i)
{
    a[i]=x;
}

public void setValue(double[] x )
{
    length=x.length;
    a=new double[length];
    for(int i=0;i<length;i++)
    {
        a[i]=x[i];
    }
}

public String cikti(int i)
{
    return ""+a[i];
}

public String cikti()
{
    String s="";
    for(int i=0;i<length;i++)
    {
        s+=a[i]+" ";
    }
    s+="\n";
    return s;
}
}

```

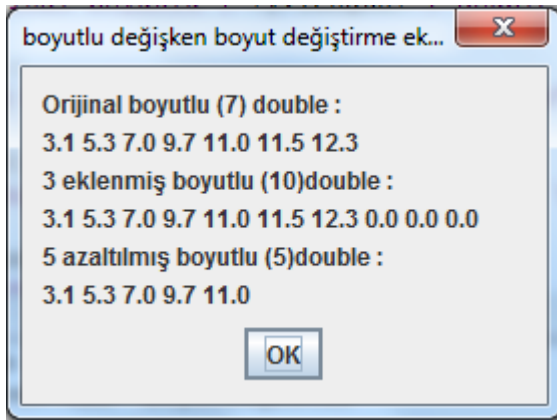
DoubleBoyut sınıfında boyutu değiştirilebilen değişken (nesne) tanımladık. Bu değişkeni boyutDegistir sınıfında test edelim :

Program 5.6.2 : boyutlu deęiřtirme testi : doubleBoyut sınıfını kullanan boyutDegistir.java

```
import javax.swing.JOptionPane;

public class boyutDegistir
{
    public static void main(String args[])
    {
        double x[]={ 3.1,5.3,7.0,9.7,11.0,11.5,12.3};
        doubleBoyut y=new doubleBoyut(x);
        String s="Orijinal boyutlu (" +y.length+" ) double : \n";
        s+=y.cikti();
        y.boyutEkle(3);
        s+="3 eklenmiř boyutlu (" +y.length+" )double : \n";
        s+=y.cikti();
        y.boyutAzalt(5);
        s+="5 azaltılmıř boyutlu (" +y.length+" )double : \n";
        s+=y.cikti();
        JOptionPane.showMessageDialog(null,s,
        "boyutlu deęiřken boyut deęiřtirme eksersizi",
        JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Program ıktısı :



5.6 ALIřTIRMALAR

1 : double tr bir boyutlu deęiřken oluřturarak verileri toplama

```
public class array1
{
    public static void main(String arg[])
    {
        double x[]=new double[3];
        x[0]=1.2;
        x[1]=2.3;
        x[2]=3.1;
        double total=0;
        for(int i=0;i<x.length;i++)
```

```
{total+=x[i];
System.out.println("x["+i+"] = "+x[i]);
}
System.out.println("total="+total);
}
}
```

----- Capture Output -----

```
"D:\java\bin\javaw.exe" array1
x[0] = 1.2
x[1] = 2.3
x[2] = 3.1
total=6.6
> Terminated with exit code 0.
```

2 : double türü bir boyutlu değişken oluşturarak verileri toplama

```
public class array2
{
public static void main(String arg[])
{
double x[]={ 1.2,2.3,3.3};
double total=0;
for(int i=0;i<x.length;i++)
{total+=x[i];
System.out.println("x["+i+"] = "+x[i]);
}
System.out.println("total="+total);
}
}
```

----- Capture Output -----

```
"D:\java\bin\javaw.exe" array2
x[0] = 1.2
x[1] = 2.3
x[2] = 3.3
total=6.8
> Terminated with exit code 0.
```

3 : String türü bir boyutlu değişken oluşturmak

```
public class array3
{
public static void main(String arg[])
{
String x[]{"Ali "," Veli"," 49 "," elli"};
String s="";
for(int i=0;i<x.length;i++)
{s+=x[i];
```

```
System.out.println("x["+i+"] = "+x[i]);
}
System.out.println("s="+s);
}
}
```

----- Capture Output -----

"D:\java\bin\javaw.exe" array3

```
x[0] = Ali
x[1] = Veli
x[2] = 49
x[3] = elli
s=Ali Veli 49 elli
```

> Terminated with exit code 0.

4 : Box3 sınıfı bir boyutlu değişken oluşturmak

```
public class box3
{
    public double length;
    public double width;
    public double height;
    public String color;
    public box3(double l,double w,double h,String s)
    {
        length=l;
        width=w;
        height=h;
        color=s;
    }
    public double volume()
    { return height*width*length;}

    public String output()
    { return "volume = "+volume()+"color = "+color;}
}
```

```
public class array4
{
    public static void main(String arg[])
    {
        box3 x[]=new box3[2];
        x[0]=new box3(1.2,2.3,2.5,"yellow");
        x[1]=new box3(1.1,1.3,1.5,"blue");
        String s="";
        for(int i=0;i<x.length;i++)
        {
            System.out.println("volume of x["+i+"] = "+x[i].volume());
        }
    }
}
```

```
}  
}
```

```
----- Capture Output -----  
"D:\java\bin\javaw.exe" array4  
volume of x[0] = 6.8999999999999995  
volume of x[1] = 2.1450000000000005  
> Terminated with exit code 0.
```

5 : complex sınıfı boyutlu değişkenini oluşturmak

```
public class complex  
{  
    public double real;  
    public double imag;  
    public complex(double r,double i)  
    {real=r;imag=i;}  
    //static addition of two complex variable result returns as complex  
    public static complex add(complex z1,complex z2)  
    {complex z3=new complex((z1.real+z2.real),(z1.imag+z2.imag));  
    return z3;  
    }  
    //dynamic addition to a complex variable into the value of the class  
    public void add(complex z1)  
    {real+=z1.real;  
    imag+=z1.imag;  
    }  
  
    public String output()  
    {String s=real+" +i*"+imag;  
    return s;  
    }  
}
```

```
public class array5  
{  
    public static void main(String arg[])  
    {  
        complex x[]=new complex[2];  
        x[0]=new complex(1.2,2.3);  
        x[1]=new complex(1.1,1.3);  
        complex z=complex.add(x[0],x[1]);  
        System.out.println(z.output());  
    }  
}
```

```
----- Capture Output -----  
"D:\java\bin\javaw.exe" array4  
volume of x[0] = 6.8999999999999995
```



```
volume of x[1] = 2.1450000000000005  
> Terminated with exit code 0.
```

6 : complex sınıfı boyutlu değişkenle hesap

```
public class array6  
{  
public static void main(String arg[])  
{  
complex x[]=new complex[2];  
x[0]=new complex(1.2,2.3);  
x[1]=new complex(1.1,1.3);  
x[0].add(x[1]);  
System.out.println(x[0].output());  
}  
}
```

```
----- Capture Output -----  
"D:\java\bin\javaw.exe" array4  
volume of x[0] = 6.8999999999999995  
volume of x[1] = 2.1450000000000005  
> Terminated with exit code 0.
```

7 : boyutlu değişken katsayıları olan polinom sınıfı f yi oluşturmak

```
public class f  
{// General polynomial function  
// y=a[0]+a[1]*x+a[2]*x^2+...  
public double a[];  
public f(double ia[])  
{int n=ia.length;  
a=new double[n];  
for(int i=0;i<n;i++){a[i]=ia[i];}  
}  
public double func(double x)  
{  
int n=a.length;  
double ff=a[n-1];  
for(int i=n-2;i>=0;i--)  
{ff=ff*x+a[i];}  
return ff;  
}  
}
```

```
public class array7  
{  
public static void main(String arg[])  
{  
double a[]={ 1.0,2.0,3.0};  
f z=new f(a);
```

```
for(double x=0;x<=10;x+=1)
{System.out.println("x="+x+"y="+z.func(x));}
}
}
```

----- Capture Output -----

```
"D:\java\bin\javaw.exe" array7
x=0.0y=1.0
x=1.0y=6.0
x=2.0y=17.0
x=3.0y=34.0
x=4.0y=57.0
x=5.0y=86.0
x=6.0y=121.0
x=7.0y=162.0
x=8.0y=209.0
x=9.0y=262.0
x=10.0y=321.0
```

> Terminated with exit code 0.

8) 10 sayılı bir double değişken seti tanımlayınız ve ortalamalarını hesaplattırınız

```
double x[]={1.2,3.3,4.76,...,1.7465};
```

9) complex sınıfını kullanarak $z[0]=1+2i$, $z[1]=1.2+3i$, $z[2]=0.2+0.1i$, $z[3]=1+i$ sayılarının karelerini hesaplattırınız

```
complex z[]=new complex[4];
```

```
z[0]=new complex(1.0,2.0);
```

...

```
z[3]= new complex(1.0,1.0);
```

daha sonra hepsini ekleyerek kareköklerini hesaplayınız.

```
import java.io.*;
import java.text.*;
import java.util.Locale;
import java.util.*;

import javax.swing.*; // giriş çıkış

// class complex
// complex number abstraction
//
class complex {
    // constructors
    public complex()
    {
        nreal=0;
        nimag=0;
    }

    public complex(double nre,double nim)
    {
        nreal=nre;
```

```

nimag=nim;
}

public complex(int nre,int nim)
{
nreal=(double)nre;
nimag=(double)nim;
}

public complex(double numer)
{
nreal=numer;
nimag=0;
}

public complex(complex value )
{
nreal=value.real();
nimag=value.imaginary();
}

// accessor functions

public double real()
{
//return real part of the complex number
return nreal;
}

public double imaginary()
{
//return imaginary part of the complex number
return nimag;
}

public double R()
{
// Return radius of polar coordinate equivalent of complex number
return Math.sqrt(nreal*nreal+nimag*nimag);
}

public double theta()
{
// Return angle of polar coordinate equivalent of complex number in radian
return Math.atan2(nimag,nreal);
}

public double dtheta()
{
// Return angle of polar coordinate equivalent of complex number in degree
return Math.atan2(nimag,nreal)*180.0/Math.PI;
}

// assignments
public void assign(complex right)
{
nreal=right.real();
nimag=right.imaginary();
}

```

```

}

public void add(complex right)
{
nimag = nimag + right.imaginary();
nreal = nreal + right.real();
}

public void add(double nr1)
{
nreal = nreal+nr1;
}

public void add(double nr1,double ni1)
{
nreal = nreal+nr1;
nimag = nimag + ni1;
}

public void subtract(complex right)
{
nimag = nimag - right.imaginary();
nreal = nreal - right.real();
}

public void multiply(complex right )
{
nreal = nreal*right.real() - nimag*right.imaginary();
nimag = nreal*right.imaginary() + nimag*right.real();
}

public void divide(complex right )
{
double a=nreal*nreal+nimag*nimag;
nreal = ( nreal*right.real() + nimag*right.imaginary())/a;
nimag = (-nreal*right.imaginary() + nimag*right.real())/a;
}

public static complex add(complex left, complex right)
{ // return sum of two complex numbers
double r1=(left.real() + right.real());
double i1=(left.imaginary() + right.imaginary());
complex result;
result=new complex(r1,i1);
return result;
}

public static complex add(complex left, double right)
{ // return sum of two complex numbers
double r1=(left.real() + right);
double i1=left.imaginary();
complex result;
result=new complex(r1,i1);
return result;
}

public void conjugate()
{ nimag*=-1;
}

```

```

public static complex conjugate(complex z)
{ complex z1=new complex(z.real(),-z.imaginary());
return z1;
}

public static complex subtract(complex left, complex right)
{ // return subtraction of two complex numbers
complex result;
result=new complex((left.real() - right.real()),
    (left.imaginary() - right.imaginary()));
return result;
}

public static complex multiply(complex left, complex right)
{ // return multiplication of two complex numbers
complex result;
result=new complex
((left.real()*right.real() - left.imaginary()*right.imaginary()),
    (left.real()*right.imaginary() + left.imaginary()*right.real()));
return result;
}

public static complex multiply(complex left, double right)
{ complex result;
result=new complex((left.real()*right),(left.imaginary()*right));
return result;
}

public static complex divide(complex left, complex right)
{ // return division of two complex numbers
double a=right.real()*right.real()+right.imaginary()*right.imaginary();
complex result;
result=new complex
((left.real()*right.real() + left.imaginary()*right.imaginary())/a,
    (-left.real()*right.imaginary() + left.imaginary()*right.real())/a);
return result;
}

public static complex divide(complex left, double right)
{ // return division of a complex number to a real number
complex result;
result=new complex((left.real()/right),(left.imaginary()/right));
return result;
}

public static complex pow(complex left, double right)
{ // return sum of two complex numbers
double Rad,th;
Rad=Math.pow(left.R(),right);
th=right*left.theta();
complex result;
result =new complex((Rad*Math.cos(th) ),
    (Rad*Math.sin(th) ) );
return result;
}

public static complex exp(complex left)
{ // exp(x+i*y)
complex result;

```

```

result =new complex((Math.exp(left.real()*Math.cos(left.imaginary() )),
                    (Math.exp(left.real()*Math.sin(left.imaginary() )));
return result;
}

public static complex exp(double left)
{ // exp(i*left) imaginary exponent
complex result;
result =new complex(Math.cos(left),
                    Math.sin(left) );
return result;
}

public static complex sqrt(complex left)
{
return pow(left,0.5);
}

public static double abs(complex left)
{
return left.R();
}

public boolean smaller(complex left,complex right)
{
// less then comparison of two complex numbers
return (left.R() < right.R());
}

public boolean smaller_equal(complex left,complex right)
{
// less then and equal comparison of two complex numbers
return (left.R() <= right.R());
}

public boolean greater(complex left,complex right)
{
// greater then comparison of two complex numbers
return left.R() > right.R();
}

public boolean greater_equal(complex left,complex right)
{
// greater then and equal comparison of two complex numbers
return left.R() >= right.R();
}

public boolean equal(complex left,complex right)
{
// equal comparison of two complex numbers
return left.R() == right.R();
}

public boolean not_equal(complex left,complex right)
{
// not equal comparison of two complex numbers
return left.R() != right.R();
}

```

```

}

public static String toString(double left, int w, int d)
// converts a double to a string with given width and decimals.
{
    NumberFormat df=NumberFormat.getInstance(Locale.US);
    df.setMaximumFractionDigits(d);
    df.setMinimumFractionDigits(d);
    df.setGroupingUsed(false);
    String s = df.format(left);
    while (s.length() < w)
        s = " " + s;
    if (s.length() > w)
    {
        s = "";
        for (int i=0; i<w; i++)
            s = s + "-";
    }
    return s;
}

public static String toString(double left)
// converts a double to a string with a constant width and constant decimals.
return toString(left,15,10);}

public static String toString(complex value)
{
String b="";
if(Math.abs(value.imaginary())!=1)
{
if(value.imaginary() >= 0)
b=b+"("+toString(value.real())+" + "+toString(value.imaginary())+"i)";
else
b=b+"("+toString(value.real())+" - "+toString(-value.imaginary())+"i)";
}
else
{
if(value.imaginary() >= 0)
b=b+"("+toString(value.real())+" + i)";
else
b=b+"("+toString(value.real())+" - i)";
}
return b;
}

public String toString()
{
String b="";
if(Math.abs(imaginary())!=1)
{
if(imaginary() >= 0)
b=b+"("+toString(real())+" + "+toString(imaginary())+"i)";
else
b=b+"("+toString(real())+" - "+toString(-imaginary())+"i)";
}
else
{
if(imaginary() >= 0)

```

```

    b=b+"("+toString(real())+" + i)";
else
    b=b+"("+toString(real())+" - i)";
}
return b;
}
public static complex toComplex(String s)
{
    //bu metod kompleks sayıyı ekrandan okur.
    //StringTokanizer kütüphane sınıfı bir stringi cümlelere ayırır
    String s1=JOptionPane.showInputDialog(s);
    StringTokenizer token=new StringTokenizer(s1);
    int n=token.countTokens()-1;
    int m=n+1;
    double a[]=new double[m];
    int j=0;
    while(token.hasMoreTokens())
    {
        Double ax=new Double(token.nextToken());
        a[j++]=ax.doubleValue();
    }
    complex b=new complex(a[0],a[1]);
    return b;
}
// data areas
public double nreal;
public double nimag;

};

```

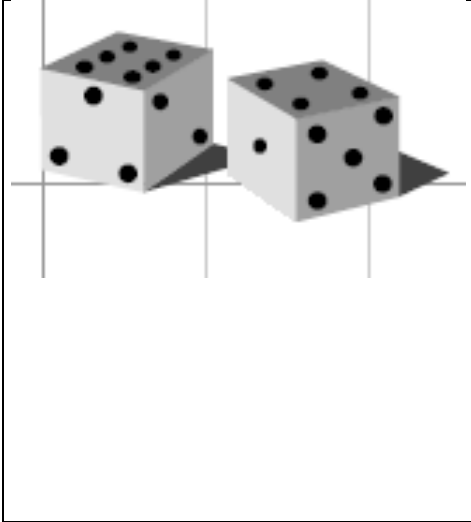
10) 3 boyutlu karteziyen koordinat sisteminde tanımlanan nokta3D P(1,1,1); P(1,2,1), P(1,0,0) and P(2,1,2). Noktalarını tanımlayınız. Bu noktalar arasındaki toplam mesafeyi hesaplayınız.

```

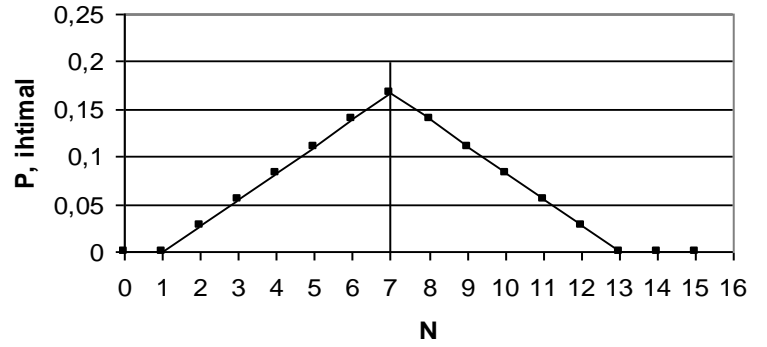
nokta3D P[]=new nokta3D[4];
P[0]= new nokta3D(1.0,1.0,1.0);
...
P[3]= new nokta3D(2.0,1.0,2.0);

```

11)Tavla oyunu iki zar atarak oynanır. Tek bir zar atımında her zarın olasılığı 1/6 dır. Ancak 2 zar atıldığında iki zarın toplamı değişik zar kombinasyonlarından etkileneceği için değişir. 2 gelme olasılığı $1/6 * 1/6 = 1/36 = 0.0277777777$ iken 3 gelme olasılığı ([1 2] veya [2 1] zarlarının toplamı 3 edeceğinden $1/6 * 1/6 + 1/6 * 1/6 = 0.0555555$ dir.



2 zar atma ihtimali



iki zar atışta sayıların dağılımı

zar toplamı	zar kombinasyonu	ihtimal		zar frekansı
0		0		0
1		0		0
2	11	0,027778	1/36	1
3	12 21	0,055556	2/36	2
4	22 13 31	0,083333	3/36	3
5	14 41 23 32	0,111111	4/36	4
6	33 15 51 42 24	0,138889	5/36	5
7	16 61 25 52 34 43	0,166667	6/36=1/6	6
8	44 26 62 35 53	0,138889	5/36	5
9	26 63 45 54	0,111111	4/36	4
10	55 46 64	0,083333	3/36	3
11	56 65	0,055556	2/36	2
12	66	0,027778	1/36	1
13		0		0
14		0		0
15		0		0
	Toplam	1		36

Bu durumu gerçek zar atarak ve gelen zarların istatistiğini hesaplayarak görelim.

```

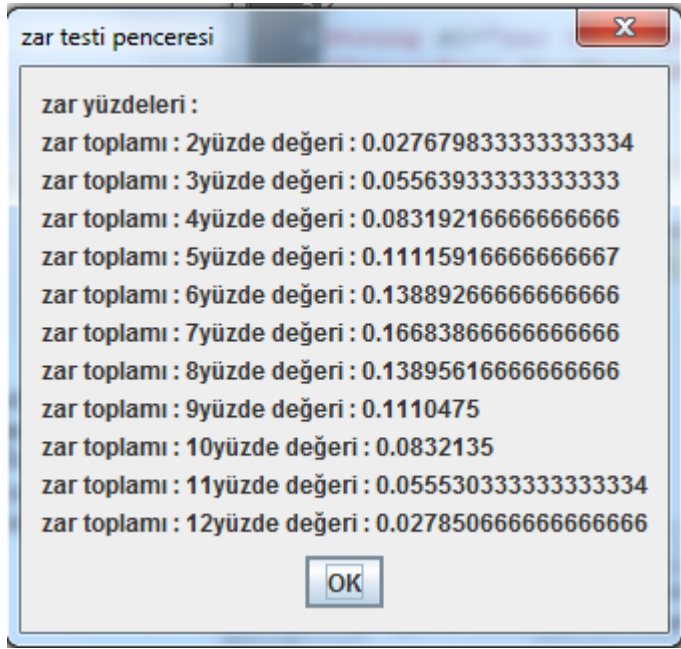
public class zaristatistigi
{
//çift zar atıldığında her zarın gelme olasılığını
//zar atarak hesaplar
public static int zar()
{
return 1+(int)(Math.random()*6);
}
public double[] zaryuzdesi()
{
double n=6000000;
double gelensayi[]=new double[13];

```

```
int sayi=0;
for(int i=0;i<n;i++)
{
sayi=zar()+zar();
for(int j=2;j<=12;j++)
{if(sayi==j) gelensayi[j]++; }
}
for(int j=2;j<=12;j++)
{gelensayi[j]/=n; }
return gelensayi;
}
public String cikti()
{
String s="zar yüzdeleri : \n";
double yuzde[]=zaryuzdesi();
for(int j=2;j<=12;j++)
{s+="zar toplamı : "+j+"yüzde değeri : "+yuzde[j]+"\\n"; }
return s;
}
}
```

```
import javax.swing.*;
class zartest1
{
public static void ciktipenceresi(String s)
{
String s1="zar testi penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
zaristatistigi x=new zaristatistigi();
ciktipenceresi(x.cikti());
}}}
```



12) Bir önceki bölümde kitap eksersizinden kitapların çıktısını elde etmek için s String değişkenine tek tek ilave etmemiz gerekmişti. Boyutlu değişken kullanarak aynı işi döngü değişkenleri kullanarak da yapabiliriz. Örnek problemi inceleyiniz.

```
import javax.swing.*;
class kitaptest3
{
public static void ciktipenceresi(String s)
{
String s1="kitap sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
kitap dergi=new kitap("Bilim ve Teknik","TÜBİTAK","dergi","Turkish");
roman r1=new roman("It is all quiet in the westen front ","Erich Maria Reamarque","İngilizce");
kitap r2=new kitap("The God Delusion","Richard Dawkins","roman","İngilizce");
siir s1=new siir("Human lanscapes from my country","Nazım Hikmet","İngilizce ve Türkçe");
ansiklopedi a1=new ansiklopedi("Encyclopedia Britannica","İngilizce");
kitap k[]=new kitap[5];
k[0]=dergi;
k[1]=r1;
k[2]=r2;
k[3]=s1;
k[4]=a1;
String s="";
for(int i=0;i<k.length;i++)
{s+=k[i].cikti()+"\n";}
ciktipenceresi(s);
}
}
```

5.8 PROBLEMLER

1) aşağıda iki matrisi çarpmak için bir metod tanımlanmıştır. Bu metodu matris sınıfına ekleyerek iki matrisi çarpacak bir örnek program yazınız.

```
public static double[][] carp(double[][] left,double[][] right)
{
//iki matrisin çarpımı
int ii,jj,i,j,k;
int m1=left[0].length;
int n1=left.length;
int m2=right[0].length;
int n2=right.length;
double[][] b;
b=new double[m1][n2];
if(n1 != m2)
{
System.out.println("inner matrix dimensions must agree");
for(ii=0;ii<n1;ii++)
{
for(jj=0;jj<m2;jj++)
b[ii][jj]=0;
}
return b;
}
for(i=0;i<m1;i++)
{
for(j=0;j<n2;j++)
{
for(k=0;k<n1;k++)
b[i][j]+=left[i][k]*right[k][j];
}
}
return b;
//end of multiply of two matrices
}
```

2. complex sınıfını kullanarak $z[0]=1+2i$, $z[1]=1.2+3i$, $z[2]=0.2+0.1i$, $z[3]=1+i$ sanal sayılarını çarpınız.

```
complex z[]=new complex[4];
z[0]=new complex(1.0,2.0);
...
z[3]= new complex(1.0,1.0);
```

3 iki vektörü çarpacak bir metod oluşturunuz.

public static double[] carp(double[] left,double[] right) bir örnek progrmda kullanarak sonuçları kontrol ediniz.

4. Bayagikesir sınıfı tanımlanmıştır. $[\frac{1}{2}]$; $1[\frac{3}{4}]$; $[\frac{5}{8}]$ ve $[\frac{3}{7}]$ bayağı kesir değerlerini bir boyutlu bayagikesir değişkenine yükledikten sonra ortalamalarını hesaplayacak bir program geliştiriniz.

5. 3 zar attığımızda zarların yüzdesinin ne şekilde değiştiğini bir bilgisayar programı aracılığıyla irdelleyiniz. (alıştırma 4 ü inceleyiniz)

6 ABSTRACT SINIFLAR, INTERFACE, İNDİREK REFERANS KULLANIMI

6.1 ABSTRACT SINIFLAR

Önceki bölümlerimizde extends kelimesini kullanarak bir sınıfın diğer sınıftan türetilmesini görmüştük. Aynı zamanda türetme zincirinde kök teşkil eden sınıfın tüm türetilen sınıflar için referans olarak kullanılabileceğini de gördük. Daha yazılmamış olan sınıfları bir indirek referans sınıfı üzerinden ifade ederek programların yazılması programcılıkta oldukça yararlıdır. Bu yüzden sadece yeni sınıf veya sınıfların türetilmesinde bir kök sınıf rolü oynaması amaçlanan, çok az bir içerik taşıyan sınıf yapıları kullanılabilir. Extends sözcüğü ile türetilen sınıf dizisinde kök sınıf rolü oynayacak sınıfa **abstract** sınıf adını veriyoruz. Abstract sınıfın tanımı şu şekildedir:

abstract class class_name

```
{  
class variable list  
class method list  
}
```

Bu yapının içinde sınıf abstract olduğu gibi metodlar da abstract (boş) olarak verilebilir. Eğer metod abstract olarak tanımlanmışsa bu sınıftan extend yoluyla türetilen sınıfta bu metodun tanımının yapılması şarttır. Bir örnek abstract sınıf tanımlayalım:

Program 6.1.1 Abstract sınıf şekil

```
public abstract class şekil  
{ public double alan(){return 0.0;}  
public double hacim() {return 0.0;}  
public abstract String isim();  
public abstract String cikti();  
}
```

Bu sınıfta isim ve cikti metodlarının abstract olduğunu not edelim. Bir başka abstract sınıf örneği f_x sınıfı

Program 6.1.2 Abstract class f_x

```
abstract class f_x  
{ abstract double func(double x);}
```

Bu sınıfın içinde de verilmiş olan tek metod func abstract olarak tanımlanmıştır (metodun içi boştur). Bu asıl metodun bu sınıftan türetilmiş sınıf içinde tanımlanacağı anlamına gelir. Sınıfların birbirinden türetilmesini göstermek için şekil sınıfından nokta3 sınıfını türetilim. Nokta3 sınıfından daire3 ve daire3 sınıfından silindir3 sınıfını türetilim. Böylece birbirinden türemiş şekil->nokta3->

Program 6.1.3 şekil sınıfından türetilmiş nokta3 sınıfı

```
public class nokta3 extends şekil  
{  
public double xi,xj;  
  
public nokta3()  
{  
xi=0;  
xj=0;  
}}
```

```

public nokta3(double yi,double yj)
{
xi=yi;
xj=yj;
}

public nokta3(nokta1 y)
{
xi=y.xi;
xj=y.xj;
}

public double alan()
{return 0.0;}

public double hacim(){return 0.0;}

public String cikti()
{
String s="";
s+=xi+" i ";
if(xj>=0 )
s+=" "+xj+" j";
else if(xj<0)
s+="- "+Math.abs(xj)+" j ";
return s;
}
public String isim(){return "nokta";}
}

```

Program 6.1.4 nokta3 sınıfından türetilmiş daire3 sınıfı

```

public class daire3 extends nokta3
{
public double yaricap;
// bu sınıfta xi ve xj olmak üzere
// iki tane daha sınıf değişkeni var
public daire3()
{ super(0.0,0.0);
yaricapgir(0.0);}

public daire3(double r, double a, double b)
{ super(a,b);
yaricapgir(r);
}
public daire3(daire1 d)
{ super(d.xi,d.xj);
yaricapgir(d.yaricap);
}
public daire3(double r, nokta1 n)
{ super(n);
yaricapgir(r);
}

public void yaricapgir(double r)
{ if(r >= 0.0)
yaricap=r;
else
yaricap=0.0;
}
}

```

```

}

public double alan()
{return Math.PI*yaricap*yaricap;}
public double hacim(){return 0.0;}
public String cikti()
{ return "Merkez = "+super.cikti()+" yarıçap = "+yaricap;}

public String isim(){return "daire";}
}

```

Program 6.1.5 daire3 sınıfından türetilmiş silindir3 sınıfı

```

public class silindir3 extends daire3
{
protected double yukseklik;

public silindir3()
{super(0.0,0.0,0.0);
yuksekligir(0);}

public silindir3( double h,daire1 d)
{super(d);
yuksekligir=h;
}

public silindir3( double h, double r, double a, double b)
{super(r,a,b);
yuksekligir=h;
}

public void yukseklikgir (double h )
{ if(h>=0) yukseklik=h;
else yukseklik=0;
}

public double alan()
{return 2*super.alan()+2*Math.PI*yaricap*yuksekligir;}

public double hacim(){return super.alan()*yuksekligir;}

public String cikti()
{return super.cikti()+" yükseklik = "+yuksekligir;}

public String isim(){return "silindir";}
}

```

When these classes needs to be used the abstract class shape can be used to get Access to any of these classes.

Program 6.1.6 output example class H9Ex1

```

import javax.swing.*;
public class abstracttest1 {
public static void ciktipenceresi(String s)
{
String s1="indirek adres kullanımı";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}
}

```

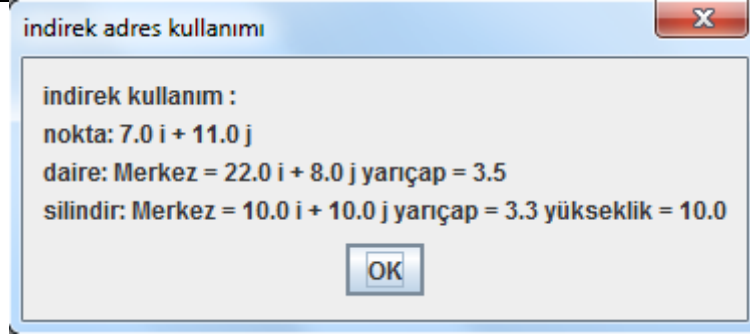
```

}

public static void main(String args[])
{
    nokta3 n = new nokta3(7,11);
    daire3 d = new daire3(3.5,22,8);
    silindir3 s = new silindir3(10,3.3,10,10);
    sekil a[] = new sekil[3];
    a[0]=n;
    a[1]=d;
    a[2]=s;

    String s2="";
    s2+="indirek kullanım : \n";
    for(int i=0;i<a.length;i++)
    {
        s2+=a[i].isim()+" : "+a[i].cikti()+"\n";
    }
    ciktipenceresi(s2);
}
}

```



Görüldüğü gibi tüm bu sınıflara abstract sınıf sekil üzerinden ulaşılabilmektedir. Diğer bir örnek olarak sayısal metodunu tanımlayalım. Bu metodun içinde integral ve türev metodları tanımlanmıştır. integrali ve türevi

Program 6.1.7 abstract f_x sınıfını kullanan sayısal sınıfı.

```

public class sayısal
{

public static double turev(f_x f,double x)
{
    double h=0.00001;
    return (-f.func(x+2.0*h)+8.0*f.func(x+h)-8.0*f.func(x-h)+f.func(x-2.0*h))/(12.0*h);
}

public static double integral(f_x f_xnt,double a,double b)
{
    // integral f(x) dx
    double r[]={-0.973906528517171,-0.865063366688984,-0.679409568299024,-0.433395394129247,-0.148874338981631,
0.148874338981631,0.433395394129247,0.679409568299024,0.865063366688984,0.973906528517171};
    double c[]={0.066671344308684,0.149451349150580,0.219086362515982,0.269266719309996,0.295524224714752,
0.295524224714752,0.269266719309996,0.219086362515982,0.149451349150580,0.066671344308684};
    double z=0,x,y;
    double k1=(b-a)/2.0;
    double k2=(b+a)/2.0;
    for(int i=0;i<r.length;i++)
    {
        x=k2+k1*r[i];

```



```
y=f_xnt.func(x);
z+=k1*c[i]*y;
}
return z;
}
}
```

Bu programda fonksiyon olarak abstract sınıf `f_x` in metodu olan `func` metodu kullanılmıştır, ancak gerçek bir fonksiyon verilmemiştir. Bu sınıftaki metodlar kullanılacağı zaman fonksiyonun içinde olduğu yeni bir sınıf `f_x` sınıfından extends yoluyla türetilerek kullanılabilir.

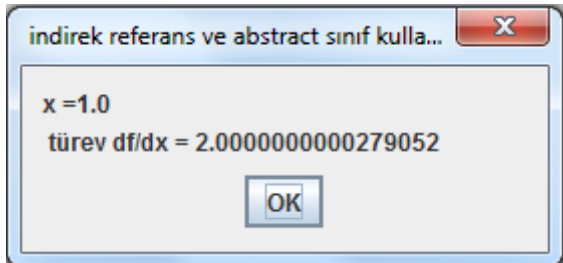
Program 6.1.8 `f_x` sınıfından türetilmiş `fb` sınıfını kullanan sayisaltest sınıfı

```
import javax.swing.*;

class fb extends f_x
{ public double func(double x)
  { return x*x-5.0;}}

public class sayisaltest {
public static void main(String args[])
{ fb ff=new fb();
  double x=1.0;
  String s="x="+x+"\n türev df/dx = "+sayisal.turev(ff,x);
  String s1=" indirek referans ve abstract sınıf kullanımı :";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}
```

Örnek problemde `f_x` sınıfından türetilen `fb` sınıfı oluşturulmuş ve bu sınıf ana programda kullanılmıştır.



Bu şekilde indirek kullanımın en önemli avantajı değişik sınıfları çağırabilme olasılığıdır.. Bunu son uygulamamıza aktarırsak değişik fonksiyonları değerlendirmemiz mümkün olabilir.

Program 6.1.9 Abstract sınıf ve indirek referans örneği

```
import javax.swing.*;

class fb extends f_x
{ public double func(double x)
  { return x*x-5.0;}}

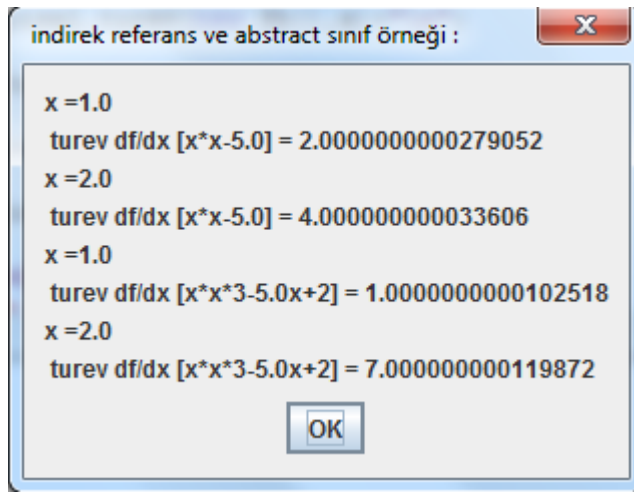
class fc extends f_x
{ public double func(double x)
  { return x*x*3-5.0*x+2;}}
```

```

public class sayisaltest2 {

public static void main(String args[])
{
    double x=1.0;
    String s="x =" +x+"\n turev df/dx [x*x-5.0] = "+sayisal.turev(new fb(),x)+"\n";
    x=2;
    s+="x =" +x+"\n turev df/dx [x*x-5.0] = "+sayisal.turev(new fb(),x)+"\n";
    x=1;
    s+="x =" +x+"\n turev df/dx [x*x*3-5.0x+2] = "+sayisal.turev(new fc(),x)+"\n";
    x=2;
    s+="x =" +x+"\n turev df/dx [x*x*3-5.0x+2] = "+sayisal.turev(new fc(),x)+"\n";
    String s1=" indirek referans ve abstract sınıf örneği :";
    JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```



6.2 INTERFACE (ARABAĞ)

Üst bölümde öğrendiğimiz abstract sınıfının kullanıldığı extend deyimi sadece bir sınıfın bağlanmasını sağlıyordu. Azen birden fazla sınıftan bağlantı gereksinmesi olabilir. Bu durumda abstract sınıf yerine interface kullanılır. Interface sınıfının içinde sadece abstract metodlar yer alır (metodun içinde kod verilmez) ancak abstract sözcüğü kullanılmaz. Bu sınıftan yenibir sınıf türetilmek istendiğinde **implements** kelimesiyle bağlantı sağlanır. Temel olarak bu bağlantı interface sile tanımlanan sanal sınıfta bulunan temel metod ve sınıf değişkenlerinin bu metotta tanımlanmasını zorunlu kılar, eğer bu metodlar tanımlanmamışsa hata verir. Interface'in kendi başına bir sınıf oluşturmadığını sadece bağlantı yapmak için bir şablon teşkil ettiğini burada hatırlatalım.

Interface tanımı

```

interface sınıf_ismi
{
Sınıf değişkenleri listesi
Sınıf metodları listesi
}

```

İnterfacete metod gövdesi mevcut değildir, sadece girdi çıktı tanımı mevcuttur. İlk interface örneği olarak şekil 1 i vereceğiz.

Program 6.2.1 interface sekil1

```
public interface sekil1
{public double alan();
 public double hacim();
 public String isim();
 public String cikti();
}
```

Bu tanımlı abstract sınıf sekil ile karşılaştırsak önce abstract kelimesinin kullanılmadığını ancak tüm metodların abstract olduğunu (metod gövdesi olmadığını) görürüz. İkinci örnek olarak f_x1 sınıfı verildi.

Program 6.2.2 interface f_x1

```
public interface f_x1
{ double func(double x);}
```

Bu metodda da diğer interface'lerde olduğu gibi herhangi bir metod gövdesi mevcut değildir. Interface'lerin sınıf oluşturmada kullanılmalarını göstermek için bir önceki bölümde kullandığımız nokta örneğini bu kavrama da uyarlıyalım.

In this interface like all other interfaces, no method definition is given. To see how a method is written by using interface and used, classes point and circle and cylinder is created as extended from class shape. The word **implements** is used in interface connection instead of word **extend**.

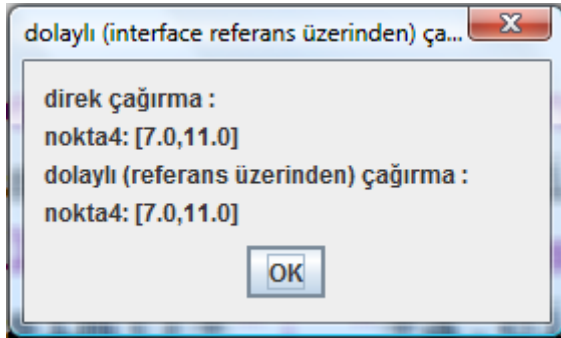
Program 6.2.3 interface sekil1 den implements kelimesiyle türetilen nokta4 sınıfı

```
public class nokta4 implements sekil1
{ double x, y;
  public nokta4(double a, double b) { x=a;y=b;}
  public String cikti(){ return "["+x+","+y+"]";}
  public double alan() {return 0.0;}
  public double hacim(){return 0.0;}
  public String isim() {return "nokta4";}
}
```

Nokta4 sınıfını incelediğimizde interface'de tanımlanan tüm sınıf değişkenlerinin ve metodların tanımlarının burada yer aldığını görüyoruz. Şimdi nokta4 sınıfını bir örnek programda kullanırsak :

Program 6.2.4 nokta4 sınıfını test eden nokta4test sınıfı

```
import javax.swing.*;
class nokta4test {
public static void main(String args[])
{ nokta4 n = new nokta4(7,11);
  sekil1 a;
  a=n;
  String s="direk çağırma : \n";
  s+=n.isim()+": "+n.cikti()+"\n";
  s+="dolaylı (referans üzerinden) çağırma : \n";
  s+=a.isim()+": "+a.cikti()+"\n";
  String s1="dolaylı (interface referans üzerinden) çağırma";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}
```



Örnekten de görüldüğü gibi burada da extends üzerinden kurduğumuz zincir gibi indirek çağırma müsaadesine sahibiz. Interface bir anlamda gölge bir abstract sınıf gibi davranmaktadır, tek önemli farkı bizi tanımlamaya zorladığı tüm değişkenlerin ve metodların bir alt sınıfta tanımlanma zorunludur.

Program 6.2.5 interface f_x1 kullanan sayisal1 sınıfı.

```
public class sayisal1
{
public static double turev(f_x1 f,double x)
{ double h=0.00001;
return (-f.func(x+2.0*h)+8.0*f.func(x+h)-8.0*f.func(x-h)+f.func(x-2.0*h))/(12.0*h);
}

public static double integral(f_x1 f_xnt,double a,double b)
{
// integral f(x) dx
double r[]={-0.973906528517171,-0.865063366688984,-0.679409568299024,-0.433395394129247,-
0.148874338981631,
0.148874338981631,0.433395394129247,0.679409568299024,0.865063366688984,0.973906528517171};
double
c[]={0.066671344308684,0.149451349150580,0.219086362515982,0.269266719309996,0.295524224714752,
0.295524224714752,0.269266719309996,0.219086362515982,0.149451349150580,0.066671344308684};
double z=0,x,y;
double k1=(b-a)/2.0;
double k2=(b+a)/2.0;
for(int i=0;i<r.length;i++)
{
x=k2+k1*r[i];
y=f_xnt.func(x);
z+=k1*c[i]*y;
}
return z;
}}
```

Program 6.2.6 f_x1 ve sayisal1 sınıfını kullanan sayisal1test test programı

```
import javax.swing.*;

class fb implements f_x1
{ public double func(double x)
{ return x*x-5.0;}}

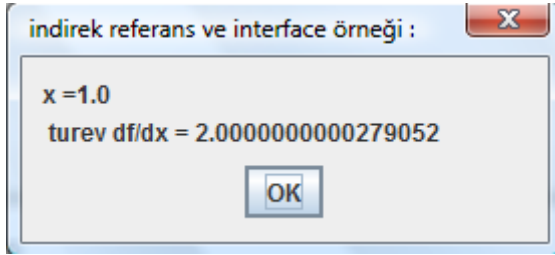
public class sayisal1test {
public static void main(String args[])
```

```

{ fb ff=new fb();
  double x=1.0;
  String s="x =" +x+"\n turev df/dx = "+sayisal1.turev(ff,x);
  String s1=" indirek referans ve interface örneği :";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```

Örnek programda fb f_x1 sınıfından implement kelimesiyle türetilmiştir. Sayisal1 sınıfının türev ve integral metodlarında girdi değişkeni f_x1 olarak verilmiş, ancak fb sınıfından değişken kullanılmıştır (indirek referans). Program çıktısı :



Örneklerden de görüldüğü gibi interface kullanımı abstract sınıfı kullanımı ile çok benzerdir. En önemli farkı abstract sınıfta extends ile kopyelerken kodların ve değişkenlerin bire bir aktarılması, interface de ise bu değişkenlerin ve metodların tanımlanma zorunluğunu getirmesidir. Yalnız java 8 den itibaren bu kural yumuşatılmıştır. **Default** deyiimiyle birlikte interface içinde metod tanımlı yapılabilir, metodu kullanmak için ikinci bir metoda interface olarak bağlanarak kullanılması gerekir.

Program 6.2.7 f_x1 ve sayisal1 sınıfını kullanan sayisal1test test programı

```

interface mesaj {
    // java 8 default metodu eklendi
    default String merhabaDe() {
        return "Merhaba nasılsın!";
    }
}

class m1 implements mesaj {}

public class interfacetest1 {
    public static void main(String [] args) {
        m1 x = new m1();
        //burada interfacede default olarak tanımlanan metodu çağırıyoruz
        System.out.println(x.merhabaDe());
    }
}

```

```

----- Capture Output -----
> "C:\java\bin\java.exe" interfacetest1
Merhaba nasılsın!

> Terminated with exit code 0.

```

6.3 İÇ SINIF TANIMI VE LAMBDA DEYİMLERİ

Yukarıda tanımladığımız f_x interface'ini göz önüne alalım. Bu interface'i sayisal1test sınıfında test etmiştik. Şimdi aynı programı bir iç sınıf olarak yazalım. İç sınıflar metodların içinde tanımlanmış sınıflardır. Burada daha önceki örneğimizde normal bir sınıf olarak tanımladığımız fb sınıfı bir iç sınıf olarak tanımlanmıştır.

Program 6.3.1 f_x1 ve sayisal1 sınıfını kullanan sayisal1test1 test programı, iç sınıf kullanan versiyon

```

import javax.swing.*;
public class sayisal1test1 {
public static void main(String args[])
{   class fb implements f_x1
    { public double func(double x) { return x*x-5.0;}}
fb ff=new fb();
double x=1.0;
String s="x="+x+"\n derivative df/dx = "+sayisal1.turev(ff,x);
String s1=" indirect referencing and interface example with inner class :";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}

```

Görüldüğü gibi iç sınıf tanımlamanın tanımlamanın metodların içinde yapılması dışında dış sınıf tanımından bir farkı yoktur. Java 8 'den itibaren implement ile belli bir sınıfa bağlanmış iç sınıf tanımlarının çok daha kısa bir şekilde tanımlanmasına izin verilmiştir. Bu yeni tanım şekline **lamda deyimleri** adını veriyoruz.

Program 6.3.2 f_x1 ve sayisal1 sınıfını kullanan sayisal1test2 test programı, lamda deyimi kullanan versiyon

```

import javax.swing.*;
public class sayisal1test2 {
public static void main(String args[])
{ f_x1 ff = x -> x*x-5.0;
double x=1.0;
String s="x="+x+"\n derivative df/dx = "+sayisal1.turev(ff,x);
String s1=" indirect referencing and interface example with inner class :";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```

Elbette interface tanımı sadece fonksiyon tanımlamak için kullanılmayabilir. Tanım her türlü metod için geçerlidir. Yalnız, interface'de sadece bir metod olmalıdır, böylece lambda tanımı hangi metodu yazacağını karıştırmaz. Lambda deyiminde deyim yazıldığı metod adının verilmediğine dikkatinizi çekeriz. Bu tür tek metodlu interface'lere fonksiyon tipi interface diyeceğiz. Lamda deyimine ikinci bir örnek daha verelim:

Program 6.3.3 f_boolean interface

```

public interface f_boolean
{ boolean karar(double x,double y);}

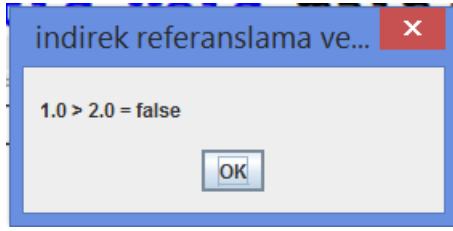
```

Program 6.3.4 f_boolean interface test programı dış sınıf fc yi kullanıyor

```

import javax.swing.*;
class fc implements f_boolean
{ public boolean karar(double x,double y)
  { return x>y;}
}
public class booleantest1 {
public static void main(String args[])
{
fb ff=new fc();
double x=1.0;
double y=2.0;
String s=x+" > "+y+" = "+ff.karar(x,y);
String s1=" indirek referanslama ve interface örneği:";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}

```



Aynı programı iç sınıf formunda verirsek:

Program 6.3.5 f_boolean interface test programı iç sınıf fc yi kullanıyor

```
import javax.swing.*;

public class booleantest2 {
public static void main(String args[])
{ class fc implements f_boolean
  { public boolean karar(double x,double y)
    { return x>y;}
  }
  fc ff=new fc();
  double x=1.0;
  double y=2.0;
  String s=x+" > "+y+" = "+ff.karar(x,y);
  String s1=" indirek referanslama ve interface örneği:";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}
```

Aynı programı lamda deyimi ile yazarsak:

Program 6.3.6 f_boolean interface test programı lambda deyimi kullanıyor

```
public class booleantest3 {
public static void main(String args[])
{ f_boolean ff=(x,y)->x>y;
  double x=1.0;
  double y=2.0;
  String s=x+" > "+y+" = "+ff.karar(x,y);
  String s1=" indirek referanslama ve interface örneği:";
  JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}
}
```

Lambda deyimlerinde kullandığımız interfacelerde sadece bir metod olma gereğini yineleyelim. Bu tür interfaceler fonksiyon interface adını alırlar. Zorunlu olmamakla birlikte fonksiyon tanımının önüne **@FunctionalInterface** terimini kullanabiliriz.

Program 6.3.7 isci_interface interface test programı lambda deyimi ve isimsiz iç sınıf kullanıyor

```
//fonksiyon interface tanımı
@FunctionalInterface
interface isci_interface {
  public void bir_is_yap();
}
public class isci_interfaceTest {

  public static void execute(isci_interface isci) {
    isci.bir_is_yap();
  }

  public static void main(String [] args) {
    //isimsiz iç sınıf tanımla
    execute(new isci_interface() {
      @Override
      public void bir_is_yap() {
```

```

        System.out.println("isci_interface isimsiz bir iç sınıfla kullanılıyor");
    }
    });
    //lambda deyimini çalıştır
    execute( () -> System.out.println("isci_interface lamda deyimiiyle kullanılıyor") );
}
}

```

Son olarak default metodunun kullanılarak fonksiyon interface hesaplaması örneğine bakalım. Interface'deki java 8 ile gelen default metodlar lambda deyimini kullanılmasını çok pratik hale getirebilece yeni eklentilerdir. inter_f_x interface'i içinde dört tane default method ve bir tane metod tanımı barındırmaktadır. Default metod tanımları türev ve integral hesaplamasında daha önceki örnekte Numeric sınıfı üzerinden yaptığımız işlemleri interface sınıfı içinde yapma olasılığı sağlamaktadır.

Program 9.3.9 örnek interface sınıfı inter_f_x

```

import static java.lang.Math.*;
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

@FunctionalInterface
interface inter_f_x
{
    //Function
    public double func(double x);

    default double dfunc(double x)
    { double h=1.0e-3;
      int n=1;
      return dfunc_n(x,n,h);
    }

    default double dfunc(double x,double h)
    {int n=1;
      return dfunc_n(x,n,h);
    }

    default double dfunc(double x,int n)
    { double h=1.0e-3;
      return dfunc_n(x,n,h);
    }

    default double dfunc_n(double x,int n,double h)
    {
      double hh=1/h;
      double df=0;
      if(n==0) df=func(x);
      else if(n==1)
      { df=(3.0*func(x-4.0*h)-32.0*func(x-3.0*h)+168.0*func(x-2.0*h)-672.0*func(x-h)+672.0*func(x+h)-
        168.0*func(x+2.0*h)+32.0*func(x+3.0*h)-3.0*func(x+4.0*h))/840.0*hh; }
      else if(n==2)
      { df=(-14350.0*func(x)-9.0*func(x-4*h)+128*func(x-3*h)-1008*func(x-2*h)+8064*func(x-h)+8064.0*func(x+h)-
        1008.0*func(x+2.0*h)+128.0*func(x+3.0*h)-9.0*func(x+4.0*h))/5040.0*hh*hh; }
      else if(n==3)
      { df=(-7.0*func(x-4.0*h)+72.0*func(x-3.0*h)-338.0*func(x-2.0*h)+488.0*func(x-h)-

```



```

488.0*func(x+h)+338.0*func(x+2.0*h)-72.0*func(x+3.0*h)+7.0*func(x+4.0*h))/240.0*hh*hh*hh;}
else if(n==4)
{ df=(2730.0*func(x)+7.0*func(x-4.0*h)-96.0*func(x-3.0*h)+676.0*func(x-2.0*h)-1952.0*func(x-h)-
1952.0*func(x+h)+676.0*func(x+2.0*h)-96.0*func(x+3.0*h)+7.0*func(x+4.0*h))/240.0*hh*hh*hh*hh;}
else if(n==5)
{ df=(func(x-4.0*h)-9.0*func(x-3.0*h)+26.0*func(x-2.0*h)-29.0*func(x-h)+29.0*func(x+h)-
26.0*func(x+2.0*h)+9.0*func(x+3.0*h)-func(x+4.0*h))/6.0*hh*hh*hh*hh*hh;}
else if(n==6)
{ df=(-150.0*func(x)-func(x-4.0*h)+12.0*func(x-3.0*h)-52.0*func(x-2.0*h)+116.0*func(x-h)+116.0*func(x+h)-
52.0*func(x+2.0*h)+12.0*func(x+3.0*h)-func(x+4.0*h))/4.0*hh*hh*hh*hh*hh*hh;}
else if(n==7)
{ df=(-func(x-4.0*h)+6.0*func(x-3.0*h)-14.0*func(x-2.0*h)+14.0*func(x-h)-14.0*func(x+h)+14.0*func(x+2.0*h)-
6.0*func(x+3.0*h)+func(x+4.0*h))/2.0*hh*hh*hh*hh*hh*hh*hh;}
else if(n==8)
{ df=(70.0*func(x)+func(x-4.0*h)-8.0*func(x-3.0*h)+28.0*func(x-2.0*h)-56.0*func(x-h)-
56.0*func(x+h)+28.0*func(x+2.0*h)-8.0*func(x+3.0*h)+func(x+4.0*h))*hh*hh*hh*hh*hh*hh*hh*hh;}
else df=0;
return df;
}
//INTEGRAL
default double integral(double a,double b)
{ //10 point Gauss-Legendre formula
//integral f(x)dx
double r[]={-0.973906528517171,-0.865063366688984,-0.679409568299024,-0.433395394129247,-
0.148874338981631,
0.148874338981631,0.433395394129247,0.679409568299024,0.865063366688984,0.973906528517171};
double c[]={0.066671344308684,0.149451349150580,0.219086362515982,0.269266719309996,0.295524224714752,
0.295524224714752,0.269266719309996,0.219086362515982,0.149451349150580,0.066671344308684};
double z=0,x,y;
double k1=(b-a)/2.0;
double k2=(b+a)/2.0;
for(int i=0;i<r.length;i++)
{
x=k2+k1*r[i];
y=func(x);
z+=k1*c[i]*y;
}
return z;
}
}

```

Program 9.3.10 örnek interface sınıfı dış sınıf olarak test programı

```

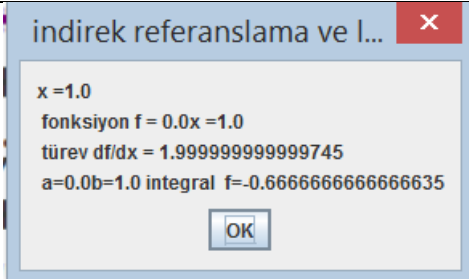
import javax.swing.*;
class ff1 implements inter_f_x
{public double func(double x)
{return x*x-1.0;}
}
public class inter_fx_Ex1 {
public static void main(String args[])
{ ff1 ff=new ff1();
double x=1.0;
String s="x =" +x+"\n fonksiyon f = "+ff.func(x);

```

```

s+="x="+x+"\n türev df/dx = "+ff.dfunc(x);
double a=0;double b=1.0;
s+="\n a="+a+"b="+b+" integral f="+ff.integral(a,b);
String s1=" indirek referanslama ve lambda deyimleri örneği:";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```



Program 9.3.11 örnek interface sınıfı lambda deyimi olarak test programı

```

import javax.swing.*;

public class inter_fx_Ex4{
public static void main(String args[])
{ inter_f_x ff=x->x*x-1.0;
double x=1.0;
String s="x="+x+"\n fonksiyon f = "+ff.func(x);
s+="x="+x+"\n türev df/dx = "+ff.dfunc(x);
double a=0;double b=1.0;
s+="\n a="+a+"b="+b+" integral f="+ff.integral(a,b);
String s1=" indirek referanslama ve lambda deyimleri örneği:";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```

Lambda deyiminde kullandığımız interface metodu girdisi tek bir değişken olmak zorunda değildir, birden fazla değişken olabileceği gibi boyutlu değişken de olabilir.

Program 9.3.12 örnek interface if_xj

```

@FunctionalInterface
interface if_xj
{
// tek fonksiyon çoklu bağımsız değişken
// tek çıktı değeri verir
public double func(double x[]);
default double dfunc(double x[],int x_ref)
{ // fonksiyonun x_ref indisli değişken göre türevi
double h0=0.0256;
int i,m;
int n=7;
double f1,f2;
double x1[];
x1=new double[x.length];
double x2[];
x2=new double[x.length];
for(i=0;i<x.length;i++)

```

```

{
x1[i]=x[i];
x2[i]=x[i];
}
//fonksiyonun türevi
double T[][];
T=new double[n][n];
double h[];
h=new double[n];
for(i=0;i<n;i++)
{
h[i]=0;
for(int j=0;j<n;j++)
T[i][j]=0;
}
h[0]=h0;
double r=0.5;
for( i=1;i<n;i++)
{
h[i]=h0*Math.pow(r,i);
}

for(i=0;i<n;i++)
{
x1[x_ref]+=h[i];
x2[x_ref]-=h[i];
f1=func(x1);
f2=func(x2);
T[i][0]=( f1 - f2)/(2.0*h[i]);
x1[x_ref]=x[x_ref];
x2[x_ref]=x[x_ref];
}
for(m=1;m<n;m++)
{
for(i=0;i<n-m;i++)
{
T[i][m]=(h[i]*h[i]*T[i+1][m-1] - h[i+m]*h[i+m]*T[i][m-1])/(h[i]*h[i]
- h[i+m]*h[i+m]);
}
}
double xx=T[0][n-1];
return xx;
}}

```

Program 9.3.13 örnek interface if_xj 'nin lambda deyimi olarak kullanımı

```

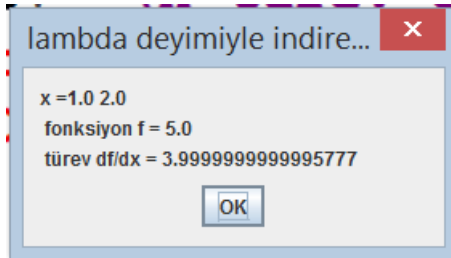
import javax.swing.*;
public class ifxj_Ex4 {
public static String array(double x[])
{ String s="";
for(int i=0;i<x.length;i++) {s+=x[i]+" ";}
return s;
}
}

```

```

public static void main(String args[])
{
if_xj ff=x->x[0]*x[0]+x[1]*x[1];
double x[]={1.0,2.0};
String s="x="+array(x)+"\n fonksiyon f = "+ff.func(x);
s+="\n türev df/dx = "+ff.dfunc(x,1);
String s1="lambda deyimiyle indirek referanslama:";
JOptionPane.showMessageDialog(null,s,s1,JOptionPane.PLAIN_MESSAGE);
}}

```



6.4 ALIŞTIRMALAR

1.) **abstract** sınıf **sekil** ve **nokta1**, **daire1** ve **silindir1** sınıflarını inceleyiniz. Bu sınıfların benzeri **koni1** sınıfını yaratınız ve bir örnek programda indirek referansla kullanınız.

	<p>Koninin yüzey alanı = $\pi r s$</p> <p>Koninin taban alanı = πr^2</p> <p>Koninin toplam alanı = $\pi r s + \pi r^2$</p> <p>Koninin hacmi = $\frac{1}{3} \pi r^2 h$</p> <p>$s = (r^2 + h^2)^{0.5}$</p>
--	--

2.) $f(x)=x*x-3*x+2$ fonksiyonunun türevini **abstract sınıf f_x**'i kullanarak $x=1.0$ noktasında hesaplayın.

3.) $f(x)=\sin(x)$ fonksiyonunun türevini **abstract sınıf f_x**'i kullanarak $x=\pi/4$ noktasında hesaplayın.

6.4 PROBLEMLER

1.) **abstract** **sekil** sınıfını kullanarak küre sınıfını yaratınız yüzey alanını ve hacmini hesaplattırınız. İndirek referansla test ediniz.

2.) **interface** **sekil1** kullanarak küre sınıfını yaratınız yüzey alanını ve hacmini hesaplattırınız. İndirek referansla test ediniz.

3.) x değişkenini ekrandan okuduktan sonra $y=e^{-x}/(x^2-1)$ fonksiyonunu ve türevini hesaplayacak ve sonuçları grafik ortamında yazdıracak bir program yazınız.

a) bir dış sınıf kullanınız

b) bir iç sınıf kullanınız

c) lamda deyimi kullanınız.

4.) a ve b integral sınırlarını ekrandan okuduktan sonra $y=e^{-x}/(x^2-1)$ fonksiyonun ve integralini hesaplayacak ve sonuçları grafik ortamında yazdıracak bir program yazınız.

a) bir dış sınıf kullanınız

b) bir iç sınıf kullanınız

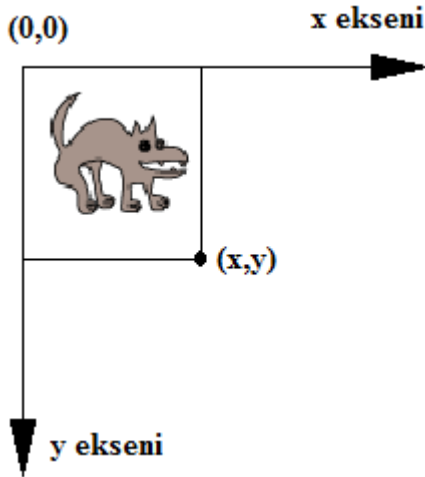
c) lambda deyimi kullanınız.

7 GRAFİK ÇİZİMİ

7.1 JAVA SWING GRAFİK ORTAMI JFRAME VE JAPPLET SINIFLARI

Java dilinde iki grafik ortamı tanımlanmıştır. Bunlar awt ve swing kütüphaneleridir. Burada grafik tanımlamalarımızı verirken daha sofistike olan swing sınıfı üzerinden gideceğiz, awt sınıfı da benzer olduğundan okuyucularımız bu bilgilenmeden sonra bu sınıfı da rahatlıkla kullanabilirler. Grafik çiziminde tamamen sınıflardan yararlanacağız. Bir çok programı hazır olarak java kütüphanelerinden alacağız, bir kısmını da burada temel kalıplar olarak kullanılmak üzere biz hazırlayarak sunacağız.

Java grafik sistemini anlamak için anlamamız gereken ilk kavram koordinat sistemidir. Koordinat birimi pixel (ekran görüntü elemanı) Digital bilgisayarda ekran görüntüsü bir çok pixelin bir arada kullanılmasıyla oluşur. Her bir pixel'in rengi, parlaklığı gibi çeşitli fiziksel özellikleri değiştirilerek ekranda veyakullanılan grafik ortamında görüntü oluşur. Java'da ve diğer bilgisayar grafik sistemlerinde piksel koordinatları karteziyen koordinat sistemi gibi düzenlenmiştir, ancak y eksenini aşağı doğru yönelmiştir. Bunun temel sebebi bilgisayar grafiklerinin ilk defa printerlar kullanılarak oluşturulmuş olması ve satır satır ileriye doğru giden printerlarda geriye doğru hareket imkanı bulunmamasıydı. Günümüzde böyle çalışma gereksinimi olmamasına rağmen , bu tarihi sebepten dolayı grafik sistemi bu şekilde standartlaştırılmıştır.



Şekil 7.1 Java grafik (pizel) koordinat sistemi

Grafik çiziminde temel olarak JFrame sınıfından yararlanılabilir. Temel grafik diğer bir sınıf olan JPanel sınıfında oluşturulacak ve JFrame sınıfına eklenecektir. Sizin için içine bir adet JPanel sınıfının eklenebileceği temel bir girdi JFrame programı olan FrameGrafik sınıfını oluşturduk. Elbette siz kendi JFrame programınızı oluşturarak çok daha kompleks grafik çıktıları JFrame sınıfından alabilirsiniz. Burada gayemiz mümkün olduğu kadar kolay bir şekilde grafik ortamına giriş yapmaktır. Sınıf bir String tipi ve bir JPanel tipi iki değişken girmemizi sağlayan bir kurucu metod ve grafik çıktı almamızı sağlayan statik tip plot sınıfından oluşmuştur. JPanel sınıfı nesnesi d, kurucu metod üzerinden sınıfta tanımlanmakta ve bu nesnenin gereği olan çizim plot sınıfı kullanılarak yapılmaktadır. Plot sınıfının içinde Frame grafik sınıfının Grafik ortamdaki tanım metodlarının çağrılması yer almaktadır. Örneğin programımızda grafik penceresi x=800 ve y=500 piksel olarak tanımlanmıştır. **f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);** komutuyla grafik penceresinin köşesindeki X ya basıldığında programın kapanacağı bildirilmektedir. **f.setVisible(true);** komutu ise grafiği göster komutudur.

Program 7.1.1 FrameGrafik grafik çizdirme sınıfı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
public class FrameGrafigi extends JFrame
{
    JPanel d;
    public FrameGraphic(String a,JPanel di)
    {
        super(a);
        d=di;
        add(d);
    }
    public static void plot(String a,JPanel di)
    {
        FrameGraphic f = new FrameGraphic(a,di);
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        f.setSize(800,500);
        f.setVisible(true);
    }
}
```

Bu sınıfta sınıf değişkeni olarak çağrılan JPanel sınıfı değişkenlerine ilk örnek olarak hoşgeldinizP sınıfı verelim. Bu sınıf Graphics sınıfından **public void paint(Graphics g)** metodunu çağırarak grafik ortamda (piksel kullanarak) çeşitli grafikleri çizdirir. Örnekte iki alt metod çağırılmıştır. Bunlardan biri yazı fontunu ayarlayan **g2.setFont(new Font("Serif",Font.ITALIC,24));** metodudur. Bu metotla serif tipi fontla (M.S. windows'daki Times-New roman yazı tipinin karşılığıdır.) italik karakter setiyle 24 piksel boyutunda karakterlerle yazı yazma istenmektedir. Java'nın şu anki versiyonunda mevcut bulunan fontlar

Tablo 7.1 Java Grafik fontları listesi

Agency FB	Century	FangSong
Aharoni	Century Gothic	Felix Titling
Algerian	Century Schoolbook	Footlight MT Light
Andalus	Chiller	Forte
Angsana New	Colonna MT	Franklin Gothic Book
AngsanaUPC	Comic Sans MS	Franklin Gothic Demi
Aparajita	Consolas	Franklin Gothic Demi Cond
Arabic Typesetting	Constantia	Franklin Gothic Heavy
Arial	Cooper Black	Franklin Gothic Medium
Arial Black	Copperplate Gothic Bold	Franklin Gothic Medium Cond
Arial Narrow	Copperplate Gothic Light	FrankRuehl
Arial Rounded MT Bold	Corbel	FreesiaUPC
Arial Unicode MS	Cordia New	Freestyle Script
Baskerville Old Face	CordiaUPC	French Script MT
Batang	Courier New	Gabriola
BatangChe	Curlz MT	Garamond
Bauhaus 93	DaunPenh	Gautami
Bell MT	David	Gentium Basic
Berlin Sans FB	DejaVu Sans	Gentium Book Basic
Berlin Sans FB Demi	DejaVu Sans Condensed	Georgia
Bernard MT Condensed	DejaVu Sans Light	Gigi
Blackadder ITC	DejaVu Sans Mono	Gill Sans MT
Bodoni MT	DejaVu Serif	Gill Sans MT Condensed
Bodoni MT Black	DejaVu Serif Condensed	Gill Sans MT Ext Condensed Bold
Bodoni MT Condensed	DFKai-SB	Gill Sans Ultra Bold
Bodoni MT Poster Compressed	Dialog	Gill Sans Ultra Bold Condensed

Book Antiqua	DialogInput	Gisha
Bookman Old Style	DilleniaUPC	Gloucester MT Extra Condensed
Bookshelf Symbol 7	DokChampa	Goudy Old Style
Bradley Hand ITC	Dotum	Goudy Stout
Britannic Bold	DotumChe	Gulim
Broadway	Ebrima	GulimChe
Browallia New	Edwardian Script ITC	Gungsuh
BrowalliaUPC	Elephant	GungsuhChe
Calibri	Engravers MT	Haettenschweiler
Californian FB	Eras Bold ITC	Harlow Solid Italic
Calisto MT	Eras Demi ITC	Harrington
Cambria	Eras Light ITC	High Tower Text
Cambria Math	Eras Medium ITC	JasmineUPC
Candara	Estrangelo Edessa	Jokerman
Castellar	EucrosiaUPC	Juice ITC
Centaur	Euphemia	KaiTi
Kalinga	Miriam	Rockwell
Kartika	Miriam Fixed	Rockwell Condensed
Khmer UI	Mistral	Rockwell Extra Bold
KodchiangUPC	Modern No. 20	Rod
Kokila	Mongolian Baiti	Sakkal Majalla
Kristen ITC	Monospaced	SansSerif
Kunstler Script	Monotype Corsiva	Script MT Bold
Lao UI	MoolBoran	Segoe Print
Latha	MS Gothic	Segoe Script
Leelawadee	MS Mincho	Segoe UI
Levenim MT	MS Outlook	Segoe UI Light
Liberation Sans Narrow	MS PGothic	Segoe UI Semibold
LilyUPC	MS PMincho	Segoe UI Symbol
Lucida Bright	MS Reference Sans Serif	Serif
Lucida Calligraphy	MS Reference Specialty	Shonar Bangla
Lucida Console	MS UI Gothic	Showcard Gothic
Lucida Fax	MT Extra	Shruti
Lucida Handwriting	MV Boli	SimHei
Lucida Sans	Narkisim	Simplified Arabic
Lucida Sans Typewriter	Niagara Engraved	Simplified Arabic Fixed
Lucida Sans Unicode	Niagara Solid	SimSun
Magneto	NSimSun	SimSun-ExtB
Maiandra GD	Nyala	Snap ITC
Malgun Gothic	OCR A Extended	Stencil
Mangal	Old English Text MT	Sylfaen
Marlett	Onyx	Symbol
Matura MT Script Capitals	OpenSymbol	Tahoma
Meiryo	Palace Script MT	Tempus Sans ITC
Meiryo UI	Palatino Linotype	Times New Roman
Microsoft Himalaya	Papyrus	Traditional Arabic
Microsoft JhengHei	Parchment	Trebuchet MS
Microsoft New Tai Lue	Perpetua	Tunga
Microsoft PhagsPa	Perpetua Titling MT	Tw Cen MT
Microsoft Sans Serif	Plantagenet Cherokee	Tw Cen MT Condensed
Microsoft Tai Le	Playbill	Tw Cen MT Condensed Extra Bold
Microsoft Uighur	PMingLiU	Utsaah
Microsoft YaHei	PMingLiU-ExtB	Vani
Microsoft Yi Baiti	Poor Richard	Verdana
MingLiU	Pristina	Vijaya
MingLiU-ExtB	Raavi	Viner Hand ITC
MingLiU_HKSCS	Rage Italic	Vladimir Script
MingLiU_HKSCS-ExtB	Ravie	Vrinda
Webdings	Wingdings 2	Imprint MT Shadow
Wide Latin	Wingdings 3	Informal Roman
Wingdings	Impact	IrisUPC

Deyimdeki ikinci terim olan **Font.ITALIC** yerine **Font.BOLD** ve **Font.PLAIN** kullanılabilir. Font türleri birlikte kullanılmak isteniyorsa :

```
int stil=Font.BOLD | Font.ITALIC;  
Font font=new Font( "Serif",stil,12);  
g2.setFont(font);
```

şeklinde de kullanılabilir. Buradaki sitil Kalın italik font tipidir.

Bir sonraki **g2.drawString("Java sınıfı grafik uygulamalarına hoş geldiniz " + isim,50,50);** komutuyla da x=50,y=50 koordinatlarına yazı (String) yazdırmaktayız .

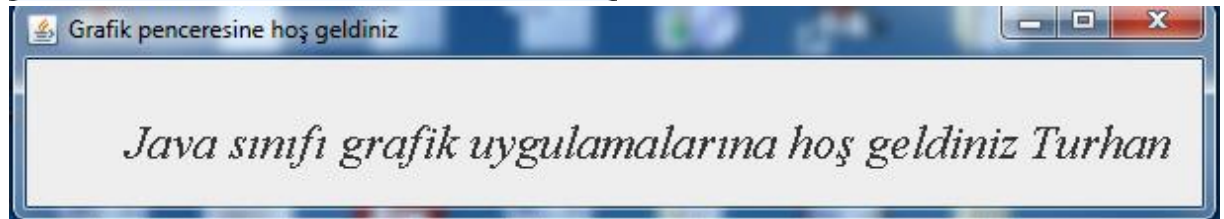
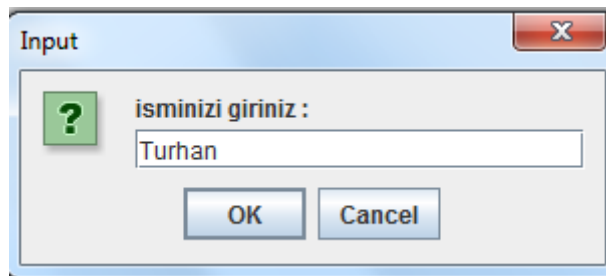
Program 7.1.2 hoşgeldinizP JPanel sınıfı hoşgeldiniz mesajını grafik ekrana çizer.

```
import javax.swing.*;  
import java.awt.Graphics;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
  
public class hoşgeldinizP extends JPanel  
{  
    String isim;  
    public hoşgeldinizP()  
    { isim=JOptionPane.showInputDialog("isminizi giriniz : ");}  
    public void paint(Graphics g)  
    {  
        Graphics2D g2=(Graphics2D)g;  
        g2.setFont(new Font("Serif",Font.ITALIC,24));  
        g2.drawString("Java sınıfı grafik uygulamalarına hoş geldiniz " + isim,50,50);  
    }  
}
```

Bu programı FrameGrafıgi programının girdisi olarak kullanarak bir test programında çağırabiliriz.

Program 7.1.3 An example program to get graphic output WelcomeP JPanel class

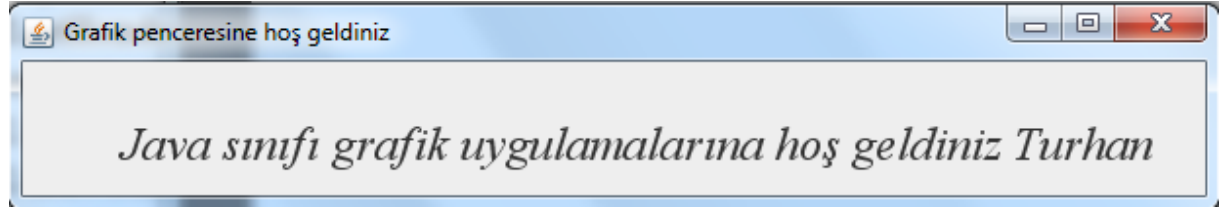
```
class hoşgeldinizTest {  
    public static void main(String args[])  
    {FrameGraphic.plot("Grafik penceresine hoş geldiniz",new hoşgeldinizP());}}
```



Çıktıdan gördüğümüz gibi harfler yazılırken oluştuğu noktalar belirgin bir şekilde belli oluyor. Yazımızın (veya herhangi bir çizimin) daha düzgün bir şekilde çıkmasını istersek görüntü düzeltme alt metodunu grafik çizim işleminin parçası olarak çağırabiliriz.

```
public void paint(Graphics g)
{
    Graphics2D g2=(Graphics2D)g;
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setFont(new Font("Serif",Font.ITALIC,24));
    g2.drawString("Java sınıfı grafik uygulamalarına hoş geldiniz " + isim,50,50);
}
```

Bu durumda çıktı çok daha düzgün karakterlerle oluşacaktır. Bu komutu her zaman için çizim programımızın bir parçası olarak kullanabiliriz.



Şimdi de değişik fontların kullanılmasını yeni bir örnekle gösterelim:

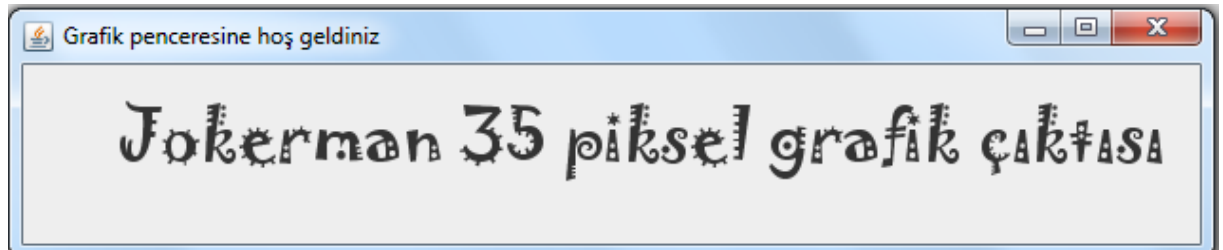
Program 7.1.4 hosgeldinizP1 JPanel sınıfı grafik ekrana Jokerman (Şakacı) fontuyla yazı yazdırır.

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class hosgeldinizP1 extends JPanel
{
    public void paint(Graphics g)
    {
        Graphics2D g2=(Graphics2D)g;
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
        Font font = new Font("Jokerman", Font.PLAIN, 35);
        g2.setFont(font);
        g2.drawString("Jokerman 35 piksel grafik çıktısı",50,50);
    }
}
```

Program 7.1.5 sınıfı test programı

```
class hosgeldinizTest1 {
    public static void main(String args[])
    {FrameGrafigi.plot("Grafik penceresine hoş geldiniz",new hosgeldinizP1());}
```



İkinci JPanel programı olarak ekranda resim gösteren bir program oluşturalım.

Program 7.1.6 resimP JPanel sınıfı bir resmi grafik ekranda gösterir.

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;
import java.net.URL;
import java.awt.image.*;
import java.util.*;

public class resimP extends JPanel
{
    private Image resim;

    public resimP(String s)
    {super();
      URL url = getClass().getResource(s);
      resim = getToolkit().getImage(url);
    }

    public void paint(Graphics g)
    {
      Graphics2D g2=(Graphics2D)g;
      Dimension d=getSize();
      int dx = d.width;
      int dy = d.height;
      g2.drawImage( resim, 0, 0,dx,dy, this);
    }
}
```



Bu resmin okunabilmesi için resim dosyası URL sınıfında **URL url = getClass().getResource(s);** Komutuyla okunduktan sonra Image sınıfından resim nesnesine yüklendi.

Image resim = getToolkit().getImage(url);. Paint programında resmi göstermek için de **g2.drawImage(resim, 0, 0,dx,dy, this);** metodunu kullandık. Bu metod pencerenin boyutunu Dimension sınıfının width ve height sınıf değişkenlerinden kendisi saptadı.

```
Dimension d=getSize();  
int dx = d.width;  
int dy = d.height;  
g2.drawImage( resim, 0, 0,dx,dy, this);
```

Program 7.1.7 resimP JPanel sınıfı test programı

```
class resimTesti {  
public static void main(String args[])  
{  
resimP pp=new resimP("M101_galaksisi.jpg");  
FrameGrafigi.plot("M101 Galaksisi 170000 ışık yılı boyutunda, bizden 25 milyon ışık yılı uzaklıktadır ",pp);  
}  
}
```

Çizgi, daire, dikdörtgen gibi değişik şekilleri de benzer yöntemlerle çizebiliriz. Çizgi çizmek için Line2D sınıfını ve Graphics kütüphanesinde draw metodunu kullanacağız. Bu örnek programda arka plan ve çizgi için renk seçmeyi de gösterdik. Renk seçimi için **g2.setColor(Color.red);** metodunu Color (renk) sınıfından Color.red statik değişkenini çağırarak oluşturduk. Burada ayrıca **g2.setStroke(new BasicStroke(5.0f));** metodu kullanılarak çizgi kalınlığının 5 pixel olması sağlanmıştır.

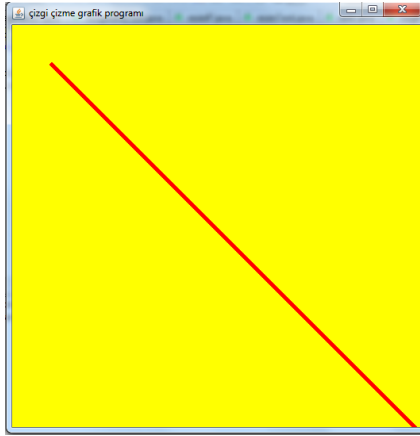
Program 7.1.8 çizgiP JPanel class to draw a line in the graphic window

```
import javax.swing.*;  
import java.awt.Graphics;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
  
public class çizgiP extends JPanel  
{  
public void paintComponent(Graphics g)  
{  
super.paintComponent(g);  
Graphics2D g2=(Graphics2D)g;  
g2.setFont(new Font("Serif",Font.BOLD,24));  
this.setBackground(Color.YELLOW);  
g2.setColor(Color.red);  
g2.setStroke(new BasicStroke(5.0f));  
Line2D x=new Line2D.Double(50,50,800,800);  
g2.draw(x);  
}}}
```

Program 7.1.9 example program to draw line by using lineP JPanel class

```
class çizgiTest {  
public static void main(String args[])  
{  
çizgiP pp=new çizgiP();  
FrameGrafigi.plot("çizgi çizme grafik programı",pp);  
}  
}
```

```
}
```



Yeri gelmişken Color sınıfına biraz daha detaylı bakalım. Grafik programlamada renk kontrolü **Color** sınıfını kullanarak yapılır. Renk oluşturmada monitör ve televizyonda da kullanılan **RGB (Red-Green-Blue=Kırmızı-Yesil-Mavi)** sistemi temel alınır. Bu üç renk monitörde 0-255 arası değerler alabilir. Üç rengin karışımı toplam $256*256*256$ renk tanımlar. Color sınıfı kurucu metodları şu şekilde tanımlanır.

```
public Color(int kırmızı,int yeşil,int mavi) //her renk 0-255 arası  
public Color(float kırmızı,float yeşil,float mavi) //her renk 0.0-1.0 arası  
Color sınıfında ayrıca renk değerlerini okuyabildiğimiz  
public int getRed() // Kırmızı tonunu oku  
public int getGreen()//Yeşil tonunu oku  
public int getBlue()//Mavi tonunu oku  
public abstract Color getColor() //rengi oku  
metotları ve rengi değiştirebildiğimiz  
public abstract Color setColor(Color c)  
metodu mevcuttur. Color sınıfında sabit olarak tanımlanmış :
```

Tablo 7.2 Color sınıfında tanımlanmış sabit renkler:

Renk sabiti	Renk	RGB değeri
Public final static Color orange Public final static Color ORANGE	portakal	255,200,0
Public final static Color pink Public final static Color PINK	Pembe	255,175,175
public final static Color cyan public final static Color CYAN	camgöbeği	0,255,255
public final static Color magenta public final static Color MAGENTA	Eflatun	255,0,255
public final static Color yellow public final static Color YELLOW	Sarı	255,255,0
public final static Color black public final static Color BLACK	Siyah	0,0,0
public final static Color white public final static Color WHITE	Beyaz	255,255,255
public final static Color gray public final static Color GRAY	Gri	128,128,128
public final static Color lightGray public final static Color LIGHT_GRAY	Açık gri	192,192,192
public final static Color darkGray	Koyu Gri	64,64,64

public final static Color DARK_GREY		
public final static Color red public final static Color RED	kırmızı	255,0,0
public final static Color green public final static Color GREEN	Yeşil	0,255,0
public final static Color blue public final static Color BLUE	mavi	0,0,255

Color sınıfının daha iyi anlaşılabilmesi amacıyla içinde Türkçe renk sabitlerinin yer aldığı Color sınıfının bir alt sınıfı olan **renk** sınıfını tanımlayabiliriz.

Program 7.1.10 Color sınıfından türetilmiş Türkçe renk tanımlarını veren renk sınıfı

```
import java.io.*;
import java.awt.*;

public class renk extends Color
{
    public final static renk kırmızı=new renk(255,0,0);
    public final static renk mavi=new renk(0,0,255);
    public final static renk siyah=new renk(0,0,0);
    public final static renk camgöbeği=new renk(0,255,255);
    public final static renk açık_camgöbeği=new renk(224,255,255);
    public final static renk koyugri=new renk(64,64,64);
    public final static renk gri=new renk(128,138,128);
    public final static renk yeşil=new renk(0,255,0);
    public final static renk açikgri=new renk(192,192,192);
    public final static renk portakal=new renk(255,200,0);
    public final static renk pembe=new renk(255,175,175);
    public final static renk beyaz=new renk(255,255,255);
    public final static renk sarı=new renk(255,255,0);
    public final static renk açikmavi=new renk(150,206,237);
    public final static renk lacivert=new renk(0,0,128);
    public final static renk mor=new renk(160,32,240);
    public final static renk turkuaz=new renk(48,213,200);
    public final static renk menekşe=new renk(238,130,238);
    public final static renk mavi_menekşe=new renk(138,43,226);
    public final static renk patlıcan=new renk(153,17,153);
    public final static renk buğday=new renk(238,130,238);
    public final static renk ten=new renk(210,180,140);
    public final static renk gök_mavi=new renk(135,206,235);
    public final static renk açık_gök_mavisi=new renk(135,206,250);
    public final static renk deniz_kabuğu=new renk(255,245,238);
    public final static renk deniz_yeşili=new renk(46,138,87);
    public final static renk açık_deniz_yeşili=new renk(32,178,170);
    public final static renk zeytin_yeşili=new renk(107,142,35);
    public final static renk fildişi=new renk(255,255,240);
    public final static renk lavanta=new renk(181,126,220);
    public final static renk bej=new renk(245,245,220);
    public final static renk kahverengi=new renk(165,42,42);
    public final static renk çikolata=new renk(210,105,30);
    public final static renk mercan=new renk(255,127,80);
    public final static renk açık_yeşil=new renk(102,255,0);
    public final static renk alev_kırmızısı=new renk(255,77,0);
    public final static renk alev_turuncusu=new renk(255,102,0);
    public final static renk alizarin=new renk(227,38,54);
    public final static renk altuni=new renk(255,215,0);
    public final static renk ametist=new renk(153,102,204);
    public final static renk armut=new renk(209,226,49);
}
```

```

public final static renk akuamarin=new renk(127,255,212);
public final static renk asker_yeşili=new renk(75,83,32);
public final static renk bakır=new renk(184,115,51);
public final static renk bordo=new renk(128,0,0);
public final static renk bronz=new renk(205,127,50);
public final static renk çivit_mavisi=new renk(75,0,130);
public final static renk gece_mavisi=new renk(0,51,102);
public final static renk gül=new renk(255,0,127);
public final static renk hardal=new renk(255,219,88);
public final static renk havuç=new renk(237,145,33);
public final static renk ilkbahar_yeşili=new renk(0,255,127);
public final static renk kabak=new renk(255,117,24);
public final static renk karanfil=new renk(249,90,97);
public final static renk kayısı=new renk(251,206,177);
public final static renk kehribar=new renk(255,191,0);
public final static renk kobalt_mavisi=new renk(0,71,171);
public final static renk krem=new renk(255,253,208);
public final static renk limon=new renk(253,233,16);
public final static renk açık_limon=new renk(255,250,205);
public final static renk leylak=new renk(200,162,200);
public final static renk mandalina=new renk(255,204,0);
public final static renk nane_yeşili=new renk(152,255,152);
public final static renk nar=new renk(243,71,35);
public final static renk orkide=new renk(218,112,214);
public final static renk pas=new renk(183,65,14);
public final static renk safran=new renk(244,196,48);
public final static renk safir=new renk(9,37,103);
public final static renk tarçın=new renk(123,63,0);
public final static renk turuncu=new renk(255,165,0);
public final static renk zümrüt_yeşili=new renk(80,200,120);
public final static renk şarap=new renk(150,0,24);

```

//burada kendi yeni renginizi tanımlayabilirsiniz.

```

public renk(float kirmizi,float yesil, float mavi)
{ super(kirmizi,yesil,mavi);}

```

```

public renk(double kirmizi,double yesil, double mavi)

```

```

{ super((float)kirmizi,(float)yesil,(float)mavi);}

```

```

public renk(int kirmizi,int yesil, int mavi)

```

```

{ super(kirmizi,yesil,mavi);}

```

```

public renk(int RGB)

```

```

{ super( RGB);}

```

```

public renk(renk r)

```

```

{ super(r.kirmiziOku(),r.yesilOku(),r.maviOku());}

```

```

public int kirmiziOku()

```

```

{ return super.getRed();}

```

```

public int yesilOku()

```

```

{ return super.getGreen();}

```

```

public int maviOku()

```

```

{ return super.getBlue();}

```

```

public int KYMOku()

```

```

{ return super.getRGB();}

```

```

public renk renkOku()

```

```

{ return (renk)this;}

```

```

public static renk renkOku(Color r)

```

```

{ return (renk)r;}

```

```

public String cikti()

```














































```




























































{return "renk[ kirmizi "+kirmiziOku()+ " mavi "+maviOku()+ " yesil "+yesilOku()+"]";}
//metodlar
//static int HSBtoRGB(float hue,float saturation,float brightness)
//getRed(),getGreen(),getBlue(),getRGB()
}



























































```






















































Bu sınıfı yarattıktan sonra Color sınıfı yerine renk sınıfını kullanabiliriz. Renk sınıfı Color sınıfından türetildiği için programda onun yerine geçebilir. Aynı zamanda renk sınıfının içinde direk olarak tanımlanmamış renkleri ve RGB kodlarını tanımlamak için Türkçe bir renk tablosu Tablo 7.3 de verilmiştir.

Tablo 7.3 Türkçe renkler ve renklerin RGB kodları

Ad	Örnek	RGB		
Açık galibarda		255	119	255
Açık mavi		173	216	230
Açık mor		224	176	255
Açık turkuaz		8	232	222
Açık yeşil		102	255	0
Açık yeşil		127	255	0
Açık yeşil sarı		223	255	0
Alev kırmızısı		255	77	0
Alev turuncusu		255	102	0
Alice mavisi		240	248	255
Alizarin rengi		227	38	54
Altunî		255	215	0
Ametist rengi		153	102	204
Armut rengi		209	226	49
Akuamarin rengi		127	255	212
Asker yeşili		75	83	32
Bakır rengi		184	115	51
Barut rengi		61	43	31
Bataklık yeşili		172	183	142
Bebek mavisi		224	255	255
Bej		245	245	220
Beyaz		255	255	255
Bondi mavisi		0	149	182
Bordo		128	0	0
Bronz rengi		205	127	50
Buğday rengi		245	222	179
Burgonya		128	0	32
Camgöbeği		0	255	255
Camgöbeği mavisi		21	96	189
Çam yeşili		1	121	111
Çay yeşili		208	240	192
Çelik mavisi		70	130	180
Çokolata rengi		210	105	30
Çivit rengi		75	0	130
Deniz mavisi		0	255	255
Deniz yeşili		46	139	87
Devedikeni rengi		216	191	216
Eğrelti yeşili		79	121	66
Elektrik mavisi		125	249	255
Elektrik yeşili (X11 yeşil)		0	255	0
Elektrik çivit rengi		102	0	255
Elektrik lime rengi		204	255	0
Elektrik mor rengi		191	0	255
Falu kırmızısı		128	24	24
Fildişi rengi		255	255	240

Fransız gül rengi		246	74	138
Galibarda (parlak mor)		255	0	255
Gece mavisi ^[1]		0	51	102
Gök mavisi		0	127	255
Gri		128	128	128
Gri-kuşkonmaz		70	89	69
Gül rengi		255	0	127
Gümüşi		192	192	192
Haki		195	176	145
Haki (X11)		240	230	140
Hardal rengi		255	219	88
Havuç rengi		237	145	33
Horozibiği rengi		229	43	80
İlkbahar yeşili		0	255	127
İslam yeşili		0	153	0
Kabak rengi		255	117	24
Kahverengi		150	75	0
Kahverengimsi gri		72	60	50
Kamuflaj yeşili		120	134	107
Karanfil pembesi		255	166	201
Karanfil rengi		249	90	97
Kardinal rengi		196	30	58
Karolina mavisi		156	186	227
Kayısi rengi		251	206	177
Kehribar rengi		255	191	0
Kestane rengi		111	53	26
Keten rengi		250	240	230
Kırmızı		255	0	0
Kırmızımsı kahverengi		233	116	81
Kırmızı-menekşe		199	21	133
Kiraz kırmızısı		222	49	99
Kobalt rengi		0	71	171
Kobalt mavisi		0	71	171
Koyu galibarda		193	84	193
Koyu haki		189	183	107
Koyu kahverengi		101	67	33
Koyu kestane		152	105	96
Koyu kırmızı		220	20	60
Koyu kızıl kahverengi		128	0	0
Koyu leylak		153	85	187
Koyu magenta		204	0	204
Koyu mandalina		255	168	18
Koyu mavi		0	0	139
Koyu menekşe		66	49	137
Koyu mercan		205	91	69
Koyu mor		204	136	153
Koyu pastel yeşil		3	192	60
Koyu pembe		231	84	128
Koyu şeftali		255	203	164
Koyu toz mavi)		0	51	153
Koyu turkuaz		17	96	98
Koyu yeşil		1	50	32
Kösele rengi		240	220	130
Krem		255	253	208
Kum kahverengisi		244	164	96
Kuşkonmaz rengi		123	160	91
Lacivert		0	0	128
Lavanta rengi		181	126	220
Lavanta mavisi		204	204	255

Lavender pembesi		255	240	245
Lavender greisi		189	187	215
Lavender magenta		238	130	238
Lavanta pembesi		251	174	210
Lavanta mor		150	120	182
Lavanta gül rengi		251	160	227
Limoni		253	233	16
Açık Limon		255	250	205
Leylak rengi		200	162	200
Lime rengi		191	255	0
Mandalina rengi		255	204	0
Malakit rengi		11	218	81
Mavi		0	0	255
Menekşe rengi		139	0	255
Menekşe rengi (web)		238	130	238
Menekşe-patlıcan rengi		153	17	153
Mısır rengi		251	236	93
Mor		102	0	153
Morsalkım rengi		201	160	220
Nane yeşili		152	255	152
Nar rengi		243	71	35
Navajo beyazı		255	222	173
Okul otobüsü sarısı		255	216	0
Orkide rengi		218	112	214
Orman yeşili		34	139	34
Parlak mor (galibarda)		255	0	255
Pas rengi		183	65	14
Pastel pembe		255	209	220
Pastel yeşili		119	221	119
Patlıcan rengi		153	0	102
Pembe		255	192	203
Pembe-turuncu		255	153	102
Peygamber çiçeği rengi		100	149	237
Prusya mavisi		0	49	83
Safran rengi		244	196	48
Safir rengi		8	37	103
Sarımsı kahverengi		210	180	140
Sarımsı pembe ^[1]		255	140	105
Sarı		255	255	0
Siyah		0	0	0
Siyahımsı koyu kahverengi		112	66	20
Soluk sarı		240	220	130
Şeftali rengi		255	229	180
Şeftali-turuncu		255	204	153
Şeftali-sarı		250	223	173
Tarçın rengi		123	63	0
Teal		0	128	128
Toz mavi		176	224	230
Turkuaz		48	213	200
Turuncu (web)		255	165	0
Turuncumsu sarı		228	155	15
Yeşil		0	255	0
Yeşil-sarı		173	255	47
Yonca yeşili		0	158	96
Yosun yeşili		173	223	173
Zeytuni		128	128	0
Zümrüt yeşili		80	200	120
Yanık turuncu		204	85	0
Yanık Toprak rengi		138	51	36

Kardinal		196	30	58
Şarap rengi		150	0	24
Celadon		172	225	175
Berrak mavi		0	123	167
Gök mavisi		42	82	190
Gül rengi		153	102	102
Mercan		255	127	80
Mercan Kırmızısı		255	64	64
Kıpkırmızı		220	20	60
Hile mavisi		30	144	255
Altınmsı		218	165	32
Soytarı		63	255	0
Siğil otu		223	115	255
Holivod kırmızısı		244	0	161
Sıcak Magenta		255	0	204
Sıcak pembe		255	105	180
Uluslararası Klein mavisi		0	47	167
Enternasyonal turuncu		255	79	0
Yeşim		0	168	107
Orta şarap		175	64	53
Orta Mor		147	112	219
Dağ pembesi		153	122	141
Aşı boyası		204	119	34
Eski altın		207	181	59
Eski iplik		253	245	230
Eski Lavanta		121	104	120
Eski gül		192	46	76
Zeytin Kahverengisi		107	142	35
Turuncu		255	127	0
Donuk turuncu		255	160	0
Papaya		255	239	213
Periwinkle		204	204	255
Pers mavisi		28	57	187
Pers yeşili		0	166	147
Persian lacivert		50	18	122
Pers pembesi		247	127	190
Persian kırmızısı		204	51	51
Pers gülü		254	40	162
Ham toprak		115	74	18
Kızıl yumurta mavisi		0	204	204
Kraliyet mavisi		65	105	225
Kırmızı şarap		146	0	10
Kırmızı		255	36	0
Deniz kabuğu		255	245	238
Ayrık sarı		255	186	0
Vurgun pembe		252	15	192
Salamura grisi		112	128	144
Tenné (Tawny) ^[1]		205	87	0
Küçük kara		226	114	91
Lacivert		18	10	143
Viridian		64	130	109
Zinnwaldite		235	194	175
				

Program 7.1.12 Color sınıfından türetilmiş Türkçe renk tanımlarını veren renk sınıfını kullanan çizgi çizdirme programı cizgiP1

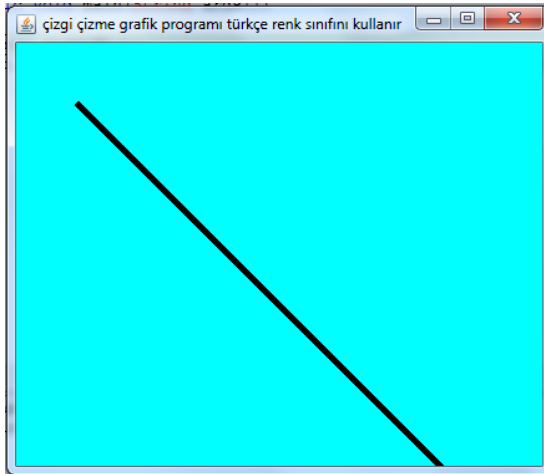
```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class cizgiP1 extends JPanel
{

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        g2.setFont(new Font("Serif",Font.BOLD,24));
        this.setBackground(renk.camgöbeği);
        g2.setColor(renk.siyah);
        g2.setStroke(new BasicStroke(5.0f));
        Line2D x=new Line2D.Double(50,50,800,800);
        g2.draw(x);
    }
}
```

Program 7.1.13 Color sınıfından türetilmiş Türkçe renk tanımlarını veren renk sınıfını kullanan çizgi çizdirme programı cizgiP1 test programı

```
class cizgiTest1 {
    public static void main(String args[])
    {
        cizgiP1 pp=new cizgiP1();
        FrameGrafigi.plot("çizgi çizme grafik programı türkçe renk sınıfını kullanır",pp);
    }
}
```



Şimdi aynı şekilde bir dikdörtgen çizdirmeyi deneyelim. Dikdörtgen çizimi, doğru çizimine çok benzer. Başlangıç noktası(x ve y piksel değerleri) ve dikdörtgenin eni ve yüksekliği tanımlanır. Aynı doğru çizilirken olduğu gibi draw metodu kullanılarak çizilir.

```
Rectangle2D R=new Rectangle2D.Double(x,y,en,yükseklik);  
g2.draw(R);
```

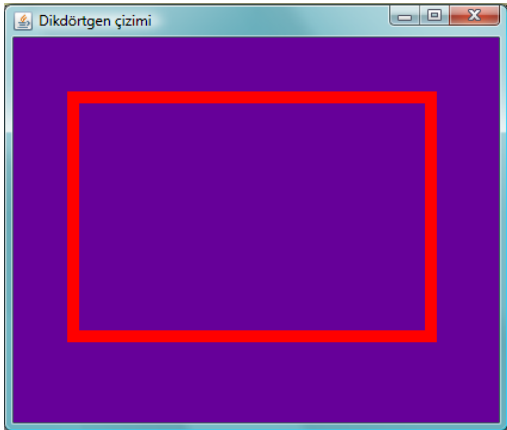
Program 7.1.14 dikdortgenP JPanel sınıfı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class dikdortgenP extends JPanel
{
    public void paint(Graphics g)
    {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        g2.setFont(new Font("Serif",Font.BOLD,24));
        g2.setColor(renk.kırmızı);
        this.setBackground(renk.mor);
        g2.setStroke(new BasicStroke(10.0f));
        Rectangle2D x=new Rectangle2D.Double(50,50,300,200);
        g2.fill(x);
    }
}
```

Program 7.1.15 dikdortgenTest örnek programı

```
class dikdortgenTest {
    public static void main(String args[])
    {
        dikdortgenP pp=new dikdortgenP();
        FrameGrafıgi.plot("Dikdörtgen çizimi",pp);
    }
}
```



Eğer şeklin içini tamamen boyamak istersek **g2.draw(x)**; metodu yerine **g2.fill(x)** metodu kullanılır.

Program 7.1.16 rectangleP1 JPanel class to draw a filled rectangle in the graphic window

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class dikdortgenP1 extends JPanel
{
```

```

public void paint(Graphics g)
{
super.paintComponent(g);
Graphics2D g2=(Graphics2D)g;
g2.setFont(new Font("Serif",Font.BOLD,24));
g2.setColor(renk.portakal);
this.setBackground(renk.açık_deniz_yeşili);
g2.setStroke(new BasicStroke(10.0f));
Rectangle2D x=new Rectangle2D.Double(50,50,300,200);
g2.fill(x);
}
}

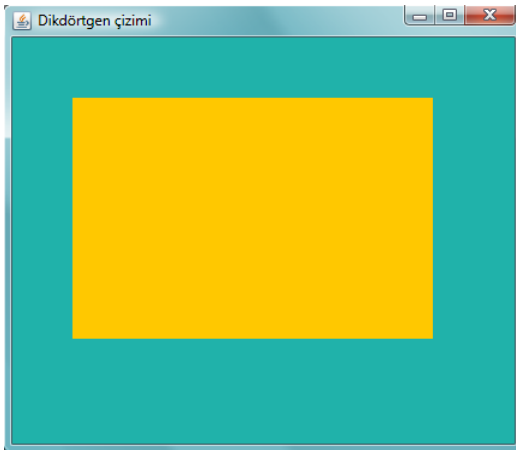
```

Program 7.1.17 example program to draw rectangle by using rectangleP JPanel class

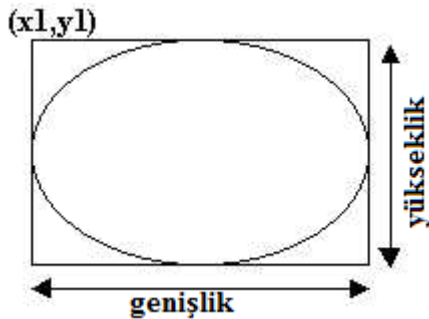
```

class dikdortgenTest1 {
public static void main(String args[])
{
dikdortgenP1 pp=new dikdortgenP1();
FrameGrafigi.plot("Dikdörtgen çizimi",pp);
}
}

```



Şimdi de elips çizimine göz atalım. Elips çizimi temel olarak dikdörtgen çizimiyle aynıdır. Dikdörtgenin içine bir elips yerleştirdiğimizi varsayabiliriz.



Program 7.1.18 elipsP1 JPanel elips çizme programı

```

import javax.swing.*.*;
import java.awt.Graphics;
import java.awt.*.*;
import java.awt.event.*.*;

```

```

import java.awt.geom.*;

public class elipsP extends JPanel
{
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        g2.setFont(new Font("Serif",Font.BOLD,24));
        g2.setColor(Color.RED);
        this.setBackground(new Color(0,0,255));
        g2.setStroke(new BasicStroke(4.0f));
        Ellipse2D x=new Ellipse2D.Double(50,50,500,200);
        g2.draw(x);
    }
}

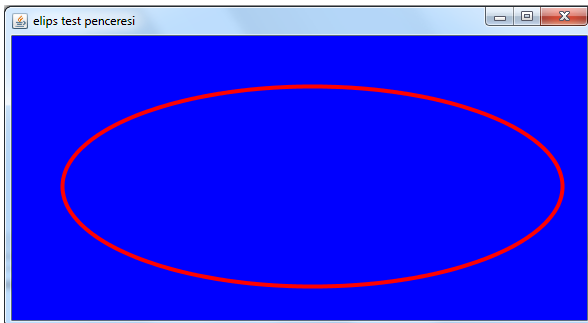
```

Program 7.1.19 elips çizme örneği

```

class elipsTest {
public static void main(String args[])
{
    elipsP pp=new elipsP();
    FrameGraphic.plot("elips test penceresi",pp);
}
}

```



Bir çizimde çizim çizgi karakterleri setStroke metodu kullanılarak değiştirilebilir.

Program 7.1.20 elipsP1 JPanel class to draw an ellipse in the graphic window

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class elipsP1 extends JPanel
{
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        g2.setFont(new Font("Serif",Font.BOLD,24));
        g2.setColor(renk.çivit_mavisi);
    }
}

```

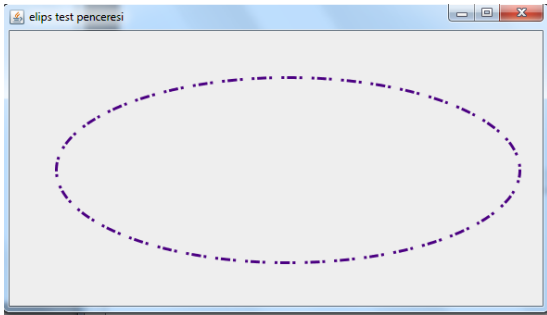
```

g2.setStroke(new BasicStroke(4.0f));
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
float dash3[] = {10.0f,3.0f,3.0f};
BasicStroke d3 = new BasicStroke(3.0f,BasicStroke.CAP_BUTT,
BasicStroke.JOIN_MITER,
3.0f, dash3, 2.0f);

g2.setStroke(d3);
Ellipse2D x=new Ellipse2D.Double(50,50,500,200);
g2.draw(x);
}
}

```

Çıktı görünümü:

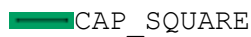
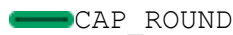
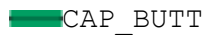


Bu programda noktalı bir eğri çizme formu oluşturduk. Bunu yapmak için de setStroke metodunu kullandık. Bu metodun içinde de BasicStroke metodunda tanımlanmış bir nesne yer aldı. Bu sınıf çizginin kalınlığı, çizgi parçacıklarının birbiriyle birleşme şekli, çizgilerin uçlarının bitiş şekli gibi bilgileri içinde taşır. Normal olarak çizgi kalınlığı 1f, 2f gibi bir bilgiyle iletilir. Burada 1f 0.35277 mm kalınlığında bir çizgi oluşturur, ancak şekilde boyutsal transformasyon yapıldığında bu değişebilir.

İki çizginin birleşme değişkeninin değerleri 3 ayrı tipte belirlenebilir:



stilleri. Çizgi bitiş stilleri olarak da yine 3 ayrı stil belirlenmiştir.



Ara boşlukların tanımını bir örnekle vermeye çalışalım

```

g2.setStroke(new BasicStroke(1f,BasicStroke.CAP_ROUND,

```



```
BasicStroke.JOIN_ROUND,0,new float[] { 10,5,2,5},0));
```

Tanımı 1f kalınlığında 10 pixel uzunluğunda çizgiyi 5 piksellik boşluk, sonra 2 piksellik çizgi ve 5 piksellik boşluk şeklinde oluşturulmuş bir çizgi ifade eder. Çizgi uçları yuvarlaktır. Çizgi bağlantıları da yuvarlaktır.

```
g2.setStroke(new BasicStroke(2f,BasicStroke.CAP_ROUND,  
BasicStroke.JOIN_ROUND,5,new float[] { 5},0));
```

Tanımı 2f kalınlığında 5 pixel çizgi ve 5 piksel boşluk ifade eder

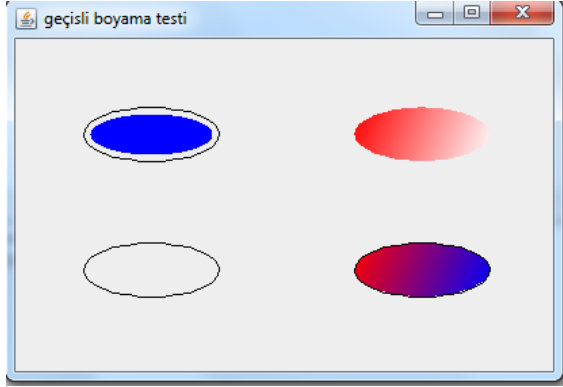
Java swing grafik türünde diğer bir ilginç özellik gradient paint dediğimiz geçişli renk oluşumudur.

Program 7.1.21 graph2P JPanel class to draw an ellipses with gradient paint in the graphic window

```
import javax.swing.*;  
import java.awt.Graphics;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
public class graph2P extends JPanel  
{  
    public void paintComponent(Graphics g)  
    {  
        super.paintComponent(g);  
        Graphics2D g2 = (Graphics2D)g;  
        g2.setPaint(Color.blue);  
        Ellipse2D elips1=new Ellipse2D.Double(55,55,90,30);  
        g2.fill(elips1);  
        g2.setPaint(Color.black);  
        Ellipse2D elips2=new Ellipse2D.Double(50,50,100,40);  
        g2.draw(elips2);  
        g2.setPaint(Color.black);  
        Ellipse2D elips3=new Ellipse2D.Double(50,150,100,40);  
        g2.draw(elips3);  
        GradientPaint kirmizidanbeyaza=new GradientPaint(250,50,Color.red,350,90,Color.white);  
        g2.setPaint(kirmizidanbeyaza);  
        Ellipse2D elips4=new Ellipse2D.Double(250,50,100,40);  
        g2.fill(elips4);  
        GradientPaint kirmizidanmaviye=new GradientPaint(250,150,Color.red,350,190,Color.blue);  
        g2.setPaint(kirmizidanmaviye);  
        Ellipse2D elips5=new Ellipse2D.Double(250,150,100,40);  
        g2.fill(elips5);  
        g2.setPaint(Color.black);  
        g2.draw(elips5);  
    }  
}
```

Program 7.1.22 gradientpaint kullanarak elips çizen örnek program

```
class graph2PTest {  
    public static void main(String args[])  
    {  
        graph2P pp=new graph2P();  
        FrameGraphic.plot("geçişli boyama testi",pp);  
    }  
}
```



Diğer bir ilginç özellikte TexturePaint dediğimiz bir şeklin içini verilen bir resimle doldurma işlemidir.

Program 7.1.22 elipsP2 TexturePaint'le elipsin içini boyayan program

```
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;
import java.net.URL;
import java.awt.image.*;
import java.util.*;

public class elipsP2 extends JPanel
{
    TexturePaint tp = getImageTexture("M101_galaksisi.jpg");

    public TexturePaint getImageTexture(String imageFile)
    {
        URL url = getClass().getResource(imageFile);
        Image img = getToolkit().getImage(url);
        try {
            MediaTracker tracker = new MediaTracker(this);
            tracker.addImage(img, 0);
            tracker.waitForID(0);
        } catch (Exception e) {}
        int width = img.getWidth(this);
        int height = img.getHeight(this);
        BufferedImage buffImg = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
        Graphics g = buffImg.getGraphics();
        g.drawImage(img, 0, 0, this);
        Rectangle2D rect = new Rectangle(0, 0, width, height);
        return new TexturePaint(buffImg, rect);
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        g2.setPaint(tp);
        g2.setFont(new Font("Serif",Font.BOLD,24));
        //g2.setColor(Color.RED);
        //this.setBackground(new Color(255,255,255));
    }
}
```

```
Ellipse2D x=new Ellipse2D.Double(100,225,800,500);
g2.fill(x);
}
}
```

Program 7.1.23 elipsP2 TexturePaint'le elipsin içini boyayan test programı

```
class elipsTest2 {
public static void main(String args[])
{
elipsP2 pp=new elipsP2();
FrameGraphic.plot("elips test penceresi",pp);
}
}
```



Tek bir çizgi yerine bir çizgi ailesini birlikte çizeceksek bu gaye için GeneralPath metodundan yararlanabiliriz.

Program 7.1.24 yıldızP GeneralPath kullanarak yıldız çizen bir program

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class yıldızP extends JPanel
{
public void paint(Graphics g)
{
int x[]={55,67,109,73,83,55,27,37,1,43};
int y[]={0,36,36,54,96,72,96,54,36,36};
Graphics2D g2=(Graphics2D)g;
GeneralPath yıldız=new GeneralPath();
yıldız.moveTo(x[0],y[0]);
for(int i=1;i<x.length;i++)
{yıldız.lineTo(x[i],y[i]);}
yıldız.closePath();
g2.setColor(Color.blue);
g2.draw(yıldız);
}}
}
```

Program 7.1.25 yıldızTest yıldızP test programı

```
class yıldızTest {
public static void main(String args[])
```

```
{
yildizP pp=new yildizP();
FrameGraphic.plot("yıldız testi",pp);
}
}
```



Doğru yerine ikinci derece veya kübik polinomlarla da noktaların birleştirilmesi mümkündür. QuadCurve2D ikinci dereceden polinom eğrilerle noktaları birleştirebilir. Tabi bir ikinci derece eğri çizmek için en az 3 noktaya ihtiyaç vardır

Program 7.1.26 ikincidereceP , ikinci derce polinom tipi eğri ile veriyi birleştirerek eğri çizme örneği

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

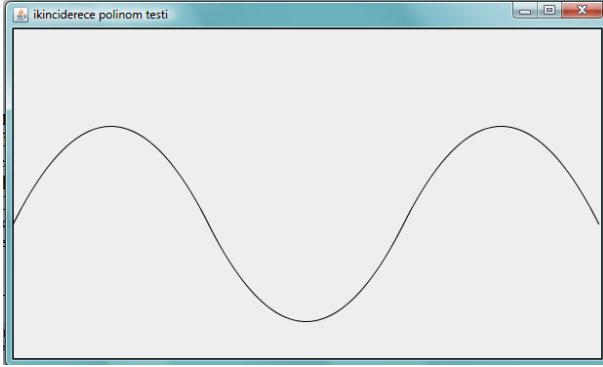
public class ikincidereceP extends JPanel
{
    public int x[];
    public int y[];
    public int xmax,ymax,xmin,ymin;
    public ikincidereceP(int xi[],int yi[])
    {
        int n=xi.length;
        x=new int[n];
        y=new int[n];
        for(int i=0;i<n;i++)
        {
            x[i]=xi[i];y[i]=yi[i];
        }
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        Dimension boyut=getSize();
        int dx=boyut.width;
        int dy=boyut.height;
        g2.setPaint(Color.BLACK);
        g2.draw3DRect(0,0,dx-1,dy-1,true);
        g2.setPaint(Color.BLACK);
        int n=x.length;
        int m=(n-1)/2;
        for(int i=0;i<n-1;i+=2)
        {
            QuadCurve2D qc2=new QuadCurve2D.Double(x[i],y[i],x[i+1],y[i+1],x[i+2],y[i+2]);
            g2.draw(qc2);
        }
    }
}
```

```
}  
}
```

Program 7.1.27 ikinci derece polinom örneği

```
class ikincidereceTest {  
public static void main(String args[])  
{int x[]={0,100,200,300,400,500,600};  
int y[]={200,0,200,400,200,0,200};  
ikincidereceP pp=new ikincidereceP(x,y);  
FrameGraphic.plot("ikinciderece polinom testi",pp);  
}  
}
```



Bir sonraki programda ikinci ve üçüncü derece polinomlar kullanılarak eğriler çizilmiştir. Bunun için yeni bir sınıf olan GeneralPath sınıfı kullanılmıştır.

Program 7.1.28 genelegriP eğri çizdirme programı

```
import javax.swing.*;  
import java.awt.Graphics;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
  
public class genelegriP extends JPanel  
{  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        Graphics2D g2 = (Graphics2D)g;  
        setBackground(Color.lightGray);  
        Dimension boyut=getSize();  
        int dx=boyut.width;  
        int dy=boyut.height;  
        g2.setStroke(new BasicStroke(3));  
        g2.draw3DRect(0,0,dx-3,dy-3,true);  
        g2.draw3DRect(3,3,dx-7,dy-7,false);  
        GeneralPath shape=new GeneralPath(GeneralPath.WIND_EVEN_ODD);  
        shape.moveTo(20,20);  
        //kuadratik ekleme
```

```

shape.quadTo(160,120,245,45);
//kübik ekleme
shape.curveTo(195,95,295,145,245,195);
shape.curveTo(-80,110,345,110,20,195);
shape.curveTo(400,250,200,250,20,20);
g2.draw(shape);
}}

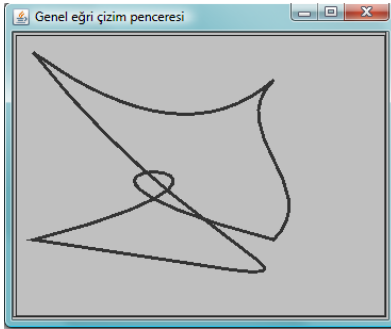
```

Program 7.1.29 genel eğri çizen örnek program

```

class genelegritest {
public static void main(String args[])
{
genelegriP pp=new genelegriP();
FrameGraphic.plot("Genel eğri çizim penceresi",pp);
}}

```



Bir sonraki eksersizde alt alta satırlar halinde yazı yazdırılacaktır.

Program 7.1.30 yazP alt alta yazı yazma programı

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
public class yazP extends JPanel
{ //grafik penceresine yazı yazar
String s[];
public yazP(String si[])
{super();
s=new String[si.length];
for(int i=0;i<si.length;i++){s[i]=si[i];}
}

public void paintComponent(Graphics g)
{ super.paintComponent(g);
Graphics2D g2 = (Graphics2D)g;
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
}
}

```

```

g2.setFont(new Font("Serif",Font.BOLD,24));
g2.setColor(Color.blue);
g2.setBackground(Color.yellow);
for(int i=0;i<s.length;i++)
    g2.drawString(s[i],50,50+25*i);
}
}

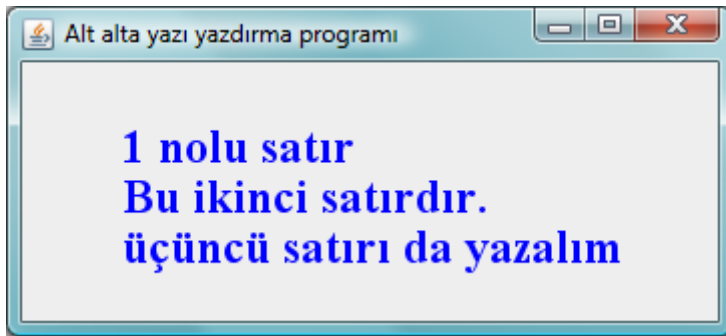
```

Program 7.1.31 alt alta yazı yazma program örneği

```

class yazTest {
public static void main(String args[])
{
String s[]{"1 nolu satır","Bu ikinci satırdır.,"üçüncü satırı da yazalım"};
yazP pp=new yazP(s);
FrameGraphic.plot("Alt alta yazı yazdırma programı",pp);
}
}

```



Son örneğimizde String kullanarak grafik çizdirmeyi deneyeceğiz.

Program 7.1.32 plotP JPanel class to draw a series of x writing following a plot function

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

abstract class f_x
{ abstract double func(double x);}

public class plotP extends JPanel
{
int x[];
int y[];
double xmin;
double xmax;
int n;
fb f;

public plotP(f_x f,double xmini,double xmaxi,int width,int height)
{

```

```

// height : height of the plot window;
// width : width of the plot window
// xmin minimum x value
// x maximum x value
xmin=xmini;
xmax=xmaxi;
n=100;
x=new int[n];
y=new int[n];
double xd[]=new double[n];
double yd[]=new double[n];
double ymin=1.0e60;
double ymax=-1.0e60;
for(int i=0;i<n;i++)
{xd[i] = (xmax-xmin)*(double)i/(double)n;
yd[i] = f.func(xd[i]);
if(yd[i]<ymin) ymin=yd[i];
if(yd[i]>ymax) ymax=yd[i];
}
for(int i=0;i<n;i++)
{x[i]=(int)(0.8*width*xd[i]/(xmax-xmin));
y[i]=height/3+(int)(0.6*height*yd[i]/(ymax-ymin));
}
}
public void paintComponent(Graphics g)
{
Graphics2D g2=(Graphics2D)g;
for(int i=0;i<x.length;i++)
{g2.drawString("x",x[i],y[i]);}
}
}

```

Program 7.1.33 An example program for plotP

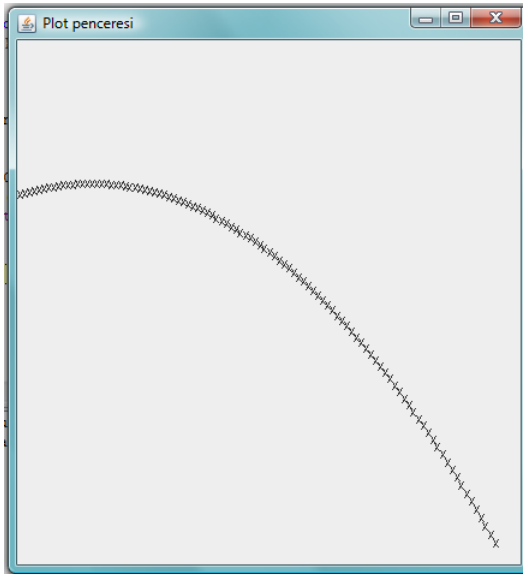
```

import javax.swing.*.*;

class fb extends f_x
{ public double func(double x)
{ return x*x-3*x-8;}
}

class plotTest {
public static void main(String args[])
{
fb ff=new fb();
plotP pp=new plotP(ff,0.0,10.0,500,500);
//cizgiP pp=new cizgiP();
FrameGraphic.plot("Plot penceresi",pp);
}}

```

Şimdi programda küçük bir değişiklik yaparak çizgi ile birleştirelim.

Program 7.1.34 plot2P JPanel sınıfı grafik programı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

abstract class f_x
{ abstract double func(double x);}

public class plot2P extends JPanel
{
    int x[];
    int y[];
    double xmin;
    double xmax;
    int n;
    fb f;
    int width;
    int height;
    public plot2P(f_x f,double xmin,double xmax,int width,int height)
    {
        // height : height of the plot window;
        // width : width of the plot window
        // xmin minimum x value
        // x maximum x value
        xmin=xmini;
        xmax=xmaxi;
        width=widthi;
        height=heighti;
        n=100;
        x=new int[n];
        y=new int[n];
        double xd[]=new double[n];
        double yd[]=new double[n];
        double ymin=1.0e60;
    }
}
```

```

double ymax=-1.0e60;
for(int i=0;i<n;i++)
{xd[i] = (xmax-xmin)*(double)i/(double)n;
yd[i] = f.func(xd[i]);
if(yd[i]<ymin) ymin=yd[i];
if(yd[i]>ymax) ymax=yd[i];
}
for(int i=0;i<n;i++)
{x[i]=(int)(0.8*width*xd[i]/(xmax-xmin));
y[i]=height/3+(int)(0.6*height*yd[i]/(ymax-ymin));
}
}

public void paintComponent(Graphics g)
{
Graphics2D g2=(Graphics2D)g;
for(int i=1;i<x.length;i++)
{ Line2D l=new Line2D.Double(x[i-1],y[i-1],x[i],y[i]);
g2.draw(l);
}
Rectangle2D r=new Rectangle2D.Double(0,0,height,width);
g2.draw(r);
}
}

```

Program 7.1.35 plot2P sınıfı örnek programı, sin(x) çizer

```

import javax.swing.*.*;

class fb extends f_x
{ public double func(double x)
{ return Math.sin(x);}
}

class H10Ex3 {
public static void main(String args[])
{
fb ff=new fb();
plot1P pp=new plot1P(ff,0.0,10.0,500,500);
//cizgiP pp=new cizgiP();
FrameGraphic.plot("Plot window",pp);
}
}

```



7.2 JAPPLET KULLANARAK GRAFİK PENCERESİ OLUŞTURMA VE KULLANMA

Bölüm 7.1 de JFrame kullanarak grafiklerimiz oluşturduk. Bu bölümde JApplet sınıfı ile aynı h grafik ortamını kullanmayı göreceğiz. Temel olarak JFrame ve JApplet programlama birbirinin aynısıdır, ancak JApplet sınıfında oluşturduğumuz programlar direkt olarak internet browser ortamına aktarılabilirler. Bu grafik sınıfı bizim için özel bir önem taşır. Bu sınıfın JFrame sınıfından en önemli farkı girdi metodu olarak **public void init()** metodunu kullanmamızdır. Bir önceki bölümde tanımladığımız Plot2P JPanel programını kullanan bir JApplet programı yazalım

Program 7.2.1 Applet_plot2P JApplet program

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

class fb extends f_x
{ public double func(double x)
  { return Math.sin(x);}
}

public class Applet_plot2P extends JApplet
{
  JPanel d;
  public void init()
  {
    fb ff=new fb();
    d=new plot2P(ff,0.0,10.0,500,500);
    add(d);
  }
}
```

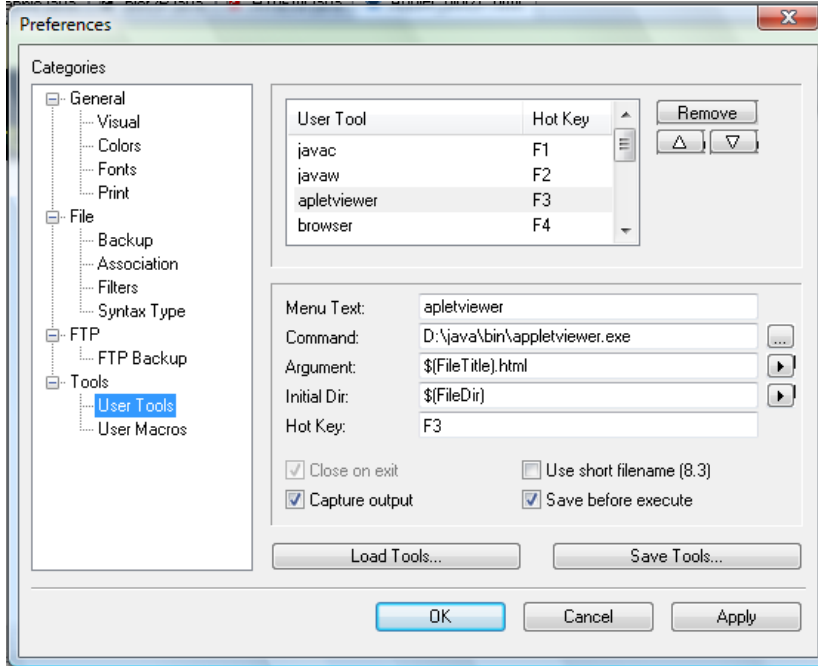
Bir Applet grafiğinin gösterilebilmesi için internet dili olan internet Hyper Text Mark-up language (HTML) kodu yazılırken program çağrılmalıdır. Aşağıda örnek bir HTML kodu görülmektedir :

Program 7.2.2 Applet_plot2P HTML programı :

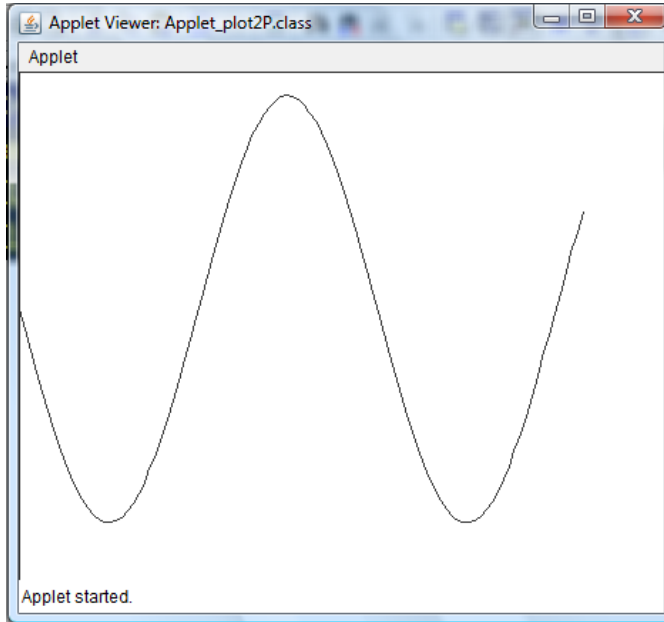
```
<html>
<applet code="Applet_plot2P.class" width=500 height=500>
</applet>
</html>
```

Bu HTML kodu herhangi bir browser programında açılabilir. Java çok basit bir browser simülasyon programını da geliştirmiştir. Applet programlarına bakmak için bu ortamı da kullanabiliriz. Java browser komutu appletviewer adını alır. Aynen java ve javaw komutları gibi bu komutu da konsol ortamından (dos) veya kullandığımız crimson editörün programlama kısmına appletviewer komutunu ekleyerek kullanabiliriz.

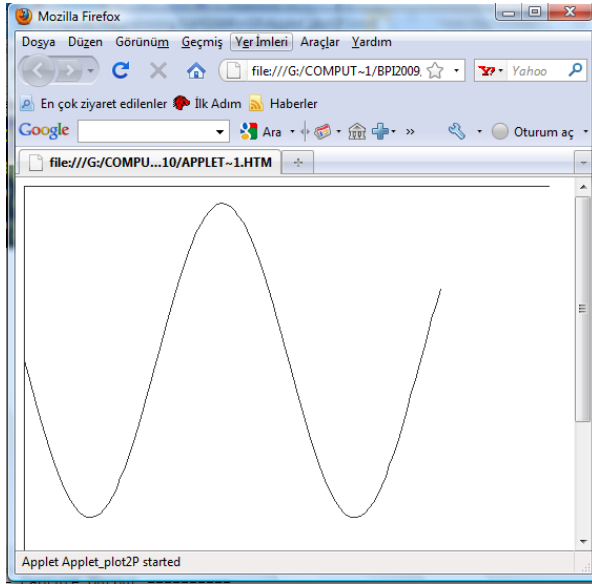
➤ **appletviewer** Applet_plot2P.html



Appletteki çıktı:



Aynı çıktının Mozilla Firefox browser'daki görünümü :



İkinci bir örneğe bakalım:

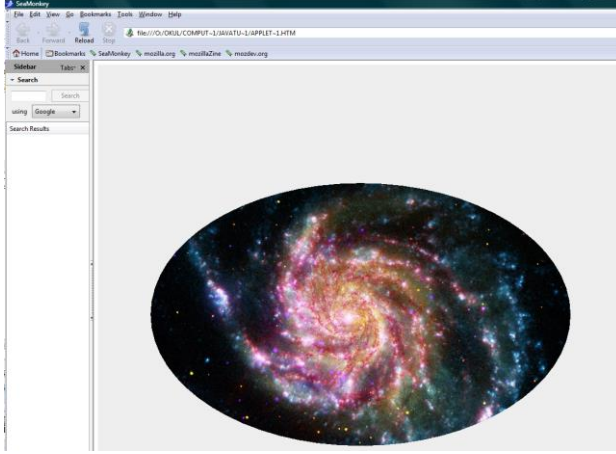
Program 7.2.3 Applet_ellipsP2 JApplet graphic program to show JPanel plot2P

```
import java.awt.event.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Applet_ellipsP2 extends JApplet
{
    JPanel d;
    public void init()
    {
        d=new elipsP2();
        add(d);
    }
}
```

Program 7.2.4 Applet_ellipsP2 HTML programı :

```
<html>
<applet code="Applet_ellipseP2.class" width=500 height=500>
</applet>
</html>
```

Çıktının seamonkey browserındaki görünümü



Son olarak JApplet programlarının da JFrame programına dönüştürülerek direk olarak main metodu içinde de çalıştırılabileceğini belirtelim.

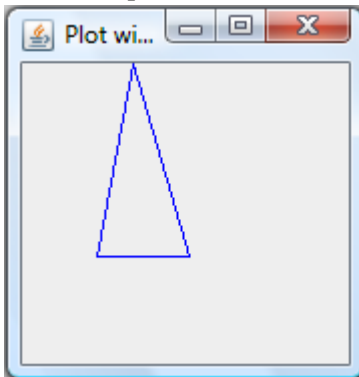
Program 7.2.5 JFrame'e dönüştürülerek main program içinden çalıştırılan bir JApplet programı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class AppletGrafigi extends JApplet
{
    public static void main(String s[])
    {
        JFrame f = new JFrame("Applet Graphics");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE );
        JApplet applet = new Applet_elipsP2();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.pack();
        f.setSize(new Dimension(600,600));
        f.setVisible(true);
    }
}
```

7.3 ALIŞTIRMALAR

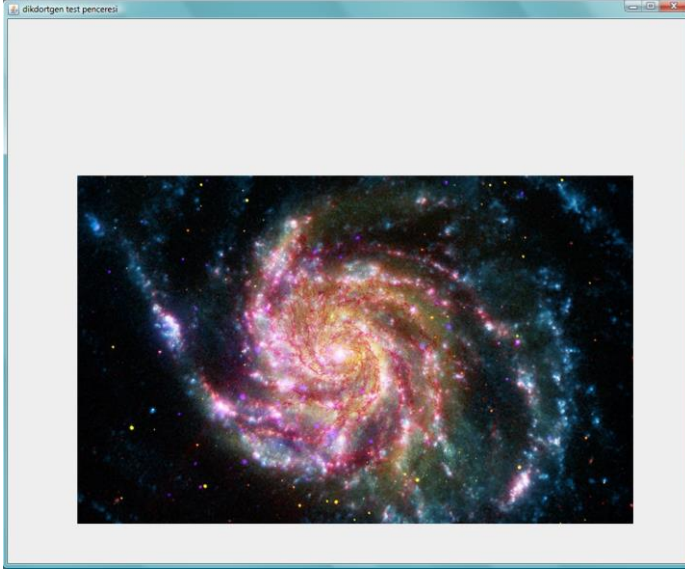
1) Grafik penceresinde bir üçgen çizdiriniz. Üçgenin koordinatlarını siz veriniz.



- 2) plot2P sınıfını inceleyiniz ve bu sınıfı kullanarak $y=x*x-2/x$ $x=1$ ile $x=5$ arasında grafik olarak gösteriniz. Grafik çıktısı almak için FrameGrafik sınıfını kullanınız.
- 3) Birbirine dik iki elips çiziniz.

7.4 PROBLEMLER

- 1) plot2P sınıfını inceleyiniz ve $y=Math.sin(x)*(0.1*Math.random())$ fonksiyonunu $x=0$ and $x=\pi$ sınırları arasında çizdiriniz. Bunun için JApplet kullanınız.
- 2) Aşağıdaki çıktıyı verecek JApplet programını yazınız



- 3) Beşgen çizecek bir JFrame programı yazınız.

8 GRAFİK KONTROL VE PROGRAMLAMA-GRAFİK KULLANICI ARAYÜZÜ (GRAPHIC USER INTERPHASE-GUI)

Önceki bölümümüzde grafik ekranını kullanarak çeşitli grafiklerin nasıl çizilebileceğini öğrendik. U bölümümüzde grafik penceresini kullanarak bilgisayar programlama ortamının nasıl kontrol edilebileceğinin detayına gireceğiz. Şu ana kadarki programlarımızda genellikle gir-hesapla-sonuçver yöntemini, yani konsol tipi program kullanımı gördük. Grafik program kontrolunda kullanırken program ekranımızda bir grafik penceresi olarak açık durur. Kullanıcı klavye, fare gibi ara birimler kullanarak programa komut gönderir. Komutlar dinleyici (Listener) dediğimiz alt metodlar tarafından alınarak gerekli işlemler yerine getirilir, bunu yapmak için grafik penceresi üzerinde düğme, yazı alanı, liste gibi çeşitli grafik-kullanıcı arabirimleri bulunur. Örneğin bir düğme verilmişse biz düğmeye bastığımızda düğmeye bağlı düğme dinleyicisi harekete geçer ve içinde tanımladığımız komutları yerine getirir. Program bundan sonra yine bekleme moduna geçer. Programı kapatmak için genellikle sağ üst köşede bulunan X işaretine fare ile basılır. Bu bölümde değişik kontrolleri ve işlevlerini ve bunları kullanarak Grafik ortamda programlamayı inceleyeceğiz. Burada bu yöntemin sadece girdi-çıkı amaçlı olduğunu ve arka planda asıl işi yapan sınıflarımızla bir ilgisi olmadığını belirtelim. Burada tüm yaptığımız kullanıcı için daha etken bir kullanım platformu oluşturmaktır. Arka planda çalışacak olan sınıflar yine aynı yapıda olacaklardır.

8.1 JBUTTON KONTROL DÜĞMESİ

JButton sınıfı grafik ekranında bir düğme oluşturur. Bu düğme ActionListener dinleyici metodu ile birlikte kullanılır. Fareyle düğmeye basıldığında

void actionPerformed(ActionEvent e)

metodunda tanımlanan işlemler devreye girerek yapılır. **actionPerformed** metodu sonunda grafiklerin tekrar çizildiğinden emin olmak için **repaint()** metodu kullanılır. İlk örnek problemimizde düğmeye basıldığında ekran arkaplan rengi tesadüfi bir renkle değiştirilmektedir.

Program 8.1.1 Jbutton test programı dugmeA0.java

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA0 extends JApplet implements ActionListener
{
    JPanel inputPanel;
    JButton b;

    public void init()
    {
        inputPanel=new JPanel();
        b=new JButton("renge deęiřtirmek için düęmeye basınız");
        b.addActionListener(this);
        inputPanel.add(b);
        add(inputPanel);
    }
    public void actionPerformed( ActionEvent e)
    {
        int red =(int)(255.0*Math.random());
        int green =(int)(255.0*Math.random());
        int blue =(int)(255.0*Math.random());
        Color c=new Color(red,green, blue);
        inputPanel.setBackground(c);
        repaint();
    }
}
```



```

public static void main(String s[])
{
    JFrame f = new JFrame("Applet Grafiği");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    JApplet applet = new dugmeA0();
    f.getContentPane().add("Center", applet);
    applet.init();
    f.setSize(800,500);
    f.setVisible(true);
}
}

```

Düğmeye her basılışında ekran renki tesadüfi seçilmiş yeni bir renkle değiştirilecektir.



Burada kullandığımız ActionListener sınıfının bir interface olduğunu ve actionPerformed metodunu tanımlamamızı sağladığını göz önüne alırsak, **lamda deyimi** üzerinden de yapabileceğimizi de görürüz.

Program 8.1.1 JButton test programı dugmeA0L.java lambda deyimi versiyonu

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA0L extends JApplet
{
    JPanel inputPanel;
    JButton b;

    public void init()
    {
        inputPanel=new JPanel();
        b=new JButton("rengi değiştirmek için düğmeye basınız");
        b.addActionListener(e->{
        int red =(int)(255.0*Math.random());
        int green =(int)(255.0*Math.random());
        int blue =(int)(255.0*Math.random());
        Color c=new Color(red,green, blue);
        inputPanel.setBackground(c);
        repaint();
        });
        inputPanel.add(b);
        add(inputPanel);
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("Applet Grafiği");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new dugmeA0();
    }
}

```

```
f.getContentPane().add("Center", applet);
applet.init();
f.setSize(800,500);
f.setVisible(true);
}
}
```

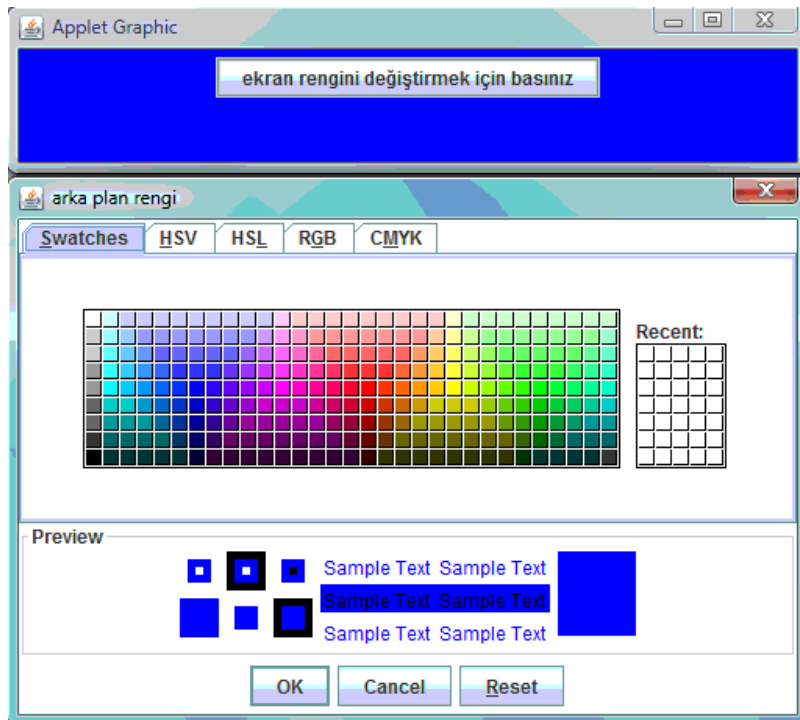
İkinci örneğimizde renk değişimi javanın hazır grafik renk seçme programı olan JColorChooser kullanarak yapılacaktır.

Program 8.1.2 JButton test program to change screen color by using JColorChooser class

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA01 extends JApplet implements ActionListener
{
    JPanel inputPanel;
    JButton b;
    Color c;
    public void init()
    {
        c=Color.blue;
        inputPanel=new JPanel();
        b=new JButton("ekran rengini değiştirmek için basınız");
        b.addActionListener(this);
        inputPanel.add(b);
        inputPanel.setBackground(c);
        add(inputPanel);
    }
    public void actionPerformed( ActionEvent e)
    {
        c=JColorChooser.showDialog(null,"arka plan rengi ",c);
        inputPanel.setBackground(c);
        repaint();
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("Applet Graphic");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new dugmeA01();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.setSize(800,500);
        f.setVisible(true);
    }
}
```



Program 8.1.2L JButton test program to change screen color by using JColorChooser class, lamda versiyonu

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA01L extends JApplet
{
    JPanel inputPanel;
    JButton b;
    Color c;
    public void init()
    {
        c=Color.blue;
        inputPanel=new JPanel();
        b=new JButton("ekran rengini deęiřtirmek iin basınız");
        b.addActionListener(e->
        { c=JColorChooser.showDialog(null,"arka plan rengi ",c);
          inputPanel.setBackground(c);
          repaint();
        });
        inputPanel.add(b);
        inputPanel.setBackground(c);
        add(inputPanel);
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("Applet Graphic");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new dugmeA01();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.setSize(800,500);
        f.setVisible(true);
    }
}

```

Üçüncü örneğimizde geçen bölümde oluşturduğumuz yıldız çizme JPanel programını biraz değiştirerek yıldız çizme ve içini doldurma arasında değiştirerek kullanacağız

Program 8.1.3 yıldizP1

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class yildizP1 extends JPanel
{
    Color c1=renk.mavi;//for shape
    Color c2=renk.beyaz;//for background
    public boolean dolu=false;

    public void icidolu() {dolu=true;}
    public void kenarcizgi() {dolu=false;}
    public void renksec(Color c) {c1=c;}
    public void renksec(int red,int green, int blue) {c1=new Color(red,green,blue);}
    public void arkaplanrenksec(Color c) {c2=c;}
    public void arkaplanrenksec(int red,int green, int blue) {c2=new Color(red,green,blue);}

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        GeneralPath yildiz=new GeneralPath();
        int x[]={55,67,109,73,83,55,27,37,1,43};
        int y[]={0,36,36,54,96,72,96,54,36,36};
        yildiz.moveTo(x[0],y[0]);
        for(int i=1;i<x.length;i++)
            {yildiz.lineTo(x[i],y[i]);}
        yildiz.closePath();
        g2.setColor(c1);
        this.setBackground(c2);
        if(dolu) g2.draw(yildiz);
        else g2.fill(yildiz);
    }
}
```

Bu örneğimizde yildizP JPanel programını yıldız çizmektedir, programdaki boolean değişken dolu çizdirmeye içini doldurma arasındaki kontrolü **public void paintComponent(Graphics g)** metodunda gerçekleştirmektedir. Dinleme metodu ActionPerformed da ise her düğmeye basılığında dolu değişkeninin değeri değişmektedir.

Program 8.1.4 dugmeF1 JFrame grafik programı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeF1 extends JFrame implements ActionListener
{
    JPanel inputPanel;
    JButton b;
    yildizP1 d;
    public dugmeF1(String a)
```

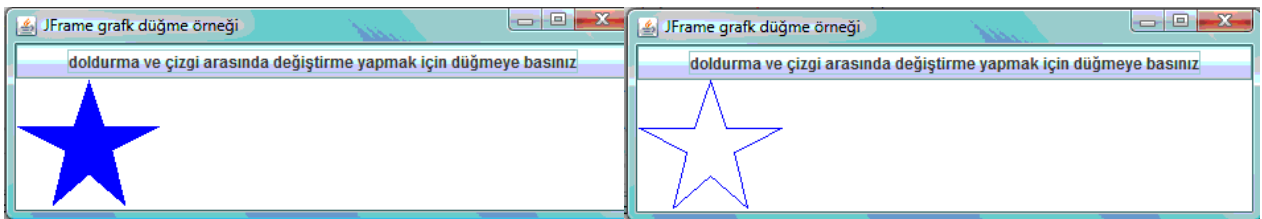
```

{
    super(a);
    inputPanel=new JPanel();
    inputPanel.setLayout(new BorderLayout());
    d=new yildizP1();
    d.icidolu();
    b=new JButton("doldurma ve çizgi arasında deęiřtirme yapmak için düğmeye basınız");
    b.addActionListener(this);
    inputPanel.add(b, BorderLayout.NORTH);
    inputPanel.add(d, BorderLayout.CENTER);
    add(inputPanel);
}

public void actionPerformed( ActionEvent e)
{
    if(d.dolu) d.kenarcizgi();
    else d.icidolu();
    repaint();
}

public static void main(String s[])
{
    JFrame f= new dugmeF1("JFrame grafik düğme örneęi");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



Bir sonraki programda grafik programı olarak JFrame yerine JApplet kullanılmıřtır. Bu arada istendięinde JApplet programını JFrame olarak main metodu üzerinden alıřtıracak program da listede verilmiřtir. Html kodu da ayrıca listelenmiřtir.

Program 8.1.5 buttonA1 JApplet graphic program

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA1 extends JApplet implements ActionListener
{
    JPanel inputPanel;
    JButton b;
    yildizP1 d;

    public void init()
    {
        inputPanel=new JPanel();
        inputPanel.setLayout(new BorderLayout());
        d=new yildizP1();
        d.icidolu();
        b=new JButton("draw ve fill arasında deęiřtir");
        b.addActionListener(this);
        inputPanel.add(b, BorderLayout.NORTH);
        inputPanel.add(d, BorderLayout.CENTER);
    }
}

```

```

add(inputPanel);
}
public void actionPerformed( ActionEvent e)
{
if(d.dolu) d.kenarcizgi();
else d.icidolu();
repaint();
}

public static void main(String s[])
{
JFrame f= new JFrame("Applet Grafiđi");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new dugmeA1();
f.getContentPane().add("Center", applet);
applet.init();
f.setSize(800,500);
f.setVisible(true);
}
}

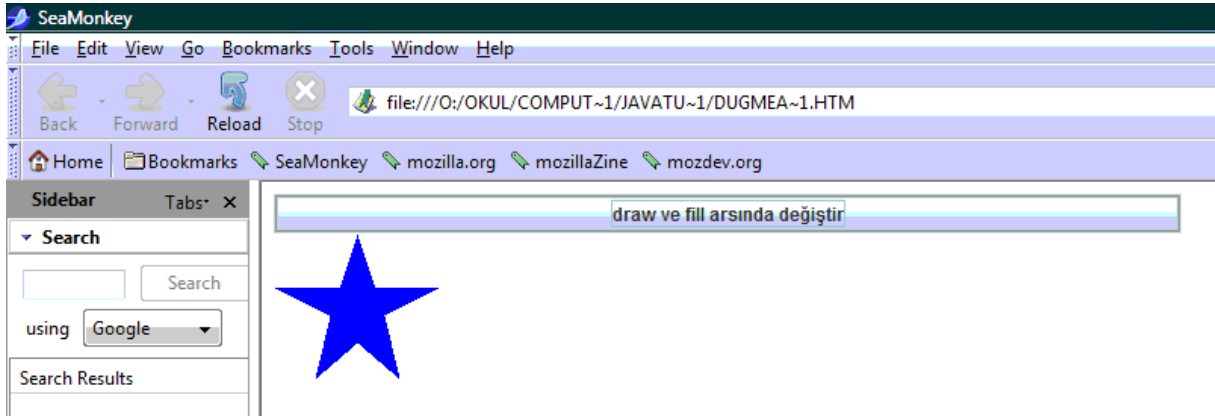
```

Program 8.1.6 dugmeA1.html html programı

```

<html>
<applet code="dugmeA1.class" width=600 height=500>
</applet>
</html>

```



Bir sonraki örnekte, programa bir düğme daha eklenerek yıldızın çizdirme şeklinin yanısıra rengi de deđiřtirilebilir duruma getirilmiřtir. Burada birden fazla düğme seęeneđi olduđundan hangi düğmenin ne görev göreceđini ayrı ayrı tanımlamak gerekir. Bunun için **public void actionPerformed(ActionEvent e)** metodunda **if(e.getSource()==b1)** gibi düğme seęme opsiyonu kullanılmıřtır.

Program 8.1.7 dugmeF2 JFrame grafik programı

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeF2 extends JFrame implements ActionListener
{
JPanel inputPanel;
JButton b1,b2;
yildizP1 d;
int kirmizi,yesil,mavi;

```

```

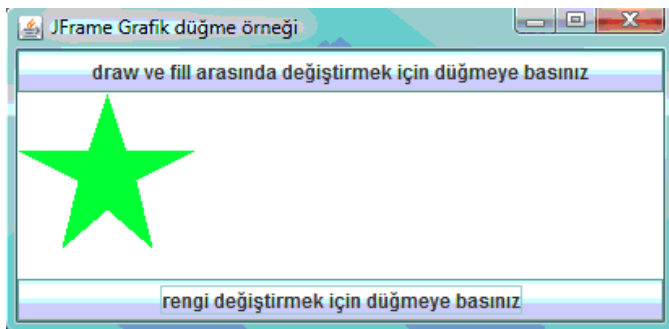
Color c;

public dugmeF2(String a)
{
    super(a);
    inputPanel=new JPanel();
    inputPanel.setLayout(new BorderLayout());
    d=new yildizP1();
    d.icidolu();
    b1=new JButton("draw ve fill arasında deęiřtirmek iin dğmeye basınız");
    b1.addActionListener(this);
    b2=new JButton("renęi deęiřtirmek iin dğmeye basınız");
    b2.addActionListener(this);
    inputPanel.add(b1,BorderLayout.NORTH);
    inputPanel.add(b2,BorderLayout.SOUTH);
    inputPanel.add(d,BorderLayout.CENTER);
    add(inputPanel);
}

public void actionPerformed( ActionEvent e)
{
    if(e.getSource()==b1)
    {
        if(d.dolu) d.kenarcizgi();
        else d.icidolu();
    }
    else if(e.getSource()==b2)
    {
        kirmizi =(int)(255.0*Math.random());
        yesil =(int)(255.0*Math.random());
        mavi =(int)(255.0*Math.random());
        d.renksec(kirmizi,yesil,mavi);
    }
    repaint();
}

public static void main(String s[])
{
    JFrame f= new dugmeF2("JFrame Grafik dğme rneęi");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



Program 8.1.7L dugmeF2 JFrame grafik programı lamda versiyonu

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeF2L extends JFrame
{

```

```

JPanel inputPanel;
JButton b1,b2;
yildizP1 d;
int kirmizi,yesil,mavi;
Color c;

public dugmeF2L(String a)
{
    super(a);
    inputPanel=new JPanel();
    inputPanel.setLayout(new BorderLayout());
    d=new yildizP1();
    d.icidolu();
    b1=new JButton("draw ve fill arasında deęiřtirmek iin dğmeye basınız");
    b1.addActionListener(e1->{ if(d.dolu) d.kenarcizgi();
                               else    d.icidolu();

                               repaint();
                               });
    b2=new JButton("rengi deęiřtirmek iin dğmeye basınız");
    b2.addActionListener(e2-> {
        kirmizi  =(int)(255.0*Math.random());
        yesil    =(int)(255.0*Math.random());
        mavi     =(int)(255.0*Math.random());
        d.renksec(kirmizi,yesil,mavi);
        repaint();
    });
    inputPanel.add(b1,BorderLayout.NORTH);
    inputPanel.add(b2,BorderLayout.SOUTH);
    inputPanel.add(d,BorderLayout.CENTER);
    add(inputPanel);
}

public static void main(String s[])
{
    JFrame f = new dugmeF2L("JFrame Grafik dğme rneęi");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,500);
    f.setVisible(true);
}
}

```

Aynı programı JApplet olarak verirsek :

Program 8.1.8 buttonA2 JApplet graphic program

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeA2 extends JApplet implements ActionListener
{
    JPanel inputPanel;
    JButton b1,b2;
    yildizP1 d;
    int kirmizi,yesil,mavi;;
    Color c;

    public void init()
    {
        inputPanel=new JPanel();
        inputPanel.setLayout(new BorderLayout());
        d=new yildizP1();
        d.icidolu();
        b1=new JButton("draw ve fill arasında deęiřtirmek iin dğmeye basınız");
    }
}

```



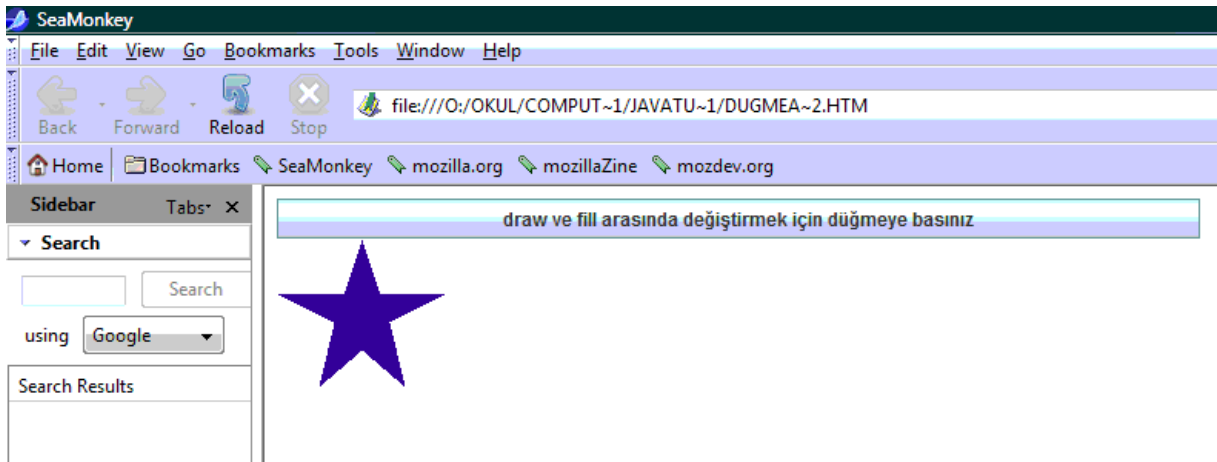
```

b1.addActionListener(this);
b2=new JButton("rengi deęiřtirmek iin dğmeye basınız");
b2.addActionListener(this);
inputPanel.add(b1, BorderLayout.NORTH);
inputPanel.add(b2, BorderLayout.SOUTH);
inputPanel.add(d, BorderLayout.CENTER);
add(inputPanel);
}

public void actionPerformed( ActionEvent e)
{
if(e.getSource()==b1)
{
if(d.dolu) d.kenarcizgi();
else d.icidolu();
}
else if(e.getSource()==b2)
{
kirmizi =(int)(255.0*Math.random());
yesil =(int)(255.0*Math.random());
mavi =(int)(255.0*Math.random());
d.renksec(kirmizi,yesil,mavi);
}
repaint();
}

public static void main(String s[])
{
JFrame f = new JFrame("Applet Graphic");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new dugmeA2();
f.getContentPane().add("Center", applet);
applet.init();
f.setSize(800,500);
f.setVisible(true);
}
}

```



Benzer bir program olarak bir lambayı aıp kapama programı oluřturalım. Burada grařik izimi yerine aık ve kapalı iki farklı lamba resmi kullanılmıřtır.

Previous program is changed a little bit. In this new version, instead of random color selection color selected through a library color change graphic window JColorChooser class

JColorChooser.showDialog method, This method returns a variable in class Color according to the color chosen in the graphic window.

Program 8.1.9 lambaP JPanel programı

```
import java.awt.image.*;
import java.util.*;

public class lambaP extends JPanel
{
    public Image lambaacik,lambakapali;
    public boolean acik;
    public lambaP()
    {super();
    URL url1 = getClass().getResource("lamba_acik.jpg");
lambaacik = getToolkit().getImage(url1);
URL url2 = getClass().getResource("lamba_kapali.jpg");
lambakapali = getToolkit().getImage(url2);
    acik=true;
    }
    public void ac() {acik=true;}
    public void kapa() {acik=false;}

    public void paint(Graphics g)
    {
    Graphics2D g2=(Graphics2D)g;
    Dimension d=getSize();
    int dx = d.width;
    int dy = d.height;
    if(acik)
    { g2.drawImage( lambaacik, 0, 0,dx,dy, this); }
    else
    { g2.drawImage( lambakapali, 0, 0,dx,dy, this);}
    }
}
```

Program 8.1.10 lambaF lamba açma kapama programı

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class lambaF extends JFrame implements ActionListener
{
    JPanel inputPanel;
    JButton b1,b2;
    lambaP d;
    Color c;

    public lambaF(String a)
    { super(a);
    c=Color.blue;
    inputPanel=new JPanel();
    inputPanel.setLayout(new BorderLayout());
    d=new lambaP();
    b1=new JButton("AÇ");
    b1.addActionListener(this);
    b2=new JButton("KAPA");
    b2.addActionListener(this);
    inputPanel.add(b1,BorderLayout.NORTH);
    inputPanel.add(b2,BorderLayout.CENTER);
```

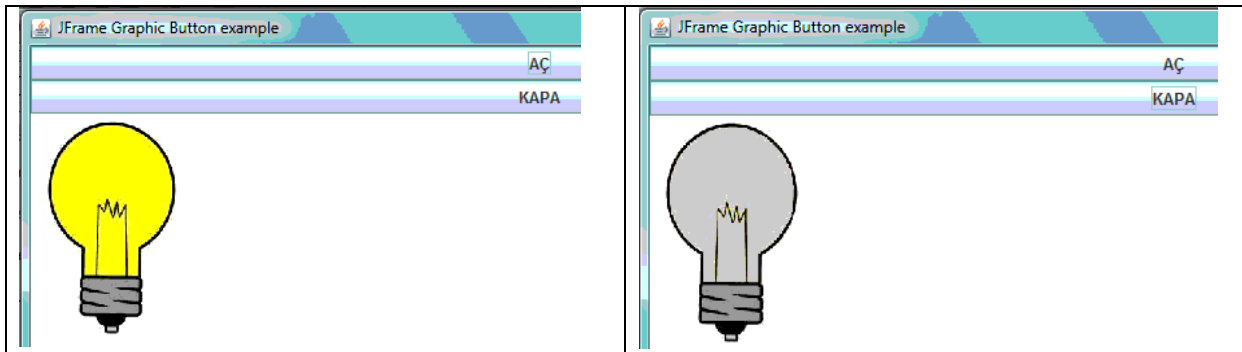
```

add(inputPanel, BorderLayout.NORTH);
add(d, BorderLayout.CENTER);
}

public void actionPerformed( ActionEvent e)
{
if(e.getSource()==b1) {d.ac();}
else if(e.getSource()==b2) {d.kapa();}
repaint();
}

public static void main(String s[])
{
    JFrame f = new lambdaF("JFrame Graphic Button example");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,800);
    f.setVisible(true);
}
}

```



Aynı programın lamda versiyonunu da verelim:

Program 8.1.10 lambdaF lamba açma kapama programı lamda versiyonu

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class lambdaFL extends JFrame
{
    JPanel inputPanel;
    JButton b1,b2;
    lambdaP d;
    Color c;

    public lambdaFL(String a)
    {
        super(a);
        c=Color.blue;
        inputPanel=new JPanel();
        inputPanel.setLayout(new BorderLayout());
        d=new lambdaP();
        b1=new JButton("AÇ");
        b1.addActionListener(e1->{d.ac();repaint();});
        b2=new JButton("KAPA");
        b2.addActionListener(e2->{d.kapa();repaint();});
    }
}

```

```

        inputPanel.add(b1, BorderLayout.NORTH);
        inputPanel.add(b2, BorderLayout.CENTER);
        add(inputPanel, BorderLayout.NORTH);
        add(d, BorderLayout.CENTER);
    }

    public static void main(String s[])
    {
        JFrame f = new lambaFL("JFrame Graphic Button example");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        f.setSize(800,800);
        f.setVisible(true);
    }
}

```

JButton düğme kontrolüne resim de ekleyebiliriz.

Program 8.1.11 JButton kontrol grafik programı lambaF1

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class lambaF1 extends JFrame implements ActionListener
{
    JPanel p1,p2;
    JButton b1,b2;
    lambaP d;
    Color c;

    public lambaF1(String a)
    {
        super(a);
        c=Color.blue;
        p1=new JPanel();
        p1.setLayout(new BorderLayout());
        p2=new JPanel();
        p2.setLayout(new BorderLayout());
        d=new lambaP();
        Icon lambaacik=new ImageIcon("lamba_acik_kucuk.jpg");
        Icon lambakapali=new ImageIcon("lamba_kapali_kucuk.jpg");
        b1=new JButton("AÇ",lambaacik);
        b1.addActionListener(this);
        b2=new JButton("KAPA",lambakapali);
        b2.addActionListener(this);
        p1.add(b1, BorderLayout.EAST);
        p1.add(b2, BorderLayout.WEST);
        p2.add(d, BorderLayout.CENTER);
        p2.add(p1, BorderLayout.SOUTH);
        add(p2);
    }

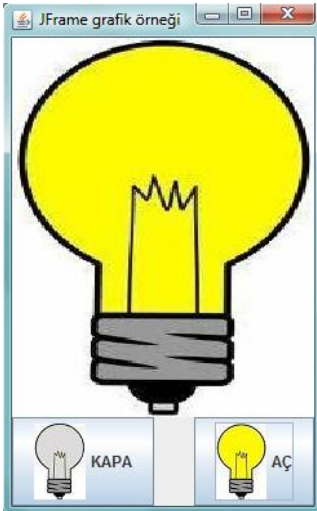
    public void actionPerformed( ActionEvent e)
    {
        if(e.getSource()==b1) { d.ac();}
        else if(e.getSource()==b2) { d.kapa();}
        repaint();
    }
}

```

```

public static void main(String s[])
{
    JFrame f = new lambaF1("JFrame grafik örneği");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(250,400);
    f.setVisible(true);
}
}

```



Aynı tür programın bir de yıldız çizdiren versiyonunu bir daha verelim. Burada çizgi, içini doldurma, yıldızı boyama ve çevreyi boyama olarak 3 kontrol düğmesi kullanılmıştır. İlk düğmede düğmeye Icon sınıfından resim eklenmiştir.

Program 8.1.11 dugmeF5 JFrame graphic program

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class dugmeF5 extends JFrame implements ActionListener
{
    JPanel inputPanel;
    JButton b1,b2,b3;
    yildizP1 d;
    int kirmizi,yesil,mavi;
    Color c1,c2;
    Icon sekil;

    public dugmeF5(String a)
    {
        super(a);
        inputPanel=new JPanel();
        inputPanel.setLayout(new BorderLayout());
        d=new yildizP1();
        d.icidolu();
        sekil=new ImageIcon("mandrill_kucuk.jpg");
        b1=new JButton("çizgi ve dolgu renk seçimi",sekil);
        b1.addActionListener(this);
        b2=new JButton("çizilen şeklin rengini değiştirin");
        b2.addActionListener(this);
        b3=new JButton("Arkaplan rengini değiştirin");
        b3.addActionListener(this);
        inputPanel.add(b1,BorderLayout.NORTH);
    }
}

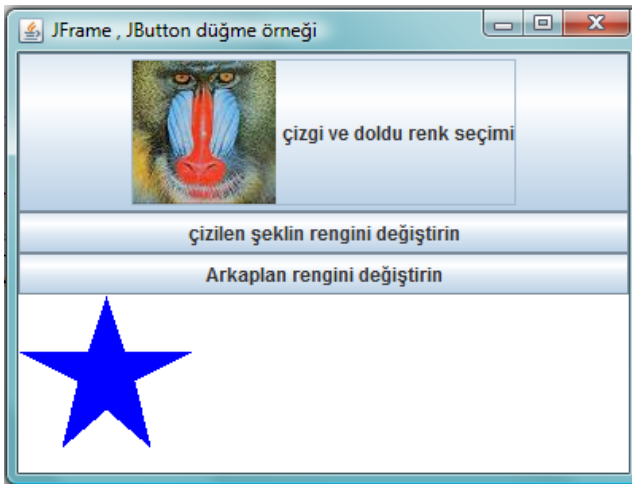
```

```

inputPanel.add(b2, BorderLayout.CENTER);
inputPanel.add(b3, BorderLayout.SOUTH);
add(inputPanel, BorderLayout.NORTH);
add(d, BorderLayout.CENTER);
}

public void actionPerformed( ActionEvent e)
{
if(e.getSource()==b1)
{
if(d.dolu) d.kenarcizgi();
else d.icidolu();
}
else if(e.getSource()==b2)
{//Random sekil color
kirmizi =(int)(255.0*Math.random());
yesil =(int)(255.0*Math.random());
mavi =(int)(255.0*Math.random());
d.renksec(kirmizi,yesil,mavi);
}
else if(e.getSource()==b3)
{//arkaplan rengini seçiniz
c2=JColorChooser.showDialog(null,"yıldızın arka planının rengi ",c2);
d.arkaplanrenksec(c2);
b1.setBackground(c2);
b2.setBackground(c2);
b3.setBackground(c2);
}
repaint();
}
public static void main(String s[])
{
JFrame f= new dugmeF5("JFrame , JButton düğme örneği");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
f.setSize(400,300);
f.setVisible(true);
}
}

```



8.2 JLABEL, JTEXTFIELD AND JTEXTAREA GRAFİK KONTROL SINIFLARI

Bu bölümümüzde üç kontrol gurubunu birlikte tanıtacağız bunlar : **JLabel**, **JTextField** and **JTextArea** sınıflarıdır. JLabel sadece yazı yazdırmak için kullanılır. JTextField ve JTextArea ise yazı yazdırmak ve gerektiğinde bu yazıları okumak (girdi, çıktı penceresi) için kullanılır. JTextField genellikle bir

satırlık kısa yazılar için, JTextArea uzun metinleri girmek için dizayn edilmiştir. JTextField ve JTextArea alt metodu **getText()** kullanılarak pencereye yazılan yazıyı okuyabiliriz. **setText(String s)** metodunu kullanarak da yeni bir String tipi değişkeni kutuya yazabiliriz.

İlk örnek programımızda bu üç sınıfı birlikte kullanacağız. İsim değişkeni JTextField alanından okunacak ve JTextArea alanına Hoşgeldiniz mesajı yazılacaktır.

Program 8.2.1 hoşgeldinizA JApplet grafik programı

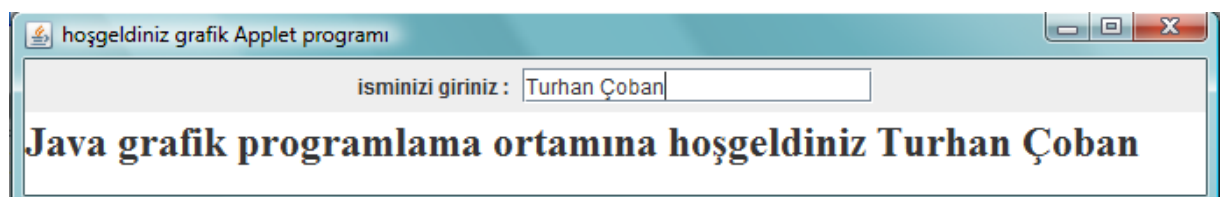
```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class hoşgeldinizA extends JApplet implements ActionListener
{
    JLabel label1;
    JTextField isimkutusu;
    JPanel girdipaneli;
    JTextArea t;
    String isim;

    public void init()
    {
        label1=new JLabel("İsminizi giriniz : ");
        t=new JTextArea(" ")
        t.setFont(new Font("Serif",Font.BOLD,24));
        isimkutusu=new JTextField(" ")
        setLayout(new BorderLayout());
        girdipaneli=new JPanel();
        girdipaneli.add(label1);
        girdipaneli.add(isimkutusu);
        add(girdipaneli,BorderLayout.NORTH);
        add(t,BorderLayout.CENTER);
        isimkutusu.addActionListener(this);
    }

    public void actionPerformed( ActionEvent e)
    {
        if(e.getSource()==isimkutusu)
        { isim=isimkutusu.getText();
          t.setText("Java grafik programlama ortamına hoşgeldiniz "+isim); }
    }

    public static void main(String s[])
    {
        JFrame f= new JFrame("hoşgeldiniz grafik Applet programı");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new hoşgeldinizA();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.setSize(800,500);
        f.setVisible(true);
    }
}
```



İkinci örnek programımız 1 den n e kadar sayıların toplamını hesaplayacak toplama işlemi ikinci bir metodun içinde yaptırılmıştır.

In the second example, numbers are added up to a given number which is defined in a JTextField area. A separate method is given for summation of the numbers.

Program 8.2.2 sumA JApplet graphic program

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class toplaa extends JApplet implements ActionListener
{
    JLabel label1;
    JTextField rakamkutusu;
    JPanel girdipaneli;
    JTextArea t;
    int n;

    public void init()
    {
        label1=new JLabel("1 den n e sayıların toplamı toplanacak son sayıyı (n) veriniz ");
        t=new JTextArea(" ");
        t.setFont(new Font("Serif",Font.BOLD,24));
        rakamkutusu=new JTextField("100");
        setLayout(new BorderLayout());
        girdipaneli=new JPanel();
        girdipaneli.add(label1);
        girdipaneli.add(rakamkutusu);
        add(girdipaneli,BorderLayout.NORTH);
        add(t,BorderLayout.CENTER);
        rakamkutusu.addActionListener(this);
    }

    public double toplaa(int n)
    { double tt=0;
      for(int i=0;i<=n;i++) { tt+=i;}
      return tt;
    }

    public void actionPerformed( ActionEvent e)
    {
        if(e.getSource()==rakamkutusu)
        { n=Integer.parseInt(rakamkutusu.getText());
          t.setText("1 den"+n+" e sayıların toplamı : "+toplaa(n)); }
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("1 den n e sayıların toplamı applet programı");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new toplaa();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.setSize(800,500);
        f.setVisible(true);
    }
}
```




Bir sonraki örnekte String bir değişken okunacak ve içindeki karakterlerin yerleri ters çevrilerek yazılacaktır. Bu işlem için **StringBuffer** sınıfı ve bu sınıfın **charAt(int i)** metodundan yararlanacağız. Bu sınıfla ilgili ilerde daha detaylı bilgi de verilecektir. Bu programda ActionPerformed metodu başlama komutunu fareden değil klavyedeki ctrl-return tuşundan almaktadır. Bu tuş basılınca addActionListener metodu devreye girmektedir.

Program 8.2.3 tersA JApplet grafik programı, yazıyı ters çevirir.

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class tersA extends JApplet implements ActionListener
{
    JLabel label1;
    JTextField cumlekutusu;
    JPanel girdipaneli;
    JTextArea t;
    String ters;

    public void init()
    {
        label1=new JLabel("cümlelerinizi yazınız : ");
        t=new JTextArea(" ");
        t.setFont(new Font("Serif",Font.BOLD,24));
        cumlekutusu=new JTextField("Methinks bulut aynı bir gelinciğe benziyor");

        setLayout(new BorderLayout());
        girdipaneli=new JPanel();
        girdipaneli.add(label1);
        girdipaneli.add(cumlekutusu);
        add(girdipaneli,BorderLayout.NORTH);
        add(t,BorderLayout.CENTER);
        cumlekutusu.addActionListener(this);
    }

    public static String invert(String source)
    { int i, len = source.length();
      StringBuffer dest = new StringBuffer(len);
      for (i = (len - 1); i >= 0; i--) {
        dest.append(source.charAt(i)); }
      return dest.toString();
    }

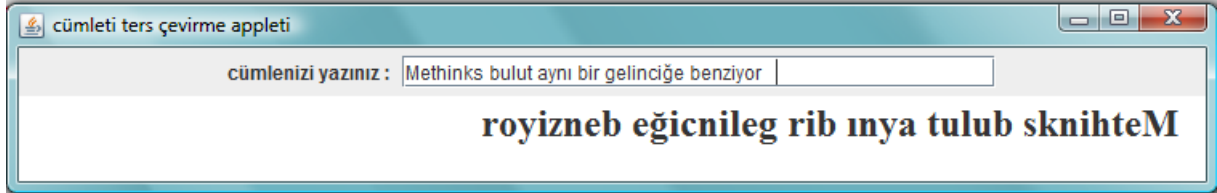
    public void actionPerformed( ActionEvent e)
    {
        if(e.getSource()==cumlekutusu)
        { ters=cumlekutusu.getText();
          t.setText(invert(ters)); }
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("cümleleri ters çevirme appleti");
```

```

f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new tersA();
f.getContentPane().add("Center", applet);
applet.init();
f.setSize(800,500);
f.setVisible(true);
}
}

```



Bir sonraki programda aynı kontrolü oluşturabilmek için hem JTextField hem de düğme(JButton) kullanılmıştır. Aynı zamanda yazı tesadüfi olarak renk değiştirmektedir.

Program 8.2.4 tersA1 JApplet grafik programı

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class tersA1 extends JApplet implements ActionListener
{
    JLabel label1;
    JTextField cumlekutusu;
    JPanel girdipaneli;
    JTextField t;
    String cumle;
    JButton b;
    String s;
    int kirmizi,yesil,mavi;

    public void init()
    {
        s="Methinks bulut aynı bir gelinciğe benziyor";
        label1=new JLabel("cümlelerinizi yazınız : ");
        t=new JTextField(s);
        t.setFont(new Font("Serif",Font.BOLD,24));
        cumlekutusu=new JTextField(s);
        b=new JButton("cümlelerinizi ters çeviriniz ");

        setLayout(new BorderLayout());
        girdipaneli=new JPanel();
        girdipaneli.add(label1);
        girdipaneli.add(cumlekutusu);
        girdipaneli.add(b);
        add(girdipaneli,BorderLayout.NORTH);
        add(t,BorderLayout.CENTER);
        cumlekutusu.addActionListener(this);
        b.addActionListener(this);
    }

    public static String invert(String source)
    { int i, len = source.length();
      StringBuffer dest = new StringBuffer(len);
      for (i = (len - 1); i >= 0; i--) {
        dest.append(source.charAt(i)); }
      return dest.toString();
    }
}

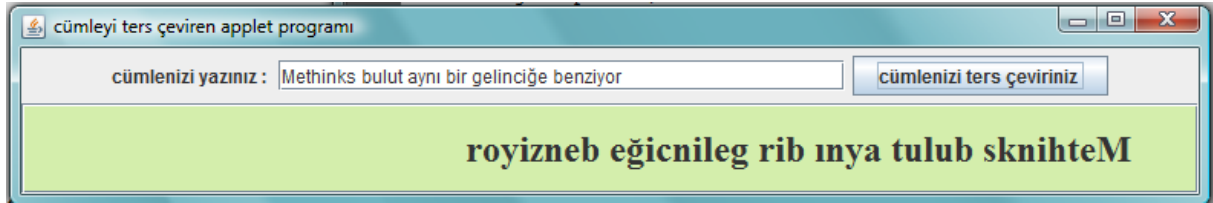
```

```

public void actionPerformed( ActionEvent e)
{
    if(e.getSource()==cumlekutusu)
    { s=cumlekutusu.getText();
      t.setText(s);
    }
    else if(e.getSource()==b)
    { t.setText(invert(t.getText()));
      kirmizi =(int)(255.0*Math.random());
      yesil =(int)(255.0*Math.random());
      mavi =(int)(255.0*Math.random());
      Color c1=new Color(kirmizi,yesil,mavi);
      t.setBackground(c1);
    }
}

public static void main(String s[])
{
    JFrame f= new JFrame("cümleyi ters çeviren applet programı");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    JApplet applet = new tersA1();
    f.getContentPane().add("Center", applet);
    applet.init();
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



Aynı programın lambda versiyonu:

Program 8.2.4 tersA1L JApplet grafik programı lamda versiyonu

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class tersA1L extends JApplet
{
    JLabel label1;
    JTextField cumlekutusu;
    JPanel girdipaneli;
    JTextField t;
    String cumle;
    JButton b;
    String s;
    int kirmizi,yesil,mavi;

    public void init()
    {
        s="Methinks bulut aynı bir gelinciğe benziyor";
        label1=new JLabel("cümlelerinizi yazınız : ");
        t=new JTextField(s);
        t.setFont(new Font("Serif",Font.BOLD,24));
    }
}

```

```

        cumlekutusu=new JTextField(s);
        b=new JButton("cümlelerinizi ters çeviriniz " );
        setLayout(new BorderLayout());
        girdipaneli=new JPanel();
        girdipaneli.add(label1);
        girdipaneli.add(cumlekutusu);
        girdipaneli.add(b);
        add(girdipaneli, BorderLayout.NORTH);
        add(t, BorderLayout.CENTER);
        cumlekutusu.addActionListener(e1->{s=cumlekutusu.getText();
            t.setText(s);
        });
        b.addActionListener(e2->{ t.setText(invert(t.getText()));
            kirmizi =(int)(255.0*Math.random());
            yesil =(int)(255.0*Math.random());
            mavi =(int)(255.0*Math.random());
            Color c1=new Color(kirmizi,yesil,mavi);
            t.setBackground(c1);
        });
    }

    public static String invert(String source)
    { int i, len = source.length();
      StringBuffer dest = new StringBuffer(len);
      for (i = (len - 1); i >= 0; i--) {
        dest.append(source.charAt(i)); }
      return dest.toString();
    }

    public static void main(String s[])
    {
        JFrame f = new JFrame("cümleleri ters çeviren applet programı");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        JApplet applet = new tersA1L();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.setSize(800,500);
        f.setVisible(true);
    }
}

```

8.3 FARE KONTROLU

Grafik programlama ortamında fare en yoğun kullanılan kontrol aygıtlarından biridir. Fare kullanarak oldukça karışık kontrol eylemlerini gerçekleştirebiliriz. Örneğin fare düğmesine basma, fare düğmesine basarak hareket ettirme, fare düğmesini bırakma gibi eylemler ayrı ayrı kontrol edilebilir.

Fare kontrollerinde kullanılan iki tür dinleme sınıfı mevcuttur. Bunlar fare dinleyicisi ve fare hareket dinleyicisi sınıflarıdır. (**MouseListener** ve **MouseMotionListener**)

Bunlar fare dinleyicisi (**MouseListener**) sınıfının alttaki metodları mevcuttur

public void mouseClicked(MouseEvent e) (fare düğmesine basıldı ve çekildi - kliklendi)

public void mousePressed(MouseEvent e) (fare düğmesine basıldı)

public void mouseReleased(MouseEvent e) (fare düğmesinden el çekildi)

public void mouseEntered(MouseEvent e) (fare tanımlanmış grafik alanın içine girdi)

public void mouseExited(MouseEvent e) (Fare tanımlanmış grafik alandan çıktı)

Fare hareket kontrolü sınıfının (**MouseMotionListener**) alttaki kontrol metodları mevcuttur.

public void mouseDragged(MouseEvent e) (fare tuşuna basılarak hareket ettirildi)

public void mouseMoved(MouseEvent e) (fare hareket ettirildi)

Fare kontrolünü saptayan sınıf ta **MouseEvent** sınıfıdır. Bu sınıfın farenin piksel koordinatlarını veren **getX()** ve **getY()** metodları mevcuttur.

İlk örneğimiz fareye önce basıp, sonra sürükleyip fare düğmesini bırakarak grafik ekranında doğru çizdireceğimiz fareP JPanel programıdır.

Program 8.3.1 fare kontrol JPanel programı fareP

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class mouseP extends JPanel implements MouseListener, MouseMotionListener
{
    //push-move-release line draw
    int x1,y1,x2,y2;
    Color c;
    boolean first;
    Line2D x;

    public mouseP()
    {super();
    x1=0;y1=0;x2=0;y2=0;
    first=true;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public mouseP(int x1,int y1,int x2,int y2)
    {super();
    x1=x1;y1=y1;x2=x2;y2=y2;
    first=true;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public void setLine(int x1,int y1,int x2,int y2)
    {x1=x1;y1=y1;x2=x2;y2=y2;}

    public void paint(Graphics g)
    {
    super.paintComponent(g);
    Graphics2D g2=(Graphics2D)g;
    g2.setFont(new Font("Serif",Font.BOLD,24));
    g2.setColor(Color.blue);
    g2.setStroke(new BasicStroke(2.0f));
    x=new Line2D.Double(x1,y1,x2,y2);
    g2.draw(x);
    }

    //MouseListener
```

```

public void mouseClicked(MouseEvent e)
{ }

public void mousePressed(MouseEvent e)
{
    x1=e.getX();y1=e.getY();}

public void mouseReleased(MouseEvent e)
{ x2=e.getX();y2=e.getY();repaint();}

public void mouseEntered(MouseEvent e)
{ }

public void mouseExited(MouseEvent e)
{ }
//MouseMotionListener

public void mouseDragged(MouseEvent e)
{ }
public void mouseMoved(MouseEvent e)
{ }
}

```

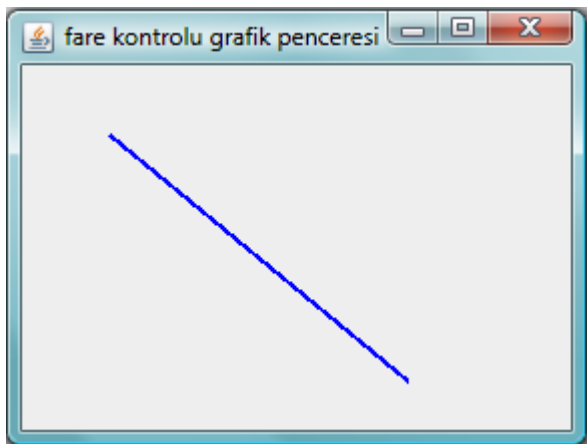
This program will be used with the previously developed FrameGrafhis class. The test program will be as follows:

Program 8.3.2 fare kontrol programı fareP test programı fareF

```

class fareF {
public static void main(String args[])
{FrameGrafigi.plot("fare kontrolu grafik penceresi",new fareP());}
}

```



İkinci programımız çizgi yerine her basışta nokta (aslında küçük bir daire) çizer

Program 8.3.3 fareP1 JPanel fare kontrol programı

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

```

```

import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class fareP1 extends JPanel implements MouseListener,MouseMotionListener
{
    //push to draw a point
    int x1,y1;
    Color c;

    public fareP1()
    {super();
    x1=0;y1=0;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public fareP1(int x1i,int y1i,int x2i,int y2i)
    {super();
    x1=x1i;y1=y1i;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public void setLine(int x1i,int y1i,int x2i,int y2i)
    {x1=x1i;y1=y1i;}

    public void paint(Graphics g)
    { Graphics2D g2 = (Graphics2D)g;
    g2.setPaint(Color.black);
    Ellipse2D elips1=new Ellipse2D.Double(x1,y1,5,5);
    g2.fill(elips1);
    }

    //MouseListener (mouse dinleyicisi)
    public void mouseClicked(MouseEvent e)
    { x1=e.getX();y1=e.getY();repaint(); }

    public void mousePressed(MouseEvent e)
    { x1=e.getX();y1=e.getY();repaint();
    }

    public void mouseReleased(MouseEvent e)
    {x1=e.getX();y1=e.getY();repaint();}

    public void mouseEntered(MouseEvent e)
    { }

    public void mouseExited(MouseEvent e)
    { }

    //MouseMotionListener (mouse hareket dinleyicisi) metotlari

    public void mouseDragged(MouseEvent e)
    {}

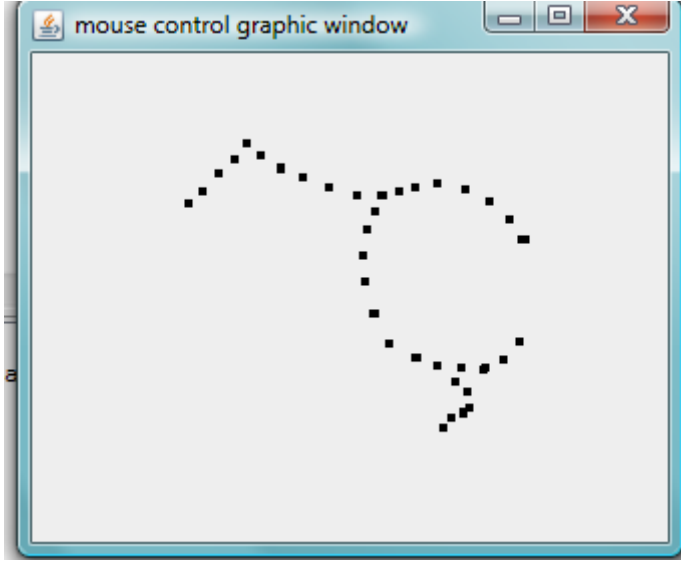
    public void mouseMoved(MouseEvent e)
    {}

}

```

Program 8.3.4 JPanel fareP1 fare kontrol test programı fareF1

```
class fareF1 {
public static void main(String args[])
{FrameGrafigi.plot("mouse control graphic window",new fareP1());}
}
```



Bir sonraki fare kontrol örneğinde bir çok çizgi çizilmesi için bir boyutlu değişken üzerinde tüm çizilen çizgilerin koordinatları depolanarak yeniden çizdirilmektedir.

Program 8.3.5 fareP2 JPanel fare kontrol programı

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class fareP2 extends JPanel implements MouseListener,MouseMotionListener
{
//BAS-SÜRÜKLE-ÇEK ÇİZGİ AİLESİ
    int x1[],y1[],x2[],y2[];

    int n;
    Line2D l[];
    Color c;

    public fareP2()
    {super();
    x1=new int[100];
    y1=new int[100];
    l=new Line2D[100];
    x2=new int[100];
    y2=new int[100];
    n=0;
    addMouseListener(this);
    addMouseMotionListener(this);
    }
}
```



```

public void setLine(int x1i,int y1i,int x2i,int y2i)
{x1[n]=x1i;y1[n]=y1i;x2[n]=x2i;y2[n]=y2i;n++;}

public void paint(Graphics g)
{
super.paintComponent(g);
Graphics2D g2=(Graphics2D)g;
g2.setFont(new Font("Serif",Font.BOLD,24));
g2.setColor(Color.blue);
g2.setStroke(new BasicStroke(2.0f));
for(int i=0;i<n;i++)
{l[i]=new Line2D.Double(x1[i],y1[i],x2[i],y2[i]);
g2.draw(l[i]);
}}

//MouseListener (fare dinleyicisi)
public void mouseClicked(MouseEvent e)
{ }

public void mousePressed(MouseEvent e)
{
x1[n]=e.getX();y1[n]=e.getY();
System.out.println(x1[n]+" "+y1[n]);
}

public void mouseReleased(MouseEvent e)
{ x2[n]=e.getX();y2[n]=e.getY();n++;repaint();}

public void mouseEntered(MouseEvent e)
{ }

public void mouseExited(MouseEvent e)
{ }
//MouseMotionListener (fare hareket dinleyicisi) metotlari

public void mouseDragged(MouseEvent e)
{ }
public void mouseMoved(MouseEvent e)
{ }
}

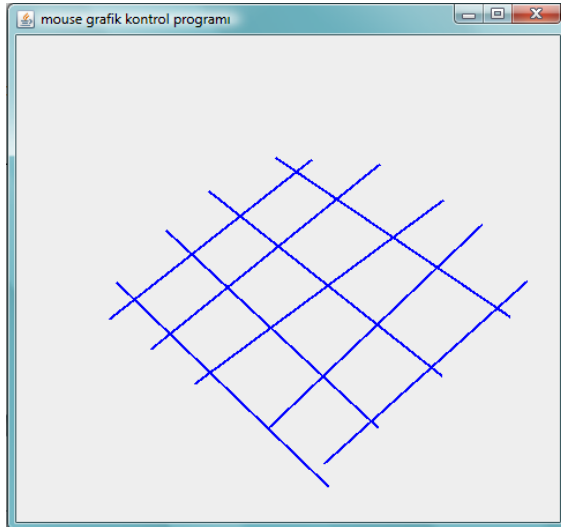
```

Program 8.3.6 fareF2 test programı fareF2

```

class fareF2 {
public static void main(String args[])
{FrameGrafigi.plot("mouse grafik kontrol programı",new fareP2());}
}

```



Bir sonraki fare kontrol programında fareye her basışta bir önce basılarak çizilen doğrunun son noktasından yeni basılan noktaya bir doğru çizdirilmektedir

Program 8.3.7 fareP3 JPanel fare kontrol programı

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class fareP3 extends JPanel implements MouseListener, MouseMotionListener
{
    //push and push again line draw
    int x1[],y1[],x2[],y2[];

    int n;
    Line2D l[];
    Color c;

    public fareP3()
    {super();
    x1=new int[100];
    y1=new int[100];
    l=new Line2D[100];
    n=0;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public void setLine(int x1i,int y1i,int x2i,int y2i)
    {x1[n]=x1i;y1[n]=y1i;x2[n]=x2i;y2[n]=y2i;n++;}
```

```

public void paint(Graphics g)
{
super.paintComponent(g);
Graphics2D g2=(Graphics2D)g;
g2.setFont(new Font("Serif",Font.BOLD,24));
g2.setColor(Color.blue);
g2.setStroke(new BasicStroke(2.0f));
for(int i=1;i<n;i++)
{l[i]=new Line2D.Double(x1[i],y1[i],x1[i-1],y1[i-1]);
g2.draw(l[i]);
}
}

//MouseListener (mouse dinleyicisi)
public void mouseClicked(MouseEvent e)
{}

public void mousePressed(MouseEvent e)
{
if(n==0) {x1[n]=e.getX();y1[n]=e.getY();System.out.println("n = "+n+" x1= "+x1[n]+"y1 = "+y1[n]);n++;}
else { x1[n]=e.getX();y1[n]=e.getY();repaint();System.out.println("n = "+n+"x1 = "+x1[n]+"y1 = "+y1[n]);n++;}
}

public void mouseReleased(MouseEvent e)
{}

public void mouseEntered(MouseEvent e)
{}

public void mouseExited(MouseEvent e)
{}
//MouseMotionListener (mouse hareket dinleyicisi) metotlari

public void mouseDragged(MouseEvent e)
{}
public void mouseMoved(MouseEvent e)
{}
}

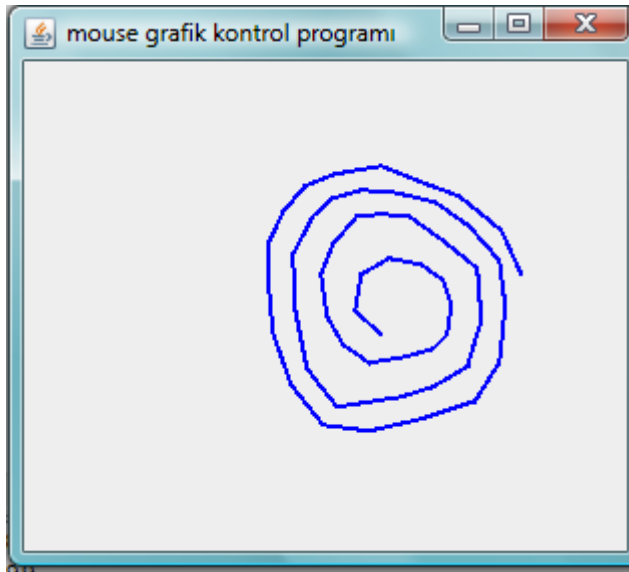
```

Program 8.3.8 fare kontrol test programı fareF3

```

class fareF3 {
public static void main(String args[])
{FrameGrafigi.plot("mouse grafik kontrol programı",new fareP3());}
}

```



Bir sonraki program da bir öncekinin benzeridir, ancak noktalar arasına doğru yerine ikinci dereceden polinomlar çizdirilmektedir.

Program 8.3.9 fareP4 fare kontrol programı

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class fareP4 extends JPanel implements MouseListener, MouseMotionListener
{
    //push and push again line draw
    int x1[], y1[], x2[], y2[];
    int n;
    QuadCurve2D l[];
    Color c;

    public fareP4()
    {
        super();
        x1=new int[200];
        y1=new int[200];
        l=new QuadCurve2D[100];
        n=0;
        addMouseListener(this);
        addMouseMotionListener(this);
    }

    public void setLine(int x1i,int y1i,int x2i,int y2i)
    {
        x1[n]=x1i;y1[n]=y1i;x2[n]=x2i;y2[n]=y2i;n++;
    }
}
```

```

public void paint(Graphics g)
{
super.paintComponent(g);
Graphics2D g2=(Graphics2D)g;
g2.setFont(new Font("Serif",Font.BOLD,24));
g2.setColor(Color.blue);
g2.setStroke(new BasicStroke(2.0f));
for(int i=2;i<n;i+=2)
{l[i]=new QuadCurve2D.Double(x1[i],y1[i],x1[i-1],y1[i-1],x1[i-2],y1[i-2]);
g2.draw(l[i]);
}
}

//MouseListener (mouse dinleyicisi)
public void mouseClicked(MouseEvent e)
{}

public void mousePressed(MouseEvent e)
{
if(n==0) {x1[n]=e.getX();y1[n]=e.getY();System.out.println("n = "+n+" x1= "+x1[n]+"y1 = "+y1[n]);n++;}
else { x1[n]=e.getX();y1[n]=e.getY();repaint();System.out.println("n = "+n+"x1 = "+x1[n]+"y1 = "+y1[n]);n++;}
}

public void mouseReleased(MouseEvent e)
{}

public void mouseEntered(MouseEvent e)
{}

public void mouseExited(MouseEvent e)
{}
//MouseMotionListener (mouse hareket dinleyicisi) metotlari

public void mouseDragged(MouseEvent e)
{}
public void mouseMoved(MouseEvent e)
{}
}

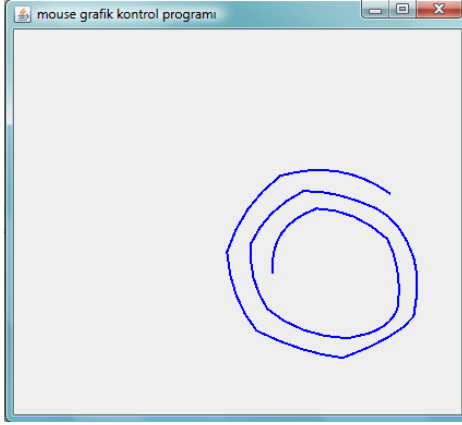
```

Program 8.3.10 Test program for Mouse control JPanel class mouseP2

```

class fareF4 {
public static void main(String args[])
{FrameGrafigi.plot("mouse grafik kontrol programi",new fareP4());}
}

```



8.4 İŞARET KUTUSU (JCHECKBOX)

Bazı seçimleri işaretlemek istediğimizde JCheckBox kullanabiliriz. JCheckBox Sınıfının kurucu metodları şunlardır:

```
public JCheckBox()  
public JCheckBox(Icon resim)  
public JCheckBox(Icon resim,boolean kutuişareti)  
public JCheckBox(String yazı)  
public JCheckBox(String yazı, boolean kutuişareti)  
public JCheckBox(String yazı, Icon resim)  
public JCheckBox(String yazı, Icon resim ,boolean kutuişareti)
```

Bu sınıfta işaretlenecek yere resim de koyabiliriz.

Program 8.4.1 kontrol_kutusuP

```
import java.io.*;  
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
import java.util.*;  
import javax.swing.*;  
import javax.swing.table.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
  
public class kontrol_kutusuP extends JPanel implements ItemListener  
{  
    JPanel inputPanel;  
    JTextField t;  
    JCheckBox bold,italic;  
    int normal_font,bold_font,italic_font;  
    String s;  
  
    public kontrol_kutusuP()  
    {  
        s="Bu yazının stilini değiştiriniz : ";  
        inputPanel=new JPanel();  
        t=new JTextField(s,40);  
        italic_font=Font.PLAIN;  
        bold_font=Font.PLAIN;
```

```

        t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
        bold=new JCheckBox("Kalin");
        italic=new JCheckBox("İtalik");
        inputPanel.add(t);
        inputPanel.add(bold);
        inputPanel.add(italic);
        add(inputPanel);
        italic.addItemListener(this);
        bold.addItemListener(this);
    }

    public void itemStateChanged(ItemEvent e)
    {
        if(e.getSource()==bold)
        {
            if(e.getStateChange()==ItemEvent.SELECTED)
            {
                bold_font=Font.BOLD;}
            else
            {
                bold_font=Font.PLAIN;}
        }
        else if(e.getSource()==italic)
        {
            if(e.getStateChange()==ItemEvent.SELECTED)
            {
                italic_font=Font.ITALIC;}
            else
            {
                italic_font=Font.PLAIN;}
        }
        t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
        repaint();
    }
}

```

Program 8.4.2 kontrol_kutusuF

```

class kontrol_kutusuF {
public static void main(String args[])
{FrameGrafigi.plot("Kontrol kutusu",new kontrol_kutusuP());}
}

```



Bir sonraki program kontrol kutusu ve JButton renk kontrolunu birlikte kullanmaktadır.

Program 8.4.3 kontrol_kutusuP1

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
public class kontrol_kutusuP1 extends JPanel implements ActionListener,ItemListener
{

```

```

JPanel inputPanel;
JButton b;
JTextField t;
JCheckBox bold,italic;
int red,green,blue;
int normal_font,bold_font,italic_font;
public Color renk;
String s;

public kontrol_kutusuP1()
{
    s="Bu yazının stilini deęiřtiriniz : ";
    renk=Color.blue;
    inputPanel=new JPanel();
    inputPanel.setLayout(new GridLayout(4,1));
    b=new JButton("Ekran rengini deęiřtirmek iin dğmeye basınız");
    t=new JTextField(s);
    italic_font=Font.PLAIN;
    bold_font=Font.PLAIN;
    t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
    bold=new JCheckBox("Bold");
    italic=new JCheckBox("Italic");
    inputPanel.add(b,BorderLayout.NORTH);
    inputPanel.add(t);
    inputPanel.add(bold);
    inputPanel.add(italic);
    add(inputPanel);
    b.addActionListener(this);
    italic.addItemListener(this);
    bold.addItemListener(this);
}

public void itemStateChanged(ItemEvent e)
{
    if(e.getSource()==bold)
    {
        if(e.getStateChange()==ItemEvent.SELECTED)
        {
            bold_font=Font.BOLD;
        }
        else
        {
            bold_font=Font.PLAIN;
        }
    }
    else if(e.getSource()==italic)
    {
        if(e.getStateChange()==ItemEvent.SELECTED)
        {
            italic_font=Font.ITALIC;
        }
        else
        {
            italic_font=Font.PLAIN;
        }
    }
    t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
    repaint();
}

public void actionPerformed( ActionEvent e)
{
    //tesadfi renk belirle dğmeye her basılıřta rengi deęiřtir.
    red=(int)(255.0*Math.random());
    green =(int)(255.0*Math.random());
    blue =(int)(255.0*Math.random());
    renk=new Color(red,green,blue);
    //input panelin rengini deęiřtiriyoruz
    inputPanel.setBackground(renk);
    //ana panelin rengini deęiřtiriyoruz

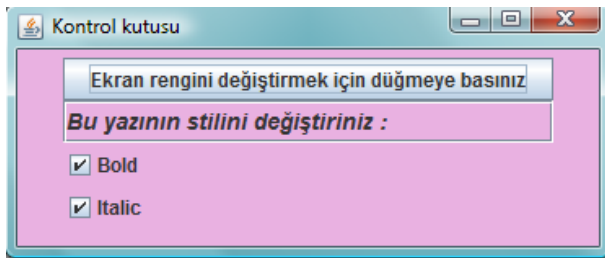
```



```
setBackground(renk);
//yazı kutusunun rengini deęiřtiriyoruz
t.setBackground(renk);
bold.setBackground(renk);
italic.setBackground(renk);
repaint();
}
}
```

Program 8.4.4 kontrol_kutusuP1

```
class control_boxF1 {
public static void main(String args[])
{
FrameGraphic.plot("check box control graphic window",new control_boxP1());
}}
```



8.5 RADYO KONTROL DÜĞMESİ

JRadioButton sınıfı JCheckBox sınıfına benzer. Kurucu metodları :

```
public JRadioButton()
public JRadioButton(Icon picture)
public JRadioButton(Icon picture,boolean boxsign)
public JRadioButton(String writing)
public JRadioButton(String writing, boolean boxsign)
public JRadioButton(String writing, Icon picture)
public JRadioButton(String writing, , Icon picture ,boolean boxsign)
```

JRadioButton sınıfıyla ilişkili dięer bir sınıf ta Buton gurup sınıfıdır. Bu sınıf birlikte kontrol oluřturacak kontrol gurubu oluřturur. Sınıfın temel metodları řu řekildedir:

```
public class ButtonGroup extends Object implements Serializable
{
protected Vector Buttons;
public ButtonGroup();
public void add(AbstractButton ab);
public Enumeration getElements();
public ButtonModel getSelection();
public boolean isSelected(ButtonModel bm);
public void remove(AbstractButton ab);
public void setSelected(ButtonModel bm,boolean b);
}
```

Program 8.5.1 radyo_kontrolu (JRadioButton sınıfı)

```
import java.io.*;
```

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class radyo_kontrolu extends JApplet implements ActionListener,ItemListener
{

    JPanel inputPanel;
    JButton b;
    JTextField t;
    JRadioButton plain,bold,italic,bolditalic;
    ButtonGroup gurup;
    int kirmizi,yesil,mavi;
    int bold_font,italic_font;
    public Color col;
    String s;
    public void init()
    {
        s="yazının stillerini deęiřtiriniz : alt N Düz yazı alt K Kalın yazı alt t italik alt a kalın-italik";
        col=Color.blue;
        inputPanel=new JPanel();
        setLayout(new BorderLayout());
        inputPanel.setLayout(new GridLayout(1,4));
        b=new JButton("renk deęiřtirmek için basınız");
        t=new JTextField(s);
        italic_font=Font.PLAIN;
        bold_font=Font.PLAIN;
        JPanel Check=new JPanel();
        Check.setLayout(new GridLayout(4,0));
        gurup=new ButtonGroup();
        plain=new JRadioButton("Plain");
        plain.setMnemonic('N');
        plain.setSelected(true);
        plain.addItemListener(this);
        gurup.add(plain);
        inputPanel.add(plain);
        bold=new JRadioButton("Bold");
        bold.setMnemonic('K');
        bold.setSelected(false);
        bold.addItemListener(this);
        gurup.add(bold);
        inputPanel.add(bold);

        italic=new JRadioButton("Italic");
        italic.setMnemonic('t');
        italic.setSelected(false);
        italic.addItemListener(this);
        gurup.add(italic);
        inputPanel.add(italic);

        bolditalic=new JRadioButton("Bold-Italic");

```

```

bolditalic.setMnemonic('a');
bolditalic.setSelected(false);
bolditalic.addItemListener(this);
gurup.add(bolditalic);
inputPanel.add(bolditalic);

        t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
inputPanel.add(t);

add(b, BorderLayout.NORTH);
add(t, BorderLayout.CENTER);
add(inputPanel, BorderLayout.SOUTH);
b.addActionListener(this);
}

```

```

public void itemStateChanged(ItemEvent e)
{   Object kutu=e.getItemSelectable();
if(kutu==plain)
{
    bold_font=Font.PLAIN;
    italic_font=Font.PLAIN;
}
else if(kutu==bold)
{
    bold_font=Font.BOLD;
    italic_font=Font.PLAIN;
}
else if(kutu==italic)
{
    bold_font=Font.PLAIN;
    italic_font=Font.ITALIC;
}
else if(kutu==bolditalic)
{
    bold_font=Font.BOLD;
    italic_font=Font.ITALIC;
}
t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
repaint();
}

```

```

public void actionPerformed( ActionEvent e)
{
//tesadüfi col belirle düğmeye her basılışta rengi değiştir.
kirmizi=(int)(255.0*Math.random());
yesil =(int)(255.0*Math.random());
mavi =(int)(255.0*Math.random());
col=new Color(kirmizi,yesil,mavi);
//input panelin rengini değiştiriyoruz
inputPanel.setBackground(col);
//ana panelin rengini değiştiriyoruz
setBackground(col);
//yazı kutusunun rengini değiştiriyoruz
t.setBackground(col);
plain.setBackground(col);
bold.setBackground(col);
italic.setBackground(col);
}

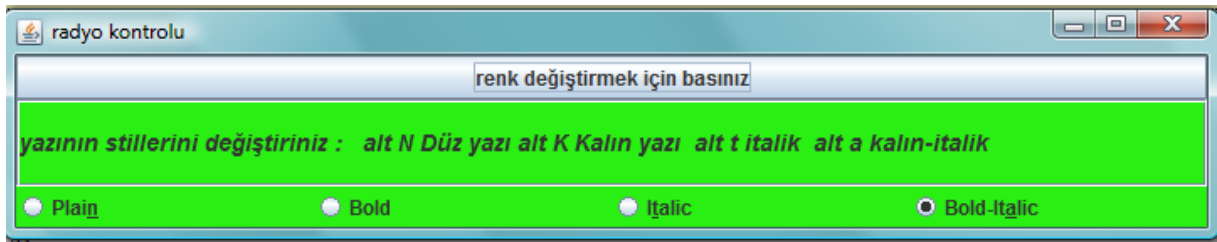
```

```

bolditalic.setBackground(col);
repaint();
}

public static void main(String s[])
{
    JFrame f = new JFrame("radyo kontrolu");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    JApplet applet = new radyo_kontrolu();
    f.getContentPane().add("Center", applet);
    applet.init();
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



8.6 LIST (LİSTE) VE JCOMBOBOX KONTROLU VE JSCROLLPANE FORMAT DEYİMİ

Bir seçim listesi oluşturmak için JList komutunu kullanabiliriz. JList komutu panel'e direkt eklenebilirse de JScrollPane format deyimiyle eklenirse daha güzel bir görüntü verir. JScrollPane formatı listenin bir kısmının görünüp, geri kalanın kayarak ekrana gelmesini sağlar. Dinleyici sınıfı olarak JList ListSelectionListener sınıfını kullanır. JList sınıfının kurucu metodları şunlardır:

public JList<E>();

public JList<E>(ListModel lm);

public JList<E>(Object o[]);

public JList<E>(Vector v);

Buradaki E herhangi bir sınıf olabilir. Bu yapıları genel tür değişken (Generic) ismini veriyoruz.

Bölüm 10.1 de genel tür değişkenler hakkında daha detaylı bilgi bulabilirsiniz. Örneğin JList<String> String sınıfından bir liste oluşturur.

Sınıfın çok kullanılan önemli metodlarının bazıları aşağıda listelenmiştir.

public int GetSelectedIndex() : seçilen liste nesnesinin index adresi (0 dan başlar)

public int[] getSelectedIndices() : birden fazla seçim yapıldığında seçilen listesi

public Object getSelectedValue() : seçilmiş değeri verir

public Object[] getSelectedValues() : seçilmiş değerleri verir

void setSelectionMode(int mode) : seçim modunu verir. Değişik mevcut seçim modları aşağıdaki gibidir.

ListSelectionModel.SINGLE_SELECTION (yalnızca tek bir seçime izin ver)

ListSelectionModel .INTERVAL_SELECTION (grup seçimine izin ver)

ListSelectionModel .MULTIPLE_INTERVAL_SELECTION (birden fazla gurup seçimine izin ver)

Public void setVisibleRowCount(int sayı) : ekranda görünen liste elemanı sayısı

Eğer ekranda görülen eleman sayısı toplam eleman sayısından azsa ve liste JScrollPane kullanılarak eklenmişse yan tarafta kayma çubuğu belirecektir. JScrollPane format sınıfının kurucu metodları aşağıda verilmiştir.

public JScrollPane();

public JScrollPane(int yan_çubuk_tipi,int type_of_column_scroll_bar);

public JScrollPane(Component JPanel_veya_bileşen_ismi);

public JScrollPane(Component JPanel_veya_bileşen_ismi , int yan_çubuk_tipi,int type_of_column_scroll_bar);

JPanel_veya_bileşen_ismi kayma çubuğunun tipini belirler, mevcut tipler şunlardır

JScrollPane.VERTICAL_SCROLLBAR_ALWAYS

JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED

JScrollPane.VERTICAL_SCROLLBAR_NEVER

JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS

JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED

JScrollPane.HORIZONTAL_SCROLLBAR_NEVER

Şimdi JList sınıfını bir örnekle inceleyelim. Programımız listeden renk seçmektedir. Renkler Color sınıfının alt sınıfı olarak tanımladığımız renk sınıfındaki sabit renkler olarak tanımlanmıştır.

Program 8.6.1 JList sınıf örnek programı listP

```
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class listP extends JPanel implements ListSelectionListener
{ private static final long serialVersionUID = 654677L;
  JPanel inputPanel;
  JList<String> renklistesi;
  String renkisimleri[]={ "kırmızı", "mavi", "siyah", "camgöbeği", "açık_camgöbeği", "koyugri", "gri", "yeşil", "açıkgri", "portakal",
  "pembe", "beyaz", "sarı", "açıkmavi", "lacivert", "mor", "turquaz", "menekşe", "mavi_menekşe", "patlıcan", "buğday", "ten", "gök_mavi",
  "açık_gök_mavisi", "deniz_kabuğu", "deniz_yeşili", "açık_deniz_yeşili", "zeytin_yeşili", "fildişi", "lavanta", "bej", "kahverengi",
  "çikolata", "mercan", "açık_yeşil", "alev_kırmızısı", "alev_turuncusu", "alizarin", "altuni", "ametist", "armut", "akuamarin",
  "asker_yeşili", "bakır", "bordo", "bronz", "çivit_mavisi", "gece_mavisi", "gül", "hardal", "havuç", "ilkbahar_yeşili", "kabak",
  "karanfil", "kayısı", "kehribar", "kobalt_mavisi", "krem", "limon", "açık_limon", "leylak", "mandalina", "nane_yeşili", "nar", "orkide",
  "pas", "safran", "safır", "tarçın", "turuncu", "zürüt_yeşili", "şarap"};
  renk
  renkler[]={ renk.kırmızı, renk.mavi, renk.siyah, renk.camgöbeği, renk.açık_camgöbeği, renk.koyugri, renk.gri, renk.yeşil, renk.açıkgri, re
```

```

nk.portakal,

renk.pembe,renk.beyaz,renk.sarı,renk.açık_mavi,renk.lacivert,renk.mor,renk.turquaz,renk.menekşe,renk.mavi_menekşe,renk.patlıcan,renk.buğday,renk.ten,renk.gök_mavi,

renk.açık_gök_mavisi,renk.deniz_kabuğu,renk.deniz_yeşili,renk.açık_deniz_yeşili,renk.zeytin_yeşili,renk.fildişi,renk.lavanta,renk.bej,renk.kahverengi,

renk.çikolata,renk.mercan,renk.açık_yeşil,renk.alev_kırmızısı,renk.alev_turuncusu,renk.alizarin,renk.altuni,renk.ametist,renk.armut,renk.akuamarin,

renk.asker_yeşili,renk.bakır,renk.bordo,renk.bronz,renk.çivit_mavisi,renk.gece_mavisi,renk.gül,renk.hardal,renk.havuç,renk.ilkbahar_yeşili,renk.kabak,

renk.karanfil,renk.kayısı,renk.kehribar,renk.kobalt_mavisi,renk.krem,renk.limon,renk.açık_limon,renk.leylak,renk.mandalina,renk.nane_yeşili,renk.nar,renk.orkide,renk.pas,renk.safran,renk.safir,renk.tarçın,renk.turuncu,renk.zümrüt_yeşili,renk.şarap};

public listP()
{
    renklistesi=new JList<String>(renkisimleri);
    renklistesi.setVisibleRowCount(5);
    inputPanel=new JPanel();
    inputPanel.add(new JScrollPane(renklistesi));
    add(inputPanel);
    renklistesi.addListSelectionListener(this);
}

public void valueChanged( ListSelectionEvent e)
{
    renk r1=renkler[renklistesi.getSelectedIndex()];
    inputPanel.setBackground(r1);
    setBackground(r1);
    repaint();
}
}

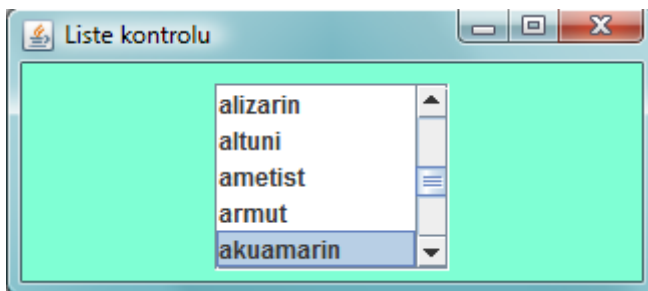
```

Program 8.6.2 JList test programı listF

```

class listF {
public static void main(String args[])
{FrameGrafigi.plot("Liste kontrolü",new listP());}}

```



İkinci örneğimizde listelerden biri rengi değiştirirken ikincisi de şekli değiştirmektedir.

Program 8.6.3 şekil P sınıfı

```
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class şekilP extends JPanel
{
    Color c;
    int şekil;
    //şekil = 0 elips
    //şekil = 1 yıldız

    public şekilP(Color c1,int şekil1)
    {c=c1;şekil=şekil1;}

    public void setColor(Color c1)
    {c=c1;}

    public void setşekil(int şekil1)
    {şekil=şekil1;}

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        if(şekil==0) //yıldız
        {
            int x[]={55,67,109,73,83,55,27,37,1,43};
            int y[]={0,36,36,54,96,72,96,54,36,36};
            Graphics2D g2=(Graphics2D)g;
            GeneralPath star=new GeneralPath();
            star.moveTo(x[0],y[0]);
            for(int i=1;i<x.length;i++)
                {star.lineTo(x[i],y[i]);}
            star.closePath();
            g2.setColor(c);
            g2.fill(star);
        }
        else if(şekil==1) //elips
        {
            Graphics2D g2=(Graphics2D)g;
            g2.setFont(new Font("Serif",Font.BOLD,24));
            g2.setColor(c);
            g2.setStroke(new BasicStroke(2.0f));
            Ellipse2D x=new Ellipse2D.Double(50,50,500,200);
            g2.fill(x);
        }
    }
}
```

Program 8.6.4 listA1 JApplet class : an example of JList class

```
import java.io.*;
import java.applet.Applet;
```

```

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class listA1 extends JApplet implements ListSelectionListener
{ private static final long serialVersionUID = 5656754L;
  sekiIP sekiI;
  JPanel inputPanel;
  JPanel outputPanel;
  JList<String> sekiIlistesi;
  JList<String> renkIlistesi;
  String sekiIisimleri[]={ "dolu yıldız","dolu elips"};
  String
renkisimleri[]={ "kırmızı","mavi","siyah","camgöbeği","açık_camgöbeği","koyugri","gri","yeşil","açıkgri","portakal",

"pembe","beyaz","sarı","açıkmavi","lacivert","mor","turquaz","menekşe","mavi_menekşe","patlıcan","buğday","ten","gök_
mavi",

"açık_gök_mavisi","deniz_kabuğu","deniz_yeşili","açık_deniz_yeşili","zeytin_yeşili","fildişi","lavanta","bej","kahverengi",
"çikolata","mercan","açık_yeşil","alev_kırmızısı","alev_turuncusu","alizarin","altuni","ametist","armut","akuamarin",
"asker_yeşili","bakır","bordo","bronz","çivit_mavisi","gece_mavisi","gül","hardal","havuç","ilkbahar_yeşili","kabak",

"karanfil","kayısı","kehribar","kobalt_mavisi","krem","limon","açık_limon","leylak","mandalina","nane_yeşili","nar","orkid
e",
"pas","safran","safir","tarçın","turuncu","zümrüt_yeşili","şarap"};
  renk
renkler[]={renk.kırmızı,renk.mavi,renk.siyah,renk.camgöbeği,renk.açık_camgöbeği,renk.koyugri,renk.gri,renk.yeşil,renk.açı
kgri,renk.portakal,

renk.pembe,renk.beyaz,renk.sarı,renk.açıkmavi,renk.lacivert,renk.mor,renk.turquaz,renk.menekşe,renk.mavi_menekşe,renk.
patlıcan,renk.buğday,renk.ten,renk.gök_mavi,

renk.açık_gök_mavisi,renk.deniz_kabuğu,renk.deniz_yeşili,renk.açık_deniz_yeşili,renk.zeytin_yeşili,renk.fildişi,renk.lavanta,renk.bej,renk.kahverengi,

renk.çikolata,renk.mercan,renk.açık_yeşil,renk.alev_kırmızısı,renk.alev_turuncusu,renk.alizarin,renk.altuni,renk.ametist,renk
.armut,renk.akuamarin,

renk.asker_yeşili,renk.bakır,renk.bordo,renk.bronz,renk.çivit_mavisi,renk.gece_mavisi,renk.gül,renk.hardal,renk.havuç,renk.
ilkbahar_yeşili,renk.kabak,

renk.karanfil,renk.kayısı,renk.kehribar,renk.kobalt_mavisi,renk.krem,renk.limon,renk.açık_limon,renk.leylak,renk.mandalina,renk.nane_yeşili,renk.nar,renk.orkide,
renk.pas,renk.safran,renk.safir,renk.tarçın,renk.turuncu,renk.zümrüt_yeşili,renk.şarap};

  public void init()
  {
    int i=0;
    sekiI=new sekiIP(renk.mavi,i);
    renkIlistesi=new JList<String>(renkisimleri);
    renkIlistesi.setVisibleRowCount(5);
  }

```



```

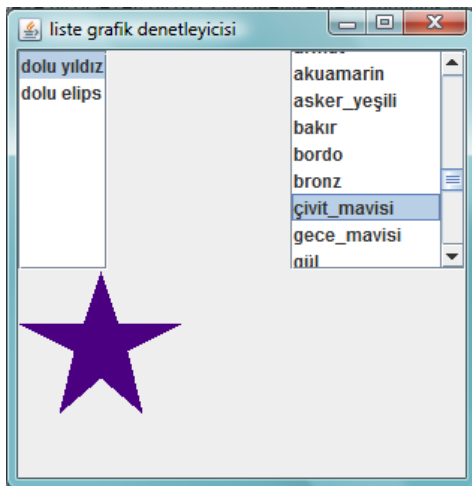
        sekillistesi=new JList<String>(sekil isimleri);
inputPanel=new JPanel();
inputPanel.setLayout(new BorderLayout());
outputPanel=new JPanel();
outputPanel.setLayout(new BorderLayout());
outputPanel.add(sekil,BorderLayout.CENTER);
inputPanel.add(new JScrollPane(renklistesi),BorderLayout.EAST);
inputPanel.add(new JScrollPane(sekillistesi),BorderLayout.WEST);
add(inputPanel,BorderLayout.NORTH);
add(outputPanel,BorderLayout.CENTER);
renklistesi.addListSelectionListener(this);
sekillistesi.addListSelectionListener(this);
}

public void valueChanged( ListSelectionEvent e)
{
if(e.getSource()==renklistesi)
{
renk r1=renkler[renklistesi.getSelectedIndex()];
sekil.setColor(r1);
}
if(e.getSource()==sekillistesi)
{sekil.setsekil(sekillistesi.getSelectedIndex());}

repaint();
}

public static void main(String s[])
{
JFrame f = new JFrame("liste grafik denetleyicisi");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new listA1();
f.getContentPane().add("Center", applet);
applet.init();
f.setSize(800,500);
f.setVisible(true);
}
}

```



JList ve JComboBox kullanım olarak benzer metodlardır. JComboBox kurucu metodları şunlardır:

```
public JComboBox<E>();  
public JComboBox <E>(ListModel lm);  
public JComboBox <E>(E[] girdi_nesneleri);  
public JList<E>(Vector<E> girdi_nesneleri);
```

Buradaki E herhangi bir sınıf olabilir. Bu yapıları genel tür değişken (Generic) ismini veriyoruz. Bölüm 10.1 de genel tür değişkenler hakkında daha detaylı bilgi bulabilirsiniz. Örneğin JList<String> String sınıfından bir liste oluşturur.

JList'te verdiğimiz aynı örneği JComboBox olarak tekrar verelim:

Program 8.6.5 JComboBox sınıf örnek programı comboP

```
import java.io.*;  
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
import java.util.*;  
import javax.swing.*;  
import javax.swing.table.*;  
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.event.*;  
import java.awt.geom.*;  
  
public class comboP extends JPanel implements ActionListener  
{ private static final long serialVersionUID =76874L;  
  JPanel inputPanel;  
  JComboBox<String> renklistesi;  
  String  
  renk isimleri[]={ "kırmızı", "mavi", "siyah", "camgöbeği", "açık_camgöbeği", "koyugri", "gri", "yeşil", "açıkgri", "portakal",  
  "pembe", "beyaz", "sarı", "açıkmavi", "lacivert", "mor", "turquaz", "menekşe", "mavi_menekşe", "patlıcan", "buğday", "ten",  
  "gök_mavi", "açık_gök_mavisi", "deniz_kabuğu", "deniz_yeşili", "açık_deniz_yeşili", "zeytin_yeşili", "fildişi", "lavanta", "bej", "k  
  ahverengi", "çikolata", "mercan", "açık_yeşil", "alev_kırmızısı", "alev_turuncusu", "alizarin", "altuni", "ametist", "armut", "akuama  
  rin", "asker_yeşili", "bakır", "bordo", "bronz", "çivit_mavisi", "gece_mavisi", "gül", "hardal", "havuç", "ilkbahar_yeşili", "kabak",  
  "karanfil", "kayısı", "kehribar", "kobalt_mavisi", "krem", "limon", "açık_limon", "leylak", "mandalina", "nane_yeşili", "nar",  
  "orkide", "pas", "safran", "safır", "tarçın", "turuncu", "zümrüt_yeşili", "şarap"};  
  renk renkleri[] =  
  {renk.kırmızı, renk.mavi, renk.siyah, renk.camgöbeği, renk.açık_camgöbeği, renk.koyugri, renk.gri, renk.yeşil, renk.açıkgri,  
  renk.portakal, renk.pembe, renk.beyaz, renk.sarı, renk.açıkmavi, renk.lacivert, renk.mor, renk.turquaz, renk.menekşe,  
  renk.mavi_menekşe, renk.patlıcan, renk.buğday, renk.ten, renk.gök_mavi, renk.açık_gök_mavisi, renk.deniz_kabuğu,  
  renk.deniz_yeşili, renk.açık_deniz_yeşili, renk.zeytin_yeşili, renk.fildişi, renk.lavanta, renk.bej, renk.kahverengi, renk.çikolata,  
  renk.mercan, renk.açık_yeşil, renk.alev_kırmızısı, renk.alev_turuncusu, renk.alizarin, renk.altuni, renk.ametist, renk.armut,  
  renk.akuamarin, renk.asker_yeşili, renk.bakır, renk.bordo, renk.bronz, renk.çivit_mavisi, renk.gece_mavisi, renk.gül, renk.hardal  
  , renk.havuç, renk.ilkbahar_yeşili, renk.kabak, renk.karanfil, renk.kayısı, renk.kehribar, renk.kobalt_mavisi, renk.krem, renk.limon,  
  renk.açık_limon, renk.leylak, renk.mandalina, renk.nane_yeşili, renk.nar, renk.orkide, renk.pas, renk.safran, renk.safır, renk.tarçın,  
  renk.turuncu, renk.zümrüt_yeşili, renk.şarap};  
  public comboP()  
  {  
    renklistesi=new JComboBox<String>(renk isimleri);  
    //renklistesi.setVisibleRowCount(5);  
    inputPanel=new JPanel();  
    inputPanel.add(new JScrollPane(renklistesi));  
    add(inputPanel);  
    renklistesi.addActionListener(this);  
  }  
  public void actionPerformed( ActionEvent e)  
  {  
    String srenk= (String)renklistesi.getSelectedItem();
```

```

renk r1=renkler[renklistesi.getSelectedIndex()];
inputPanel.setBackground(r1);
setBackground(r1);
repaint();
}}

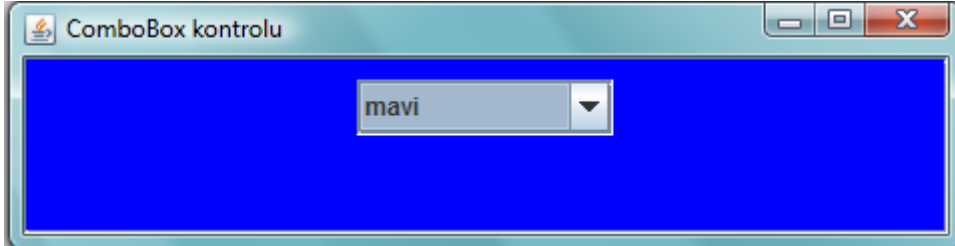
```

Program 8.6.6 JComboBox test programı comboF

```

class comboF {
public static void main(String args[])
{FrameGrafigi.plot("ComboBox kontrolu",new comboP());}
}

```



JComboBox'ın JList'ten en önemli farkı alan'a JTextField'de olduğu gibi girdi verebilmemiz ve hem JList hem de JTextField gibi kullanabilmemizdir. Edit edilebilir bir formdadır. Bunu sağlayabilmek için edit edilebilirlik özelliğini **setEditable(true)** komutuyla açmamız gerekir. Bir sonraki programda üstte yazdığımız renk seçim programının edit edilebilir formu verilmiştir. Burada JComboBox kutusuna bir renk ismi girilebilir. Program eğer girilen isim renk olarak mevcut değilse beyaz rengin seçilmesi şeklinde programlanmıştır.

Program 8.6.7 yazarak girdi verilebilen JComboBox sınıf örnek programı comboP1

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class comboP1 extends JPanel implements ActionListener
{ private static final long serialVersionUID =76874L;
  JPanel inputPanel;
  JComboBox<String> renklistesi;
  String
renkisimleri[]={ "kırmızı", "mavi", "siyah", "camgöbeği", "açık_camgöbeği", "koyugri", "gri", "yeşil", "açıkgri", "portakal",
"pembe", "beyaz", "sarı", "açıkmavi", "lacivert", "mor", "turquaz", "menekşe", "mavi_menekşe", "patlıcan", "buğday", "ten",
"gök_mavi", "açık_gök_mavisi", "deniz_kabuğu", "deniz_yeşili", "açık_deniz_yeşili", "zeytin_yeşili", "fildişi", "lavanta", "bej", "k
ahverengi", "çikolata", "mercan", "açık_yeşil", "alev_kırmızısı", "alev_turuncusu", "alizarin", "altuni", "ametist", "armut", "akuama
rin", "asker_yeşili", "bakır", "bordo", "bronz", "çivit_mavisi", "gece_mavisi", "gül", "hardal", "havuç", "ilkbahar_yeşili", "kabak",
"karanfil", "kayısı", "kehribar", "kobalt_mavisi", "krem", "limon", "açık_limon", "leylak", "mandalina", "nane_yeşili", "nar",
"orkide", "pas", "safran", "safir", "tarçın", "turuncu", "zümrüt_yeşili", "şarap"};
  renk renkler[] = {renk.kırmızı, renk.mavi, renk.siyah, renk.camgöbeği, renk.açık_camgöbeği, renk.koyugri, renk.gri,
renk.yeşil, renk.açıkgri, renk.portakal, renk.pembe, renk.beyaz, renk.sarı, renk.açıkmavi, renk.lacivert, renk.mor, renk.turquaz,
renk.menekşe, renk.mavi_menekşe, renk.patlıcan, renk.buğday, renk.ten, renk.gök_mavi, renk.açık_gök_mavisi,
renk.deniz_kabuğu, renk.deniz_yeşili, renk.açık_deniz_yeşili, renk.zeytin_yeşili, renk.fildişi, renk.lavanta, renk.bej,

```

```
renk.kahverengi,renk.çikolata,renk.mercan,renk.açık_yeşil,renk.alev_kırmızısı,renk.alev_turuncusu,renk.alizarin,renk.altuni,renk.ametist,renk.armut,renk.akuamarin,renk.asker_yeşili,renk.bakır,renk.bordo,renk.bronz,renk.çivit_mavisi,renk.gece_mavisi,renk.gül,renk.hardal,renk.havuç,renk.ilkbahar_yeşili,renk.kabak,renk.karanfil,renk.kayı,renk.kehribar,renk.kobalt_mavisi,renk.krem,renk.limon,renk.açık_limon,renk.leylak,renk.mandalina,renk.nane_yeşili,renk.nar,renk.orkide,renk.pas,renk.safran,renk.safir,renk.tarçın,renk.turuncu,renk.zümrüt_yeşili,renk.şarap};
```

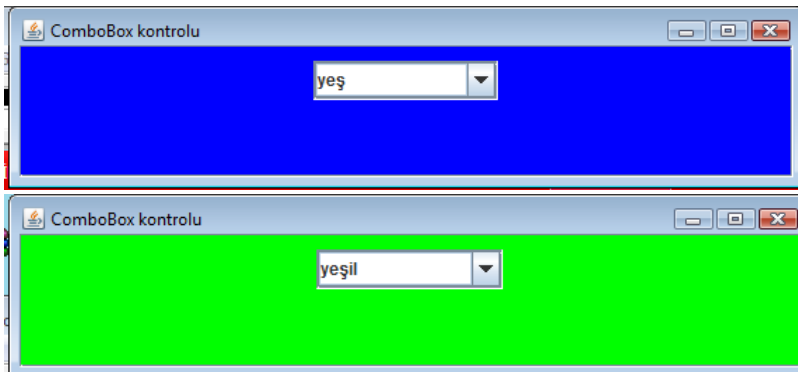
```
public renk renkbul(String r)
{for(int i=0;i<renkisimleri.length;i++)
{if(r.equals(renkisimleri[i])) return renkleri[i];}
return renk.beyaz;
}
```

```
public comboP1()
{
renklistesi=new JComboBox<String>(renkisimleri);
renklistesi.setEditable(true);
inputPanel=new JPanel();
inputPanel.add(new JScrollPane(renklistesi));
add(inputPanel);
renklistesi.addActionListener(this);
}
```

```
public void actionPerformed( ActionEvent e)
{
if(e.getSource()==renklistesi)
{
JComboBox<String> cb = (JComboBox<String>)e.getSource();
String srenk= (String)cb.getSelectedItem();
renk r1=renkbul(srenk);
inputPanel.setBackground(r1);
setBackground(r1);
repaint();
}}}
```

Program 8.6.8 yazılabilen JComboBox test programı comboF1

```
class comboF1 {
public static void main(String args[])
{FrameGrafigi.plot("ComboBox kontrolu",new comboP1());}}
```



Program her seçimde yeni bir JComboBox oluşturmakta ve rengi buradan seçmektedir. Girdiyi hem listeden seçmek hem de JTextField’de olduğu gibi yazarak girmek mümkündür.

JComboBox kontrolunu **ComboBoxModel** sınıfı üzerinden de tanımlayarak kullanabiliriz.

Program 8.6.9 yazarak girdi verilebilen ve JComboBoxModel üzerinden tanımlanmış JComboBox sınıf örnek programı comboP2

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

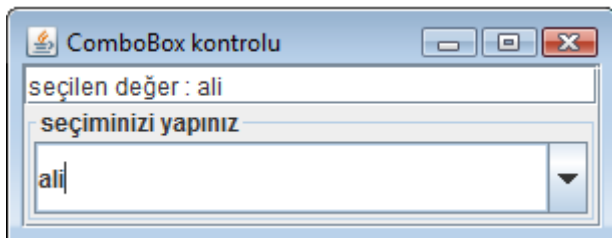
public class comboP2 extends JPanel implements ActionListener
{
    String[] s = { "1", "2", "3", "4", "5", "6", "7", "8" };
    JComboBoxModel<String> mode;
    JTextField jtf;
    public comboP2() {
        JFrame f = new JFrame("JComboBox2");
        Container contentPane = f.getContentPane();
        jtf=new JTextField(s[0]);
        mode = new UserDefineComboBoxModel();
        JComboBox<String> combo = new JComboBox<String>(mode);
        combo.setEditable(true);
        combo.setBorder(BorderFactory.createTitledBorder("seçiminizi yapınız"));
        setLayout(new BorderLayout());
        add(jtf,BorderLayout.NORTH);
        add(combo,BorderLayout.CENTER);
        combo.addActionListener(this);
    }

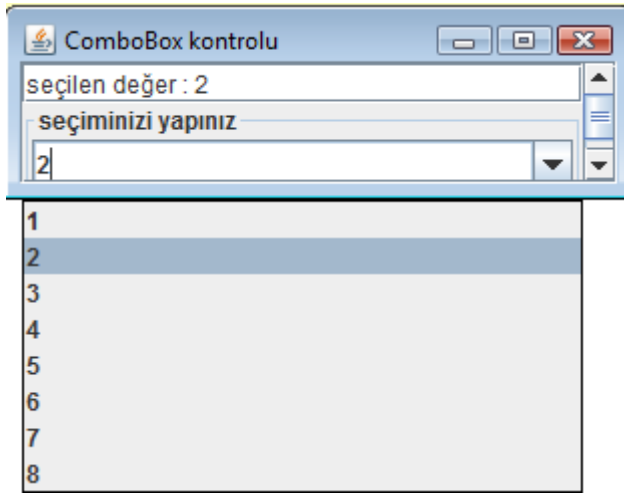
    public void actionPerformed( ActionEvent e)
    { JComboBox<String> cb = (JComboBox<String>)e.getSource();
      String srenk= (String)cb.getSelectedItem();
      jtf.setText("seçilen değer : "+srenk);
    }

    class UserDefineComboBoxModel<String> extends AbstractListModel implements
      ComboBoxModel {
        String item = null;
        public Object getElementAt(int index)
        {return s[index++];}
        public int getSize()
        {return s.length;}
        public void setSelectedItem(Object anItem) {item = (String) anItem;}
        public Object getSelectedItem() {return item;}
    }
}
```

Program 8.6.10 yazılabilen JComboBox test programı comboF2

```
class comboF2 {
    public static void main(String args[])
    {FrameGrafigi.plot("ComboBox kontrolu",new comboP2());}
}
```





Aynı programın JComboBox'a yazılı girdi verilemeyen versiyonunu da ekleyelim.

Program 8.6.11 JComboBoxModel üzerinden tanımlanmış JComboBox sınıf örnek programı comboP23 (bu versiyonda yazarak girdi verilemez)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class comboP3 extends JPanel implements ActionListener
{
    String[] s = { "1", "2", "3", "4", "5", "6", "7", "8" };
    JComboBoxModel<String> mode;
    JComboBox<String> combo;
    JTextField jtf;
    public comboP3() {
        JFrame f = new JFrame("JComboBox ");
        Container contentPane = f.getContentPane();
        jtf=new JTextField(s[0]);
        mode = new UserDefineComboBoxModel();
        combo = new JComboBox<String>(mode);
        combo.setBorder(BorderFactory.createTitledBorder("seçiminizi yapınız"));
        setLayout(new BorderLayout());
        add(jtf,BorderLayout.NORTH);
        add(combo,BorderLayout.CENTER);
        combo.addActionListener(this);
    }

    public void actionPerformed( ActionEvent e)
    { String srenk= (String)combo.getSelectedItem();
      jtf.setText("seçilen değer : "+srenk);
    }

    class UserDefineComboBoxModel<String> extends AbstractListModel implements
        JComboBoxModel {
        String item = null;
        public Object getElementAt(int index)
        {return s[index++];}
        public int getSize()
        {return s.length;}
        public void setSelectedItem(Object anItem) {item = (String) anItem;}
        public Object getSelectedItem() {return item;}
    }
}
```

```
}
```

Program 8.6.12 JComboBox test programı comboF3

```
class comboF2 {  
public static void main(String args[])  
{FrameGrafigi.plot("ComboBox kontrolu",new comboP2());}}
```



8.7 MENU KONTROLU

Eğer pencere (window) tipi programlar kullanıyorsak menülerle haşır neşiriz demektir. Genelde window tipi programların üst tarafında bulunan açılır kapanır tür liste yapılarına menü ismi verilir. Java menüleri JMenuBar, JMenu ve JMenuItem sınıflarından oluşur. Menülerin içinde, JRadioButtonMenuItem, JCheckBoxItem gibi elemanlar da kullanılabilir. **JMenuBar** kurucu metodu:

```
public JMenuBar();
```

JMenuBar sınıfının önemli metodları :

```
public JMenu add(JMenu menu); :
```

```
public Component getComponent();
```

```
public Component getComponentAtIndex(int indeks);
```

```
public Insets getMargin();
```

```
public JMenu getMenu(int)
```

```
public int getMenuCount();
```

```
public SingleSelectionModel getSelectionModel();
```

```
public MenuElement[] getSubElements();
```

```
protected void paintBorder(Graphics g);
```

```
public void processMouseEvent(MouseEvent e,MouseElement path[],MouseSelectionManager mgr);
```

```
public void setBorderPainted(boolean b);
```

```
public void setHelpMenu(Jmenu menu);
```

```
public void setMargin(Insets in);
```

```
public void setSelected(Component c);
```

Jmenu kurucu metodları:

```
public JMenu();
```

```
public JMenu(String Label);
```

```
public JMenu(String label,boolean menudenayır);
```

JMenu sınıfının önemli metodları şunlardır:

```
public JMenuItem add(Action act);
```

```
public JMenuItem add(JMenuItem item);
```

```
public Component add(Component c);
```

```
public void add(String Jlabel);
```

```
public void addMenuListener(MenuListener l);  
public void addSeperator();  
public JMenuItem getItem(int pos);  
public int getItemCount();  
public Component getMenuComponent(int pos);  
public int getMenuComponentCount();  
public Component[] getMenuComponents(public MenuElement[] getSubElements());  
public JMenuItem insert(JMenuItem item,int pos);  
public void insert(String etiket,int pos);  
public void remove(int pos);  
public void remove(MenuItem item);  
public void removeAll();  
public void removeMenuListener(MenuListener l);  
public void setAccelerator(KeyStroke s);  
public void setDealy(int delay);  
public void setMenuLocation(int x,int y);  
public void setPopUpMenuVisible(boolean b)  
public void setSelected(boolean b);
```

JMenuItem sınıfı kurucu metodları

```
public JMenuItem();  
public JMenuItem(Icon picture);  
public JMenuItem(String Label);  
public JMenuItem(String Label,int mnemonic);  
public JMenuItem(String Label,Icon picture);
```

JMenuItem sınıfı önemli metodları:

```
public Component getComponent();  
public menuElement[] getSubElements();
```

JCheckBoxMenuItem, JMenuItem sınıfının alt metodudur. JCheckBox sınıfına benzer davranır.

Kurucu metodları şunlardır :

```
public JCheckBoxMenuItem();  
public JCheckBoxMenuItem(Icon pictue);  
public JCheckBoxMenuItem(String Label);  
public JCheckBoxMenuItem(String Label,Icon picture);  
public JCheckBoxMenuItem(String Label,Boolean isselected);  
public JCheckBoxMenuItem(String Label,Icon picture, Boolean isselected);
```

JRadioButtonMenuItem da JMenuItem sınıfının alt sınıfıdır. Kurucu metodları :

```
public JRadioButtonMenuItem ();  
public JRadioButtonMenuItem (Icon picture);  
public JRadioButtonMenuItem (String Label);  
public JRadioButtonMenuItem (String Label,Icon picture);
```

Şimdi bazı menü örnekleri oluşturarak inceleyelim:

Program 8.7.1 Menu1


```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;
import java.awt.Graphics;

public class Menu1 extends JFrame implements ListSelectionListener
{
    JPanel girdipaneli;
    JMenuBar menuBar;
    JMenu m1,m2;
    JList renklisterisi;
    String
renkisimleri[]={ "kırmızı","mavi","siyah","camgöbeği","açık_camgöbeği","koyugri","gri","yeşil","açıkgri","portakal",

"pembe","beyaz","sarı","açıkmavi","lacivert","mor","turquaz","menekşe","mavi_menekşe","patlıcan","buğday","ten","gök_
mavi",

"açık_gök_mavisi","deniz_kabuğu","deniz_yeşili","açık_deniz_yeşili","zeytin_yeşili","fildişi","lavanta","bej","kahverengi",
"çikolata","mercan","açık_yeşil","alev_kırmızısı","alev_turuncusu","alizarin","altuni","ametist","armut","akuamarin",
"asker_yeşili","bakır","bordo","bronz","çivit_mavisi","gece_mavisi","gül","hardal","havuç","ilkbahar_yeşili","kabak",

"karanfil","kayısı","kehribar","kobalt_mavisi","krem","limon","açık_limon","leylak","mandalina","nane_yeşili","nar","orkid
e",
"pas","safran","safir","tarçın","turuncu","zümrüt_yeşili","şarap"};
renk
renkler[]={renk.kırmızı,renk.mavi,renk.siyah,renk.camgöbeği,renk.açık_camgöbeği,renk.koyugri,renk.gri,renk.yeşil,renk.açı
kgri,renk.portakal,

renk.pembe,renk.beyaz,renk.sarı,renk.açıkmavi,renk.lacivert,renk.mor,renk.turquaz,renk.menekşe,renk.mavi_menekşe,renk.
patlıcan,renk.buğday,renk.ten,renk.gök_mavi,

renk.açık_gök_mavisi,renk.deniz_kabuğu,renk.deniz_yeşili,renk.açık_deniz_yeşili,renk.zeytin_yeşili,renk.fildişi,renk.lavanta,
renk.bej,renk.kahverengi,

renk.çikolata,renk.mercan,renk.açık_yeşil,renk.alev_kırmızısı,renk.alev_turuncusu,renk.alizarin,renk.altuni,renk.ametist,renk
.armut,renk.akuamarin,

renk.asker_yeşili,renk.bakır,renk.bordo,renk.bronz,renk.çivit_mavisi,renk.gece_mavisi,renk.gül,renk.hardal,renk.havuç,renk.
ilkbahar_yeşili,renk.kabak,

renk.karanfil,renk.kayısı,renk.kehribar,renk.kobalt_mavisi,renk.krem,renk.limon,renk.açık_limon,renk.leylak,renk.mandalin
a,renk.nane_yeşili,renk.nar,renk.orkide,
renk.pas,renk.safran,renk.safir,renk.tarçın,renk.turuncu,renk.zümrüt_yeşili,renk.şarap};

public Menu1()
{
    super("Menugrafik kontrolü");
    girdipaneli=new JPanel();
    renklisterisi=new JList(renkisimleri);
    renklisterisi.setVisibleRowCount(10);

```

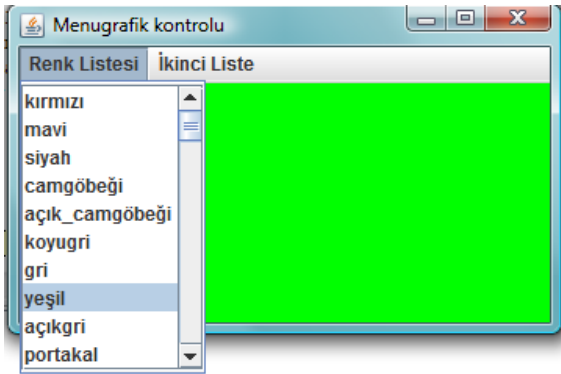
```

renklistesi.addListSelectionListener(this);
menuBar=new JMenuBar();
    setJMenuBar(menuBar);
    m1 = new JMenu("Renk Listesi");
m1.setMnemonic('M');
m1.getAccessibleContext().setAccessibleDescription("Programdaki sadece bu Menunun alt listeleri mevcuttur");
m1.add(new JScrollPane(renklistesi));
menuBar.add(m1);
m2 = new JMenu("İkinci Liste");
menuBar.add(m2);
add(girdipaneli);
}

public void valueChanged( ListSelectionEvent e)
{
renk r1=renkler[renklistesi.getSelectedIndex()];
girdipaneli.setBackground(r1);
}

public static void main(String s[])
{
    Menu1 f = new Menu1();
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



Program 8.7.2 Menu2 menü kontrol programı

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;
import java.awt.Graphics;

```

```

public class Menu2 extends JFrame implements ListSelectionListener
{
    Container c;
    shapeP sekil;
    JMenuBar menuBar;
    JMenu m1,m2;
    JPanel outputPanel;
    JList sekillistesi;
    JList renklistesi;
    String sekilisimleri[]={ "Filled Star", "Filled Ellipse"};
    String renk isimleri[]={ "black", "blue", "cyan", "dark gray", "gray",
        "green", "light grey", "magenta", "orange", "pink", "red", "white", "yellow"};
    Color renkler[]={ Color.black,Color.blue,Color.cyan,Color.darkGray,Color.gray,
        Color.green,Color.lightGray,Color.magenta,Color.orange,Color.pink,
        Color.red,Color.white,Color.yellow};

    public Menu2()
    {
        super("Menu Graphic controller");
        int i=0;
        sekil=new shapeP(Color.black,i);
        renklistesi=new JList(renk isimleri);
        renklistesi.setVisibleRowCount(5);
        sekillistesi=new JList(sekil isimleri);
        c=getContentPane();
        outputPanel=new JPanel();
        outputPanel.setLayout(new BorderLayout());
        outputPanel.add(sekil, BorderLayout.CENTER);
        menuBar=new JMenuBar();
        setJMenuBar(menuBar);
        m1 = new JMenu("Color List");
        m1.setMnemonic('R');
        m1.getAccessibleContext().setAccessibleDescription("Şekillerin hangi renkte olacağını seçer");
        m1.add(new JScrollPane(renklistesi));
        menuBar.add(m1);
        m2 = new JMenu("Shape List");
        m2.setMnemonic('e');
        m2.getAccessibleContext().setAccessibleDescription("Şekillerin hangi şekil olacağını seçer");
        m2.add(new JScrollPane(sekillistesi));
        menuBar.add(m2);
        c.add(outputPanel, BorderLayout.CENTER);
        renklistesi.addListSelectionListener(this);
        sekillistesi.addListSelectionListener(this);
    }

    public void valueChanged( ListSelectionEvent e)
    {
        if(e.getSource()==renklistesi)
        {
            Color r1=renkler[renklistesi.getSelectedIndex()];
            sekil.setColor(r1);
        }
        if(e.getSource()==sekillistesi)
        {
            sekil.setShape(sekillistesi.getSelectedIndex());
        }

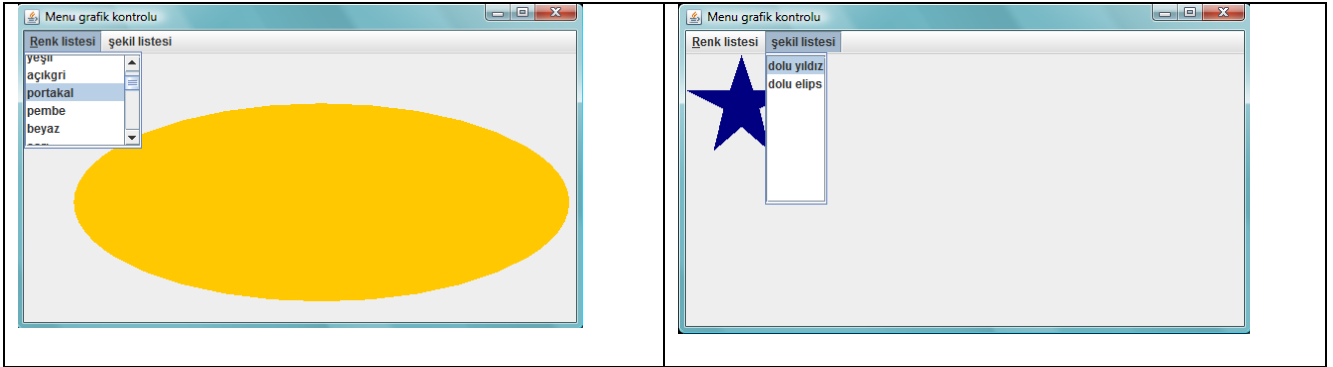
        repaint();
    }
}

```

```

public static void main(String s[])
{
    Menu2 f = new Menu2();
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



8.8 JSLIDER GRAFİK KONTROLU

Kayma çubuğu, JSlider verilen sınırlar içinde bir değişkenin çubuğu kaydırılarak kontrol edilmesi amacıyla kullanılır. Örneğin % (yüzde) olarak değişen bir değişkenin kontrolü için uygun bir yöntemdir. JSlider sınıfının kurucu metodları şunlardır:

```

public JSlider();
public JSlider(BoundedRangeModel brm);
public JSlider(int konum);
public JSlider(int minimumdeğer, int maksimumdeğer);

```

Örnek programımızda JSlider sınıfını kullanarak bir dairenin çapını değiştireceğiz.

Program 8.8.1 ovalP , daire çizen bir JPanel sınıfı

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.geom.*;

public class ovalP extends JPanel
{
    int xi,yi,radius;

    public ovalP(int xi1, int yi1,int radius1)
    {
        xi=xi1;
        yi=yi1;
        radius=(radius1 >= 0 ? radius1:10);
    }

    public void setOvalP(int xi1, int yi1,int radius1)
    {
        xi=xi1;
        yi=yi1;
    }
}

```

```

radius=(radius1 >= 0 ? radius1:10);
repaint();
}

public void paintComponent(Graphics g)
{
super.paintComponent(g);
g.setColor(Color.BLUE);
Graphics2D g2=(Graphics2D)g;
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
Ellipse2D elips1=new Ellipse2D.Double(xi,yi,radius,radius);
g2.draw(elips1);
}
}

```

Program 8.8.2 kaymacubugutesti JSlider test programı

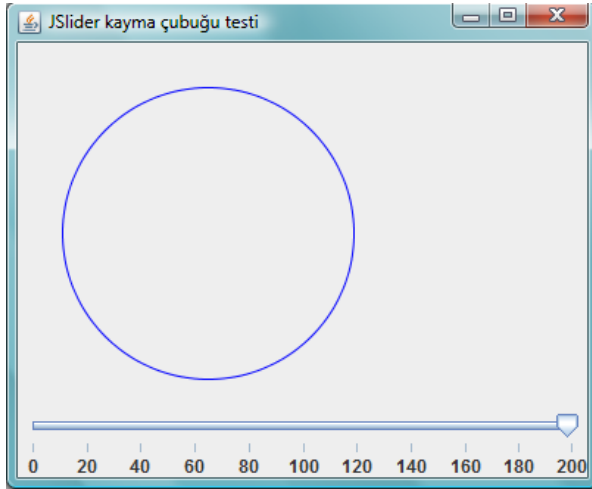
```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;

public class kaymacubugutesti extends JFrame implements ChangeListener
{
//
private ovalP p;
private JSlider cap;
Color r=Color.lightGray;
// pencereyi baslatma metodu
public kaymacubugutesti()
{
super("JSlider kayma çubuğu testi");
p=new ovalP(30,30,30);
cap=new JSlider(SwingConstants.HORIZONTAL,0,200,10);
cap.setMajorTickSpacing(20);
cap.setPaintTicks(true);
cap.setPaintLabels(true);
cap.addChangeListener(this);
add(p,BorderLayout.CENTER);
add(cap,BorderLayout.SOUTH);
}
// girdi alanındaki olan olaylari dinleme metodu
public void stateChanged(ChangeEvent e)
{
p.setOvalP(30, 30,cap.getValue());
repaint();
}

//=====
public static void main(String[] args)
{
kaymacubugutesti f= new kaymacubugutesti();
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
f.setSize(350,200);
f.setVisible(true);
}
}

```



İkinci bir örnek olarak analog resim olarak sıcaklık gösteren termometrelerin sıcaklık girdisini JSlider ile kontrol eden bir program örneği verelim. Bu program gerçek bir uygulama için hazırlanmıştır. Bu yüzden de diğer uygulamalara göre size biraz daha kompleks görülebilir. Dial kontrol polar koordinat sisteminde verilmiş bir ibreyi döndürmektedir:

Program 8.8.3 polar, polar koordinat sistemi tanımı

```
public class polar
{
// polar coordinates
// teta radyan
protected double R,teta;

public polar()
{
R=0;
teta=0;
}

public polar(double Ri,double tetai)
{
R=Ri;
teta=tetai;
}

public polar(polar y)
{
R=y.ROku();
teta=y.tetaOku();
}

public void R_input(double Ri)
{R=Ri;}

public void teta_input(double tetai)
{teta=tetai;}

public void polar_input(double Ri,double tetai)
{R=Ri;
teta=tetai;}
}
```

```

public void kartesien_input(double xi,double xj)
{R=Math.sqrt(xi*xi+xj*xj);
teta=Math.atan2(xj,xi);}

public double ROku()
{return R;}

public double tetaOku()
{return teta;}

public double xi()
{return R*Math.cos(teta);}

public double xj()
{return R*Math.sin(teta);}

public void topla(polar y)
{kartesien_input((xi()+y.xi()),(xj()+y.xj()));}

public void fark(polar y)
{ kartesien_input((xi()-y.xi()),(xj()-y.xj()));}

public polar polarOku()
{return this;}

public boolean esittir(polar v)
{boolean b=((R==v.ROku())&&(teta==v.tetaOku()));
return b;}

public boolean buyuktur(polar v)
{return (this.R>v.ROku());}

public boolean kucuktur(polar v)
{ return (this.R<v.ROku());}
}

```

Program 8.8.4 analogDialP analog gösterge tanımı

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;
import java.awt.Graphics;
import java.util.*;
import java.text.*;
import java.util.Locale;
import java.awt.image.*;

import java.awt.Font;
import java.awt.Graphics2D;
import java.awt.RenderingHints;

```

```

import java.awt.geom.*;
import java.awt.font.*;

// =====
public class analogDialP extends JComponent
{
    double x,minx,maxx;
    double minteta,maxteta;
    int n;
    int n1;
    polar p;
    double angle1,angle2,angle3;
    String name;
    String unit;
    public analogDialP(String namei,String uniti,double minxi,double maxx,double mintetai,double maxtetai,int ni,int n1i)
    {
        name=namei;
        unit=uniti;
        minx=minxi;
        maxx=maxxi;
        minteta=mintetai;
        maxteta=maxtetai;
        x=minx;
        n=ni;
        n1=n1i;
        angle1=minteta;
        angle2=maxteta;
    }

    public void setx(double xi)
    {x=xi;}

    public void drawPlotLabel(Graphics g2,String s, int xi,int yi,int R1i)
    {Graphics2D g=(Graphics2D)g2;
    int x1,y1;
    int R1=R1i;
    x1=xi+R1;
    y1=yi+R1;
    g.drawString(s,x1,y1);
    //AffineTransform origTransform = g.getTransform();
    //g.translate(x1,y1);
    //g.rotate(90*Math.PI/180);
    //g.drawString(s,0,0);
    //g.setTransform(origTransform);
    //g.drawString("xxxxxxxxxxxx",0,0);
    }

    public void drawLabel1(Graphics g2,int xi,int yi,int R1i)
    {
        double angle;
        Graphics2D g=(Graphics2D)g2;
        double R1=R1i;
        polar P1=new polar();
        int x[]=new int[n];
        int y[]=new int[n];
        GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
        x.length);
    }

```



```

int x1,y1;
double teta=(maxteta-minteta)/n;
double dx=(maxx-minx)/n;

for(double xx=minx;xx<=maxx;xx+=dx)
{ angle=(xx-minx)/(maxx-minx)*(maxteta-minteta)+minteta;
  double teta1=-angle+Math.PI*3.0/2.0;
  //System.out.println("xx="+xx+"angle="+angle+"teta1="+teta1);
  P1.polar_input(R1+20,teta1);
  x1=(xi+(int)P1.xi());
  y1=(yi+(int)P1.xj());

  if(teta1>Math.PI/2.0 && teta1<3.0*Math.PI/2.0)
  { x1=(xi+(int)P1.xi())-40;
  }
  else
  {
    x1=(xi+(int)P1.xi());
  }
  if(teta1>Math.PI && teta1<2.0*Math.PI)
  { y1=(yi-(int)P1.xj()+10);}
  else
  { y1=(yi-(int)P1.xj());}

  g.setFont(new Font("Serif",Font.BOLD,18));
  g.drawString(toString(xx,5,1),x1,y1);
}
}

public void drawLabel(Graphics g2,int xi,int yi,int R1i)
{
  double angle;
  Graphics2D g=(Graphics2D)g2;
  double R1=R1i;
  polar P1=new polar();
  polar P2=new polar();
  int x[]=new int[n];
  int y[]=new int[n];
  int x1,y1;
  double teta=(maxteta-minteta)/n;
  double teta2;
  double dx=(maxx-minx)/n;
  g.setFont(new Font("Serif",Font.BOLD,18));
  AffineTransform at;
  AffineTransform origTransform = g.getTransform();
  for(double xx=minx;xx<=maxx;xx+=dx)
  { angle=(xx-minx)/(maxx-minx)*(maxteta-minteta)+minteta;
    double teta1=-angle+Math.PI*3.0/2.0;
    teta2=angle+Math.PI/2.0-Math.PI/40.0;
    // dik yazi istersek
    // teta2=angle+Math.PI/2.0+Math.PI/80.0;
    P1.polar_input(R1,teta2);
    x1=(xi+(int)P1.xi());
    y1=(yi+(int)P1.xj());
    g.translate(x1,y1);
    g.rotate(teta2+Math.PI/2.0);
    // dik yazi istersek
    // g.rotate(teta2);
  }
}

```

```

    g.drawString(toString(xx,3,0),0,0);
    g.setTransform(origTransform);
}

} //drawAkrep metodu sonu

public static String toString(double left, int w, int d)
// converts a double to a string with given width and decimals.
{
    NumberFormat df=NumberFormat.getInstance(Locale.US);
    df.setMaximumFractionDigits(d);
    df.setMinimumFractionDigits(d);
    df.setGroupingUsed(false);
    String s = df.format(left);
    while (s.length() < w)
        s = " " + s;
    if (s.length() > w)
    {
        s = "";
        for (int i=0; i<w; i++)
            s = s + "-";
    }
    return s;
}

public void drawLine(Graphics g2,int xi,int yi,int R1,int R2,int ni)
{
    double angle;
    Graphics2D g=(Graphics2D)g2;
    double R1=R1i;
    double R2=R2i;
    polar P1=new polar();
    polar P2=new polar();
    int x[]=new int[ni];
    int y[]=new int[ni];
    GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
    x.length);
    double teta=(maxteta-minteta)/ni;
    double dx=(maxx-minx)/ni;
    for(double xx=minx;xx<=maxx;xx+=dx)
    { angle=(xx-minx)/(maxx-minx)*(maxteta-minteta)+minteta;
      double teta1=-angle+Math.PI*3.0/2.0;
      P1.polar_input(R1,teta1);
      P2.polar_input(R2,teta1);
      polygon.moveTo((xi+(int)P1.xi()),(yi-(int)P1.xj()));
      polygon.lineTo((xi+(int)P2.xi()),(yi-(int)P2.xj()));
    }
    g.draw(polygon);
} //drawAkrep metodu sonu

public void drawAkrep(Graphics g2,int xi,int yi,int Ri)
{
    double angle=(x-minx)/(maxx-minx)*(maxteta-minteta)+minteta;
    Graphics2D g=(Graphics2D)g2;
    double R=Ri;
    double r=Ri*0.3;
    polar P1=new polar();

```

```

polar P2=new polar();
polar P3=new polar();
int x[]=new int[5];
int y[]=new int[5];
GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
x.length);
x[0]=xi;
y[0]=yi;
x[4]=xi;
y[4]=yi;
double teta=Math.PI/10.0;
    double teta1=-angle+Math.PI*3.0/2.0;
    double teta2=teta1+teta/2.0;
    double teta3=teta1-teta/2+Math.PI*2;
    P1.polar_input(R,teta1);
    P2.polar_input(r,teta2);
    P3.polar_input(r,teta3);
    x[1]=xi+(int)P3.xi();
    y[1]=yi-(int)P3.xj();
    x[2]=xi+(int)P1.xi();
    y[2]=yi-(int)P1.xj();
    x[3]=xi+(int)P2.xi();
    y[3]=yi-(int)P2.xj();
for(int i=0;i<5;i++)
{
    if(i==0)
    {
        polygon.moveTo(x[0],y[0]);
    }
    else
    {
        polygon.lineTo(x[i],y[i]);
    }
}
g.draw(polygon);

} //drawAkrep metodu sonu
public void fillAkrep(Graphics g2,int xi,int yi,int Ri)
{
    double angle=(x-minx)/(maxx-minx)*(maxteta-minteta)+minteta;
    Graphics2D g=(Graphics2D)g2;
    double R=Ri;
    double r=Ri*0.3;
    polar P1=new polar();
    polar P2=new polar();
    polar P3=new polar();
    int x[]=new int[5];
    int y[]=new int[5];
    GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
x.length);
x[0]=xi;
y[0]=yi;
x[4]=xi;
y[4]=yi;
double teta=Math.PI/10.0;
    double teta1=-angle+Math.PI*3.0/2.0;
    double teta2=teta1+teta/2.0;
    double teta3=teta1-teta/2+Math.PI*2;

```

```

P1.polar_input(R,teta1);
P2.polar_input(r,teta2);
P3.polar_input(r,teta3);
x[1]=xi+(int)P3.xi();
y[1]=yi-(int)P3.xj();
x[2]=xi+(int)P1.xi();
y[2]=yi-(int)P1.xj();
x[3]=xi+(int)P2.xi();
y[3]=yi-(int)P2.xj();
for(int i=0;i<5;i++)
{
    if(i==0)
    {
        polygon.moveTo(x[0],y[0]);
    }
    else
    {
        polygon.lineTo(x[i],y[i]);
    }
}
g.fill(polygon);

} //drawAkrep metodu sonu

public static void drawDaire(Graphics g2, int xi,int yi, int R)
{
    Graphics2D g=(Graphics2D)g2;
    g.draw(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2*R,2*R));
}

public static void fillDaire(Graphics g2, int xi,int yi, int R)
{
    Graphics2D g=(Graphics2D)g2;
    g.fill(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2*R,2*R));
}

    public TexturePaint getImageTexture(String imageFile,int xi,int yi, int R)
    {
        URL url = getClass().getResource(imageFile);
        Image img = getToolkit().getImage(url);
        try {
            MediaTracker tracker = new MediaTracker(this);
            tracker.addImage(img, 0);
            tracker.waitForID(0);
        } catch (Exception e) {}
        int width = 2*R;
        int height = 2*R;
        BufferedImage buffImg = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
        Graphics g = buffImg.getGraphics();
        g.drawImage(img,-30,-20, this);
        Rectangle2D rect = new Rectangle((int)(xi-R),(int)(yi-R), width, height);
        return new TexturePaint(buffImg, rect);
    }

public void paintComponent(Graphics g)
{
    final Graphics2D g2 = (Graphics2D) g;
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,

```

```

RenderingHints.VALUE_ANTIALIAS_ON);
try{
    final BasicStroke s1 = new BasicStroke(1);
    final BasicStroke s2 = new BasicStroke (2);
    final BasicStroke s10 = new BasicStroke (10);
    FontMetrics metrics=g.getFontMetrics();
    //System.out.println(metrics.getAscent());
    //System.out.println(metrics.getDescent());
    //System.out.println(metrics.getHeight());
    int xi=300;
    int yi=200;
    int R1=170;
    int R2=160;
    int R3=150;
    g2.setColor(Color.yellow);
    TexturePaint tp = getImageTexture("deneysan1.jpg",xi,yi,R1);
    g2.setPaint(tp);
    star.fillDaire(g2,xi,yi,R1);
    g2.setColor(Color.red);
    fillAkrep(g2,xi,yi,R2);
    g2.setColor(Color.black);

    g2.setStroke (s2);
    g2.setPaint (Color.black);
    drawDaire(g2,xi,yi,R1);
    drawDaire(g2,xi,yi,R2);

    drawDaire(g2,xi,yi,5);
    drawAkrep(g2,xi,yi,R2);
    drawLine(g2,xi,yi,R1,R2,n);
    g2.setStroke (s1);
    drawDaire(g2,xi,yi,R3);
    drawLine(g2,xi,yi,R2,R3,n1);
    g2.setStroke (s2);
    drawLabel(g2,xi,yi,R1+10);
    drawPlotLabel(g2,name,xi,yi,R1);
    drawPlotLabel(g2,unit,xi,yi+30,R1);
    }
    catch(NullPointerException e){}
}
}

```

Program 8.8.5 analog göstergely slider ile süren sliderP sınıfı

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;

public class sliderP extends JPanel implements ChangeListener
{
    private analogDialP p;
    private JSlider cap;
    private JTextArea tf;
    Color r=Color.lightGray;
    public double x;
    // pencereyi baslatma metodu

```

```

public sliderP()
{
    x=40.0;
    this.setLayout(new BorderLayout());
    //p=new analogDialP("Sıcaklık kanal 1", ""+"\u00B0"+"C",0.0,100.0,0.0,2*Math.PI*9.0/10.0,10,100);
    p=new analogDialP("Sıcaklık kanal 1", ""+"\u00B0"+"C",0.0,100.0,Math.PI/4,Math.PI*2-Math.PI/4,10,100);
    p.setx(x);
    Color c=p.getBackground();
    cap=new JSlider(SwingConstants.HORIZONTAL ,0,100,(int)x);
    cap.setMajorTickSpacing(10);
    cap.setMinorTickSpacing(1);
    cap.setPaintTicks(true);
    cap.setPaintLabels(true);
    cap.addChangeListener(this);
    tf=new JTextArea(""+x);
    tf.setBackground(c);
    tf.setFont(new Font("TimesRoman",Font.BOLD,32));
    add(tf,BorderLayout.NORTH);
    add(p,BorderLayout.CENTER);
    add(cap,BorderLayout.SOUTH);
}
// girdi alanındaki olan olayları dinleme metodu
public void stateChanged(ChangeEvent e)
{
    x=(double)cap.getValue();
    p.setx(x);
    tf.setText(""+x);
    repaint();
}

//=====
}

```

Program 8.8.6 analog göstergeli slider ile süren sliderPtest sınıfı

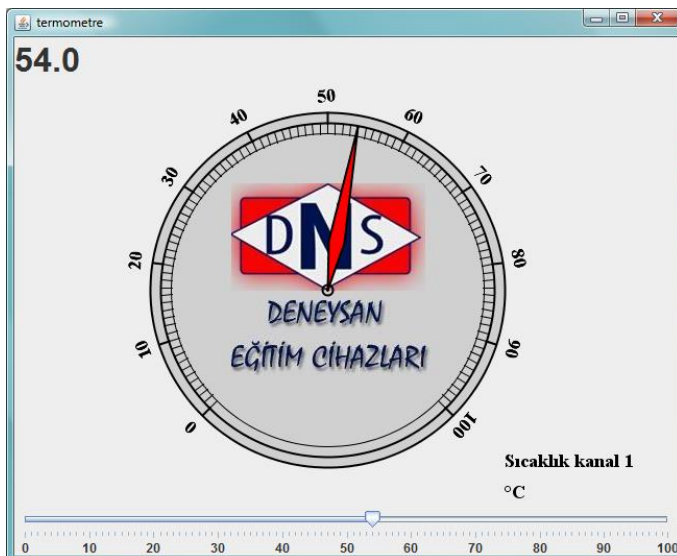
```

class sliderPtest {

public static void main(String args[])

{FrameGrafigi.plot("termometre ",new sliderP());}
}

```



Şimdi de düz termometre şekli tanımlayan ve bu şekli JSlider ile süren bir program gurubu tanımlayalım:

Program 8.8.7 analog termometre göstergesi dialP1

```
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;
import java.awt.Graphics;
import java.util.*;
import java.text.*;
import java.util.Locale;
import java.awt.image.*;
// ===== //
public class dialP1 extends JPanel
{
    double x,minx,maxx;
    double minteta,maxteta;
    int n;
    int n1;
    JPanel out;
    String name;
    String unit;
    JTextArea jta;
    Color col;
    int width;
    int height;
    public dialP1(String namei,String uniti,double minxi,double maxx, int ni,int n1i,int widthi,int heighti)
    { // input parameters
        // Name : name
        // Unit : unit
        // minx : minimum x
        // maxx : maximum x
        // n : big scale number
        // n1 : small scale number
        // width: width of the column
        // height: height of the column
        out=new JPanel();
        out.setLayout(new BorderLayout());
        name=namei;
        unit=uniti;
        minx=minxi;
        maxx=maxxi;
        width=widthi;
        height=heighti;
        x=minx;
        n=ni;
        n1=n1i;
        jta=new JTextArea("");
        jta.setFont(new Font("TimesRoman",Font.BOLD,32));
        col=this.getBackground();
        jta.setBackground(col);
    }
}
```

```

out.add(jta, BorderLayout.SOUTH);
add(out);
}

public void setColor(Color c)
{col=c;
}

public void setx(double xi)
{x=xi;}

public static String toString(double left, int w, int d)
// converts a double to a string with given width and decimals.
{
    NumberFormat df=NumberFormat.getInstance(Locale.US);
    df.setMaximumFractionDigits(d);
df.setMinimumFractionDigits(d);
df.setGroupingUsed(false);
String s = df.format(left);
while (s.length() < w)
    s="" + s;
if (s.length() > w)
{ s="";
    for(int i=0; i<w; i++)
        s = s + "-";
    }
return s;
}

public TexturePaint getImageTexture(String imageFile,int xi,int yi, int widthi,int heighti)
{
URL url = getClass().getResource(imageFile);
Image img = getToolkit().getImage(url);
try {
    MediaTracker tracker = new MediaTracker(this);
    tracker.addImage(img, 0);
    tracker.waitForID(0);
} catch (Exception e) {}
//width = widthi;
//height = heighti;
BufferedImage buffImg = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
Graphics g = buffImg.getGraphics();
g.drawImage(img,-100,-100, this);
Rectangle2D rect = new Rectangle((int)(xi-width),(int)(yi-height), width, height);
return new TexturePaint(buffImg, rect);
}

public void drawYTic(Graphics2D g2,int xi,int yi)
{
    //draw a series of y axis tics
    int dtic=(int)(height/n);
    int dtic1=(int)(height/n/n1);
    int x1=xi;
    int y1=yi;
    int R1=height;
    int R2=width;
    int dR=5;
    int dR1=5;
    int dR2=20;
    double dx=(maxx-minx)/n;

```



```

double xx1,xx2,yy1,yy2;
double xx1a,xx2a,yy1a,yy2a;
double xx=minx;
xx1=x1+R2+dR;
yy1=y1+R1;
xx2=x1+R2+dR+dR2;
yy2=y1+R1;
xx1a=x1+R2+dR;
xx2a=x1+R2+dR+dR1;
for(int i=0;i<n;i++)
{g2.draw(new Line2D.Double( xx1,yy1,xx2,yy1));
g2.drawString(""+xx,(int)xx2,(int)yy1);
yy1a=yy1;yy2a=yy2;
for(int j=0;j<n1;j++)
{
g2.draw(new Line2D.Double( xx1a,yy1a,xx2a,yy1a));
yy1a-=dtic1;
}
yy1-=dtic;
xx+=dx;
}
g2.draw(new Line2D.Double( xx1,yy1,xx2,yy1));
g2.drawString(""+xx,(int)xx2,(int)yy1);
}

public void paintComponent(Graphics g)
{
final Graphics2D g2 = (Graphics2D) g;
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
try{
final BasicStroke s1 = new BasicStroke(1);
final BasicStroke s2 = new BasicStroke(2);
final BasicStroke s3 = new BasicStroke(3);
final BasicStroke s4 = new BasicStroke(4);
final BasicStroke s6 = new BasicStroke(6);
final BasicStroke s10 = new BasicStroke(10);
FontMetrics metrics=g.getFontMetrics();
//System.out.println(metrics.getAscent());
//System.out.println(metrics.getDescent());
//System.out.println(metrics.getHeight());
int xi=100;
int yi=100;
int R1=height;
int R2=width;
int dh=30;
g2.setColor(Color.yellow);
TexturePaint tp = getImageTexture("deneysan1.jpg",xi,yi,R1,R1);
g2.setPaint(tp);
g2.setColor(Color.blue);
RoundRectangle2D rect = new RoundRectangle2D.Double((int)xi,(int)(yi-dh),R2, (R1+2*dh),20,20);
g2.setStroke( s4);
g2.draw(rect);
g2.setStroke( s1);
g2.fill(rect);
Ellipse2D e1=new Ellipse2D.Double((int)(xi-R2/2),(int)(yi+R1-R2/2+dh),2*R2,2*R2);
g2.setStroke( s6);
g2.draw(e1);
}

```

```

g2.setStroke (s1);
g2.setColor(Color.red);
g2.fill(e1);
int xx=(int)((x-minx)/(maxx-minx)*R1);
RoundRectangle2D rect1 = new RoundRectangle2D.Double((int)xi,(int)(yi+(R1-xx)),R2,(int)(xx+dh),10,10);
g2.setStroke (s6);
g2.fill(rect1);
g2.setPaint (Color.black);
g2.setStroke (s1);
drawYTic(g2,xi,yi);
}
catch(NullPointerException e){}
}
}

```

Program 8.8.8 analog termometre göstergesi sliderP1

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import javax.swing.event.*;

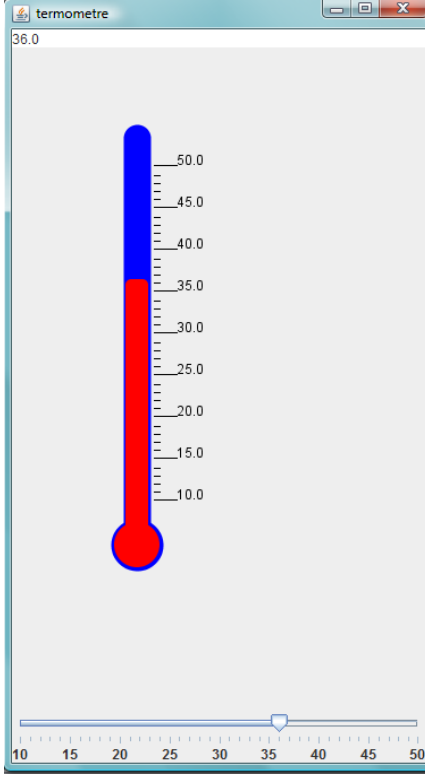
public class sliderP1 extends JPanel implements ChangeListener
{
    private dialP1 p;
    private JSlider cap;
    private JTextArea tf;
    Color r=Color.lightGray;
    public double x;
    // pencereyi baslatma metodu
    public sliderP1()
    {
        x=40.0;
        this.setLayout(new BorderLayout());
        p=new dialP1("isim","derece K",10.0,50.0,8,5,20,300);
        p.setx(x);
        cap=new JSlider(SwingConstants.HORIZONTAL,10,50,(int)x);
        cap.setMajorTickSpacing(5);
        cap.setMinorTickSpacing(1);
        cap.setPaintTicks(true);
        cap.setPaintLabels(true);
        cap.addChangeListener(this);
        tf=new JTextArea(""+x);
        add(tf,BorderLayout.NORTH);
        add(p,BorderLayout.CENTER);
        add(cap,BorderLayout.SOUTH);
    }
    // girdi alanindaki olan olaylari dinleme metodu
    public void stateChanged(ChangeEvent e)
    { x=(double)cap.getValue();
      p.setx(x);
      tf.setText(""+x);
      repaint();
    }

    //=====
}

```

Program 8.8.9 analog termometre göstergesi test programı sliderP1test

```
class sliderP1test {  
public static void main(String args[])  
{FrameGrafigi.plot("termometre ",new sliderP1());}}
```



8.9 JTABLE GRAFİK SINIFI

JTable sınıfı Tablo formunda (veri hücreleri şeklinde) grafik çıktı almaya yarayan bir sınıftır. Bu sınıfı gösterme amacıyla JTableP JPanel sınıfını hazırladık.

Program 8.9.1 JTableP JTable sınıfı örneği

```
import java.io.*;  
import java.util.*;  
import javax.swing.*;  
import javax.swing.table.*;  
import java.awt.*;  
public class JTableP extends JPanel  
{  
    private static final long serialVersionUID = 43567894L;  
    public void kur(Object a[][],String baslik[])  
    {  
        int n=a.length;  
        int m=a[0].length;  
        JTable jt=new JTable(a,baslik);  
        jt.setPreferredScrollableViewportSize(new Dimension(500,80));  
        jt.setFillViewportHeight(true);  
        add(new JScrollPane(jt));  
    }  
    public void kur(Object a[][])
```

```

        { int n=a[0].length;
          String baslik[]=new String[n];
          char b='\u0041';
for(int i=0;i<n;i++)
{ baslik[i]=""+b;b++;}
kur(a,baslik);
}
public JTableP(Object a[][],String baslik[])
{ kur(a,baslik);}
    public JTableP(Object a[][])
    { kur(a);}
    public JTableP(double b[][])
    { Object a[][]=doubletoDouble(b);
      kur(a);
    }
    public JTableP(double b[][],String baslik[])
    { Object a[][]=doubletoDouble(b);
      kur(a,baslik);
    }
    public JTableP(int b[][])
    { Object a[][]=inttoInteger(b);
      kur(a);
    }
    public JTableP(int b[][],String baslik[])
    { Object a[][]=inttoInteger(b);
      kur(a,baslik);
    }
    public static void print(Object a[][],String baslik[],String bb)
{
    JTableP pp=new JTableP(a,baslik);
JFrame cerceve=new JFrame(bb);
cerceve.addWindowListener(new BasicWindowMonitor());
cerceve.getContentPane().add(pp);
cerceve.pack();
cerceve.setVisible(true);
}
    public static void print(Object a[][],String bb)
    {
        int n=a[0].length;
        String baslik[]=new String[n];
        char b='\u0041';
for(int i=0;i<n;i++)
{ baslik[i]=""+b;b++;}
JTableP pp=new JTableP(a,baslik);
    JFrame cerceve=new JFrame(bb);
cerceve.addWindowListener(new BasicWindowMonitor());
cerceve.getContentPane().add(pp);
cerceve.pack();
cerceve.setVisible(true);
}
    public static void print(Object a[][])
    {
        int n=a[0].length;
        String baslik[]=new String[n];
        char b='\u0041';
for(int i=0;i<n;i++)
{ baslik[i]=""+b;b++;}
JTableP pp=new JTableP(a,baslik);
    JFrame cerceve=new JFrame("JTable");

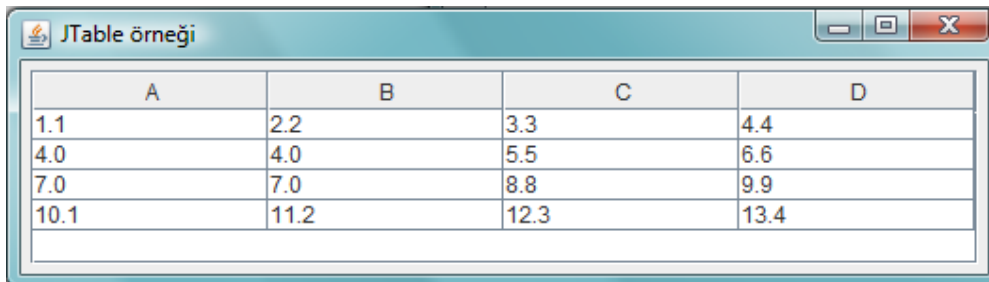
```

```

cerceve.addWindowListener(new BasicWindowMonitor());
cerceve.getContentPane().add(pp);
cerceve.pack();
cerceve.setVisible(true);
}
public static void print(double a[],String baslik[],String bb)
{print(doubletoDouble(a),baslik,bb);}
public static void print(double a[],String bb)
{print(doubletoDouble(a),bb);}
public static void print(double a[][])
{print(doubletoDouble(a));}
public static void print(int a[],String baslik[],String bb)
{print(inttoInteger(a),baslik,bb);}
public static void print(int a[],String bb)
{print(inttoInteger(a),bb);}
public static void print(int a[][])
{print(inttoInteger(a));}
public static Double[][] doubletoDouble(double xi[][])
{ int n=xi.length;
int m=xi[0].length;
Double X[][]=new Double[n][m];
for(int j=0;j<m;j++)
{for(int i=0;i<n;i++)
{X[i][j]=new Double(xi[i][j]);}
}
return X;
}
public static Integer[][] intoInteger(int xi[][])
{ int n=xi.length;
int m=xi[0].length;
Integer X[][]=new Integer[n][m];
for(int j=0;j<m;j++)
{for(int i=0;i<n;i++)
{X[i][j]=new Integer(xi[i][j]);}
}
return X;
}

public static void main(String arg[])
{double a[][]={{ 1.1,2.2,3.3,4.4},{4.4,5.5,6.6,7.7},{7.7,8.8,9.9,10.1},{10.1,11.2,12.3,13.4}};
print(a,"JTable örneği");}
}

```



A	B	C	D
1.1	2.2	3.3	4.4
4.0	4.0	5.5	6.6
7.0	7.0	8.8	9.9
10.1	11.2	12.3	13.4

Bu sınıf içerisindeki print metodlarını direk olarak kullanabileceğimiz gibi, bir JPanel programı olarak herhangi bir grafik programı içerisinde de yararlanabiliriz.

Program 8.9.2 JTabletest JTable sınıfı örneği

```
import java.io.*;
```

```

import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
public class JTabletest
{
    public static void main(String arg[])
    {
        double a[][]={{ 1,2,3},{4,5,6},{7,8,9}};
        String s[]={ "c1","c2","c3"};
        String s1="JTable örneği";
        JTableP pp=new JTableP(a,s);
        FrameGrafigi.plot(s1,pp);
    }
}

```

JTable kullanımında javanın son versiyonlarına kadar tek yol TableModel üzerinden işlem yaparak JTable sınıfını oluşturmaktı. Versiyon 7 den itibaren bu zorunluluk kalkmış bulunuyor, ancak bir çok programlama hala TableModel üzerinden daha kolay gerçekleştirilebilir. JTable'ın TableModel üzerinden çağrılan bir örneği altta verilmiştir.

Program 8.9.3 JTable sınıfını kurmak için kullanılan TableModel sınıfı örneği

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.event.*;
import java.io.*;
import java.util.*;

class gm extends AbstractTableModel implements TableModelListener
{ private static final long serialVersionUID = 7432664L;
  Object[][] veri;
  String[] baslik;

  public gm(Object[][] verii,String basliki[])
  {
    int n=verii.length;
    int m=verii[0].length;
    veri=new Object[n][m];
    baslik=new String[m];
    for(int j=0;j<m;j++)
    { for(int i=0;i<n;i++)
      { veri[i][j]=verii[i][j];}
      baslik[j]=basliki[j];
    }
    addTableModelListener(this);
  }

  public int getRowCount() {return veri.length;}

  public int getColumnCount() {return baslik.length;}

  public Object getValueAt(int satir,int sutun) {return veri[satir][sutun];}

  public String getColumnName(int c) {return baslik[c];}

  public void setValueAt(Object val, int row, int col)
  { veri[row][col] = val;}
}

```

```

public void tableChanged(TableModelEvent e)
{ }

public void setValues(Object[][] verii,String basliki[])
{ verii=verii;
  baslik=basliki;
}

public Object[][] getValues(){return verii;}

public boolean isCellEditable(int row, int col) {return true;}
}

```

Program 8.9.4 TabModel sınıfı üzerinden kurulan JTable sınıfını örneği

```

import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
public class JTabletest1
{
    public static void main(String arg[])
    {
        String a[][]={{ "A", "B", "C"}, {"D", "E", "F"}, {"G", "H", "I"} };
        String s[]={ "c1", "c2", "c3" };
        String s1="JTable örneği";
        gm g=new gm(a,s);
        JTable pp=new JTable(g);
        JPanel jp=new JPanel();
        jp.add(new JScrollPane(pp));
        JFrame jf=new JFrame(s1);
        jf.add(jp);
        jf.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        jf.setSize(800,500);
        jf.setVisible(true);
    }
}

```

	c1	c2	c3
A	A	B	C
D	D	E	F
G	G	H	I

8.10 GRAFİK ÇIKTI SAYFALARINI FORMATLAMA

Grafik format kontrol yapılarını kullanarak grafik çıktı programlarını istediğimiz formda oluşmasını sağlayabiliriz. Bu bölümde değişik formatlama sistemlerine detaylı bakacağız. Bu terimlerin bir kısmını zaten şu ana kadar örnek olarak verdiğimiz programların içinde kullandık, örneğin BorderLayout isimli grafik format kontrol yapısı sık olarak örneklerimizde kullanıldı

Border Layout

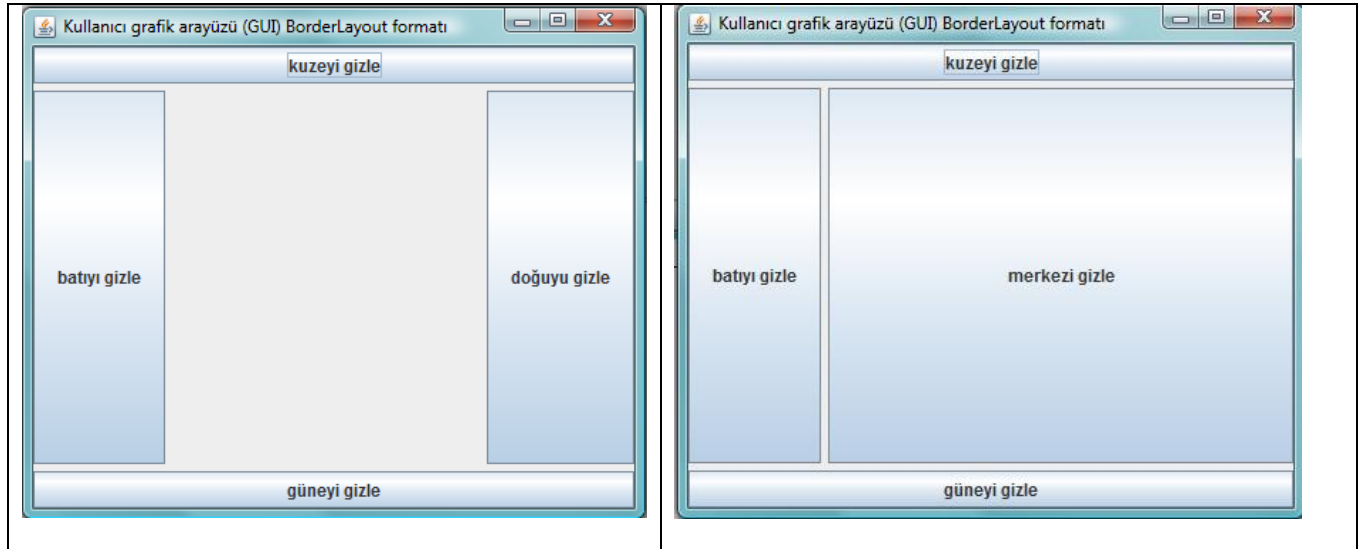
BorderLayout ekranı 5 bölgeye ayırır. DOĞU,BATI,KUZZEY ,GÜNEY ve MERKEZ (WEST,EAST,NORTH,SOUTH ve CENTER). Kurucu metodlarımız şunlardır:

```
public BorderLayout()  
public BorderLayout(int yatay_bosluk,int düşey_bosluk);
```

yatay_bosluk ve düşey_bosluk değişkenleri formatlamada kutular arasında kalan boşlukların piksel olarak değerleridir. Örnek problemimizde basılan düğme gizlenecektir.

Program 8.10.1 BorderLayoutTest örnek problemi

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
public class BorderLayoutTest extends JFrame implements ActionListener  
{  
    private JButton B[];  
    private String S[]={ "Hide North", "Hide South", "Hide East",  
        "Hide West", "Hide Center"};  
  
    public BorderLayoutTest(String s)  
    {  
        super(s);  
        setLayout(new BorderLayout(5,5));  
        B=new JButton[5];  
        for(int i=0;i<B.length;i++)  
        {  
            B[i]=new JButton(S[i]);  
            B[i].addActionListener(this);  
        }  
        add(B[0],BorderLayout.NORTH);  
        add(B[1],BorderLayout.SOUTH);  
        add(B[2],BorderLayout.EAST);  
        add(B[3],BorderLayout.WEST);  
        add(B[4],BorderLayout.CENTER);  
    }  
  
    public void actionPerformed(ActionEvent e)  
    {  
        for(int i=0;i<B.length;i++)  
        {  
            if(e.getSource()==B[i])  
                B[i].setVisible(false);  
            else B[i].setVisible(true);  
        }  
        validate();  
    }  
  
    public static void main(String[] args)  
    {  
        BorderLayoutTest f = new BorderLayoutTest("Border Layout format Grafic User Interphase(GUI) ");  
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );  
        f.setSize(800,500);  
        f.setVisible(true);  
    }  
}
```

Flow Layout

FlowLayout sınıfı arka arkaya eklenen elementleri bir sıra içinde gösterir. Sıra dolunca alt satıra geçer.

Kurucu metodları:

public Flowlayout()

public Flowlayout(int formatregion)

public Flowlayout(int formatregion, int horizontal_space, int vertical_space);

satır sağa, sola veya ortaya göre ayarlanabilir. BU ayarlamalar için gerekli sabitler:

FlowLayout.RIGHT,

FlowLayout.CENTER or

FlowLayout.LEFT

Düğmeleri sağ sol veya ortaya kaydıran bir örnek verilmiştir.

Program 8.10.2 FlowLayoutTest örneği

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class FlowLayoutTest extends JFrame implements ActionListener
{
    private JButton B[];
    private Container c;

    public FlowLayoutTest(String s)
    {
        super(s);
        setLayout(new FlowLayout());
        B=new JButton[3];
        B[0]=new JButton("Sola kaydır");
        B[1]=new JButton("Ortala");
        B[2]=new JButton("Sağa kaydır");
        for(int i=0;i<B.length;i++)
        {
            B[i].addActionListener(this);
```

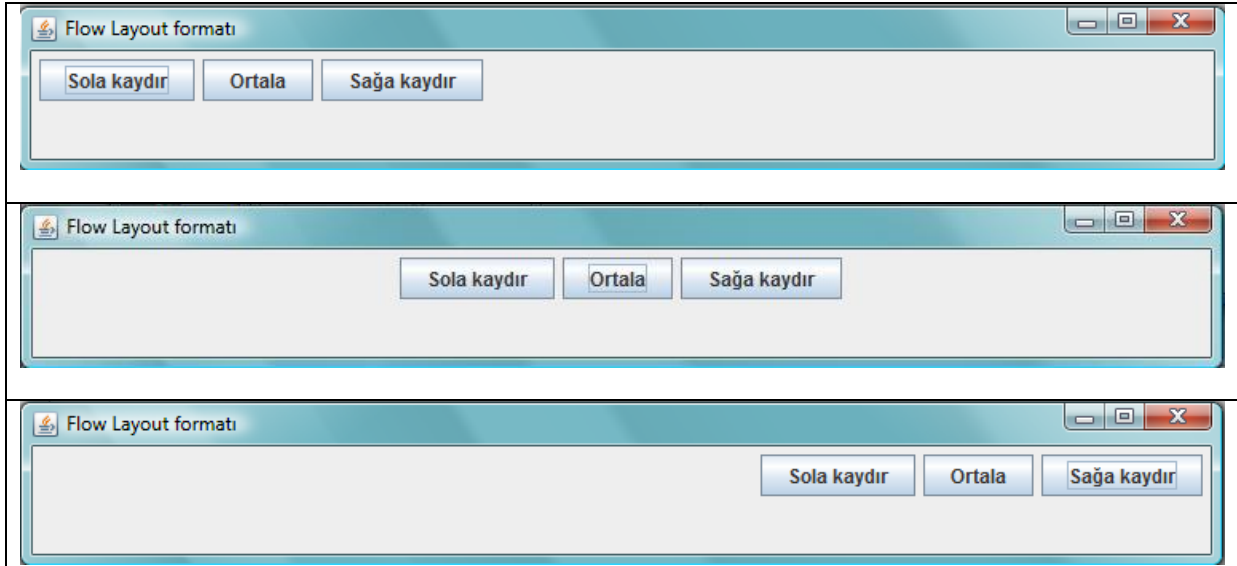
```

add(B[i]);
}
}

public void actionPerformed(ActionEvent e)
{
int position=FlowLayout.LEFT;;
if(e.getSource()==B[0])
{
position=FlowLayout.LEFT;
}
else if(e.getSource()==B[1])
{
position=FlowLayout.CENTER;
}
else if(e.getSource()==B[2])
{
position=FlowLayout.RIGHT;
}
setLayout(new FlowLayout(position));
validate();
}

public static void main(String[] args)
{
FlowLayoutTest f = new FlowLayoutTest("Flow Layout formatı");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
f.setSize(800,500);
f.setVisible(true);
}
}

```



Grid Layout

GridLayout sınıfı ekranı eşit büyüklükte gridlere böler ve elemntleri sırasıyla bu gridlere yerleştirir.

Kurucu metodları:

```

public GridLayout(int rows,int columns)
publicGridLayout(int rows,int columns,int horizontal_space,int vertical_space);

```

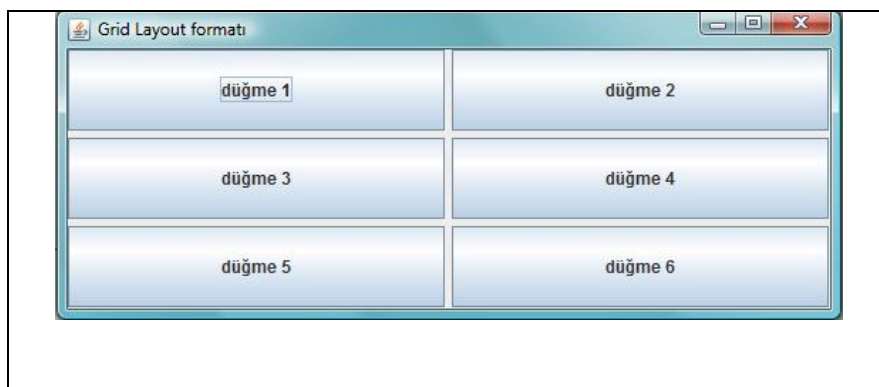
Program 8.10.3 FlowLayoutTest örneği

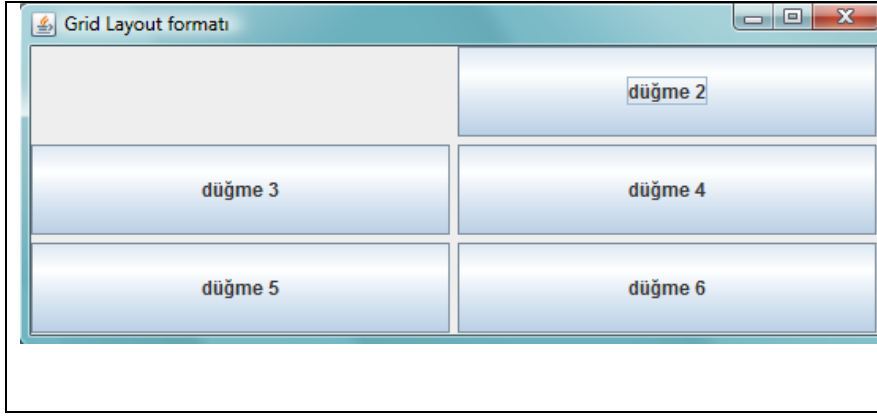
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class GridLayoutTest extends JFrame implements ActionListener
{
    private JButton B[];
    private String S[]={ "Button 1","Button 2","Button 3",
    "Button 4","Button 5","Button 6"};

    public GridLayoutTest(String s)
    {
        super(s);
        setLayout(new GridLayout(3,2,5,5));
        B=new JButton[6];
        for(int i=0;i<B.length;i++)
        {
            B[i]=new JButton(S[i]);
            B[i].addActionListener(this);
            add(B[i]);
        }
    }

    public void actionPerformed(ActionEvent e)
    {
        for(int i=0;i<B.length;i++)
        {
            if(e.getSource()==B[i])
            B[i].setVisible(false);
            else B[i].setVisible(true);
        }
        validate();
    }

    public static void main(String[] args)
    {
        GridLayoutTest f = new GridLayoutTest("Grid Layout formatting of Graphic User Interphase (GUI)");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        f.setSize(800,500);
        f.setVisible(true);
    }
}
```





Absolute (mutlak) Layout

Bu formatlama yapısında aslında hiçbir format tanımlanmamıştır. Tüm tanımlar Dimension sınıfı ve dimension sınıfının setBounds metodu kullanılarak program içinde mutlak yer olarak tanımlanır. Mutlak tanımlamalar değişmeyeceğinden dinamik değişken sayfalar için çık uygun değildir.

Program 8.10.4 AbsoluteLayoutTest örneği

```
import java.awt.Container;
import java.awt.Insets;
import java.awt.Dimension;
import javax.swing.JButton;
import javax.swing.JFrame;

public class AbsoluteLayoutDemo {
    public static void addComponentsToPane(Container pane) {
        pane.setLayout(null);
        JButton b1 = new JButton("bir");
        JButton b2 = new JButton("iki");
        JButton b3 = new JButton("üç");
        pane.add(b1);
        pane.add(b2);
        pane.add(b3);
        Insets insets = pane.getInsets();
        Dimension size = b1.getPreferredSize();
        b1.setBounds(25 + insets.left, 5 + insets.top,
            size.width, size.height);
        size = b2.getPreferredSize();
        b2.setBounds(55 + insets.left, 40 + insets.top,
            size.width, size.height);
        size = b3.getPreferredSize();
        b3.setBounds(150 + insets.left, 15 + insets.top,
            size.width + 50, size.height + 20);
    }

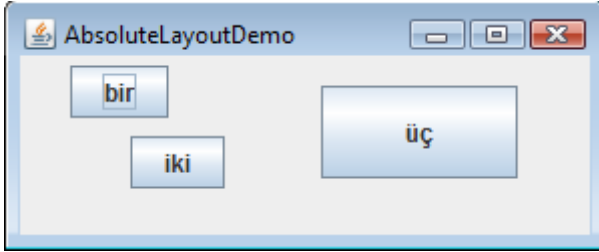
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("AbsoluteLayoutDemo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addComponentsToPane(frame.getContentPane());
        Insets insets = frame.getInsets();
        frame.setSize(300 + insets.left + insets.right,
            125 + insets.top + insets.bottom);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
```

```

    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```



Card (oyun kartı destesi) Layout

Card Layout oyun kartı destesi gibi davranır. Değişik Container gurubu formatlanmış elemanları Layout kontrolüyle destedeki oyun kağıtlarını değiştirir gibi sırayla gösterir. Her seferinde sadece bir Container görülür, ancak bu kontainerın içinde değişik elemanlar bulunabilir.

Program 8.10.5 CardLayoutTest örneği

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class CardLayoutDemo implements ItemListener {
    JPanel kartlar; //a panel that uses CardLayout
    final static String DUGMEPANELI = "JButtons kartı";
    final static String YAZIPANELI = "JTextField kartı";

    public void addComponentToPane(Container pane) {
        JPanel comboBoxPaneli = new JPanel(); //use FlowLayout
        String comboBoxelemanlari[] = { DUGMEPANELI, YAZIPANELI };
        JComboBox<String> cb = new JComboBox<String>(comboBoxelemanlari);
        cb.setEditable(false);
        cb.addItemListener(this);
        comboBoxPaneli.add(cb);
        JPanel kart1 = new JPanel();
        kart1.add(new JButton("Düğme 1"));
        kart1.add(new JButton("Düğme 2"));
        kart1.add(new JButton("Düğme 3"));
        JPanel kart2 = new JPanel();
        kart2.add(new JTextField("TextField", 20));
        kartlar = new JPanel(new CardLayout());
        kartlar.add(kart1, DUGMEPANELI);
        kartlar.add(kart2, YAZIPANELI);

        pane.add(comboBoxPaneli, BorderLayout.PAGE_START);
        pane.add(kartlar, BorderLayout.CENTER);
    }

    public void itemStateChanged(ItemEvent evt) {
        CardLayout cl = (CardLayout)(kartlar.getLayout());
        cl.show(kartlar, (String)evt.getItem());
    }

    private static void createAndShowGUI() {

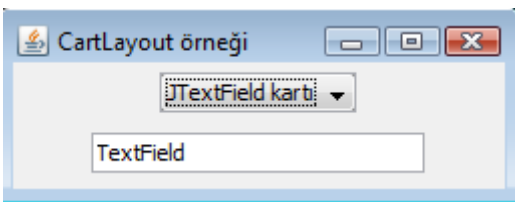
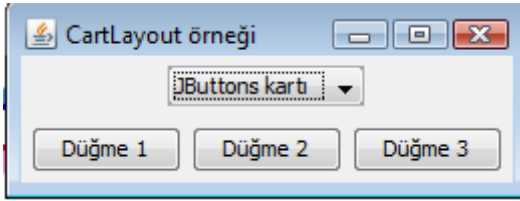
```

```

JFrame frame = new JFrame("CartLayout örneđi");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
CardLayoutDemo demo = new CardLayoutDemo();
demo.addComponentToPane(frame.getContentPane());
frame.pack();
frame.setVisible(true);
}

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        //UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    UIManager.put("swing.boldMetal", Boolean.FALSE);
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```



Box Layout

Javadaki diđer bir formatlama yapısı BoxLayout'tur. BoxLayout içindeki elemanları ya alt alta, ya da yan yana sıralayan bir formatlama sistemidir. Bir anlamda (en azından içindeki elemanları yan yana dizerken) biraz FlowLayout'u andırır, fakat çok daha esnek bir yapısı vardır. Program 8.10.6 da altalta sıralama, program 8.10.7 de yan yana sıralama, program 8.10.8 de alt alt sıralanmış iki JPanel'in yan yana sıralanmış şekli görölmektedir. Bu yapıyla GridLayout formatının çok daha esnek bir formu oluşturulabilmiştir.

Program 8.10.6 BoxLayoutTest örneđi

```

import java.awt.Component;
import java.awt.Container;
import javax.swing.*.*;

```

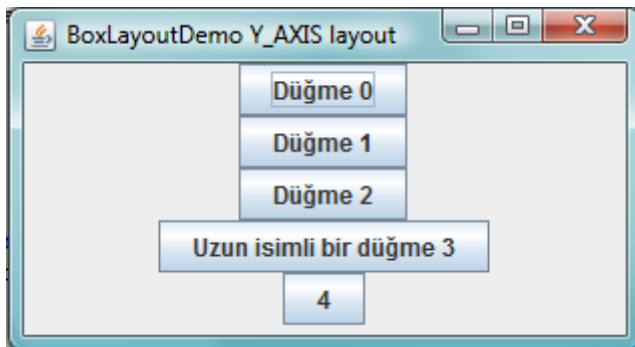
```

public class BoxLayoutDemo {
    public static void addComponentsToPane(Container pane)
    {
        JButton dugme[]=new JButton[10];
        dugme[0]=new JButton("Düğme 0");
        dugme[1]=new JButton("Düğme 1");
        dugme[2]=new JButton("Düğme 2");
        dugme[3]=new JButton("Uzun isimli bir düğme 3");
        dugme[4]=new JButton("4");
        JPanel b=new JPanel();
        b.setLayout(new BoxLayout(b, BoxLayout.Y_AXIS ));
        for(int i=0;i<5;i++) { dugme[i].setAlignmentX(Component.CENTER_ALIGNMENT);b.add(dugme[i]); }
        pane.add(b);
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("BoxLayoutDemo Y_AXIS layout");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addComponentsToPane(frame.getContentPane());
        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}

```



Program 8.10.7 BoxLayoutTest örneği

```

import java.awt.Component;
import java.awt.Container;
import javax.swing.*;

public class BoxLayoutDemo1 {
    public static void addComponentsToPane(Container pane)
    {
        JButton dugme[]=new JButton[10];
        dugme[0]=new JButton("Düğme 5");
        dugme[1]=new JButton("Düğme 6");
    }
}

```

```

dugme[2]=new JButton("Düğme 7");
dugme[3]=new JButton("Uzun isimli bir düğme 8");
dugme[4]=new JButton("9");
JPanel b=new JPanel();
b.setLayout(new BorderLayout(b, BorderLayout.X_AXIS ));
for(int i=0;i<5;i++) { dugme[i].setAlignmentX(Component.CENTER_ALIGNMENT); b.add(dugme[i]); }
pane.add(b);
}

private static void createAndShowGUI() {
    JFrame frame = new JFrame("BoxLayoutDemo1 X_AXIS Layout");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    addComponentsToPane(frame.getContentPane());
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```



Program 8.10.8 BoxLayoutTest örneği

```

import java.awt.Component;
import java.awt.Container;
import javax.swing.*;

public class BoxLayoutDemo2 {
    public static void addComponentsToPane(Container pane)
    {
        JPanel b[]=new JPanel[3];
        JButton dugme[]=new JButton[10];
        dugme[0]=new JButton("Düğme 0");
        dugme[1]=new JButton("Düğme 1");
        dugme[2]=new JButton("Düğme 2");
        dugme[3]=new JButton("Uzun isimli bir düğme 3");
        dugme[4]=new JButton("4");
        dugme[5]=new JButton("Düğme 5");
        dugme[6]=new JButton("Düğme 6");
        dugme[7]=new JButton("Düğme 7");
        dugme[8]=new JButton("Uzun isimli bir düğme 8");
        dugme[9]=new JButton("9");
        for(int i=0;i<3;i++) { b[i]=new JPanel();}
        b[0].setLayout(new BorderLayout(b[0], BorderLayout.Y_AXIS ));
        b[1].setLayout(new BorderLayout(b[1], BorderLayout.Y_AXIS ));
        b[2].setLayout(new BorderLayout(b[2], BorderLayout.X_AXIS ));
        for(int i=0;i<10;i++) { dugme[i].setAlignmentX(Component.CENTER_ALIGNMENT); }
        for(int i=0;i<5;i++) { b[0].add(dugme[i]); b[1].add(dugme[i+5]); }
    }
}

```



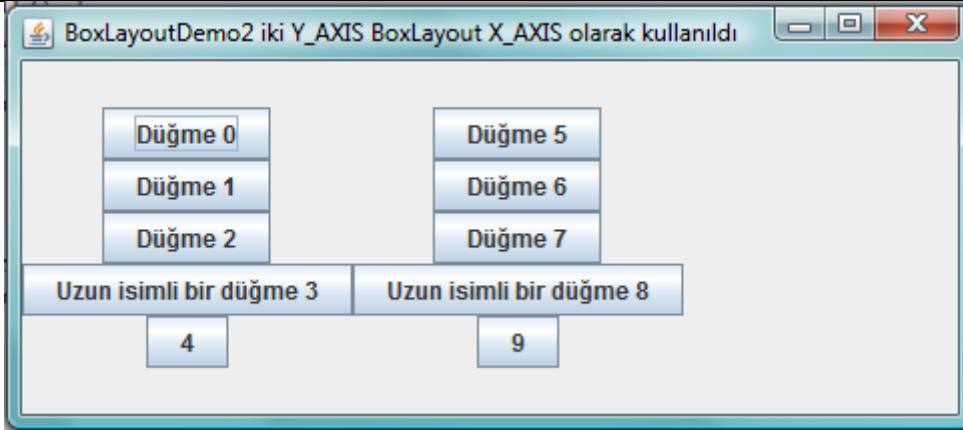
```

    b[2].add(b[0]);
    b[2].add(b[1]);
    pane.add(b[2]);
}

private static void createAndShowGUI() {
    JFrame frame = new JFrame("BoxLayoutDemo2 iki Y_AXIS BoxLayout X_AXIS olarak kullanıldı");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    addComponentsToPane(frame.getContentPane());
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```



JOptionPane sınıfı

JOptionPane sınıfı genel bir java kütüphanelerinde verilmiş hazır girdi-çıkı sınıfıdır. Buraya kadar bu sınıfı girdi-çıkı sınıfı olarak yoğun bir şekilde kullandık. Şimdi biraz daha derinleşerek sınıfın detaylarını inceleyelim. Bu sınıf **Object** sınıfından nesnelere girdi ve çıkı olarak kullanılabilir. Tüm nesne tipi sınıf değişkenleri **Object** sınıfından türetildiğinden tüm nesne tipi değişkenler için girdi ve çıkı yaratabilir. **JOptionPane** sınıfının bazı metodları aşağıda listelenmiştir.

public static String showInputDialog(Object message)

public static String showInputDialog(Component mainwindow, Object message)

public static String showInputDialog(Component mainwindow, Object message, String heading, int messagetype)

public static String showInputDialog(Component mainwindow, Object message, String heading, int messagetype, Icon resim, Object[] listselectionvalue, Object entrancevalue)

public static String showInternalInputDialog(Component mainwindow, Object message)

public static String showInternalInputDialog(Component mainwindow, Object message, String heading, int messagetype)

public static String showInternalInputDialog(Component mainwindow, Object message, String heading, int messagetype, Icon resim, Object[] listselectionvalue, Object girişdeğeri)

```

public static void showMessageDialog(Component mainwindow, Object message);
public static void showMessageDialog(Component mainwindow, Object message, String
heading, int messagetype);
public static void showMessageDialog(Component mainwindow, Object message, String
heading, int messagetype, Icon resim);
public static void showInternalMessageDialog(Component mainwindow, Object message);
public static void showInternalMessageDialog(Component mainwindow, Object message, String
heading, int messagetype);
public static void showInternalMessageDialog(Component mainwindow, Object message, String
heading, int messagetype, Icon resim);
public static int showConfirmDialog(Component mainwindow, Object message);
public static int showConfirmDialog(Component mainwindow, Object message, String
heading, int messagetype);
public static int showConfirmDialog(Component mainwindow, Object message, String
heading, int opsiyontürü, int messagetype);
public static int showConfirmDialog(Component mainwindow, Object message, String
heading, int opsiyontürü, int messagetype, Icon resim);
public static int showInternalConfirmDialog(Component mainwindow, Object message);
public static int showInternalConfirmDialog(Component mainwindow, Object message, String
heading, int messagetype);
public static int showInternalConfirmDialog(Component mainwindow, Object message, String
heading, int opsiyontürü, int messagetype);
public static int showInternalConfirmDialog(Component mainwindow, Object message, String
heading, int opsiyontürü, int messagetype, Icon resim);
public static int showOptionDialog(Component mainwindow, Object message, String heading, int
opsiyontürü, int messagetype, Icon resim, Object[] listselectionvalues Object listedekiilkdeğer);
public static int showInternalOptionDialog(Component mainwindow, Object message, String
heading, int opsiyontürü, int messagetype, Icon resim, Object[] listselectionvalues Object
listedekiilkdeğer);

```

Message type değişkeni alttaki değerleri alabilir.

JOptionPane.ERROR_MESSAGE hata mesajı

JOptionPane.INFORMATION_MESSAGE bilgi mesajı

JOptionPane.PLAIN_MESSAGE normal (sıradan mesaj)

JOptionPane.QUESTION_MESSAGE soru mesajı

JOptionPane.WARNING_MESSAGE uyarı mesajı

Opsiyontürü değişkeni aşağıdaki türleri kabul eder

JOptionPane.DEFAULT_OPTION : ok düğmesi verir

JOptionPane.OK_CANCEL_OPTION : ok veya iptal (cancel) düğmesi verir

JOptionPane.YES_NO_CANCEL_OPTION : Yes (evet) , No(hayır) veya Cancel (iptal) düğmesi

JOptionPane.YES_NO_OPTION : Yes (evet) , No(hayır)

OptionDialog aşağıdaki seçenekler kullanılabilir

JOptionPane.CANCEL_OPTION //iptal seçeneği

JOptionPane.CLOSED_OPTION kapatma seçeneği

JOptionPane.NO_OPTION hayır seçeneği

JOptionPane.OK_OPTION tamam (ok) seçeneği

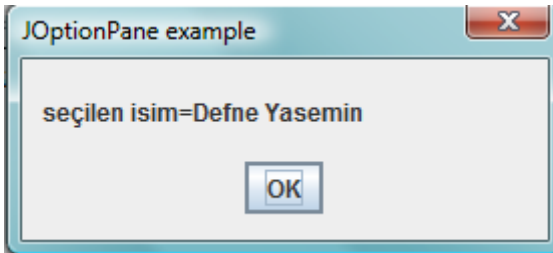
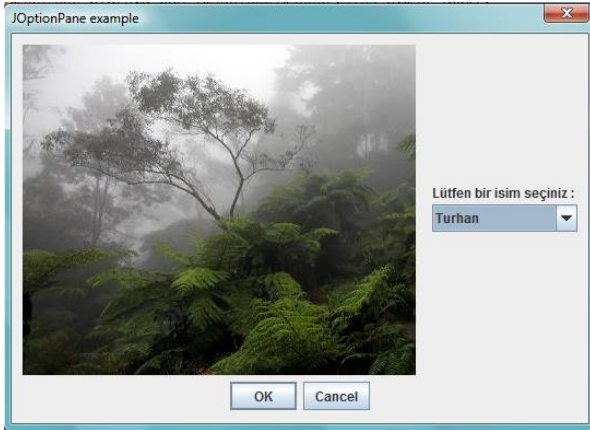
JOptionPane.YES_OPTION evet seçeneđi

JOptionPane.UNINITIALIZED_VALUE tanımlanmamış deđer

Program 8.10.9 JOptionPane1 Test örneđi

```
import javax.swing.*;
import java.awt.*;

public class JOptionPane1
{
    public static void main(String[] args)
    {
        Object[] o=new Object[]{"Turhan","Meral","İrfan","Hatice","Osman",
        "Nurhan","Birsen","Defne Yasemin","Selin","Mustafa","Ahmet","Mehmet","Ayhan"};
        ImageIcon picture=new ImageIcon("avustralya_orman.jpg");
        String name=(String)JOptionPane.showInputDialog(null,"Lütfen bir isim seçiniz : ",
        "JOptionPane example",JOptionPane.QUESTION_MESSAGE,picture,o,"Turhan");
        String s1="seçilen isim="+name;
        String s2="JOptionPane example";
        JOptionPane.showMessageDialog(null,s1,s2,JOptionPane.PLAIN_MESSAGE);
    }
}
```



JOptionPane sınıfını herhangi bir grafik çıktısı almak için de kullanabiliriz.

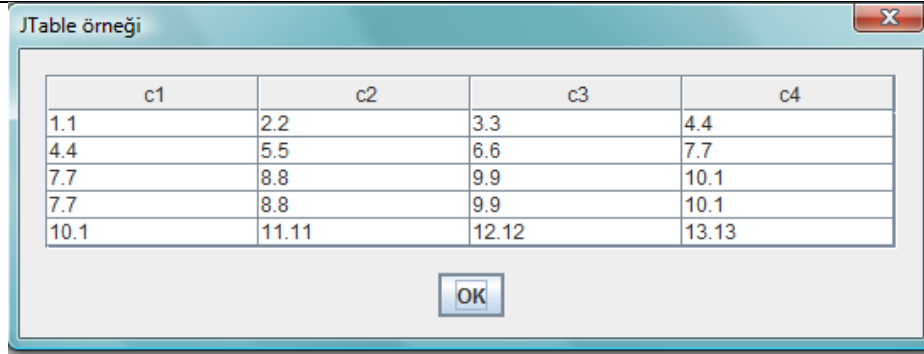
Program 8.10.10 JOptionPane sınıfının JTable çıktısı için kullanımı Test örneđi (JTableP Programı 8.9.1 de verilmiştir)

```
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
public class JTabletest2
```

```

{
public static void main(String arg[])
{
double a[][]={{1.1,2.2,3.3,4.4},{4.4,5.5,6.6,7.7},{7.7,8.8,9.9,10.10},{7.7,8.8,9.9,10.10},{10.10,11.11,12.12,13.13}};
String s[]={"c1","c2","c3","c4"};
String s1="JTable örneği";
JTableP pp=new JTableP(a,s);
JOptionPane.showMessageDialog(null,pp,s1,JOptionPane.PLAIN_MESSAGE);
}}

```



JPanel sınıfını format amacıyla kullanma

Format amacıyla sık olarak kullanılan bir sınıf ta JPanel'dir. Çünkü JPanel sınıfı nesnelere birbirinin içine istenildiği gibi eklenebildiği için farklı format yapısındaki alt elemanları birleştirebilirler. Böylece kendimize özgü format şekilleri elde edebiliriz.

Program 8.10.6 JPanel1test example

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

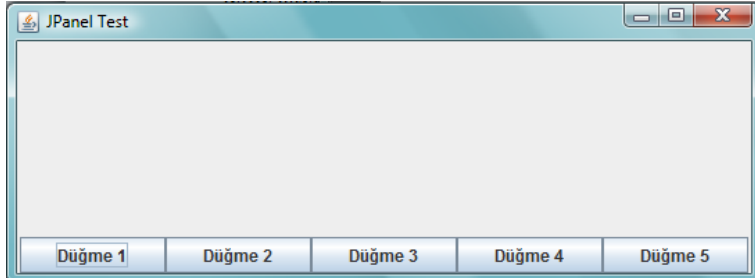
public class JPanel1test extends JApplet
{
private JPanel buttonPanel;
private JButton buttons[];

public void init()
{
buttonPanel=new JPanel();
buttons=new JButton[5];
buttonPanel.setLayout(new GridLayout(1,buttons.length));
for(int i=0;i<buttons.length;i++)
{
buttons[i]=new JButton("Düğme "+(i+1));
buttonPanel.add(buttons[i]);
}
setLayout(new BorderLayout());
add(buttonPanel,BorderLayout.SOUTH);
}

public static void main(String s[])
{
JFrame f = new JFrame("JPanel Test");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new JPanel1test();
f.getContentPane().add("Center", applet);
}
}

```

```
applet.init();
f.pack();
f.setSize(new Dimension(545,400));
f.setVisible(true);
}
}
```



JTabbedPane format sınıfı

JTabbedPane sınıfı birden fazla sayfayı dosya ayrıçlarıyla bir araya tutturarak verir. Her sayfa içinde tüm diğer format yapılarını kullanabiliriz.

Program 8.10.7 JTabbedPane örneği

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class JTabbedPaneTest extends JFrame implements ActionListener
{
    private JButton B[];
    private String S[]={"düğme 1","düğme 2","düğme 3",
    "düğme 4","düğme 5","düğme 6"};
    private Container c;

    public JTabbedPaneTest(String s)
    {
        super(s);
        setLayout(new BorderLayout(5,5));
        B=new JButton[6];
        JTabbedPane jtp=new JTabbedPane();
        for(int i=0;i<B.length;i++)
        {
            B[i]=new JButton(S[i]);
            B[i].addActionListener(this);
            jtp.addTab(S[i],B[i]);
        }
        add(jtp);
    }

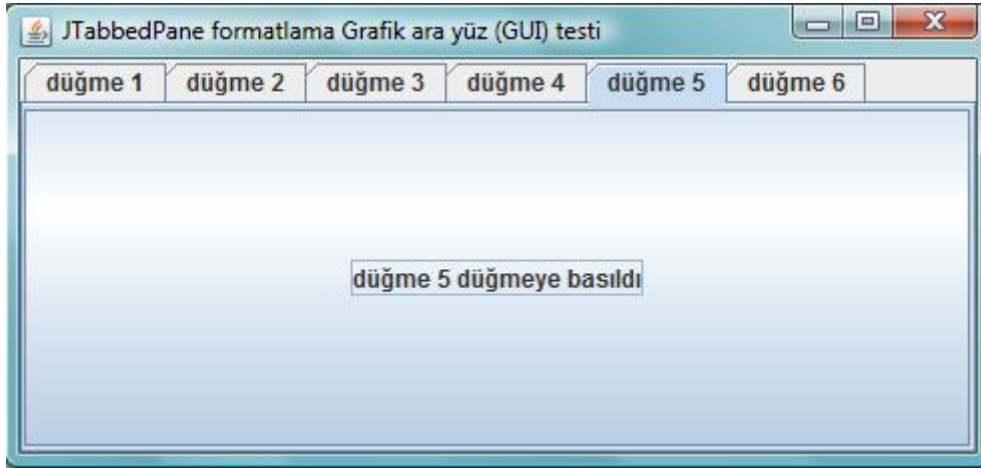
    public void actionPerformed(ActionEvent e)
    {
        for(int i=0;i<B.length;i++)
        {
            if(e.getSource()==B[i])
            B[i].setText(S[i]+" düğmeye basıldı");
            else B[i].setText(S[i]);
        }
    }
}
```

```

validate();
}
public static void main(String[] args)
{

    JTabbedPaneTest f = new JTabbedPaneTest("JTabbedPane formatlama Grafik ara yüz (GUI) testi");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );    f.setSize(800,500);
    f.setVisible(true);
}
}

```



JScrollPane format sınıfı

JScrollPane sınıfını daha önce JList ile birlikte kullanmıştık. Bu yüzden sadece ek örnek vermekle yetineceğiz. Hatırlanacağı gibi bileşik kullanım

add(new JScrollPane(inputPanel)); şeklindeydi

In the example program, JScrollPane will be added to a previously used example

Program 8.10.8 JScrollPaneTest örneği

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class JScrollPaneTest extends JApplet implements ItemListener
{
    JPanel inputPanel;
    JTextField t;
    JCheckBox bold,italic;
    int normal_font,bold_font,italic_font;
    String s;
    public void init()

```

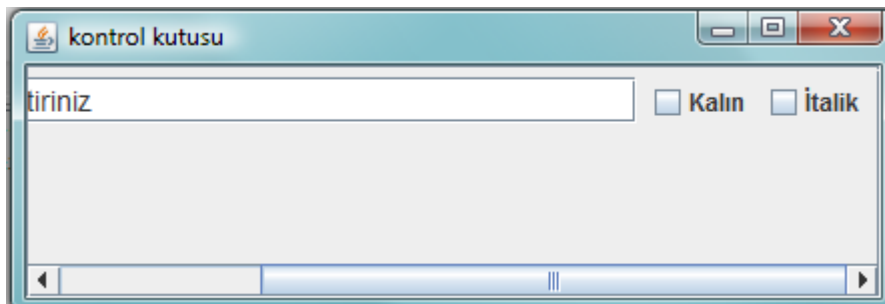
```

{
    s="Bu yazının stilini deęiřtiriniz";
    inputPanel=new JPanel();
    t=new JTextField(s,40);
    italic_font=Font.PLAIN;
    bold_font=Font.PLAIN;
    t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
    bold=new JCheckBox("Kalın");
    italic=new JCheckBox("İtalik");
    inputPanel.add(t);
    inputPanel.add(bold);
    inputPanel.add(italic);
    add(new JScrollPane(inputPanel));
    italic.addItemListener(this);
    bold.addItemListener(this);
}

public void itemStateChanged(ItemEvent e)
{
    if(e.getSource()==bold)
    {
        if(e.getStateChange()==ItemEvent.SELECTED)
            {bold_font=Font.BOLD;}
        else
            {bold_font=Font.PLAIN;}
    }
    else if(e.getSource()==italic)
    {
        if(e.getStateChange()==ItemEvent.SELECTED)
            {italic_font=Font.ITALIC;}
        else
            {italic_font=Font.PLAIN;}
    }
    t.setFont(new Font("TimesRoman",bold_font+italic_font,14));
    repaint();
}

public static void main(String s[])
{
    JFrame f = new JFrame("kontrol kutusu");
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    JApplet applet = new JScrollPaneTest();
    f.getContentPane().add("Center", applet);
    applet.init();
    f.setSize(800,500);
    f.setVisible(true);
}
}

```



8.11 ALIŖTIRMALAR

1) **toplaA.java** programını inceleyiniz **JFrame** kullanarak o dan n e kadar sayıların karelerinin toplamının karekökünü hesaplayan bir program yazınız.

2) **tersA1.java** programını inceleyiniz. Bu kodu biraz deęiştirerek düğmeye her basıŖya farklı büyüklüklerde karakterler verecek bir program yazınız.

3) **duymeF1.java** kodunu inceleyiniz. Yıldız yerine eşkenar dörtgen çizecek **squareP.java** programını geliřtiriniz. Test programını da yazarak çalıştırınız.

4) Listelenen program menü ve fare kullanarak yıldız. Çizer yıldızın rengi ve köře sayısı menülerden ayarlanmaktadır. Fare basılıp çekilerek ve bırakılarak yıldız oluşturulur. Farenin ilk basılıŖı nokta yıldızın merkezi, çekildikten sonra bırakıldıęı nokta ise yıldızın çapını oluşturur.

```
public class polar
{
// polar coordinates
// teta radyan
protected double R,teta;

public polar()
{
R=0;
teta=0;
}

public polar(double Ri,double tetai)
{
R=Ri;
teta=tetai;
}

public polar(polar y)
{
R=y.ROku();
teta=y.tetaOku();
}

public void R_input(double Ri)
{
R=Ri;
}

public void teta_input(double tetai)
{
teta=tetai;
}

public void polar_input(double Ri,double tetai)
{
R=Ri;
teta=tetai;
}

public void kartesien_input(double xi,double xj)
```



```

{
R=Math.sqrt(xi*xi+xj*xj);
teta=Math.atan2(xj,xi);
}

public double ROku()
{
return R;
}

public double tetaOku()
{
return teta;
}

public double xi()
{
return R*Math.cos(teta);
}

public double xj()
{
return R*Math.sin(teta);
}

public void topla(polar y)
{
kartesien_input((xi()+y.xi()),(xj()+y.xj()));
}

public void fark(polar y)
{
kartesien_input((xi()-y.xi()),(xj()-y.xj()));
}

public polar polarOku()
{
return this;
}

public boolean esittir(polar v)
{
boolean b=((R==v.ROku())&&(teta==v.tetaOku()));
return b;
}

public boolean buyuktur(polar v)
{
return (this.R>v.ROku());
}

public boolean kucuktur(polar v)
{
return (this.R<v.ROku());
}

```

```
import java.io.*;
```

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

public class yildizP2 extends JPanel implements MouseListener,MouseMotionListener
{
    int x1,y1,x2,y2;
    int yildizcapi;
    renk c;
    int side;
    boolean ilk;
    GeneralPath x;

    public yildizP2(renk c1,int n)
    {super();
    side=n;
    c=c1;
    x1=100;y1=100;x2=x1;y2=y1+30;
    ilk=true;
    addMouseListener(this);
    addMouseMotionListener(this);
    }

    public void renksec(renk c1)
    {c=c1;}

    public void setside(int n)
    {side=n;}

    public void setCizgi(int x1i,int y1i,int x2i,int y2i)
    {x1=x1i;y1=y1i;x2=x2i;y2=y2i;}

    public GeneralPath yildiz(int x,int y, int n,int yildizcapi,double aci)
    {// bu yıldız çizime teta=pi/2+aci radyandan baslar
    GeneralPath yildiz1=new GeneralPath();
    double teta=2.0*Math.PI/n;
    double R=yildizcapi;
    double r=yildizcapi*0.25;
    polar P1=new polar();
    polar P2=new polar();
    polar P3=new polar();
    for(int i=0;i<n;i++)
    {
        double teta1=teta*i+Math.PI/2.0+aci;
        double teta2=teta/2+teta1;
        P1.polar_input(R,teta1);
        P2.polar_input(r,teta2);
        if(i==0) yildiz1.moveTo((x+(int)P1.xi()),(y-(int)P1.xj()));
        else yildiz1.lineTo((x+(int)P1.xi()),(y-(int)P1.xj()));
        yildiz1.lineTo((x+(int)P2.xi()),(y-(int)P2.xj()));
    }
    }

```

```

    double teta3=teta*(i+1)+Math.PI/2.0+aci;
    P3.polar_input(R,teta3);
    yildiz1.lineTo((x+(int)P3.xi()),(y-(int)P3.xj()));
    }
    yildiz1.closePath();
    return yildiz1;
    }

public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    Graphics2D g2=(Graphics2D)g;
    g2.setFont(new Font("Serif",Font.BOLD,24));
    g2.setColor(c);
    g2.setStroke(new BasicStroke(2.0f));
    yildizcapi=(int)Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
    x=yildiz(x1,y1,side,yildizcapi,0);
    g2.fill(x);
    }

//MouseListener (fare dinleyicisi)
public void mouseClicked(MouseEvent e)
{ }

public void mousePressed(MouseEvent e)
{
    x1=e.getX();y1=e.getY();}

public void mouseReleased(MouseEvent e)
{ x2=e.getX();y2=e.getY();repaint();}

public void mouseEntered(MouseEvent e)
{ }

public void mouseExited(MouseEvent e)
{ }

//MouseMotionListener methods

public void mouseDragged(MouseEvent e)
{ }
public void mouseMoved(MouseEvent e)
{ }

}

```

```

import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.awt.geom.*;

```

```

public class yildiz1 extends JApplet implements ListSelectionListener
{
    yildizP2 sekil;
    JMenuBar menuBar;
    JMenu m1,m2;
    JPanel outputPanel;
    JList sekilListesi;
    JList renkListesi;
    String sekilNames[]={ "3 köşe yıldız", "4 köşe yıldız", "5 köşe yıldız",
        "6 köşe yıldız", "7 köşe yıldız", "8 köşe yıldız", "9 köşe yıldız", "10 köşe yıldız"};
    String renk isimleri[]={ "kırmızı", "mavi", "siyah", "camgöbeği", "açık_camgöbeği", "koyugri", "gri", "yeşil", "açıkgri", "portakal",

    "pembe", "beyaz", "sarı", "açıkmavi", "lacivert", "mor", "turquaz", "menekşe", "mavi_menekşe", "patlıcan", "buğday", "ten", "gök_
    mavi",

    "açık_gök_mavisi", "deniz_kabuğu", "deniz_yeşili", "açık_deniz_yeşili", "zeytin_yeşili", "fildişi", "lavanta", "bej", "kahverengi",
    "çikolata", "mercan", "açık_yeşil", "alev_kırmızısı", "alev_turuncusu", "alizarin", "altuni", "ametist", "armut", "akuamarin",
    "asker_yeşili", "bakır", "bordo", "bronz", "çivit_mavisi", "gece_mavisi", "gül", "hardal", "havuç", "ilkbahar_yeşili", "kabak",

    "karanfil", "kayısı", "kehribar", "kobalt_mavisi", "krem", "limon", "açık_limon", "leylak", "mandalina", "nane_yeşili", "nar", "orkid
    e",
    "pas", "safran", "safır", "tarçın", "turuncu", "zümrüt_yeşili", "şarap"};
    renk
    renkler[]={ renk.kırmızı, renk.mavi, renk.siyah, renk.camgöbeği, renk.açık_camgöbeği, renk.koyugri, renk.gri, renk.yeşil, renk.açı
    kgri, renk.portakal,

    renk.pembe, renk.beyaz, renk.sarı, renk.açıkmavi, renk.lacivert, renk.mor, renk.turquaz, renk.menekşe, renk.mavi_menekşe, renk.
    patlıcan, renk.buğday, renk.ten, renk.gök_mavi,

    renk.açık_gök_mavisi, renk.deniz_kabuğu, renk.deniz_yeşili, renk.açık_deniz_yeşili, renk.zeytin_yeşili, renk.fildişi, renk.lavanta,
    renk.bej, renk.kahverengi,

    renk.çikolata, renk.mercan, renk.açık_yeşil, renk.alev_kırmızısı, renk.alev_turuncusu, renk.alizarin, renk.altuni, renk.ametist, renk
    .armut, renk.akuamarin,

    renk.asker_yeşili, renk.bakır, renk.bordo, renk.bronz, renk.çivit_mavisi, renk.gece_mavisi, renk.gül, renk.hardal, renk.havuç, renk.
    ilkbahar_yeşili, renk.kabak,

    renk.karanfil, renk.kayısı, renk.kehribar, renk.kobalt_mavisi, renk.krem, renk.limon, renk.açık_limon, renk.leylak, renk.mandalina,
    renk.nane_yeşili, renk.nar, renk.orkide,
    renk.pas, renk.safran, renk.safır, renk.tarçın, renk.turuncu, renk.zümrüt_yeşili, renk.şarap};

    public void init()
    {
        int i=3;
        sekil=new yildizP2(renk.mavi,i);
        renkListesi=new JList(renk isimleri);
        renkListesi.setVisibleRowCount(4);
        sekilListesi=new JList(sekilNames);
        sekilListesi.setVisibleRowCount(4);
        outputPanel=new JPanel();
        outputPanel.setLayout(new BorderLayout());
        outputPanel.add(sekil, BorderLayout.CENTER);
        menuBar=new JMenuBar();
        setJMenuBar(menuBar);
        m1 = new JMenu("Color List");
        m1.setMnemonic('R');
    }
}

```

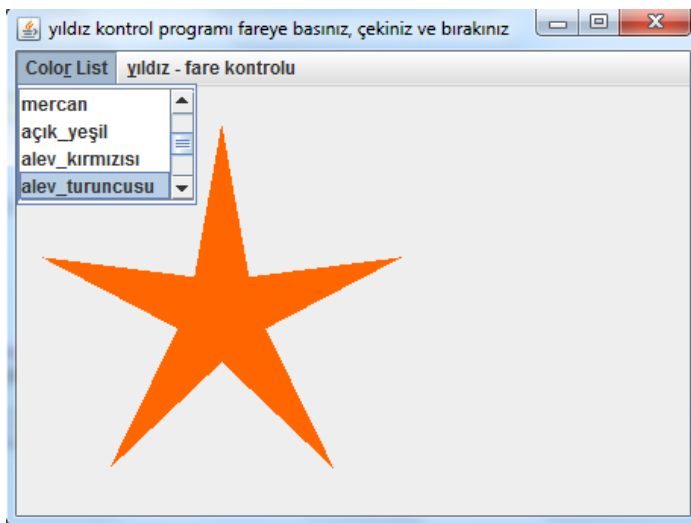
```

m1.getAccessibleContext().setAccessibleDescription("yıldızların hangi renkte olacağını seçer");
m1.add(new JScrollPane(renklistesi));
menuBar.add(m1);
m2 = new JMenu("yıldız - fare kontrolü");
m2.setMnemonic('Y');
m2.getAccessibleContext().setAccessibleDescription("yıldızların kaç köşeli olacağını seçer");
m2.add(new JScrollPane(sekillistesi));
menuBar.add(m2);
add(outputPanel, BorderLayout.CENTER);
renklistesi.addListSelectionListener(this);
sekillistesi.addListSelectionListener(this);
}

public void valueChanged( ListSelectionEvent e)
{
if(e.getSource()==renklistesi)
{
renk r1=renkler[renklistesi.getSelectedIndex()];
sekil.renksec(r1);
}
if(e.getSource()==sekillistesi)
{sekil.setside(sekillistesi.getSelectedIndex()+3);}
repaint();
}

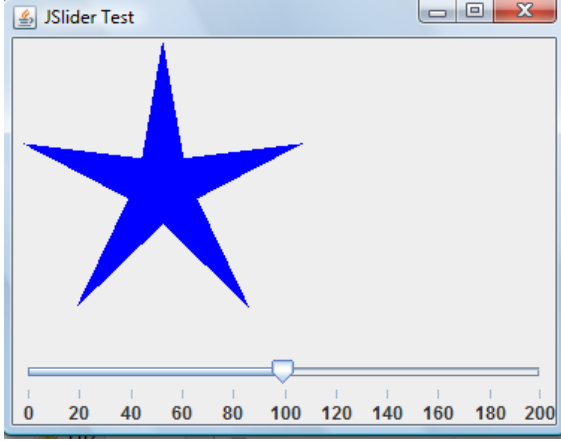
public static void main(String s[])
{
JFrame f = new JFrame("yıldız kontrol programı fareye basınız, çekiniz ve bırakınız");
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
JApplet applet = new yildiz1();
f.getContentPane().add("Center", applet);
applet.init();
f.pack();
f.setSize(new Dimension(545,400));
f.setVisible(true);
}
}

```

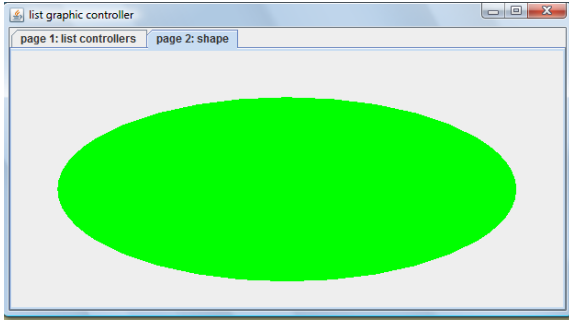


Bu programı inceleyiniz, ve programı çeşitli çokgenleri çizdirecek şekle dönüştürünüz (üçgen, dörtgen, beşgen, altıgen...). Bunun için yıldızP programını değiştirerek cokgenP programını oluşturun.

5) Problem 8.8.1 de ovalP, daire çizen bir JPanel sınıfı verilmişti. Benzer olarak JSlider kullanarak yıldızın çapını büyütüp küçülten bir program yazınız.



6) sekilP ve listA1 programlarını inceleyiniz. Yebni bir programda JTabbedPane formatıyla iki sayfalık bir format yapısı hazırlayın. Birinci sayfada liste kontrolleri yer alsın, ikinci sayfada şekil çizilsin.



8.12 PROBLEMLER

1) Bir grafik ortamı programında iki JTextField alanına iki sayı girilsin ve ve bir JTextArea alanında sayıların toplamı gösterilsin.

2) Bir fonksiyonun türevini ve integralini hesaplayan sayisal.java programı önceki bölümlerimizde verilmişti. Grafik ekranına (JTextField olarak) bir sayı giriniz ve hesaplana $f(x)=x*x-5$ fonksiyonu ve türevinin değerleri JTextArea penceresinde yazdırılsın.

```
public class sayisal
{
public static double turev(f_x f,double x)
{ double h=0.00001;
return (-f.func(x+2.0*h)+8.0*f.func(x+h)-8.0*f.func(x-h)+f.func(x-2.0*h))/(12.0*h);
}

public static double integral(f_x f_xnt,double a,double b)
{
// integral f(x) dx
double r[]={-0.973906528517171,-0.865063366688984,-0.679409568299024,-0.433395394129247,-0.148874338981631,
0.148874338981631,0.433395394129247,0.679409568299024,0.865063366688984,0.973906528517171};
```

```
double c[]={0.066671344308684,0.149451349150580,0.219086362515982,0.269266719309996,0.295524224714752,
0.295524224714752,0.269266719309996,0.219086362515982,0.149451349150580,0.066671344308684};
double z=0,x,y;
double k1=(b-a)/2.0;
double k2=(b+a)/2.0;
for(int i=0;i<r.length;i++)
{
x=k2+k1*r[i];
y=f_xnt.func(x);
z+=k1*c[i]*y;
}
return z;
}
}
```

```
class fb extends f_x
{ public double func(double x)
{ return x*x-5.0;}}
```

3) **duymeF4.java** programını inceleyin ve JApplet formuna dönüştürün.

9 HATA KONTROLÜ

9.1 GENEL TANIMLAR

Java dili yapısında hata kontrol mekanizmasına yer verilmiştir. Bunun genel olarak Exception adını alan bir grup sınıf tanımıyla yapılırlar. Temel hata yakalama yapısı try {...} catch gurubuyla olur. Bu guruptan sonra hata oluştuğunda belirli bir hata (Exception sınıfından bir değişken çağrılır ve gerektiğinde hata mesajı verilir. Genel yapı şu şekilde gösterilebilir.

```
try {  
    java deyimleri;  
}  
catch( herhangi_bir__java_Exception_types e1)  
{ hata kontrol deyimleri }  
( herhangi_bir__java_Exception_types e2)  
{ hata kontrol deyimleri }  
finally  
{ hata kontrol deyimleri }
```

Java Kütüphanesinde hazır olarak verilen hata kontrol sınıfları aşağıda listelenmiştir.:

Java.lang hata listesi

AbstractMethodError
ClassCircularityError
ClassFormatError
Error
ExceptionInInitializerError
IllegalAccessError
IncompatibleClassChangeError
InstantiationError
InternalError
LinkageError
NoClassDefFoundError
NoSuchFieldError
NoSuchMethodError
OutOfMemoryError
StackOverflowError
ThreadDeath
UnknownError
UnsatisfiedLinkError
VerifyError
VirtualMachineError

Java.lang Exception listesi

ArithmeticException
ArrayIndexOutOfBoundsException
ArrayStoreException

ClassCastException
ClassNotFoundException
CloneNotSupportedException
Exception
IllegalAccessException
IllegalArgumentException
IllegalMonitorStateException
IllegalStateException
IllegalThreadStateException
IndexOutOfBoundsException
InstantiationException
InterruptedException
NegativeArraySizeException
NoSuchFieldException
NoSuchMethodException
NullPointerException
NumberFormatException
RuntimeException
SecurityException
StringIndexOutOfBoundsException

Java.util exception listesi

EmptyStackException
MissingResourceException
NoSuchElementException
TooManyListenersException

Java.io exception list

CharConversionException
EOFException
FileNotFoundException
IOException
InterruptedIOException
InvalidClassException
InvalidObjectException
NotActiveException
NotSerializableException
ObjectStreamException
OptionalDataException
StreamCorruptedException
SyncFailedException
UTFDataFormatException
UnsupportedEncodingException
WriteAbortedException

Herbir hata sınıfının ne yaptığı burada tek tek tanımlanmamıştır. Java döküman kütüphanelerinden detaylı olarak incelenebilir. Bloкта tanımlanmış olan hata sınıfında bir hata oluştuğunda hata try-catch gurubu tarafından yakalanır. Hata oluşturan metodların girdi kısmında **throws** sözcüğü kullanılarak bir hata sinyalinin gönderileceği belirtilir. Bu belirtilmişse, bu metod bir try-catch bloğu içinde çağrılmalıdır. Bu yapının açıklanmasının en iyi yolunun örnekler üzerinden yapılacağına inanıyoruz. İlk örneğimizde önce kendi hata sınıfımızı kütüphane hata sınıfı olan **ArithmeticException** (matematik hatası) sınıfından türettik bu sınıfa sifira bolmehatasi ismini verdik. **sifirbolmehatasi**

When an error happened in java, it will carried to try-catch group with the same Exception group given at the description of the method error happend. Exception with the keyword **throws** will also be located after the name and input block of the method. Inside of the method when and where error supposed to be happen keyword throw and a new Exception object is used. In order to explain these structures, few examples will be given in the first example two numbers are divided. If the divisor is zero number goes to infinity. In order to avoid this, an exception can be used. This exception will be created as a class extended from ArithmeticException class

Program 9.1.1 sifirabolmehatasi sınıfı

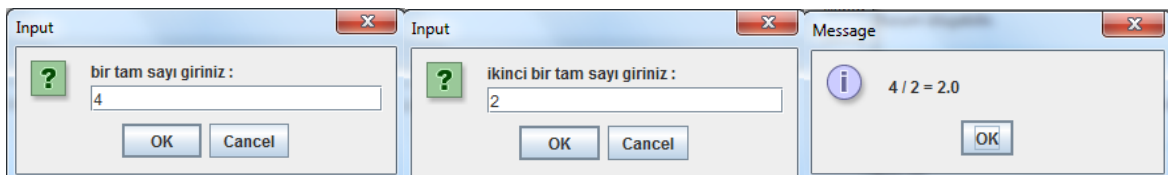
```
public class sifirabolmehatasi extends ArithmeticException
{
    public sifirabolmehatasi() { super("sayı sifira (0) bölündü "); }
}
```

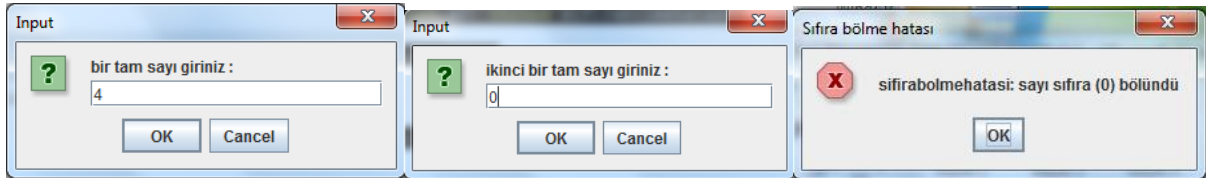
Program 9.1.2 sifirabolmetesti sınıfı

```
import javax.swing.*;
import javax.swing.table.*;
import java.text.DecimalFormat;

public class sifirabolmetesti
{
    public static double bol(int s1,int s2) throws sifirabolmehatasi
    {
        if(s2==0) throw new sifirabolmehatasi();
        return (double) s1/s2;
    }

    public static void main(String args[])
    {
        String s;
        double sonuc=0;
        int sayi1,sayi2;
        sayi1=Integer.parseInt(
        JOptionPane.showInputDialog("bir tam sayı giriniz : "));
        sayi2=Integer.parseInt(
        JOptionPane.showInputDialog("ikinci bir tam sayı giriniz : "));
        try{
            sonuc=bol(sayi1,sayi2);
            JOptionPane.showMessageDialog(null,sayi1+" / "+sayi2+" = "+sonuc);
        } catch(sifirabolmehatasi e1) {JOptionPane.showMessageDialog(null,e1.toString(),"Sıfıra
        bölme hatası",JOptionPane.ERROR_MESSAGE);}
    }
}
```





İkinci örneğimizde iki hata deyimi birlikte kullanılacak. Birincisi bir önceki örnekte kullandığımız sifira bölme hatası, ikincisi hazır bir hata sınıfı olan NumberFormatException sınıfı. Bu sınıf sayı girişi yanlış yapılırsa (örneğin tam sayı yerine gerçek sayı girişi yapılırsa) devreye girer.

In the second Example two Exception will be used in series. The first one will be the dividebyzeroException defined above, the second one will be NumberFormatException. Two Number required in the program below are integers, in case any any number or string entered, A Number format exception will be thrown. In this program, while loop is also used loop continues until correct input statements are given.

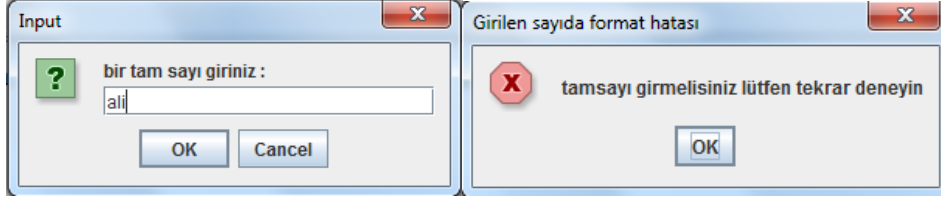
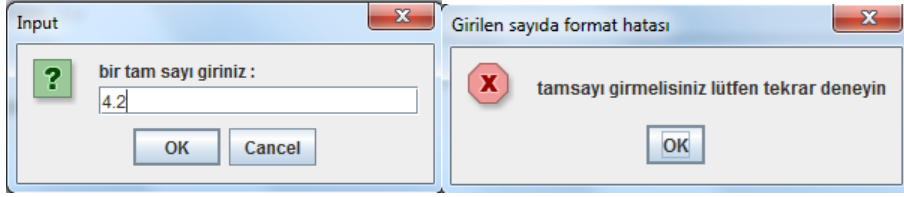
Program 9.1.3 dividebyzeroTest1 class

```
import javax.swing.* ;
import java.text.DecimalFormat;

public class dividebyzeroTest1
{
public static double divide(int s1,int s2) throws dividebyzeroException
{
if(s2==0) throw new dividebyzeroException();
return (double) s1/s2;
}

public static void main(String args[])
{
String s;
double result=0;
int number1,number2;
boolean continueloop=true;
while(continueloop)
{
try{
number1=Integer.parseInt(
JOptionPane.showInputDialog("enter an integer number : "));
number2=Integer.parseInt(
JOptionPane.showInputDialog("enter a second integer number : "));
result=divide(number1,number2);
continueloop=false;
JOptionPane.showMessageDialog(null,number1+" / "+number2+" = "+result);
}
catch(NumberFormatException e2)
{JOptionPane.showMessageDialog(null,"You must enter integers, please try
again","ERROR",JOptionPane.ERROR_MESSAGE);}
catch(dividebyzeroException e1)
{JOptionPane.showMessageDialog(null,e1.toString(),"ERROR",JOptionPane.ERROR_MESSAGE);}
}
}
```

```
}  
}
```



Bir sonraki örnekte sınıf yapısı hazırlanırken hata yapısı integra olarak hazırlanarak sınıf yapısıyla birlikte oluşturulmuştur.

In the third example a class previously used to explain class structure, box class will be used with zero_or_negative_sizeException that will be created from ArithmeticException.

Program 9.1.4 class sifir_veya_eksi_boyut_hatasi hata sınıfı

```
public class sifir_veya_eksi_boyut_hatasi extends ArithmeticException  
{  
    public sifir_veya_eksi_boyut_hatasi() { super("verilen boyut 0 veya eksi "); }  
}
```

Program 9.1.5 sifir_veya_eksi_boyut_hatasi hata sınıfı ile kontrol edilen jutu2 sınıfı

```
class kutu2  
{  
    private double boy;  
    private double en;  
    private double yukseklik;  
  
    kutu2(double l,double w, double h) throws sifir_veya_eksi_boyut_hatasi  
    { if(l<=0 || w<=0 || h<=0) throw new sifir_veya_eksi_boyut_hatasi();  
      else  
      {  
        boy=l;  
        en=w;  
        yukseklik=h;  
      }  
    }  
  
    kutu2(kutu2 bi) throws sifir_veya_eksi_boyut_hatasi  
    {if(bi.boy<=0 || bi.en<=0 || bi.yukseklik<=0) throw new sifir_veya_eksi_boyut_hatasi();  
      else  
      {  
        boy=bi.boy;  
        en=bi.en;  
        yukseklik=bi.yukseklik;  
      }  
    }  
}
```

```

// This methods are written to access private boy,
// en and yukseklik variables
public double oku_boy() {return boy;}
public double oku_en() {return en;}
public double oku_yukseklik() {return yukseklik;}
//This methods are written to change private boy,
// en and yukseklik variables
public void gir_boy(double l) throws sifir_veya_eksi_boyut_hatasi
{if(l<=0 ) throw new sifir_veya_eksi_boyut_hatasi();
boy=l;
}
public void gir_en(double w) throws sifir_veya_eksi_boyut_hatasi
{if(w<=0 ) throw new sifir_veya_eksi_boyut_hatasi();
en=w;
}
public void gir_yukseklik(double h) throws sifir_veya_eksi_boyut_hatasi
{if(h<=0 ) throw new sifir_veya_eksi_boyut_hatasi();
yukseklik=h;
}

double volume()
{return boy*en*yukseklik;}

public String toString()
{ String s="";
s+="boy = "+boy+"\n";
s+="en = "+en+"\n";
s+="yukseklik = "+yukseklik+"\n";
s+="Volume = "+volume()+"\n";
return s;
}
}

```

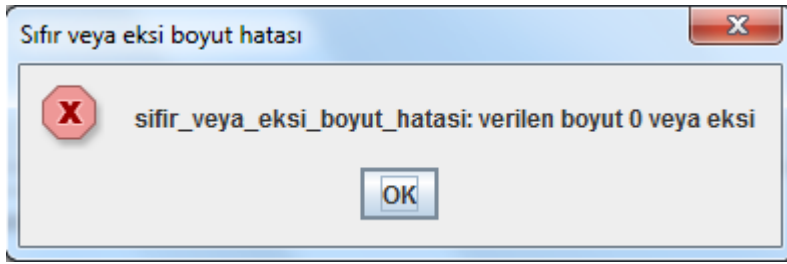
Program 9.1.6 kututesti6

```

import javax.swing.JOptionPane;

class kututesti6
{
    public static void main(String args[])
    {
        double volume;
        try{
            kutu2 k= new kutu2(1.0,1.2,0.2);
            String s="kutunun orijinal deđeri : \n"+k.cikti()+"\n";
            k.gir_boy(3.0);
            k.gir_en(2.0);
            k.gir_yukseklik(0.0);
            s+="kutunun deđiřtirilmiř deđeri : \n"+k.toString()+"\n";
            JOptionPane.showMessageDialog(null,s,
            "sımf 6rneđi kutu testi6 ",JOptionPane.PLAIN_MESSAGE);
        } catch(sifir_veya_eksi_boyut_hatasi e1)
        { JOptionPane.showMessageDialog(null,e1.toString(),"Sifir veya eksi boyut hatasi",JOptionPane.ERROR_MESSAGE);}
        System.exit(0);
    }
}

```



The last example will be classes circle and cylinder, **zero_or_negative_sizeException** will also be used with these classes.

Program 9.1.7 daire2 sınıfı

```
public class daire2 extends nokta1
{
    public double yaricap;
    // bu sınıfta xi ve xj olmak üzere
    // iki tane daha sınıf değişkeni var
    public daire2()
    { super(0.0,0.0);
      yaricapgir(0.0);}

    public daire2(double r, double a, double b) throws sifir_veya_eksi_boyut_hatasi
    { super(a,b);
      yaricapgir(r);
    }
    public daire2(daire1 d) throws sifir_veya_eksi_boyut_hatasi
    { super(d.xi,d.xj);
      yaricapgir(d.yaricap);
    }
    public daire2(double r, nokta1 n) throws sifir_veya_eksi_boyut_hatasi
    { super(n);
      yaricapgir(r);
    }

    public void yaricapgir(double r) throws sifir_veya_eksi_boyut_hatasi
    { if(r > 0.0)
      yaricap=r;
      else
      throw new sifir_veya_eksi_boyut_hatasi();
    }

    public double alan()
    {return Math.PI*yaricap*yaricap;}
    public double hacim(){return 0.0;}
    public String cikti()
    { return "Merkez = "+super.cikti()+" yarıçap = "+yaricap;}

    public String isim(){return "daire";}
}
```

Program 9.1.8 silindir2 sınıfı

```
public class silindir2 extends daire2
{
    protected double yukseklik;
```

```

public silindir2() throws sifir_veya_eksi_boyut_hatasi
{super(0.0,0.0,1.0);
yuksekligir(1.0);}

public silindir2( double h,daire1 d) throws sifir_veya_eksi_boyut_hatasi
{super(d);
yuksekligir(h);
}

public silindir2( double h, double r, double a, double b) throws sifir_veya_eksi_boyut_hatasi
{super(r,a,b);
yuksekligir(h);
}

public void yuksekligir (double h ) throws sifir_veya_eksi_boyut_hatasi
{ if(h>0) yukseklik=h;
else throw new sifir_veya_eksi_boyut_hatasi();
}

public double alan()
{return 2*super.alan()+2*Math.PI*yaricap*yukseklig;}

public double hacim(){return super.alan()*yukseklig;}

public String cikti()
{return super.cikti()+" yukseklik = "+yukseklig;}

public String isim(){return "silindir";}
}

```

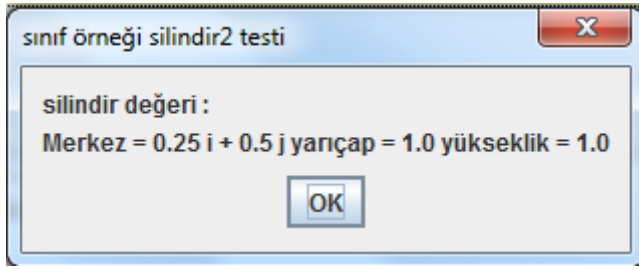
Program 9.1.9 silindirtesti sınıfı

```

import javax.swing.JOptionPane;

class silindirtesti
{
public static void main(String args[])
{
double volume;
try{
silindir2 k= new silindir2(1.0,1.0,0.25,0.5);
String s="silindir değeri : \n"+k.cikti()+"\n";
JOptionPane.showMessageDialog(null,s,
"sınıf örneği silindir2 testi ",JOptionPane.PLAIN_MESSAGE);
} catch(sifir_veya_eksi_boyut_hatasi e1)
{ JOptionPane.showMessageDialog(null,e1.toString(),"Sıfır veya eksi boyut hatası",JOptionPane.ERROR_MESSAGE);}
System.exit(0);
}
}

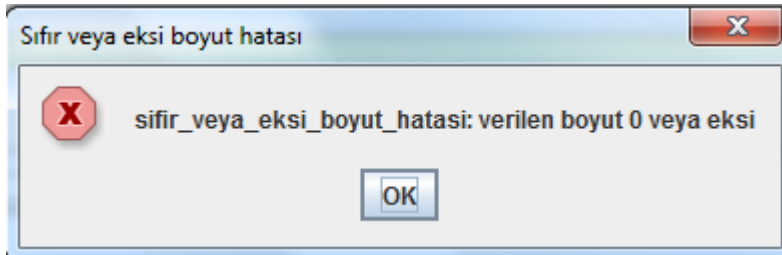
```



Program 9.1.10 silindirtesti1 sınıfı

```
import javax.swing.JOptionPane;

class silindirtesti1
{
    public static void main(String args[])
    {
        double volume;
        try{
            silindir2 k= new silindir2(0.0,1.0,2.0,0.5);
            String s="silindir değeri : \n"+k.cikti()+"\n";
            JOptionPane.showMessageDialog(null,s,
                "sınıf örneği silindir2 testi ",JOptionPane.PLAIN_MESSAGE);
        } catch(sifir_veya_eksi_boyut_hatasi e1)
        { JOptionPane.showMessageDialog(null,e1.toString(),"Sıfır veya eksi boyut hatası",JOptionPane.ERROR_MESSAGE);}
        System.exit(0);
    }
}
```



Son örnek olarak bayağı kesirleri vereceğiz. Bayağı kesirler tanımları icabı eğer payda 0 olarak verilmişse hata oluştururlar. Bu hatayı sıfıra bölme hatası kullanarak giderebiliriz.

Program 9.1.11 bayagikesir1 sınıfı

```
=====
// bayagikesir1 sinifi tanimi
// Dr. Turhan Coban
//=====
import java.io.*;
// sinif bayagikesir
// bu sinif bayagikesir sayilarin matematik
// islemlerini tanimlar
//
class bayagikesir1 {
    // sinif degiskenleri
    int pay;
    int payda;
    // kurucu metotlar
```



```

public bayagikesir1()
{
pay=0;
payda=1;
}

public bayagikesir1(int npay,int npayda) throws sifirbolmehtasi
{
pay=npay;
payda=npayda;
if(payda==0) throw new sifirbolmehtasi();
}

public bayagikesir1(int tamsayi,int npay,int npayda) throws sifirbolmehtasi
{
payda=npayda;
pay=npay+tamsayi*npayda;
if(payda==0) throw new sifirbolmehtasi();
}

public bayagikesir1(int numer)
{
pay=numer;
payda=1;
}
public bayagikesir1(bayagikesir1 c ) throws sifirbolmehtasi
{
pay=c.Pay();
payda=c.Payda();
if(payda==0) throw new sifirbolmehtasi();
}

// giriş- çıkış metodları
public int enbuyukortakbolen()
{
int n=pay;
int m=payda;
// iki tam sayinin en buyuk ortak bolenini hesaplar
if(n==0)
return m;
if(m==0)
return n;
while(m != n)
{
if(n>m)
n=n-m;
else
m=m-n;
}
return n;
}

public void sadelestir()
{
//sadelestir
int isaret=1;

```

```

if(pay<0)
{
isaret=-isaret;
pay=-pay;
}
if(payda<0)
{
isaret=-isaret;
payda=-payda;
}
if(payda==0) {pay=0;payda=1;}
int ebob=enbuyukortakbolen();
ebob=Math.abs(ebob);
pay=isaret*pay/ebob;
payda=payda/ebob;
}

public int Pay()
{
return pay;
}

public int Payda()
{
return payda;
}

public void payGir(int r)
{
pay=r;
}

public void paydaGir(int i)
{
payda=i;
}
public void bayagikesirGir(bayagikesir1 sag) throws sifirabilmehatasi
{
pay=sag.Pay();
payda=sag.Payda();
if(payda==0) throw new sifirabilmehatasi();
}

public void bayagikesirGir(int nr,int ni) throws sifirabilmehatasi
{
pay=nr;
payda=ni;
if(payda==0) throw new sifirabilmehatasi();
}

public void bayagikesirGir(int nr)
{
pay=nr;
payda=1;
}

public void bayagikesirGir(double d)
{

```

```
// tam sayinin bayagikesir esitini yükle
// eh birazcık yaklaşım var tabi
bayagikesirGir((int)d*10000,10000);
sadelestir();
}

public double toDouble()
{
//bayagikesir sayinin gercek sayi esidi
return ((double)pay/(double)payda);
}

public static bayagikesir1 BayagikesireCevir(double d)
{
// tam sayinin bayagikesir esiti
// eh birazcık yaklaşım var tabi
bayagikesir1 b=new bayagikesir1();
b.bayagikesirGir((int)d*10000,10000);
b.sadelestir();
return b;
}

public void topla(bayagikesir sag)
{
//ikinci bir bayagikesirle topla
pay = pay*sag.Payda() + sag.Pay()*payda;
payda = payda*sag.Payda();
sadelestir();
}

public void topla(int sag)
{
//bir gercek sayıyla topla
pay = pay + sag*payda;
sadelestir();
}

public void cikar(bayagikesir sag)
{
//ikinci bir bayagikesirle topla
pay = pay*sag.Payda() - sag.Pay()*payda;
payda = payda*sag.Payda();
sadelestir();
}

public void cikar(int sag)
{
//bir gercek sayıyla topla
pay = pay - sag*payda;
sadelestir();
}

public void carp(bayagikesir sag )
{
//bir bayagikesir sayıyla carp
pay = pay*sag.Pay();
payda = payda*sag.Payda();
sadelestir();
}
```

```

}

public void carp(int sag )
{
//bir gercek sayıyla carp
pay = pay*sag;
sadelestir();
}

public void bol(bayagikesir sag )
{
//bir bayagikesir sayıyla böl
pay = pay*sag.Payda();
payda = payda*sag.Pay();
sadelestir();
}

public void bol(int sag )
{
//bir pay sayıyla böl
payda = payda*sag;
sadelestir();
}

public static bayagikesir1 topla(bayagikesir sol, bayagikesir sag)
{
// iki bayagikesir sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1= sol.Pay() * sag.Payda() + sag.Pay() * sol.Payda();
int i1= sol.Payda()* sag.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 topla(bayagikesir sol, int sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sol.Pay() + sag*sol.Payda();
int i1=sol.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 topla(int sol, bayagikesir1 sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sag.Pay() + sol*sag.Payda();
int i1=sag.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

```

```

}
//=====

public static bayagikesir1 cikar(bayagikesir sol, bayagikesir1 sag)
{
// iki bayagikesir sayının farkını return deyimiyle bayagikesir olarak
// aktarır
int r1= sol.Pay() * sag.Payda() - sag.Pay() * sol.Payda();
int i1= sol.Payda()* sag.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 cikar(bayagikesir1 sol, int sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sol.Pay() - sag*sol.Payda();
int i1=sol.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 cikar(int sol, bayagikesir1 sag)
{
// bir bayagikesir ve bir Pay sayının toplamını return deyimiyle bayagikesir olarak
// aktarır
int r1=sag.Pay() - sol*sag.Payda();
int i1=sag.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1(r1,i1);
sonuc.sadelestir();
return sonuc;
}

//=====

public static bayagikesir1 carp(bayagikesir1 sol, bayagikesir sag)
{ // iki bayagikesir sayının carpimini aktar
bayagikesir1 sonuc;
sonuc=new bayagikesir1(sol.Pay()*sag.Pay(),sol.Payda()*sag.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 carp(bayagikesir1 sol, int sag)
{ // bir bayagikesir ve bir double sayinin carpimini aktar
bayagikesir1 sonuc;
sonuc=new bayagikesir1(sol.Pay()*sag.sol.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 carp(int sol, bayagikesir sag)
{ // bir bayagikesir ve bir double sayinin carpimini aktar

```

```

bayagikesir1 sonuc;
sonuc=new bayagikesir1(sag.Pay()*sol,sag.Payda());
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 bol(bayagikesir1 sol, bayagikesir1 sag)
{ // iki bayagikesir sayinin bölümlerini aktar
double a=sag.Pay()*sag.Pay()+sag.Payda()*sag.Payda();
bayagikesir1 sonuc;
sonuc=new bayagikesir1((int)(sol.Pay()*sag.Payda()),(int)(sol.Payda()*sag.Pay()));
sonuc.sadelestir();
return sonuc;
}

public static bayagikesir1 bol(bayagikesir1 sol, int sag)
{ // bayagikesir sayıyı double sayıya bol
bayagikesir1 sonuc;
sonuc=new bayagikesir1(sol.Pay(),(sol.Payda()*sag));
sonuc.sadelestir();
return sonuc;
}

public boolean kucuktur(bayagikesir1 sol,bayagikesir1 sag)
{
// less then karşılaştırması of two bayagikesir numbers
return (sol.toDouble() < sag.toDouble());
}

public boolean kucuktur_esittir(bayagikesir1 sol,bayagikesir1 sag)
{
// less then and esittir karşılaştırması of two bayagikesir numbers
return (sol.toDouble() <= sag.toDouble());
}

public boolean buyuktur(bayagikesir1 sol,bayagikesir1 sag)
{
// buyuktur then karşılaştırması of two bayagikesir numbers
return sol.toDouble()>sag.toDouble();
}

public boolean buyuktur_esittir(bayagikesir1 sol,bayagikesir1 sag)
{
// buyuktur then and esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() >= sag.toDouble();
}

public boolean esittir(bayagikesir1 sol,bayagikesir1 sag)
{
// esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() == sag.toDouble();
}

public boolean esit_degildir(bayagikesir1 sol,bayagikesir1 sag)
{
// not esittir karşılaştırması of two bayagikesir numbers
return sol.toDouble() != sag.toDouble();
}

```

```

public static String cikti(bayagikesir1 value)
{
String b="";
if(Math.abs(value.Payda())!=1)
{
b=b+" (" +value.Pay()+ " / "+value.Payda()+ " )";
}
else
{
b=b+value.Pay()+" ";
}
return b;
}

public String cikti()
{
// yazima hazır bayagikesir formda String değişkeni iletir.
String b="";
if(Math.abs(Payda())!=1)
{
b=b+" (" +pay+ " / "+Payda()+ " )";
}
else
{
b=b+Pay()+" ";
}
return b;
}
}

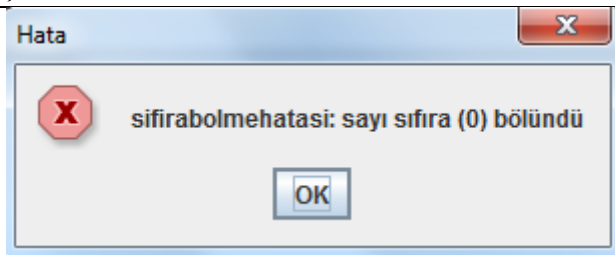
```

Problem 9.1.12 bayagikesir1test sınıfı

```

import javax.swing.JOptionPane;
public class bayagikesir1test
{
public static void main(String arg[])
{
try{
bayagikesir1 r1=new bayagikesir1(1,3);
bayagikesir1 r2=new bayagikesir1(1,0);
r2.topla(r1);
JOptionPane.showMessageDialog(null,r2.toString(),
"bayagikesir number class",JOptionPane.PLAIN_MESSAGE);
} catch(sifirabilmehatasi e1) {JOptionPane.showMessageDialog(null,e1.toString(),"Hata",JOptionPane.ERROR_MESSAGE);}
}
}

```



9.2 ALIŞTIRMALAR

1) Şirket kar-zarar bilanço durumunu veren sirket programı verilmiştir.

```

public class sirket
{ public String isim;
  public double sermaye,islem_hacmi,kar;
  public int islem;

  //constructor method
  public sirket(String is,double Ri)
  { isim=is;
    sermaye=Ri;
    islem_hacmi=Ri;
    kar=0;
    islem=0;
  }

  public void buy(double x)
  { islem_hacmi-=x;
    islem++;
    if(islem_hacmi<=0)
      {System.out.println("şirket iflas etmiş durumda "+islem);}
  }

  public void sell(double x)
  { islem_hacmi+=x;
    kar=islem_hacmi-sermaye;
    islem++;
  }

  public String account()
  { String s="-----"+isim+"-----\n";
    if(islem_hacmi<=0)
      s+="sirket is broken \n";
    else
      s+=" sermaye = "+sermaye+" TL\n";
      s+=" islem_hacmi = "+islem_hacmi+" TL\n";
      s+=" kar = "+kar+" TL\n";
      s+=" number of islemes = "+islem+" \n";
    return s;
  }
}

```

Şirket iflas etti hatası aşağıdaki gibi verilmiştir.

```

public class sirket_iflas_etti_hatasi extends Exception
{ public companyisbrokenException() { super("Şirket iflas etmiştir."); }
}

```

Bu sirket sınıfını değiştirerek sirket_iflas_etti_hatasi hata programı ile integre ediniz.

2) Aşağıdaki programlar verilmiştir.


```
public class minuslogarithmException extends ArithmeticException
{   public minuslogarithmException() { super("logarithm of a minus number "); }
}
```

```
import javax.swing.*; // program poweres class Scanner

public class logarithmatesti
{
    public static double log(double x) throws minuslogarithmException
    {
        // ln(x) = 1+y^2/3+y^4/5+y^6/7+...  y=(x-1)/(x+1)
        if(x<0) throw new minuslogarithmException();
        double power=1;
        double ln=1;
        double y=(x-1)/(x+1);
        double n=1;
        do
        {
            power*=y*y;
            ln+=power/(2.0*n+1.0);
            n++;
        } while(n<=100000);
        ln*=2*y;
        return ln;
    }
    // main method begins execution of Java application
    public static void main( String args[] )
    {
        String s;
        double x=Double.parseDouble(JOptionPane.showInputDialog("Enter a real number: ")); // read number
        s="log("+x+") = "+log(x)+" "+"Math library "+Math.log(x);
        JOptionPane.showMessageDialog(null,s,"natural logarithm calculation",JOptionPane.PLAIN_MESSAGE);
    } // end method main
} // end class
```

Bu programı değiştirerek log metoduna eksi bir rakam girildiğinde eksi rakam hatası verecek şekilde getiriniz.

9.3 PROBLEMLER

- 1) Bayağı kesir programında sıfıra bölme hatası verilmiştir. Programa ArithmeticException da ilave edilerek tam sayı olarak verilmemiş bayağıkesir pay ve paydası girildiğinde hata mesajı verecek hale getiriniz.
- 2) Altteki complex sınıfı verilmiştir.

```
public class complex
{   public double real;
    public double imaginary;

    public complex(double g,double s)
    {real=g;
     imaginary=s;
    }
}
```

```
public complex(complex s1)
{real=s1.real;
imaginary=s1.imaginary;
}

public void add(complex s1)
{real+=s1.real;
imaginary+=s1.imaginary;
}

public static complex add(complex s1,complex s2)
{complex z3=new complex((s1.real+s2.real),(s1.imaginary+s2.imaginary));
return z3;
}

public String toString()
{ String s="+real+" + i*"+imaginary+" \n";
return s;
}
}
```

Ana programda iki gerçekte sayı değişken okunması gerekmektedir. Eğer sayılar gerçekte sayı veya tam sayı olarak girilmediyse, örneğin String tipi değişken girildiyse hata mesajı verecek düzenlemeleri yapınız.

10 İLERİ VERİ YAPILARI

10.1 GENEL(GENERIC) METODLAR

Şu ana kadarki programlarımızda girdi ve çıktı değişkenlerinin türlerinin tanımlanması gereğinden bahsettik. Bu bize bazen değişik girdi ve çıktı değişkenleri kullandığımız için aynı metodun birden çok kopyesini oluşturmamız gibi bir gereksinme doğurur. Eğer girdi ve çıktı değişkenleri genel bir formda tanımlanabilseydi programlama bu tür uygulamalarda çok daha kısa bir şekilde yapılabilirdi. Burada Genel (Generic) değişken türü tanımlama ve kullanma üzerinde duracağız. İlk örneğimizde boyutlu değişik tür değişkenleri bütün olarak yazdırmak için for döngüsü kullanan print metodları oluşturduk.

Program 10.1.1 aynı isimli 4 print metodunun kullanıldığı yazdır sınıfı

```
public class yazdir
{
public static void print(Integer[] x)
{for(int i=0;i<x.length;i++)
System.out.print(" "+x[i]);
System.out.println();
}
public static void print(Double[] x)
{for(int i=0;i<x.length;i++)
System.out.print(" "+x[i]);
System.out.println();
}
public static void print(Character[] x)
{for(int i=0;i<x.length;i++)
System.out.print(" "+x[i]);
System.out.println();
}
public static void print(String[] x)
{for(int i=0;i<x.length;i++)
System.out.print(" "+x[i]);
System.out.println();
}
public static void main(String arg[])
{
Integer[] i={1,2,3,4,5,6};
Double[] d={1.1,2.2,3.3,4.4,5.5,6.6};
Character[] c={'a','b','c','d','f','g'};
String[] s={"ali","veli","49","elli"};
print(i);
print(d);
print(c);
print(s);
}
}
```

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" yazdir
1 2 3 4 5 6
```

```
1.1 2.2 3.3 4.4 5.5 6.6
```

```
a b c d f g
```

```
ali veli 49 elli
```

> Terminated with exit code 0.

Şimdi bu programı daha kısa hale getirmek için genel tür değişken (Generic) kullanacağız. Programda metod çıktı genel tür değişken tanımlamak için **<genel_değişken_ismi>** tanımını kullandık. Girdi değişkenimizin boyutlu olduğunu bildirmek için ise **değişken_ismi[]** tanımını kullandık.

Program 10.1.2 Genel (generic) olarak tanımlanmış print metodunun kullanıldığı yazdir1 sınıfı

```
public class yazdir1
{
    public static<genel> void printArray( genel[] x)
    { for(int i=0;i<x.length;i++)
      System.out.print(" "+x[i]);
      System.out.println();
    }

    public static void main(String arg[])
    {
      Integer[] i={1,2,3,4,5,6};
      Double[] d={1.1,2.2,3.3,4.4,5.5,6.6};
      Character[] c={'a','b','c','d','f','g'};
      String[] s={"ali","veli","49","elli"};
      printArray(i);
      printArray(d);
      printArray(c);
      printArray(s);
    }
}
```

İki programın sonuçları birbirinin aynıdır. Bu programı hazırlarken sınıf çıktı değişkeni tanımını **<genel>** tanımıyla değiştirildi. Sınıf girdi boyutlu değişkeni olarak ta **genel[]** kullanıldı. Burada değişkenlerimizin hepsinin nesne tipi değişkenler olarak tanımlandığına dikkatinizi çekelim. Basit değişken türleri için (double, int, char, boolean..) bu işlemi yapamayız. Nesne tipi değişkenlerin hepsi Object sınıfından türetilmiştir. Aslında genel tanımını bize Object sınıfı üzerinden değişken transferi yapmaktadır. Print metodunun tanımını Object sınıfı olarak yaparak ta aynı sonucu elde edebildik.

Program 10.1.3 Genel (generic) yapıdaki Object sınıfı kullanılarak tanımlanmış print metodunun kullanıldığı yazdir2 sınıfı

```
public class yazdir2
{
    public static void print( Object[] x)
    { for(int i=0;i<x.length;i++)
      System.out.print(" "+x[i]);
      System.out.println();
    }

    public static void main(String arg[])
    {
```

```
Integer[] i={1,2,3,4,5,6};
Double[] d={1.1,2.2,3.3,4.4,5.5,6.6};
Character[] c={'a','b','c','d','f','g'};
String[] s={"ali","veli","49","elli"};
print(i);
print(d);
print(c);
print(s);
}
}
```

10.2 GENEL(GENERIC) SINIFLAR

Genel metod kavramında gördüğümüz genelleştirme sınıflara da uygulanabilir. Sınıf değişkenleri genel değişken tanımıyla verilebilir.

Program 10.2.1 genel olarak (genel sınıf değişkeni ile) tanımlanmış matris sınıf

```
public class matris<E>
{
    public E[][] A;
    //constructor methods
    public matris(E[][] Ai)
    { try{
        A = (E[][]) new Object[Ai.length][Ai[0].length];
    }catch(bosmatrishatasi e){}
    catch(dolumatrishatasi f){};
    input_matris(Ai);
}

    public matris(matris< E > M)
    {
        try{
            A=(E[][])new Object[M.A.length][M.A[0].length];
        }catch(bosmatrishatasi e){}
        catch(dolumatrishatasi f){};
        input_matris(M.A);
    }

    public matris(int n,int m)
    { try{
        A=(E[][])new Object[n][m];
    }catch(bosmatrishatasi e){}
    catch(dolumatrishatasi f){};
    }

    public void input_matris(E[][] Ai)
    { for(int i=0;i<A.length;i++)
        { for(int j=0;j<A[0].length;j++)
            { A[i][j]=Ai[i][j];}
        }
    }

    public String toString(matris< E > M)
    { int n=M.A.length;
    int m=M.A[0].length;
    String s="";
    for(int i=0;i<n;i++)
```

```

        { for(int j=0;j<m;j++)
          { s+=M.A[i][j]+" ";
            s+="\n";
          }
        return s;
      }

      public String toString()
      {
        int n=A.length;
        int m=A[0].length;
        String s1="";
        for(int i=0;i<n;i++)
        { for(int j=0;j<m;j++)
          { s1+=String.format(" %s ",A[i][j]);}
          s1+="\n";
        }
        return s1;
      } //end of method toString

    } //end of class matrix

```

Program 10.2.2 bosmatrishatasi hata yakalama sınıfı

```

public class bosmatrishatasi extends RuntimeException
{public bosmatrishatasi()
 {super("Matris tamamen dolu");}
public bosmatrishatasi(String exception)
 {super(exception);}
}

```

Program 10.2.3 dolumatrishatasi hata yakalama sınıfı

```

public class dolumatrishatasi extends RuntimeException
{public dolumatrishatasi()
 {super("Matris tamamen dolu");}
public dolumatrishatasi(String exception)
 {super(exception);}
}

```

Program 10.2.4 Generic matrix test sınıfı matristesti

```

public class matristesti
{public static void main(String arg[])
{Double[][] a={{1.1,2.2},{3.3,4.4}};
Integer[][] b={{1,2},{3,4}};
Character[][] c={{'a','b'},{'c','d'}};
matrix<Double> A=new matrix<Double>(a);
matrix<Integer> B=new matrix<Integer>(b);
matrix<Character> C=new matrix<Character>(c);
System.out.println(A.toString());
System.out.println(B.toString());
System.out.println(C.toString());
}}

```

----- Capture Output -----

```
> "C:\turhan\java\bin\javaw.exe" matristesti
```

```
1.1 2.2
3.3 4.4
```

```
1 2
3 4
```

```
a b
c d
```

```
> Terminated with exit code 0.
```

Burada boyutlu deęişken tanımı aktarımının Object üzerinden yapıldığına dikkatiniz çekmek isteriz.

```
public E[][] A=(E[][])new Object[M.A.length][M.A[0].length];
```

örneğin

```
degerL=(JComboBox<String>)new Object[np1];
```

Genel sınıf tanımları ileri veri yapılarının temel aktarım sınıfları olarak karşımıza çıkar. Bu yüzden bu sınıfları kullanırken bu özelliklerini göz önünde bulundurmanız gerekir.

10.3 LAMDA DEYİMLERİNİ GENEL SINIF TANIMLARIYLA KULLANMA

Lamda deyimi kullanımına önceki bölümlerimizde değinmiştik. Lamda deyiminin javaya eklendiği java8 aynı zamanda çeşitli fonksiyon interface yapılarını da kütüphaneye eklemiştir. Tüm bu eklenen fonksiyonlar genel deęişken yapıları üzerinden tanımlandığı için rahatlıkla genel bir yapı olarak kullanılabilirler. Böylece bir çok işlemi yapmak için gerekli çeşitli interface'ler hazır olarak bulunabilir ve kullanılabilir. Bu interface'lerden bazıları alttaki tabloda verilmiş ve örnekler üzerinden lamda deęişkenleri olarak kullanılmalrı gösterilmiştir. Daha detaylı fonksiyon interface tanımları için java doküman kütüphanesini inceleyebilirsiniz.

Tablo 10.3.1 java 8 de tanımlanmış bazı fonksiyon interface tanımları

Interface ismi	Giriş deęişkenleri	Çıkış deęişkeni
Predicate<T>	T	boolean
Consumer<T>	T	void
Function<T,R>	T	R
BiFunction<T,W,R>	(T,W)	R
Supplier<T>	None	T
UnaryOperator<T>	T	T
BinaryOperator<T>	(T,T)	T
Binary Function<T,W,R>	(T,W)	R

Şimdi bu interface'lerin bazılarının lamda deyimleriyle birlikte kullanımlarını örneklerle göz atalım:

10.3.1 Predicate Fonksiyonunun lamda deyimiyle kullanımı

```
import java.util.function.*;
public class Predicate_L
{
    public static void main(String[] arg)
    {
        Predicate<Integer> atLeast5 = x -> x > 5;
        Boolean b=atLeast5.test(2);
        System.out.println(b);
    }
}
```

```
----- Capture Output -----  
> "C:\java\bin\java.exe" Predicate_L  
false  
> Terminated with exit code 0.
```

10.3.2 Consumer Fonksiyonunun lamda deyimiyle kullanımı

```
import java.util.function.*;  
public class Function_L  
{ public static void main(String[] arg)  
  { Function<Double,Double> y = x -> {return x*x-2.0*x+5;};  
    System.out.println(z.apply(2.0));  
  }  
}
```

```
----- Capture Output -----  
> "C:\java\bin\java.exe" Consumer_L  
-6.0  
> Terminated with exit code 0.
```

10.3.3 Function Fonksiyonunun lamda deyimiyle kullanımı

```
import java.util.function.*;  
public class Function_L  
{ public static void main(String[] arg)  
  { Function<Double,Double> y = x -> x*x-2.0*x+5;  
    System.out.println(y.apply(2.0));  
  }  
}}
```

```
----- Capture Output -----  
> "C:\java\bin\java.exe" Function_L  
5.0  
> Terminated with exit code 0.
```

10.3.4 Function Fonksiyonunun lamda deyimiyle kullanımı

```
import java.util.function.*;  
public class Function_L1  
{ public static void main(String[] arg)  
  { Function<Double,Boolean> y = x -> (x*x-2.0*x+5)>2;  
    System.out.println(y.apply(2.0));  
  }  
}}
```

```
----- Capture Output -----  
> "C:\java\bin\java.exe" Function_L1  
true  
> Terminated with exit code 0.
```

10.3.5 BiFunction Fonksiyonunun lamda deyimiyle kullanımı

```
import java.util.function.*;  
public class BiFunction_L  
{ public static void main(String[] arg)  
  { BiFunction<Double,Double,Boolean> z = (x,y) -> (x*x+y*y)>3;  
    System.out.println(z.apply(2.0,3.0));  
  }  
}
```



```
}}
```

```
----- Capture Output -----  
> "C:\java\bin\java.exe" BiFunction_L  
true  
> Terminated with exit code 0.
```

10.4 STRINGTOKENIZER SINIFI

Eğer bir cümle kelimelere bölünecekse, bu işi yapabilen bir sınıfımız hazır olarak java sınıf kütüphanelerinde mevcut bulunmaktadır. Bu sınıfın adı **StringTokenizer** sınıfıdır. StringTokenizer sınıfı bütün olarak verilmiş bir metni boşluk veya verilen diğer anahtar bir harf veya sembol kullanarak bir String boyutlu değişkeni haline dönüştürür (kelimelerine böler). String tokenizer sınıfının metodları şunlardır:

Kurucu metodlar:

[StringTokenizer\(String str\)](#) : kurucu metod str girdi String değişkeni

[StringTokenizer\(String str, String delim\)](#) : Constructs a string tokenizer for the specified string.

[StringTokenizer\(String str, String delim, boolean returnDelims\)](#) : Constructs a string tokenizer for the specified string.

Diğer metodlar:

int [countTokens\(\)](#) String Tokenizer sınıfının böldüğü toplam kelime sayısını verir.

Boolean [hasMoreElements\(\)](#) : eğer listede hala kelime varsa true(doğru) yoksa false(yanlış) değeri verir.

Boolean [hasMoreTokens\(\)](#) : eğer listede hala kelime varsa true(doğru) yoksa false(yanlış) değeri verir.

Object [nextElement\(\)](#) : Bir sonraki kelimeyi verir

String [nextToken\(\)](#) : Bir sonraki kelimeyi String tipi değişken olarak verir.

String [nextToken\(String delim\)](#) : ayraç olarak denim kullanarak bir sonraki kelimeyi String olarak verir.

Program 10.4.1 StringTokenizer test programı

```
import java.io.*;  
import java.util.*;  
  
public class StringTokenizerTest  
{  
    public static void main(String arg[])  
    {  
        String s="Evvel zaman içinde kalbur saman içinde ben dedemin beşiğini tıngır mıngır salları iken";  
        System.out.println("sentence : "+s);  
        StringTokenizer t=new StringTokenizer(s);  
        System.out.println("kelime sayısı : "+t.countTokens());  
        int i=0;  
        while(t.hasMoreTokens())  
        {System.out.println("kelime indeksi : "+(i++)+" kelime : "+t.nextToken());}  
    }  
}
```

```
----- Capture Output -----  
> "D:\java\bin\javaw.exe" StringTokenizerTest  
sentence : Evvel zaman içinde kalbur saman içinde ben dedemin beşiğini tıngır mıngır salları iken  
kelime sayısı : 13
```

```
kelime indeksi : 0 kelime : Evvel
kelime indeksi : 1 kelime : zaman
kelime indeksi : 2 kelime : içinde
kelime indeksi : 3 kelime : kalbur
kelime indeksi : 4 kelime : saman
kelime indeksi : 5 kelime : içinde
kelime indeksi : 6 kelime : ben
kelime indeksi : 7 kelime : dedemin
kelime indeksi : 8 kelime : beşğini
kelime indeksi : 9 kelime : tıngır
kelime indeksi : 10 kelime : mıngır
kelime indeksi : 11 kelime : sallar
kelime indeksi : 12 kelime : iken
```

```
> Terminated with exit code 0.
```

İlk örneğimizde String StringTokenizer sınıfına yüklenmiş ve

```
while(t.hasMoreTokens())
{System.out.println("kelime indeksi : "+(i++)+" kelime : "+t.nextToken());}
}
```

Komutuyla kelimelere bölünmüştür. İkinci örneğimizde double türü bir boyutlu değişken JOptionPane sınıfında girilen değerler kullanılarak ayrı ayrı değişkenler olarak boyutlu yapıya aktarılacaktır. Görüldüğü gibi bu yapı boyutları ve alt elemanları kendisi otomatik olarak değerlendirebildiği için çok önemli bir programlama alt yapısı oluşturmaktadır.

Program 10.4.2 StringTokenizerTest1 sınıfı

```
import java.io.*;
import java.util.*;
import javax.swing.*;

public class StringTokenizerTest1
{
    public static double[] boyutludoublegir(String s)
    {
        String s1=JOptionPane.showInputDialog(s);
        StringTokenizer token=new StringTokenizer(s1);
        int n=token.countTokens()-1;
        int m=n+1;
        double a[]=new double[m];
        int j=0;
        while(token.hasMoreTokens())
        {
            Double ax=new Double(token.nextToken());
            a[j++]=ax.doubleValue();
        }
        return a;
    }

    public static String print(double a[])
    {
        String s="";
        int n=a.length;
        int i=0;
        while(i<n) {s+=a[i]+"\\n";i++;}
        return s;
    }
}
```

```
}  
  
public static void main(String arg[])  
{  
double x[]=boyutludoublegir("boyutlu double deęişken deęerlerini her biri arasında bir boşluk bırakarak giriniz");  
JOptionPane.showMessageDialog(null,print(x),"Boyutlu double deęişken String Tokenizer  
Testi",JOptionPane.PLAIN_MESSAGE);  
}  
}
```





10.5 STRINGBUFFER SINIFI

StringBuffer sınıfı String sınıfının kardeř sınıfıdır. Bu sınıf kullanılarak String sınıfındaki deęişkenin char alt deęişkenlerine direk olarak ulaşmak mümkündür. Örnek problemimizde String Buffer sınıfı kullanılarak String ters çevrilecektir. String buffer sınıfı listesi :

[StringBuffer\(\)](#)

başlangıç kapasitesi olarak 16 karakter verilmiş boş bir sınıf oluşturur..

[StringBuffer\(String str\)](#)

str Stringinin karakter deęerlerini StringBuffer sınıfına yükler

StringBuffer sınıfının dięer metodları şunlardır:

Deęişken ekleme metodları

- [StringBuffer](#) [append](#)(boolean b)
- [StringBuffer](#) [append](#)(char c)
- [StringBuffer](#) [append](#)(char[] str)
- [StringBuffer](#) [append](#)(char[] str, int offset, int len) .
- [StringBuffer](#) [append](#)(CharSequence s)
- [StringBuffer](#) [append](#)(CharSequence s, int start, int end) .
- [StringBuffer](#) [append](#)(double d)
- [StringBuffer](#) [append](#)(float f)
- [StringBuffer](#) [append](#)(int i)
- [StringBuffer](#) [append](#)(long lng)
- [StringBuffer](#) [append](#)(Object obj) .
- [StringBuffer](#) [append](#)(String str)
- [StringBuffer](#) [append](#)(StringBuffer sb)
- [StringBuffer](#) [appendCodePoint](#)(int codePoint)

- İnt [capacity](#)()
şu andaki depolama kapasitesini verir
- Char [charAt](#)(int index)
index teki karakteri verir
- İnt [codePointAt](#)(int index)
indexteki karakterin Unicode karşılığını verir

İnt codePointBefore(int index)
 indeksten bir önceki karakteri verir

İnt codePointCount(int beginIndex, int endIndex)
 indeksten bir öncelki karakterin Unicode karşılığını verir

StringBuffer delete(int start, int end)
 yüklenen karakterleri siler

StringBuffer deleteCharAt(int index)
 index adresindeki karakteri siler

Void ensureCapacity(int minimumCapacity)
 depolama kapasitesini en az minimumCapacity değişkeninde verilen değere getirir.

Void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)
 belli iki indeks arasındaki alt karakter dizinini dst değişkenine yükler

İnt indexOf(String str)
 belli bir str alt gurubunun başlangıç adresini verir.

İnt indexOf(String str, int fromIndex)
 bir stringin belli bir indeksten başlayarak ilk defa oluştuğu yeri verir.
 Değişkenleri listeye ekleme metodları:

StringBuffer insert(int offset, boolean b)

StringBuffer insert(int offset, char c)

StringBuffer insert(int offset, char[] str) .

StringBuffer insert(int index, char[] str, int offset, int len)

StringBuffer insert(int dstOffset, CharSequence s)

StringBuffer insert(int dstOffset, CharSequence s, int start, int end)

StringBuffer insert(int offset, double d)

StringBuffer insert(int offset, float f)

StringBuffer insert(int offset, int i)

StringBuffer insert(int offset, long l)

StringBuffer insert(int offset, Object obj)

StringBuffer insert(int offset, String str)

İnt lastIndexOf(String str)
 verilen alt Stringin en son verildiği adresi verir.

İnt lastIndexOf(String str, int fromIndex)
 verilen alt Stringin en son verildiği adresi verir.

İnt length()
 Stringin kaç karakter boyunda olduğunu verir.

StringBuffer replace(int start, int end, String str)
 alt stringteki karakterleri yeni verilen string ile değiştirir.

StringBuffer reverse()
 Causes this character sequence to be replaced by the reverse of the sequence.

Void setCharAt(int index, char ch)
 The character at the specified index is set to ch.

Void setLength(int newLength)
 Sets the length of the character sequence.

CharSequence subSequence(int start, int end)
 Returns a new character sequence that is a subsequence of this sequence.

String substring(int start)
 Returns a new String that contains a subsequence of characters currently contained in this character

sequence.

String **substring(int start, int end)**
Returns a new String that contains a subsequence of characters currently contained in this sequence.

String **toString()**
Returns a string representing the data in this sequence.

Void **trimToSize()**
Attempts to reduce storage used for the character sequence.

Methods inherited from class java.lang.Object

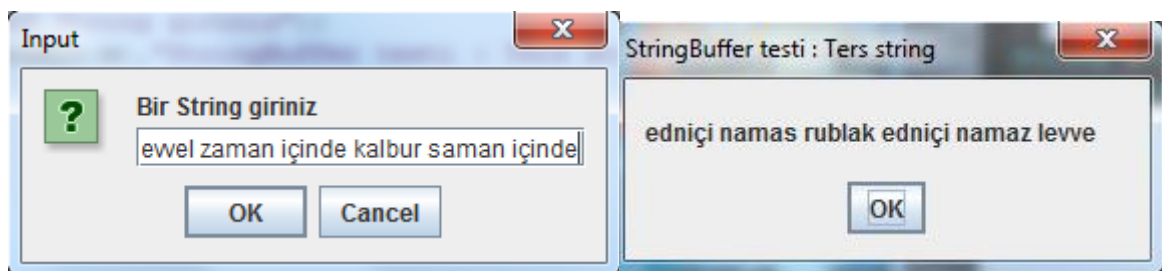
[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Program 10.5.1 StringBufferTest class test

```
import java.io.*;
import java.util.*;
import javax.swing.*;

public class StringBufferTest
{
    public static String reverseIt(String source) {
        int i, len = source.length();
        StringBuffer dest = new StringBuffer(len);
        for (i = (len - 1); i >= 0; i--) {
            dest.append(source.charAt(i));
        }
        return dest.toString();
    }

    public static void main(String arg[])
    {
        String s=JOptionPane.showInputDialog("Bir String giriniz");
        JOptionPane.showMessageDialog(null,reverseIt(s),"StringBuffer testi : Ters string",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Program 10.5.2 StringBuffer class test

```
import java.io.*;
class javakahvesi{
    public static void main (String args[])
    {
        StringBuffer sb = new StringBuffer("Java kahvesi içiniz!");
        sb.insert(12, "ni sıcak ");
        System.out.println(sb.toString());
    }
}
```

```
}  
}
```

```
----- Capture Output -----  
> "D:\java\bin\javaw.exe" javakahvesi  
Java kahvesini sıcak içiniz!  
> Terminated with exit code 0.
```

10.6 DİNAMİK BOYUTLU DEĞİŞKEN KULLANIMI : LIST, ARRAYLIST, VECTOR, LINKEDLIST,QUE SINIFLARI

Şu ana kadar birden fazla değişkeni bir arada kullanmak gerektiğinde boyutlu değişken yapısını (Arrays) kullandık. Boyutlu değişken yapısında yapının boyutu statik olarak birbiri ardına sırayla oluşturulur. Örneğin **double b[]=new double[3];**
Deyimi bilgisayar belleğinde

Tablo 10.6.1 boyutlu değişkenlerin bilgisayar belleğinde yerleşimi

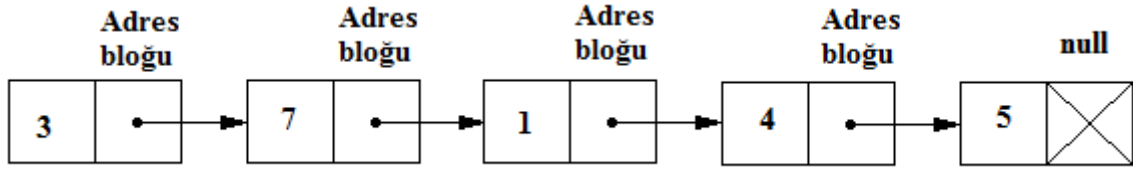
1023400	b[0]
1023464	b[1]
1023528	b[2]

yerlerini birbirini izleyen adreslerde ayırır.

Buradaki sorun eğer bilgisayar belleğindeki toplam yer değiştirilmek istenirse bunun zor olmasıdır. Aynı zamanda bilgisayar boyutlu değişkenin adreslerini bilgisayar hafızasında bir referans tablosu olarak tuttuğundan boyut değiştirme bilgisayar hafızasını da zorlayan bir işlemdir. Pratik olarak yapılan uygulama yeni bir boyutlu değişken oluşturup adresleri değiştirmektir.

```
double c[]=new double[5];  
for(int i=0,i<b.length;i++)  
{ c[i]=b[i];}  
b=c;
```

Statik yapıdaki boyutlu değişken yapıları yerine dinamik yapıda boyutlu değişken yapıları oluşturan sınıflar oluşturabiliriz. Dinamik yapılarda toplam boyut sayısı program çalışırken değiştirilebilir. Boyutlu dinamik elemanlar bir göze (kaynak) elemandan başlayarak oluşturulur. Her göze veri değeri ve yeni bir göze bağlantısı için adres değeri içerir. Bu adres üzerinden göze elemanı bir sonraki gözeye bağlanır. Serinin en sonunda adres olarak null (boş) değeri taşıyan bir göze bulunur ve böylece seri sona erdirilir. Eğer yeni bir eleman eklenmek istenirse null değeri silinerek yeni eklenecek elemanın adresiyle değiştirilir. Bu tür elemanlara ilk örnek olarak List sınıfını vereceğiz. List sınıfı java dilinde kütüphane programı olarak verilmektedir. Göze elemanlarının bağlantısını daha iyi anlatabilmek için burada bir örnek program verilmiştir.



Program 10.6.1Kendi tanımladığımız dinamik boyutlu değişken ListGoze and List sınıfları

```

class ListGoze
{
  Object veri;
  ListGoze sonraki;
  ListGoze(Object o) {this(o,null);}
  ListGoze(Object o,ListGoze sonrakiGoze)
  {
    veri=o;
    sonraki=sonrakiGoze;
  }
  Object ObjectOku() {return veri;}
  ListGoze sonrakiGoze() {return sonraki;}
}

public class List
{
  //List
  private ListGoze ilkGoze;
  private ListGoze sonGoze;
  private String isim;
  public List(String s)
  {
    isim=s;
    ilkGoze=sonGoze=null;
  }
  public List(){this("Liste ");}
  public synchronized void ListeninOnuneEkle(Object yenidegergir)
  {
    if( bosmu() )
      ilkGoze = sonGoze = new ListGoze(yenidegergir);
    else
      ilkGoze = new ListGoze(yenidegergir,ilkGoze);
  }
  public synchronized void ListeninSonunaEkle(Object yenidegergir)
  {
    if(bosmu())
      ilkGoze=sonGoze=new ListGoze(yenidegergir);
    else
      sonGoze=sonGoze.sonraki=new ListGoze(yenidegergir);
  }
  public synchronized Object ListeninOnundenCikar()
  throws bosListhatasi
  {
    Object cikarilanNesne=null;
    if(bosmu())
      throw new bosListhatasi(isim);
    cikarilanNesne=ilkGoze.veri;
    if(ilkGoze.equals(sonGoze))
      ilkGoze=sonGoze=null;
  }
}
  
```



```

else
    ilkGoze=ilkGoze.sonraki;
return cikarilanNesne;
}
public synchronized Object listeninSonundanCikar() throws bosListhatasi
{
Object cikarilanNesne=null;
if(bosmu())
    throw new bosListhatasi(isim);
cikarilanNesne=sonGoze.veri;
if(ilkGoze.equals(sonGoze))
    ilkGoze=sonGoze=null;
else
{
    ListGoze secilenGoze=ilkGoze;
    while(secilenGoze.sonraki != sonGoze)
        secilenGoze=secilenGoze.sonraki;
    sonGoze=secilenGoze;
    secilenGoze.sonraki=null;
}
return cikarilanNesne;
}
public boolean bosmu() {return ilkGoze==null; }

public String cikti() throws bosListhatasi
{ String s="";
if(bosmu())
    { throw new bosListhatasi(isim); }
s+=" "+isim+" : ";
ListGoze secilenGoze=ilkGoze;
while(secilenGoze!=null)
    {
s+=secilenGoze.veri.toString()+" ";
secilenGoze=secilenGoze.sonraki;
    }
s+="\n";
return s;
}
}

```

Program 10.6.2 List sınıfı hata tanım sınıfı bosListHatasi

```

public class bosListhatasi extends RuntimeException
{
    public bosListhatasi(String isim)
    { super(" "+isim+" Liste boş "); }
}

```

Program 10.6.3 List sınıfı test programı

```

import javax.swing.*.*;
public class ListTest
{
    public static void main(String args[])
    {
        List liste=new List();
        Boolean b=new Boolean(true);
        Character c=new Character('\u03B2');
        Integer i=new Integer(34567);
        String s1=new String("merhaba");
    }
}

```

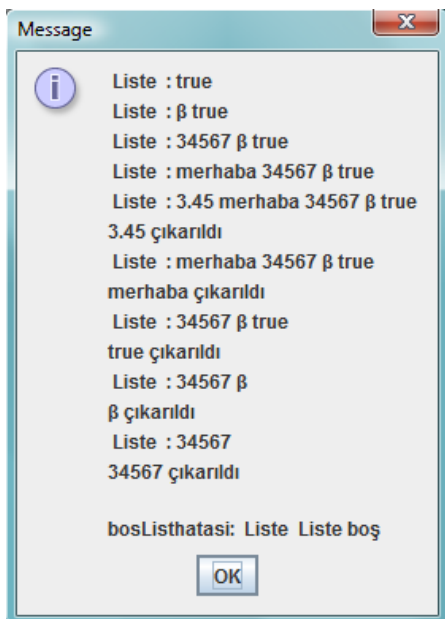
```

Double d=new Double(3.45);
String s="";

try{
    liste.ListeninOnuneEkle(b);
    s+=liste.cikti();
    liste.ListeninOnuneEkle(c);
    s+=liste.cikti();
    liste.ListeninOnuneEkle(i);
    s+=liste.cikti();
    liste.ListeninOnuneEkle(s1);
    s+=liste.cikti();
    liste.ListeninOnuneEkle(d);
    s+=liste.cikti();
    Object cikarilanNesne;
    cikarilanNesne=liste.ListeninOnundenCikar() ;
    s+=cikarilanNesne.toString()+" çıkarıldı \n";
    s+=liste.cikti();
    cikarilanNesne=liste.ListeninOnundenCikar() ;
    s+=cikarilanNesne.toString()+" çıkarıldı \n";
    s+=liste.cikti();
    cikarilanNesne=liste.listeninSonundanCikar();
    s+=cikarilanNesne.toString()+" çıkarıldı \n";
    s+=liste.cikti();
    cikarilanNesne=liste.listeninSonundanCikar();
    s+=cikarilanNesne.toString()+" çıkarıldı \n";
    s+=liste.cikti();
    cikarilanNesne=liste.listeninSonundanCikar();
    s+=cikarilanNesne.toString()+" çıkarıldı \n";
    s+=liste.cikti();
}
catch(bosListhasasi e) {s+="\n"+e.toString();}
OptionPane.showMessageDialog(null,s);
}
}

```

Program çıktısı:



Program örneğinden de görüleceği gibi listeye yeni elemanlar ekleyip çıkarmak çok kolaydır. Şimdi de aynı tür yapının Java Kütüphanesinde verilen sınıflarına göz atalım örnek programda verilen List sınıfının benzer yapılarını içeren ArrayList ve LinkedList sınıfları javada tanımlanmıştır, Bu yüzden de bu gaye için kendi tanımladığımız sınıfları kullanmamız gerekmez. İlk örneğimiz ArrayList sınıfından olacak. Burada Java sınıfında tanımlanan Dinamik yapıların girdilerinin Generic sınıftan olduğunu bildirelim.

Program 10.6.4 Java kütüphane sınıfı ArrayListList test programı

```
import java.util.*;
import javax.swing.*;

public class ArrayList1
{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static void main(String args[])
    {
        ArrayList<Integer> a=new ArrayList<Integer>();
        for(int i=0;i<5;i++)
        {int sayi=((int)(Math.random()*100));
          a.add(new Integer(sayi));
        }
        String s1="Array List sınıfı";
        String s=cikti(a);
        s+="sıraya dizilmiş liste\n";
        Collections.sort(a);
        s+=cikti(a);
        s+="ters liste\n";
        Collections.reverse(a);
        s+=cikti(a);
        s+="karıştırılmış liste\n";
        Collections.shuffle(a);
        s+=cikti(a);
        JOptionPane.showMessageDialog(null,s,s1,
        JOptionPane.PLAIN_MESSAGE);
    }
}
```

Programdaki ArrayList kurucu metodu **ArrayList<Integer> a=new ArrayList<Integer>();** **Generic olarak <Integer>** java Object gurubu değişkenini yarattı. ArrayList ile beraber herhangi bir nesnetemelli değişken türü kullanılabilir, ancak nesne temelli olmayan temel değişken türleri kullanılamaz(double, int, char..). Örnek olarak eğer String tipi bir değişken listesi tanımlamak istersek: **ArrayList<String> a=new ArrayList<String>();** Tanımını kullanırız. Veriyi geri okumak için **Iterator** sınıfı tanımlanmıştır. Döngü oluşturmak için bu sınıftan hasNext ve next metodlarını kullanıyoruz. (bakınız **public static String cikti(Collection c)** metodu). Bu kullanımın en büyük avantajı liste boşaldığında hasNext boolean tip metod yardımıyla döngünün otomatik durmasını sağlanmasıdır.



Bir sonraki programımızda döngüyü ArrayListsınıfının alt metodları olan get ve size metodlarını kullanarak oluşturduk. Sonuç benzer olacaktır.

Program 10.6.5 Java kütüphane programı ArrayList test programı ArrayList2

```
import java.util.*;
import javax.swing.*;

public class ArrayList2
{
    public static void main(String args[])
    {
        ArrayList<Integer> a=new ArrayList<Integer>();
        for(int i=0;i<5;i++)
        {int sayi=((int)(Math.random()*100));
        a.add(new Integer(sayi));
        }
        String s1="Array List class";
        String s="";
        for (int i=0, n=a.size(); i < n; i++) {s+=a.get(i)+"\n"};
        JOptionPane.showMessageDialog(null,s,s1,
        JOptionPane.PLAIN_MESSAGE);
    }
}
```

Bir sonraki örneğimizde kesişim ve bileşim kümelerini String tipi girdi verileri üzerinden değerlendirebilen bir test programı oluşturacağız.

Program 10.6.6 kumeA.java ve kumeTest sınıfları (ArrayList örneği)

```
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class kumeA
{
    public ArrayList<String> bilesim(ArrayList<String> list1, ArrayList<String> list2)
    {
        ArrayList<String> bilesimArray = new ArrayList<String>();
        String s1,s2;
        Iterator n1=list1.iterator();
        Iterator n2=list2.iterator();
        while(n1.hasNext())
        { s1=(String)n1.next();
          bilesimArray.add(s1);
        }
        while(n2.hasNext())
        {
            s2=(String)n2.next();
            if(!bilesimArray.contains(s2))
                bilesimArray.add(s2);
        }
        return bilesimArray;
    }

    public ArrayList<String> kesisim(ArrayList<String> list1, ArrayList<String> list2)
    {
        ArrayList<String> kesisimArray = new ArrayList<String>();
        String s;
        Iterator n = list1.iterator();
        while(n.hasNext())
        {
            s=(String)n.next();
            if(list2.contains(s))
                kesisimArray.add(s);
        }
        return kesisimArray;
    }
}

public class kumeTest extends JFrame implements ActionListener
{
    kumeA kume = new kumeA(); // kumeA
    JLabel L1, P1, P2;
    JTextField T1,T2;
    JTextArea output;

    // dinamik boyutlu degisken sinifi ArrayList kullanarak
    // iki vector listesi yarat
    ArrayList<String> list1 = new ArrayList<String>();
    ArrayList<String>list2 = new ArrayList<String>();

    // Sonuclari yine vektor cinsinden degiskenler
    // kullanarak aktar
    ArrayList<String> ansUn;
```

```

ArrayList<String> ansInt;
Container c;

public kumeTest()
{
    super("kume testi ");
    c=getContentPane();
    c.setLayout(new FlowLayout());
    P1 = new JLabel("İlk listeyi gir : ");
    T1 = new JTextField(30);
    T1.setBackground(c.getBackground());
    P2 = new JLabel("İkinci listeyi gir : ");
    T2 = new JTextField(30);
    T2.setBackground(c.getBackground());
    output = new JTextArea();
    output.setEditable(false);
    output.setText("");
    output.setBackground(c.getBackground());
    c.add(P1);
    c.add(T1);
    T1.addActionListener(this);
    c.add(P2);
    c.add(T2);
    T2.addActionListener(this);
    JScrollPane jp=new JScrollPane(output);
    jp.setPreferredSize(new Dimension(320,200));
    c.add(jp);
}

public void inputA(String string, ArrayList<String> list)
{
    StringTokenizer tokens = new StringTokenizer(string);
    while(tokens.hasMoreTokens()) {
        String test = tokens.nextToken();
        list.add(test);
    }
}

public void outputA(ArrayList<String> v)
{
    Iterator enm = v.iterator();
    while(enm.hasNext())
    {
        String ans = (String)enm.next();
        output.append(ans + "\n");
    }
}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==T1)
    {
        String stringToTokenize = T1.getText();
        inputA(stringToTokenize,list1);
    }
    else if(e.getSource()==T2)
    {
        String stringToTokenize = T2.getText();

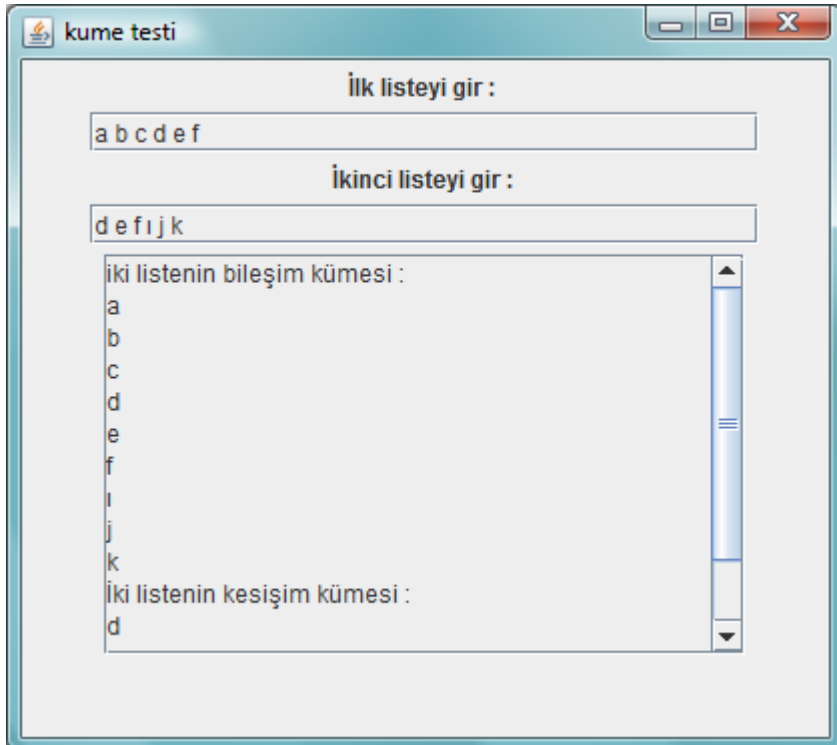
```

```

inputA(stringToTokenize,list2);
ansUn = kume.bilesim(list1,list2);
ansInt = kume.kesisim(list1,list2);
output.setText("");
output.append("iki listenin bileşim kümesi : \n");
outputA(ansUn);
output.append("İki listenin kesişim kümesi : \n");
outputA(ansInt);
}
} //end of actionPerformed method
public static void main(String[] args)
{
kumeTest pencere = new kumeTest();
pencere.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
pencere.setSize(400,350);
pencere.setVisible(true);
}
}

```

The program output:

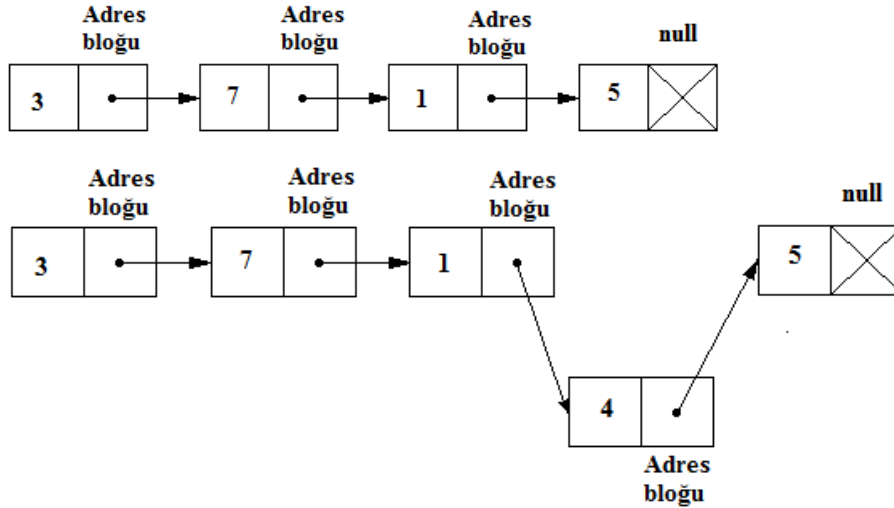


ArrayList sınıfının alt metodları aşağıda listelenmiştir.

Sonuç	Metod	Tanım
<i>Kurucu Metodlar</i>		
$a =$	new ArrayList<E>()	on doldurma kapasiteli boş bir liste hazırlar
$a =$	new ArrayList<E>(ilkkapasite)	verilen ilk kapasitede bir boş liste oluşturur
$a =$	new ArrayList<E>(coll<E>)	Collection sınıfında verilen elementlerden bir liste oluşturur.
<i>Eleman ekleme</i>		
	a.add(e)	verilen elemanı listenin sonuna yerleştirir.

	<code>a.add(i, e)</code>	verilen elemanı verilen indeksteki yere yerleştirir.
Elemanı değiştirme		
	<code>a.set(i,e)</code>	verilen elemanı verilen indeksteki elemanla değiştirir.
Elemana ulaşma		
<code>e =</code>	<code>a.get(i)</code>	I indeksindeki elemanın değerini verir.
<code>oarray =</code>	<code>a.toArray()</code>	Tüm elemanların değerini boyutlu değişkene aktarır
<code>earray =</code>	<code>a.toArray(E[])</code>	Tüm elemanların değerini E tipi boyutlu değişkene aktarır
İteratörler (döngü yapıları)		
<code>iter =</code>	<code>a.iterator()</code>	Döngü için iterator değeri verir
<code>liter =</code>	<code>a.listIterator(i)</code>	Döngü için ileriye veya geriye doğru sıralabilen ListIterator verir. İndeks I den başlayarak ileriye doğru, a.listIterator(a.size()) başlayarak geriye doğru ilerleyebilir.
<code>liter =</code>	<code>a.listIterator()</code>	Döngü için ileriye veya geriye doğru sıralabilen ListIterator verir.
Arama		
<code>b =</code>	<code>a.contains(e)</code>	Eğer ArrayList a, e değerini içeriyorsa true verir.
<code>i =</code>	<code>a.indexOf(e)</code>	E değerinin ilk geçtiği indeksi verir, eğer iç geçmiyorsa -1 verir.
<code>i =</code>	<code>a.lastIndexOf(e)</code>	Returns index of last occurrence of e, or -1 if not there.
Elemanı silme		
	<code>a.clear()</code>	ArrayList a'dan tüm elemanlarını siler
	<code>a.remove(i)</code>	I indeksindeki elemanı siler
	<code>a.removeRange(i,j)</code>	I den j ye kadar olan elemanları siler
Diğer		
<code>i =</code>	<code>a.size()</code>	Listedeki eleman sayısını verir.

Dinamik yapıda **a.add(i, e)** metodu kullanarak istenilen yere yeni bir liste elemanı yerleştirilebilir.



Şekil 10.5.1 ArrayList sınıfında Dinamik listeye yeni eleman ekleme prosesi

Vector sınıfı ArrayList sınıfının hemen hemen aynısıdır. Bu iki sınıfında kütüphanede bulunması daha ziyade tarihi sebeplere bağlıdır. Vector sınıfı Java dilinin ilk oluşturulduğunda eklenmiş bir sınıftır. ArrayList ise daha sonradan oluşturulmuştur.

Program 10.5.7 Java kütüphanesi Vector sınıfı test programı

```
import java.util.*;
import javax.swing.*;
```



```

public class Vector1
{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static void main(String args[])
    {

        Vector<Integer> a=new Vector<Integer>();
        for(int i=0;i<5;i++)
        {int sayi=((int)(Math.random()*100));
          a.add(new Integer(sayi));
        }
        String s1="Vector sınıfı";
        String s=cikti(a);
        s+="sıraya koyulmuş liste \n";
        Collections.sort(a);
        s+=cikti(a);
        JOptionPane.showMessageDialog(null,s,s1,
        JOptionPane.PLAIN_MESSAGE);
    }
}

```

Vector sınıfı için ikinci bir örnek:

Program 10.6.8 Vector sınıfı örneği gezegenler1

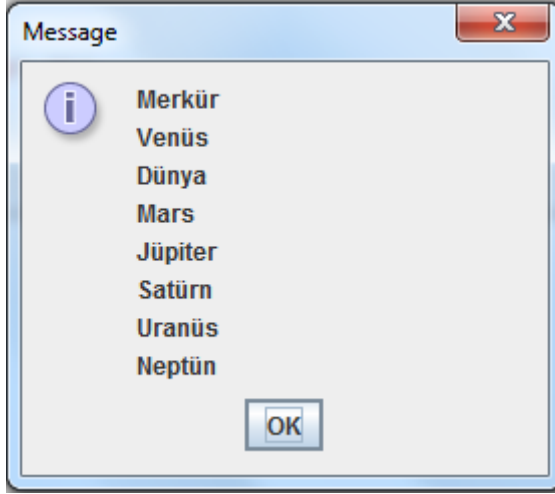
```

import java.util.*;
import javax.swing.JOptionPane;

public class gezegenler1 {
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static void main (String args[]) {
        String names[] = {"Merkür", "Venüs", "Dünya", "Mars", "Jüpiter", "Satürn", "Uranüs", "Neptün", "Pluton"};
        Vector<String> gezegenler = new Vector<String> ();
        for(int i=0;i<names.length;i++)
        { gezegenler.add(names[i]);}
        //Pluton artık gezegen sayılmıyordu değil mi?
        gezegenler.remove("Pluton");
        String ss=cikti(gezegenler);
        JOptionPane.showMessageDialog(null,ss);
    }
}

```



Vector sınıfının önemli metodları aşağıda verilmiştir.

<i>Metod</i>	<i>Tanım</i>
v.add(o)	O nesnesini(Object) v Vector nesnesine ekler
v.add(i, o)	O nesnesini(Object) v Vector nesnesine I inci eleman olarak ekler
v.clear()	V Vector nesnesindeki tüm elemanları siler
v.contains(o)	Eğer vector nesnesi v o Object nesnesini içeriyorsa true değeri verir.
v.firstElement(i)	Ilk elemanı verir
v.get(i)	I index elemanını verir
v.lastElement(i)	Son elemanı verir
v.listIterator()	Vector nesnesi v ile kullanılacak ListUterator sınıfı nesne oluşturur.
v.remove(i)	i' inci elemanı siler
v.set(i,o)	O nesnesini(Object) v Vector nesnesine I inci elemanı ile değiştirir
v.size()	V Vector nesnesinin eleman sayısını verir.
v.toArray(Object[])	Object türü boyutlu değişken oluşturur ve v Nesnesindeki elemanları yükleyerek bu boyutlu değişken türünden çıktı verir.

Vector javada ilk tanımlanmış dinamik boyutlu değişken tanımlayan sınıftır. İlk tanımlardan bazıları daha yeni ve diğer boyutlu dinamik değişken sınıflarıyla uyumlu metodlar haline getirilmiştir. Eski tanımlar hala çalışmakla birlikte yeni program yazarlar için tavsiye edilmez.

Eski metod	Yeni Metod
void addElement (Object)	boolean add (Object)
void copyInto (Object[])	Object[] toArray ()
Object elementAt (int)	Object get (int)
Enumeration elements ()	Iterator iterator () ListIterator listIterator()
void insertElementAt (Object, int)	void add (index, Object)
void removeAllElements ()	void clear ()
boolean removeElement (Object)	boolean remove (Object)
void removeElementAt (int)	void remove (int)
void setElementAt (int)	Object set (int, Object)

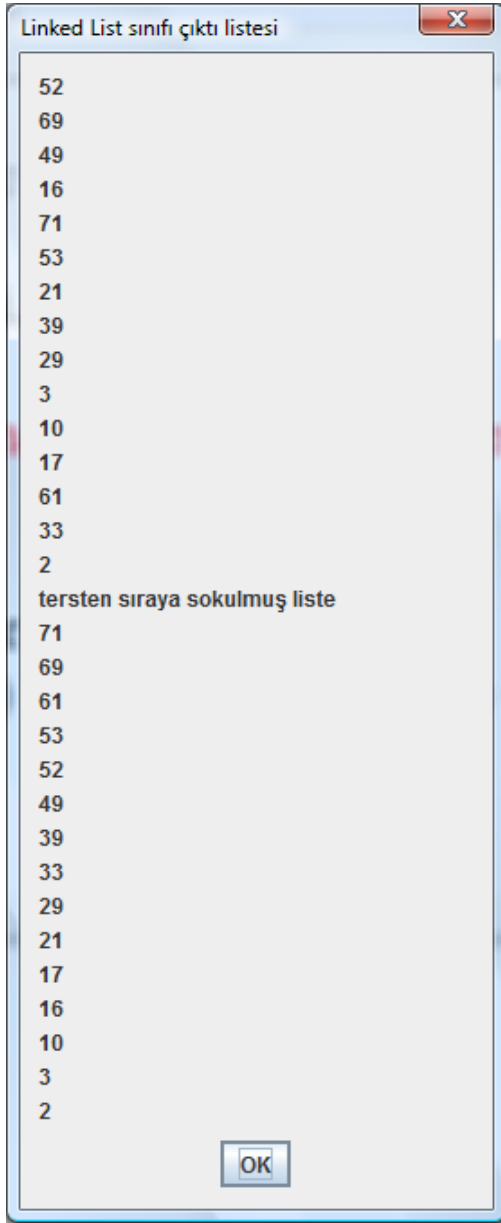
LinkedList sınıfının kullanımı da ArrayList ve Vector sınırlarının kullanımına benzer. En önemli değişikliği ArrayList ve Vector sınıfında yeni eleman dinamik boyutlu değişkenin herhangi bir noktasına eklenebiliyorken, LinkedList'te sadece iki ucuna eklenebilir.

Program 10.6.9 LinkedList inamik boyutlu yapı örnek programı

```
import java.util.*;
import javax.swing.*;

public class LinkedList1
{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static void main(String args[])
    {
        LinkedList<Integer> a=new LinkedList<Integer>();
        for(int i=0;i<15;i++)
        {int sayi=((int)(Math.random()*100));
        a.add(new Integer(sayi));
        }
        String s1="LinkedList sınıfı çıktı listesi";
        String s=cikti(a);
        s+="tersten sıraya sokulmuş liste\n";
        Collections.sort(a,Collections.reverseOrder());
        s+=cikti(a);
        JOptionPane.showMessageDialog(null,s,s1,
        JOptionPane.PLAIN_MESSAGE);
    }
}
```



Aşağıdaki tabloda LinkedList ile ilgili metodlar verilmektedir.

Sonuç	Metod	Tanım
Kurucu Metodlar		
$a =$	new ArrayList<E>()	on doldurma kapasiteli boş bir liste hazırlar
$a =$	new ArrayList<E>(ilkkapasite)	verilen ilk kapasitede bir boş liste oluşturur
$a =$	new ArrayList<E>(coll<E>)	Collection sınıfında verilen elementlerden bir liste oluşturur.
Eleman ekleme		
	a.add(e)	verilen elemanı listenin sonuna yerleştirir.
	a.add(i, e)	verilen elemanı verilen indeksteki yere yerleştirir.
	a.addFirst(e)	verilen elemanı listenin sonuna yerleştirir.
	a.addLast(e)	verilen elemanı listenin sonuna yerleştirir.
Elemanı değiştirme		
	a.set(i,e)	verilen elemanı verilen indeksteki elemanla değiştirir.
Elemana ulaşma		

$e =$	a.get(i)	I indeksindeki elemanın değerini verir.
	a.element()	Değeri çağırır fakat listeden silmez.
	a.getFirst()	İlk elemanı çağırır
	a.getLast()	Son elemanı çağırır
oarray =	a.toArray()	Tüm elemanların değerini boyutlu değişkene aktarır
earray =	a.toArray(E[])	Tüm elemanların değerini E tipi boyutlu değişkene aktarır
Iteratörler (döngü yapıları)		
iter =	a.iterator()	Döngü için iteratör değeri verir
liter =	a.listIterator(i)	Döngü için ileriye veya geriye doğru sınırlanabilir ListIterator verir. İndeks I den başlayarak ileriye doğru, a.listIterator(a.size()) başlayarak geriye doğru ilerleyebilir.
liter =	a.listIterator()	Döngü için ileriye veya geriye doğru sınırlanabilir ListIterator verir.
Arama		
b =	a.contains(e)	Eğer ArrayList a, e değerini içeriyorsa true verir.
i =	a.indexOf(e)	E değerinin ilk geçtiği indeksi verir, eğer iç geçmiyorsa -1 verir.
i =	a.lastIndexOf(e)	Returns index of last occurrence of e, or -1 if not there.
Elemanı silme		
	a.clear()	ArrayList a'dan tüm elemanlarını siler
	a.remove(i)	I indeksindeki elemanı siler
	a.removeRange(i,j)	I den j ye kadar olan elemanları siler
Diğer		
i =	a.size()	Listedeki eleman sayısını verir.

Türkçe sıra diyebileceğimiz ArrayDeque dinamik boyutlu değişkeni, bir önceki sınıfla aynıdır. Normal olarak ArrayDeque sınıfına her iki taraftan elemanlar eklenip çıkarılabilirse de normal pratik yeni elemanın sıranın sonuna eklenip başından çekilmesi şeklindedir. Aşağıda bir örnek problem verilmiştir.

Program 10.6.10 Gezegener2.java ,ArrayDeque (sıra) sınıfı örneği

```
import java.util.*;
import javax.swing.JOptionPane;

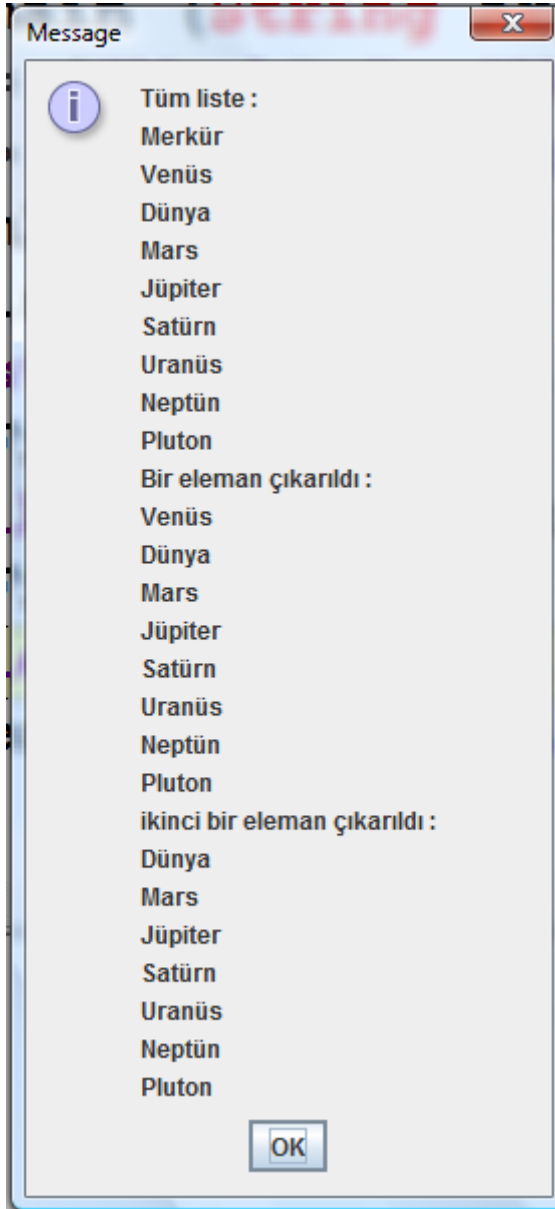
public class Gezegener2{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static void main (String args[]) {
        String isimler[] = {"Merkür", "Venüs", "Dünya","Mars", "Jüpiter", "Satürn", "Uranüs","Neptün",
        "Pluton"};
        ArrayDeque<String> Gezegener = new ArrayDeque<String> ();
        for(int i=0;i<isimler.length;i++)
        { Gezegener.add(isimler[i]);}
        String ss="Tüm liste : \n"+cikti(Gezegener);
        Gezegener.removeFirst();
    }
}
```

```

ss+="Bir eleman çıkarıldı :\n"+cikti(Gezegenler);
Gezegenler.removeFirst();
ss+="ikinci bir eleman çıkarıldı :\n"+cikti(Gezegenler);
JOptionPane.showMessageDialog(null,ss);
}
}

```



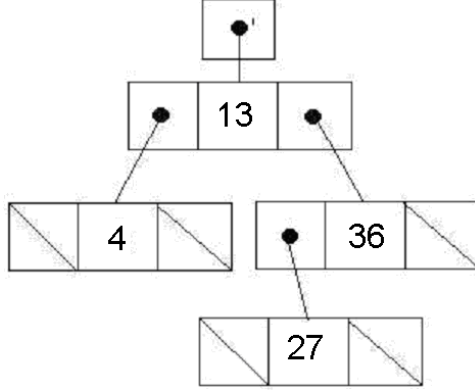
ArrayDeque metodları aşağıdaki tabloda verilmiştir:

Sonuç	Metod	Tanım
<i>Kurucu metodlar</i>		
$a =$	new ArrayDeque<E>(0)	Önceden tanımlanmış doldurma kapasiteli ArrayDeque yaratır
$a =$	new ArrayDeque<E>(ilkkapasite)	verilen ilk kapasitede bir boş liste oluşturur
$a =$	new ArrayDeque<E>(coll<E>)	Collection sınıfında verilen elementlerden bir liste oluşturur.
<i>Ön kısım için verilen işlemler (head, start, first)</i>		

	a.addFront(e)	e nesnesini a ArrayDeque sınıfı dinamik boyutlu nesnesinin başına ekler
b =	a.offerFirst(e)	addFront'la aynı , fakat sınırlı kapasiteli
e =	a.getFirst()	Listedeki ilk elementi verir, fakat listeden silmez.
e =	a.element()	Listedeki ilk elementi verir, fakat listeden silmez.
e =	a.peekFirst()	getFirst'le aynı fakat liste boşsa null değeri verir
e =	a.peek()	peekFirst'le aynı
e =	a.removeFirst()	İlk elemanın değerini verir ve değeri siler, eğer sıra boşsa hata mesajı (exception) yollar.
e =	a.remove()	pollFirst ile aynıdır
e =	a.pollFirst()	İlk elemanın değerini verir ve değeri siler, eğer sıra boşsa false değeri verir
e =	a.poll()	pollFirst ile aynıdır
e =	a.pop()	removeFirst ile aynıdır
e =	a.push(e)	addFront ile aynıdır.
<i>Son kısım (kuyruk) için verilen işlemler</i>		
	a.addLast(e)	e nesnesini a ArrayDeque sınıfı dinamik boyutlu nesnesinin sonuna(kuyruğuna) ekler
	a.add(e)	addLast ile aynıdır.
b =	a.offerLast(e)	addLast'la aynı , fakat sınırlı kapasiteli
b =	a.offer(e)	offerLast ile aynı
e =	a.getLast()	Son elemanı return yapar fakat listeden silmez. Sıra boşsa hata mesajı(exception) verir.
e =	a.peekLast()	Son elemanı return yapar fakat listeden silmez. Sıra boşsa null verir.
e =	a.removeLast()	Son elemanı return eder ve siler. Sıra boşsa hata mesajı(exception) verir.
e =	a.pollLast()	Son elemanı return eder ve siler. Sıra boşsa null verir.
<i>Boyutlu static değişkene dönüştürme</i>		
oarray =	a.toArray()	ArrayDeque içindeki elemanları boyutlu static değişkenlere dönüştürür
earray =	a.toArray(E[])	ArrayDeque içindeki elemanları E sınıftan boyutlu static değişkenlere dönüştürür
<i>Döngü yapıları, Iterators</i>		
iter =	a.iterator()	İleriye doğru döngü oluşturur.
iter =	a.descendingIterator()	Geriye doğru döngü oluşturur.
<i>Arama</i>		
b =	a.contains(e)	Eğer ArrayDeque dinamik sınıfı nesnesi a, e nesnesini barındırıyorsa true değeri döner
b =	a.removeFirstOccurrence(e)	E nesnesinin ilk değerini siler, ve silebilirse true Boolean değeri döner
b =	a.removeLastOccurrence(e)	E nesnesinin sıradaki en son değerini siler, ve silebilirse true Boolean değeri döner
b =	a.remove(e)	removeFirstOccurrence() ile aynıdır.
<i>Elemanları silme</i>		
<i>diğer</i>		
i =	a.size()	ArrayDeque a'nın boyutu.
	a.clear()	ArrayDeque içindeki tüm elemanları siler.

10.7 DİNAMİK HAFIZAYLA TANIMLANAN BOYUTLU SINIFLAR : TREE (AĞAÇ YAPISI)

Tree (ağaç) dinamik boyutlu yapılarının şu ana kadar baktığımız dinamik boyutlu yapılardan farkı her elemanın veri haricinde iki referans verisi içermesidir. İlk veri girildikten sonra, ikinci veri girildiğinde belli bir kurala göre yerleştirilir, örneğin daha küçük değerdeki veriler sol taraftaki referans noktasına daha büyük veriler sağ taraftaki referans noktasına adreslenerek eklenir. Bu şekilde depolanan bir dinamik değişkenden adresbel her çağrıldığında otomatik olarak sıraya dizilmiş olacaklardır. Bu tür bir ağaç yapısı şekil 14.6.1 de görülmektedir.



Şekil 14.6.1 içerdiği sayıların büyüklüğüne göre guruplandırılmış bir ağaç yapısı

Ağaç yapılarını daha iyi göz önüne getirebilmek için burada önce kendi ağaç yapısı sınıfını geliştirdik.

Program 10.7.1 TreeGoze ve Tree sınıfları (Ağaç yapısını daha iyi anlamak için kendi yazdığımız sınıf)

```

public class Tree
{
    private TreeGoze kok;
    public Tree() {kok=null;}
    public synchronized void gir(int d)
    {
        if(kok==null) kok=new TreeGoze(d);
        else kok.gir(d);
    }
    public void Goze_sol_sag_sirala(){Goze_sol_sag_siralayici(kok);}
    public void sol_Goze_sag_sirala(){sol_Goze_sag_siralayici(kok);}
    public void sol_sag_Goze_sirala(){sol_sag_Goze_siralayici(kok);}
    public void Goze_sag_sol_sirala(){Goze_sag_sol_siralayici(kok);}
    public void sag_Goze_sol_sirala(){sag_Goze_sol_siralayici(kok);}
    public void sag_sol_Goze_sirala(){sag_sol_Goze_siralayici(kok);}
    public void Goze_sol_sag_siralayici(TreeGoze Goze)
    {
        if(Goze==null) return;
        System.out.print(Goze.veri+" ");
        Goze_sol_sag_siralayici(Goze.sol);
        Goze_sol_sag_siralayici(Goze.sag);
    }
    public void Goze_sag_sol_siralayici(TreeGoze Goze)
    {
        if(Goze==null) return;
        System.out.print(Goze.veri+" ");
        Goze_sag_sol_siralayici(Goze.sag);
        Goze_sag_sol_siralayici(Goze.sol);
    }
}
  
```



```

public void sol_Goze_sag_siralayici(TreeGoze Goze)
{
if(Goze==null) return;
sol_Goze_sag_siralayici(Goze.sol);
System.out.print(Goze.veri+" ");
sol_Goze_sag_siralayici(Goze.sag);
}
public void sag_Goze_sol_siralayici(TreeGoze Goze)
{
if(Goze==null) return;
sag_Goze_sol_siralayici(Goze.sag);
System.out.print(Goze.veri+" ");
sag_Goze_sol_siralayici(Goze.sol);
}
public void sol_sag_Goze_siralayici(TreeGoze Goze)
{
if(Goze==null) return;
sol_Goze_sag_siralayici(Goze.sol);
sol_Goze_sag_siralayici(Goze.sag);
System.out.print(Goze.veri+" ");
}
public void sag_sol_Goze_siralayici(TreeGoze Goze)
{
if(Goze==null) return;
sag_Goze_sol_siralayici(Goze.sag);
sag_Goze_sol_siralayici(Goze.sol);
System.out.print(Goze.veri+" ");
}
}

```

Program 10.7.2 TreeTest sınıfı

```

public class TreeTest
{
public static void main(String args[])
{
Tree agac=new Tree();
int sayi;
System.out.println("Örijinal olarak ağaç yapısında tanımlanan veriler: ");
for(int i=0;i<=6;i++)
{
sayi=(int)(Math.random()*45)+1;
System.out.print(sayi+" ");
agac.gir(sayi);
}
System.out.println("\nsol ağaç-göze-sağ ağaç düzenlemesi : ");
agac.sol_Goze_sag_sirala();
System.out.println("\ngöze sol ağaç sağ ağaç düzenlemesi : ");
agac.Goze_sol_sag_sirala();
System.out.println("\nsol ağaç sağ ağaç göze düzenlemesi : ");
agac.sol_sag_Goze_sirala();
System.out.println("\nsağ ağaç göze sol ağaç düzenlemesi : ");
agac.sag_Goze_sol_sirala();
System.out.println("\ngöze sol ağaç sağ ağaç düzenlemesi : ");
agac.Goze_sag_sol_sirala();
System.out.println("\nsağ ağaç sol ağaç göze düzenlemesi : ");
agac.sag_sol_Goze_sirala();
}
}

```

```
}  
}
```

```
----- Capture Output -----  
> "C:\turhan\java\bin\javaw.exe" TreeTest  
orijinal olarak ağaç yapısında tanımlanan veriler:  
4 37 2 25 33 4 21  
sol ağaç-göze-sağ ağaç düzenlemesi :  
2 4 21 25 33 37  
göze sol ağaç sağ ağaç düzenlemesi :  
4 2 37 25 21 33  
sol ağaç sağ ağaç göze düzenlemesi :  
2 21 25 33 37 4  
sağ ağaç göze sol ağaç düzenlemesi :  
37 33 25 21 4 2  
göze sol ağaç sağ ağaç düzenlemesi :  
4 37 25 33 21 2  
sağ ağaç sol ağaç göze düzenlemesi :  
37 33 25 21 2 4  
> Terminated with exit code 0.
```

Java sınıfında benzer yapıyı veren TreeSet sınıfı mevcuttur. Bu set aynı değeri birden fazla kabul etmez. Burada verilen verilerin üstteki şekilde aynı olduğuna dikkati çekmek istiyoruz.

Problem 10.7.3 TreeSet sınıf örneği

```
import java.util.*;  
import javax.swing.JOptionPane;  
  
public class TreeTest1  
{  
    public static String print(Collection c)  
    {  
        String s="";  
        Iterator i=c.iterator();  
        while(i.hasNext()) {s+=i.next()+"\n";}  
        return s;  
    }  
    public static void main(String args[])  
    {  
        TreeSet<Integer> tree=new TreeSet<Integer>();  
        int number;  
        System.out.println("orijinal olarak ağaç yapısında tanımlanan veriler: ");  
        tree.add(new Integer(13));  
        tree.add(new Integer(4));  
        tree.add(new Integer(36));  
        tree.add(new Integer(27));  
        String s=print(tree);  
        System.out.print(s);  
    }  
}
```

```
----- Capture Output -----  
> "C:\turhan\java\bin\javaw.exe" TreeTest1
```

orijinal olarak ağaç yapısında tanımlanan veriler:

4
13
27
36

> Terminated with exit code 0.> Terminated with exit code 0.

TreeSet sınıfı için ikinci bir örnek daha verelim:

Program 10.7.4 TreeSet class example

```
import java.util.*;
import javax.swing.*;

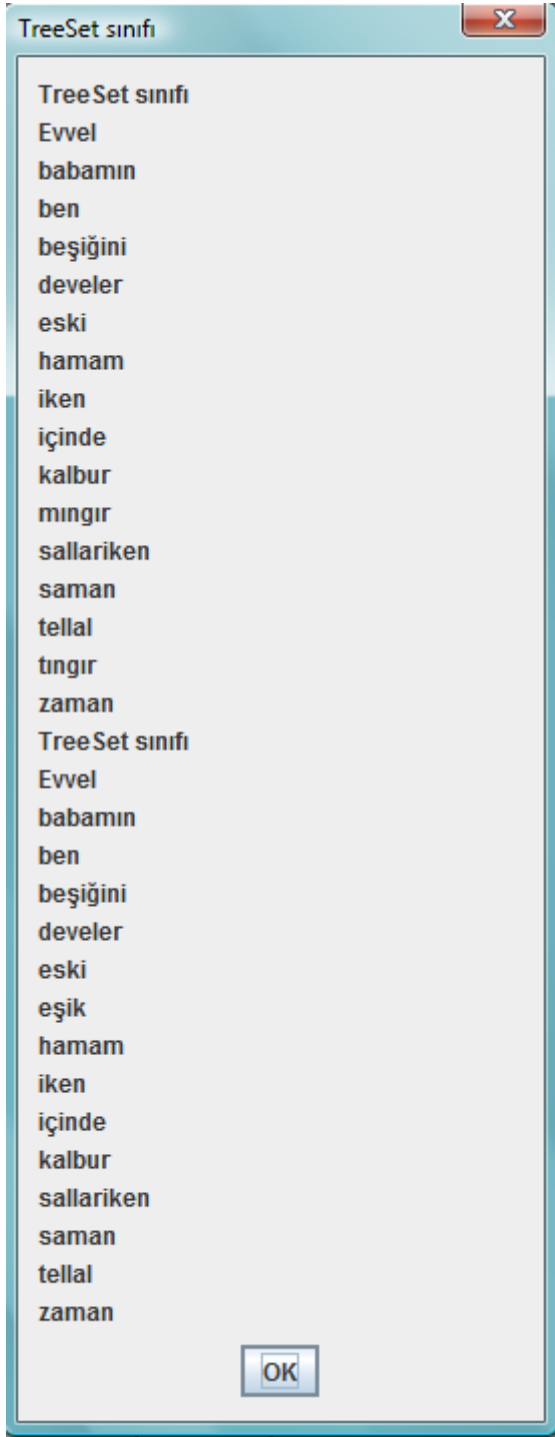
public class TreeSet2
{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static String[] boyutluStringgir(String s)
    {
        StringTokenizer token=new StringTokenizer(s);
        int n=token.countTokens()-1;
        int m=n+1;
        String a[]=new String[m];
        int j=0;
        while(token.hasMoreTokens())
        {
            String ax=new String(token.nextToken());
            a[j++]=ax;
        }
        return a;
    }

    public static void main(String args[])
    {
        TreeSet<String> a=new TreeSet<String>();
        String s3="Evvel zaman içinde kalbur saman içinde develer tellal iken eski hamam içinde ben babamın beşiğini tıngır mıngır sallarken";

        String s[]=boyutluStringgir(s3);
        for(int i=0;i<s.length;i++) a.add(s[i]);
        String s1="TreeSet sınıfı\n";
        String s2="TreeSet sınıfı";
        s1+=cikti(a);
        a.remove("tıngır");
        a.remove("mıngır");
        a.add("eşik");
        s1+="TreeSet sınıfı\n";
        s1+=cikti(a);
        JOptionPane.showMessageDialog(null,s1,s2,
```

```
JOptionPane.PLAIN_MESSAGE);  
}  
}
```



Bir TreeSet örneği daha verelim. Bu örnekte ağaç yapımıza Double sayılar yüklüyoruz.

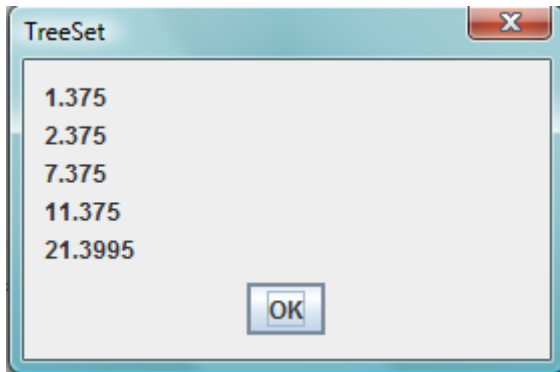
Program 10.7.5 TreeSet sınıf örneği

```
class TreeSet3 {  
  
    public static void main(String args[])
```

```

{
    TreeSet<Double> a=new TreeSet<Double>();
    Double s1=new Double(1.375);
    Double s2=new Double(2.375);
    Double s3=new Double(21.3995);
    Double s4=new Double(1.375);
    Double s5=new Double(7.375);
    Double s6=new Double(11.375);
    a.add(s1);
    a.add(s2);
    a.add(s3);
    a.add(s4);
    a.add(s5);
    a.add(s6);
    Iterator i=a.iterator();
    String s="";
    while(i.hasNext())
    {s+=i.next()+"\n";}
    JOptionPane.showMessageDialog(null,s,"TreeSet",
    JOptionPane.PLAIN_MESSAGE);
}
}

```



Örnekte 1.375 değeri 2 kere yüklendiği halde sadece 1 adet mevcuttur. TreeSet'teki sıralama mantığını Comparator (karşılaştırma) sınıfı kullanarak değiştirebiliriz. Bir önceki örnekteki sayı dizinini ters olarak elde etmek istersek

Program 10.7.6 TreeSet sınıf örneği, Double değişkenleri büyükten küçüğe sıralar

```

import javax.swing.*;
import java.util.*;
class Comparator1 implements Comparator<Object>
{
    public int compare(Object o1,Object o2)
    {
        Double I1=(Double)o1;
        Double I2=(Double)o2;
        double i1=I1.doubleValue();
        double i2=I2.doubleValue();
        if(i1 > i2) return -1;
        else if(i1 < i2) return 1;
        else return 0;
    }
}

```

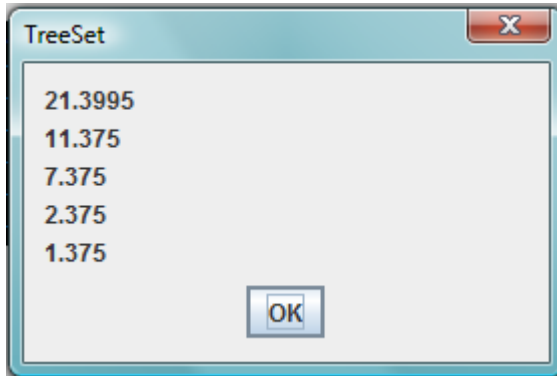
```

public class TreeSet3A {

    public static void main(String args[])
    {
        TreeSet<Double> a=new TreeSet<Double>(new Comparator1());
        Double s1=new Double(1.375);
        Double s2=new Double(2.375);
        Double s3=new Double(21.3995);
        Double s4=new Double(1.375);
        Double s5=new Double(7.375);
        Double s6=new Double(11.375);
        a.add(s1);
        a.add(s2);
        a.add(s3);
        a.add(s4);
        a.add(s5);
        a.add(s6);
        Iterator i=a.iterator();
        String s="";
        while(i.hasNext())
        {s+=i.next()+"\n";}
        JOptionPane.showMessageDialog(null,s,"TreeSet",
        JOptionPane.PLAIN_MESSAGE);
    }
}

```

Program sonucu



Görüldüğü gibi program kurucu metod dışında aynı, ancak kurucu metodu Comparator interface kullanarak yazdığımız Compare1 metoduyla çağırdık. Bu metodu kullanarak ağaç yapısındaki karşılaştırmanın nasıl yapılacağını sınıfa bildirmiş olduk. Benzer yapıyı Integer olarak kuralım:

Program 10.7.7 TreeSet sınıf örneği, Integer değişkenleri Comparator kullanarak büyükten küçüğe sıralar

```

import java.util.*;

class TreeSet4
{
    public static void main(String arg[])
    {
        TreeSet<Integer> t=new TreeSet<Integer>(new Comparator2());
        t.add(new Integer(20));
    }
}

```

```

t.add(new Integer(10));
t.add(new Integer(30));
t.add(new Integer(100));
System.out.println(t); // 10 ,20 ,30,100
}
}
class Comparator2 implements Comparator<Object>
{
public int compare(Object o1,Object o2)
{
Integer I1=(Integer)o1;
Integer I2=(Integer)o2;
int i1=I1.intValue();
int i2=I2.intValue();
if(i1>i2) return -1;
else if(i1 < i2) return 1;
else return 0;
}}

```

```

----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" TreeSet4
[100, 30, 20, 10]

> Terminated with exit code 0.

```

Compare interface' i kullanarak diğ er sınıfları da karşılaştırabiliriz, bir sonraki örnekte String karşılaştırması mevcuttur.

Program 10.7.8 TreeSet sınıf örneği, String değişkenleri Comparator kullanarak büyükten küçüğe sıralar

```

import java.util.*;

public class TreeSet5
{
public static void main(String arg[])
{
TreeSet<String> t=new TreeSet<String>(new Comparator3());
t.add("Ali");
t.add("veli");
t.add("49");
t.add("elli");
System.out.println(t); // 10 ,20 ,30,100
}
}

class Comparator3 implements Comparator<Object>
{
public int compare(Object o1, Object o2)
{
//parameter are of type Object, so we have to downcast it to Employee objects
String emp1 = (String)o1;

```

```
String emp2 = (String)o2;
//String sınıfının compareTo metodunu kullanıyor
return emp1.compareTo(emp2);
}
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" TreeSet5
[49, Ali, elli, veli]

> Terminated with exit code 0.
```

Bu son 2 programda System.out.print metodunun TreeSet metodu çıktısını direk olarak verdiğini de not edelim. Compare kullanarak sadece göreceli basit sınıfları değil daha kompleks yapıdaki sınıfları da sıraya koyabiliriz. Bir sonraki örnekte

Program 10.7.9 TreeSet sınıf örneği, Personel sınıfı değişkenleri Comparator kullanarak isme ve yaşa göre büyükten küçüğe sıralar

```
import java.util.*;
import javax.swing.*.*;
class Personel{
    public int yas;
    public String isim;
    public Personel(String isimi,int yasi)
    { isim=isimi;
      yas=yasi;
    }
    public String cikti()
    { String s="isim = "+isim+" Yaş = "+yas;
      return s;
    }
}

class yasComparator implements Comparator<Object>{

    public int compare(Object emp1, Object emp2)
    {
        int emp1yas = ((Personel)emp1).yas;
        int emp2yas = ((Personel)emp2).yas;

        if(emp1yas > emp2yas)
            return 1;
        else if(emp1yas < emp2yas)
            return -1;
        else
            return 0;
    }
}

class isimComparator implements Comparator<Object>{
```



```

public int compare(Object emp1, Object emp2){
    String emp1isim = ((Personel)emp1).isim;
    String emp2isim = ((Personel)emp2).isim;
    return emp1isim.compareTo(emp2isim);
}
}

public class ComparatorTest{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext())
        { Personel p=(Personel)i.next();
          s+=p.cikti()+"\n";}
        return s;
    }
    public static String cikti(Personel p[])
    {
        String s="";
        for(int i=0; i < p.length; i++)
        {s+="Personel " + (i+1) + " isim : " + p[i].cikti()+"\n";}
        return s;
    }
}

public static void main(String args[]){
    Personel p[] = new Personel[4];
    p[0] = new Personel("Turhan",55);
    p[1] = new Personel("Nurhan",50);
    p[2] = new Personel("Defne",10);
    p[3] = new Personel("Birsen",57);

    String s="Orijinal personel listesi \n";
    s+=cikti(p);
    //Şimdi listeyi TreeSet dinamik yapısına Comperatör kullanarak yükleyeceğiz
    TreeSet<Personel> t1=new TreeSet<Personel>(new isimComparator());
    for(int i=0;i<p.length;i++)
    {t1.add(p[i]);}
    s+="İsme göre sıralanmış liste : \n"+cikti(t1);
    TreeSet<Personel> t2=new TreeSet<Personel>(new yasComparator());
    for(int i=0;i<p.length;i++)
    {t2.add(p[i]);}
    s+="Yaşa göre sıralanmış liste : \n"+cikti(t2);
    JOptionPane.showMessageDialog(null,s,"TreeSet ve Comparator sınıfları kullanılması örneği",
    JOptionPane.PLAIN_MESSAGE);
}
}

```



Görüldüğü gibi programımız TreeSet kullanarak listeyi isme ve yaşa göre sıralayabiliyor. Aynı işlemi TreeSet yerine diğer boyutlu yapılardan bazılarıyla da yapabiliriz. Bir sonraki örnekte TreeSet yerine Arrays sınıfını kullanarak bu işlemin yapılmasını görmekteyiz.

Program 10.7.10 Arrays sınıf örneği, Personel sınıfı değişkenleri Comparator kullanarak isme ve yaşa göre büyükten küçüğe sıralar

```
import java.util.*;
import javax.swing.*;

public class ComparatorTest1 {

    public static String cikti(Personel p[])
    {
        String s="";
        for(int i=0; i < p.length; i++)
        {s+="Personel " + (i+1) + " isim : " + p[i].cikti()+"\n";}
        return s;
    }
    public static void main(String args[]){
        Personel p[] = new Personel[4];
        p[0] = new Personel("Turhan",55);
        p[1] = new Personel("Nurhan",50);
        p[2] = new Personel("Defne",10);
        p[3] = new Personel("Birsen",57);

        String s="Orijinal personel listesi \n";
        s+=cikti(p);
        //Şimdi listeyi TreeSet dinamik yapısına Comperatör kullanarak yükleyeceğiz
        Arrays.sort(p, new isimComparator());
        s+="isme göre sıralanmış personel listesi \n";
        s+=cikti(p);
        s+="Yaşa göre sıralanmış liste : \n";
    }
}
```

```

Arrays.sort(p, new yasComparator());
s+=cikti(p);
OptionPane.showMessageDialog(null,s,"Array ve Comparator sınıfları kullanılması örneği",
OptionPane.PLAIN_MESSAGE);
}
}

```

Şimdi TreeSet sınıfının metodlarına bakalım. TreeSet sınıfının Kurucu metodları:

Sonuç	Kurucu metod	Tanım
tset =	new TreeSet<E>()	Doğal karşılaştırma yapısında bir TreeSet yaratır
tset =	new TreeSet<E>(comp)	Bizim tanımladığımız bir Comparator comp karşılaştırma yapısında TreeSet yaratır
tset =	new TreeSet<E>(coll)	Collection coll sınıfındaki listeyi aktararak TreeSet yaratır.
tset =	new TreeSet<E>(sset)	SortedSet smp den veriyi okuyarak yeni bir TreeSet yaratır.

TreeSet, temel işlemleri için Collection sınıfı metodlarından yararlanır. Bu metodları da Set sınıfı üzerinden aktarır. Bu metodları temel olarak

Collection coll; boolean b; Object obj; int i; Iterator it;

dönüş	Metod	İşlem
<i>Elemanları dinamik değişken listeye ekleme</i>		
b =	coll.add(obj)	Obj nesnesini dinamik boyutlu değişkene aktarır. Eğer liste değişmişse true değeri verir (TreeList gibi sınıflarda yeni eklenmeye çalışılan eleman listede varsa yeni değer eklenmez, bu yüzden liste değişmez)
b =	coll.addAll(coll)	Collection sınıfından (veya Collection sınıfının TreeSet gibi alt sınıflarından herhangi birinde tanımlanmış) dinamik değişken listesini dinamik boyutlu değişkene aktarır. Eğer liste değişmişse true değeri verir
<i>Elemanları dinamik değişken listeden silme</i>		
	coll.clear()	Tüm elemanları siler.
b =	coll.remove(obj)	Obj nesnesini listeden siler.
b =	coll.removeAll(coll)	Collection sınıfından (veya Collection sınıfının TreeSet gibi alt sınıflarından herhangi birinde tanımlanmış) dinamik değişken listesini dinamik boyutlu değişkenden siler. Eğer liste değişmişse true değeri verir.
b =	coll.retainAll(coll)	Collection sınıfından (veya Collection sınıfının TreeSet gibi alt sınıflarından herhangi birinde tanımlanmış) dinamik değişken listesinde mevcut bulunmayan liste elemanlarını dinamik boyutlu değişkenden siler. Eğer liste değişmişse true değeri verir.
<i>Collectionda mevcut olup olmadığının testi</i>		
b =	coll.contains(obj)	Obj Collectionda mevcutsa true değeri verir
b =	coll.containsAll(coll)	Coll listesindeki tüm nesnelere Collectionda mevcutsa true değeri verir.
b =	coll.isEmpty()	Liste boşsa true değeri verir.
<i>Diğer</i>		
it =	coll.iterator()	Bu Collection için sayıcı (Iterator) atar
i =	coll.size()	Listenin boyutunu verir.
Object[] =	coll.toArray()	Listeyi static boyutlu değişkene (Array) çevirir.
Object[] =	coll.toArray(Object[])	Listeyi Object te verilen sınıfta static boyutlu değişkene çevirir.

Aynı zamanda SortedSet metodunun elemanlarını da implement yöntemiyle alır.

SortedSet<E> sset;

sonuç	Metod	Tanım
<i>comp</i> =	<code>sset.comparator()</code>	Kullanılan Comparator metodunu verir
<i>obj</i> =	<code>sset.firstKey()</code>	İlk eleman (sıraya girmiş formda)
<i>obj</i> =	<code>sset.lastKey()</code>	Son eleman(sırayagirmiş formda)
<i>sset</i> =	<code>sset.headMap(<i>obj</i>)</code>	Sıraya girmiş obj nesnesinden küçük olan tüm seti verir
<i>sset</i> =	<code>sset.tailMap(<i>obj</i>)</code>	Sıraya girmiş obj nesnesinden büyük olan tüm seti verir
<i>sset</i> =	<code>sset.subMap(<i>from,to</i>)</code>	Sıraya girmiş <i>from</i> nesnesinden büyük ve <i>to</i> nesnesinden küçük olan seti verir

Bir sonra inceleyeceğimiz gurup TreeMap gurubu. Bu gurup her bir Gözesinde iki nesne barındırır ve aynı zamanda TreeSEtte olduğu gibi iki adres içerir ve ağaç yapısında sıralar. Genellikle barındırdığı iki nesneden birisi anahtar olarak kullanılır, diğeri ise gerçek anlamda kullanılan veriyi içerir. İlk örneğimizde basit bir problem verdik. Burada haftanın günleri gün numarasıyla birlikte TreeMap sınıfında depolandı.

10.7.11 TreeMap sınıfı örneği

```

import java.util.*;

public class TreeMapTest{
    public static void main(String[] args) {
        System.out.println("Tree Map örneği!\n");
        TreeMap <Integer, String>tMap = new TreeMap<Integer, String>();
        tMap.put(1, "Pazar");
        tMap.put(2, "Pazartesi");
        tMap.put(3, "Salı");
        tMap.put(4, "Çarşamba");
        tMap.put(5, "Perşembe");
        tMap.put(6, "Cuma");
        tMap.put(7, "Cumartesi");
        System.out.println("TreeMap anahtarları: " + tMap.keySet());
        System.out.println("TreeMap değerleri: " + tMap.values());
        System.out.println("5 numaralı anahtara karşı gelen değer: " + tMap.get(5)+ "\n");
        System.out.println("Birinci anahtar: " + tMap.firstKey() + " Değeri: "
            + tMap.get(tMap.firstKey()) + "\n");
        System.out.println("Son anahtar: " + tMap.lastKey() + " Değeri: "
            + tMap.get(tMap.lastKey()) + "\n");
        System.out.println("ilk veriyi siliyoruz: " + tMap.remove(tMap.firstKey()));
        System.out.println("ilk veri silindikten sonraki anahtarlar: " + tMap.keySet());
        System.out.println("ilk veri silindikten sonraki değerler: " + tMap.values() + "\n");
        System.out.println("Son veriyi siliyoruz: " + tMap.remove(tMap.lastKey()));
        System.out.println("Son veri silindikten sonraki anahtarlar : " + tMap.keySet());
        System.out.println("Son veri silindikten sonraki değerler : " + tMap.values());
    }
}

```

```

----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" TreeMapTest
Tree Map örneği!

TreeMap anahtarları: [1, 2, 3, 4, 5, 6, 7]
TreeMap değerleri: [Pazar, Pazartesi, Salı, Çarşamba, Perşembe, Cuma, Cumartesi]
5 numaralı anahtara karşı gelen değer: Perşembe

Birinci anahtar: 1 Değeri: Pazar

Son anahtar: 7 Değeri: Cumartesi

```

ilk veriyi siliyoruz: Pazar
ilk veri silindikten sonraki anahtarlar: [2, 3, 4, 5, 6, 7]
ilk veri silindikten sonraki deęerler: [Pazartesi, Salı, arşamba, Perşembe, Cuma, Cumartesi]

Son veriyi siliyoruz: Cumartesi
Son veri silindikten sonraki anahtarlar : [2, 3, 4, 5, 6]
Son veri silindikten sonraki deęerler : [Pazartesi, Salı, arşamba, Perşembe, Cuma]

> Terminated with exit code 0.

İkinci örneğimizde daha önce gördüğümüz kitap sınıfını statik boyutlu deęişken olarak kullanan kütüphane sınıfının dinamik boyutlu deęişken (TreeMap) kullanan versiyonunu oluşturacağız.

Program 4.2.11 kitap sınıfı

```
public class kitap
{
String isim;
String yazar;
String kitap_turu;
String dil;

public kitap(String kisim,String kyazar,String kkitap_turu,String kdil)
{ isim=kisim;
yazar=kyazar;
kitap_turu=kkitap_turu;
dil=kdil;
}

public kitap(kitap k)
{ isim=k.isim;
yazar=k.yazar;
kitap_turu=k.kitap_turu;
dil=k.dil;
}

public String toString()
{
String s="isim = "+isim+"\n";
s+="yazar = "+yazar+"\n";
s+="kitap türü = "+kitap_turu+"\n";
s+="kitap dili = "+dil+"\n";
return s;
}

public String cikti()
{
String s="isim = "+isim+"\n";
s+="yazar = "+yazar+"\n";
s+="kitap türü = "+kitap_turu+"\n";
s+="kitap dili = "+dil+"\n";
return s;
}
}
```

Program 4.2.12 roman sınıfı

```
public class roman extends kitap
{
public roman(String kisim,String kyazar,String klanguage)
```

```
{ super(kisim,kyazar,"roman",klanguage);}
}
```

Program 4.2.13 siir sınıfı

```
public class siir extends kitap
{
public siir(String kisim,String kyazar,String klanguage)
{ super(kisim,kyazar,"şiiir",klanguage);}
}
```

Program 4.2.14 ansiklopedi sınıfı

```
public class ansiklopedi extends kitap
{
public ansiklopedi(String kisim,String kyazar,String klanguage)
{ super(kisim,kyazar,"ansiklopedi",klanguage);}
}
```

Program 10.7.12 TreeMap boyutlu değişken yapısı kullanan kütüphane1 sınıfı

```
import java.util.*;

public class kutuphane1
{
String isim;
int kitap_sayisi;
TreeMap <String, kitap> kitap_ismi = new TreeMap<String, kitap>();
TreeMap <String, kitap> yazar = new TreeMap<String, kitap>();

public kutuphane1(String kisim,kitap b[])
{ isim=kisim;
  kitap_sayisi=0;
  ekle(b);
}

public void ekle(kitap b)
{
  kitap_ismi.put(b.isim,b);
  yazar.put(b.yazar,b);
  kitap_sayisi=kitap_ismi.size();
}

public void ekle(kitap b[])
{
  for(int i=0;i<b.length;i++)
  { ekle(b[i]);}
}

public void sil(kitap b)
{
  kitap_sayisi--;
  kitap_ismi.remove(b.isim);
  yazar.remove(b.yazar);
  kitap_sayisi=kitap_ismi.size();
}
```

```

public String ismiyle_listele()
{String s=kitap_ismi.entrySet().toString();
return s;
}
public String yazariyla_listele()
{String s=yazar.entrySet().toString();
return s;
}
public String cikti_isim()
{      String s="kutuphane isim : "+isim+"\n";
s+="Toplam kitap sayısı : "+kitap_sayisi+"\n\n";
s+=ismiyle_listele();
return s;
}
public String cikti_yazar()
{      String s="kutuphane isim : "+isim+"\n";
s+="Toplam kitap sayısı : "+kitap_sayisi+"\n\n";
s+=yazariyla_listele();
return s;
}
}
}

```

Program 10.7.13 TreeMap boyutlu değişken yapısı kullanan kütüphane1 sınıfı test programı

```

import javax.swing.*;
class kutuphanetest1
{
public static void ciktipenceresi(String s)
{
String s1="kutuphane sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}
public static void main(String args[])
{
kitap b[]=new kitap[5];
b[0]=new kitap("Bilim ve Teknik","TÜBİTAK","magazine","Turkish");
b[1]=new roman("It is all quiet in the westen front ","Erich Maria Reamarque","İngilizce");
b[2]=new kitap("The God Delusion","Richard Dawkins","fiction","İngilizce");
b[3]=new siir("Human lanscapes from my country","Nazım Hikmet","İngilizce ve Türkçeh");
b[4]=new ansiklopedi("Encyclopedia Britannica","İngilizce");
kutuphane l=new kutuphane("izmir milli kütüphanesi",b);
String s=l.cikti();
ciktipenceresi(s);
}}

```



TreeMap sınıfının kurucu metodları şunlardır:

Result	Constructor	Description
<i>tmap</i> =	<code>new TreeMap()</code>	Yeni TreeMap sınıfı oluşturur.
<i>tmap</i> =	<code>new TreeMap(comp)</code>	Comparator <i>comp</i> kullanan yeni TreeMap sınıfı oluşturur.
<i>tmap</i> =	<code>new TreeMap(mp)</code>	Map sınıfını kullanarak yeni TreeMap oluşturur
<i>tmap</i> =	<code>new TreeMap(smp)</code>	<i>SortedMap</i> sınıfını kullanarak yeni TreeMap oluşturur

TreeMap temel olarak Map sınıfından türemiş olup Map sınıfı metodlarını kullanır.

sonuç	Metod	Tanım
<i>Map gurubuna key, val çiftini girme</i>		
<i>obj</i> =	<code>m.put(key, val)</code>	Key anahtar değeri ve val değerini birlikte yükler
	<code>m.putAll(map2)</code>	Diğer bir map gurubunun tüm elemanlarını yeni map gurubuna yükler
<i>Key-value gurubunu silme</i>		

	<code>m.clear()</code>	Map gurubunun tüm elemanlarını siler
<code>obj =</code>	<code>m.remove(key)</code>	Key anahtarlı elemanı siler
<i>Veri çağırma</i>		
<code>b =</code>	<code>m.containsKey(key)</code>	Eğer m key anahtarını içeriyorsa true değerini döner
<code>b =</code>	<code>m.containsValue(val)</code>	Eğer m val değerini içeriyorsa true değerini döner
<code>obj =</code>	<code>m.get(key)</code>	Key değerine karşı gelen val değerini döner
<code>b =</code>	<code>m.isEmpty()</code>	Eğer m boşsa true döner
<code>i =</code>	<code>m.size()</code>	M nin boyutunu verir
<i>Tüm değerleri veya anahtarları çağırma</i>		
<code>set =</code>	<code>m.entrySet()</code>	Tüm değerleri (val) döner
<code>set =</code>	<code>m.keySet()</code>	Tüm anahtarları (key) döner
<code>col =</code>	<code>m.values()</code>	Tüm değerleri (val) döner

10.8 DİNAMİK BOYUTLU DEĞİŞKENLER HASHTABLORARI

Hashtablosu sınıfı HashMap bir önceki bölümde incelediğimiz TreeMap sınıfına benzer bir yapı gösterir, ancak TreeMap'ten farklı olarak ikili referansla oluşan ağaç yapısı içermez ArrayList gibi tekli referans içeren ancak bir anahtar ve değer taşıyan ve sadece bir tane dinamik boyutlu adres referansı içeren bir guruptur.

Program 10.8.1 HashMap1 sınıfı

```
import java.util.*;
import javax.swing.*;

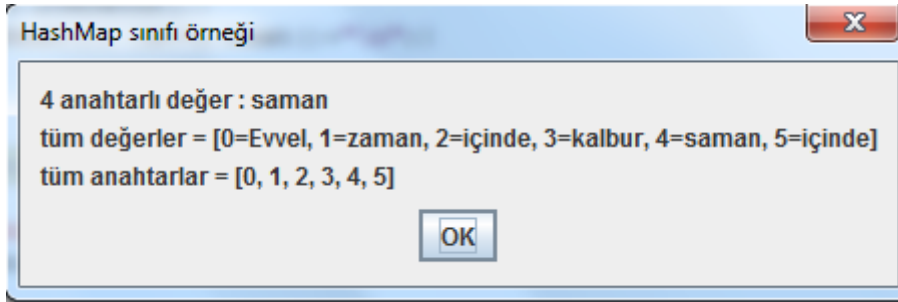
public class HashMap1
{
    public static String cikti(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static String[] boyutluStringgir(String s)
    {
        StringTokenizer token=new StringTokenizer(s);
        int n=token.countTokens()-1;
        int m=n+1;
        String a[]=new String[m];
        int j=0;
        while(token.hasMoreTokens())
        {
```

```

String ax=new String(token.nextToken());
a[j++]=ax;
}
return a;
}
public static void main(String args[])
{
HashMap<Integer,String> a=new HashMap<Integer,String>();
String s1="Evvel zaman içinde kalbur saman içinde";
String s[]=boyutluStringgir(s1);
for(int i=0;i<s.length;i++) a.put(i,s[i]);
String s2="HashMap sınıfı örneği\n";
String s3="4 anahtarlı değer : "+a.get(4)+"\n";
s3+="tüm değerler = "+a.entrySet()+"\n";
s3+="tüm anahtarlar = "+a.keySet()+"\n";
JOptionPane.showMessageDialog(null,s3,s2,JOptionPane.PLAIN_MESSAGE);
}
}

```



İkinci örnek olarak daha önce geliştirdiğimiz araç ve araç_kayıt sınıflarını ThreeMap yapısına uyarlayalım.

Program 10.8.2 arac1 sınıfı

```

import javax.swing.*;

public class arac1
{
//geçerli araç tipleri bus,truck,minitruck,minivan,auto
public String aractipi;
public String markasi;
public double guc; //KW olarak
public String renk;
public double agirlik; //yüksüz net ağırlık
public String plaka;
//kurucu metodlar

public arac1()
{aractipi="tanımlanmadı";
markasi="tanımlanmadı";
guc=0;
renk="tanımlanmadı";
plaka="tanımlanmadı";
agirlik=0;
}
}

```

```

public arac1(String i_aractipi,String i_markasi,double i_motor,String i_renk,double i_agirlik,String plakai)
{//arac tipleri "otobüs","kamyon","kamyonet","minibüs","otomobil","tır","motosiklet","bisiklet","yol dışı araç"
//if not define as "tanımlanmadı"
if(i_aractipi.equals("kamyon") ||
i_aractipi.equals("kamyonet") ||
i_aractipi.equals("otobüs")||
i_aractipi.equals("minibüs")||
i_aractipi.equals("otomobil") ||
i_aractipi.equals("motosiklet") ||
i_aractipi.equals("bisiklet") ||
i_aractipi.equals("tır") ||
i_aractipi.equals("yol dışı araç")
)
{aractipi=i_aractipi;}
else
{aractipi="tanımlanmadı";}
markasi=i_markasi;
guc=i_motor;
renk=i_renk;
agirlik=i_agirlik;
plaka=plakai;
}

public arac1(arac A)
{//gerekli tanımları yap
aractipi=A.aractipi;
markasi=A.markasi;
guc=A.guc;
renk=A.renk;
agirlik=A.agirlik;
plaka=A.plaka;
}

public void define_arac(String i_aractipi,String i_markasi,double i_motor,String i_renk,double i_agirlik,String i_plaka)
{//arac type "otobüs","kamyon","kamyonet","minibüs","otomobil","tır"
//if not define as "tanımlanmadı"
if(i_aractipi.equals("kamyon") ||
i_aractipi.equals("kamyonet") ||
i_aractipi.equals("otobüs")||
i_aractipi.equals("minibüs")||
i_aractipi.equals("otomobil") ||
i_aractipi.equals("motosiklet") ||
i_aractipi.equals("bisiklet") ||
i_aractipi.equals("tır") ||
i_aractipi.equals("yol dışı araç")
)
{aractipi=i_aractipi;}
else
{aractipi="tanımlanmadı";}
markasi=i_markasi;
guc=i_motor;
renk=i_renk;
agirlik=i_agirlik;
plaka=i_plaka;
}

public String cikti()
{String s1="";

```

```

s1+="aracın tipi : "+aractipi+"\n";
s1+="Aracın markası : "+markasi+"\n";
s1+="Motor gücü : "+guc+" KW\n";
s1+="renk : "+renk+"\n";
s1+="ağırlık : "+agirlik+" kg\n";
s1+="plaka = "+plaka+"\n";
return s1;//gerekli tanımları yap
}

public String toString() {return cikti();}

public void ciktipenceresi()
{
String s1="arac sınıfı çıktı penceresi";
JOptionPane.showMessageDialog(null,cikti(),s1, JOptionPane.PLAIN_MESSAGE);
}

}

```

Program 10.8.3 minibus1 sınıfı

```

public class minibus1 extends arac1
{
public minibus1(String i_marka,double i_mminir,String i_renk,double i_agirlik,String i_plaka)
{super("minibüs",i_marka,i_mminir,i_renk,i_agirlik,i_plaka);}

public minibus1()
{ super();
aractipi="minibüs";
}
public minibus1(minibus A)
{super(A.v);
aractipi="minibüs";
}
}

```

Program 10.8.4 otobus1 sınıfı

```

public class otobus1 extends arac1
{
public otobus1(String i_marka,double i_motor,String i_renk,double i_agirlik,String i_plaka)
{super("otobüs",i_marka,i_motor,i_renk,i_agirlik,i_plaka);}

public otobus1()
{ super();
aractipi="otobüs";
}
public otobus1(otobus A)
{super(A.v);
aractipi="otobüs";
}}

```

Program 10.8.5 otobus1 sınıfı

```

public class kamyon1 extends arac1
{

```

```

public kamyon1(String i_marka,double i_motor,String i_renk,double i_agirlik,String i_plaka)
{super("kamyon",i_marka,i_motor,i_renk,i_agirlik,i_plaka);}

public kamyon1()
{ super();
  aractipi="kamyon";
}
public kamyon1(kamyon A)
{super(A.v);
  aractipi="kamyon";
}}

```

Program 10.8.6 minibus1 sınıfı

```

public class minibus1 extends arac1
{

public minibus1(String i_marka,double i_mminir,String i_renk,double i_agirlik,String i_plaka)
{super("minibüs",i_marka,i_mminir,i_renk,i_agirlik,i_plaka);}

public minibus1()
{ super();
  aractipi="minibüs";
}
public minibus1(minibus A)
{super(A.v);
  aractipi="minibüs";
}
}

```

Program 10.8.7 arac_kayit1 sınıfı

```

import java.util.*;
public class arac_kayit1
{
String isim;
int arac_sayisi;
HashMap<String,arac1> v;

public arac_kayit1(String kisim)
{isim=kisim;
  arac_sayisi=0;
  v=new HashMap<String,arac1>();
}
public String toString()
{
  String s="araç kayıt istasyonu adı : "+isim+"\n";
  s+="Toplam arac sayısı : "+arac_sayisi+"\n\n";
  s+=listele();
  return s;
}
public void ekle(arac1 b)
{
  v.put(b.plaka,b);
  arac_sayisi++;
}

public void sil(arac1 b)
{v.remove(b.plaka);
}
}

```

```
arac_sayisi--;}

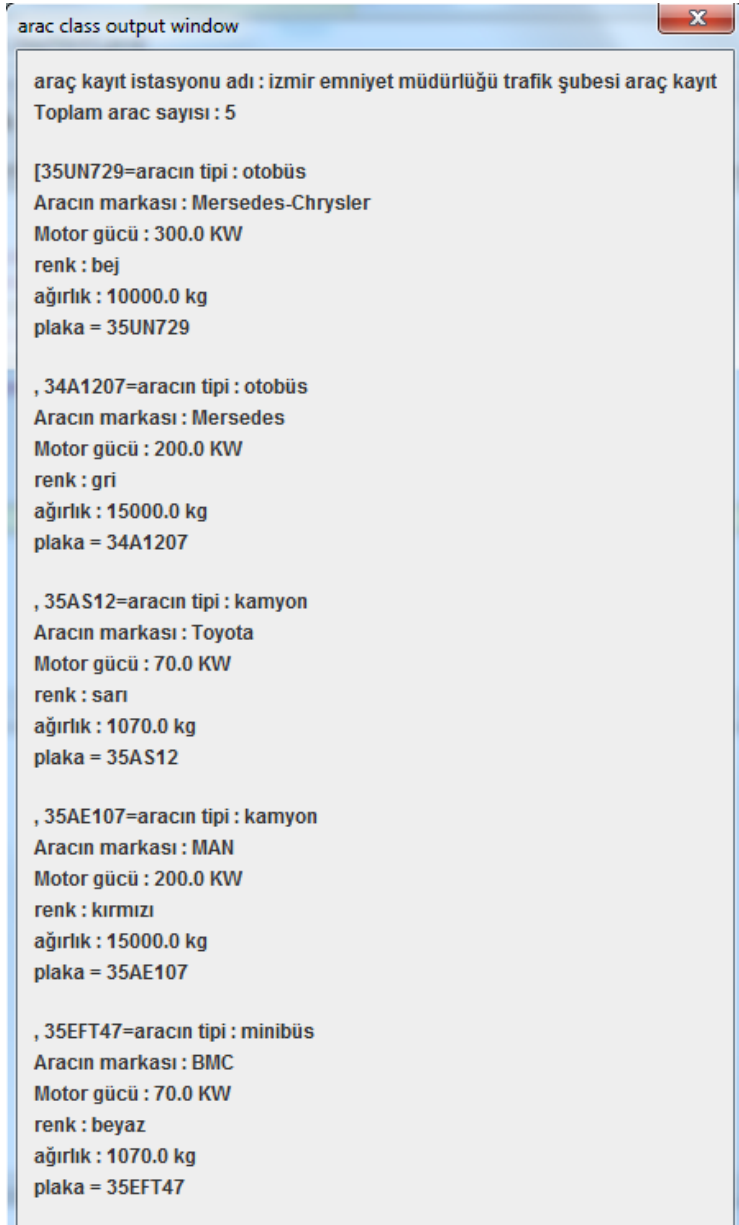
public String listele()
{String s=v.entrySet().toString();
return s;}

public String get(String regno)
{String s=v.get(regno).toString();
return s; }
}
```

Program 10.8.8 arac_kayittest1 test sınıfı

```
import javax.swing.*;
class arac_kayittest1
{
public static void outputwindow(String s)
{
String s1="arac class output window";
JOptionPane.showMessageDialog(null,s,s1, JOptionPane.PLAIN_MESSAGE);
}

public static void main(String args[])
{
arac1 v1=new arac1("kamyon","MAN",200.0,"kırmızı",15000,"35AE107");
arac1 v2=new arac1("otobüs","Mercedes-Chrysler",300.0,"bej",10000,"35UN729");
minibus1 v3=new minibus1("BMC",70,"beyaz",1070,"35EFT47");
otobus1 v4=new otobus1("Mercedes",200.0,"gri",15000,"34A1207");
kamyon1 v5=new kamyon1("Toyota",70,"sarı",1070,"35AS12");
arac_kayit1 vr=new arac_kayit1("izmir emniyet müdürlüğü trafik şubesi araç kayıt");
vr.ekle(v1);
vr.ekle(v2);
vr.ekle(v3);
vr.ekle(v4);
vr.ekle(v5);
String s=vr.toString()+"\n";
outputwindow(s);
}}
```



HashMap sınıfının kurucu metodları :

Result	Constructor	Description
<i>hmap</i> =	<code>new HashMap()</code>	16 ilk kapasiteli HashMap sınıfını oluşturur.
<i>hmap</i> =	<code>new HashMap(<i>ilkkapasite</i>)</code>	İlkkapasite boyutlu HashMap sınıfını oluşturur.
<i>hmap</i> =	<code>new HashMap(<i>ilkkapasite</i>, <i>yükfaktörü</i>)</code>	İlkkapasite ve yükfaktörü değerleriyle HashMap oluşturur.
<i>hmap</i> =	<code>new HashMap(<i>mp</i>)</code>	Map mp elemanlarının yüklendiği HashMap sınıfını oluşturur.

HashMap sınıfının diğer metodları Map sınıfından alınmış olup TreeMap bölümünde listelenmişti.

Diğer bir boyutlu dinamik yapı HashSet'dir. HashSet TreeSet'e benzer bir değer yüklenebilir, ve bir bağlantı referansına sahiptir, TreeSet'te olduğu gibi aynı değer tekrar yüklenirse yinelemz ve o değeri ek değer olarak listeye almaz.

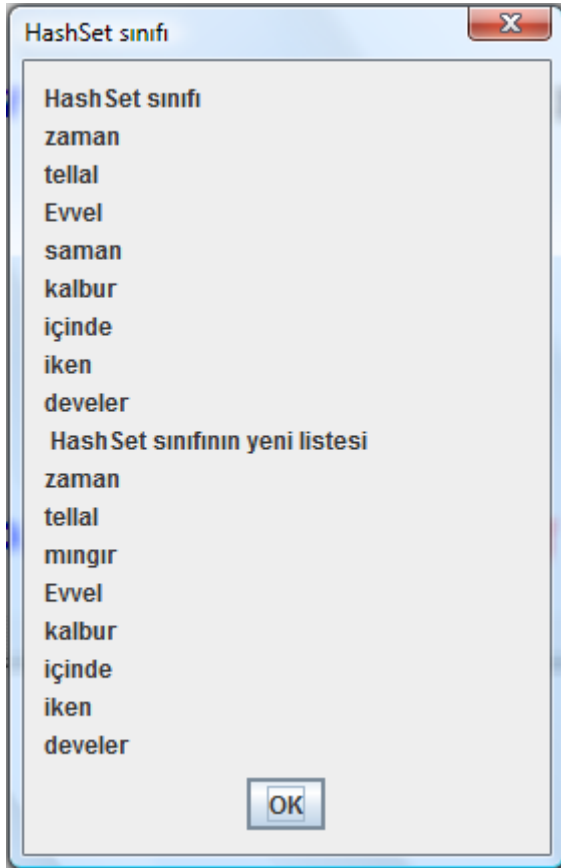
Program 10.8.5 HasSet1 class

```
import javax.swing.*;

public class HasSet1
{
    public static String print(Collection c)
    {
        String s="";
        Iterator i=c.iterator();
        while(i.hasNext()) {s+=i.next()+"\n";}
        return s;
    }

    public static String[] boyutluStringgir(String s)
    {
        StringTokenizer token=new StringTokenizer(s);
        int n=token.countTokens()-1;
        int m=n+1;
        String a[]=new String[m];
        int j=0;
        while(token.hasMoreTokens())
        {
            String ax=new String(token.nextToken());
            a[j++]=ax;
        }
        return a;
    }

    public static void main(String args[])
    {
        HashSet<String> a=new HashSet<String>();
        String s3="Evvel zaman içinde kalbur saman içinde develer tellal iken";
        String s[]=boyutluStringgir(s3);
        for(int i=0;i<s.length;i++) a.add(s[i]);
        String s1="HashSet sınıfı\n";
        String s2="HashSet sınıfı";
        s1+=print(a);
        a.remove("saman");
        a.add("mıngır");
        s1+=" HashSet sınıfının yeni listesi\n";
        s1+=print(a);
        JOptionPane.showMessageDialog(null,s1,s2,
        JOptionPane.PLAIN_MESSAGE);
    }
}
```

HashSet sınıfının kurucu metodları aşağıda verilmiştir. HashSet<E> sınıfı Set sınıfı (extend) ve HashMap sınıfından (implement) türetilmiştir.

Sonuç	Kurucu metod	Tanım
<code>hset = new HashSet<E>()</code>		16 ilk kapasiteli bir HashSet yaratır.
<code>hset = new HashSet<E>(ilkkapasite)</code>		<i>Ilkkapasite</i> ilk kapasiteli bir HashSet yaratır.
<code>hset = new HashSet<E>(ilkkapasite,yüklemefaktörü)</code>		<i>Ilkkapasite</i> ilk kapasiteli ve <i>yüklemefaktörü</i> yükleme faktörlü bir HashSet yaratır.
<code>hset = new HashSet<E>(coll)</code>		<i>Collection coll</i> sınıfı elemanlarını yükler

10.9 DİNAMİK BOYUTLU YAPILAR : NAVIGABLESET

NavigableSet çok kolay bir karşılaştırma sistemine sahip alt sistemlere kolaylıkla ayrılabilen bir sistem oluşturur. SortedSet sınıfından türetilmiştir. Lower,floor,celing ve higher metodlarıdaha küçük, eşit veya daha büyük elemanları guruplar olarak ayırabilir. NavigableSet seti azalan veya çoğulan yapıda olabilir.

`descendingIterator()` and `descendingSet()`

NavigableSet'in en ilginç metodlarından ikisi

`descendingIterator()` ve `descendingSet()` metodlarıdır.

`descendingSet()` metodu içindeki elemanların sırası ters çevrilmiş bir NavigableSet verir.

Basit bir örnek verirsek:

NavigableSet reverse=original.descendingSet();

`descendingIterator()` metodu NavigableSet'i ters sırayla iteasyonunu yapar. İç elemanların sırası değişmez.

Iterator reverse=original.descendingIterator();

headSet(), tailSet() and subSet()

`headSet()` metodu verilen elemandan daha küçük olan alt listeyi verir. Aşağıda küçük bir örnekle bu gösterilmiştir.

```
NavigableSet original=new TreeSet();
Original.add("1");
Original.add("2");
```

```
Original.add("3");
//Bu set "1" ve "2" değerlerini içerecektir:
SortedSet headset=original.headSet("3");
//Bu set "1","2" ve "3" değerlerini içerecektir çünkü "inclusive"=true olarak verilmiştir
NavigableSet headset=original.headSet("3",true);
```

tailSet() metodu headset metodu gibi çalışır, ancak verilen değerden daha büyük değerleri içerecek şekilde seçim yapar.

subSet() alt ve üst limitleri parameter olarak verir ve bu limitler arasındaki değerleri çağırır. Aşağıda bir örnek verilmiştir:

```
NavigableSet original = new TreeSet();
original.add("1");
original.add("2");
original.add("3");
original.add("4");
original.add("5");
//bu alt seçimde sadece "3" eđeri bulunacaktır
SortedSet subset = original.subSet("2", "4");
// bu alt seçimde "2", "3" ve "4" deđeri bulunacaktır, çünkü
// fromInclusive=true, ve toInclusive=true olarak verilmiştir.
NavigableSet subset = original.subSet("2", true, "4", true);
```

ceiling(), floor(), higher(), and lower()

ceiling() metodu en küçük elemanı çağırır. Altta bir örnek verilmiştir:

```
NavigableSet original = new TreeSet();
original.add("1");
original.add("2");
original.add("3");
//ceiling deđeri "2".
Object ceiling = original.ceiling("2");
//floor deđeri "2".
Object floor = original.floor("2");
```

floor() metodu **ceiling()** metodunun tersi olarak çalışır.

higher() metodu kendine eşit olmamak üzere daha büyük deđer alan elemanları verir. Örnek :

```
NavigableSet original = new TreeSet();
original.add("1");
original.add("2");
original.add("3");
//higher deđeri: "3".
Object higher = original.higher("2");
//lower deđeri: "1"
Object lower = original.lower("2");
```

lower()metodu **higher()** metodunun tersidir. Eşit olmamak üzere kendinden daha küçük deđerleri verir.

pollFirst() and pollLast()

pollFirst() metodu ilk elemanı return eder ve NavigableSet listesinden siler. Eğer liste boş ise null değeri döner. pollLast()metodu son elemanı return eder ve NavigableSet listesinden siler. Eğer liste boş ise null değeri döner. Alta iki örnekle bu metodların kullanımını verelim:

```
NavigableSet original = new TreeSet();
original.add("1");
original.add("2");
original.add("3");
//first değeri: "1"
Object first = original.pollFirst();
//last değeri: "3"
Object last = original.pollLast();
```

10.9.1 NavigableSetTest

```
import java.util.*;
import java.util.concurrent.*;

public class NavigableSetTest{
    public static void main(String[] args) {
        System.out.println("Navigable set Örneği!\n");
        NavigableSet <Integer>nSet = new ConcurrentSkipListSet<Integer>();
        nSet.add(10);
        nSet.add(20);
        nSet.add(50);
        nSet.add(30);
        nSet.add(100);
        nSet.add(80);
        // Returns an iterator over the elements in navigable set, in ascending order.
        Iterator iterator = nSet.iterator();
        System.out.print("yukarı doğru ilerleyen navigable set: ");
        //Ascending order list
        while (iterator.hasNext()){
            System.out.print(iterator.next() + " ");
        }
        System.out.println();
        //Descending order list
        System.out.println("Aşağı doğru ilerleyen navigable set: " + nSet.descendingSet() + "\n");
        //Greater than or equal to the given element
        System.out.println(" Navigable settteki 35 ten büyük veya eşit değer alan elemanlar: " + nSet.ceiling(35));
        //Less than or equal to the given element
        System.out.println(" Navigable settteki 35 ten büyük veya eşit değer alan elemanlar: " + nSet.floor(35) + "\n");
        //Viewing the portion of navigable set whose elements are strictly less than the given element
        System.out.println("Navigable set whose elements are strictly less than '40': " + nSet.headSet(40));
        //Viewing the portion of navigable set whose elements are greater than or equal to the given element
        System.out.println("Navigable settteki 35 ten büyük veya eşit değer alan elemanlar: '40': " + nSet.tailSet(40) + "\n");
        //Removing first element from navigable set
        System.out.println("Sil: "+nSet.pollFirst());
        //After removing the first element, now get navigable set
        System.out.println("yeni navigable set: " + nSet.descendingSet() + "\n");
        //Removing last element from navigable set
        System.out.println("elemanı sil: " + nSet.pollLast());
```

```
//After removing the last element, now get navigable set
System.out.println("yeni navigable set: " + nSet.descendingSet());
}
}
```

The result will be;

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" NavigableSetTest
Navigable set Örneği!

yukarı doğru ilerleyen navigable set: 10 20 30 50 80 100
Aşağı doğru ilerleyen navigable set: [100, 80, 50, 30, 20, 10]

Navigable setteki 35 ten büyük veya eşit değer alan elemanlar: 50
Navigable setteki 35 ten büyük veya eşit değer alan elemanlar: 30

Navigable set whose elements are strictly less than '40': [10, 20, 30]
Navigable setteki 35 ten büyük veya eşit değer alan elemanlar: '40': [50, 80, 100]

Sil: 10
yeni navigable set: [100, 80, 50, 30, 20]

elemanı sil: 100
yeni navigable set: [80, 50, 30, 20]

> Terminated with exit code 0.
```

Navigable Set metodları şunlardır:

Sonuç	Kurucu metod	Tanım
<code>nset =</code>	<code>new NavigableSet<E>()</code>	Doğal karşılaştırma yapısında bir NavigableSet yaratır
<code>nset =</code>	<code>new NavigableSet<E>(comp)</code>	Bizim tanımladığımız bir Comparator <i>comp</i> karşılaştırma yapısında NavigableSet yaratır
<code>nset =</code>	<code>new NavigableSet<E>(coll)</code>	<i>Collection coll</i> sınıfındaki listeyi aktararak NavigableSet yaratır.
<code>nset =</code>	<code>new NavigableSet<E>(sset)</code>	<i>SortedSet smp</i> den veriyi okuyarak yeni bir NavigableSet yaratır.

Sonuç	metod	Tanım
<code>Iterator<E> iter=</code>	<code>descendingIterator()</code>	16 ilk kapasiteli bir HashSet yaratır.
<code>nset =</code>	<code>new HashSet<E>(ilkkapasite)</code>	<i>Ilkkapasite</i> ilk kapasiteli bir HashSet yaratır.
<code>nset =</code>	<code>new HashSet<E>(ilkkapasite,yüklemefaktörü)</code>	<i>Ilkkapasite</i> ilk kapasiteli ve <i>yüklemefaktörü</i> yükleme faktörlü bir HashSet yaratır.
<code>nset =</code>	<code>new HashSet<E>(coll)</code>	<i>Collection coll</i> sınıfı elemanlarını yükler

10.10 DİNAMİK BOYUTLU DEĞİŞKEN YAPILARI: BİR GERÇEK UYGULAMA ÖRNEĞİ

İdeal gaz yasası

$$Pv=R*T$$

Denklemiyle tanımlanabilir. Burada P basınç(Pascal), T sıcaklık derece santigrad, v özgül hacim (m^3/kg), R gaz sabiti, değeri R0 üniversal gaz sabiti kullanılarak hesaplanabilir.) $R=R0/M$. Burada $R0=8314.5 J/(kmol K)$. İdeal gaz için özgül ısı denklemi $C_p(T) = a + bT + cT^2 + dT^3$ olarak tanımlanabilir, burada a,b,c ve d uydurulan polinom sabit katsayılarıdır. Bu tanımları kullanarak entalpi(J/kg) ve entropi (J/kgK) tanımlarına ulaşabiliriz.

$$h(T) = \int_{T_0}^T C_p(T) dT + h(T_0)$$

$$s(T) = \int_{T_0}^T \frac{C_p(T)}{T} dT + s(T_0) - R \ln \left(\frac{P}{P_0} \right)$$

$$h(T) = \int_{T_0}^T C_p(T) dT + h(T_0)$$

$$s(T) = \int_{T_0}^T \frac{C_p(T)}{T} dT + s(T_0) - R \ln \left(\frac{P}{P_0} \right)$$

Denklemlerden görüleceği gibi bir gaz için temel katsayılar verildiğinde diğer özellikler hesaplanarak bulunabilir. Temel veri programa ArrayList dinamik yapısı olarak girilecektir. idealgasData sınıfı ArrayList yapısına girilecek temel sınıf olarak oluşturulmuştur. Aşağıda program kodları verilmiştir.

10.10.1 idealgasData sınıfı

```
public class idealgasData extends Object
{
public String name; //ideal gaz ismi
public String formula; //ideal gas formülü
public double M; // ideal gaz moleküler ağırlığı
public double A[]; // Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T

public idealgasData(String namei,String formulai,double Mi,double a,double b,double c,double d)
{ name=namei;;
formula=formulai;
M=Mi;
A=new double[4];
A[0]=a;
A[1]=b;
A[2]=c;
A[3]=d;
}
public String toString()
{
String s="";
s+="ideal gaz ismi = "+name+"\n";
s+="ideal gaz formülü = "+formula+"\n";
s+="ideal gaz moleküler ağırlığı = "+M+" kg/kmol \n";
s+="Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T katsayıları\n";
for(int i=0;i<4;i++)
```

```
s+="A["+i+"] = "+A[i];
return s;
}
}
```

Artık idealgaz sınıfını tanımlayabiliriz. Ideal gaz verisi bu sınıfa idealgasData sınıfı ArrayList olarak okunacaktır. (idealgaInput sınıfını inceleyiniz)

10.10.2 idealgas sınıfı

```
//=====
// java Thermodynamik paketi
// sınıf: idealgas , ideal gazların termodinamik özellikleri
// Dr. Turhan Coban
// Ege Üniversitesi Mühendislik fakultesi
// Makina Bölümü
//
//=====
// Dosya ismi : idealgas.java
// Bu sınıf ideal gazların termodinamik özelliklerini hesaplar
//
//=====

import java.io.*;
import java.net.*;
import java.util.*;

public class idealgas
{
public idealgasData rd;
static double T0=298.15; // derece K
static double P0=1; //1 bar
public static double R=8.3145; //KJ/kgK

public idealgas(String igName)
{idealgasInput(igName);}

public String[] idealgaslist()
{ String s1=rd.name;String s2=idealgasInput("null");idealgasInput(s1);
StringTokenizer t=new StringTokenizer(s2);
int n=t.countTokens();
String s3[]=new String[n];
int i=0;
while(t.hasMoreTokens())
{
s3[i]=t.nextToken();
i++;
}
return s3;
}

public double Cp(double t)
{ double T=t+273.15;
return (rd.A[0]+rd.A[1]*T+rd.A[2]*T*T+rd.A[3]*T*T*T);
//KJ/kmol K
}
}
```

```

public double h(double t)
{ double T=t+273.15;
  double hi=rd.A[0]*(T-T0)+rd.A[1]/2.0*(T*T-T0*T0)+
  rd.A[2]/3.0*(T*T*T-T0*T0*T0)+rd.A[3]/4.0*(T*T*T*T-T0*T0*T0*T0);
  //KJ/kmol
  return hi;
}

public double s0(double t)
{ double T=t+273.15;
  double si=rd.A[0]*Math.log(T/T0)+rd.A[1]*(T-T0)+
  rd.A[2]/2.0*(T*T-T0*T0)+rd.A[3]/3.0*(T*T*T-T0*T0*T0);
  return si;
}

public double s(double t,double P)
{ double si=s0(t)-R*Math.log(P/P0);
  return si;
}

public String idealgasInput(String ig_name)
{
  String s="";
  ArrayList<idealgasData> gi = new ArrayList<idealgasData>();
  rd = new idealgasData("azot","N2",28.013,28.90,-0.1571e-2,0.8081e-5,-2.873e-9);
  gi.add(rd);
  rd=new idealgasData("oksijen","O2",31.999,25.48,1.52e-2,-0.7155e-5,1.312e-9);
  gi.add(rd);
  rd=new idealgasData("karbondioksit","CO2",44.01,22.26,5.981e-2,-3.501e-5,7.469e-9);
  gi.add(rd);
  rd=new idealgasData("karbonmonoksit","CO",28.011,28.16,0.1675e-2,0.5372e-5,-2.222e-9);
  gi.add(rd);
  rd=new idealgasData("hidrojen","H2",2.016,29.11,-0.1916e-2,0.4003e-5,-0.8704e-9);
  gi.add(rd);
  rd=new idealgasData("hava","0.21O2 0.79N2",28.97,28.11,0.1967e-2,0.4802e-5,-1.966e-9);
  gi.add(rd);
  rd=new idealgasData("su_buhari","H2O",18.015,32.24,0.1923e-2,1.055e-5,-3.595e-9);
  gi.add(rd);
  rd=new idealgasData("azotmonoksit","NO",30.006,29.34,-0.09395e-2,0.9747e-5,-4.187e-9);
  gi.add(rd);
  rd = new idealgasData("metan","CH4",16.043,19.89,5.024e-2,1.269e-5,-11.01e-9);
  gi.add(rd);
  rd = new idealgasData("etan","C2H6",30.07,6.90,17.27e-2,-6.406e-5,7.285e-9);
  gi.add(rd);
  rd = new idealgasData("propan","C3H8",44.097,-4.04,30.48e-2,-15.72e-5,31.74e-9);
  gi.add(rd);
  rd = new idealgasData("n-butan","C4H10",58.124,3.96,37.15e-2,-18.34e-5,35e-9);
  gi.add(rd);
  rd = new idealgasData("n-pentan","C5H12",72.151,6.774,45.43e-2,-22.46e-5,42.29e-9);
  gi.add(rd);
  rd = new idealgasData("n-hekzan","C6H14",86.178,6.938,55.22e-2,-28.65e-5,57.69e-9);
  gi.add(rd);
  rd = new idealgasData("diazotoksit","N2O",44.013,22.9,5.715e-2,-3.52e-5,7.87e-9);
  gi.add(rd);
  rd = new idealgasData("amonyak","NH3",17.03,27.568,2.563e-2,0.99072e-5,-6.6909e-9);
  gi.add(rd);
  rd = new idealgasData("kükürt","S",32.064,27.21,2.218e-2,-1.628e-2,3.986e-9);
  gi.add(rd);
}

```

```

rd = new idealgasData("kükürtdioksit","SO2",64.063,25.78,5.795e-2,-3.812e-5,8.612e-9);
gi.add(rd);
rd = new idealgasData("kükürttrioksit","SO3",80.0625,16.40,14.58e-2,-11.20e-5,32.42e-9);
gi.add(rd);
rd = new idealgasData("asetilen","C2H2",26.038,21.8,9.2143e-2,-6.527e-5,18.21e-9);
gi.add(rd);
rd = new idealgasData("etilen","C2H4",28.054,3.95,15.64e-2,-8.344e-5,17.67e-9);
gi.add(rd);
rd = new idealgasData("propilen","C3H6",42.081,3.15,23.83e-2,-12.18e-5,24.62e-9);
gi.add(rd);
rd = new idealgasData("benzen","C6H6",78.115,-36.22,48.475e-2,-31.57e-5,77.62e-9);
gi.add(rd);
rd = new idealgasData("metil_alkol","CH3OH",32.042,19.0,9.152e-2,-1.22e-5,-8.039e-9);
gi.add(rd);
rd = new idealgasData("etil_alkol","C2H5OH",46.07,19.9,20.96e-2,-10.38e-5,20.05e-9);
gi.add(rd);
rd = new idealgasData("hidrojenklorür","HCl",36.461,30.33,-0.762e-2,1.327e-5,-4.338e-9);
gi.add(rd);
int i;
Iterator n1=gi.iterator();
while(n1.hasNext())
{ rd=(idealgasData)n1.next();
s+=rd.name+" ";
if ((rd.name.equals(ig_name)) || (rd.formula.equals(ig_name))) {break;}
}
return s;
}

```

```

public String[][] output(double t,double P)
{double a[]=new double[6];
String b[][]=new String[6][3];
a[0]=+t;
a[1]=P;
a[2]=Cp(t)/rd.M;
a[3]=h(t)/rd.M;
a[4]=s0(t)/rd.M;
a[5]=s(t,P)/rd.M;
b[0][0]=" sıcaklık, t ";
b[1][0]=" basınç, P ";
b[2][0]=" sabit basınçta özgül ısı, Cp ";
b[3][0]=" entalpi h(t) ";
b[4][0]=" entropi s0(t) ";
b[5][0]=" entropi s(t,P) ";
b[0][2]=" derece C";
for(int i=0;i<a.length;i++) b[i][1]=""+a[i];
b[1][2]=" bar";
b[2][2]=" KJ/kg K";
b[3][2]=" KJ/kg";
b[4][2]=" KJ/kg K";
b[5][2]=" KJ/kg K";
return b;
}
}

```

İdeal gaz sınıfını test etme amacıyla bir test programı hazırlanmıştır.

10.10.3 idealgastest sınıfı

```
public class idealgastest
```



```

{
public static void main(String arg[])
{
idealgas O2=new idealgas("hava");
System.out.println(O2.rd.toString());

for(double t=(270-273);t<=100;t+=10)
{System.out.println("t = "+t+" Cp = "+
O2.Cp(t)+"h = "+O2.h(t)+" s = "+O2.s(t,1.0));}
String s[]=O2.idealgaslist();

System.out.println("Listedeki gaz isimleri : ");
for(int i=0;i<s.length;i++) System.out.println(s[i]);
}
}

```

----- Capture Output -----

```

> "D:\java\bin\javaw.exe" idealgastest
ideal gaz ismi = hava
ideal gaz formülü = 0.21O2 0.79N2
ideal gaz moleküler ağırlığı = 28.97 kg/kmol
Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T katsayıları
A[0] = 28.11A[1] = 0.001967A[2] = 4.802E-6A[3] = -1.966E-9
t = -3.0 Cp = 28.953078611578015h = -812.3287606046259 s = -2.8610235100083043
t = 7.0 Cp = 28.99470829640096h = -522.5903559256143 s = -1.8078923370305282
t = 17.0 Cp = 29.036967916283917h = -232.43249489307325 s = -0.7902301629983612
t = 27.0 Cp = 29.079845675226867h = 58.15106286359728 s = 0.19438833312726914
t = 37.0 Cp = 29.123329777229813h = 349.1664397549974 s = 1.1481495568466065
t = 47.0 Cp = 29.167408426292766h = 640.6196402317267 s = 2.0730313357801218
t = 57.0 Cp = 29.212069826415714h = 932.516550784386 s = 2.9708285897503064
t = 67.0 Cp = 29.257302181598664h = 1224.8629399435747 s = 3.843175169374883
t = 77.0 Cp = 29.303093695841614h = 1517.6644582798926 s = 4.691562529586142
t = 87.0 Cp = 29.349432573144565h = 1810.9266384039404 s = 5.517355773063849
t = 97.0 Cp = 29.396307017507517h = 2104.6548949663174 s = 6.321807495953667
Listedeki gaz isimleri :
azot
oksijen
karbondioksit
karbonmonoksit
hidrojen
hava
su_buharı
azotmonoksit
metan
etan
propan
n-butan
n-pentan
n-hekzan
diazotoksit
amonyak
kükürt
kükürtdioksit
kükürttrioksit
asetilen
etilen
propilen
benzen

```

```
metil_alkol
etil_alkol
hidrojenklorür
```

```
> Terminated with exit code 0.
```

Programımız çalışıyor ve sonuçları alıyor. Genellikle bir programı çalıştırdıktan sonraki ilk adım yukarıdaki gibi bir test programıyla test etmektir. Şimdi programın çalıştığını gördüğümüze göre sonuçları grafik ortamında bize aktaracak grafik çıktı ortamını oluşturalım. İlk programımız sonuçları tablo formuna getirmekte kullanacağımız idealgasModel sınıfı, ikincisi de grafik çıktı sınıfı idealgasTablosu olacaktır.

Now it is seen that class is working properly, a human graphical user interface can be prepared for a better output. Two classes will be prepared. The first class idealgasModel is prepared to get table output format. The main output program is idealgasTable

10.10.4 idealgasModel class

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.event.*;

class idealgasModel extends AbstractTableModel
{
    Object[][] veri;
    String[] baslik={"Name","Value","Units"};
    idealgas g;

    public idealgasModel()
    {
        g=new idealgas("air");
        veri=g.output(25.0,1.01325);
    }
    public idealgasModel(String idealgasname,double t,double P)
    {setValues(idealgasname,t,P);}
    public idealgasModel(idealgas gi,double t,double P)
    { setValues(gi,t,P);}
    public int getRowCount() {return veri.length;}
    public int getColumnCount() {return baslik.length;}
    public Object getValueAt(int satir,int sutun) {return veri[satir][sutun];}
    public String getColumnName(int c) {return baslik[c];}
    public void setValueAt(Object val, int row, int col)
    {
        veri[row][col] = val;
    }

    public void setValues(String idealgasname,double t,double P)
    {
        g=new idealgas(idealgasname);
        veri=g.output(t,P);
    }
    public void setValues(idealgas g1,double t,double P)
    {
        g=g1;
        veri=g.output(t,P);
    }
}
```

```
public boolean isCellEditable(int row, int col) {return true;}  
}
```

10.10.5 idealgasTable class

```
import java.awt.event.*;  
import java.util.*;  
import javax.swing.*;  
import javax.swing.table.*;  
  
public class idealgazTablosu extends JApplet implements ActionListener,ItemListener,Serializable  
{ private static final long serialVersionUID = 1L;  
  JLabel l1[]=new JLabel[3];    // Label  
  JLabel l2[]=new JLabel[3];    // Label  
  JTextField t1[]=new JTextField[2]; // giriş penceresi  
  JComboBox<String> c1; // gaz seçme alanı  
  JTextArea outputTextArea;  
  idealgasModel gm;  
  public idealgas g1;  
  JTable jt;  
  double T;  
  double P;  
  public String gazismi;  
  protected StringTokenizer token;  
  String st[];  
  Color bg;  
  Color fg;  
  JPanel inputPanel;  
  JPanel outputPanel;  
  JPanel totalPanel;  
  JScrollPane panel1;  
  public void init()  
  {  
    Container c=getContentPane();  
    c.setLayout(new FlowLayout());  
    bg=c.getBackground();  
    fg=c.getForeground();  
    gazismi=new String("N2");  
    T=25.0;  
    P=1.0;  
    inputPanel=new JPanel();  
    inputPanel.setLayout(new GridLayout(3,3,80,1));  
    outputPanel=new JPanel();  
    outputPanel.setLayout(new BorderLayout());  
    totalPanel=new JPanel();  
    totalPanel.setLayout(new BorderLayout());  
    g1=new idealgas(gazismi);  
    String s1=g1.idealgasInput("null");  
    token=new StringTokenizer(s1);  
    st=new String[token.countTokens()];  
    int i=0;  
    while(token.hasMoreTokens())  
    {st[i++]=new String(token.nextToken());}  
    c1=new JComboBox<String>(st);  
    t1[0] = new JTextField();  
    t1[1] = new JTextField();  
    g1=new idealgas(gazismi);
```

```

c1.setSelectedItem(gazismi);
l1[0]=new JLabel("gaz ismi ");
l1[1]=new JLabel("sicaklık ");
l1[2]=new JLabel("basinç ");
l2[0]=new JLabel(" ");
l2[1]=new JLabel(" derece C ");
l2[2]=new JLabel(" bar ");
outputTextArea=new JTextArea();
Font fn=l1[0].getFont();
for(i=0;i<2;i++)
{ l1[i].setBackground(Color.lightGray);
  l2[i].setBackground(Color.lightGray);
}
t1[0]=new JTextField(""+T);
t1[1]=new JTextField(""+P);
for(i=0;i<2;i++)
{ t1[i].setBackground(bg);t1[i].setFont(fn); }
c1.setBackground(bg);
outputTextArea.setBackground(bg);
outputTextArea.setForeground(fg);

outputTextArea.setFont(fn);

inputPanel.add(l1[0]);
inputPanel.add(c1);
inputPanel.add(l2[0]);

inputPanel.add(l1[1]);
inputPanel.add(t1[0]);
inputPanel.add(l2[1]);

inputPanel.add(l1[2]);
inputPanel.add(t1[1]);
inputPanel.add(l2[2]);

c1.addItemListener(this);
t1[0].addActionListener(this);
t1[1].addActionListener(this);
double TK=T+273.0;
gm=new idealgasModel(g1,T,P);
jt=new JTable(gm);
jt.setBackground(bg);
jt.setFont(fn);
outputPanel.add(outputTextArea,BorderLayout.NORTH);
panel1=new JScrollPane(jt);
outputPanel.add(panel1,BorderLayout.SOUTH);
totalPanel.add(inputPanel,BorderLayout.NORTH);
totalPanel.add(outputPanel,BorderLayout.SOUTH);
c.add(totalPanel,BorderLayout.NORTH);
}
public void itemStateChanged(ItemEvent ev)
{
String st1=(String)c1.getSelectedItem();
gazismi=st1;
gm.setValues(gazismi,T,P);
repaint();
}

```

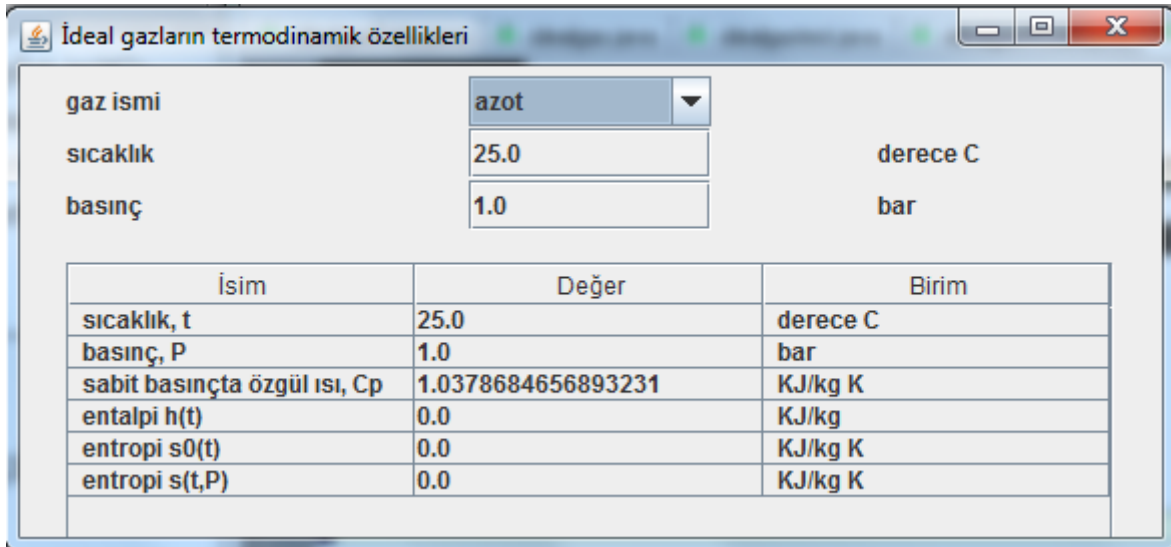
```

public void actionPerformed( ActionEvent e)
{
    Double valTemp=new Double(t1[0].getText());
    T=valTemp.doubleValue();
    Double valPressure=new Double(t1[1].getText());
    P=valPressure.doubleValue();
    gm.setValues(gazismi,T,P);
    repaint();
}

    public static void main(String s[]) {
        JFrame f= new JFrame("İdeal gazların termodinamik özellikleri ");
        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {System.exit(0);}
        });
        JApplet applet = new idealgazTablosu();
        f.getContentPane().add("Center", applet);
        applet.init();
        f.pack();
        f.setSize(new Dimension(600,400));
        f.setVisible(true);
    }
}

```

Sınıf çıktısı:



İsim	Değer	Birim
sıcaklık, t	25.0	derece C
basınç, P	1.0	bar
sabit basınçta özgül ısı, Cp	1.0378684656893231	KJ/kg K
entalpi h(t)	0.0	KJ/kg
entropi s0(t)	0.0	KJ/kg K
entropi s(t,P)	0.0	KJ/kg K

Programda kullanılan katsayılar Termodinamik kitabından direk olarak alınmıştır.

TABLE A – 1

Molar mass, gas constant, and critical-point properties

Substance	Formula	Molar mass, <i>M</i> kg/kmol	Gas constant, <i>R</i> kJ/kg · K*	Critical-point properties		
				Temperature, K	Pressure, MPa	Volume, m ³ /kmol
Air	—	28.97	0.2870	132.5	3.77	0.0883
Ammonia	NH ₃	17.03	0.4882	405.5	11.28	0.0724
Argon	Ar	39.948	0.2081	151	4.86	0.0749
Benzene	C ₆ H ₆	78.115	0.1064	562	4.92	0.2603
Bromine	Br ₂	159.808	0.0520	584	10.34	0.1355
<i>n</i> -Butane	C ₄ H ₁₀	58.124	0.1430	425.2	3.80	0.2547
Carbon dioxide	CO ₂	44.01	0.1889	304.2	7.39	0.0943
Carbon monoxide	CO	28.011	0.2968	133	3.50	0.0930
Carbon tetrachloride	CCl ₄	153.82	0.05405	556.4	4.56	0.2759
Chlorine	Cl ₂	70.906	0.1173	417	7.71	0.1242
Chloroform	CHCl ₃	119.38	0.06964	536.6	5.47	0.2403
Dichlorodifluoromethane (R-12)	CCl ₂ F ₂	120.91	0.06876	384.7	4.01	0.2179
Dichlorofluoromethane (R-21)	CHCl ₂ F	102.92	0.08078	451.7	5.17	0.1973
Ethane	C ₂ H ₆	30.070	0.2765	305.5	4.48	0.1480
Ethyl alcohol	C ₂ H ₅ OH	46.07	0.1805	516	6.38	0.1673
Ethylene	C ₂ H ₄	28.054	0.2964	282.4	5.12	0.1242
Helium	He	4.003	2.0769	5.3	0.23	0.0578
<i>n</i> -Hexane	C ₆ H ₁₄	86.179	0.09647	507.9	3.03	0.3677
Hydrogen (normal)	H ₂	2.016	4.1240	33.3	1.30	0.0649
Krypton	Kr	83.80	0.09921	209.4	5.50	0.0924
Methane	CH ₄	16.043	0.5182	191.1	4.64	0.0993
Methyl alcohol	CH ₃ OH	32.042	0.2595	513.2	7.95	0.1180
Methyl chloride	CH ₃ Cl	50.488	0.1647	416.3	6.68	0.1430
Neon	Ne	20.183	0.4119	44.5	2.73	0.0417
Nitrogen	N ₂	28.013	0.2968	126.2	3.39	0.0899
Nitrous oxide	N ₂ O	44.013	0.1889	309.7	7.27	0.0961
Oxygen	O ₂	31.999	0.2598	154.8	5.08	0.0780
Propane	C ₃ H ₈	44.097	0.1885	370	4.26	0.1998
Propylene	C ₃ H ₆	42.081	0.1976	365	4.62	0.1810
Sulfur dioxide	SO ₂	64.063	0.1298	430.7	7.88	0.1217
Tetrafluoroethane (R-134a)	CF ₃ CH ₂ F	102.03	0.08149	374.2	4.059	0.1993
Trichlorofluoromethane (R-11)	CCl ₃ F	137.37	0.06052	471.2	4.38	0.2478
Water	H ₂ O	18.015	0.4615	647.1	22.06	0.0560
Xenon	Xe	131.30	0.06332	289.8	5.88	0.1186

*The unit kJ/kg · K is equivalent to kPa · m³/kg · K. The gas constant is calculated from $R = R_u/M$, where $R_u = 8.31447$ kJ/kmol · K and M is the molar mass.

Source: K. A. Kobe and R. E. Lynn, Jr., *Chemical Review* 52 (1953), pp. 117–236; and ASHRAE, *Handbook of Fundamentals* (Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1993), pp. 16.4 and 36.1.

TABLE A-2

Ideal-gas specific heats of various common gases (Concluded)

(c) As a function of temperature

$$\bar{c}_p = a + bT + cT^2 + dT^3$$

(T in K, c_p in kJ/kmol · K)

Substance	Formula	a	b	c	d	Temperature range, K	% error	
							Max.	Avg.
Nitrogen	N ₂	28.90	-0.1571 × 10 ⁻²	0.8081 × 10 ⁻⁵	-2.873 × 10 ⁻⁹	273–1800	0.59	0.34
Oxygen	O ₂	25.48	1.520 × 10 ⁻²	-0.7155 × 10 ⁻⁵	1.312 × 10 ⁻⁹	273–1800	1.19	0.28
Air	—	28.11	0.1967 × 10 ⁻²	0.4802 × 10 ⁻⁵	-1.966 × 10 ⁻⁹	273–1800	0.72	0.33
Hydrogen	H ₂	29.11	-0.1916 × 10 ⁻²	0.4003 × 10 ⁻⁵	-0.8704 × 10 ⁻⁹	273–1800	1.01	0.26
Carbon monoxide	CO	28.16	0.1675 × 10 ⁻²	0.5372 × 10 ⁻⁵	-2.222 × 10 ⁻⁹	273–1800	0.89	0.37
Carbon dioxide	CO ₂	22.26	5.981 × 10 ⁻²	-3.501 × 10 ⁻⁵	7.469 × 10 ⁻⁹	273–1800	0.67	0.22
Water vapor	H ₂ O	32.24	0.1923 × 10 ⁻²	1.055 × 10 ⁻⁵	-3.595 × 10 ⁻⁹	273–1800	0.53	0.24
Nitric oxide	NO	29.34	-0.09395 × 10 ⁻²	0.9747 × 10 ⁻⁵	-4.187 × 10 ⁻⁹	273–1500	0.97	0.36
Nitrous oxide	N ₂ O	24.11	5.8632 × 10 ⁻²	-3.562 × 10 ⁻⁵	10.58 × 10 ⁻⁹	273–1500	0.59	0.26
Nitrogen dioxide	NO ₂	22.9	5.715 × 10 ⁻²	-3.52 × 10 ⁻⁵	7.87 × 10 ⁻⁹	273–1500	0.46	0.18
Ammonia	NH ₃	27.568	2.5630 × 10 ⁻²	0.99072 × 10 ⁻⁵	-6.6909 × 10 ⁻⁹	273–1500	0.91	0.36
Sulfur	S ₂	27.21	2.218 × 10 ⁻²	-1.628 × 10 ⁻⁵	3.986 × 10 ⁻⁹	273–1800	0.99	0.38
Sulfur dioxide	SO ₂	25.78	5.795 × 10 ⁻²	-3.812 × 10 ⁻⁵	8.612 × 10 ⁻⁹	273–1800	0.45	0.24
Sulfur trioxide	SO ₃	16.40	14.58 × 10 ⁻²	-11.20 × 10 ⁻⁵	32.42 × 10 ⁻⁹	273–1300	0.29	0.13
Acetylene	C ₂ H ₂	21.8	9.2143 × 10 ⁻²	-6.527 × 10 ⁻⁵	18.21 × 10 ⁻⁹	273–1500	1.46	0.59
Benzene	C ₆ H ₆	-36.22	48.475 × 10 ⁻²	-31.57 × 10 ⁻⁵	77.62 × 10 ⁻⁹	273–1500	0.34	0.20
Methanol	CH ₄ O	19.0	9.152 × 10 ⁻²	-1.22 × 10 ⁻⁵	-8.039 × 10 ⁻⁹	273–1000	0.18	0.08
Ethanol	C ₂ H ₆ O	19.9	20.96 × 10 ⁻²	-10.38 × 10 ⁻⁵	20.05 × 10 ⁻⁹	273–1500	0.40	0.22
Hydrogen chloride	HCl	30.33	-0.7620 × 10 ⁻²	1.327 × 10 ⁻⁵	-4.338 × 10 ⁻⁹	273–1500	0.22	0.08
Methane	CH ₄	19.89	5.024 × 10 ⁻²	1.269 × 10 ⁻⁵	-11.01 × 10 ⁻⁹	273–1500	1.33	0.57
Ethane	C ₂ H ₆	6.900	17.27 × 10 ⁻²	-6.406 × 10 ⁻⁵	7.285 × 10 ⁻⁹	273–1500	0.83	0.28
Propane	C ₃ H ₈	-4.04	30.48 × 10 ⁻²	-15.72 × 10 ⁻⁵	31.74 × 10 ⁻⁹	273–1500	0.40	0.12
n-Butane	C ₄ H ₁₀	3.96	37.15 × 10 ⁻²	-18.34 × 10 ⁻⁵	35.00 × 10 ⁻⁹	273–1500	0.54	0.24
i-Butane	C ₄ H ₁₀	-7.913	41.60 × 10 ⁻²	-23.01 × 10 ⁻⁵	49.91 × 10 ⁻⁹	273–1500	0.25	0.13
n-Pentane	C ₅ H ₁₂	6.774	45.43 × 10 ⁻²	-22.46 × 10 ⁻⁵	42.29 × 10 ⁻⁹	273–1500	0.56	0.21
n-Hexane	C ₆ H ₁₄	6.938	55.22 × 10 ⁻²	-28.65 × 10 ⁻⁵	57.69 × 10 ⁻⁹	273–1500	0.72	0.20
Ethylene	C ₂ H ₄	3.95	15.64 × 10 ⁻²	-8.344 × 10 ⁻⁵	17.67 × 10 ⁻⁹	273–1500	0.54	0.13
Propylene	C ₃ H ₆	3.15	23.83 × 10 ⁻²	-12.18 × 10 ⁻⁵	24.62 × 10 ⁻⁹	273–1500	0.73	0.17

Source: B. G. Kyle, *Chemical and Process Thermodynamics* (Englewood Cliffs, NJ: Prentice-Hall, 1984). Used with permission.

10.11 ALIŞTIRMALAR

1) Altındaki StringTokenizer2 sınıfı eksik olarak verilmiştir. Sınıfa bir String olarak bir cümle girin (String boşluklarla ayrılmış kelimelerden oluşsun örneğin: "evvel zaman içinde kalbur saman içinde ben babam beşiğini tıngır5 mıngır sallarıken") sonra **public static String[] enterStringarray(String s)** metodunu kullanarak kelimelere bölün, sonucu yazdırın.

```
import java.io.*;

import java.util.*;
import javax.swing.*;

public class StringTokenizerTest2
{
    public static String[] enterStringarray(String s)
    {
        String s1=JOptionPane.showInputDialog(s);
        StringTokenizer token=new StringTokenizer(s1);
        int n=token.countTokens()-1;
    }
}
```

```
int m=n+1;
String a[]=new String[m];
int j=0;
while(token.hasMoreTokens())
{a[j++]=token.nextToken();}
return a;
}

public static void main(String arg[])
{
}
}
```

2) StringBuffer sınıfının alt metodlarından biri replace metodudur. Bu metod başlangıç ile bitiş arasındaki kısmı değiştirir.

[StringBuffer](#) [replace](#)(int başlangıç, int bitiş, [String](#) str)

Replaces the characters in a substring of this sequence with characters in the specified String.

String s="evvel zaman içinde kalbur saman içinde ben babam beşiğini tıngır5 mıngır sallarken"

Tanımını girip her boşluğu * karakteriyle değiştirin. Bunun için yeni bir metod yazın

```
public static String bos_karekteri_yildizla_degistir(String girdi)
{
}
}
```

Çıktıyı yazdırın.

3) HashMap2.java sınıfını geliştirin. Bu sınıfta öğrenci numaralarını ve öğrenci isimlerini iki String boyutlu değişkeni olarak girin ve HashMap sınıfına yükleyin. Sonra öğrenci numarasını vererek öğrenci ismini listeden okuyun.

4) 3 nolu eksersizizi TreeNMap sınıfı kullanarak yapın.

10.12 ÖDEVLER

1) Alttaki yazı verilmiştir :

String s= "Java programı **isim.java** seklinde editörle yazılır. Java programınız yazıldıktan sonra ilk step java derleyici (compiler) programıyla **isim.class** ara bilgisayar kodunu oluşturmaktır.

Java programını yazarken dikkat etmeniz gereken diğer bir özellik de javanın nesne sınıfının ismiyle dosyadaki isminin aynı olmasıdır. Bilgisayar konsol ekranında (DOS veya UNIX Console)

javac isim.java

Komutunu bilgisayara gönderdiğinizde programınızda hata yoksa ve bilgisayarınız java komutunun yerini biliyorsa (path deyimi) java derleyicisi **isim.class** dosyasını yaratır. Eğer programınızda hata varsa hataların listesini verir. Programlamanın dili ingilizce olduğundan bu hata açıklamaları da ingilizcedir. Editör programına geri dönüp hataları düzeltmeniz, programı tekrar kaydetmeniz ve sonra bu komutu tekrar denemeniz gerekecektir. Eğer derlerken sınıf isimleri ve değişken isimlerini Türkçe olarak kullanabilmek istiyorsak üstteki deyimi"

Bir program yazarak kelime sayısını , ortalama kelime (harf) uzunluğunu ve maksimum kelime uzunluğunu hesaplatılarak yazdırınız.

2) Ödev 1 de verilen String’i alınız. String tipi boyutlu değişkenlere tüm kelimeleri yükledikten sonra ArrayList dinamik değişkenine yükleyiniz. Kelimeleri sıraya soktukten sonra yazdırınız.

3) Aşağıda otel programı verilmiştir. Bu programda otel konukları boyutlu değişkende tutulmaktadır. **double odadaki_konuk_sayisi []**; Bu yapı yerine **HashMap<Integer,Integer>** kullanan yeni bir versiyon geliştiriniz, burada ilk Integer otel oda numarası ikinci Integer odada kalan kişi sayısını belirlesin.

```
public class otel
{
String otel_ismi;
int oda_sayisi;
double odadaki_konuk_sayisi[];

public otel(String i_otel_ismi,int i_oda_sayisi)
{ otel_ismi=i_otel_ismi;
oda_sayisi=i_oda_sayisi;
odadaki_konuk_sayisi=new double[oda_sayisi];
}

public otel(String i_otel_ismi,double i_odadaki_konuk_sayisi[])
{ otel_ismi=i_otel_ismi;
oda_sayisi=i_odadaki_konuk_sayisi.length;
odadaki_konuk_sayisi=new double[oda_sayisi];
for(int oda_numarasi=0;oda_numarasi<oda_sayisi;oda_numarasi++)
{odadaki_konuk_sayisi[oda_numarasi]=i_odadaki_konuk_sayisi[oda_numarasi];}
}

public void oda_kaydi(int oda_numarasi,int konuk_sayisi)
{ odadaki_konuk_sayisi[oda_numarasi]=konuk_sayisi;}

public void oda_kaydi(double i_odadaki_konuk_sayisi[])
{ int n=i_odadaki_konuk_sayisi.length;
for(int oda_numarasi=0;oda_numarasi<n;oda_numarasi++)
{odadaki_konuk_sayisi[oda_numarasi]=i_odadaki_konuk_sayisi[oda_numarasi];}
}

public int oteldeki_konuk_sayisi()
{ int nog=0;
for(int oda_numarasi=0;oda_numarasi<oda_sayisi;oda_numarasi++)
{nog+=odadaki_konuk_sayisi[oda_numarasi];}
return nog;
}

public String toString()
{ String s="otel ismi : "+otel_ismi+"\n";
s+="oteldeki konuk sayısı : "+oteldeki_konuk_sayisi()+"\n";
for(int oda_numarasi=0;oda_numarasi<oda_sayisi;oda_numarasi++)
{s+="oda numarası = "+oda_numarasi+" odadaki konuk sayısı = "+odadaki_konuk_sayisi[oda_numarasi)+"\n";}
return s;
}

public String name(){return otel_ismi;}
}
```

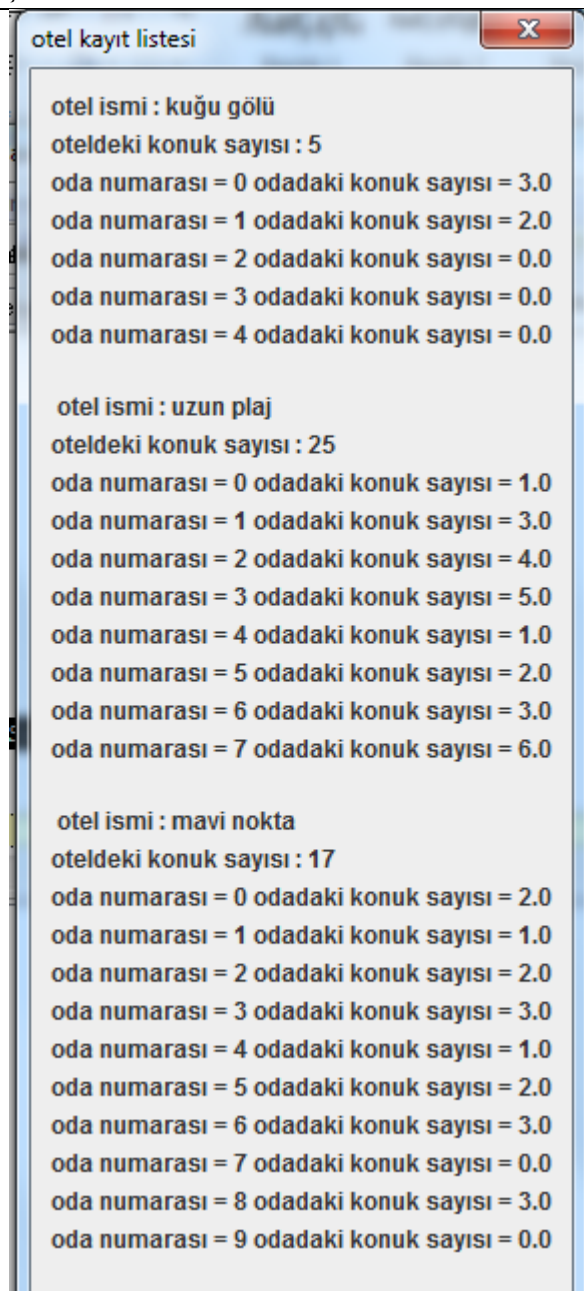
```
import javax.swing.*;

class oteltesti1
{
public static void main(String args[])
}
```

```

{
otel h1=new otel("kuğu gölü",5);
h1.oda_kaydi(0,3);
h1.oda_kaydi(1,2);
double a[]={ 1,3,4,5,1,2,3,6};
otel h2=new otel("uzun plaj",a);
otel h3=new otel("mavi nokta",10);
double b[]={2,1,2,3,1,2,3};
h3.oda_kaydi(b);
h3.oda_kaydi(8,3);
String s=h1.toString()+"\n "+
        h2.toString()+"\n "+
        h3.toString()+"\n ";
JOptionPane.showMessageDialog(null,s,
        "otel kayıt listesi",JOptionPane.PLAIN_MESSAGE);
System.exit(0);}
}

```



4) NavigableSet1.java programı verilmiştir. Bu listeden SERCAN BALTA dan daha küçük (harf sırası olarak) isimlerin listesini çıkarınız.

```
import java.util.*;
import javax.swing.*;
import java.util.concurrent.*;

public class NavigatableSet1
{
public static void main(String args[])
{
String student_name[]={ "MERT      KULA",
"İLYAS BAYRAKTAR",
"SAMET      YILDIZ",
"NURİ HARMANKAYA",
"GÜNAY      DÜRMÜŞ ",
"CAN ORAL ",
"EMRE ERDOĞAN ",
"ÇAĞLAR      ASARLI ",
"RAMAZAN ORAL",
"DOVLETGELDI ANNAGURBANOV",
"AHMET ULAŞ BAYKAL",
"MURAT      KARAKURT",
"SELMAN      ÜNAL",
"MERT KOÇER",
"BURAK CAN ÇELENK",
"MEHMET AKİF SAÇIKARA",
"AHMET      NALÇABASMAZ",
"AKİF ÖZEN",
"TANER      IŞIK",
"DİLEK AKTÜRK",
"MUHAMMET CANER MARANGOZ",
"MEHMET      ATEŞ",
"HİKMET      BAYOĞLU",
"SONER      AŞIK",
"MEHMET MURAT YARDIMCI",
"MEKAN      CHARIYEV",
"VURAL      ASLAN",
"Cihan AŞKIN",
"Volkan KORKUT",
"Selçuk MUTLU",
"Serdar DEMİR",
"Can UYSAL",
"Gürhan ÇOBAN",
"Erdal Şahin ACAR",
"Hüseyin DOĞANCAN",
"Ahmet Mert KILIÇ",
"Hüseyin ALSOY",
"Gözde BERKAY",
"Başak ŞİLEN",
"Burak MOKAN",
"Emel KIVRAK",
"Koray SARKAN",
"Uğur ÖZKAN",
"Yasin CAN",
"Bahtiyar ARSLAN",
"Mehmet SAĞIR",
"Onur GEZER",
```

```

"Kerem EREN",
"Oğuz ÇELİK",
"Hasan DEMİR",
"Yiğit Taylan YOLAÇAN",
"Mert ACAR",
"Barış AKDUR",
"Bahattar Gence SONYOL",
"Necip KİRAZCI",
"Anıl AVCIOĞLU",
"HASAN IŞIK",
"FERHAT ÖKMEN",
"EMİNE TÜRKECAN",
"MERVE HİLAL ÖZBELLİ",
"GÜVEN ONUR ALTIN",
"YAVUZ DURSUN",
"ALİ KIRAL",
"ALİCAN GÜLİŞLİ",
"ARDA İNÇELİ",
"SERCAN GÜRSEL",
"BUĞRAHAN ÇETİNKALE",
"MERT KARACA",
"SEVİM, SİNEM DEDELEROĞLU",
"BERAT CAN ŞENEL",
"SERKAN EMRE ERFİDAN",
"İBRAHİM CÖMERT BARTO",
"ANIL ENSARİ",
"MUSTAFA OKAN DOĞAN",
"SEMİH ŞİMŞEK",
"NURSEDA YILDIRIM",
"EMİR DAĞ",
"SERCAN BALTA",
"GÖZDE ERSAL",
"HÜSEYİN BARIŞ KÜSÜLÜ",
"AYBÜKE ORTAÇ",
"BAŞAR SUDİ KARATAŞ",
"CUMHUR ÖZEKEN",
"ANIL BALIKÇI",
"MUSTAFA KAHVECİ",
"KORHAN KÖKSAL",
"ABDULLAH BUZ",
"BURAK TEMİZ",
"YALÇIN KAÇAR",
"HAKAN KARAASMA",
"BAHADIR ÖZGİRAY",
};
NavigableSet <String>nSet = new ConcurrentSkipListSet<String>();
for(int i=0;i<student_name.length;i++)
{nSet.add(student_name[i]);}
//add additional lines here
}
}

```

5) Termodinamikte Van der Waals hal denklemi:

$$P = \frac{a}{v^2} (P - b) - R$$

Olarak verilmiştir. Bu denklemde a ve b katsayılarıdır. Bu katsayılar gazların kritik özellikleri kullanılarak hesaplanabilir.

$$a = \frac{2RT_{critic}^2}{64P_{critic}} \quad b = \frac{RT_{critic}}{8P_{critic}}$$

Kritik özellikler yukardaki tablolarda verilmiştir. Basınç

$$P = \frac{RT}{(v-b)} - \frac{a}{v^2}$$

Eşitliğinden hesaplanabilir. Eğer özgül hacim v değeri hesaplanmak istenirse direk olarak hesaplayamayız, bunun için kök bulma yöntemlerini kullanmamız gerekir

$$f(v) = \frac{RT}{(v-b)} - \frac{a}{v^2} - P$$

Bu denklemin kökünü çözmek için örneğin Newton-Raphson yöntemini kullanabiliriz. Newton_raphson yönteminde ilk değer tahmininde ideal gaz özgül hacim

değeri kullanılabilir. $v = \frac{RT}{P}$

vanderWaals vevanderWaalsData programları sizin için oluşturulmuştur çalıştırınız ve izleyiniz.

```
public class vanderWaalsData extends Object
{
public String isim; //ideal gazın ismi
public String formul; //ideal gazın formülü
public double M; // ideal gaz molekül ağırlığı
public double A[]; // Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T
public double Tc; //gazın kritik sıcaklığı
public double Pc; //gazın kritik basıncı
public vanderWaalsData(String isimi,String formuli,double Mi,double a,double b,double c,double
d,double Tci,double Pci)
{isim=isimi;;
formul=formuli;
M=Mi;
A=new double[4];
A[0]=a;
A[1]=b;
A[2]=c;
A[3]=d;
```

```

Pc=Pci;
Tc=Tci;
}
public String toString()
{
String s="";
s+=" gaz ismi = "+isim+"\n";
s+=" gas formülü = "+formul+"\n";
s+=" gaz moleküler ağırlığı = "+M+" kg/kmol \n";
s+=" gaz kritik sıcaklığı = "+Tc+" K \n";
s+=" gaz kritik basıncı = "+Pc+" Pa \n";
s+="Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T katsayıları\n";
for(int i=0;i<4;i++)
s+="A["+i+"] = "+A[i];
return s;
}
}

```

```

import java.io.*;
import java.net.*;
import java.util.*;
import javax.swing.*;

public class vanderWaals
{
public vanderWaalsData rd;
double a,b; //van der Waals katsayıları
double R;

public vanderWaals(String igName)
{ vanderWaalsInput(igName);
R=8314.5/rd.M; //J/(kgK)
a=28.0*R*R*rd.Tc*rd.Tc/64.0/rd.Pc;
b=R*rd.Tc/8.0/rd.Pc;
}

public double P(double T,double v)
{ //P basınç Pa
//T derece K
//v m^3/kg
double PP=R*T/(v-b)-a/(v*v);
return PP;}

public double T(double P,double v)
{ // T derece K
// P Pa
double TT=(P+a/(v*v))*(v-b)/R;
}
}

```

```

return TT;
}

public double func(double T,double P,double v)
{
//kök hesabı için fonksiyon
return R*T/(v-b)-a/(v*v)-P;
}

public double dfunc(double T,double P,double v)
{
//kök hesabı fonksiyonunun türevi
return -R*T/((v-b)*(v-b))+2*a/(v*v*v);
}

public double newton(double T,double P)
{
//Newton-Raphson metodu
double v=R*T/P;
int nmax=100;
double tolerance=1.0e-10;
double fx,dfx;
for(int i=0;i<nmax;i++)
{
fx=func(T,P,v);
dfx=dfunc(T,P,v);
v-=fx/dfx;
//System.out.println("i="+i+"x="+x+"fx="+fx+"dfx = "+dfx);
if(Math.abs(fx)<tolerance) { return v;}
}
JOptionPane.showMessageDialog(null,"Uyarı: Maksimum iterasyon sayısı aşıldı \n"+
" özgül hacim v hesap sonucu doğru olmayabilir","MAKSİMUM İTERASYON SAYISI
UYARISI",JOptionPane.WARNING_MESSAGE);
return v;
}

public double v(double T,double P)
{
//özgül hacim hesabı
return newton(T,P);
}

public String[] vanderWaalslist()
{
String s1=rd.isim;String s2=vanderWaalsInput("null");vanderWaalsInput(s1);
StringTokenizer t=new StringTokenizer(s2);
int n=t.countTokens();
String s3[]=new String[n];
int i=0;
while(t.hasMoreTokens())
{
s3[i]=t.nextToken();
}
}

```

```

i++;
}
return s3;
}

public double Cp(double t)
{ //ideal gaz Cp değeri t derece C
double T=t+273.15; //derece K
return (rd.A[0]+rd.A[1]*T+rd.A[2]*T*T+rd.A[3]*T*T*T);
//KJ/kmol K
}

public String vanderWaalsInput(String ig_isim)
{
String s="";
ArrayList<vanderWaalsData> gi = new ArrayList<vanderWaalsData>();
rd = new vanderWaalsData("azot","N2",28.013,28.90,-0.1571e-2,0.8081e-5,-28.73e-
9,126.2,3.39e6);
gi.add(rd);
rd=new vanderWaalsData("oksijen","O2",31.999,25.48,1.52e-2,-0.7155e-5,1.312e-
9,154.8,5.08e6);
gi.add(rd);
rd=new vanderWaalsData("karbondioksit","CO2",44.01,22.26,5.981e-2,-3.501e-5,7.469e-
9,304.2,7.39e6);
gi.add(rd);
int i;
Iterator n1=gi.iterator();
while(n1.hasNext())
{ rd=(vanderWaalsData)n1.next();
s+=rd.isim+" ";
if ((rd.isim.equals(ig_isim)) || (rd.formul.equals(ig_isim))) {break;}
}
return s;
}
}

```

```

public class vanderWaalstest
{
public static void main(String arg[])
{
vanderWaals O2=new vanderWaals("O2");
System.out.println(O2.rd.toString());
double P=1.01325e5;//Pa
String s[]=O2.vanderWaalslist();
System.out.println("Gaz isimleri listesi : ");
for(int i=0;i<s.length;i++) System.out.println(s[i]);
}
}

```



```
}
```

```
----- Capture Output -----  
> "D:\java\bin\javaw.exe" vanderWaalstest  
gaz ismi = oksijen  
gas formülü = O2  
gaz moleküler ağırlığı = 31.999 kg/kmol  
gaz kritik sıcaklığı = 154.8 K  
gaz kritik basıncı = 5080000.0 Pa  
Cp(T)=A[0]+A[1]*T+A[2]*T*T+A[3]*T*T*T katsayıları  
A[0] = 25.48A[1] = 0.0152A[2] = -7.155E-6A[3] = 1.312E-9  
Gaz isimleri listesi :  
azot  
oksijen  
karbondioksit  
  
> Terminated with exit code 0.
```

Bu programda sadece üç gaza yer verilmiş (oksijen, azot ve karbondioksit). Tablo1 ve2 de verilen diğer gazları da programa ilave ediniz.

public String vanderWaalsInput(String ig_isim) of class vanderWaals

Programı yeni gazlar için test ediniz

6) Bir test programı yazarak vanderWaals ve ideqal gaz denklemlerinin hesapladığı özgül ısı değerlerini karşılaştırınız, yüzde hatayı hesaplatınız.

11 DOSYADAN VERİ OKUMA VE YAZMA

11.1 FILE SINIFI

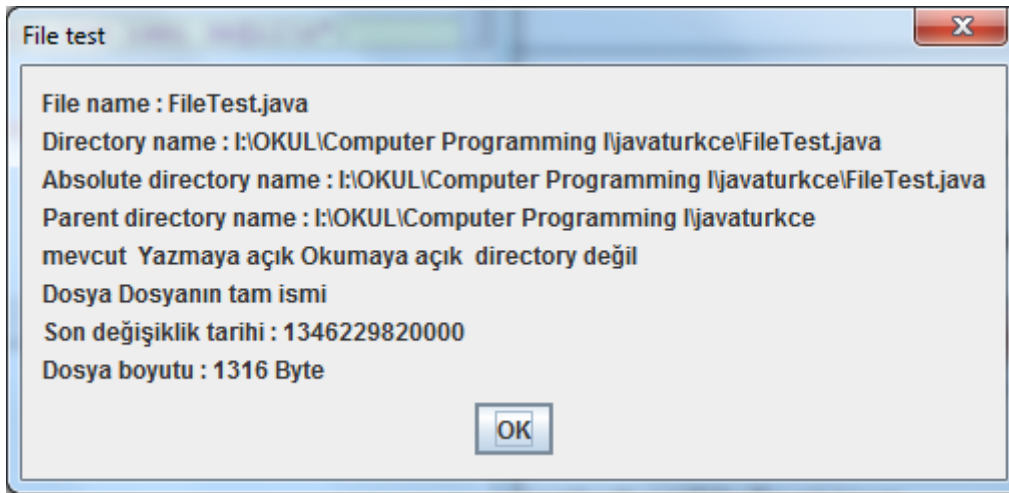
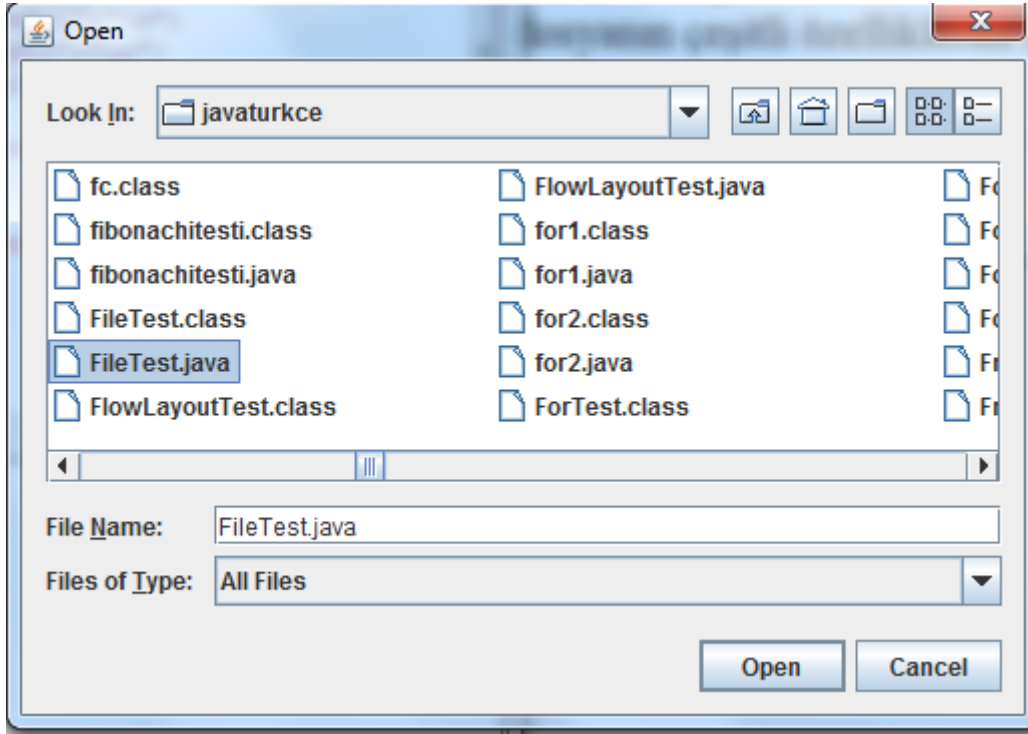
File sınıfı okunacak bir dosya ile ilgili detayları saptamak, işlemek ve daha sonra bu dosyadan işlem yapmak için hazırlık yapma amacıyla kullanılan bir sınıftır. Bir dosya File sınıfında tanımlandığında bu dosya ile ilgili çeşitli bilgilere ulaşabiliriz. Örneğin yazma-okuma müsaadelerini, dosya boyutunu görebiliriz. Bir dosyanın çeşitli özelliklerini File sınıfının metodları üzerinden listelediğimiz bir örnek program aşağıda verilmiştir.

11.1-1 FileTest programı

```
import java.awt.*;
import java.io.File;
import java.util.HashMap;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.filechooser.*;
import javax.swing.event.*;

public class FileTest
{
    public static void main(String args[])
    {
        //önce JFileChooser kullanarak dosyayı seç, sonra dosya özelliklerini göster.
        File f=new File("readfromfile1.java");
        JFileChooser dosyasec=new JFileChooser();
        int approval = dosyasec.showOpenDialog(null);
        if(approval==JFileChooser.APPROVE_OPTION)
        {
            f=dosyasec.getSelectedFile();
        }
        String s="";
        s+="File name : "+f.getName()+"\n";
        s+="Directory name : "+f.getPath()+"\n";
        s+="Absolute directory name : "+f.getAbsolutePath()+"\n";
        s+="Parent directory name : "+f.getParent()+"\n";
        s+=f.exists() ? "mevcut " : " mevcut değil "+"\n";
        s+=f.canWrite() ? "Yazmaya açık " : " Yazmaya açık değil"+"\n";
        s+=f.canRead() ? "Okumaya açık " : " Okumaya açık değil"+"\n";
        s+=f.isDirectory() ? "Directory " : " directory değil"+"\n";
        s+=f.isFile() ? "Dosya " : " Dosya değil"+"\n";
        s+=f.isAbsolute() ? "Dosyanın tam ismi " : " dosyanın tam ismi değil"+"\n";
        s+="Son değişiklik tarihi : "+f.lastModified()+"\n";
        s+="Dosya boyutu : "+f.length()+" Byte"+"\n";
        JOptionPane.showMessageDialog(null,s,"File test",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

Bu programda dosyayı seçmek için JFileChooser sınıfından yararlandık. Bu sınıf dosya seçmek için java kütüphanesinde bize hazır kullanım amacıyla sunulmuştur.



File sınıfının metodları aşağıda tablo olarak verilmiştir.

Kurucu metodlar

[File](#)([File](#) parent, [String](#) child)

[File](#)([String](#) pathname)

[File](#)([String](#) parent, [String](#) child)

[File](#)([URI](#) uri)

Method Summary

boolean [canExecute](#)()

boolean [canRead](#)()

boolean [canWrite](#)()

int [compareTo](#)([File](#) dosyayolu)

boolean [createNewFile](#)()

static [File](#) [createTempFile\(String](#) prefix, [String](#) suffix)
static [File](#) [createTempFile\(String](#) prefix, [String](#) suffix, [File](#) directory)
boolean [delete\(\)](#)
void [deleteOnExit\(\)](#)
boolean [equals\(Object](#) obj)
boolean [exists\(\)](#)
[File](#) [getAbsolutePath\(\)](#)
[String](#) [getAbsolutePath\(\)](#)
[File](#) [getCanonicalFile\(\)](#)
[String](#) [getCanonicalPath\(\)](#)
long [getFreeSpace\(\)](#)
[String](#) [getName\(\)](#)
[String](#) [getParent\(\)](#)
[File](#) [getParentFile\(\)](#)
[String](#) [getPath\(\)](#)
long [getTotalSpace\(\)](#)
long [getUsableSpace\(\)](#)
int [hashCode\(\)](#)
boolean [isAbsolute\(\)](#)
boolean [isDirectory\(\)](#)
boolean [isFile\(\)](#)
boolean [isHidden\(\)](#)
long [lastModified\(\)](#)
long [length\(\)](#)
[String\[\]](#) [list\(\)](#)
[String\[\]](#) [list\(FilenameFilter](#) filter)
[File\[\]](#) [listFiles\(\)](#)
[File\[\]](#) [listFiles\(FileFilter](#) filter)
[File\[\]](#) [listFiles\(FilenameFilter](#) filter)
static [File\[\]](#) [listRoots\(\)](#)
boolean [mkdir\(\)](#)
boolean [mkdirs\(\)](#)
boolean [renameTo\(File](#) dest)
boolean [setExecutable](#)(boolean executable)
boolean [setExecutable](#)(boolean executable, boolean ownerOnly)
boolean [setLastModified](#)(long time)
boolean [setReadable](#)(boolean readable)
boolean [setReadable](#)(boolean readable, boolean ownerOnly)
boolean [setReadOnly\(\)](#)
boolean [setWritable](#)(boolean writable)
boolean [setWritable](#)(boolean writable, boolean ownerOnly)
[String](#) [toString\(\)](#)

11.2 ARDIŞIK DOSYA OKUMA VE KANAL YAPISI

Ardışık dosya yapısı sıralı bir giriş ve yazım yapan bir yapıdır. Örneğin text tipi dosya yapısı, teyp tipi yazım ardışık (sıralı) yazım ve okuma ardışık sistemlere örnek olarak verilebilir. Dosyaların yazılması için kullanılan diğer bir metotta tesadüfi ulaşım (random Access) metodudur. Günümüzde bilgisayar hafızaları, CD gibi ortamlar bu metodu kullanırlar. Java'da dosyalara girdi çıktı yapabilmek için iki temel kütüphane tanımlanmıştır. Birincisi girdi çıktı kütüphanesi (input-output library- io) diğeri yeni girdi-çıkıtı kütüphanesidir (new input-output library-nio). Io, girdi-çıkıtı kütüphanesinde kanal okuma prosesleri için tanımlanan sınıflar şunlardır:

InputStream (girdi akışı sınıfı)

- FilterInputStream
 - LineNumberInputStream
 - BufferedInputStream
 - DataInputStream
 - PushbackInputStream
- ByteArrayInputStream
- FileInputStream
- ObjectInputStream
- PipedInputStream
- SequenceInputStream
- StringBufferInputStream

OutputStream (Çıktı akışı sınıfı)

- ByteArrayOutputStream
- FileOutputStream
- FilterOutputStream
 - BufferedOutputStream
 - DataOutputStream
 - PrintStream
- ObjectOutputStream
- PipedOutputStream

ObjectStreamClass

StreamTokenizer

Reader

- FileReader
- BufferedReader
 - LineNumberReader
- CharArrayReader
- FilterReader
 - PushbackReader
- InputStreamReader
 - FileReader
- PipedReader
- StringReader

Writer

- BufferedWriter
- CharArrayWriter
- FilterWriter
- OutputStreamWriter

PipedWriter
PrintWriter
StringWriter

Tüm bu sınıflar arasında biz en temel ardışık okuma kanalları olarak BufferedReader ve FileReader sınıflarını kullanacağız. Veri akışında kullandığımız sistemlere veri akış kanalı ismini veriyoruz. BufferedReader akış kanalının tanımı:

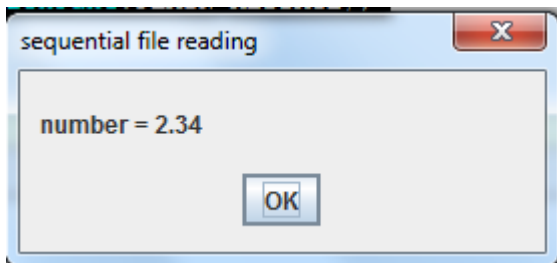
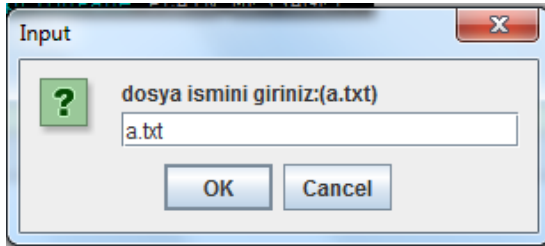
BufferedReader fin=new BufferedReader(new FileReader(f));

Şeklindedir. Bunu bir örnek programda kullanmaya çalışalım:

11.2-1 dosyadanoku sınıfı

```
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

public class dosyadanoku
{
public static void main(String args[]) throws IOException
{
String isim=JOptionPane.showInputDialog("dosya ismini giriniz:(a.txt)");
File f=new File(isim);
BufferedReader fin=new BufferedReader(new FileReader(f));
double number=Double.parseDouble(fin.readLine());
String s="number = "+number;
JOptionPane.showMessageDialog(null,s,"sequential file reading",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}
```



İkinci program da birincisinin benzeridir, ancak bu sınıfta dosya okumasını JFileChooser hazır grafik kütüphane sınıfını kullanarak yapmaktayız.

11.2-2 readfromfile1 class example

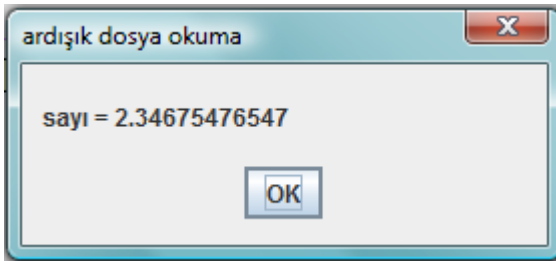
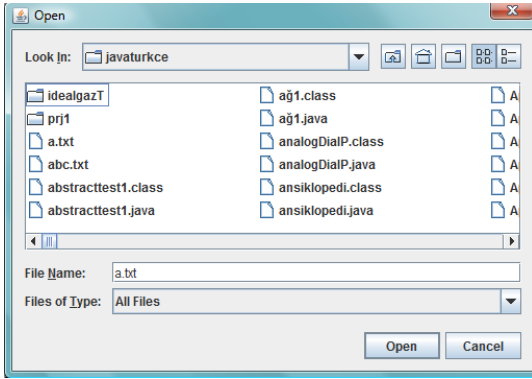
```
import java.io.*;
```

```

import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

public class dosyadanokul
{public static void main(String args[]) throws IOException
{
File f=new File("a.txt");
JFileChooser dosyasec=new JFileChooser();
int onay = dosyasec.showOpenDialog(null);
if(onay==JFileChooser.APPROVE_OPTION)
{f=dosyasec.getSelectedFile();}
BufferedReader fin=new BufferedReader(new FileReader(f));
double sayi=Double.parseDouble(fin.readLine());
String s="sayı = "+sayi;
JOptionPane.showMessageDialog(null,s,"ardışık dosya okuma",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}}

```



JFileChooser sınıfını kullanırken ExampleFileFilter sınıfı kullanılarak sadece istenen bir tip dosyanın görüntülenmesi sağlanabilir. Altta ExampleFileFilter sınıfı kodu ve bu sınıfı kullanan JOptionPane sınıfı görülmektedir.

11.2-3 ExampleFileFilter class

```

/*
 *
 *A file filter when displaying files in a file dialog. The specific use for this class
 *is to only display files with the .gal extension.
 *
 * @(#)ExampleFileFilter.java      1.16 04/07/26
 *
 * Copyright (c) 2004 Sun Microsystems, Inc. All Rights Reserved.
 *
 */

```

```

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*
* -Redistribution of source code must retain the above copyright notice, this
* list of conditions and the following disclaimer.
*
* -Redistribution in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
*
* Neither the name of Sun Microsystems, Inc. or the names of contributors may
* be used to endorse or promote products derived from this software without
* specific prior written permission.
*
* This software is provided "AS IS," without a warranty of any kind. ALL
* EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING
* ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MIDROSYSTEMS, INC. ("SUN")
* AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE
* AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS
* DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST
* REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL,
* INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY
* OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE,
* EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
* You acknowledge that this software is not designed, licensed or intended
* for use in the design, construction, operation or maintenance of any
* nuclear facility.
*/

/*
 * @(#)ExampleFileFilter.java      1.16 04/07/26
 */

import java.io.File;
import java.util.Hashtable;
import java.util.Enumeration;
import javax.swing.*.*;
import javax.swing.filechooser.*;

/**
 * A convenience implementation of FileFilter that filters out
 * all files except for those type extensions that it knows about.
 *
 * Extensions are of the type ".foo", which is typically found on
 * Windows and Unix boxes, but not on Macintosh. Case is ignored.
 *
 * Example - create a new filter that filters out all files
 * but gif and jpg image files:
 *
 * JFileChooser chooser = new JFileChooser();
 * ExampleFileFilter filter = new ExampleFileFilter(
 *     new String{ "gif", "jpg" }, "JPEG & GIF Images")
 * chooser.addChoosableFileFilter(filter);
 * chooser.showOpenDialog(this);
 */

```



```

* @version 1.16 07/26/04
* @author Jeff Dinkins
*/
public class ExampleFileFilter extends FileFilter {

    private static String TYPE_UNKNOWN = "Type Unknown";
    private static String HIDDEN_FILE = "Hidden File";

    private Hashtable filters = null;
    private String description = null;
    private String fullDescription = null;
    private boolean useExtensionsInDescription = true;

    /**
     * Creates a file filter. If no filters are added, then all
     * files are accepted.
     *
     * @see #addExtension
     */
    public ExampleFileFilter() {
        this.filters = new Hashtable();
    }

    /**
     * Creates a file filter that accepts files with the given extension.
     * Example: new ExampleFileFilter("jpg");
     *
     * @see #addExtension
     */
    public ExampleFileFilter(String extension) {
        this(extension,null);
    }

    /**
     * Creates a file filter that accepts the given file type.
     * Example: new ExampleFileFilter("jpg", "JPEG Image Images");
     *
     * Note that the "." before the extension is not needed. If
     * provided, it will be ignored.
     *
     * @see #addExtension
     */
    public ExampleFileFilter(String extension, String description) {
        this();
        if(extension!=null) addExtension(extension);
        if(description!=null) setDescription(description);
    }

    /**
     * Creates a file filter from the given string array.
     * Example: new ExampleFileFilter(String {"gif", "jpg"});
     *
     * Note that the "." before the extension is not needed and
     * will be ignored.
     *
     * @see #addExtension
     */
    public ExampleFileFilter(String[] filters) {

```

```

        this(filters, null);
    }

    /**
     * Creates a file filter from the given string array and description.
     * Example: new ExampleFileFilter(String {"gif", "jpg"}, "Gif and JPG Images");
     *
     * Note that the "." before the extension is not needed and will be ignored.
     *
     * @see #addExtension
     */
    public ExampleFileFilter(String[] filters, String description) {
        this();
        for (int i = 0; i < filters.length; i++) {
            // add filters one by one
            addExtension(filters[i]);
        }
        if(description!=null) setDescription(description);
    }

    /**
     * Return true if this file should be shown in the directory pane,
     * false if it shouldn't.
     *
     * Files that begin with "." are ignored.
     *
     * @see #getExtension
     * @see FileFilter#accepts
     */
    public boolean accept(File f) {
        if(f != null) {
            if(f.isDirectory()) {
                return true;
            }
            String extension = getExtension(f);
            if(extension != null && filters.get(getExtension(f)) != null) {
                return true;
            }
        }
        return false;
    }

    /**
     * Return the extension portion of the file's name .
     *
     * @see #getExtension
     * @see FileFilter#accept
     */
    public String getExtension(File f) {
        if(f != null) {
            String filename = f.getName();
            int i = filename.lastIndexOf('.');
            if(i>0 && i<filename.length()-1) {
                return filename.substring(i+1).toLowerCase();
            }
        }
        return null;
    }
}

```

```

/**
 * Adds a filetype "dot" extension to filter against.
 *
 * For example: the following code will create a filter that filters
 * out all files except those that end in ".jpg" and ".tif":
 *
 * ExampleFileFilter filter = new ExampleFileFilter();
 * filter.addExtension("jpg");
 * filter.addExtension("tif");
 *
 * Note that the "." before the extension is not needed and will be ignored.
 */
public void addExtension(String extension) {
    if(filters == null) {
        filters = new Hashtable(5);
    }
    filters.put(extension.toLowerCase(), this);
    fullDescription = null;
}

/**
 * Returns the human readable description of this filter. For
 * example: "JPEG and GIF Image Files (*.jpg, *.gif)"
 *
 * @see setDescription
 * @see setExtensionListInDescription
 * @see isExtensionListInDescription
 * @see FileFilter#getDescription
 */
public String getDescription() {
    if(fullDescription == null) {
        if(description == null || isExtensionListInDescription()) {
            fullDescription = description==null ? "(" : description + "(";
            // build the description from the extension list
            Enumeration extensions = filters.keys();
            if(extensions != null) {
                fullDescription += "." + (String) extensions.nextElement();
                while (extensions.hasMoreElements()) {
                    fullDescription += ", ." + (String) extensions.nextElement();
                }
            }
            fullDescription += ")";
        } else {
            fullDescription = description;
        }
    }
    return fullDescription;
}

/**
 * Sets the human readable description of this filter. For
 * example: filter.setDescription("Gif and JPG Images");
 *
 * @see setDescription
 * @see setExtensionListInDescription
 * @see isExtensionListInDescription

```

```

    */
    public void setDescription(String description) {
        this.description = description;
        fullDescription = null;
    }

    /**
     * Determines whether the extension list (.jpg, .gif, etc) should
     * show up in the human readable description.
     *
     * Only relevant if a description was provided in the constructor
     * or using setDescription();
     *
     * @see getDescription
     * @see setDescription
     * @see isExtensionListInDescription
     */
    public void setExtensionListInDescription(boolean b) {
        useExtensionsInDescription = b;
        fullDescription = null;
    }

    /**
     * Returns whether the extension list (.jpg, .gif, etc) should
     * show up in the human readable description.
     *
     * Only relevant if a description was provided in the constructor
     * or using setDescription();
     *
     * @see getDescription
     * @see setDescription
     * @see setExtensionListInDescription
     */
    public boolean isExtensionListInDescription() {
        return useExtensionsInDescription;
    }
}

```

11.2-4 readfromfile2 class example

```

    import java.io.*;
    import java.util.*;
    import javax.swing.*;
    import javax.swing.table.*;
    import java.awt.*;

    public class dosyadanoku2
    {

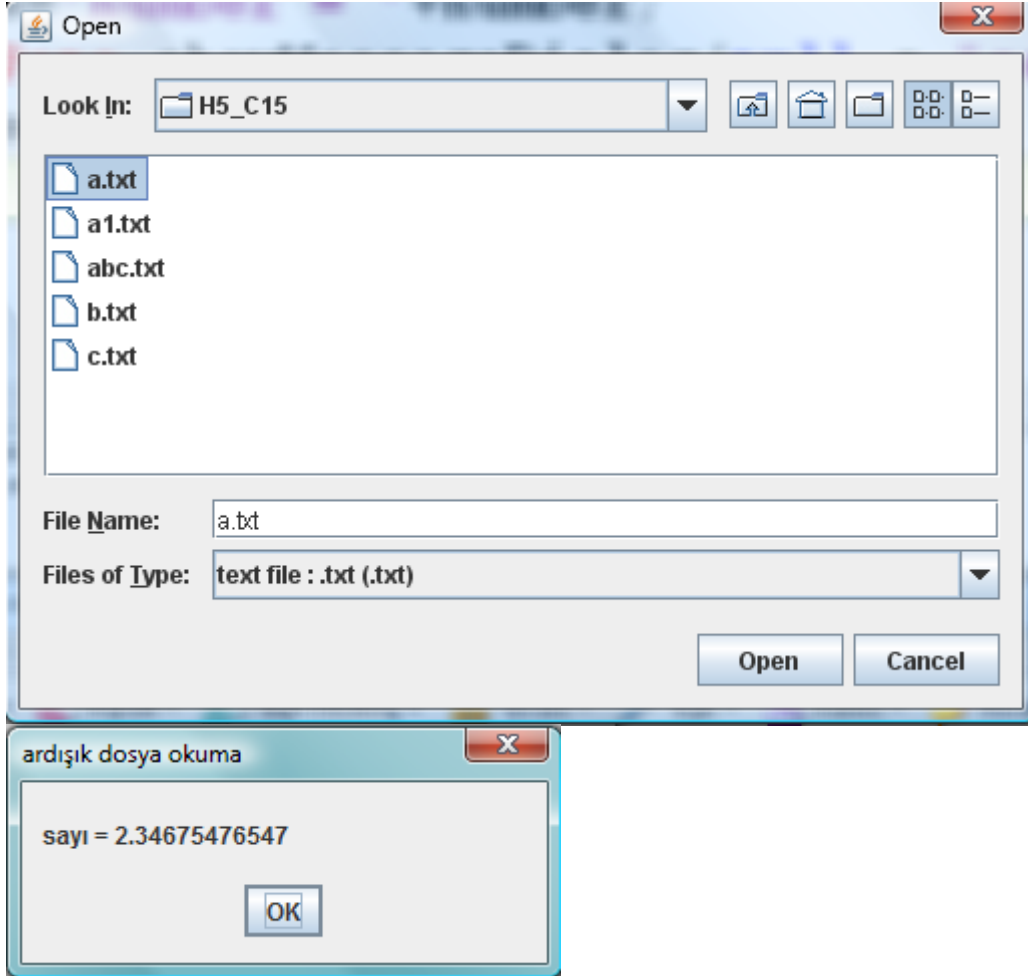
    public static void main(String args[]) throws IOException
    {
        String s1="a.txt";
        JFileChooser fc=new JFileChooser();
        ExampleFileFilter filter = new ExampleFileFilter();
        filter.addExtension("txt");
        filter.setDescription("text file : .txt");
    }
}

```

```

fc.setFileFilter(filter);
if (fc.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {File file = fc.getSelectedFile();s1=file.getName(); }
File f=new File(s1);
BufferedReader fin=new BufferedReader(new FileReader(f));
double sayi=Double.parseDouble(fin.readLine());
String s="sayi = "+sayi;
JOptionPane.showMessageDialog(null,s,"ardışık dosya okuma",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}

```



Bir sonraki örneğimizde bir sayı yerine bir sayı seti okumayı deneyeceğiz. Bunu gerçekleştirmek için bir satır halinde String değişkeni olarak okuduğumuz değerleri StringTokenizer sınıfını kullanarak böldükten sonra double değişkenine dönüştüreceğiz.

11.2-5 dosyadanoku3 class example

```

import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

public class dosyadanoku3
{

```

```

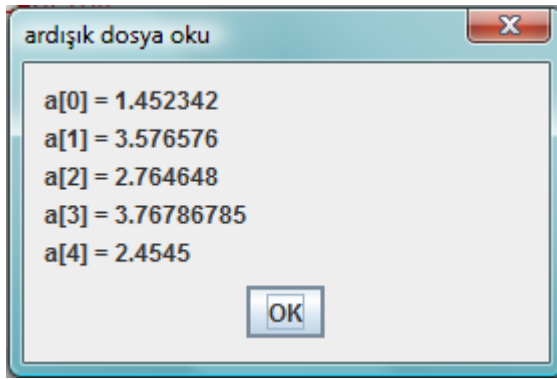
public static double[] doublegir(String filename) throws IOException
{
    //reading double values from one line of data
    BufferedReader fin=new BufferedReader(new FileReader(filename));
    double a[]=doublegir(fin);
    return a;
}

public static double[] doublegir(BufferedReader fin) throws IOException
{
    //reading double values from one line of data
    String s1=fin.readLine();
    StringTokenizer token=new StringTokenizer(s1);
    int n=token.countTokens()-1;
    int m=n+1;
    double a[]=new double[m];
    int j=0;
    while(token.hasMoreTokens())
    {
        Double ax=new Double(token.nextToken());
        a[j++]=ax.doubleValue();
    }
    return a;
}

public static String toString(double a[])
{
    String s="";
    for(int i=0;i<a.length;i++)
    {s+="a["+i+"] = "+a[i]+"\\n";}
    return s;
}

public static void main(String args[]) throws IOException
{
    String s1="a.txt";
    JFileChooser fc=new JFileChooser();
    ExampleFileFilter filter = new ExampleFileFilter();
    filter.addExtension("txt");
    filter.setDescription("text file : .txt");
    fc.setFileFilter(filter);
    if (fc.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {File file = fc.getSelectedFile();s1=file.getName(); }
    File f=new File(s1);
    BufferedReader fin=new BufferedReader(new FileReader(f));
    double number[]=doublegir(fin);
    JOptionPane.showMessageDialog(null,toString(number),"ardışık dosya oku",JOptionPane.PLAIN_MESSAGE);
    System.exit(0);
}
}

```



Girdi dosyası a.txt de veri alttaki gibi verilmiştir.

```
1.452342 3.576576 2.764648 3.76786785 2.4545
```

Bir sonraki örnekte bir matris okuyacağız. Matriste her sütunda kaç eleman olduğu StringTokenizer tarafından kontrol edilirken kaç satır olduğu **EOFException (dosyasonu)** hata kontrolü terimiyle (try-catch) kontrol edilir. Bu sayede bilinmeyen boyuttaki bir dosya rahatlıkla okunabilir.

11.2-6 dosyadanoku4

```
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

class data
{
    public double x[];
    public data(double[] xi)
    {
        int n=xi.length;
        x=new double[n];
        for(int i=0;i<n;i++) {x[i]=xi[i];}
    }
    public String toString() {return ""+x[0]+" "+x[1];}
}

public class readfromfile4
{
    public static double[][] inputdouble(String filename) throws IOException
    {
        //reading double values from one line of data
        BufferedReader fin=new BufferedReader(new FileReader(filename));
        double a[][]=inputdouble(fin);
        return a;
    }

    public static double[][] inputdouble(BufferedReader fin) throws IOException
    {
        ArrayList<data> a=new ArrayList<data>();
        int mmax=0;
        double b[][];
        int n=0,m=0;
```

```

        String s1="";
        try{
        while(fin!=null)
        { s1=fin.readLine();
        if(s1==null)break;
        StringTokenizer token=new StringTokenizer(s1);
        m=token.countTokens();
        if(m>mmax) mmax=m;
        double aa[]=new double[m];
        int j=0;
        while(token.hasMoreTokens())
        { Double ax=new Double(token.nextToken());
        aa[j++]=ax.doubleValue();
        }
        data vx=new data(aa);
        a.add(vx);
        n++;
        }
        } catch(EOFException e_eof)
        {
        //close ffile
        try{fin.close();
        }
        catch(NullPointerException e) {}
        catch(IOException e)
        {
        System.err.println("Error Closing File\n"+e.toString());
        System.exit(1);
        }
        } //End of EOFException
        b=new double[n][mmax];
        int k=0;
        Iterator i=a.iterator();
        while(i.hasNext()) { data vx=(data)i.next();b[k++]=vx.x;}
        return b;
        }

```

```

public static String toString(double a[][])
{
        String s="";
        for(int i=0;i<a.length;i++)
        {
                for(int j=0;j<a[0].length;j++)
                {s+="a["+i+"]"+"["+j+"] = "+a[i][j]+" ";}
                s+="\n";
        }
        return s;
}

```

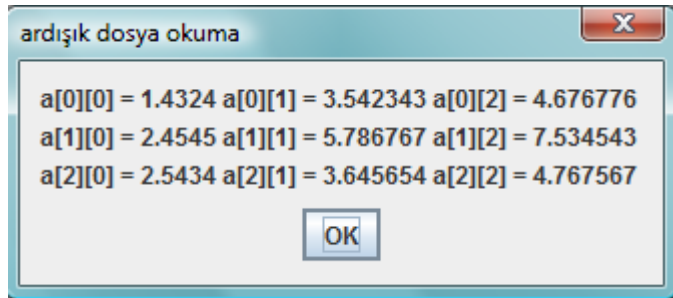
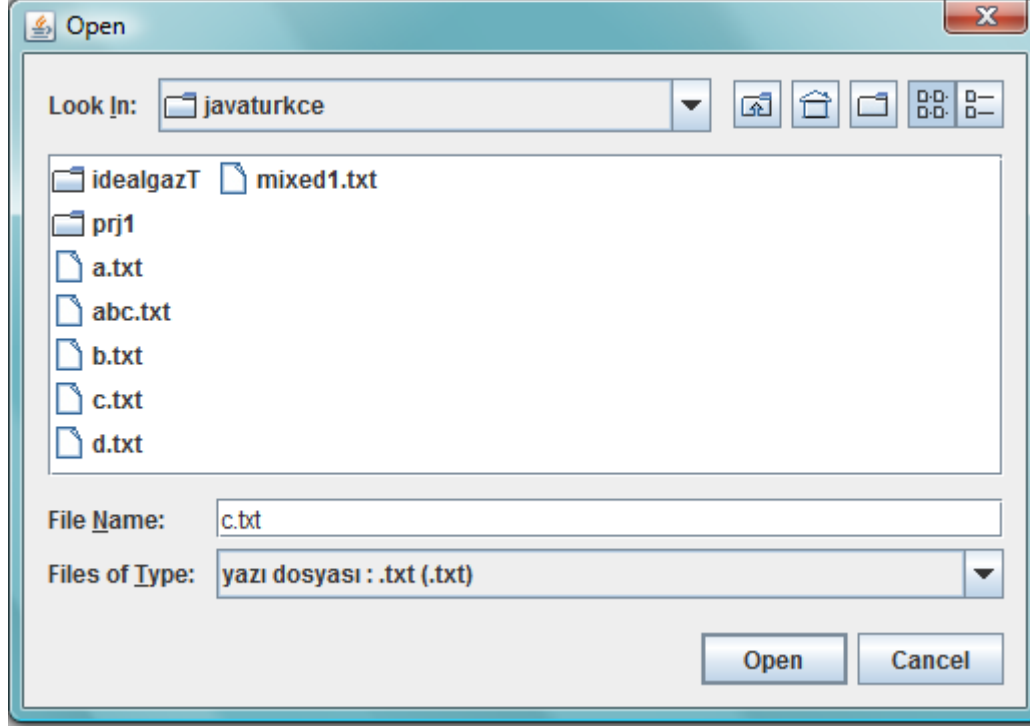
```

public static void main(String args[]) throws IOException
{
        String s1="a.txt";
        JFileChooser fc=new JFileChooser();
        ExampleFileFilter filter = new ExampleFileFilter();
        filter.addExtension("txt");
        filter.setDescription("text file : .txt");
        fc.setFileFilter(filter);
        if (fc.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {File file = fc.getSelectedFile();s1=file.getName(); }
}

```



```
File f=new File(s1);
BufferedReader fin=new BufferedReader(new FileReader(f));
double number[][]=inputdouble(fin);
JOptionPane.showMessageDialog(null,toString(number),"sequential file reading",JOptionPane.PLAIN_MESSAGE);
System.exit(0);
}
}
```



BufferedReader ve FilerReader sınıflarının bazı metodları aşağıda tanımlanmıştır:

Kurucu metodlar:

[BufferedReader\(Reader in\)](#)

[BufferedReader\(Reader in, int sz\)](#)

Diğer metotlardan bazıları

void [close\(\)](#)

kanalı kapatır

int [read\(\)](#)

tek bir karakter okur

int [read\(char\[\] cbuf, int off, int len\)](#)

karakterleri cbuf boyutlu değişkenine aktarır

[String readLine\(\)](#)

bir satır veriyi String olarak okur.

boolean [ready\(\)](#)

kanalın yeni veri okumak için hazır olup olmadığını bildirir.

Long [skip](#)(long n)
n karakteri okumadan atlar

Kurucu metodlar:

[FileReader\(File file\)](#)

[FileReader\(FileDescriptor fd\)](#)

[FileReader\(String fileName\)](#)

Sınıf java.io.InputStreamReader dan aktarılan metodlar

[close](#), [getEncoding](#), [read](#), [read](#), [ready](#)

Sınıf java.io.Reader dan aktarılan metodlar

[mark](#), [markSupported](#), [read](#), [reset](#), [skip](#)

Sınıf java.lang.Object den aktarılan metodlar:

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

11.2.2 SCANNER SINIFI

Scanner sınıfı javanın daha yeni versiyonlarında oluşturulmuş bir sınıftır. Kelime kontrolünü kendi içinde yaparak kelime kelime okuma yetisine sahiptir. Daha önceki programlarımızda StringBuffer kullanarak yaptığımız kelime (girdi) ayırma işlemini otomatik olarak gerçekleştirir.

11.2-7 scanner1

```
import java.util.Scanner; // program uses class Scanner
import java.util.Locale; // international format library Locale
import javax.swing.*; //Java swing graphic library
import java.io.*; //Java girdi library for File class

public class scanner1
{
    // main method begins execution of Java application
    public static void main( String args[] ) throws FileNotFoundException
    {
        // create Scanner to obtain girdi from a file
        String filename=JOptionPane.showInputDialog("dosya ismini giriniz (a.txt): ");
        Scanner girdi = new Scanner(new File(filename));
        Locale us=new Locale("US");
        girdi.useLocale(us);
        double number1; //ilk sayı
        double number2; // ikinci sayı
        double toplam; // iki sayının toplamı
        number1 = girdi.nextDouble(); //
        number2 = girdi.nextDouble(); //
        toplam = number1 + number2; // sayıları topla
        String s=String.format(us,"toplam = %10.5f\n", toplam );
        System.out.println(s); // toplamı yaz
    } // end method main
} // metod sonu
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" scanner1
toplam = 5.00140

> Terminated with exit code 0.
```

Bir sonraki örneğimizde ArrayList boyutlu değişken ve Scanner sınıfını birlikte kullanarak dosya okuma oluşturduk. Burada dosyanın sonlanmasını hasNext() boolean metodu sağlamaktadır. Bu programda verilerin aynı satırda olması gerekliliği yoktur.

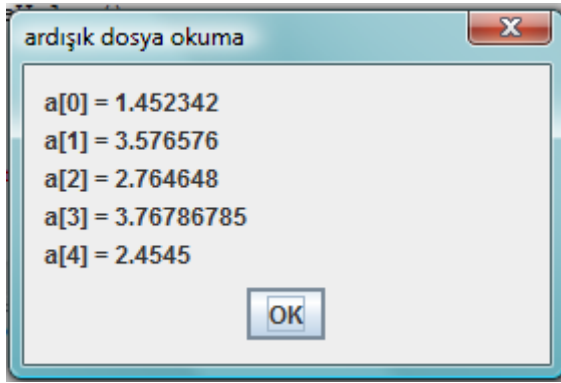
11.2-8 dosyadanoku5

```
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

public class dosyadanoku5
{
    public static double[] doublegir(String dosyaadi) throws IOException
    {
        ArrayList<String> b=new ArrayList<String>();
        Scanner fin = new Scanner(new File(dosyaadi));
        int n=0;
        int k=0;
        while(fin.hasNext())
        {b.add((String)fin.next());}
        double a[]=new double[b.size()];
        Iterator i=b.iterator();
        while(i.hasNext())
        {Double ax=new Double((String)i.next());
        a[k++]=ax.doubleValue();
        }
        return a;
    }

    public static String toString(double a[])
    {
        String s="";
        for(int i=0;i<a.length;i++)
        { s+="a["+i+"] = "+a[i)+"\n";}
        return s;
    }

    public static void main(String args[]) throws IOException
    {
        String s1="b.txt";
        double number[]=doublegir(s1);
        JOptionPane.showMessageDialog(null,toString(number),"ardışık dosya okuma",JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```



Bir diğer benzer örnekte String türü boyutlu değişkeni yine Scanner sınıfını kullanarak okuyacağız. BU örnekte Konsol ortamından java komutunda dosya adı program isminden sonra verilecek şekilde dizayn edilmiştir.

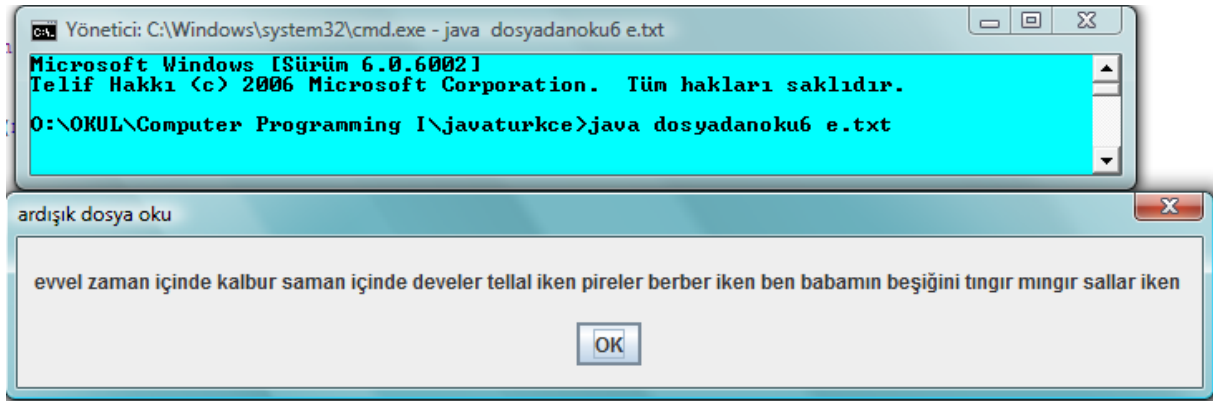
11.2-9 readfromfile6 class example

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
import javax.swing.*;

public class dosyadanoku6 {

    private static String dosyadanoku(String fileName) {
        String s="";
        try {
            File file = new File(fileName);
            Scanner scanner = new Scanner(file);
            while (scanner.hasNext()) {s+=scanner.next()+" ";}
            scanner.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        return s;
    }

    public static void main(String[] args) {
        if (args.length != 1) {
            System.err.println("kullanım: java dosyadanoku6"+ " dosya ismi");
            System.exit(0);
        }
        JOptionPane.showMessageDialog(null,dosyadanoku(args[0]),"ardışık dosya oku",JOptionPane.PLAIN_MESSAGE);
    }
}
```



Scanner sınıfının diğer bir önemli özelliği de karışık tür veriyi okuyabilme özelliğidir.

11.2-10 dosyadanoku7

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
import java.util.Locale;
public class dosyadanoku7 {
    public static void main(String args[]) throws IOException {

        int i;
        double d;
        boolean b;
        String str;
        Locale us=new Locale("US");
        FileReader fin = new FileReader("karisik.txt");
        Scanner src = new Scanner(fin);
        src.useLocale(us);
        while (src.hasNext()) {
            if (src.hasNextInt()) {
                i = src.nextInt();
                System.out.println("int: " + i);
            } else if (src.hasNextDouble()) {
                d = src.nextDouble();
                System.out.println("double: " + d);
            } else if (src.hasNextBoolean()) {
                b = src.nextBoolean();
                System.out.println("boolean: " + b);
            } else {
                str = src.next();
                System.out.println("String: " + str);
            }
        }

        fin.close();
    }
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" dosyadanoku7
String: Scanner
```

```
String: testi  
int: 10  
double: 12.2  
String: bir  
boolean: true  
String: two  
int: 15  
double: 5.123  
boolean: false
```

```
> Terminated with exit code 0.
```

11.3 ARDIŞIK DOSYALARA YAZI YAZDIRILMASI

Yazı yazdırma okutmaya çok benzer bir şekilde yapılır. JAVada kanal açmanın bir çok yolu olduğunu tekrar belirtelim. İlk örneğimizde PrintWriter ve BufferedWriter sınıflarını kullanacağız.

15.3-1 okuyaz

```
import java.io.*;  
class okuyaz  
{  
    public static void main (String args[]) throws IOException  
    {  
        String s1="ilk";  
        String s2="ikinci";  
        String s3="üçüncü";  
        PrintWriter cfout=new PrintWriter(new BufferedWriter(new FileWriter("abc.txt")));  
        cfout.println("Merhaba");  
        cfout.println("İsminiz nedir?");  
        cfout.println("tanıştığımıza memnun oldum");  
        cfout.close();  
        BufferedReader cfin=new BufferedReader(new FileReader("abc.txt"));  
        s1=cfin.readLine();  
        s2=cfin.readLine();  
        s3=cfin.readLine();  
        cfin.close();  
        System.out.println("s1 = "+s1);  
        System.out.println("s2 = "+s2);  
        System.out.println("s3 = "+s3);  
    }  
}
```

----- Capture Output -----

```
> "C:\turhan\java\bin\javaw.exe" okuyaz  
s1 = Merhaba  
s2 = İsminiz nedir?  
s3 = tanıştığımıza memnun oldum  
> Terminated with exit code 0.
```

İkinci örneğimizde FileWriter sınıfını tek başına kullanacağız.

```
import java.io.FileReader;  
import java.io.FileWriter;
```

```

import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
import java.util.Locale;

public class FileWriterTest {
    public static void main(String args[]) throws IOException {

        int i;
        double d;
        boolean b;
        String str;

        FileWriter fout = new FileWriter("karisik.txt");
        fout.write("Scanner testi 10 12.2 bir true two 15 5.123 false");
        fout.close();

        FileReader fin = new FileReader("karisik.txt");
        Scanner src = new Scanner(fin);
        Locale us=new Locale("US");
        src.useLocale(us);
        while (src.hasNext()) {
            if (src.hasNextInt()) {
                i = src.nextInt();
                System.out.println("int: " + i);
            } else if (src.hasNextDouble()) {
                d = src.nextDouble();
                System.out.println("double: " + d);
            } else if (src.hasNextBoolean()) {
                b = src.nextBoolean();
                System.out.println("boolean: " + b);
            } else {
                str = src.next();
                System.out.println("String: " + str);
            }
        }

        fin.close();
    }
}

```

```

----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" FileWriterTest
String: Scanner
String: testi
int: 10
double: 12.2
String: bir
boolean: true
String: two
int: 15
double: 5.123
boolean: false

> Terminated with exit code 0.

```

Ardışık dosya okumanın daha bir çok yolu olmasına rağmen bizim için temel olarak bu yeterli olacaktır. Daha fazlası için io ve nio kütüphanelerini inceleyebilirsiniz.

11.4 ZIP,UNZIP SINIFLARI

Bir veriyi sıkıştırmak istersek **ZipOutputStream** sınıfı java io kütüphanesinde tanımlanmıştır. Örnek problemimizi istersek konsol ortamından istersek grafik ortamından çalışacak şekilde hazırladık:

11.4-1 zip.java

```
//file : zip.java
//Turhan Çoban
import java.io.*;
import java.util.*;
import java.util.zip.*;
import javax.swing.JOptionPane;

public class zip {
    public static int sChunk = 8192;

    public static String[] dosyalistesi(String p)
    {
        File path=new File(p);
        if(path.isDirectory())
        {
            ArrayList<String> x=new ArrayList<String>();
            File files[];
            files=path.listFiles();
            int n=files.length;
            int n1=0;
            for(int i=0;i<n;i++)
            {
                if(!files[i].isDirectory())
                {
                    x.add(files[i].toString());
                    n1++;
                }
            }
            String s1[];
            s1=new String[n1];
            Iterator nn=x.iterator();
            int k=0;
            while(nn.hasNext())
            {s1[k++]=(String)nn.next();}
            return s1;
        }
        else
        {
            String[] s1 = new String[1];
            s1[0]=p;
            return s1;
        }
    }

    public static void zip(String[] args)
    {
        String str="";
    }
}
```



```

FileOutputStream out;
ZipOutputStream zout;
String zipname, source;
int m,mmax=1;
String s="NORMAL USAGE: \n"+
"java zip zipfile file(or directory)name1 file(or directory)name2...\n"+
" create a zip file and zip files(or files in the given directories)"
"Please define the zip file(name.zip)";
// anahtarı kontrol et
String zipismi="";
String dosyaismi[];
if(args.length==0 )
{
String s1="zip dosyası isim)\n"+s;
zipismi=JOptionPane.showInputDialog(null,s);

String s2="file(or directory) names are undefined\n"+
" please enter names\n";
String s3=JOptionPane.showInputDialog(null,s2);
StringTokenizer token=new StringTokenizer(s3);
m=token.countTokens();
if(m>mmax) mmax=m;
dosyaismi=new String[m];
int j=0;
while(token.hasMoreTokens())
{ dosyaismi[j++]=token.nextToken();}

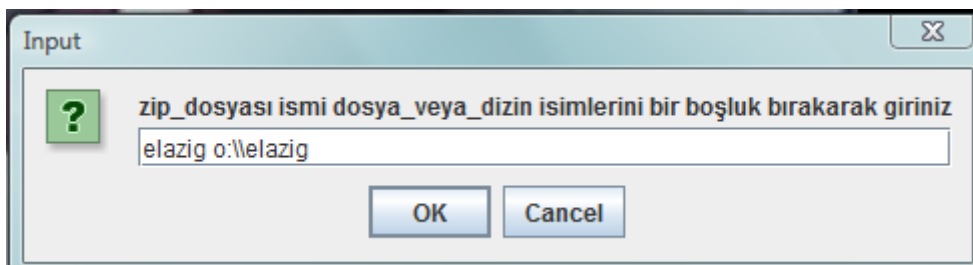
} // girdi dosyasını incele
else if(args.length==1)
{zipismi=args[0];
String s1="file(or directory) to be zipped is undefined please enter the name)\n"+s;
dosyaismi=new String[1];
dosyaismi[0]=JOptionPane.showInputDialog(null,s1);
}
else
{ zipismi=args[0];
int n=args.length-1;
dosyaismi=new String[n];
for(int i=0;i<n;i++)
{dosyaismi[i]=args[i+1];}
}
if (zipismi.endsWith(".zip")) {
zipname = zipismi;
source = zipismi.substring(0, zipismi.length( ) - 3);
}
else {
zipname = zipismi + ".zip";
source = zipismi;
}
//zip kanalını tanımla
try {
out = new FileOutputStream(zipname);
zout=new ZipOutputStream(out);
}
catch (IOException e) {
JOptionPane.showMessageDialog(null,"Dosya oluşturma hatası : " + zipname + ".");
return;
}
}

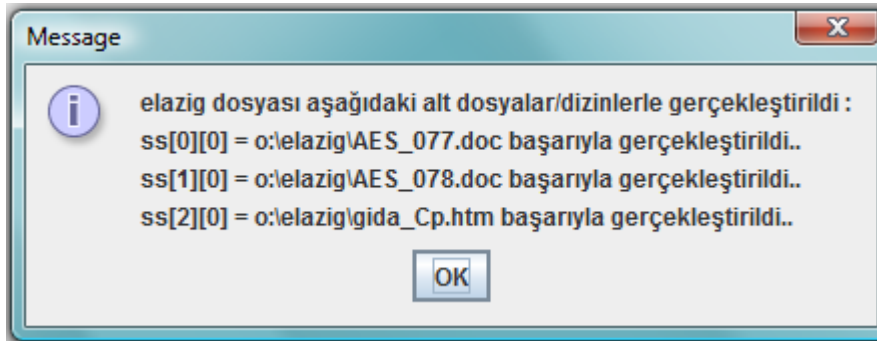
```

```

int k=0;
while(k<dosyaismi.length)
{
try
{
String ss[];
ss=dosyalistesi(dosyaismi[k]);
int j=0;
byte[] buffer=new byte[sChunk];
while(j<ss.length)
{
ZipEntry giris=new ZipEntry(ss[j]);
zout.putNextEntry(giris);
try {
FileInputStream in=new FileInputStream(ss[j]);
int length;
while((length=in.read(buffer,0,sChunk))!=-1)
zout.write(buffer,0,length);
in.close();
} //try sonu
catch (IOException e)
{
JOptionPane.showMessageDialog(null,"file zip error : " +ss[j] );
} //catch sonu
str+="ss["+j+"]["+k+"] = "+giris+" done..\n";
j++;
} //while sonu
} //try sonu
catch(IOException io) {}
k++;
} //while(k< sonu
try{
zout.close();
out.close();
}
catch(IOException e){}
JOptionPane.showMessageDialog(null,zipname+" file is created with the following files : \n"+str);
System.exit(0);
}
}

```





Elbette eğer dosyayı zipliyorsak unzip (tekrar açma) da gerekecektir. Bu gaye için **ZipInputStream** kanalını kullanıyoruz.

11.4-2 unzip.java

```
// dosya : unzip.java
import java.io.*;
import java.util.zip.*;
import javax.swing.JOptionPane;

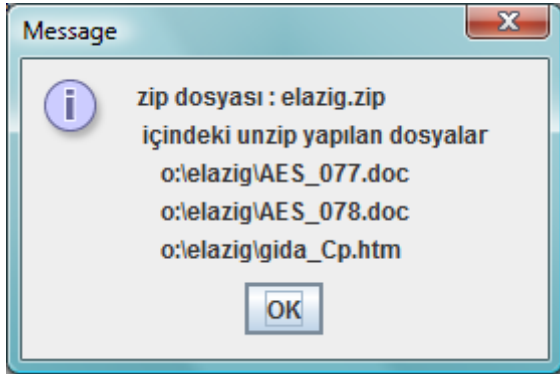
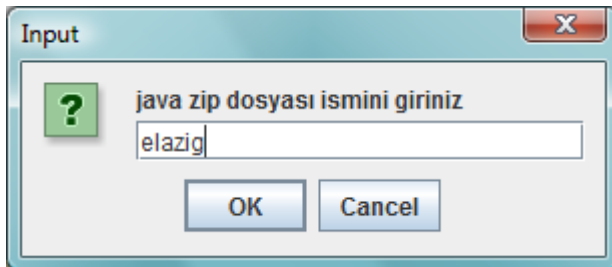
public class unzip {
    public static int sChunk = 8192;

    public static void main(String[] args) {
        String zipname, source;
        String s="";
        if (args.length != 1) {
            zipname=JOptionPane.showInputDialog(null,"USAGE: java Unzip zipfile, \n zip file is not given, please
define(name.zip)");
        }
        else{zipname=args[0];}
        // girdi dosyasını incele
        if (zipname.endsWith(".zip"))
            {source = zipname.substring(0, zipname.length( ) - 3);
        }
        else
        {
            zipname = zipname + ".zip";
            source = zipname;
        }
        s+="zip file : "+zipname+"\n files inside unzipped \n";
        try {
            FileInputStream in = new FileInputStream(zipname);
            ZipInputStream zin=new ZipInputStream(in);
            byte[] buffer = new byte[sChunk];
            // dosyayı aç (decompress
            try{
                ZipEntry z;
                do
                {
                    z=zin.getNextEntry();
                    try {
                        FileOutputStream out = new FileOutputStream(z.getName());
                        int length;
                        while ((length = zin.read(buffer, 0, sChunk)) != -1)
                            out.write(buffer, 0, length);
                    }
                }
            }
        }
    }
}
```

```

        out.close( );
    }
    catch (IOException e) {
        JOptionPane.showMessageDialog(null,
            "File zip error : " + zipname + ".");
        System.exit(0);
    }
    s+=" "+z.getName()+" \n";
}while(z!=null);
} catch (NullPointerException npe)
{ JOptionPane.showMessageDialog(null,s);
  System.exit(0);
}
try { zin.close(); } catch (IOException e) {}
}
catch(IOException e)
{
    JOptionPane.showMessageDialog(null,"File open error : " + zipname);
    System.exit(0);
}
}
}

```



11.5 EKSERSİZLER

EX1) karbondioksit (CO₂) sıcaklık ve özgül ısı verileri co2_cv.txt . Dosyayı okuyunuz ve çıktısını alınız. Kanalı açmak için

BufferedReader fin=new BufferedReader(new FileReader("co2_cv.txt")); deyimini kullanabilirsiniz.

EX 2) : x ve y değerlerini d.txt dosyasından okuyunuz ve x ve \sqrt{y} değerlerini e.txt dosyasına yazdırınız.

EX3) zip.java programını kullanarak dosyadanoku.java dosyasını zipleyiniz.

11.6 PROBLEMLER

HW 1 : dosyadanoku.java dosyasını Strings olarak okuyunuz, toplam kelime sayılarını sayınız, kelimeleri sıraya göre listeleyiniz.

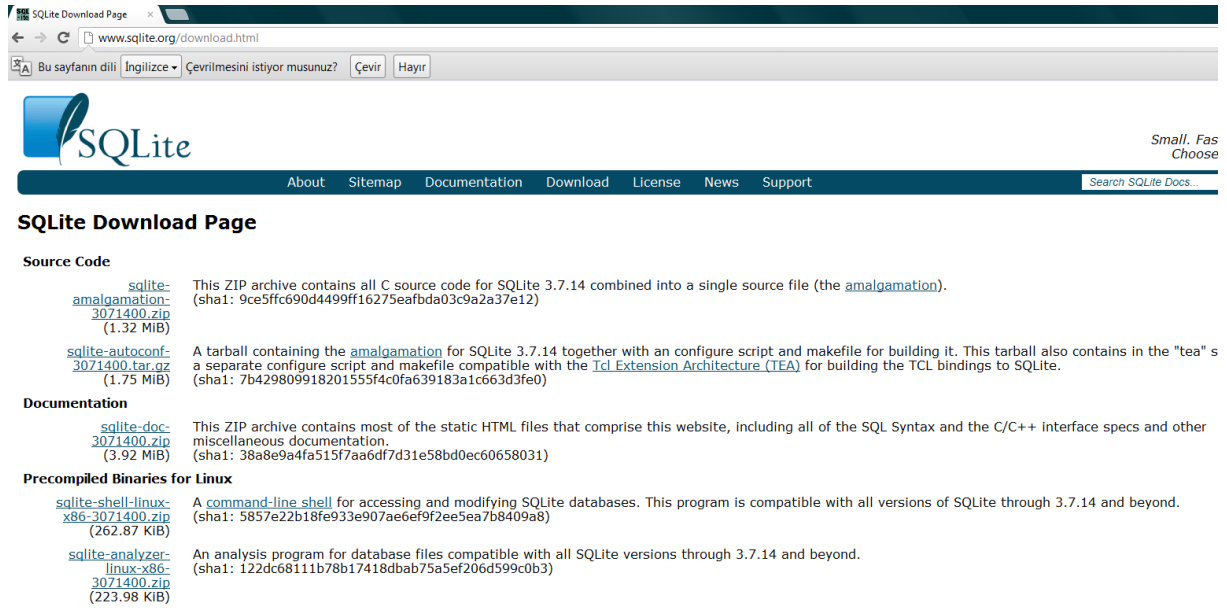
HW 2 : karbondioksit (CO₂) sıcaklık ve özgül ısı verileri co2_cv.txt . Dosyayı okuyunuz ve çıktısını alınız. Ortalama özgül ısı değerini hesaplayınız ve yazdırınız.

12 VERİTABANI DİLİ (SEQUENTIAL QUERY LANGUAGE-ARDIŞIK SORGU DİLİ) İLE PROGRAMLAMA VE JAVA ÜZERİNDEN KULLANIMI

Veri tabanı bir tür organize edilmiş veri koleksiyonudur. Veriler çeşitli şekillerde organize edilebilir ve kullanılabilir. Günümüzdeki en yaygın uygulama ilişkisel veri tabanları denilen sistemlerdir bu sistemler neredeyse tamamen Sequential Query Language-Ardışık sorgulama dili denilen bir dil üzerinden programlanır ve kullanılırlar. Kısaca SQL diye anılan bu dil çok basit sorgulama deyimleriyle bir veri dizininden istenilen alt dizilere ulaşma olasılığı sağlar. Piyasada çeşitli şirketlerin geliştirdiği ve herbiri birbirinden biraz farklı SQL dilleri mevcuttur. Bunlar arasında en popülerleri olarak Microsoft SQL server, Microsoft Office Access, Oracle, Sybase, IBM DB2, MySQL SQLite, Java DB sayılabilir. Biz burada örnek olarak SQLite veri tabanını kullanacağız. Bu veri tabanı kullanım kolaylığıyla son yıllarda ön plana çıkmıştır. Günümüzde yoğun olarak kullanılmaya başlayan Android sisteminin arkasında SQLite veri tabanı çalışmaktadır.

12.1 SQLite VERİTABANI

SQLite çok çeşitli şekillerde rahatlıkla kullanabileceğimiz bir veri tabanıdır. Veri tabanlarını öğrenmek için de öğrencilere iyi bir başlangıç ortamı sunar. Çeşitli ortamlarda, bilgisayarınızda direk olarak, internet üzerinden kullanılabilir. Veri tabanımız <http://www.sqlite.org/download.html> adresinden indirilebilir. Bu sitede doküman kütüphaneleri, C ile yazılmış orijinal kodlar ve çeşitli bilgisayar sistemlerinde çalışan derlenmiş binary kodlar bulunmaktadır.



Source Code

- [sqlite-amalgamation-3071400.zip](#) (1.32 MiB) This ZIP archive contains all C source code for SQLite 3.7.14 combined into a single source file (the [amalgamation](#)). (sha1: 9ce5ffc690d4499ff16275eafbd03c9a2a37e12)
- [sqlite-autoconf-3071400.tar.gz](#) (1.75 MiB) A tarball containing the [amalgamation](#) for SQLite 3.7.14 together with an configure script and makefile for building it. This tarball also contains in the "tea" s a separate configure script and makefile compatible with the [Tcl Extension Architecture \(TEA\)](#) for building the TCL bindings to SQLite. (sha1: 7b429809918201555f4c0fa639183a1c663d3fe0)

Documentation

- [sqlite-doc-3071400.zip](#) (3.92 MiB) This ZIP archive contains most of the static HTML files that comprise this website, including all of the SQL Syntax and the C/C++ interface specs and other miscellaneous documentation. (sha1: 38a8e9a4fa515f7aa6df7d31e58bd0ec60658031)

Precompiled Binaries for Linux

- [sqlite-shell-linux-x86-3071400.zip](#) (262.87 KIB) A [command-line shell](#) for accessing and modifying SQLite databases. This program is compatible with all versions of SQLite through 3.7.14 and beyond. (sha1: 5857e22b18fe933e907ae6ef9f2ee5ea7b8409a8)
- [sqlite-analyzer-linux-x86-3071400.zip](#) (223.98 KIB) An analysis program for database files compatible with all SQLite versions through 3.7.14 and beyond. (sha1: 122dc6811b78b17418dbab75a5ef206d599c0b3)

Siteden yükleyeceğimiz [sqlite-dll-win32-x86-3070800.zip](#) dosyasını açarsak `sqlite3.exe` dosyasını bulacağız, bu dosyayı dos ortamında çalıştırabiliriz. SQLite veritabanı programını java'ya iki türlü bağlayabiliriz. İlki wrapper denilen direk olarak java anadil programlamasıyla (C dilinde) yazılmış programlar aracılığıyla SQLite veri tabanı kullanımını oluşturmaktır. Bunlara en iyi örnek olarak <http://code.google.com/p/sqlite4java> adresinde verilen `sqlite4java` verilebilir. Bu programı bilgisayarınıza yüklemek için önce `sqlite4java-282.zip` sıkıştırılmış dosyasını yükleyiniz. Bu dosyayı açıp içindeki `sqlite4java.jar` dosyasını `JAVA_HOME/jre/lib/ext` dizinine atınız. Sıkıştırılmış dosyada bulunan `sqlite4java-win32-x64.dll` (Windows 64 bit bilgisayar) veya `sqlite4java-win32-x86.dll` (Windows 32 bit bilgisayar) dosyasını `JAVA_HOME/jre/bin` dizinine kopyalayınız. Veri tabanı ile bağlantı kurmak içinde `import com.almworks.sqlite4java.*;` deyimini kullanınız. Diğer

yol ise javanın veri tabanlarına bağlanmak için geliştirdiği Java Data Base Connectivity _JAVA Veri tabanı bağlanma sistemi (JDBC) denilen sınıflar üzerinden yapmaktır. JDBC örneği olarak <http://www.xerial.org/maven/repository/artifact/org/xerial/sqlite-jdbc/3.7.2/> sitesinde bulunan [sqlite-jdbc-3.7.2.jar](#) dosyasını kullanarak gerçekleştirilebilir. Bu paketi kullanabilmemiz için jar dosyasını JAVA_HOME/jre/lib/ext dizinine kopyelememiz yeterli olacaktır. Bu pakete bağlanmak için standart java ODBC paketi olan **import java.sql.Connection;** deyimini kullanmak yeterli olacaktır. Diğer paketlerin kullanımları biraz daha farklıdır, ilgilenenler lütfen internet sitelerine müracat etsinler. Şimdi önce SQLite3 kullanarak dos ortamında SQLite veri tabanını ve çeşitli deyimlerine göz atalım. Oluşturacağımız veri tabanının ismi temp olsun. Oluşturduğumuz veri tabanında yeni veri tabanı tablosu oluşturmamız gerekir. Bunun için **create table** SQL komutunu kullanıyoruz.

```
ca. Yönetici: C:\Windows\system32\cmd.exe - sqlite3 temp
Microsoft Windows [Sürüm 6.0.6002]
Telif Hakkı (c) 2006 Microsoft Corporation. Tüm hakları saklıdır.
O:\OKUL\Computer Programing I\javaturkce>sqlite3 temp
SQLite version 3.7.8 2011-09-19 14:49:19
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> create table KAHVE(isim String,kot int,fiyat double,miktar int,toplam do
uble);
sqlite> .separator ,
sqlite> .import veri.txt KAHVE
sqlite> select * from KAHVE;
Brezilya,123,12.99,30,0.0
Kolombia,111,15.99,20,0.0
Espresso,143,9.95,29,0.0
sqlite> -
```

.separator , komutu verinin virgül (,) ile ayrılacağını belirtti, .import komutu da veriyi veri.txt dosyasından okudu. Burada dosyada verilen alan sayısı ve tiplerinin veri tabanı tablosu tanımında verilenle aynı olması gerektiğini hatırlatalım. Veri dosyası içeriği:

```
zip.java  unzip.java  co2_cv.txt
1 Brezilya,123,12.99,30,0
2 Kolombia,111,15.99,20,0
3 Espresso,143,9.95,29,0
4
```

Veri dosyasından çıktı almak için **select * form KAHVE** komutunu kullandık. Veriyi dosyadan okuma yerine direk olarak kendimiz de girebiliriz. Bunun için **insert into** SQL komutunu kullanabiliriz.

```
ca. Yönetici: C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> insert into KAHVE values('Venezuela',145,13.75,40,0);
sqlite> select * from KAHVE;
Brezilya,123,12.99,30,0.0
Kolombia,111,15.99,20,0.0
Espresso,143,9.95,29,0.0
Venezuela,145,13.75,40,0.0
sqlite>
```

Select deyiminin en genel yapısı şu şekildedir.

```
SELECT [DISTINCT|ALL] {*| liste}
FROM [tablo ismi],[ikinci tablo ismi]....
[WHERE şart deyimi]
[GROUP BY kolon liste]
```

[HAVING şart deyimi]
[ORDER BY kolon liste]

Bu tanımlamada **DISTINCT** terimi veri tablosunda yinelenen değerler varsa onların tek olarak gelmesini sağlar. Eğer **ALL** terimi kullanılırsa tüm veri gelir.

```
ca: Yönetici: C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> select * from KAHVE;
isim      kot      fiyat    miktar   toplam
-----
Brezilya  123      12.99    30       0.0
Kolombia  111      15.99    20       0.0
Espresso  143      9.95     29       0.0
Venezuela 145      13.75    40       0.0
Venezuela 145      13.75    40       0.0
sqlite> select distinct * from KAHVE;
isim      kot      fiyat    miktar   toplam
-----
Brezilya  123      12.99    30       0.0
Kolombia  111      15.99    20       0.0
Espresso  143      9.95     29       0.0
Venezuela 145      13.75    40       0.0
sqlite> select all * from KAHVE;
isim      kot      fiyat    miktar   toplam
-----
Brezilya  123      12.99    30       0.0
Kolombia  111      15.99    20       0.0
Espresso  143      9.95     29       0.0
Venezuela 145      13.75    40       0.0
Venezuela 145      13.75    40       0.0
sqlite>
```

Eğer * terimi kullanılmışsa tüm tablo kolonları listelenir. Bunun yerine sadece listelenmesini istediğimiz kolonları verebiliriz.

```
ca: Yönetici: C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> select distinct isim,fiyat from KAHVE;
isim      fiyat
-----
Brezilya  12.99
Kolombia  15.99
Espresso  9.95
Venezuela 13.75
sqlite>
```

WHERE deyime java'daki if deyiminin benzeridir. Bu deyimi kullanarak tablodan seçim yapabiliriz.

Select komutunu kullanırken if yapısına benzer seçim kriterleri kullanabiliyoruz. Örneğin

Select * from KAHVE where fiyat>10 and fiyat<15;

Deyimi fiyatı 10 ila 15 TL arasında olan kahveleri seçer.

```
ca: Yönetici: C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> .mode column
sqlite> .headers on
sqlite> select * from KAHVE where fiyat>10 and fiyat<15;
isim      kot      fiyat    miktar   toplam
-----
Brezilya  123      12.99    30       0.0
Venezuela 145      13.75    40       0.0
sqlite>
```

Where ile kullanılabilen mantıksal seçim deyimleri şunlardır:

AND ve

OR veya

NOT değil

BETWEEN (genellikle AND deyimi ile birlikte kullanılır örneğin **BETWEEN 20 AND 30** 20 ile 30 arasında seçim yapar)

IN içinde

LIKE (String değişkenleri için eşittir deyimi

LIKE '%ali%' içinde ali olan tüm deyimleri bulur

NULL : boş

ANY herhangi bir değer

ALL hepsi

EXIST mevcutsa

Rakamsal karşılaştırmalarda:

= : eşittir

<> : eşit değildir

> : büyüktür

< : küçüktür

>= : büyük veya eşittir

<= : küçük veya eşittir

ORDER BY deyimi veriyi sıraya sokar. Eğer ASC kullanıldıysa küçükten büyüğe doğru sıralanır, eğer DESC kullanıldıysa küçükten büyüğe doğru sıraya sokar.

```
C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> select * from KAHUE order by fiyat asc;
isim      kot      fiyat    miktar   toplam
-----
Espresso  143      9.95     29       0.0
Brezilya  123      12.99    30       0.0
Venezuela 145      13.75    40       0.0
Venezuela 145      13.75    40       0.0
Kolombia  111      15.99    20       0.0
sqlite> _
```

Eğer String türü bir karşılaştırma gerekiyorsa like terimini kullanabiliriz.

```
C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> select distinct * from KAHUE where isim like '%Kol%';
isim      kot      fiyat    miktar   toplam
-----
Kolombia  111      15.99    20       0.0
sqlite> select distinct * from KAHUE where isim like 'Kolombia';
isim      kot      fiyat    miktar   toplam
-----
Kolombia  111      15.99    20       0.0
sqlite> _
```

Diğer önemli bir SQL komutu hesaplama yapmakta kullandığımız set komutudur. Bu komutla yeni bir hesap yaparak daha önceden tanımlanmış bir kolona veya yeniden oluşturulacak bir kolona veri aktarılabilir. Örneğimizde toplam kolonunun değerleri hesaplanmıştır.

```
C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> update KAHUE set toplam=fiyat*miktar;
sqlite> select distinct * from KAHUE order by fiyat asc;
isim      kot      fiyat    miktar   toplam
-----
Espresso  143      9.95     29       288.55
Brezilya  123      12.99    30       389.7
Venezuela 145      13.75    40       550.0
Kolombia  111      15.99    20       319.8
sqlite> _
```

Bir tabloyu silmek için drop table komutu kullanırız.

```
C:\Windows\system32\cmd.exe - sqlite3 temp
sqlite> drop table KAHUE;
sqlite> select * from KAHUE;
Error: no such table: KAHUE
sqlite> _
```

Tüm veri tabanını silmek içinse drop database terimini kullanırız.

Şimdi SQLite programlamayı java ortamında yapmak için örneklere bakalım. Başlangıç bölümünde java için SQLite kurulumundan bahsetmiştik. Burada tekrar yinelemeye gerek görmüyoruz. İlk örnek olarak üsteki eksersizinin benzerini java ortamında gerçekleştirelim. Bu ilk java programımızda temel olarak Java Data Base Connectivity _JAVA Veri tabanı bağlanma sistemi (JDBC) kullanılacaktır.

Program 12.1-1 KAHVEjava

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class KAHVE
{
    public static void main(String[] args) throws ClassNotFoundException
    {
        // sqlite-JDBC sürücüsünü yükle
        Class.forName("org.sqlite.JDBC");
        Connection connection = null;
        try
        {
            // veritabanı ile bağlantı kur.
            connection = DriverManager.getConnection("jdbc:sqlite:temp.db");
            Statement statement = connection.createStatement();
            statement.setQueryTimeout(30); // set timeout to 30 sec.

            statement.executeUpdate("drop table if exists KAHVE");
            statement.executeUpdate("create table KAHVE(isim String, kot int, fiyat double, miktar int,toplam double)");
            statement.executeUpdate("insert into KAHVE values('Espresso',150,9.99,110,0);");
            statement.executeUpdate("insert into KAHVE values('Venezuela',173,12.99,57,0);");
            statement.executeUpdate("insert into KAHVE values('Brezilya',123,12.99,30,0);");
            statement.executeUpdate("insert into KAHVE values('Kolombiya',111,15.99,20,0);");
            statement.executeUpdate("update KAHVE set toplam=fiyat*miktar;");
            ResultSet rs = statement.executeQuery("select * from KAHVE");
            while(rs.next())
            {
                // read the result set
                System.out.print("KAHVE İSMİ = " + rs.getString("isim"));
                System.out.print(" KOT = " + rs.getInt("kot"));
                System.out.print(" FİYAT = " + rs.getDouble("fiyat"));
                System.out.print(" MİKTAR = " + rs.getInt("miktar"));
                System.out.println(" TOPLAM = " + rs.getDouble("toplam"));
            }
        }
        catch(SQLException e)
        {
            // if the error message is "out of memory",
            // it probably means no database file is found
            System.err.println(e.getMessage());
        }
        finally
        {
            try
            {
                if(connection != null)
                    connection.close();
            }
            catch(SQLException e)
            {
                // connection close failed.
                System.err.println(e);
            }
        }
    }
}
```

```
}
```

----- Capture Output -----

```
> "D:\java\bin\javaw.exe" KAHVE  
KAHVE İSMİ = Espresso KOT = 150 FİYAT = 9.99 MİKTAR = 110 TOPLAM = 1098.9  
KAHVE İSMİ = Venezuela KOT = 173 FİYAT = 12.99 MİKTAR = 57 TOPLAM = 740.43  
KAHVE İSMİ = Brezilya KOT = 123 FİYAT = 12.99 MİKTAR = 30 TOPLAM = 389.7  
KAHVE İSMİ = Kolombia KOT = 111 FİYAT = 15.99 MİKTAR = 20 TOPLAM = 319.8
```

```
> Terminated with exit code 0.
```

Programdan da görüldüğü gibi SQLite veritabanı programına JDBC olarak `Class.forName("org.sqlite.JDBC");` deyimiyle bağlanıldı. Veri tabanı `temp.db` ye bağlanmak için: `Connection connection == DriverManager.getConnection("jdbc:sqlite:temp.db");` terimi kullanıldı. Veri tabanımızın ismi **temp.db**'dir. veri tabanı tablomuzun ismi de **KAHVE** olarak tanımlandı.

İkinci örneğimizde `KAHVE.java` programında JDBC olarak verilen kodun `SQLite4` wrapper paketi için yazılmış eşdeğeri verilmektedir.

Program 12.1-2 KAHVE1.java (SQLite wrapper program using SQLite4)

```
import com.alworks.sqlite4java.*;  
import java.io.*;  
  
public class KAHVE1  
{  
    public static void main(String arg[]) throws SQLException  
    {  
        SQLiteConnection db = new SQLiteConnection(new File("/temp"));  
        db.open(true);  
        db.exec("drop table if exists KAHVE");  
        db.exec("create table KAHVE(isim String, kot int, fiyat double, miktar int,toplam double)");  
        db.exec("drop table if exists KAHVE");  
        db.exec("create table KAHVE(isim String, kot int, fiyat double, miktar int,toplam double)");  
        db.exec("insert into KAHVE values('Espresso',150,9.99,110,0);");  
        db.exec("insert into KAHVE values('Venezuela',173,12.99,57,0);");  
        db.exec("insert into KAHVE values('Brezilya',123,12.99,30,0);");  
        db.exec("insert into KAHVE values('Kolombia',111,15.99,20,0);");  
        db.exec("update KAHVE set toplam=fiyat*miktar;");  
        SQLiteStatement statement = db.prepare("select * from KAHVE");  
        String isim;  
        int kot;  
        double fiyat;  
        int miktar;  
        double toplam;  
        try {  
            while (statement.step()) {  
                isim=statement.columnString(statement.getBindParameterIndex("isim"));  
                kot=statement.columnInt(statement.getBindParameterIndex("kot"));  
                fiyat=statement.columnDouble(statement.getBindParameterIndex("fiyat"));  
                miktar=statement.columnInt(statement.getBindParameterIndex("miktar"));  
                toplam=statement.columnDouble(statement.getBindParameterIndex("toplam"));  
                System.out.print("KAHVE İSMİ = " + isim);  
                System.out.print(" KOT = " + kot);  
                System.out.print(" FİYAT = " + fiyat);  
                System.out.print(" MİKTAR = " + miktar);  
                System.out.println(" TOPLAM = " + toplam);  
            }  
        }  
    }  
}
```

```
} finally {  
    statement.dispose();  
}  
}
```

----- Capture Output -----

```
> "D:\java\bin\javaw.exe" KAHVE1  
Oca 21, 2013 12:12:12 PM com.almworks.sqlite4java.Internal log  
INFO: [sqlite] DB[1]: instantiated [\temp]  
Oca 21, 2013 12:12:12 PM com.almworks.sqlite4java.Internal log  
INFO: [sqlite] Internal: loaded sqlite4java-win32-x86 from system path  
Oca 21, 2013 12:12:12 PM com.almworks.sqlite4java.Internal log  
INFO: [sqlite] Internal: loaded sqlite 3.7.10, wrapper 0.2  
Oca 21, 2013 12:12:12 PM com.almworks.sqlite4java.Internal log  
INFO: [sqlite] DB[1]: opened  
KAHVE İSMİ = Espresso KOT = 150 FİYAT = 9.99 MİKTAR = 110 TOPLAM = 1098.9  
KAHVE İSMİ = Venezuela KOT = 173 FİYAT = 12.99 MİKTAR = 57 TOPLAM = 740.4300000000001  
KAHVE İSMİ = Brezilya KOT = 123 FİYAT = 12.99 MİKTAR = 30 TOPLAM = 389.7  
KAHVE İSMİ = Kolombia KOT = 111 FİYAT = 15.99 MİKTAR = 20 TOPLAM = 319.8  
  
> Terminated with exit code 0.
```

Programdan da görüleceği gibi, burada veri tabanı sistemine bağlanmak için **import com.almworks.sqlite4java.*;** komutu kullanılmıştır. Veri tabanını açmak ve SQL deyimlerini çalıştırmak için **SQLiteConnection** sınıfı kullanıldı. Veri tabanından verileri java'ya aktarmak için ise **SQLiteStatement** sınıfından yararlanıldı. Görüldüğü gibi temel komutlar farklı olmakla beraber veriye bağlanma ve çağırma prosesi oldukça benzerdir. Bundan sonraki örneklerimizde karışıklığa sebep vermeme için programlarımızı JDBC kullanarak oluşturacağız. Bir sonraki programımızda genel bir grafik sorgu giriş çıkış sistemi yaratılmıştır. Program tablo formunda çıkış almak için bir TableModel programı ve Grafik girdi-çıkış programından oluşmuştur.

Program 12.1-3 SQLite_Table.java

```
import javax.swing.table.*;  
import java.sql.*;  
  
public class SQLite_Table extends JFrame  
{  
    JTextField hostField;  
    JTextField tableField;  
    JTextArea queryField;  
    QueryTableModel qtm;  
    String s="jdbc:sqlite:";  
  
    static final long serialVersionUID = 8597941330711875390L;  
    public SQLite_Table() throws ClassNotFoundException  
    {  
        super("SQLite veritabanı SQL tablo sorgusu");  
        Class.forName("org.sqlite.JDBC");  
        Connection connection = null;  
  
        //addWindowListener(new BasicWindowMonitor());  
        setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );  
        setSize(500,300);  
        qtm=new QueryTableModel();  
    }  
}
```

```

JTable table=new JTable(qtm);
JScrollPane scrollpane=new JScrollPane(table);
JPanel p1=new JPanel();
p1.setLayout(new GridLayout(3,2));
p1.add(new JLabel(" veritabanı'nın ismini giriniz :"));
p1.add(hostField=new JTextField("temp.db"));
p1.add(new JLabel("SQL deyimini giriniz :"));
p1.add(queryField=new JTextArea("select *\n from KAHVE"));
p1.add(new JLabel("veri tabanını gönderiniz:"));
JButton jb=new JButton("veri tabanı sorgusunu çalıştırınız");
jb.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
qtm.setHostURL(s+hostField.getText().trim());
qtm.setQuery(queryField.getText().trim());
}
}); // jb.addActionListener
p1.add(jb);
getContentPane().add(p1, BorderLayout.NORTH);
getContentPane().add(scrollpane, BorderLayout.CENTER);
} //DatabaseTest()

public static void main(String args[])
{
try
{
SQLite_Table tt=new SQLite_Table();
tt.setVisible(true);
} catch(ClassNotFoundException e)
{
// "out of memory" mesajı veri tabanının bulunamadığı anlamına gelir
System.err.println(e.getMessage());
}
}
}
}

```

Program 12.1-3 QueryTableModel

```

import java.sql.*;
import java.io.*;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.*;

public class QueryTableModel extends AbstractTableModel
{
Vector<Object> cache;
int colCount;
String[] headers;
Connection db;
Statement statement;
String currentURL;
static final long serialVersionUID = 67895236875L;
public QueryTableModel()
{
cache=new Vector<Object>();

```

```

}

public String getColumnName(int i) {return headers[i];}
public int getColumnCount() {return colCount;}
public int getRowCount() {return cache.size();}

public Object getValueAt(int row,int col)
{
return ((String[])cache.elementAt(row))[col];
}

public void setHostURL(String url)
{
if(url.equals(currentURL))
{return;}
closeDB();
initDB(url);
currentURL=url;
}

public void setQuery(String q)
{
cache= new Vector<Object>();
try{
ResultSet rs=statement.executeQuery(q);
ResultSetMetaData meta=rs.getMetaData();
colCount=meta.getColumnCount();
headers=new String[colCount];
for (int h=1;h<=colCount;h++)
{
headers[h-1]=meta.getColumnName(h);
}
while(rs.next())
{
String[] record=new String[colCount];
for(int i=0;i<colCount;i++)
{record[i]=rs.getString(i+1);}
cache.add(record);
} //while'in sonu
fireTableChanged(null);
} //try'in sonu
catch(Exception e)
{
cache=new Vector<Object>();
e.printStackTrace();
}
} // setQuery sonu

public void initDB(String url)
{
try {
db=DriverManager.getConnection(url);
statement=db.createStatement();
}
catch(Exception e)
{
System.out.println("veri tabanı kapatılmadı");
e.printStackTrace();
}
}

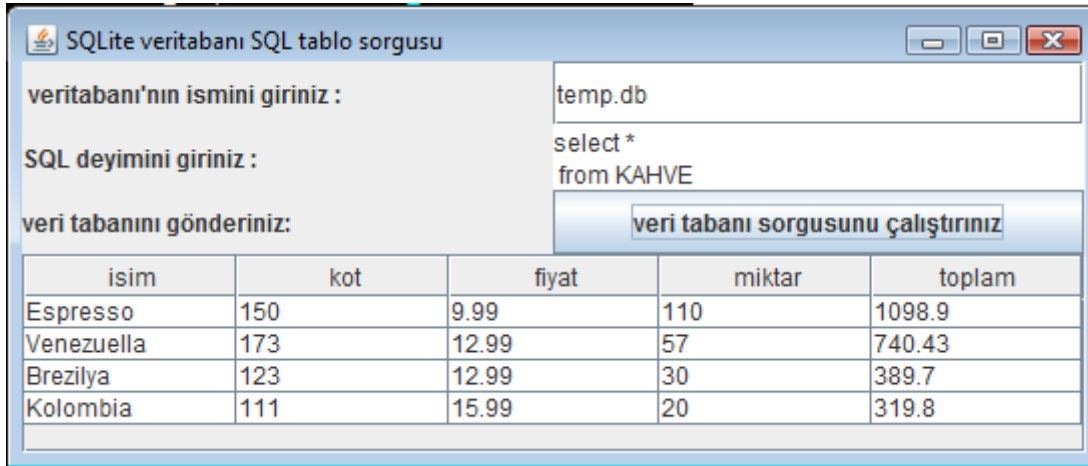
```

```

    }
} //initDB sonu

public void closeDB()
{
    try {
        if(statement!= null) {statement.close();}
        if(db != null) {db.close();}
    }
    catch(Exception e)
    {
        System.out.println("veri tabanı kapatılmadı");
        e.printStackTrace();
    }
} //closeDB sonu
}

```



SQLite anahtar kelimeleri:

SQLite veritabanı ve tablo ismi olarak kullanılma müsadesi olmayan bir çok anahtar kelime barındırır. Bu kelimeler çeşitli komutların çalıştırılması için kullanılırlar. Aşağıda SQLite’da kullanılan anahtar kelimelerin bir listesi verilmiştir.

ABORT	CREATE	FROM	NATURAL	ROW
ACTION	CROSS	FULL	NO	SAVEPOINT
ADD	CURRENT_DATE	GLOB	NOT	SELECT
AFTER	CURRENT_TIME	GROUP	NOTNULL	SET
ALL	CURRENT_TIMESTAMP	HAVING	NULL	TABLE
ALTER	DATABASE	IF	OF	TEMP
ANALYZE	DEFAULT	IGNORE	OFFSET	TEMPORARY
AND	DEFERRABLE	IMMEDIATE	ON	THEN
AS	DEFERRED	IN	OR	TO
ASC	DELETE	INDEX	ORDER	TRANSACTION
ATTACH	DESC	INDEXED	OUTER	TRIGGER
AUTOINCREMENT	DETACH	INITIALLY	PLAN	UNION
BEFORE	DISTINCT	INNER	PRAGMA	UNIQUE
BEGIN	DROP	INSERT	PRIMARY	UPDATE

BETWEEN	EACH	INSTEAD	QUERY	USING
BY	ELSE	INTERSECT	RAISE	VACUUM
CASCADE	END	INTO	REFERENCES	VALUES
CASE	ESCAPE	IS	REGEXP	VIEW
CAST	EXCEPT	ISNULL	REINDEX	VIRTUAL
CHECK	EXCLUSIVE	JOIN	RELEASE	WHEN
COLLATE	EXISTS	KEY	RENAME	WHERE
COLUMN	EXPLAIN	LEFT	REPLACE	
COMMIT	FAIL	LIKE	RESTRICT	
CONFLICT	FOR	LIMIT	RIGHT	
CONSTRAINT	FOREIGN	MATCH	ROLLBACK	

SQLite kullanırken java kullanırken olduğu gibi hazır olarak tanımlanmış fonksiyonları kullanabiliriz. Örnek programımız verilerin ortalamasını almaktadır. Bunun için SQLite avg metodu kullanılmıştır.

Program 12.1-4 ortalama.java programı

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ortalama
{
    public static void main(String[] args) throws ClassNotFoundException
    {
        // sqlite-JDBC sürücüsünü yükle
        Class.forName("org.sqlite.JDBC");
        Connection connection = null;
        ResultSet rs = null;
        try
        {
            // veritabanı ile bağlantı kur.
            connection = DriverManager.getConnection("jdbc:sqlite:temp.db");
            Statement statement = connection.createStatement();
            statement.setQueryTimeout(30); // set timeout to 30 sec.
            if(statement.execute("select avg(fiyat) as T from KAHVE")) {rs = statement.getResultSet();}
            while(rs.next())
            {System.out.print("Ortalama = " + rs.getDouble("T"));}
        }
        catch(SQLException e)
        {
            // if the error message is "out of memory",
            // it probably means no database file is found
            System.err.println(e.getMessage());
        }
        finally
        {
            try
            {
                if(connection != null)
                    connection.close();
            }
        }
    }
}

```



```
catch(SQLException e)
{
    // connection close failed.
    System.err.println(e);
}
}
}
}
```

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" ortalama
Ortalama = 12.99
> Terminated with exit code 0.
```

Bir sonraki programda aynı ortalamaları java ile hesaplayarak sonucu alalım

Program 12.1-5 ortalama1.java programı

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ortalama1
{
    public static void main(String[] args) throws ClassNotFoundException
    {
        // sqlite-JDBC sürücüsünü yükle
        Class.forName("org.sqlite.JDBC");
        Connection connection = null;
        ResultSet rs = null;
        try
        {
            // veritabanı ile bağlantı kur.
            connection = DriverManager.getConnection("jdbc:sqlite:temp.db");
            Statement statement = connection.createStatement();
            statement.setQueryTimeout(30); // set timeout to 30 sec.
            if(statement.execute("select fiyat from KAHVE")) {rs = statement.getResultSet()};
            int j=0;
            double sum=0;
            while (rs.next()) {
                double number= rs.getDouble("fiyat");
                sum+=number;
                j++;
            }
            double ortalama=sum/j;
            System.out.println("ortalama : "+ortalama);
        }
        catch(SQLException e)
        {
            // if the error message is "out of memory",
            // it probably means no database file is found
            System.err.println(e.getMessage());
        }
        finally
        {
            try
```

```

    {
        if(connection != null)
            connection.close();
    }
    catch(SQLException e)
    {
        // connection close failed.
        System.err.println(e);
    }
}
}
}
}

```

Veri tabanından okuma ve yazma yapmak üzere hazırladığımız genel bir sınıfı verelim, bu sınıfı kullanarak detaylara girmeden veri tabanındaki verileri double değişken olarak okuyup yazmak mümkündür.

Program 12.1-6 SQLite.java, genel kullanım veri tabanı okuma programı

```

import java.sql.Statement;
import java.sql.*;
import java.util.Vector;
import java.io.*;
// This class reads, writes and plot data into/from a given SQL Database table
//
public class SQLite
{
// Reads given data column from a specified Lite database table
public static double[][] read_double(String databasename, String tablei,String column[]) throws
IOException,ClassNotFoundException
{
// load the sqlite-JDBC driver using the current class loader
double a[][];
int n=1,m=1;
Class.forName("org.sqlite.JDBC");
Connection connection = null;
try
{
// create a database connection
connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
Statement statement = connection.createStatement();
statement.setQueryTimeout(30); // set timeout to 30 sec.
//statement.executeUpdate("drop table if exists H9Ex3");
ResultSet rs = statement.executeQuery("select * from "+tablei);
int i=0;
n=column.length;
double x;
while(rs.next())
{
x=rs.getDouble(column[0]);
i++;
}
m=i;
}
catch(SQLException e)
{
// if the error message is "out of memory",
// it probably means no database file is found
System.err.println(e.getMessage());
}
}
}

```

```

}
finally
{
    try
    {
        if(connection != null)
            connection.close();
    }
    catch(SQLException e)
    {
        // connection close failed.
        System.err.println(e);
    }
}
a=new double[n][m];
try
{
    // create a database connection
    connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
    Statement statement = connection.createStatement();
    statement.setQueryTimeout(30); // set timeout to 30 sec.
    //statement.executeUpdate("drop table if exists H9Ex3");
    String s="select ";
    ResultSet rs = statement.executeQuery("select * from "+tablei);
    for(int i=0;i<m;i++)
    {rs.next();
    for(int j=0;j<n;j++)
    {
        a[j][i]=rs.getDouble(column[j]);
    }
    }
}
catch(SQLException e)
{
    // if the error message is "out of memory",
    // it probably means no database file is found
    System.err.println(e.getMessage());
}
finally
{
    try
    {
        if(connection != null)
            connection.close();
    }
    catch(SQLException e)
    {
        // connection close failed.
        System.err.println(e);
    }
}
return a;
}
//Reads all data columns from a specified Lite database table
public static double[][] read_double(String databasename, String tablei) throws IOException,ClassNotFoundException
{
    // load the sqlite-JDBC driver using the current class loader
    double a[][];

```

```

int n=1,m=1;
Class.forName("org.sqlite.JDBC");
Connection connection = null;
try
{
    // create a database connection
    connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
    Statement statement = connection.createStatement();
    statement.setQueryTimeout(30); // set timeout to 30 sec.
    //statement.executeUpdate("drop table if exists H9Ex3");
    ResultSet rs = statement.executeQuery("select * from "+tablei);
    ResultSetMetaData meta=rs.getMetaData();
    int colCount=meta.getColumnCount();
    String column[]=new String[colCount];
    for (int h=0;h<colCount;h++)
    {
        column[h]=meta.getColumnName(h+1);
    }
    int i=0;
    n=colCount;
    double x;
    while(rs.next())
    {
        x=rs.getDouble(column[0]);
        i++;
    }
    m=i;
}
catch(SQLException e)
{
    // if the error message is "out of memory",
    // it probably means no database file is found
    System.err.println(e.getMessage());
}
finally
{
    try
    {
        if(connection != null)
            connection.close();
    }
    catch(SQLException e)
    {
        // connection close failed.
        System.err.println(e);
    }
}
a=new double[n][m];
try
{
    // create a database connection
    connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
    Statement statement = connection.createStatement();
    statement.setQueryTimeout(30); // set timeout to 30 sec.
    //statement.executeUpdate("drop table if exists H9Ex3");
    String s="select ";
    ResultSet rs = statement.executeQuery("select * from "+tablei);
    ResultSetMetaData meta=rs.getMetaData();

```

```

int colCount=meta.getColumnCount();
String column[]=new String[colCount];
for (int h=0;h<colCount;h++)
{
column[h]=meta.getColumnName(h+1);
}
for(int i=0;i<m;i++)
{rs.next();
for(int j=0;j<n;j++)
{
a[j][i]=rs.getDouble(column[j]);
}
}
}
}
catch(SQLException e)
{
// if the error message is "out of memory",
// it probably means no database file is found
System.err.println(e.getMessage());
}
finally
{
try
{
if(connection != null)
connection.close();
}
catch(SQLException e)
{
// connection close failed.
System.err.println(e);
}
}
return a;
}

```

public static void write_double(String databasename, String tablei,String column[],double a[][]) throws IOException,ClassNotFoundException

```

{
// load the sqlite-JDBC driver using the current class loader
int n=1,m=1;
int i=0,j=0;
Class.forName("org.sqlite.JDBC");
Connection connection = null;
try
{
// create a database connection
connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
Statement statement = connection.createStatement();
String s="create table "+tablei+"(";
n=column.length;
m=a[0].length;
for(i=0;i<n-1;i++)
{s+=column[i]+" double, ";}
s+=column[i]+" double)";
statement.setQueryTimeout(30); // set timeout to 30 sec.
statement.executeUpdate("drop table if exists "+tablei);
statement.executeUpdate(s);
}
}

```

```

String s2="";
for(j=0;j<a[0].length;j++)
{
s2="";
for(i=0;i<a.length-1;i++)
{s2+=""+a[i][j]+",";}
s2+=""+a[i][j];
String s3="insert into "+tablei+" values("+s2+")";
statement.executeUpdate(s3);
}
}
catch(SQLException e)
{
// if the error message is "out of memory",
// it probably means no database file is found
System.err.println(e.getMessage());
}
finally
{
try
{
if(connection != null)
connection.close();
}
catch(SQLException e)
{
// connection close failed.
System.err.println(e);
}
}
}
}

public static void add_double(String databasename, String tablei,String column[],double a[][]) throws
IOException,ClassNotFoundException
{
// load the sqlite-JDBC driver using the current class loader
int n=1,m=1;
int i=0,j=0;
Class.forName("org.sqlite.JDBC");
Connection connection = null;
try
{
// create a database connection
connection = DriverManager.getConnection("jdbc:sqlite:"+databasename+".db");
Statement statement = connection.createStatement();
/*
String s="create table "+tablei+"(
n=column.length;
m=a[0].length;
for(i=0;i<n-1;i++)
{s+=column[i]+" double, ";}
s+=column[i]+" double)";
statement.setQueryTimeout(30); // set timeout to 30 sec.
statement.executeUpdate("drop table if exists "+tablei);
statement.executeUpdate(s);
*/
String s2="";
for(j=0;j<a[0].length;j++)
{s2="";
for(i=0;i<a.length-1;i++)

```

```

        {s2+=""+a[i][j]+",";}
        s2+=""+a[i][j];
        String s3="insert into "+tablei+" values("+s2+");";
        statement.executeUpdate(s3);
    }
}
catch(SQLException e)
{
    // if the error message is "out of memory",
    // it probably means no database file is found
    System.err.println(e.getMessage());
}
finally
{
    try
    {
        if(connection != null)
            connection.close();
    }
    catch(SQLException e)
    {
        // connection close failed.
        System.err.println(e);
    }
}
}

public static void plot(String databasename, String tablei) throws IOException,ClassNotFoundException
{
    double a[][]=read_double(databasename,tablei);
    Plot pp=new Plot(a);
    pp.plot();
}

public static void plot(String databasename, String tablei,String column[]) throws IOException,ClassNotFoundException
{
    double a[][]=read_double(databasename,tablei,column);
    Plot pp=new Plot(a);
    pp.plot();
}
}
}

```

Program 12.1-7 SQLitetest.java, SQLite sınıfı test programı

```

import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class SQLitetest
{
    public static void main(String arg[]) throws IOException,ClassNotFoundException
    {
        double a[][]=Text.T(Text.readDouble("a.txt"));
        double b[][]=Text.T(Text.readDouble("b.txt"));
    }
}

```

```

String column[]={ "column1", "column2" };
Text.print(Text.T(a),column,"a");
Text.print(Text.T(b),column,"b");
int n=a.length;
int m=a[0].length;
SQLite.write_double("DB1", "tableX",column,a);
SQLite.add_double("DB1", "tableX",column,b);
double c[][]=SQLite.read_double("DB1", "tableX");
Text.print(Text.T(c),column,"c");
SQLite.plot("DB1", "tableX");
}
}

```

column1	column2
0.0	1.29376886
1.0	1.294863598
2.0	1.301698887
3.0	1.301197701
4.0	1.285973527
5.0	1.285973527
6.0	1.285973527
7.0	1.279204108
8.0	1.275875027
9.0	1.278780069
10.0	1.273546229
11.0	1.274211099
12.0	1.274211099
13.0	1.274211099
14.0	1.281326315
15.0	1.266981081
16.0	1.277890023
17.0	1.279962854
18.0	1.289421705
19.0	1.289421705

column1	column2	2	3	4
1.452342	3.576576	2.764648	3.76786785	2.4545

Son örnek olarak profesyonel seviyede bir veri tabanı programı geliştirelim. Bu örneğimizde bir adres defteri oluşturacağız. Adres defterinde isim, soyisim, adres, telefon numarası gibi alanlar bulunacak, verileri istediğimizde değiştirebileceğiz, istersek tüm defteri aynı anda inceleyebileceğiz, istersek istediğimiz kişinin verilerini ad ve soyad bilgilerinden ekrana taşıyabileceğiz. Bu program bir çok küçük sınıfın bir araya getirilmesiyle oluşturulmuştur.

Program 12.1-8 createaddressbook.java, adres defteri alt sınıfı (veri tabanı tablosunu tanımlar)

```

import java.sql.*;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class createaddressbook{
public createaddressbook() throws ClassNotFoundException
{
Class.forName("org.sqlite.JDBC");
try {
Connection conn =DriverManager.getConnection("jdbc:sqlite:adres.db");
Statement stmt = null;
ResultSet rs1 = null;
stmt = conn.createStatement();
String s1;
s1 = "create table adres(" +
" name String, " +
" lastname String, " +
" adres String, " +
" city String, " +
" state String, " +

```



```

        " zipcode String, " +
        " country String, " +
        " email String, " +
        " home_phone String, " +
        " work_phone String, " +
        " mobile_phone String, " +
        " fax String, " +
        " web_adress String, " +
        " home_adress String) if not exist adres";
if (stmt.execute(s1)) {rs1 = stmt.getResultSet();}

} catch (SQLException ex) {
// handle any errors
System.out.println("SQLException: " + ex.getMessage());
System.out.println("SQLState: " + ex.getSQLState());
System.out.println("VendorError: " + ex.getErrorCode());
}}
}

```

Program 12.1-9 adres.java, adres defteri ana sınıfı

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class adres extends JFrame {
    private outputpanel controls;
    private adresspanel scrollArea;
    private JTextArea output;
    private String url;
    private Connection connect;
    private JScrollPane textpane;

    public adres() throws ClassNotFoundException
    { super( "Adres defteri " );
        Class.forName("org.sqlite.JDBC");
        Container c = getContentPane();
        scrollArea = new adresspanel();
        output = new JTextArea( 6, 30 );
        c.setLayout( new BorderLayout() );
        c.add( new JScrollPane( scrollArea ),
            BorderLayout.CENTER );
        textpane = new JScrollPane( output );
        c.add( textpane, BorderLayout.SOUTH );
        try {
            // veritabanı ile bağlantı kur.
            connect = DriverManager.getConnection("jdbc:sqlite:adres.db");
            Statement statement = connect.createStatement();
            statement.setQueryTimeout(2); // set timeout to 2 sec.
            output.append( "Bağlantı sağlandı\n" );
        }
        catch ( SQLException sqllex ) {
            sqllex.printStackTrace();
            output.append( "Bağlantı sağlanamadı\n" +
                sqllex.toString() );
        }
    }
}

```

```

    }
    catch ( Exception ex ) {
        ex.printStackTrace();
        output.append( ex.toString() );
    }
    controls =
        new outputpanel( connect, scrollArea, output);
    c.add( controls, BorderLayout.NORTH );
    setSize( 500, 500 );
    setVisible(true);
}

public static void main( String args[] ) throws ClassNotFoundException
{
    createaddressbook cab=new createaddressbook();
    adres app = new adres();
    app.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
}
}

```

Program 12.1-10 adrespanel.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class adrespanel extends JPanel {
    private JPanel labelPanel, fieldsPanel;
    private String labels[] =
        {
            " ID : ",
            " isim : ",
            " soyisim : ",
            " adres : ",
            " ilçe : ",
            " il : ",
            " zip : ",
            " ülke : ",
            " eposta : ",
            " ev tel : ",
            " iş tel : ",
            " cep tel : ",
            " faks : ",
            " web adresi : ",
            " ev adresi " };
    JTextField id, name, lastname, address, // alan nameleri
        city, state, zipcode, country, email, home_phone,
        work_phone, mobile_phone, fax, web_adress, home_adress;
    public adrespanel()
    { // Label
        labelPanel = new JPanel();
        labelPanel.setLayout(
            new GridLayout( labels.length, 1 ) );
        ImageIcon ii = new ImageIcon( "images/icon.jpg" );
        for ( int i = 0; i < labels.length; i++ )
            labelPanel.add( new JLabel( labels[ i ], ii, 0 ) );
        // TextField panel
        fieldsPanel = new JPanel();
        fieldsPanel.setLayout(
            new GridLayout( labels.length, 1 ) );
    }
}

```

```

id = new JTextField( 20 );
id.setEditable( false );
fieldsPanel.add( id );
name = new JTextField( 20 );
fieldsPanel.add( name );
lastname = new JTextField( 20 );
fieldsPanel.add( lastname );
adres = new JTextField( 20 );
fieldsPanel.add( adres );
city = new JTextField( 20 );
fieldsPanel.add( city );
state = new JTextField( 20 );
fieldsPanel.add( state );
zipcode = new JTextField( 20 );
fieldsPanel.add( zipcode );
country = new JTextField( 20 );
fieldsPanel.add( country );
email = new JTextField( 20 );
fieldsPanel.add( email );
home_phone = new JTextField( 20 );
fieldsPanel.add( home_phone );
work_phone = new JTextField( 20 );
fieldsPanel.add( work_phone );
mobile_phone = new JTextField( 20 );
fieldsPanel.add( mobile_phone );
fax = new JTextField( 20 );
fieldsPanel.add( fax );
web_adress = new JTextField( 20 );
fieldsPanel.add( web_adress );
home_adress = new JTextField( 20 );
fieldsPanel.add( home_adress );
setLayout( new GridLayout( 1, 2 ) );
add( labelPanel );
add( fieldsPanel );
}
}

```

Program 12.1-11 outputpanel.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class outputpanel extends JPanel {
    private JButton findName, addName,
        updateName, clear, help, listall;

    public outputpanel( Connection c, adresspanel s,
        JTextArea t )
    {
        setLayout( new GridLayout( 1, 6 ) );
        // 0
        findName = new JButton( "BUL" );
        findName.addActionListener( new findaddress( c, s, t ) );
        add( findName );
        // 1
        addName = new JButton( "EKLE" );
        addName.addActionListener( new inputadress( c, s, t ) );
        add( addName );
    }
}

```

```

// 2
updateName = new JButton( "DEĞİŞTİR" );
updateName.addActionListener(
    new changeaddress( c, s, t ) );
add( updateName );
// 3
clear = new JButton( "SİL" );
clear.addActionListener( new deleteaddress( s ) );
add( clear );
// 4
help = new JButton( "YARDIM" );
help.addActionListener( new help( t ) );
add( help );
// 5
listall=new JButton("TÜM ADRESLER");
listall.addActionListener(new ActionListener()
{ public void actionPerformed(ActionEvent e)
{ try { SQLightF tt=new SQLightF();
    tt.setVisible(true);
} catch(ClassNotFoundException e1)
{
// eğer hafıza yeterli değil "out of memory",
// mesajı gelmişse veri tabanı bulunamadı demektir.
System.err.println(e1.getMessage());
}
}
});
add(listall);
}
}

```

Program 12.1-12 findaddress.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class findaddress implements ActionListener {
    private adressedpanel fields;
    private JTextArea output;
    private Connection connection;

    public findaddress( Connection c, adressedpanel f,
        JTextArea o )
    {
        connection = c;
        fields = f;
        output = o;
    }

    public void actionPerformed((ActionEvent e)
    {
        try {
            if ( !fields.name.getText().equals( "" ) &&
                fields.lastname.getText().equals( "" ) ) {
                Statement statement =connection.createStatement();
                String query = "SELECT * FROM adres " +
                    "WHERE name = " +
                    fields.name.getText() + """;
            }
        }
    }
}

```

```

output.append( "\nname query is send: " +
    connection.nativeSQL( query )
    + "\n" );
ResultSet rs = statement.executeQuery( query );
display( rs );
output.append( "\n name query s successfull\n" );
statement.close();
}
else if ( !fields.lastname.getText().equals( "" ) &&
fields.name.getText().equals( "" ) ) {
Statement statement =connection.createStatement();
String query = "SELECT * FROM adres " +
    "WHERE lastname = " +
    fields.lastname.getText() + """;
output.append( "\nQuery is send: " +
    connection.nativeSQL( query )
    + "\n" );
ResultSet rs = statement.executeQuery( query );
display( rs );
output.append( "\n lastname query s successfull\n" );
statement.close();
}
else if ( !((fields.lastname.getText().equals( "" )
&& fields.lastname.getText().equals( "" ))) ) {
Statement statement =connection.createStatement();
String query = "SELECT * FROM adres " +
    "WHERE lastname = " +
    fields.lastname.getText() + """+
    " AND name = " +
    fields.name.getText() + """;
output.append( "\n name+lastname Query is sending: " +
    connection.nativeSQL( query )
    + "\n" );
ResultSet rs = statement.executeQuery( query );
display( rs );
output.append( "\n name + lastname Query is successful\n" );
statement.close();
}
else
fields.lastname.setText(
    "Burada soyismi (ve/veya üste ismi) girin ve bul düğmesine basın" );
}
catch ( SQLException sqlx ) {
sqlx.printStackTrace();
output.append( sqlx.toString() );
}
}

public void display( ResultSet rs )
{
try {
rs.next();
//int recordNumber = rs.getInt( 1 );
//if ( recordNumber != 0 )
{
fields.name.setText(rs.getString( 1 ));
fields.lastname.setText(rs.getString( 2 ));
fields.adress.setText(rs.getString( 3 ));
}
}
}

```

```

        fields.city.setText(rs.getString( 4 ));
        fields.state.setText(rs.getString( 5 ));
        fields.zipcode.setText(rs.getString( 6 ));
        fields.country.setText(rs.getString( 7 ));
        fields.email.setText(rs.getString( 8 ));
        fields.home_phone.setText(rs.getString( 9 ));
        fields.work_phone.setText(rs.getString( 10 ));
        fields.mobile_phone.setText(rs.getString( 11 ));
        fields.fax.setText(rs.getString( 12 ));
        fields.web_adress.setText(rs.getString( 13 ));
        fields.home_adress.setText(rs.getString( 14 ));
    }
}
catch ( SQLException sqllex ) {
    sqllex.printStackTrace();
    output.append( sqllex.toString() );
}
}
}
}

```

Program 12.1-13 inputadress.java, adres defteri alt sinifi

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class inputadress implements ActionListener {
    private adresspanel fields;
    private JTextArea output;
    private Connection connection;
    public inputadress( Connection c, adresspanel f,JTextArea o )
    { connection = c;
      fields = f;
      output = o;
    }
    public void actionPerformed((ActionEvent e )
    {
        try {
            Statement statement = connection.createStatement();
            String a[]=new String[14];
            a[0]=fields.name.getText();
            a[1]=fields.lastname.getText();
            a[2]=fields.adress.getText();
            a[3]=fields.city.getText();
            a[4]=fields.state.getText();
            a[5]=fields.zipcode.getText();
            a[6]=fields.country.getText();
            a[7]=fields.email.getText();
            a[8]=fields.home_phone.getText();
            a[9]=fields.work_phone.getText();
            a[10]=fields.mobile_phone.getText();
            a[11]=fields.fax.getText();
            a[12]=fields.web_adress.getText();
            a[13]=fields.home_adress.getText();
            for(int i=0;i<a.length;i++)
            {
                if(a[i].equals("")) a[i]=" ";
            }
        }
    }
}

```

```

if ( !fields.lastname.getText().equals( "" ) &&
!fields.name.getText().equals( "" ) ) {
String query = "INSERT INTO adres (" +
" name," +
" lastname," +
" adres," +
" city," +
" state," +
" zipcode," +
" country," +
" email," +
" home_phone," +
" work_phone," +
" mobile_phone," +
" fax," +
" web_adress," +
" home_adress )" +
" VALUES (";
query+=a[0]+ ",";
query+=a[1]+ ",";
query+=a[2]+ ",";
query+=a[3]+ ",";
query+=a[4]+ ",";
query+=a[5]+ ",";
query+=a[6]+ ",";
query+=a[7]+ ",";
query+=a[8]+ ",";
query+=a[9]+ ",";
query+=a[10]+ ",";
query+=a[11]+ ",";
query+=a[12]+ ",";
query+=a[13]+ ")";
output.append( "\nSQL query is sent: " +
connection.nativeSQL( query )
+ "\n" );
int result = statement.executeUpdate( query );

if ( result == 1 )
output.append( "\nQuery operated successfully\n" );
else {
output.append( "\nQuery gives error\n" );
fields.name.setText("");
fields.lastname.setText("");
fields.adress.setText("");
fields.city.setText("");
fields.state.setText("");
fields.zipcode.setText("");
fields.country.setText(" ");
fields.email.setText("");
fields.home_phone.setText("");
fields.work_phone.setText("");
fields.mobile_phone.setText("");
fields.fax.setText("");
fields.web_adress.setText("");
fields.home_adress.setText("");
}
}
else

```

```

        output.append( "\nat least name and last name should be entered " +
            "push the add button\n" );
        statement.close();
    }
    catch ( SQLException sqlx ) {
        sqlx.printStackTrace();
        output.append( sqlx.toString() );
    }
}
}
}

```

Program 12.1-14 changeadress.java, adres defteri alt simfi

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class changeadress implements ActionListener {
    private adresspanel fields;
    private JTextArea output;
    private Connection connection;

    public changeadress( Connection c, adresspanel f, JTextArea o )
    {
        connection = c;
        fields = f;
        output = o;
    }

    public void actionPerformed((ActionEvent e)
    {
        try {
            Statement statement = connection.createStatement();
            if ( ! fields.id.getText().equals( "" ) )
            {
                String a[]=new String[14];
                a[0]=fields.name.getText();
                a[1]=fields.lastname.getText();
                a[2]=fields.adress.getText();
                a[3]=fields.city.getText();
                a[4]=fields.state.getText();
                a[5]=fields.zipcode.getText();
                a[6]=fields.country.getText();
                a[7]=fields.email.getText();
                a[8]=fields.home_phone.getText();
                a[9]=fields.work_phone.getText();
                a[10]=fields.mobile_phone.getText();
                a[11]=fields.fax.getText();
                a[12]=fields.web_adress.getText();
                a[13]=fields.home_adress.getText();
                for(int i=0;i<a.length;i++)
                    {if(a[i].equals("")) a[i]=" ";}
                if ( !a[1].equals( "" ) &&
                    !a[0].equals( "" ) )
                {
                    String query = "UPDATE adres SET ";
                    query+="name =";query+=fields.name.getText();
                    query+="",lastname =";query+=fields.lastname.getText();

```



```

query+="",adress ="";query+=fields.adress.getText();
query+="",city ="";query+=fields.city.getText();
query+="",state ="";query+=fields.state.getText();
query+="",zipcode ="";query+=fields.zipcode.getText();
query+="",country ="";query+=fields.country.getText();
query+="",email ="";query+=fields.email.getText();
query+="",home_phone ="";query+=fields.home_phone.getText();
query+="",work_phone ="";query+=fields.work_phone.getText();
query+="",mobile_phone ="";query+=fields.mobile_phone.getText();
query+="",fax="";query+=fields.fax.getText();
query+="",web_adress ="";query+=fields.web_adress.getText();
query+="",home_adress ="";query+=fields.home_adress.getText();
query+=" WHERE ID=";query+=fields.id.getText();
output.append( "\nquery is send : " +
connection.nativeSQL( query ) + "\n" );
int result = statement.executeUpdate( query );
if ( result == 1 )
    output.append( "\nkayıt başarı ile değiştirildi\n" );
else {
    output.append( "\nkayıt değiştirme prosesi başarısız oldu\n" );
    fields.name.setText("");
    fields.lastname.setText("");
    fields.adress.setText("");
    fields.city.setText("");
    fields.state.setText("");
    fields.zipcode.setText("");
    fields.country.setText("");
    fields.email.setText("");
    fields.home_phone.setText("");
    fields.work_phone.setText("");
    fields.mobile_phone.setText("");
    fields.fax.setText("");
    fields.web_adress.setText("");
    fields.home_adress.setText("");
    }
statement.close();
} //
} //end of if
else
    output.append( "\nsadece mevcut bulunan kayıtlar değiştirilebilir\n" );
}
catch ( SQLException sqllex ) {
    sqllex.printStackTrace();
    output.append( sqllex.toString() );
}
}
}

```

Program 12.1-15 deleteadress.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;

public class deleteadress implements ActionListener {
    private adresspanel fields;

    public deleteadress( adresspanel f )
    {
        fields = f;
    }
}

```

```

}

public void actionPerformed((ActionEvent e)
{
    fields.name.setText("");
    fields.lastname.setText("");
    fields.adress.setText("");
    fields.city.setText("");
    fields.state.setText("");
    fields.zipcode.setText("");
    fields.country.setText(" ");
    fields.email.setText("");
    fields.home_phone.setText("");
    fields.work_phone.setText("");
    fields.mobile_phone.setText("");
    fields.fax.setText("");
    fields.web_adress.setText("");
    fields.home_adress.setText("");
}
}

```

Program 12.1-16 help.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class help implements ActionListener {
    private JTextArea output;

    public help( JTextArea o )
    { output = o; }

    public void actionPerformed( ActionEvent e )
    {
        output.append( "\nBir kaydı bulmak için isim veya soyismi yazınız ve BUL düğmesine basınız.\n" +
            "Yeni kayıt için EKLE düğmesine basınız\n" +
            "Kayıtlı veriyi değiştirmek için DEĞİŞTİR düğmesine basınız\n" +
            "Veriyi silmek için SİL düğmesine basınız\n" );
    }
}

```

Program 12.1-17 help.java, adres defteri alt sınıfı

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class SQLightF extends JFrame
{
    JTextField hostField;
    JTextField tableField;
    JTextArea queryField;
    QueryTableModel qtm;
    String s="jdbc:sqlite:";

```

```

static final long serialVersionUID = 8597941330711875390L;
public SQLightF() throws ClassNotFoundException
{
    super("adres defteri SQL sorgu tablosu");
    Class.forName("org.sqlite.JDBC");
    Connection connection = null;
    //addWindowListener(new BasicWindowMonitor());
    setDefaultCloseOperation( JFrame.DISPOSE_ON_CLOSE );
    setSize(500,300);
    qtm=new QueryTableModel();
    JTable table=new JTable(qtm);
    JScrollPane scrollpane=new JScrollPane(table);
    JPanel p1=new JPanel();
    p1.setLayout(new GridLayout(3,2));
    p1.add(new JLabel(" veritabanının ismini giriniz :"));
    p1.add(hostField=new JTextField("adres.db"));
    p1.add(new JLabel("çalıştırmak istediğiniz SQL detimini giriniz :"));
    p1.add(queryField=new JTextArea("select distinct * from adres order by name asc"));
    p1.add(new JLabel("SQL komutunu veritabanına göndermek için düğmeye basınız:"));
    JButton jb=new JButton("SQL sorgusunu çalıştır");
    qtm.setHostURL(s+hostField.getText().trim());
    qtm.setQuery(queryField.getText().trim());
    jb.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            qtm.setHostURL(s+hostField.getText().trim());
            qtm.setQuery(queryField.getText().trim());
        }
    }); // jb.addActionlistener
    p1.add(jb);
    getContentPane().add(p1, BorderLayout.NORTH);
    getContentPane().add(scrollpane, BorderLayout.CENTER);
} //DatabaseTest()
public void actionPerformed( ActionEvent e )
{
    try
    {
        SQLight_Table tt=new SQLight_Table();
        tt.setVisible(true);
    } catch(ClassNotFoundException e1)
    {
        // eğer hafıza yeterli değil "out of memory",
        // mesajı gelmişse veri tabanı bulunamadı demektir.
        System.err.println(e1.getMessage());
    }
}
}

```

Adres defteri

BUL	EKLE	DEĞİŞ...	SİL	YARDIM	TÜM A...
ID :					
isim :		irfan			
soyisim :		çoban			
adres :					
ilçe :					
il :					
zip :					
ülke :					
eposta :					
ev tel :					
iş tel :					
cep tel :		5381122233			
faks :					
web adresi :					
ev adresi)					

Bağlantı sağlandı

name query is send: SELECT * FROM adres WHERE name = 'irfan'

name query s successfull

adres defteri SQL sorgu tablosu

veritabanının ismini giriniz : adres.db

çalıştırmak istediğiniz SQL detimini giriniz : select distinct * from adres order by name asc

SQL komutunu veritabanına göndermek için düğmeye basınız:

SQL sorgusunu çalıştır													
name	lastname	adress	city	state	zipcode	country	email	home_pho...	work_phone	mobile_ph...	fax	web_adress	home_adr...
ali	demir									05645 6744			
ali	erdoğdu									3364845			
irfan	çoban									53811222...			
meral	çoban							23233648...					
turhan	çoban									53793333...			

12.2 ALIŞTIRMALAR

1) Karbondioksit sıcaklık ve sabit hacimde özgül ısı değerleri co2_cv.txt d osyasında verilmiştir. veritabanı temp içinde CO2 tablosunu oluşturunuz. Bunun için **create table CO2 (T float, Cv float, TK float);** komutunu kullanabilirsiniz. Yeni bir kolon olarak TK=T+273 kolonunu Update kullanarak oluşturunuz **update CO2 set TK=T+273.15;**

2) x ve y kolonunu kapsayan **sinx** tablosunu oluşturunuz. X değeri olarak 0.1 0.2 .03 0.4 0.5 değerlerini giriniz. Y değeri olarak 0 giriniz. y değerlerini update kullanarak sin(x) değerine değiştiriniz. değiştiriniz.

EX3) : adres programını çalıştır, tüm detaylarıyla 5 adres ekle, bir adresi ara ve bir adresi sil.

12.3 PROBLEMLER

1) **student** tablosunu

name String, surname String, not1 double, not22 double, sonuc double alanlarıyla oluşturunuz, alttaki veriyi giriniz.

'Ali' 'Çiçek' 24 58

'Veli' 'Durmuş' 49 73

'Hasan' 'Yücel' 72 68

'Mehmet' 'Demir' 63 55

Sonucu %40 not1 ve %60 not2 olmak üzere hesaplayınız

2) Alıştırma 1 de CO₂ sıcaklık ve özgül ısı değerlerini girmiştik. Veritabanı CO₂ den değerleri okuyun ve ortalamaları hesaplayarak yazdırın.

13. HESAP ÇİZELGESİ (SPREADSHEET) PROGRAMLARINDAN VERİ OKUMA VE YAZMA

13.1 GENEL TANIMLAR

Microsoft excel, Lotus, OpenOffice calc, StarOffice calc türü matrisel hücreler barındıran bir sayfa üzerinde hesap yapan sistemlere hesap çizelgesi adı verilir. Bu bölümümüzde hesap çizelgelerinden veri okuma ve yazma işlevini inceleyeceğiz. Java dili bu işlem için alt yapı hazırlamışsa da direk olarak hesap çizelgelerinin formatını vermemiştir. Biraz programlama ile bu işlev gerçekleştirilebilir, ancak bir burada serbest yazılım programı olarak internet ortamında sunulan jxl paketini kullanarak bu işlevin kolaylıkla yapılabileceğini düşündük. Piyasada bu işlevi gören bir çok program olmakla beraber, genellikle ücretli programlar olmaları sebebiyle bu paketi seçtik. **jxl** apketi <http://jexcelapi.sourceforge.net/> adresinden kopyalanabilir. Kopyalamadan sonra jxl.jar paketini **JAVA_HOME/jre/lib/ext** dizinine kopyalamamız gerekmektedir. Bunu yapınca bu kütüphanedeki tüm sınıfları rahatlıkla kullanabiliriz.

13.2 HESAP ÇİZELGELERİNDEN VERİ OKUMA

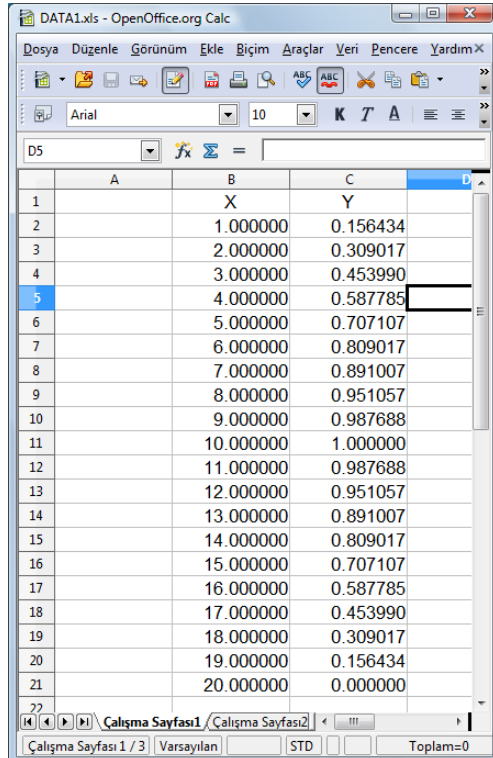
JExcelApi Excel verisini yerel dosyadan veya herhangi bir inputStream den (örneğin internetden) veriyi okuyabilir. Temel olarak ardışık dosya açma ile aynı prensipte çalışır. Data1.xls dosyasından veri okumak için veri kanalı açmak istersek aşağıdaki yapıyı kullanmamız gereklidir.

```
the local filesystem.  
import java.io.File;  
import java.util.Date;  
import jxl.*;
```

....

Workbook workbook = Workbook.getWorkbook(new File("data1.xls")); Assuming to have the

Örnek programımız data1.java da veri data1.xls den okunmaktadır. Hesap çizelgesindeki veri aşağıdaki gibidir.



	A	B	C	D
1		X	Y	
2		1.000000	0.156434	
3		2.000000	0.309017	
4		3.000000	0.453990	
5		4.000000	0.587785	
6		5.000000	0.707107	
7		6.000000	0.809017	
8		7.000000	0.891007	
9		8.000000	0.951057	
10		9.000000	0.987688	
11		10.000000	1.000000	
12		11.000000	0.987688	
13		12.000000	0.951057	
14		13.000000	0.891007	
15		14.000000	0.809017	
16		15.000000	0.707107	
17		16.000000	0.587785	
18		17.000000	0.453990	
19		18.000000	0.309017	
20		19.000000	0.156434	
21		20.000000	0.000000	

Veriyi okumak için kanal açtıktan sonra, hesap çizelgesinin hangi sayfasından veriyi okuyacağımıza karar vermeliyiz. Sayfa indeksleri 0 (sıfır) dan başlamaktadır. Yani ilk sayfanın indeksi 0, ikinci sayfanın indeksi bir dir. Bu API içindeki metodları kullanarak

```
Sheet sheet = workbook.getSheet(0);
```

İstediğimiz sayfaya eriştikten sonra sayfadaki hücre verilerini okumaya başlayabiliriz. Hücre içeriklerini okumak için getContents() metodu kullanırız. Alttaki örnek programda A1 hücresi yazı, B2 hücresi sayısal veri barındırmaktadır. Bu verilere şu şekilde ulaşırız:

Eğer hücrede yazı varsa:

```
Cell a1 = sheet.getCell(0,0);  
String stringa1 = a1.getContents();
```

Eğer hücrede rakam varsa:

```
Cell b2 = sheet.getCell(1,1);  
NumberCell nc = (NumberCell) b2;  
double numberb2 = nc.getValue();
```

Eğer hücrede tarih varsa:

```
Cell c2 = sheet.getCell(2,1);  
Date datec2 = null;  
DateCell dc = (DateCell) c2;  
datec2 = dc.getDate();
```

Örnek programın referansları ile ilgili olarak A1 hücresini (0,0) A2 hücresini (0,1), B1 hücresini (1,0), B2 hücresini (1,1), C1 hücresini (2,0), C2 hücresini (2,1), C3 hücresini (2,2) olarak belirttiğimiz not edelim. Parantesin birinci rakamı kolonları, ikinci rakamı satırları göstermektedir. Programımız veriyi ilk sayfadan okumaktadır.

PROGRAM 13.2.1 Data1 programı, hesap çizelgesi örneği

```
import java.io.*;  
import javax.swing.*;  
import jxl.*;  
import jxl.write.*;  
import jxl.format.*;  
  
//import java.util.Locale;  
  
public class data1  
{  
    public static String toString(double x[],double y[])  
    {  
        String s="";  
        for(int i=0;i<x.length;i++)  
            {s+="x["+i+"][" +x[i]+"] y["+i+"] = "+y[i]+"\\n ";}  
  
        return s;  
    }  
  
    public static void main(String args[]) throws IOException  
    {  
        double x[]=new double[21];  
        double y[]=new double[21];
```

```

try
{
Workbook workbook = Workbook.getWorkbook(new File("data1.xls"));
Sheet sheet = workbook.getSheet(0);

Cell cx;
Cell cy;
for(int i=1;i<x.length;i++)
{
cx=sheet.getCell(1,i);
cy=sheet.getCell(2,i);
NumberCell ncx = (NumberCell) cx;
NumberCell ncy = (NumberCell) cy;
x[i-1]=ncx.getValue();
y[i-1]=ncy.getValue();
}
}
catch(jxl.read.biff.BiffException e1) {};
System.out.println(toString(x,y));
}
}

```

----- Capture Output ----- Capture Output -----

```

> "C:\turhan\java\bin\javaw.exe" data1
x[0][ 1.0 y[0] = 0.15643446504023087
x[1][ 2.0 y[1] = 0.3090169943749474
x[2][ 3.0 y[2] = 0.45399049973954675
x[3][ 4.0 y[3] = 0.5877852522924731
x[4][ 5.0 y[4] = 0.7071067811865475
x[5][ 6.0 y[5] = 0.8090169943749475
x[6][ 7.0 y[6] = 0.8910065241883678
x[7][ 8.0 y[7] = 0.9510565162951535
x[8][ 9.0 y[8] = 0.9876883405951378
x[9][ 10.0 y[9] = 1.0
x[10][ 11.0 y[10] = 0.9876883405951378
x[11][ 12.0 y[11] = 0.9510565162951536
x[12][ 13.0 y[12] = 0.8910065241883679
x[13][ 14.0 y[13] = 0.8090169943749475
x[14][ 15.0 y[14] = 0.7071067811865476
x[15][ 16.0 y[15] = 0.5877852522924732
x[16][ 17.0 y[16] = 0.45399049973954686
x[17][ 18.0 y[17] = 0.3090169943749475
x[18][ 19.0 y[18] = 0.15643446504023098
x[19][ 20.0 y[19] = 1.2246467991473532E-16
x[20][ 0.0 y[20] = 0.0

> Terminated with exit code 0.

```


İkinci öğrenimizde de birinci örneğimizdeki aynı veri okunacak, ama bu sefer bitiş noktası verilmeyecek boş hücre gelince döngü duracaktır. BU program veriyi depolamak için ArrayList kullanmaktadır.

PROGRAM 13.2.2 Data1B programı, hesap çizelgesi örneği

```
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.lang.*;
import jxl.*;
import jxl.write.*;
import jxl.format.*;

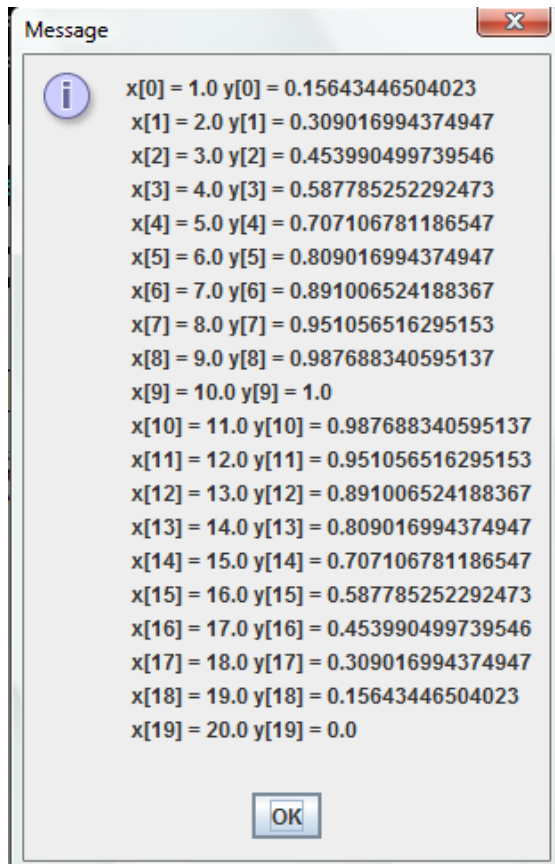
class data
{ double x,y;
  data(double xi,double yi)
  { x=xi;y=yi;}
}

public class data1B
{
  public static String toString(double x[],double y[])
  {
    String s="";
    for(int i=0;i<x.length;i++)
    {s+="x["+i+"] = "+x[i]+" y["+i+"] = "+y[i]+"\\n ";}
    return s;
  }
  public static void main(String args[]) throws IOException
  {
    try
    {
      String stringa1="xx";
      ArrayList<data> b=new ArrayList<data>();
      data d;
      double xi=1;
      double yi;
      Workbook workbook = Workbook.getWorkbook(new File("data1.xls"));
      Sheet sheet = workbook.getSheet(0);
      Cell cx;
      Cell cy;
      int i=1;
      cx=sheet.getCell(1,i);
      cy=sheet.getCell(2,i);
      stringa1=cx.getContents();
      try{
        do
        {
          NumberCell ncx =(NumberCell)cx;
          NumberCell ncy =(NumberCell)cy;
          xi=ncx.getValue();
          yi=ncy.getValue();
          d=new data(xi,yi);
          b.add(d);
          i++;
        }
      }
    }
  }
}
```

```

cx=sheet.getCell(1,i);
cy=sheet.getCell(2,i);
stringa1=cx.getContents();
} while(!stringa1.equals(" "));
}
catch(ArrayIndexOutOfBoundsException ai) {}
catch(ClassCastException ci) {}
double x[]=new double[b.size()];
double y[]=new double[b.size()];
int k=0;
    Iterator ii=b.iterator();
    while(ii.hasNext())
    {d=(data)ii.next();
        Double ax=new Double(d.x);
        Double ay=new Double(d.y);
        x[k]=ax.doubleValue();
        y[k]=ay.doubleValue();
        k++;
    }
OptionPane.showMessageDialog(null,toString(x,y));
}
catch(jxl.read.biff.BiffException e1) {};
}
}

```



Bir sonraki örneğimizde veri ve rakam karışık olarak verilen bir veriyi okuyacağız.

ATA1.xls - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

File Edit View Insert Format Tools Data Window Help

Arial 10

A	B	C	D	E
	Ali	Çiçek	25	58
	Veli	Durmuş	49	73
	Hasan	Yücel	72	68
	Mehmet	Demir	63	55

PROGRAM 13.2.3 Data2 program, an example of reading mixed String and real data from spreadsheet

```

import java.io.*;
import javax.swing.*;
import jxl.*;
import jxl.write.*;
import jxl.format.*;

//import java.util.Locale;

public class data2
{
    public static String toString(String s1[], String s2[],double x[],double y[])
    {
        String s="";
        for(int i=0;i<x.length;i++)
        {s+=s1[i]+" "+s2[i]+" x["+i+"] = "+x[i]+" y["+i+"] = "+y[i]+"\\n ";}

        return s;
    }

    public static void main(String args[]) throws IOException
    {
        String s1[]=new String[4];
        String s2[]=new String[4];
        double x[]=new double[4];
        double y[]=new double[4];

        try
        {
            Workbook workbook = Workbook.getWorkbook(new File("data2.xls"));
            Sheet sheet = workbook.getSheet(1);

            Cell cs1;
            Cell cs2;
            Cell cx;
            Cell cy;
            for(int i=1;i<5;i++)
            {
                cs1=sheet.getCell(1,i);
                cs2=sheet.getCell(2,i);
                cx=sheet.getCell(3,i);
            }
        }
    }
}

```

```

cy=sheet.getCell(4,i);
LabelCell lc1 = (LabelCell) cs1;
LabelCell lc2 = (LabelCell) cs2;
s1[i-1] = lc1.getString();
s2[i-1] = lc2.getString();
NumberCell ncx = (NumberCell) cx;
NumberCell ncy = (NumberCell) cy;
x[i-1]=ncx.getValue();
y[i-1]=ncy.getValue();
}
}
catch(jxl.read.biff.BiffException e1) {};
System.out.println(toString(s1,s2,x,y));
}
}

```

```

----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" data2
Ali Çiçek x[0] = 25.0 y[0] = 58.0
Veli Durmuş x[1] = 49.0 y[1] = 73.0
Hasan Yücel x[2] = 72.0 y[2] = 68.0
Mehmet Demir x[3] = 63.0 y[3] = 55.0

> Terminated with exit code 0.

```

Bir sonraki programımız üniversal olarak veri okuyabilmek için hazırlandı. Bu programda veriyi okumak için çeşitli alt metodlar hazırlanmıştır, böylece her türlü girdi türünü kullanabiliriz.

```

double a[][]=excel.read_double_from_excel("a.xls","Page2","D1",2);
double a[][]=excel.read_double_from_excel("a.xls",1,"D1",2);
double a[][]=excel.read_double_from_excel("a.xls",1,"D1","E20");
double a[][]=excel.read_double_from_excel("a.xls",1,0,0,2,20);
double a[][]=excel.read_double_from_excel("a.xls",1,0,0,2);

```

String çıktı okumak için de

```

String a[][]=read_String_from_excel("a.xls","Page2","D1","E20");
String a[][]=read_String_from_excel("a.xls",1,"D1","E20");
String a[][]=read_String_from_excel("a.xls",1,0,0,2,20);

```

Metodları mevcuttur. Metodlardaki ilk isim hesap çizelgesinin ismidir. İkinci isim veya numara hesap çizelgesinin sayfa ismi veya indeks numarasıdır. Üçüncü ve dördüncü referanslar giriş ve çıkış adreslerinin referanslarıdır.

PROGRAM 13.2.4 excel program, an example of reading real data from spreadsheet

```

import java.io.*;
import javax.swing.*;
import jxl.*;
import jxl.write.*;
import jxl.format.*;
import java.io.File;
import java.util.Hashtable;

```

```

import java.util.Enumeration;
import javax.swing.filechooser.*;
import java.util.*;
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
import java.text.*;
import java.util.Locale;

public class excel
{
public static String[][] read_String_from_excel(String filename, int pageno,int x0,int y0,int nx,int ny)
{
String a[][]=new String[nx][ny];
try{
File f=new File(filename);
jxl.Workbook workbook = jxl.Workbook.getWorkbook(f);
int nos=workbook.getNumberOfSheets();
Sheet sheet= workbook.getSheet(pageno);
Cell a1 = sheet.getCell(x0,y0);
LabelCell lc = (LabelCell)a1;
for(int i=0;i<nx;i++)
{ for(int j=0;j<ny;j++)
{ a1 = sheet.getCell(x0+i,y0+j);
lc = (LabelCell) a1;
a[i][j] = lc.getString();
}
}
} catch(jxl.read.biff.BiffException be) {System.err.println("BiffException");}
catch(IOException bio) {System.err.println("Input output exception");}
return a;
}

public static String[][] read_String_from_excel(String filename, int pageno,String xy0,String xyn)
{int n0[]=excelcode(xy0);
int n1[]=excelcode(xyn);
int nx=n1[0]-n0[0]+1;
int ny=n1[1]-n0[1]+1;
return read_String_from_excel(filename,pageno,n0[0],n0[1],nx,ny);
}

public static String[][] read_String_from_excel(String filename, int pageno,String xy0,int nx,int ny)
{int n0[]=excelcode(xy0);
return read_String_from_excel(filename,pageno,n0[0],n0[1],nx,ny);
}

public static double[][] read_double_from_excel(String filename, String pagetitle,int x0,int y0,int nx,int ny)
{
double a[][]=new double[nx][ny];
try{
File f=new File(filename);
jxl.Workbook workbook = jxl.Workbook.getWorkbook(f);
int nos=workbook.getNumberOfSheets();
String sheets="";
int pageno=0;
Sheet sheet;
for(int i=0;i<nos;i++)
{sheet = workbook.getSheet(i);
sheets=sheet.getName();
}
}
}

```

```

    if(sheets.equals(pagetitle)) {pageno=i;break;}
    }
    sheet= workbook.getSheet(pageno);
    Cell a1 = sheet.getCell(x0,y0);
    NumberCell nc = (NumberCell) a1;
    for(int i=0;i<nx;i++)
    { for(int j=0;j<ny;j++)
    { a1 = sheet.getCell(x0+i,y0+j);
      nc = (NumberCell) a1;
      a[i][j] = nc.getValue();
    }
    }
} catch(jxl.read.biff.BiffException be) {System.err.println("BiffException");}
catch(IOException bio) {System.err.println("Input output exception");}
return a;
}

public static double[][] read_double_from_excel(String filename, String pagetitle,String xy0,String xyn)
{int n0[]=excelcode(xy0);
int n1[]=excelcode(xyn);
int nx=n1[0]-n0[0]+1;
int ny=n1[1]-n0[1]+1;
System.out.println("nx="+nx+"ny="+ny);
return read_double_from_excel(filename,pagetitle,n0[0],n0[1],nx,ny);
}

public static double[][] read_double_from_excel(String filename, String pagetitle,String xy0,int nx,int ny)
{int n0[]=excelcode(xy0);
return read_double_from_excel(filename,pagetitle,n0[0],n0[1],nx,ny);
}

public static double[][] read_double_from_excel(String filename, int pageno,int x0,int y0,int nx,int ny)
{
double a[][]=new double[nx][ny];
try{
File f=new File(filename);
jxl.Workbook workbook = jxl.Workbook.getWorkbook(f);
int nos=workbook.getNumberOfSheets();
Sheet sheet= workbook.getSheet(pageno);
Cell a1 = sheet.getCell(x0,y0);
NumberCell nc = (NumberCell) a1;
for(int i=0;i<nx;i++)
{ for(int j=0;j<ny;j++)
{ a1 = sheet.getCell(x0+i,y0+j);
nc = (NumberCell) a1;
a[i][j] = nc.getValue();
}
}
} catch(jxl.read.biff.BiffException be) {System.err.println("BiffException");}
catch(IOException bio) {System.err.println("Input output exception");}
return a;
}

public static double[][] read_double_from_excel(String filename, int pageno,int x0,int y0,int nx)
{
try{

```

```

File f=new File(filename);
jxl.Workbook workbook = jxl.Workbook.getWorkbook(f);
int nos=workbook.getNumberOfSheets();
Sheet sheet= workbook.getSheet(pageno);
//dolu excel satırlarını say
String stringa1="xx";
Cell a1 = sheet.getCell(x0,y0);
int j=0;
try
{
while(stringa1!="")
{
a1 = sheet.getCell(x0,j);
stringa1 = a1.getContents();
j++;
}
}catch(ArrayIndexOutOfBoundsException ai) { }
//bu excell sayfası için satır sayısı = n
int ny=j-2;
double a[][]=new double[nx][ny];

NumberCell nc = (NumberCell) a1;
for(int i=0;i<nx;i++)
{ for(j=0;j<ny;j++)
{ a1 = sheet.getCell(x0+i,y0+j);
nc = (NumberCell) a1;
a[i][j] = nc.getValue();
}
}
return a;
}catch(jxl.read.biff.BiffException be) {System.err.println("BiffException");}
catch(IOException bio) {System.err.println("Input output exception");}
double b[][]=new double[1][1];
return b;
}

public static double[][] read_double_from_excel(String filename, String pagetitle,int x0,int y0,int nx)
{
try{
File f=new File(filename);
jxl.Workbook workbook = jxl.Workbook.getWorkbook(f);
int nos=workbook.getNumberOfSheets();
String sheets="";
int pageno=0;
Sheet sheet;
for(int i=0;i<nos;i++)
{sheet = workbook.getSheet(i);
sheets=sheet.getName();
if(sheets.equals(pagetitle)) {pageno=i;break;}
}
sheet= workbook.getSheet(pageno);
//dolu excel satırlarını say
String stringa1="xx";
Cell a1 = sheet.getCell(x0,y0);
int j=0;
try
{

```

```

while(stringa1!="")
{
a1 = sheet.getCell(x0,j);
stringa1 = a1.getContents();
j++;
}
}catch(ArrayIndexOutOfBoundsException ai) { }
//bu excell sayfası için satır sayısı = n
int ny=j-2;
double a[][]=new double[nx][ny];

NumberCell nc = (NumberCell) a1;
for(int i=0;i<nx;i++)
{ for(j=0;j<ny;j++)
{ a1 = sheet.getCell(x0+i,y0+j);
nc = (NumberCell) a1;
a[i][j] = nc.getValue();
}
}
return a;
}catch(jxl.read.biff.BiffException be) {System.err.println("BiffException");}
catch(IOException bio) {System.err.println("Input output exception");}
double b[][]=new double[1][1];
return b;
}

public static double[][] read_double_from_excel(String filename, int pageno,String xy0,String xyn)
{int n0[]=excelcode(xy0);
int n1[]=excelcode(xyn);
int nx=n1[0]-n0[0]+1;
int ny=n1[1]-n0[1]+1;
System.out.println("nx="+nx+"ny="+ny);
return read_double_from_excel(filename,pageno,n0[0],n0[1],nx,ny);
}

public static double[][] read_double_from_excel(String filename, int pageno,String xy0,int nx,int ny)
{int n0[]=excelcode(xy0);
return read_double_from_excel(filename,pageno,n0[0],n0[1],nx,ny);
}

public static double[][] read_double_from_excel(String filename, int pageno,String xy0,int nx)
{int n0[]=excelcode(xy0);
return read_double_from_excel(filename,pageno,n0[0],n0[1],nx);
}

public static double[][] read_double_from_excel(String filename, String pagetitle,String xy0,int nx)
{int n0[]=excelcode(xy0);
return read_double_from_excel(filename,pagetitle,n0[0],n0[1],nx);
}

public static boolean isLetter(char x)
{ boolean x1;
if((x>='A' && x<='Z') || (x>='a' && x<='z')) x1=true;
else x1=false;
return x1;
}

public static boolean isNumber(char x)
{ boolean x1;
if(x>='0' && x<='9') x1=true;
}

```



```

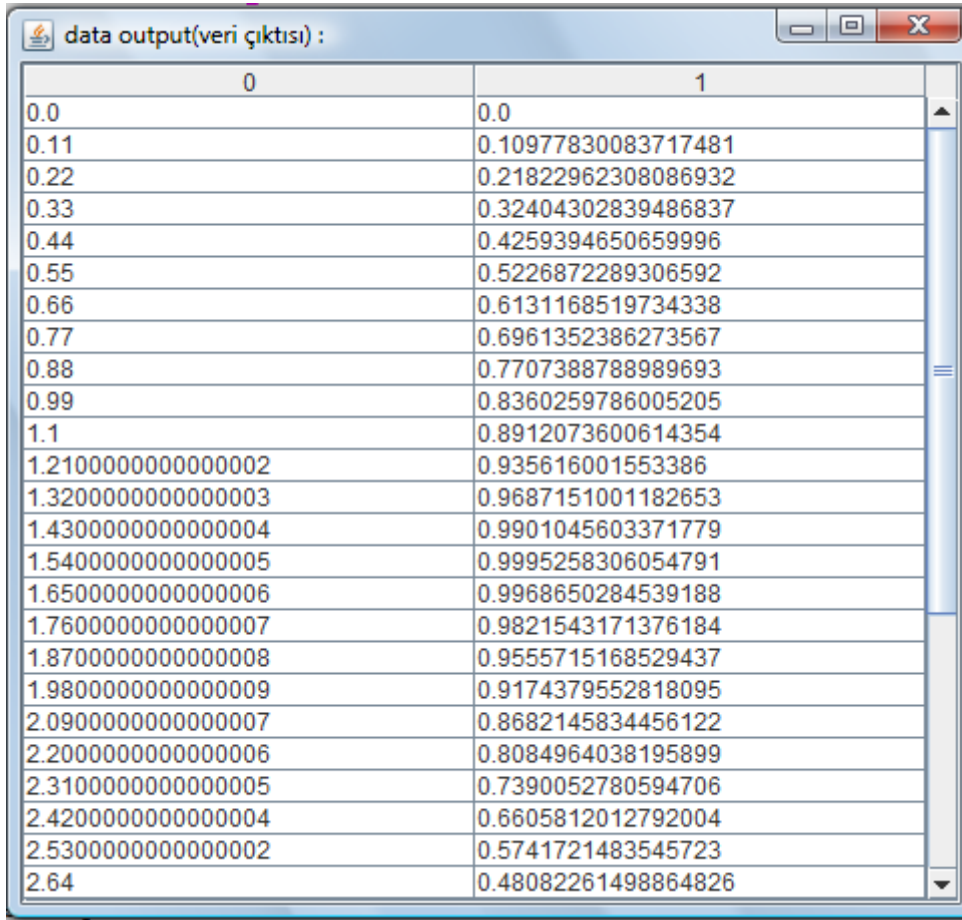
else x1=false;
return x1;
}

public static int charnumber(char x)
{
char b1[]={ 'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};
char b2[]={ 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
for(int i=0;i<26;i++)
{if(x==b1[i] || x==b2[i]) return i;}
return 0;
}

public static int[] excelcode(String s)
{int len = s.length();
int n[]=new int[2];
String s1="";
char a[]=new char[len];
int n1[]=new int[2];
for(int i=0;i<len;i++)
{a[i]=s.charAt(i);
}
if(isLetter(a[0]) && isLetter(a[1]))
{
n1[0]=(charnumber(a[0])+1)*26+charnumber(a[1]);
for(int i=2;i<len;i++) s1+=a[i];
}
else
{n1[0]=charnumber(a[0]);
for(int i=1;i<len;i++) s1+=a[i];
}
n1[1]=Integer.parseInt(s1)-1;
return n1;
}

public static void main(String args[]) throws IOException
{
double a[][]=read_double_from_excel("a.xls","Sayfa2","D1",2);
Text.print(Text.T(a));
}
}

```



0	1
0.0	0.0
0.11	0.10977830083717481
0.22	0.21822962308086932
0.33	0.32404302839486837
0.44	0.4259394650659996
0.55	0.5226872289306592
0.66	0.6131168519734338
0.77	0.6961352386273567
0.88	0.7707388788989693
0.99	0.8360259786005205
1.1	0.8912073600614354
1.2100000000000002	0.935616001553386
1.3200000000000003	0.9687151001182653
1.4300000000000004	0.9901045603371779
1.5400000000000005	0.9995258306054791
1.6500000000000006	0.9968650284539188
1.7600000000000007	0.9821543171376184
1.8700000000000008	0.9555715168529437
1.9800000000000009	0.9174379552818095
2.0900000000000007	0.8682145834456122
2.2000000000000006	0.8084964038195899
2.3100000000000005	0.7390052780594706
2.4200000000000004	0.6605812012792004
2.5300000000000002	0.5741721483545723
2.64	0.48082261498864826

13.3 HESAP ÇİZELGELERİNE YAZMA

Bu bölümde hesap çizelgelerine nasıl yazacağımızı inceleyeceğiz. Yazı yazdırma, okumaya çok benzer metodlarla yapılmaktadır. Temel fark açılan kanalın tanımında odaklanmaktadır. Tabiki yazdırırken yazdırma formatlarının da verilmesi gerekir. Yazı kanalı açmak için **WritableWorkbook** sınıfını kullanıyoruz.

```
import java.io.File;
import java.util.Date;
import jxl.*;
import jxl.write.*;
...
```

```
WritableWorkbook workbook = Workbook.createWorkbook(new File("data3.xls"));
```

Bu hesap çizelgesi dosyası yerel dizinde bulunacak ve data3.xls olarak çağrılacaktır. Bir sonraki adım hangi sayfaya yazdıracağımızın belirlenmesidir. Örneğin “**birinci sayfa**” isimli sayfaya yazmak için

```
WritableSheet sheet = workbook.createSheet("First Sheet", 0);
```

Deyimini kullanırız. Alttaki kısa kod A3 hücresine yazı yazdırmakta ve D5 hücresine de 3.14159 rakamını yazdırmaktadır.

```
Label label = new Label(0, 2, "A label record");  
sheet.addCell(label);
```

```
Number number = new Number(3, 4, 3.1459);  
sheet.addCell(number);
```

Burada yine hücre referansının (sütun, satır) olarak verildiğini yineleyelim. A1 (0,0), B1 (1,0), A2 (0,1) ile gösterilir.

İstenilen satır ve sütunlar eklendikten sonra write deyimi kullanılarak gerçek hesap çizelgelerine yazdırılır. Daha sonra close deyimiyle bağlantı kesilir.

```
workbook.write();  
workbook.close();
```

PROGRAM 17.3.1 Data3 program, hesap çizelgesine yazma örneği

```
import java.io.*;  
import javax.swing.*.*;  
import jxl.*;  
import jxl.write.*;  
import jxl.format.*;  
  
//import java.util.Locale;  
  
public class data3  
{  
    public static String toString(double x[],double y[])  
    {  
        String s="";  
        for(int i=0;i<x.length;i++)  
            {s+="x["+i+"][" +x[i]+"y["+i+"] = "+y[i)+"\n ";}  
        return s;  
    }  
  
    public static void main(String args[]) throws IOException  
    {  
        double x[]=new double[21];  
        double y[]=new double[21];  
  
        try  
        {  
            WritableWorkbook workbook = Workbook.createWorkbook(new File("DATA3.xls"));  
            WritableSheet sheet = workbook.createSheet("First Sheet", 0);  
  
            for(int i=0;i<21;i++)  
            {  
                jxl.write.Number nx = new jxl.write.Number(1, i+1, i);  
                sheet.addCell(nx);  
                jxl.write.Number ny = new jxl.write.Number(2, i+1, Math.sin(i*Math.PI/20));  
                sheet.addCell(ny);  
            }  
            workbook.write();  
            workbook.close();  
        }catch(jxl.write.WriteException we) {System.out.println("write exception");}  
    }  
}
```

	A	B	C	D	E
1					
2		0	0		
3		1	0.156434		
4		2	0.309017		
5		3	0.45399		
6		4	0.587785		
7		5	0.707107		
8		6	0.809017		
9		7	0.891007		
10		8	0.951057		
11		9	0.987688		
12		10	1		
13		11	0.987688		
14		12	0.951057		
15		13	0.891007		
16		14	0.809017		
17		15	0.707107		
18		16	0.587785		
19		17	0.45399		
20		18	0.309017		
21		19	0.156434		
22		20	1.22E-16		
23					
24					
25					

13.4 HESAP ÇİZELGELERİNİ FORMATLAMA

Önceki bölüde verileri hesap çizelgesine yazdık ancak format kullanmadık. Format kullanarak hangi fontla ne büyüklükte yazacağımızı ayarlayabiliriz. Altaki satır 10 Arial fontu oluşturmaktadır.

```
WritableFont arial10font = new WritableFont(WritableFont.ARIAL, 10);  
WritableCellFormat arial10format = new WritableCellFormat (arial10font);  
Label label2 = new Label(1,0, "Arial 10 point label", arial10format);  
sheet.addCell(label2);
```

Hücre fontları paylaşılabilir, birden fazla hücre aynı fontu kullanabilir.

```
Label label3 = new Label(2, 0, "Another Arial 10 point label", arial10format);  
sheet.addCell(label3);
```

Bu formatlar birden fazla yerde kullanılabilirdiğinden bir kere oluşturulduktan sonra değiştirilemez şekilde dizayn edilmişlerdir.

Aşağıdaki örnek 16 piksellik Times-new roman Kalın italik formatı kullanmaktadır. Ve bu fontu D1 hücresinde kullanılmak üzere belirlemektedir.

```
WritableFont times16font = new WritableFont(WritableFont.TIMES, 16, WritableFont.BOLD, true);  
WritableCellFormat times16format = new WritableCellFormat (times16font);
```

```
Label label4 = new Label(3,0, "Times 16 bold italic label", times16format);
sheet.addCell(label4);
```

Rakamsal formatlama da da benzer bir yol izlenir. Bir çok sayısal format hazır olarak da verilmiştir.

```
WritableCellFormat integerFormat = new WritableCellFormat (NumberFormats.INTEGER);
Number number2 = new Number(0, 4, 3.141519, integerFormat);
sheet.addCell(number2);
```

```
WritableCellFormat floatFormat = new WritableCellFormat (NumberFormats.FLOAT);
Number number3 = new Number(1, 4, 3.141519, floatFormat);
sheet.addCell(number3);
```

Üstteki kodlar A5 ve B5 hücrelerine İNTEGER ve FLOAT formatları kullanarak rakam yazdırmaktadır. A5 3 ve B5 3.14 rakamlarını gösterecektir. Biz kendi istediğimiz kadar digit isteyen formatları kendimiz de yaratabiliriz.

```
NumberFormat fivedps = new NumberFormat("#.#####");
WritableCellFormat fivedpsFormat = new WritableCellFormat(fivedps);
Number number4 = new Number(2, 4, 3.141519, fivedpsFormat);
sheet.addCell(number4);
```

Elbette aynı hücre için hangi fontun kullanılacağı da belirtilebilir.

```
WritableCellFormat fivedpsFontFormat = new WritableCellFormat (times16font, fivedps);
Number number5 = new Number(3, 4, 3.141519, fivedpsFontFormat);
sheet.addCell(number5);
```

PROGRAM 13.4.1 Data4 program, fromatlama örneği

```
import java.io.*;
import javax.swing.*;
import jxl.*;
import jxl.write.*;
import jxl.format.*;

//import java.util.Locale;

public class data4
{
    public static String toString(double x[],double y[])
    {
        String s="";
        for(int i=0;i<x.length;i++)
        {s+="x["+i+"][" +x[i]+"y["+i+"] = "+y[i)+"\n ";}

        return s;
    }

    public static void main(String args[]) throws IOException
    {
        double x[]=new double[21];
        double y[]=new double[21];

        try
        {
```

```

WritableWorkbook workbook = Workbook.createWorkbook(new File("DATA4.xls"));
WritableSheet sheet = workbook.createSheet("birinci sayfa", 0);
NumberFormat besdps = new NumberFormat("#.#####");
WritableCellFormat besdpsFormat = new WritableCellFormat(besdps);
for(int i=0;i<21;i++)
{
jxl.write.Number nx = new jxl.write.Number(1, i+1, i, besdpsFormat);
sheet.addCell(nx);
jxl.write.Number ny = new jxl.write.Number(2, i+1, Math.sin(i*Math.PI/20), besdpsFormat);
sheet.addCell(ny);
}
workbook.write();
workbook.close();
} catch(jxl.write.WriteException we) {System.out.println("hatayı yaz");}
}
}

```

	A	B	C	D
1				
2		.	.	
3		1.	.15643	
4		2.	.30902	
5		3.	.45399	
6		4.	.58779	
7		5.	.70711	
8		6.	.80902	
9		7.	.89101	
10		8.	.95106	
11		9.	.98769	
12		10.	1.	
13		11.	.98769	
14		12.	.95106	
15		13.	.89101	
16		14.	.80902	
17		15.	.70711	
18		16.	.58779	
19		17.	.45399	
20		18.	.30902	
21		19.	.15643	
22		20.	.	
23				
24				

PROGRAM 13.4.2 Data5programı, formatlama örneği

```
import java.io.*;
import javax.swing.*;
import jxl.*;
import jxl.write.*;
import jxl.format.*;

//import java.util.Locale;

public class data5
{
    public static String toString(double x[],double y[])
    {
        String s="";
        for(int i=0;i<x.length;i++)
            {s+="x["+i+"][" +x[i]+"y["+i+"] = "+y[i)+"\n ";}

        return s;
    }


    public static void main(String args[]) throws IOException
    {
        double x[]=new double[21];
        double y[]=new double[21];
        jxl.write.WritableImage image;
        File ff=new File("fall.png");

        try
        {
            WritableWorkbook workbook = Workbook.createWorkbook(new File("DATA5.xls"));
            WritableSheet sheet = workbook.createSheet("Birinci sayfa", 0);
            NumberFormat fivedps = new NumberFormat("##.#####");
            WritableCellFormat fivedpsFormat = new WritableCellFormat(fivedps);
            image=new jxl.write.WritableImage(0,0,2,5,ff);
            //WritableCellFormat wcf1=new WritableCellFormat();
            //wcf1.setBorder(jxl.format.Border.RIGHT,jxl.format.BorderLineStyle.THICK);
            WritableCellFormat wcf1=new WritableCellFormat();
            wcf1.setBorder(jxl.format.Border.ALL,jxl.format.BorderLineStyle.THICK);
            jxl.write.Label x_label = new Label(0, 6, "x");
            x_label.setCellFormat(wcf1);
            sheet.addCell(x_label);
            jxl.write.Label y_label = new Label(1, 6, "y");
            y_label.setCellFormat(wcf1);
            sheet.addCell(y_label);

            for(int i=0;i<21;i++)
            {
                jxl.write.Number nx = new jxl.write.Number(0, i+7, i, fivedpsFormat);
                nx.setCellFormat(wcf1);
                sheet.addCell(nx);

                sheet.addImage(image);
                jxl.write.Number ny = new jxl.write.Number(1, i+7, Math.sin(i*Math.PI/20), fivedpsFormat);
                ny.setCellFormat(wcf1);
                sheet.addCell(ny);
            }
            workbook.write();
            workbook.close();
        }
    }
}
```

```
}catch(jxl.write.WriteException we) {System.out.println(yazma hatası");}  
}  
}
```

F20		
	A	B
1		
2		
3		
4		
5		
6		
7	x	y
8	0	0
9	1	0.156434
10	2	0.309017
11	3	0.45399
12	4	0.587785
13	5	0.707107
14	6	0.809017
15	7	0.891007
16	8	0.951057
17	9	0.987688
18	10	1
19	11	0.987688
20	12	0.951057
21	13	0.891007
22	14	0.809017
23	15	0.707107
24	16	0.587785
25	17	0.45399
26	18	0.309017
27	19	0.156434
28	20	1.22E-16

13.5 ALIŖTIRMALAR

1) co2_cv.xls dosyasında sıcaklık ve sabit basınçta özgül enerji verilmiştir. Beriyi oku ve ekrana yazdır.

2) 2 sütunlu sinx tablosunu oluşturun birinci sütunda x ikinci sütunda sin(x) değerleri olsun. X değerleri : 0.1 0.2 .03 0.4 0.5 , enter all y values as sin(x). Sonuçları data6.xls hesap çizelgesine (excel) yazdırın

3) : data6.xls hesap çizelgesine fall.png resmini ekleyin.

13.6 PROBLEMLER

1 : student.xls hesap çizelgesini yaratın ve alttaki veriyi isim , soyisim , not1, not 2, sonuç olarak girin

'Ali' 'Çiçek' 24 58

'Veli' 'Durmuş' 49 73

'Hasan' 'Yücel' 72 68

'Mehmet' 'Demir' 63 55

Sonucu %40 not1 ve %40 not2 olarak hesapladıktan sonra aynı hesap çizelgesine yazdırın.

2) co2_cv.xls , çizelgesinden veriyi okuyun ortalama sıcaklık ve özgül enerjiyi hesaplayarak yazdırın.

14. GERÇEK ZAMAN PROGRAMLAMA

14.1 CPU ZAMAN DİLİMCİKLERİ TREDLERLE PROGRAMLAMA

Gerçek dünyada her şey sırayla gitseydi belki her şey daha düzenli olurdu, ama maalesef gerçek dünyada işlevler birbirine paralel olarak yürümektedir. İnsan vücudu da dahil olmak üzere doğadaki sistemlerin çoğu paralel olarak çalışmaktadır. Bir işlev oluşurken diğer işlevler de ona paralel olarak oluşur. Son yıllarda paralel işlemciler kullanmaya başladılar, ilerki yıllarda çok büyük sayıda paralel işlemcinin bir arada çalışmasını bekleyebiliriz. Birden fazla işlemci olduğunda işlevleri işlemciler arasında paylaşmak mümkündür. Paralel çalışmanın diğer bir yolu da tek bir işlemcide zaman paylaşımıdır. Çoklu kullanım (multithreading) birden fazla programın aynı anda bilgisayar işlemcisinde zaman paylaşımı prensibi üzerinden beraber çalışmasıdır. Ana işlem ünitesinin (Central Processing Unit-CPU) zamanı küçük işlem paketçiklerine(tred) bölünür ve her bir paketçikte ayrı bir program çalıştırılır. Bir program çalışırken diğerleri sırasını bekler. Fakat günümüzdeki bilgisayarlar yeterince hızlı olduklarından bu bekleme işlevini farketmeden biz programların paralel çalıştığı algısını alırız. Bu işlem çoğu dil için bilgisayar işletim sistemi tarafından kontrol edilir. Fakat bu durumda program bir işletim sistemi dışında çalışamaz. Java işletim sistemlerinden bağımsız bir dil olarak bu tred kontrolünü kendi içinde gerçekleştirir. Bunun için Thread ve Runnable sınıflarını kullanır. Tablo 14.1 de değişik programların tred işlem paketçikleri kullanarak çalışması gösterilmiştir.

TABLO 14.1.1 A tred işlem paketçikleri örneği

Tred1	Tred 2	Tred 3	Tred 4	Tred 5	Tred 6	Tred 7	Tred 8	Tred 9
PR1	PR1	PR2	PR3	PR1	PR2	PR2	PR3	PR3

İşlem paketçiklerini kullanan ilk program örneğimiz aşağıda verilmiştir. İşçi sınıfı bir işlem paketçigi örneği. Programdan da görüleceği gibi Thread sınıftan türetilmiştir. Bu sınıfta iki nesne türettik ve start() komutuyla çalıştırmaya başladık. Bu programda iki nesne de aynı önceliğe sahip. Bu yüzden CPU zamanında eşit büyüklükte parçacıklar (eşit zaman) kullanacaklardır.

PROGRAM 14.1.1 İŞÇİ SINIFI

```
class isci extends Thread
{
    public void run()
    { for(int i=0;i<5;i++) {System.out.println(this.getName()+"-->"+i);} }
}

public class C14E1
{
    public static void main(String args[])
    {isci i1=new isci();
    isci i2=new isci();
    i1.start();
    i2.start();
    }
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" C14E1
Thread-0-->0
Thread-1-->0
```

```
Thread-0-->1
Thread-1-->1
Thread-0-->2
Thread-1-->2
Thread-0-->3
Thread-1-->3
Thread-0-->4
Thread-1-->4
```

> Terminated with exit code 0.

İkinci örneğimizde Thread sınıfının sleep metodu kullanılarak değişik Thread nesnelerinin zaman ayarlaması gerçekleştirilecektir. Sleep metodu verilen mikrosaniye kadar nesneyi bekletir.

PROGRAM 14.1.2 Thread sınıfı örneği

```
class isci1 extends Thread
{
String isim;
public isci1(String isim1)
{isim=isim1;}

public void run()
{ for(int i=0;i<5;i++) {
    try
    {
    if(isim.equals("ali")) Thread.sleep(1000);
    if(isim.equals("veli")) Thread.sleep(2000);
    } catch(InterruptedException ex) {System.out.println("error...");}
    System.out.println(this.getName()+"-->"+i);}
}

public class C14E2
{
public static void main(String args[])
{isci1 i1=new isci1("ali");
isci1 i2=new isci1("veli");
i1.start();
i2.start();
}
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" C18E2
Thread-0-->0
Thread-0-->1
Thread-1-->0
Thread-0-->2
Thread-1-->1
Thread-0-->3
Thread-0-->4
Thread-1-->2
Thread-1-->3
Thread-1-->4

> Terminated with exit code 0.
```

Bir sonraki örneğimizde değişik Thread öncelikleri kullanılarak parçacıkların zamanlama kontrolüne gidilmiştir. MIN_PRORITY en yavaş çalışma (en uzun bekleme) yı verir.

PROGRAM 14.1.3 Thread class example with changing thread priorities

```
class Robot extends Thread
{
public Robot(String isim) {super(isim);}
public void run()
{ try{
    for(int i=0;i<10;i++)
    {System.out.println("Priority: "+this.getPriority()+this.getName()+"-->" +i);}
    } catch(Exception ex) {System.out.println("hata..");}
}
}

public class C14E3
{
public static void main(String args[])
{Robot r1=new Robot("A");
 Robot r2=new Robot("B");
 Robot r3=new Robot("C");
 Robot r4=new Robot("D");
 r1.setPriority(Thread.MIN_PRIORITY);
 r2.setPriority(Thread.NORM_PRIORITY);
 r3.setPriority(Thread.MAX_PRIORITY);
 r4.setPriority(Thread.MAX_PRIORITY-2);

 r1.start();
 r2.start();
 r3.start();
 r4.start();
}
}
```

```
----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" C14E3
Priority: 1A-->0
Priority: 10C-->0
Priority: 5B-->0
Priority: 10C-->1
Priority: 8D-->0
Priority: 1A-->1
Priority: 8D-->1
Priority: 10C-->2
Priority: 5B-->1
Priority: 10C-->3
Priority: 8D-->2
Priority: 1A-->2
Priority: 8D-->3
Priority: 10C-->4
Priority: 5B-->2
Priority: 10C-->5
Priority: 8D-->4
Priority: 1A-->3
Priority: 8D-->5
Priority: 10C-->6
```

```
Priority: 5B-->3
Priority: 10C-->7
Priority: 8D-->6
Priority: 1A-->4
Priority: 8D-->7
Priority: 10C-->8
Priority: 5B-->4
Priority: 10C-->9
Priority: 8D-->8
Priority: 1A-->5
Priority: 8D-->9
Priority: 5B-->5
Priority: 1A-->6
Priority: 5B-->6
Priority: 1A-->7
Priority: 5B-->7
Priority: 1A-->8
Priority: 5B-->8
Priority: 1A-->9
Priority: 5B-->9
```

> Terminated with exit code 0.

Şimdi bir bilgisayar saatini Thread.sleep metodu kullanarak oluşturmaya çalışalım. Thread'i 1000 mikrosaniye uyutursak 1 saniye zaman geçmiş olacaktır.

PROGRAM 14.1.4 çok basit bir saat

```
class saat extends Thread
{
int second;
int minute;
long counter;
public saat()
{counter=0;}

public void run()
{ while(true)
{
try
{
second=(int)counter%60;
minute=(int)counter/60;
Thread.sleep(1000);
counter++;
} catch(InterruptedException ex) {System.out.println("error...");}
System.out.println("minute : "+minute+" second : "+second);
}
}
}

public class C14E5
{
public static void main(String args[])
{saat c1=new saat();
c1.start();
}
```

```
}
```

```
----- Capture Output -----  
> "C:\turhan\java\bin\javaw.exe" C14E5  
minute : 0 second : 0  
minute : 0 second : 1  
minute : 0 second : 2  
minute : 0 second : 3  
minute : 0 second : 4  
minute : 0 second : 5  
minute : 0 second : 6  
minute : 0 second : 7  
minute : 0 second : 8  
minute : 0 second : 9  
minute : 0 second : 10  
minute : 0 second : 11  
> Terminated with exit code 1.
```

Bu programı çok daha gelişmiş bir forma sokabiliriz. Genellikle daha hassas zaman sayımı için bilgisayar saatinden (counter) zamanı direk okuyan Date sınıfını kullanırız, böylece bilgisayar proseslerinin aldığı zamandan dolayı hata oluşmamış olur.

PROGRAM 14.1.5 Thread sınıfı örneği saat1

```
import java.awt.*;  
import java.awt.event.*;  
import java.awt.geom.*;  
import javax.swing.*;  
import java.net.*;  
import java.io.*;  
import java.awt.Graphics;  
import java.util.*;  
import java.text.*;  
import java.util.Locale;  
  
class saat1 extends Thread  
{  
    JFrame a;  
    public saat1(JFrame ai)  
    {a=ai;}  
  
    public void run()  
    {  
        while(true)  
        {  
            a.validate();  
            a.repaint();  
            try {Thread.sleep(1000);}  
            catch(InterruptedException e)  
            {System.err.println(e.toString());}  
        }  
    }  
  
    public class C14E6 extends JFrame implements Serializable  
    {  
        JLabel jb;  
        Font F;  
        Date D;
```

```

saat1 r;
GregorianCalendar calender;

public C14E6()
{
super("Digital saat : Bilgisayar Programlama");
Container cc=getContentPane();
jb=new JLabel("");
F=new Font("TimesRoman",Font.BOLD,24);
jb.setFont(F);
// Türkiye standart (Greenwich zamanına göre iki saat ileride)
String[] ids = TimeZone.getAvailableIDs(+3 * 60 * 60 * 1000);
SimpleTimeZone pdt = new SimpleTimeZone(+3 * 60 * 60 * 1000, ids[0]);
calender=new GregorianCalendar(pdt);
D=new Date();
calender.setTime(D);
r=new saat1(this);
cc.add(jb);
r.start();
}

public void paint(Graphics g)
{
D=new Date();
calender.setTime(D);
String s=" ["+
calender.get(Calendar.DATE) + " / "+
(calender.get(Calendar.MONTH)+1) + " / "+
calender.get(Calendar.YEAR) + " ] "+
calender.get(Calendar.HOUR_OF_DAY) + " : "+
calender.get(Calendar.MINUTE) + " : "+
calender.get(Calendar.SECOND);
jb.setText(s);
}

public static void main(String s[])
{
C14E6 f = new C18E6();
f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
f.setSize(400,110);
f.setVisible(true);
}
}

```



Thread sınıfı yerine Runnable interface'i de kullanabiliriz. Thread sınıfı Extends yoluyla eklenen bir sınıftır, ve extends sadece bir sınıf için geçerlidir, eğer başka bir sınıfın daha bağlanması gerekiyorsa Thread sınıfı yerine Runnable interface'i kullanabiliriz. Örnekte bir önceki programın aynısını Runnable interface ile yarattık.

PROGRAM 141.6 Thread sınıf Runnable interphase örneği

```
import java.awt.Graphics;
import java.awt.*;
import java.util.*;
import javax.swing.*;
import java.text.*;
import java.awt.event.*;

class saat2 extends Thread
{
    JFrame a;
    public saat2(JFrame ai)
    {a=ai;}

    public void run()
    {
        while(true)
        {
            a.validate();
            a.repaint();
            try {Thread.sleep(1000);}
            catch(InterruptedException e)
            {System.err.println(e.toString);}
        }
    }
}

public class C14E7 extends JFrame implements Runnable
{
    JLabel jb;
    Font F;
    Date D;
    saat2 r;
    GregorianCalendar calender;
    Thread runner;

    public C14E7()
    {
        super("Digital saat implements Runnable Thread");
        Container cc=getContentPane();
        jb=new JLabel("");
        F=new Font("TimesRoman",Font.BOLD,24);
        jb.setFont(F);
        // Türkiye standart time + 1 hour( 3 hours advanced from Greenwich)
        String[] ids = TimeZone.getAvailableIDs(+3 * 60 * 60 * 1000);
        SimpleTimeZone pdt = new SimpleTimeZone(+3 * 60 * 60 * 1000, ids[0]);
        calender=new GregorianCalendar(pdt);
        D=new Date();
        calender.setTime(D);
        r=new saat2(this);
        cc.add(jb);
        r.start();
    }

    public void start()
    {
        if(runner==null)
    
```



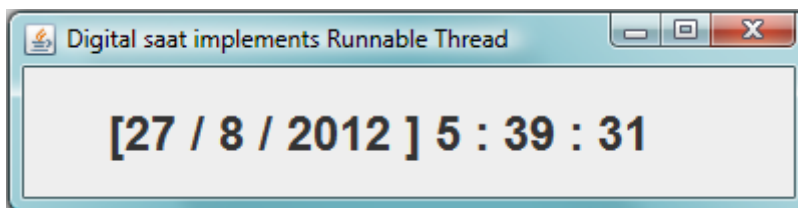
```

    {
        runner=new Thread(this);
        runner.start();
    }
}

public void run()
{
    while(true)
    {
        repaint();
        try {Thread.sleep(1000);}
        catch(InterruptedException e)
        {System.err.println(e.toString());}
    }
}

public void paint(Graphics g)
{
    D=new Date();
    calender.setTime(D);
    String s="  ["+
calender.get(Calendar.DATE) +" / "+
calender.get(Calendar.MONTH)+" / "+
calender.get(Calendar.YEAR) +" ] "+
calender.get(Calendar.HOUR_OF_DAY)+" : "+
calender.get(Calendar.MINUTE)+" : "+
calender.get(Calendar.SECOND);
    jb.setText(s);
}
public static void main(String s[])
{
    C14E7 f= new C14E7();
    f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    f.setSize(400,100);
    f.setVisible(true);
}
}

```



Programımızı bilgisayar grafikleri üzerinden rahatlıkla analog hale de getirebiliriz.

PROGRAM 141.7 Thread sınıf Runnable interphase örneği

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;
public class C14E7A extends JApplet
{
    public void init()

```

```

{ saatP pp= new saatP();
  add(pp);
}
public static void main(String s[])
{
  JFrame f = new JFrame("Saat");
  f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
  JApplet applet = new C14E7A();
  f.getContentPane().add("Center", applet);
  applet.init();
  f.setSize(140,160);
  f.setVisible(true);
}
}

```

PROGRAM 141.8 Thread sınıf Runnable interphase örneği saatP

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;
import java.awt.Graphics;
import java.util.*;
import java.text.*;
import java.util.Locale;
public class saatP extends JPanel implements Runnable
{
double saniye,dakika,saat;
Thread runner;
Date c;

public saatP()
{ c=new Date();
  // Türkiye is 3 saat advanced than Greenwich
  String[] ids = TimeZone.getAvailableIDs(+3 * 60 * 60 * 1000);
  if (ids.length == 0)
  System.exit(0);
  SimpleTimeZone pdt = new SimpleTimeZone(+3 * 60 * 60 * 1000, ids[0]);
  GregorianCalendar calender=new GregorianCalendar(pdt);
  calender.setTime(c);
  saniye=calender.get(Calendar.SECOND );
  dakika=calender.get(Calendar.MINUTE);
  saat=calender.get(Calendar.HOUR);
  runner=new Thread(this);
  runner.start();
}

public Line2D line(double A)
{ double teta=(A*2.0*Math.PI)/60.0;
int xx=(int)(60+50.0*Math.cos(teta));
int yy=(int)(60-50.0*Math.sin(teta));
Line2D x3=new Line2D.Double(60,60,xx,yy);
return x3;
}
}

```

```

public Line2D line1(double A,int x0,int y0,int r1,int r2)
{ double teta=(A*2.0*Math.PI)/60.0;
int xx1=(int)(x0+r1*Math.cos(teta));
int yy1=(int)(y0-r1*Math.sin(teta));
int xx2=(int)(x0+r2*Math.cos(teta));
int yy2=(int)(y0-r2*Math.sin(teta));
Line2D x3=new Line2D.Double(xx1,yy1,xx2,yy2);
return x3;
}

public Line2D[] setTime()
{ //I will use the computer time
c=new Date();
// Türkiye is 3 saat advanced than Greenwich
String[] ids = TimeZone.getAvailableIDs(+3 * 60 * 60 * 1000);
if (ids.length == 0)
System.exit(0);
SimpleTimeZone pdt = new SimpleTimeZone(+3 * 60 * 60 * 1000, ids[0]);
GregorianCalendar calender=new GregorianCalendar(pdt);
saniye=calender.get(Calendar.SECOND );
dakika=calender.get(Calendar.MINUTE);
saat=calender.get(Calendar.HOUR);
Line2D x3[]=new Line2D[3];
double teta1=((15-saniye)*2.0*Math.PI)/60.0;
int xx=(int)(60+50.0*Math.cos(teta1));
int yy=(int)(60-50.0*Math.sin(teta1));
x3[0]=new Line2D.Double(60,60,xx,yy);
double dakika1=(dakika+saniye/60);
double teta2=((15-dakika1)*2.0*Math.PI)/60.0;
xx=(int)(60+40.0*Math.cos(teta2));
yy=(int)(60-40.0*Math.sin(teta2));
x3[1]=new Line2D.Double(60,60,xx,yy);
double saat1=(saat+dakika/60)% 12;
double teta3=((3-saat1)*2.0*Math.PI)/12.0;
xx=(int)(60+30.0*Math.cos(teta3));
yy=(int)(60-30.0*Math.sin(teta3));
x3[2]=new Line2D.Double(60,60,xx,yy);
return x3;
}

public void paint(Graphics g)
{
super.paintComponent(g);
Graphics2D g2=(Graphics2D)g;
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
g2.setStroke(new BasicStroke(1.0f));
Ellipse2D x=new Ellipse2D.Double(10,10,100,100);
g2.setColor(Color.yellow);
g2.fill(x);
g2.setColor(Color.black);
g2.draw(x);
x=new Ellipse2D.Double(5,5,110,110);
g2.draw(x);
Ellipse2D x6=new Ellipse2D.Double(0,0,120,120);
g2.draw(x6);
for(int i=0;i<12;i++)

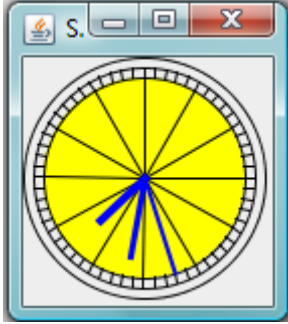
```

```

{ Line2D x2=line(15+5.0*i); g2.draw(x2); }
Line2D x4[]=new Line2D[60];
for(int i=0;i<60;i++)
{ x4[i]=line1((double)i,60,60,50,55);
g2.draw(x4[i]);
}
g2.setColor(Color.blue);
g2.setStroke(new BasicStroke(2.0f));
Line2D x3[]=setTime();
g2.draw(x3[0]);
g2.setStroke(new BasicStroke(3.0f));
g2.draw(x3[1]);
g2.setStroke(new BasicStroke(4.0f));
g2.draw(x3[2]);
}

public void run()
{
while(true)
{
try {Thread.sleep(1000);}
catch(InterruptedException e)
{System.err.println(e.toString());}
repaint();
}}}

```



Örneği biraz daha renkli hale getirmek için yıldızlardan oluşan bir analog saat oluşturabiliriz. Burada iç içe geçmiş sınıf uygulamasının da iyi bir örneğini görmekteyiz. Saat hesaplarında polar sınıfı kullanıldı, ve star sınıfıyla yıldız oluşturuldu.

PROGRAM 14.1.8 Thread sınıf Runnable interphase örneği polar.java

```

public class polar
{
//polar coordinates
protected double R,teta;
public polar()
{R=0;
teta=0;}
public polar(double Ri,double tetai)
{R=Ri;
teta=tetai;}
public polar(polar y)
{R=y.ROku();
teta=y.tetaOku();}
public void R_input(double Ri)
{R=Ri;}
public void teta_input(double tetai)
{teta=tetai;}
}

```

```

public void polar_input(double Ri,double tetai)
{R=Ri;
teta=tetai;}
public void kartesien_input(double xi,double xj)
{R=Math.sqrt(xi*xi+xj*xj);
teta=Math.atan2(xj,xi);}
public double ROku()
{return R;}
public double tetaOku()
{return teta;}
public double xi()
{return R*Math.cos(teta);}
public double xj()
{return R*Math.sin(teta);}
public void topla(polar y)
{kartesien_input((xi()+y.xi()),(xj()+y.xj()));}
public void fark(polar y)
{kartesien_input((xi()-y.xi()),(xj()-y.xj()));}
public polar polarOku()
{return this;}
public boolean esittir(polar v)
{boolean b=((R==v.ROku())&&(teta==v.tetaOku()));
return b;}
public boolean buyuktur(polar v)
{return (this.R>v.ROku());}
public boolean kucuktur(polar v)
{return (this.R<v.ROku());}
public String toString()
{return ""+R+"*exp("+teta+"i) ";}
}

```

PROGRAM 14.1.9 Thread sınıf Runnable interphase örneği yıldiz.java

```

import javax.swing.*;
import java.io.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class yildiz
{
public static void drawYildiz1(Graphics g2,int xi,int yi, int n,int yildizboyu,double angle )
{
// this yildiz yildizts to draw from angle teta=pi/2+angle radian
Graphics2D g=(Graphics2D)g2;
double teta=2.0*Math.PI/n;
double R=yildizboyu;
double r=yildizboyu*0.25;
polar P1=new polar();
polar P2=new polar();
polar P3=new polar();
for(int i=0;i<n;i++)
{
double teta1=teta*i+Math.PI/2.0+angle;
double teta2=teta/2+teta1;
P1.polar_input(R,teta1);
P2.polar_input(r,teta2);
}
}
}

```

```

g.drawLine((xi+(int)P1.xi()),(yi-(int)P1.xj()),
            (xi+(int)P2.xi()),(yi-(int)P2.xj()));
double teta3=teta*(i+1)+Math.PI/2.0+angle;
P3.polar_input(R,teta3);
g.drawLine((xi+(int)P2.xi()),(yi-(int)P2.xj()),
            (xi+(int)P3.xi()),(yi-(int)P3.xj()));
g.drawLine(xi,yi,
            (xi+(int)P1.xi()),(yi-(int)P1.xj()));
g.drawLine(xi,yi,
            (xi+(int)P2.xi()),(yi-(int)P2.xj()));
}
} //drawYildiz1 metodu sonu
public static void drawYildiz1(Graphics g2,int xi,int yi, int n,int yildizboyu )
{
    drawYildiz1(g2,xi,yi,n,yildizboyu,0);
} //drawYildiz1 metodu sonu

public static void drawYildiz(Graphics g2,int xi,int yi, int n,int yildizboyu,double angle )
{
    Graphics2D g=(Graphics2D)g2;
    // bu yıldız çizime teta=pi/2+angle radyandan baslar
    double teta=2.0*Math.PI/n;
    double R=yildizboyu;
    double r=yildizboyu*0.25;
    polar P1=new polar();
    polar P2=new polar();
    polar P3=new polar();
    int x[]=new int[2*n+2];
    int y[]=new int[2*n+2];
    GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
    x.length);

    for(int i=0;i<=n;i++)
    {
        double teta1=teta*i+Math.PI/2.0+angle;
        double teta2=teta/2.0+teta1;
        P1.polar_input(R,teta1);
        P2.polar_input(r,teta2);
        x[2*i]=xi+(int)P1.xi();
        y[2*i]=yi-(int)P1.xj();
        x[2*i+1]=xi+(int)P2.xi();
        y[2*i+1]=yi-(int)P2.xj();
        if(i==n)
        {
            x[2*i]=x[0];
            y[2*i]=y[0];
        }
    }

    for(int i=0;i<=2*n;i++)
    {
        if(i==0)
        {
            polygon.moveTo(x[0],y[0]);
        }
        else
        {
            polygon.lineTo(x[i],y[i]);
        }
    }
}

```

```

    }
}
g.draw(polygon);
} //drawYildiz1 metodu sonu

public static void fillYildiz(Graphics g2,int xi,int yi, int n,int yildizboyu,double angle )
{
    Graphics2D g=(Graphics2D)g2;
    // bu yıldız çizime teta=pi/2+angle radyandan baslar
    double teta=2.0*Math.PI/n;
    double R=yildizboyu;
    double r=yildizboyu*0.25;
    polar P1=new polar();
    polar P2=new polar();
    polar P3=new polar();
    int x[]=new int[2*n+2];
    int y[]=new int[2*n+2];
    GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
    x.length);

    for(int i=0;i<=n;i++)
    {
        double teta1=teta*i+Math.PI/2.0+angle;
        double teta2=teta/2.0+teta1;
        P1.polar_input(R,teta1);
        P2.polar_input(r,teta2);
        x[2*i]=xi+(int)P1.xi();
        y[2*i]=yi-(int)P1.xj();
        x[2*i+1]=xi+(int)P2.xi();
        y[2*i+1]=yi-(int)P2.xj();
        if(i==n)
        {
            x[2*i]=x[0];
            y[2*i]=y[0];
        }
    }

    for(int i=0;i<=2*n;i++)
    {
        if(i==0)
        {
            polygon.moveTo(x[0],y[0]);
        }
        else
        {
            polygon.lineTo(x[i],y[i]);
        }
    }
    g.fill(polygon);
} //fillYildiz metodu sonu

public static void fillAkrep(Graphics g2,int xi,int yi, int n,int yildizboyu,double angle,int i )
{
    // bu yıldız çizime teta=pi/2+angle radyandan baslar
    // yıldızın içini boyar
    Graphics2D g=(Graphics2D)g2;
    double teta=2.0*Math.PI/n;
    double R=yildizboyu;

```

```

double r=yildizboyu*0.25;
polar P1=new polar();
polar P2=new polar();
polar P3=new polar();
int x[]=new int[5];
int y[]=new int[5];
GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
x.length);
x[0]=xi;
y[0]=yi;
x[4]=xi;
y[4]=yi;
double teta1=teta*i+Math.PI/2.0+angle;
double teta2=teta1+teta/2.0;
double teta3=teta1-teta/2+Math.PI*2;
P1.polar_input(R,teta1);
P2.polar_input(r,teta2);
P3.polar_input(r,teta3);
x[1]=xi+(int)P3.xi();
y[1]=yi-(int)P3.xj();
x[2]=xi+(int)P1.xi();
y[2]=yi-(int)P1.xj();
x[3]=xi+(int)P2.xi();
y[3]=yi-(int)P2.xj();
for(i=0;i<5;i++)
{
    if(i==0)
    {
        polygon.moveTo(x[0],y[0]);
    }
    else
    {
        polygon.lineTo(x[i],y[i]);
    }
}
g.fill(polygon);
} //fillAkrep metodu sonu

public static void drawAkrep(Graphics g2,int xi,int yi, int n,int yildizboyu,double angle,int i )
{
    Graphics2D g=(Graphics2D)g2;
    double teta=2.0*Math.PI/n;
    double R=yildizboyu;
    double r=yildizboyu*0.25;
    polar P1=new polar();
    polar P2=new polar();
    polar P3=new polar();
    int x[]=new int[5];
    int y[]=new int[5];
    GeneralPath polygon=new GeneralPath(GeneralPath.WIND_EVEN_ODD,
x.length);
x[0]=xi;
y[0]=yi;
x[4]=xi;
y[4]=yi;
double teta1=teta*i+Math.PI/2.0+angle;
double teta2=teta1+teta/2.0;
double teta3=teta1-teta/2+Math.PI*2;

```



```

P1.polar_input(R,teta1);
P2.polar_input(r,teta2);
P3.polar_input(r,teta3);
x[1]=xi+(int)P3.xi();
y[1]=yi-(int)P3.xj();
x[2]=xi+(int)P1.xi();
y[2]=yi-(int)P1.xj();
x[3]=xi+(int)P2.xi();
y[3]=yi-(int)P2.xj();
for(i=0;i<5;i++)
{
    if(i==0)
    {
        polygon.moveTo(x[0],y[0]);
    }
    else
    {
        polygon.lineTo(x[i],y[i]);
    }
}
g.draw(polygon);

} //drawAkrep metodu sonu

public static void drawDaire(Graphics g2, int xi,int yi, int R)
{
    Graphics2D g=(Graphics2D)g2;
    g.draw(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2*R,2*R));
}

public static void fillDaire(Graphics g2, int xi,int yi, int R)
{
    Graphics2D g=(Graphics2D)g2;
    g.fill(new Ellipse2D.Double((int)(xi-R),(int)(yi-R),2*R,2*R));
}
}

```

PROGRAM 14.1.10 Thread sınıf Runnable interphase örneği yıldiz.java

```

import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;
import java.awt.Graphics;
import java.util.*;
import java.text.*;
import java.util.Locale;

class drawyildiz extends Thread
{
    JPanel a;
    public drawyildiz(JPanel ai)
    {a=ai;}
}

```

```

public void run()
{ while(true)
{ a.repaint();
try { Thread.sleep(1000);}
catch(InterruptedException e)
{ System.err.println(e.toString());}
}}
} //yildizciz2Thread sınıfının sonu

public class analogSaatP extends JPanel
{ drawyildiz y;
double angle1,angle2,angle3;
Date c;
polar p;

public analogSaatP()
{ //super("Analog Saat");
c=new Date();
// Türkiye standart (3 hour advanced than Greenwith time)
String[] ids = TimeZone.getAvailableIDs(+3 * 60 * 60 * 1000);
if (ids.length == 0)
System.exit(0);
SimpleTimeZone pdt = new SimpleTimeZone(+3 * 60 * 60 * 1000, ids[0]);
GregorianCalendar calender=new GregorianCalendar(pdt);
calender.setTime(c);
p=new polar();
setBackground(Color.white);
angle1=-calender.get(Calendar.SECOND)/60.0*Math.PI*2;
angle2=-calender.get(Calendar.MINUTE)/60.0*Math.PI*2;
angle3=-((calender.get(Calendar.HOUR))%12)/12.0*Math.PI*2+angle2/12.0;
y=new drawyildiz(this);
y.start();
}

public void paint(Graphics g)
{
try{
g.setColor(Color.yellow);
yildiz.fillDaire(g,200,200,170);
g.setColor(Color.blue);
yildiz.fillAkrep(g,200,200,16,160,angle2,0);
g.setColor(Color.green);
yildiz.fillAkrep(g,200,200,8,110,angle3,0);
g.setColor(Color.black);
yildiz.fillAkrep(g,200,200,10,60,angle1,0);
g.setColor(Color.black);
yildiz.drawDaire(g,200,200,165);
yildiz.drawDaire(g,200,200,170);
yildiz.drawAkrep(g,200,200,16,160,angle2,0);
yildiz.drawAkrep(g,200,200,8,110,angle3,0);
yildiz.drawYildiz1(g,200,200,10,60,angle1);
angle1-=Math.PI/30.0;
angle2-=Math.PI/1800.0;
angle3-=Math.PI/21600.0;
for(int i=0;i<12;i++)
{
p.polar_input(165,Math.PI*i/6);
g.setColor(Color.magenta);

```

```

        if(i==0||i==3||i==6||i==9||i==12)
        {
            yildiz.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),3,20,0);
            g.setColor(Color.black);
            yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),3,20,0);
        }
        else
        {
            g.setColor(Color.black);
            yildiz.fillYildiz(g,(int)(p.xi()+200),(int)(200-p.xj()),4,10,5);
            //yildiz.drawYildiz1(g,(int)(p.xi()+200),(int)(200-p.xj()),15,10,0);

        }
    }
}

}
catch(NullPointerException e){}
}
}
}

```

PROGRAM 14.1.11 Thread sınıfı Runnable interphase örneği yildiz.java

```

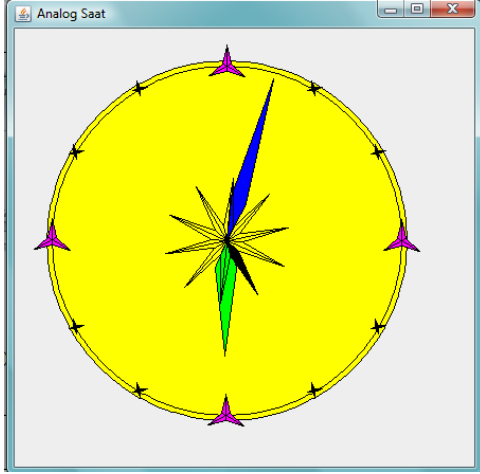
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

public class AnalogSaat extends JFrame
{
    Container c;
    JPanel d;

    public AnalogSaat(String a)
    {
        super(a);
        c=getContentPane();
        d=new analogSaatP();
        c.add(d);
    }

    public static void main(String s[])
    {
        AnalogSaat f = new AnalogSaat("Analog Saat");
        f.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        f.setSize(450,450);
        f.setVisible(true);
    }
}

```



14.2 SYNCHRONISATION (BİRARADA ÇALIŞTIRMA)

Birden fazla Thread bir arada çalışırken eğer aynı bir hafıza birimiyle işlem yapıyorlarsa, gerçek zamanda veriyi alma sırasından emin olamayız. Ortak kullanılan bir veriyi ilk programın oluşturduğu, ikinci programın kullandığını düşünelim. Yeni bir veri oluştuğunda eski veri yok olacaktır, okuma programı bu durumda en son oluşan veriyi okuyacaktır. Bu programın verileri birlikte senkronize şekilde çalışmasını gösteren çeşitli program örnekleri verilmiştir. Programda üç sınıf mevcuttur ortakHucre, IntegerYarat, IntegerKullan, IntegerHarca

14.2-1 OrtakHucre

```
class OrtakHucre{ public static void main(String args[])
{
TutInteger h=new TutInteger();
UretInteger p=new UretInteger(h);
KullanInteger c= new KullanInteger(h);
p.start();
c.start(); }
}
class UretInteger extends Thread
{
private TutInteger pTut;
public UretInteger(TutInteger h)
{
pTut=h;
}
public void run()
{
for(int count=0; count<10; count++)
{
pTut.setOrtakInt(count);
System.out.println("OrtakInteger "+count+" degerini uretti");
try{
sleep( (int) (Math.random()*5000 ) );
}
catch(InterruptedException e)
{
System.err.println("Exeption "+e.toString());
}
}
}
```

```

}
class KullanInteger extends Thread
{
private TutInteger cTut;
public KullanInteger(TutInteger h)
{
cTut=h;
}
public void run()
{
int Intdegeri=0;
while (Intdegeri !=9)
{
try{
sleep( (int) (Math.random()*5000));
}
catch(InterruptedExpection e)
{
System.err.println("Exeption "+e.toString());
}
Intdegeri=cTut.getOrtakInt();
System.out.println("Kullan Integer"+Intdegeri+" degerini okudu");
}
}}
class TutInteger{ private int OrtakInt; public void setOrtakInt(int Intdegeri)
{
OrtakInt=Intdegeri;
}
public int getOrtakInt()
{
return OrtakInt;
}
}

```

```

----- Capture Output -----
> "D:\java\bin\javaw.exe" OrtakHucre
OrtakInteger 0 degerini uretti
OrtakInteger 1 degerini uretti
Kullan Integer1 degerini okudu
Kullan Integer1 degerini okudu
Kullan Integer1 degerini okudu
OrtakInteger 2 degerini uretti
OrtakInteger 3 degerini uretti
OrtakInteger 4 degerini uretti
Kullan Integer4 degerini okudu
OrtakInteger 5 degerini uretti
OrtakInteger 6 degerini uretti
Kullan Integer6 degerini okudu
OrtakInteger 7 degerini uretti
Kullan Integer7 degerini okudu
OrtakInteger 8 degerini uretti
OrtakInteger 9 degerini uretti
Kullan Integer9 degerini okudu
> Terminated with exit code 0.

```

Çıktıdan da görüleceği gibi Kullan Integer 0 değerini hiç görmedi, 1 değerini 3 kere okudu, Kullan Integer 2 ve 3 değerini görmedi. İkinci programımızda Integeri tek bir yere koyma yerine bir buffer hafızası (birden çok yer) kullanacağız böylece sorunu kısmen çözmek mümkün olabilir.

14.2-2 OrtakHucresuf Buffer çok boyutlu hafızası kullanan ortak hücre

```
import java.awt.*;
import java.applet.Applet;
import java.text.DecimalFormat;
public class OrtakHucresuf extends Applet
{
    private TextArea cikti;
    public void init()
    {
        setLayout(new BorderLayout());
        cikti = new TextArea();
        add(cikti, BorderLayout.CENTER);
    }
    public void start()
    {
        Tutinteger h=new Tutinteger(cikti);
        Uretinteger p=new Uretinteger(h);
        Consumeinteger c= new Consumeinteger(h);
        p.start();
        c.start();
    }
}
class Uretinteger extends Thread
{
    private Tutinteger pTut;
    private TextArea cikti;
    public Uretinteger(Tutinteger h)
    {
        pTut=h;
    }
    public void run()
    {
        for(int count=0; count<10; count++)
        {
            pTut.setOrtakint(count);
            try{
                sleep( (int) (Math.random()*500) );
            }
            catch(InterruptedException e)
            {
                System.err.println("Hata : "+e.toString());
            }
        }
    }
}
class Consumeinteger extends Thread
{
    private Tutinteger cTut;
    public Consumeinteger(Tutinteger h)
    {
        cTut=h;
    }
    public void run()
```

```

{
    int intdegeri;
    intdegeri=cTut.getOrtakint();
    while (intdegeri !=9)
    {
        try{
            sleep( (int) (Math.random()*500));
        }
        catch(InterruptedExceotion e)
        {
            System.err.println("Hata "+e.toString());
        }
        intdegeri=cTut.getOrtakint();
    }
}
}
class Tutinteger
{
    private int Ortakint[] = {9,9,9,9,9};
    private boolean writeable = true;
    private boolean okunabilir = false;
    private int YerelOku =0, YerelYaz=0;
    private TextArea cikti;
    public Tutinteger(TextArea out)
    {
        cikti = out;
    }
    public synchronized void setOrtakint(int intdegeri)
    {
        while (!writeable)
        {
            try{
                cikti.append(" Beklemede "+intdegeri);
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Hata :"+e.toString() );
            }
        }
        Ortakint[YerelYaz]=intdegeri;
        okunabilir = true;
        cikti.append("\n Uretilen "+intdegeri+" Yazilan "+YerelYaz);
        YerelYaz = ++YerelYaz % 5;
        cikti.append("\t yazilan hucre "+YerelYaz+"\t okunan hucre "+YerelOku);
        printBuffer(cikti, Ortakint);
        if(YerelYaz==YerelOku)
        {
            writeable=false;
            cikti.append("\nDEPOLAMA YERLERİ DOLU ");
        }
        notify();
    }
    public synchronized int getOrtakint()
    {
        int intdegeri;
        while (!okunabilir)
        {

```

```

try{
    cikti.append(" KULLANMAK iCiN BEKLiYOR ");
    wait();
}
catch (InterruptedException e)
{
    System.err.println("hata:"+e.toString() );
}
}
writeable=true;
intdegeri=Ortakint[YerelOku];
cikti.append("\n Kullanılan "+intdegeri+" Okundugu Hücre "+YerelOku);
YerelOku = ++YerelOku % 5;
cikti.append("\t yazılan hücre "+YerelYaz+"\t okunan hücre "+YerelOku);
printBuffer(cikti, Ortakint);
if (YerelOku==YerelYaz)
{okunabilir=false;
cikti.append("\nİNTEGER DEPOSU BOŞ");
}

notify();
return intdegeri;
}
public void printBuffer(TextArea out,int buf[])
{
    out.append("\t depo: ");
    for(int i=0; i<buf.length; i++)
        out.append(" "+buf[i]);
}
}
}

```

This time 0 is never consumed, 2 and 5 consume twice. Two version of the program will use for synchronized programs. In the first one, an integer array buffer is used, instead of a single common integerBuffer size can be adjusted to secure a completely filled buffer.

14.2-3 CommonCellBuff program, synchronised common variable by using a buffer

```

import java.awt.*;
import java.applet.Applet;
import java.text.DecimalFormat;
import javax.swing.*;
import java.awt.Graphics;
import java.awt.*;
import java.awt.event.*;

class OrtakHucreBuf
{
    public static void main(String args[])
    {
        TutInteger h=new TutInteger();
        UretInteger p=new UretInteger(h);
        KullanInteger c= new KullanInteger(h);
        p.start();
    }
}

```



```

    c.start();
    }
}

class UretInteger extends Thread
{
    private TutInteger pTut;
    private TextArea cikii;

    public UretInteger(TutInteger h)
    {
        pTut=h;
    }

    public void run()
    {
        for(int count=0; count<10; count++)
        {
            pTut.setCommonInteger(count);
            //System.out.println("UretInteger created "+count);

            try{
                sleep( (int) (Math.random()*500) );
            }
            catch(InterruptedExpection e)
            {
                System.err.println("Error: "+e.toString());
            }
        }
    }
}

class KullanInteger extends Thread
{
    private TutInteger cTut;

    public KullanInteger(TutInteger h)
    {
        cTut=h;
    }

    public void run()
    {
        int intValue;

        intValue=cTut.getCommonInteger();

        while (intValue !=9)
        {
            try{
                sleep( (int) (Math.random()*500));
            }
            catch(InterruptedExpection e)
            {
                System.err.println("Error "+e.toString());
            }
            intValue=cTut.getCommonInteger();
        }
    }
}

```

```

        System.out.println("KullanInteger caught "+intValue);
    }
}

class TutInteger
{
    private int Commonint[] = {9,9,9,9,9};
    private boolean writeable = true;
    private boolean readable = false;
    private int LocalRead =0, LocalPrint=0;
    public synchronized void setCommonInteger(int intValue)
    {
        while (!writeable)
        {
            try{
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Error :"+e.toString() );
            }
        }

        Commonint[LocalPrint]=intValue;
        readable = true;
        LocalPrint = ++LocalPrint % 5;
        System.out.println("Üretilen Integer : "+intValue);
        printBuffer(Commonint);
        if(LocalPrint==LocalRead)
        {
            writeable=false;
            System.out.println("\nDEPO DOLU ");
        }
        //notify();
    }

    public synchronized int getCommonInteger()
    {
        int intValue;
        while (!readable)
        {
            try{
                System.out.println(" KULLANILMAK İÇİN BEKLİYOR ");
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Hata:"+e.toString() );
            }
        }

        writeable=true;
        intValue=Commonint[LocalRead];
        LocalRead = ++LocalRead % 5;
        if (LocalRead==LocalPrint)
        {readable=false;

```

```

        System.out.println("\nDEPO BOŞ");
    }
    notify();
    return intValue;
}

public void printBuffer(int buf[])
{
    String s="";
    s+="\t storage: ";
    for(int i=0; i<buf.length; i++)
        s+=" "+buf[i];
    System.out.println(s);
}
}

```

```

----- Capture Output -----
> "D:\java\bin\javaw.exe" OrtakHucresyn
Üretilen Integer : 0
    storage: 0 9 9 9 9

DEPO BOŞ
KULLANILMAK İÇİN BEKLİYOR
Üretilen Integer : 1
    storage: 0 1 9 9 9
Üretilen Integer : 2
    storage: 0 1 2 9 9
Üretilen Integer : 3
    storage: 0 1 2 3 9
Üretilen Integer : 4
    storage: 0 1 2 3 4
Üretilen Integer : 5
    storage: 5 1 2 3 4

DEPO DOLU

```

Bir sonraki program sinkronize versiyondur. Program Integer okunana kadar bekler okuduktan sonra yeni değer oluşturur. Kavşataki bir trafik ışığı gibi çalışmaktadır.

14.2-4 OrtakHucresyn programı

```

class OrtakHucresyn
{
    public static void main(String args[])
    {
        TutInteger h=new TutInteger();
        UretInteger p=new UretInteger(h);
        KullanInteger c= new KullanInteger(h);
        p.start();
        c.start();
    }
}
class UretInteger extends Thread
{
    private TutInteger pTut;
    public UretInteger(TutInteger h)

```

```

    {
        pTut=h;
    }
    public void run()
    {
        for(int count=0; count<10; count++)
        {
            pTut.setOrtakInt(count);
            System.out.println("UretInteger OrtakInteger degerini"+count+"olarak olusturdu");
            try{
                sleep( (int) (Math.random()*3000 ) );
            }
            catch(InterruptedException e)
            {
                System.err.println("Exeption "+e.toString());
            }
        }
    }
}
class KullanInteger extends Thread
{
    private TutInteger cTut;
    public KullanInteger(TutInteger h)
    {
        cTut=h;
    }
    public void run()
    {
        int Intedegeri=0;
        while (Intedegeri !=9)
        {
            try{
                sleep( (int) (Math.random()*3000));
            }
            catch(InterruptedException e)
            {
                System.err.println("Exeption "+e.toString());
            }
            Intedegeri=cTut.getOrtakInt();
            System.out.println("KullanInteger "+Intedegeri+" yakalad □");
        }
    }
}
class TutInteger
{
    private int OrtakInt;
    private boolean Yazilabilir = true;

    public synchronized void setOrtakInt(int Intedegeri)
    {
        while (!Yazilabilir)
        {
            try{
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Exception:"+e.toString() );
            }
        }
    }
}

```

```

    }
    }
    OrtakInt=Intedegeri;
    Yazilabilir=false;
    notify();
    }
    public synchronized int getOrtakInt()
    {
        while (Yazilabilir)
        {
            try{
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Exception:"+e.toString() );
            }
        }

        Yazilabilir=true;
        notify();
        return OrtakInt;
    }
}

```

```

----- Capture Output -----
> "D:\java\bin\javaw.exe" OrtakHucreSyn
UretInteger OrtakInteger degerini 0 olarak olusturdu
KullanInteger 0 yakaladi
UretInteger OrtakInteger degerini 1 olarak olusturdu
KullanInteger 1 yakaladi
UretInteger OrtakInteger degerini 2 olarak olusturdu
KullanInteger 2 yakaladi
UretInteger OrtakInteger degerini 3 olarak olusturdu
KullanInteger 3 yakaladi
UretInteger OrtakInteger degerini 4 olarak olusturdu
KullanInteger 4 yakaladi
UretInteger OrtakInteger degerini 5 olarak olusturdu
KullanInteger 5 yakaladi
UretInteger OrtakInteger degerini 6 olarak olusturdu
KullanInteger 6 yakaladi
UretInteger OrtakInteger degerini 7 olarak olusturdu
KullanInteger 7 yakaladi
UretInteger OrtakInteger degerini 8 olarak olusturdu
KullanInteger 8 yakaladi
UretInteger OrtakInteger degerini 9 olarak olusturdu
KullanInteger 9 yakaladi
> Terminated with exit code 0.

```

Programlardan da görüleceği gibi UretInteger sınıfı (thread'i) TutInteger sınıfını kullanarak yeni bir integer degeri yaratmakta ve KullanInteger sınıfı da bu degeri ortak hafızadan okumaktadır. Burada orijinal programlarda iki thread'in çalışma sıralarının sırayla gitmeme problemi çıkmıştır. Bazen UretInteger programı birden fazla çalışmakta bazende KullanInteger programı birden fazla çalışmaktadır. Bunu onlemek için OrtakHucreSync programında TutInteger sınıfının içinde Uyum kontrolü yapılmıştır. boolean yazilabilir degişkenini bir trafik işareti gibi kullanıp UretInteger ve TutInteger thread programlarının sırasıyla işleme girmesi, bu arada diğlerinin beklemesi sağlanmıştır. Buradaki zorluk. Bazen bir programın çok fazla

bekleyebilmesi olasılığıdır. Bunu engellemek içinde OrtakHucreBuf programında yazmak için bir yerine beş integer yeri tanımlanmış, ve okutma ve yazma işlemlerinde iki ayrı kontrol boolean deyimi kullanılmıştır.

Bir sonraki programda benzer yapı bir havuz problemi olarak verildi. Havuz bir taraftan doldurulmakta, diğertaraftan boşaltılmaktadır. Depo doldurma ve boşaltma öiktarı tesadüfi olarak değişmektedir (sabit değildir)

14.2-5 Pump_Tank_simulation program

```
class Pump_Tank_simulation
{

public static void main(String args[])
{
    Tank h=new Tank();
        Pump1 p1 = new Pump1(h);
        Pump2 p2 = new Pump2(h);
    p1.start();
    p2.start();
}
}

class Pump1 extends Thread
{
    private Tank pTut;
    private double flow_rate;

    public Pump1(Tank h)
    {
        pTut=h;
    }

    public void run()
    {
        while(true)
        {
            while((flow_rate = Math.abs((Math.random()*5))) == 0);
            pTut.addWater(flow_rate);
            try{
                sleep( (int) (Math.random()*3000 ) );
            }
            catch(InterruptedException e)
            {
                System.err.println("Exception "+e.toString());
            }
        }
    }
}

class Pump2 extends Thread
{
    private Tank cTut;
    private double flow_rate;

    public Pump2(Tank h)
    {
        cTut=h;
    }
}
```

```

public void run()
{
    while (true)
    {
        try{
            sleep( (int) (Math.random()*3000));
        }
        catch(InterruptedException e)
        {
            System.err.println("Exeption "+e.toString());
        }
        while((flow_rate = Math.abs((Math.random()*5))) == 0);
        cTut.cikarSu(flow_rate);
    }
}

class Tank
{
    private double TankWater = 0.0;
    private boolean Pump1 = true;
    private boolean Pump2 = false;
    private final double kapasite = 10.0;

    public synchronized void addWater(double amount_of_water)
    {
        while (!Pump1)
        {
            try{
                wait();
            }
            catch (InterruptedException e)
            {
                System.err.println("Exception:"+e.toString() );
            }
        }

        TankWater += amount_of_water;
        if (TankWater >= kapasite){
            Pump1 = false;
            System.out.println("Flow rate of pump 1 " + amount_of_water + " Storage tank is Full(10).");
            TankWater = kapasite;
        }
        else
            System.out.println("Flow rate of pump 1 " + amount_of_water + "water content of Storage tank is " +
TankWater);

        Pump2 = true;
        notify();
    }

    public synchronized void cikarSu(double amount_of_water)
    {
        while (!Pump2)
        {
            try{
                wait();
            }
        }
    }
}

```

```

    }
    catch (InterruptedException e)
    {
        System.err.println("Exception:"+e.toString() );
    }
}

TankWater -= amount_of_water;
    if (TankWater <= 0)
    {
        Pump2=false;
            System.out.println("Flow rate of pump 2 " + amount_of_water + " Storage tank is empty(0)");
            TankWater = 0;
        }
    else
        System.out.println("Flow rate of pump 2 " + amount_of_water + "water content of Storage tank is " +
TankWater);
        Pump1 = true;
    notify();
}
}

```

```

----- Capture Output -----
> "C:\turhan\java\bin\javaw.exe" Pump_Tank_simulation
Flow rate of pump 1 2.3467302907416965water content of Storage tank is 2.3467302907416965
Flow rate of pump 2 4.561512486251294 Storage tank is empty(0)
Flow rate of pump 1 0.5867555472676406water content of Storage tank is 0.5867555472676406
Flow rate of pump 2 2.600533616696703 Storage tank is empty(0)
Flow rate of pump 1 3.80887023404572water content of Storage tank is 3.80887023404572
Flow rate of pump 1 3.9397103967673575water content of Storage tank is 7.748580630813077
Flow rate of pump 2 0.9527778587374092water content of Storage tank is 6.7958027720756675
Flow rate of pump 1 0.6006220170762883water content of Storage tank is 7.396424789151956
Flow rate of pump 1 0.021814412492205393water content of Storage tank is 7.418239201644162
Flow rate of pump 2 2.8249175749160464water content of Storage tank is 4.593321626728115
Flow rate of pump 1 0.5102548266882029water content of Storage tank is 5.103576453416318
Flow rate of pump 1 0.5141968938471103water content of Storage tank is 5.617773347263428
Flow rate of pump 1 2.1940998083020187water content of Storage tank is 7.811873155565447
Flow rate of pump 1 2.42826921346704 Storage tank is Full(10).
Flow rate of pump 2 2.2087842517393335water content of Storage tank is 7.791215748260667

```

14.3 ALIŞTIRMALAR

- 1) C14E5 programındaki saat sınıfını inceleyiniz. Programı saat değerini de yazacak şekilde getiriniz.
- 2) Pump_Tank_simulation programını inceleyiniz. Havuz hacmini 50 m³ e değiştirerek programı tekrar çalıştırınız.
- 3) saatP ve C14E7A programlarını inceleyin ve bir jar programına dönüştürün . **HOMEWORK**

14.4 PROBLEMLER

- 1) Isı transferi simülasyonu incelenecektir. Dış havasıcaklığı bir sinüs fonksiyonu olarak oluşturulacak ve belli tesadüfi salınımlar içerecektir. Da içi sıcaklık 20 C olarak

alınacaktır. Program anlık ısı kayıplarını ve toplam ısı kaybını hesaplayacaktır. Isı transferi denklemini $Q=10(T_{oda} - T_{çevre})$ Watt

$$T_{çevre} = \frac{T_{maksimum} + T_{minimum}}{2} + \frac{T_{maksimum} + T_{minimum}}{2} * \sin\left(\frac{t * (1 - 0.1 * tesadüfü_sayı)}{3600 * 24}\right)$$

$$Enerji = \sum_{t=0}^{24saat} Q(t) * \Delta t$$

Burada Δt her tred dönemindeki zaman farkıdır. Programı kontrol için $T_{minimum}=0$ C ve $T_{maksimum}=20$ C alabiliriz.

15. JAVA PROGRAMLARINI BAŞKA JAVA PROGRAMLARI İÇİNDE DERLEME VE GERÇEK ZAMANDA ÇALIŞTIRMA

15.1 JAVA PROGRAMLARINI GERÇEK ZAMANDA DERLEME

Java derleyicisi java programlarının içinde çalıştırılabilir. Java derleyicisini çalıştırmak için alttaki kod kullanabiliriz.

```
try{
    JavaCompiler javac = ToolProvider.getSystemJavaCompiler();
    int rc = javac.run(null,null,null,fileName);}
catch(NullPointerException e1) {System.err.println("NULL POINTER");}
```

Bu deyimdeki javac java derleyici nesnesinin ismidir.

ToolProvider.getSystemJavaCompiler(); java derleyicisinin yerini saptar.

javac.run(null,null,null,dosyaismi); deyim “dosya ismi” dosyasını derler. Ve class dosyasını yaratır. İlk örnek olarak bir **hosgeldiniz** programı yaratalım.

Program 15.1-1 HosgeldinizGZ.java example class

```
class HosgeldinizGZ
{
public HosgeldinizGZ()
{
System.out.println("Java Sinifina Hos Geldiniz!");
}
}
```

Program 19.1-2 gercekszaman.java example class

```
import javax.swing.table.*;
import static java.lang.Math.*;
import java.awt.image.*;
import javax.swing.*;
import java.util.Locale;
import java.text.*;
import java.util.*;
import javax.tools.*;

public class gercekszaman
{
    public static void compileandrun(String fileName)
    {
        try{
            JavaCompiler javac = ToolProvider.getSystemJavaCompiler();
            int rc = javac.run(null,null,null,fileName);}
        catch(NullPointerException e1) {System.err.println("NULL POINTER");}
    }
    public static void main(String arg[])
    {
        //String fileName=JOptionPane.showInputDialog("Enter file name (a.txt): ");
        compileandrun("HosgeldinizGZ.java");
        HosgeldinizGZ x=new HosgeldinizGZ();
    }
}
```

HosgeldinizGZ programının gercekszaman.java programı içinde derlendiğinden emin olmak için ilk iş olarak HosgeldinizGZ.class dosyasını silelim.

```
C:\Windows\system32\cmd.exe
G:\OKUL\Computer Programming I\javaturkce>dir HosgeldinizGZ.class
G sürücüsü birimi: TURHAN_32G
Birim Seri Numarası: 5AD2-8A4D

G:\OKUL\Computer Programming I\javaturkce dizini
27.09.2012 13:07          380 HosgeldinizGZ.class
                1 Dosya          380 bayt
                0 Dizin        1 173 028 864 bayt boş

G:\OKUL\Computer Programming I\javaturkce>del HosgeldinizGZ.class
G:\OKUL\Computer Programming I\javaturkce>dir HosgeldinizGZ.class
G sürücüsü birimi: TURHAN_32G
Birim Seri Numarası: 5AD2-8A4D

G:\OKUL\Computer Programming I\javaturkce dizini
Dosya Bulunamadı
G:\OKUL\Computer Programming I\javaturkce>
```

Ve bundan sonra gercekszaman programını çalıştıralım

```
----- Capture Output -----
> "D:\java\bin\javaw.exe" gercekszaman
Java Sinifina Hos Geldiniz!
> Terminated with exit code 0.
```

Görüldüğü gibi HosgeldiniGZ önce derlenmiş, sonra da sonucu program içinde alınmıştır. Burada bir noktaya dikkatinizi çekelim, bir programın içinde diğer sınıfı çağırmak için çalıştıracağımız kodları kurucu metod içine çektik.

15.2 PROGRAM YAZARAK GERÇEK ZAMANDA DERLEYEREK ÇALIŞTIRMAK

Java program kodları temel olarak yazı dosyalarıdır. Ve bu sebeple başka bir program tarafından oluşturulabilirler. Program kodu dosyaya yazıldıktan sonra da java programı tarafından derlenerek çalıştırılabilir, böylece tüm proses programın çalışması esnasında oluşabilir. Buna örnek olarak aşağıdaki program kodları verilmiştir.

Program 15.2-1 gercekszaman3A, f2D.java dosyasını yaratır.

```
import java.io.*;
import javax.swing.*;
import javax.tools.*;
import java.util.*;
import java.lang.Integer;
import java.awt.*;
import java.awt.event.*;
import java.awt.font.*;
import java.awt.geom.*;
```

```

import java.awt.image.*;
import javax.swing.*;
import java.util.Locale;
import java.text.*;
import java.util.*;
import javax.swing.table.*;
import static java.lang.Math.*;

public class gercekzaman3A
{
    public static void compile(String name)
    {
        try{
            JavaCompiler javac = ToolProvider.getSystemJavaCompiler();
            int rc = javac.run(null,null,null,name);}
            catch(NullPointerException e1) {System.err.println("NULL POINTER");}
        }

    public static void createfunction(String name,String function)
    {
        String s="";
        String name1=name+".java";
        try {PrintWriter ffout=new PrintWriter(new BufferedWriter(new FileWriter(name1)));
        s+="class "+name+" extends f_x\n";
        s+=" {\n";
        s+=" public double func (double x)\n";
        s+=" { double y="+function+"\n";
        s+=" return y;\n";
        s+=" }\n";
        s+=" }\n";
        ffout.println(s);
        //System.out.println(s);
        ffout.close();
        }
        catch(IOException e1) {System.err.println("girdi hatası.);}
        catch(NumberFormatException e2){};
    }

    public gercekzaman3A()
    {
        String fileName="f2D";
        String function=JOptionPane.showInputDialog("Enter function statement f(x) ");
        createfunction(fileName,function);
        compile(fileName+".java");
    }
}

```

Program 15.2-2 gercekzaman3B, f2D.java dosyasını çalıştırarak değer hesaplar

```

import static java.lang.Math.*;

public class gercekzaman3B
{
    public gercekzaman3B()
    {
        f2D ff=new f2D();
        double x=Double.parseDouble(JOptionPane.showInputDialog("x = "));
        System.out.println("x="+x+"y="+ff.func(x));
    }
}

```

Program 19.2-3 realtime3B evaluating the value of created function fa.java

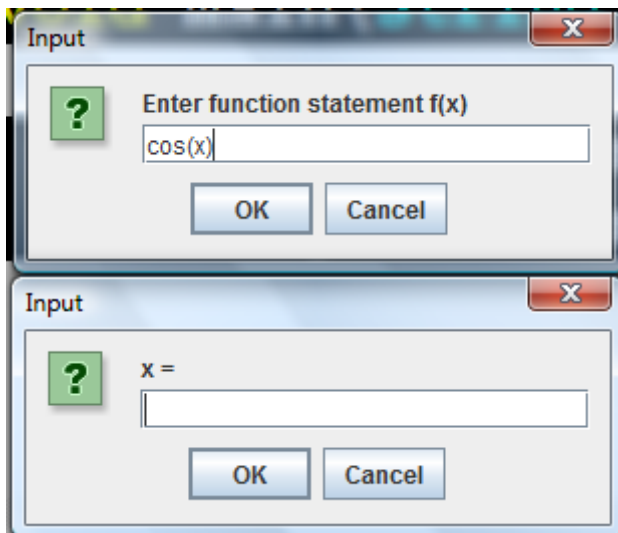
```
import java.awt.image.*;
import javax.swing.*;
import java.util.Locale;
import java.text.*;
import java.util.*;
import javax.swing.table.*;
import static java.lang.Math.*;

public class realtime3B
{
    public realtime3B()
    {
        fa ff=new fa();
        double x=Double.parseDouble(JOptionPane.showInputDialog("x = "));
        System.out.println("x="+x+"y="+ff.func(x));
    }
}
```

Program 15.2-3 gercekszaman3C gercekszaman3A ve gercekszaman3B sınıflarını çağırır.

```
import static java.lang.Math.*;

public class gercekszaman3C
{
    public static void main(String arg[])
    {
        gercekszaman3A x1=new gercekszaman3A();
        gercekszaman3B x2=new gercekszaman3B();
    }
}
```



----- Capture Output -----

```
> "D:\java\bin\javaw.exe" gercekszaman3C
x=1.0 y=0.5403023058681398
> Terminated with exit code 0.
```

Bir sonraki örnekte benzer bir problem çalıştırılacak, fakat bu sefer fonksiyonu değil, türevini hesaplayacağız.

Program 15.2-4 sayisal.java sayisal analiz programı

```
public class sayisal
{

public static double turev(f_x f,double x)
{ double h=0.00001;
return (-f.func(x+2.0*h)+8.0*f.func(x+h)-8.0*f.func(x-h)+f.func(x-2.0*h))/(12.0*h);
}

public static double integral(f_x f_xnt,double a,double b)
{
// integral f(x) dx
double r[]={-0.973906528517171,-0.865063366688984,-0.679409568299024,-0.433395394129247,-
0.148874338981631,
0.148874338981631,0.433395394129247,0.679409568299024,0.865063366688984,0.973906528517
171};
double
c[]={0.066671344308684,0.149451349150580,0.219086362515982,0.269266719309996,0.29552422
4714752,
0.295524224714752,0.269266719309996,0.219086362515982,0.149451349150580,0.066671344308
684};
double z=0,x,y;
double k1=(b-a)/2.0;
double k2=(b+a)/2.0;
for(int i=0;i<r.length;i++)
{
x=k2+k1*r[i];
y=f_xnt.func(x);
z+=k1*c[i]*y;
}
return z;
}
}
```

Program 15.2-5 gercekszaman4B

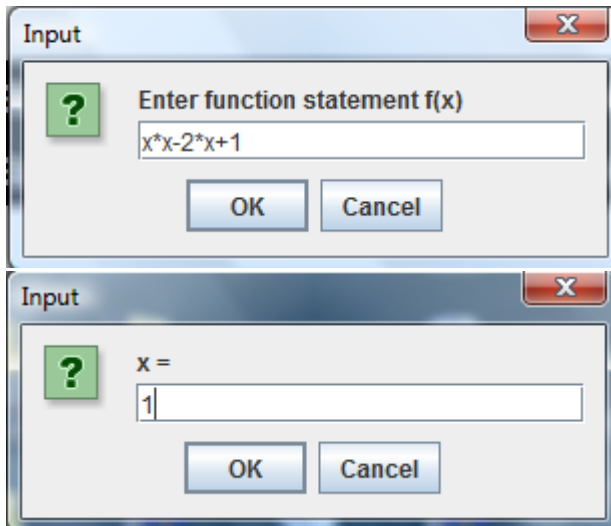
```
import static java.lang.Math.*;

public class gercekszaman4B
{
public gercekszaman4B()
{
f2D ff=new f2D();
double x=Double.parseDouble(JOptionPane.showInputDialog("x = "));
System.out.println("x = "+x+" y = "+ff.func(x)+"türev = "+sayisal.turev(ff,x));
}
}
```

```
}  
}
```

Program 15.2-6 gercekszaman4C

```
import java.io.*;  
import javax.swing.*;  
import javax.tools.*;  
import java.util.*;  
import java.lang.Integer;  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.font.*;  
import java.awt.geom.*;  
import java.awt.image.*;  
import javax.swing.*;  
import java.util.Locale;  
import java.text.*;  
import java.util.*;  
import javax.swing.table.*;  
import static java.lang.Math.*;  
  
public class gercekszaman4C  
{  
    public static void main(String arg[])  
    {  
        gercekszaman3A x1=new gercekszaman3A();  
        gercekszaman4B x2=new gercekszaman4B();  
    }  
}
```



```
----- Capture Output -----  
> "D:\java\bin\javaw.exe" gercekszaman4C  
x = 1.0 y = 0.0türev = 0.0  
> Terminated with exit code 0.
```

15.3 ALIŞTIRMALAR

1) **sayisal.java** programı yukarda verilmişti ve programda integral metodu da vardı. Gerçek zamada fonksiyonu yazarak integralini alacak bir program yazınız.

PROBLEMLER

- 1) Alttaki program seti diferansiyel denklem setini gerçek zamanda çözmek için dizayn edilmiştir. Bu program Runge-Kutta 6 diferansiyel çözüm yöntemi kullanmaktadır.

public static double[][] RK6(f_xy f,double x0,double y0,double xn,double h)

Bu metodu yine verilmiş olan

public static double[][] RKF6(f_xy f,double x0,double y0,double xn,double h,double eps)

metoduyla değiştirerek

$$\frac{dy}{dx} = 0.5 - 10y^2$$

x=0 y(0)=0.5 x=0 dan 4 e kadar h=0.1 sınırlıya çözünüz

fonksiyonuretcil

```
import java.io.*;
import javax.swing.*;
import javax.tools.*;
import java.util.*;
import java.lang.Math.*;
import java.lang.*;

abstract class f_xy
{
    //single function single independent variable
    // example f=x*x
    abstract double func(double x,double y);
}

class fonksiyonuretcil
{
    String fonksiyon;
    String dosyaismi;
    String sinif;
    PrintStream fout;
    PrintStream ferr;

    public fonksiyonuretcil(String dosyaismi,String fonksiyon)
    { this.fonksiyon=fonksiyon;
      this.dosyaismi=dosyaismi;
      try {
          fout=new PrintStream(new FileOutputStream("print.txt"));
          ferr=new PrintStream(new FileOutputStream("error.txt"));
      } catch(FileNotFoundException e1) {System.err.println("file is not found");}
    }

    public fonksiyonuretcil(String fonksiyon)
    {this.fonksiyon=fonksiyon;
     dosyaismi="fDEq";
    }

    public void addfunction(String fonksiyon)
    {this.fonksiyon=fonksiyon;}

    public String toString()
    {return fonksiyon;}
}
```



```

public fonksiyonuretici1()
{ fonksiyon="x*x*y";
  dosyaismi="fDEq";
}

public String createfunction()
{String isim=dosyaismi+".java";
String s="";
try {PrintWriter ffout=new PrintWriter(new BufferedWriter(new FileWriter(isim)));
s+="class "+dosyaismi+" extends f_xy\n";
s+=" {\n";
s+=" public double func (double x,double y)\n";
s+=" { double z="+fonksiyon+"\n";
s+=" return z;\n";
s+=" }\n";
s+=" }\n";
sinif=s;
ffout.println(s);
System.out.println(s);
ffout.close();
}
catch(IOException e1) {System.err.println("input error.");}
catch(NumberFormatException e2){};
//compiling it
try{
JavaCompiler javac = ToolProvider.getSystemJavaCompiler();
int rc = javac.run(null,null,null,isim);}
catch(NullPointerException e1) {System.err.println("NULL POINTER");}
return fonksiyon;
}

protected void finalize() throws Throwable
{super.finalize();}
}

```

girdiFDE.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.event.*;
import java.io.*;
import java.util.*;
public class girdiFDE extends Thread implements ActionListener
{
    Container c;
    String s=" ";
    String s1="";
    fonksiyonuretici1 fg;
    JTextArea jt;
    JButton bb;
    boolean basildi;

    public girdiFDE(String isim,String fonksiyon)
    {
        jt=new JTextArea(s,5,30);
        jt.setFont(new Font("SansSerif",Font.PLAIN,20));
    }
}

```

```

        fg=new fonksiyonuretici1();
        bb=new JButton("push button after entering the function into to the window : x");
        JScrollPane jsp=new JScrollPane(jt);
        JFrame cerceve=new JFrame("function girdi window example:x*x*y-y");
        cerceve.addWindowListener(new BasicWindowMonitor());
        c=cerceve.getContentPane();
        c.setLayout(new BorderLayout());
        c.add(jsp,BorderLayout.NORTH);
        c.add(bb,BorderLayout.SOUTH);
        bb.addActionListener(this);
        cerceve.pack();
        cerceve.setVisible(true);
        basildi=false;
    }

    public String vericiktisi()
    {
        while(!basildi)
        {
            try {Thread.sleep(100);}
            catch(InterruptedException e) {System.err.println(e.toString());}
        }
        return s1;
    }

    protected void finalize() throws Throwable
    {super.finalize();}

    public void actionPerformed( ActionEvent e)
    {
        if(e.getSource()==bb)
        {s=jt.getText();fg.addfunction(s);s1=fg.createfunction();
        basildi=true;
        try{fg.finalize();finalize();
        } catch(Throwable e1) {System.err.println("girdi hatası.");}
        }
    }
}

```

de1.java

```

import java.io.*;
import javax.swing.*;
import javax.tools.*;
import java.util.*;
import java.lang.Integer;
import java.awt.*;
import java.awt.event.*;
import java.awt.font.*;
import java.awt.geom.*;
import java.awt.image.*;
import javax.swing.*;
import java.util.Locale;
import java.text.*;
import java.util.*;
import javax.swing.table.*;
import static java.lang.Math.*;

class de1

```

```

{
    String s;
    public de1()
    { girdiFDE g=new girdiFDE("fDEq", " ");
      s=g.vericiktisi();
      try{ g.finalize();}
      catch(Throwable e1) {System.err.println("girdi error.");}
    }

    public static double[] doubleboyutgir(String s)
    {
        String s1=JOptionPane.showInputDialog(s);
        StringTokenizer token=new StringTokenizer(s1);
        int n=token.countTokens()-1;
        int m=n+1;
        double a[]=new double[m];
        int j=0;
        while(token.hasMoreTokens())
        {
            Double ax=new Double(token.nextToken());
            a[j++]=ax.doubleValue();
        }
        return a;
    }

    public static void main(String arg[])
    {
        de1 x0=new de1();
        double a[]=doubleboyutgir("x[0] y[0] y[n] dx");
        differential1 xx=new differential1(a[0],a[1],a[2],a[3]);
    }
}

```

differential1.java

```

import java.awt.event.*;
import java.awt.font.*;
import java.awt.geom.*;
import java.awt.image.*;
import javax.swing.*;
import java.util.Locale;
import java.text.*;
import java.util.*;
import javax.swing.table.*;

class differential1
{
    public differential1(double x0,double y0,double xn,double h)
    {
        fDEq ff=new fDEq();
        double r[][]= RK6(ff,x0,y0,xn,h);
        Text.print(Text.T(r));
    }

    public static double[] enterdoublearray(String s)
    {
        String s1=JOptionPane.showInputDialog(s);
        StringTokenizer token=new StringTokenizer(s1);
        int n=token.countTokens()-1;
        int m=n+1;
    }
}

```

```

double a[]=new double[m];
int j=0;
while(token.hasMoreTokens())
{
Double ax=new Double(token.nextToken());
a[j++]=ax.doubleValue();
}
return a;
}

public static double[][] RK6(f_xy f,double x0,double y0,double xn,double h)
{
// Runge-Kutta 6
int n=(int)((xn-x0)/h);
double a[][]=new double[2][n+1];
int i=0;
a[0][0]=x0;
a[1][0]=y0;
double k1,k2,k3,k4,k5,k6;
for(double xx=x0;xx<xn;xx+=h)
{ a[0][i+1]=xx+h;
k1=f.func(a[0][i],a[1][i]);
k2=f.func(a[0][i]+0.25*h,a[1][i]+0.25*k1*h);
k3=f.func(a[0][i]+0.25*h,a[1][i]+1.0/8.0*k1*h+1.0/8.0*k2*h);
k4=f.func(a[0][i]+0.5*h,a[1][i]-0.5*k2*h+k3*h);
k5=f.func(a[0][i]+3.0/4.0*h,a[1][i]+3.0/16.0*k1*h+9.0/16.0*k4*h);
k6=f.func(a[0][i]+h,a[1][i]-3.0/7.0*k1*h+2.0/7.0*k2*h+12.0/7.0*k3*h-12.0/7.0*k4*h+8.0/7.0*k5*h);
a[1][i+1]=a[1][i]+(7.0*k1+32.0*k3+12.0*k4+32.0*k5+7*k6)/90.0*h;
i++;
}
return a;
}

public static double[][] RKF6(f_xy f,double x0,double y0,double xn,double h,double eps)
{
// Runge-Kutta Fehlberg 6 deđişken stepli
//eps bađıl hata
double alfa;
//double a[][]=new double[2][n+1];
int i=0;
//a[0][0]=x0;
//a[1][0]=y0;
double delta=h;
double deltayeni=eps*y0;
double k1,k2,k3,k4,k5,k6;
Vector<Double> xi=new Vector<Double>();
Vector<Double> yi=new Vector<Double>();
double x,y,yp1_5,yp1_6,xeski,yeski;
x=x0;
y=y0;
xeski=x;
yeski=y;
Double Xi=new Double(x);
Double Yi=new Double(y);
xi.add(Xi);
yi.add(Yi);
while((x+h)<=xn)
{
xeski=x;

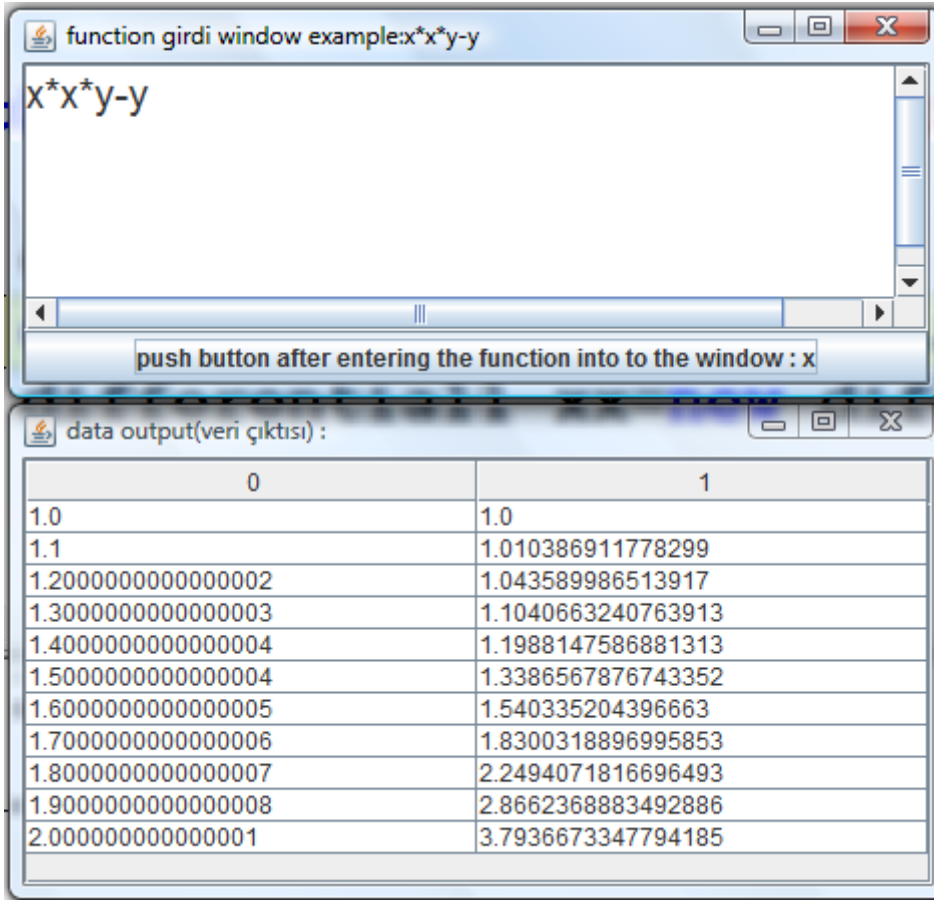
```

```

yeski=y;
k1=f.func(x,y);
k2=f.func(x+1.0/5.0*h,y+1.0/5.0*k1*h);
k3=f.func(x+3.0/10.0*h,y+3.0/40.0*k1*h+9.0/40.0*k2*h);
k4=f.func(x+3.0/5.0*h,y+3.0/10.0*k1*h-9.0/10.0*k2*h+6.0/5.0*k3*h);
k5=f.func(x+h,y-11.0/54.0*k1*h+5.0/2.0*k2*h-70.0/27.0*k3*h+35/27.0*k4*h);
k6=f.func(x+7.0/8.0*h,y+1631.0/55296.0*k1*h+175.0/512.0*k2*h+575.0/13824.0*k3*h+44275.0/110592.0*k4*h+253.0/4096.0*k5*h);
yp1_5=y+(37.0/378.0*k1+250.0/621.0*k3+125.0/594.0*k4+512.0/1771.0*k6)*h;
yp1_6=y+(2825.0/27648.0*k1+18575.0/48384.0*k3+13525.0/55296.0*k4+277.0/14336.0*k5+1.0/4.0*k6)*h;
y=yp1_6;
x=x+h;
Xi=new Double(x);
Yi=new Double(y);
xi.add(Xi);
yi.add(Yi);
delta=Math.abs(yp1_6-yp1_5);
deltayeni=Math.abs(eps*yp1_6);
if(delta<=deltayeni) alfa=0.2;
else alfa=0.25;
h=h*Math.pow((deltayeni/delta),alfa);
i++;
}
if(x<xn)//son bir step daha
{
h=xn-x;
k1=f.func(x,y);
k2=f.func(x+1.0/5.0*h,y+1.0/5.0*k1*h);
k3=f.func(x+3.0/10.0*h,y+3.0/40.0*k1*h+9.0/40.0*k2*h);
k4=f.func(x+3.0/5.0*h,y+3.0/10.0*k1*h-9.0/10.0*k2*h+6.0/5.0*k3*h);
k5=f.func(x+h,y-11.0/54.0*k1*h+5.0/2.0*k2*h-70.0/27.0*k3*h+35/27.0*k4*h);
k6=f.func(x+7.0/8.0*h,y+1631.0/55296.0*k1*h+175.0/512.0*k2*h+575.0/13824.0*k3*h+44275.0/110592.0*k4*h+253.0/4096.0*k5*h);
yp1_5=y+(37.0/378.0*k1+250.0/621.0*k3+125.0/594.0*k4+512.0/1771.0*k6)*h;
yp1_6=y+(2825.0/27648.0*k1+18575.0/48384.0*k3+13525.0/55296.0*k4+277.0/14336.0*k5+1.0/4.0*k6)*h;
y=yp1_6;
//System.out.println("x="+x+"h="+h+"y="+yp1_6);
x=xn;
Xi=new Double(x);
Yi=new Double(y);
xi.add(Xi);
yi.add(Yi);
i++;
}
//vektöre aktarılan değerleri double matrise aktar
int n=i;
double aa[][]=new double[2][n+1];
Enumeration nx=xi.elements();
Enumeration ny=yi.elements();
for(i=0;i<=n;i++)
{
//String ss1=(String)nx.nextElement();
//String ss2=(String)ny.nextElement();
Double ax=(Double)nx.nextElement();
Double ay=(Double)ny.nextElement();
aa[0][i]=ax.doubleValue();
aa[1][i]=ay.doubleValue();
}
return aa;
}

```

}



2) Altta kronometre sınıfı verilmiştir. Sınıfı bir jar dosyası haline getiriniz.

```
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import javax.swing.*;
import java.net.*;
import java.io.*;
import java.awt.Graphics;
import java.util.*;
import java.text.*;
import java.util.Locale;

class chronometerThread extends Thread
{
    public long i;
    JFrame a;
    public boolean done;
    public chronometerThread(JFrame ai)
    {a=ai;i=0;done=true;}

    public void reset() {i=0;}

    public void run()
    {
        while(done)
```

```

{
a.validate();
a.repaint();
try {Thread.sleep(10);i++;}
catch(InterruptedExceotion e)
{System.err.println(e.toString());}
}
}
}

public class chronometer extends JFrame implements Serializable,ActionListener
{
private static final long serialVersionUID=358044808532797990L;
    JTextField jtb;
    JButton b;
    Font F;
    chronometerThread r;
    int minute;
    int second;
    int centisecond;

    public chronometer()
    {
    super("chronometer");
    second=0;
    Container cc=getContentPane();
    jtb=new JTextField("");
    b=new JButton("start/stop");
    F=new Font("TimesRoman",Font.BOLD,24);
    jtb.setFont(F);
    b.setFont(F);
    r=new chronometerThread(this);
    cc.add(jtb,FlowLayout.NORTH);
    cc.add(b,FlowLayout.SOUTH);
    b.addActionListener(this);
    }

    public void actionPerformed( ActionEvent e)
    {
    if(r.i==0) r.start();
    else {r.done=false;r=new chronometerThread(this);}
    }

    public void paint(Graphics g)
    {
    minute=(int)(r.i/6000);
    second=(int)(r.i/100%60);
    centisecond=(int)(r.i%100);
    jtb.setText(""+minute+" "+second+" "+centisecond);
    }

    public static void main(String s[])
    {
    chronometer f = new chronometer();
    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {System.exit(0);}
    });
    f.setSize(200,120);
    f.setVisible(true);
    }
}

```



16. NETWORK PROGRAMLAMAYA GİRİŞ

16.1 TCP/IP PROTOKOLÜ

Java network protokolü olarak TCP/IP protokolünü kullanır. TCP İngilizce Transmission interface protokol (Veritransferi arayüz protokolü) IP ise Internet protokolü anlamına gelir.

IP kodu veriyi şu alt paketçikler olarak taşır :

- 0-10 : başlık
- 10-12 : kontrol
- 12-16 gönderici adresi
- 16-20 gideceği adres
- 20-24 : seçenekler
- 24....gönderilen bilgi

IP bilgi paketlerinin yerine ulaşma garantisizdir. TCP protokolü gitme garantili paketler oluşturur. IP paketini telgrafa benzetebiliriz. TCP paketleri ise telefon servisi gibidir. Karşılıklı iki tarafı bağlar. TCP paketinde alt birimler şunlardır :

- 0-2 gönderici adresi
- 2-4 gidecek yerin adresi
- 4-8 paket numarası
- 8-12 paket varış numarası
- 12-20 başlıklar
- 20+ veri

TCP kanallarından bazıları özel görevler görürler. Örneğin kanal 7 gönderilen bilgiyi geriye yansıtır. Kanal 13 verinin gittiği makinanın yerel zamanını bildirir. Kanal 20-21 ftp protokolü, kanal 23 telnet protokolü, kanal 80 http protokolü tarafından kullanılır. Java'da internet ulaşımı deyince ilk olarak bakmamız gereken URL sınıfıdır. URL sınıfı internet üzerinden direk olarak dosya açmamızı sağlayan bir sınıftır. URL'in kullanımı text okumada kullandığımız diğer kanal kullanımların aynı prensiple çalışır.

Kurucu metodlar	
	URL(String spec)
	URL(String protocol, String host, int port, String file)
	URL(String protocol, String host, int port, String file, URLStreamHandler handler)
	URL(String protocol, String host, String file)
	URL(URL context, String spec)
	URL(URL context, String spec, URLStreamHandler handler)

Diğer metotlardan bazıları	
boolean	equals(Object obj)
String	getAuthority()

Object	getContent()
Object	getContent(Class[] classes)
int	getDefaultPort()
String	getFile()
String	getHost()
String	getPath()
int	getPort()
String	getProtocol()
String	getQuery()
String	getRef()
String	getUserInfo()
int	hashCode()
URLConnection	openConnection()
URLConnection	openConnection(Proxy proxy)
InputStream	openStream() .
boolean	sameFile(URL other)
protected void	set(String protocol, String host, int port, String file, String ref)
protected void	set(String protocol, String host, int port, String authority, String userInfo, String path, String query, String ref)
static void	setURLStreamHandlerFactory(URLStreamHandlerFactory fac) Sets an application's URLStreamHandlerFactory.
String	toExternalForm()
String	toString()
URI	toURI()

PROGRAM 16.1.1 İnternet site seçici

```
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
```

```

import java.awt.geom.*;

public class sitesecici extends JApplet implements ListSelectionListener
{
    private static final long serialVersionUID = 1L;

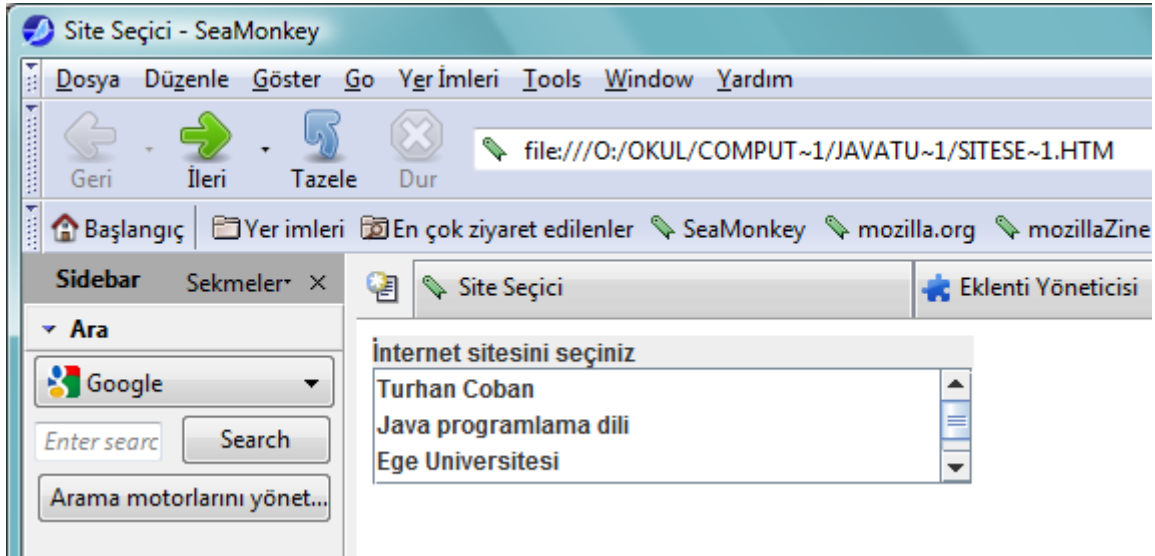
    private HashMap<Object, URL > sites; // site isimleri
    private ArrayList<String > siteNames; //
    private JList<String> siteChooser; // list of sites to choose from

    private String name[]={ "Turhan Coban",
        "Java programlama dili",
        "Ege Üniversitesi",
        "E.Ü. Mühendislik fakültesi makina mühendisliği"};
    private String adress[]={ "http://www.turhancoban.com/",
        "http://www.oracle.com/technetwork/java/index.html",
        "http://www.ege.edu.tr/",
        "http://me.ege.edu.tr/" };

    public void init()
    {
        // create GUI components and layout interface
        add( new JLabel( "İnternet sitesini seçiniz" ), BorderLayout.NORTH );
        siteChooser = new JList<String>(name);
        siteChooser.setVisibleRowCount(5);
        siteChooser.addListSelectionListener(this);
        add(new JScrollPane(siteChooser ), BorderLayout.CENTER );
    }

    public void valueChanged( ListSelectionEvent event )
    {
        // bu sistenin bağlantı adresini bulun
        String site=adress[siteChooser.getSelectedIndex()];
        try{
            URL newDocument = new URL(site);
            System.out.println("www="+newDocument.toString());
            AppletContext browser = getAppletContext();
            browser.showDocument(newDocument);
        } catch ( MalformedURLException urlException )
        { urlException.printStackTrace(); } // end catch
    } // end method valueChanged
} // end class sitesecici

```



İkinci uygulamada basit bir browser oluşturacağız, tabiki içinde ekleri ve diğer programları kullanan herhangi bir şey olmayacak

PROGRAM 16.1.1 İnternet site seçici

```
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.event.HyperlinkEvent;
import javax.swing.event.HyperlinkListener;

public class web extends JFrame implements ActionListener,HyperlinkListener
{
    private JTextField enterField; // JTextField to enter site name
    private JEditorPane contentsArea; // to display website
    private static final long serialVersionUID = 1L;
    // set up GUI
    public web()
    {
        super( "Basit bir browser program" );

        // create enterField and register its listener
        enterField = new JTextField( "Enter file URL here" );
        enterField.addActionListener(this);
        add( enterField, BorderLayout.NORTH );

        contentsArea = new JEditorPane(); // create contentsArea
        contentsArea.setEditable( false );
        contentsArea.addHyperlinkListener(this);
        add( new JScrollPane( contentsArea ), BorderLayout.CENTER );
        setSize( 400, 300 ); // set size of window
        setVisible( true ); // show window
    } // end web constructor

    public void actionPerformed( ActionEvent event )
```

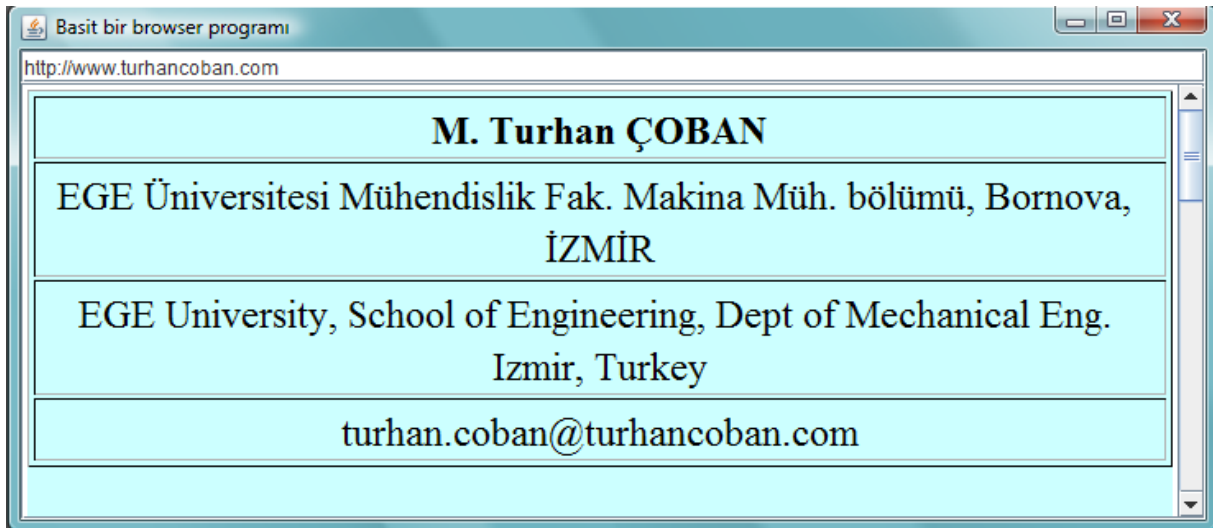
```

    {
        getThePage( event.getActionCommand() );
    } // end method actionPerformed

public void hyperlinkUpdate( HyperlinkEvent event )
    {
        if ( event.getEventType() ==
            HyperlinkEvent.EventType.ACTIVATED )
            getThePage( event.getURL().toString() );
    } // end method hyperlinkUpdate

// load document
private void getThePage( String location )
    {
        try // load document and display location
        {
            contentsArea.setPage( location ); // set the page
            enterField.setText( location ); // set the text
        } // end try
        catch ( IOException ioException )
        {
            JOptionPane.showMessageDialog( this,
                "Error retrieving specified URL", "Bad URL",
                JOptionPane.ERROR_MESSAGE );
        } // end catch
    } // end method getThePage
public static void main( String args[] )
    {
        web application = new web();
        application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    } // end main
} // end class web

```



16.2 TCP/IP PROTOKOLÜ

Bu bölümde **ServerSocket** ve **ClientSocket** isimli java sınıflarını kullanarak internet üzerinden haberleşmeyi ne şekilde oluşturabileceğimize bakacağız. Temel olarak bir istek gönderme ve bu isteğe cevap verme şeklinde çalışan bir sistem söz konusudur. **ServerSocket**'i bağlamak için

```

ServerSocket server = new ServerSocket( portNumber, queueLength );
ServerSocket

```

Program 20.2-1 Server.java

```
/ Set up a server that will receive a connection from a client, send
// a string to the client, and close the connection.
import java.io.EOFException;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;

public class Server extends JFrame
{
    private static final long serialVersionUID = 2345L;
    private JTextField enterField; // inputs message from user
    private JTextArea displayArea; // display information to user
    private ObjectOutputStream output; // output stream to client
    private ObjectInputStream input; // input stream from client
    private ServerSocket server; // server socket
    private Socket connection; // connection to client
    private int counter = 1; // counter of number of connections

    // set up GUI
    public Server()
    {
        super( "Server" );

        enterField = new JTextField(); // create enterField
        enterField.setEditable( false );
        enterField.addActionListener(
            new ActionListener()
            {
                // send message to client
                public void actionPerformed( ActionEvent event )
                {
                    sendData( event.getActionCommand() );
                    enterField.setText( "" );
                } // end method actionPerformed
            } // end anonymous inner class
        ); // end call to addActionListener

        add( enterField, BorderLayout.NORTH );

        displayArea = new JTextArea(); // create displayArea
        add( new JScrollPane( displayArea ), BorderLayout.CENTER );

        setSize( 300, 150 ); // set size of window
        setVisible( true ); // show window
    } // end Server constructor

    // set up and run server
```

```

public void runServer()
{
try // set up server to receive connections; process connections
{
server = new ServerSocket( 12345, 100 ); // create ServerSocket

while ( true )
{
try
{
waitForConnection(); // wait for a connection
getStreams(); // get input & output streams
processConnection(); // process connection
} // end try
catch ( EOFException eofException )
{
displayMessage( "\nServer terminated connection" );
} // end catch
finally
{
closeConnection(); // close connection
counter++;
} // end finally
} // end while
} // end try
catch ( IOException ioException )
{
ioException.printStackTrace();
} // end catch
} // end method runServer

// wait for connection to arrive, then display connection info
private void waitForConnection() throws IOException
{
displayMessage( "Bağlanmak için bekliyor\n" );
connection = server.accept(); // allow server to accept connection
displayMessage( connection.getInetAddress().getHostName() + " kanalından "+counter+ "bağlantı gerçekleşti");
}

// get streams to send and receive data
private void getStreams() throws IOException
{
// set up output stream for objects
output = new ObjectOutputStream( connection.getOutputStream() );
output.flush(); // flush output buffer to send header information

// set up input stream for objects
input = new ObjectInputStream( connection.getInputStream() );

displayMessage( "\n I/O mesajı alındı\n" );
} // end method getStreams

// process connection with client
private void processConnection() throws IOException
{
String message = "Connection successful";
sendData( message ); // send connection successful message

```

```

// enable enterField so server user can send messages
setTextFieldEditable( true );

do // process messages sent from client
{
try // read message and display it
{
message = ( String ) input.readObject(); //read new message
displayMessage( "\n" + message ); // displaymessage
} // end try
catch ( ClassNotFoundException classNotFoundException )
{
displayMessage( "\nbilinmeyen tip nesne" );
} // end catch

} while ( !message.equals( "CLIENT>>> SONLANDIRILDI" ));
} // end method processConnection

// close streams and socket
private void closeConnection()
{
displayMessage( "\nbağlantı sonlandırılıyor\n" );
setTextFieldEditable( false ); // disable enterField

try
{
output.close(); // close output stream
input.close(); // close input stream
connection.close(); // close socket
} // end try
catch ( IOException ioException )
{
ioException.printStackTrace();
} // end catch
} // end method closeConnection

// send message to client
private void sendData( String message )
{
try // send object to client
{
output.writeObject( "SERVER>>> " + message );
output.flush(); // flush output to client
displayMessage( "\nSERVER>>> " + message );
} // end try
catch ( IOException ioException )
{
displayArea.append( "\nYazım hatası");
} // end catch
} // end method sendData

// manipulates displayArea in the event-dispatch thread
private void displayMessage( final String messageToDisplay )
{
SwingUtilities.invokeLater(
new Runnable()
{
public void run() // updates displayArea

```



```

{
displayArea.append( messageToDisplay );// append message
} // end method run
} // end anonymous inner class
); // end call to SwingUtilities.invokeLater
} // end method displayMessage

// manipulates enterField in the event-dispatch thread
private void setTextFieldEditable( final boolean editable )
{
SwingUtilities.invokeLater(
new Runnable()
{
public void run() // sets enterField's editability
{
enterField.setEditable( editable );
} // end method run
} // end inner class
); // end call to SwingUtilities.invokeLater
} // end method setTextFieldEditable

public static void main( String args[] )
{
Server application = new Server(); // create server
application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
application.runServer(); // run server application
} // end main
} // end class Server

```

Program 20.2-2 Client.java

```

// Client.java
// Client that reads and displays information sent from a Server.
import java.io.EOFException;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.InetAddress;
import java.net.Socket;
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;

public class Client extends JFrame
{
private static final long serialVersionUID = 2346L;
private JTextField enterField; // enters information from user
private JTextArea displayArea; // display information to user
private ObjectOutputStream output; // output stream to server
private ObjectInputStream input; // input stream from server
private String message = ""; // message from server
private String chatServer; // host server for this application
private Socket client; // socket to communicate with server

// initialize chatServer and set up GUI
public Client( String host )

```

```

{
    super( "Client" );

    chatServer = host; // set server to which this client connects

    enterField = new JTextField(); // create enterField
    enterField.setEditable( false );
    enterField.addActionListener(
        new ActionListener()
        {
            // send message to server
            public void actionPerformed( ActionEvent event )
            {
                sendData( event.getActionCommand() );
                enterField.setText( "" );
            } // end method actionPerformed
        } // end anonymous inner class
    ); // end call to addActionListener

    add( enterField, BorderLayout.NORTH );

    displayArea = new JTextArea(); // create displayArea
    add( new JScrollPane( displayArea ), BorderLayout.CENTER );

    setSize( 300, 150 ); // set size of window
    setVisible( true ); // show window
} // end Client constructor

// connect to server and process messages from server
public void runClient()
{
    try // connect to server, get streams, process connection
    {
        connectToServer(); // create a Socket to make connection
        getStreams(); // get the input and output streams
        processConnection(); // process connection
    } // end try
    catch ( EOFException eofException )
    {
        displayMessage( "\nClient bağlantısı sonlandırdı" );
    } // end catch
    catch ( IOException ioException )
    {
        ioException.printStackTrace();
    } // end catch
    finally
    {
        closeConnection(); // close connection
    } // end finally
} // end method runClient

// connect to server
private void connectToServer() throws IOException
{
    displayMessage( "Attempting connection\n" );

    // create Socket to make connection to server
    client = new Socket( InetAddress.getByName( chatServer ), 12345 );

```

```

// display connection information
displayMessage( "Bağlanılan yer : " +
    client.getInetAddress().getHostName() );
} // end method connectToServer

// get streams to send and receive data
private void getStreams() throws IOException
{
    // set up output stream for objects
    output = new ObjectOutputStream( client.getOutputStream() );
    output.flush(); // flush output buffer to send header information

    // set up input stream for objects
    input = new ObjectInputStream( client.getInputStream() );

    displayMessage( "\nGot I/O streams\n" );
} // end method getStreams

// process connection with server
private void processConnection() throws IOException
{
    // enable enterField so client user can send messages
    setTextFieldEditable( true );

    do // process messages sent from server
    {
        try // read message and display it
        {
            message = ( String ) input.readObject(); // read new message
            displayMessage( "\n" + message ); // display message
        } // end try
        catch ( ClassNotFoundException classNotFoundException )
        {
            displayMessage( "\nbilinmeyen tip nesne gönderildi" );
        } // end catch

    } while ( !message.equals( "SERVER>>> TERMINATE" ) );
} // end method processConnection

// close streams and socket
private void closeConnection()
{
    displayMessage( "\nBağlantı kapatılıyor" );
    setTextFieldEditable( false ); // disable enterField

    try
    {
        output.close(); // close output stream
        input.close(); // close input stream
        client.close(); // close socket
    } // end try
    catch ( IOException ioException )
    {
        ioException.printStackTrace();
    } // end catch
} // end method closeConnection

```

```

// send message to server
private void sendData( String message )
{
    try // send object to server
    {
        output.writeObject( "CLIENT>>> " + message );
        output.flush(); // flush data to output
        displayMessage( "\nCLIENT>>> " + message );
    } // end try
    catch ( IOException ioException )
    {
        displayArea.append( "\nYazım hatası" );
    } // end catch
} // end method sendData

// manipulates displayArea in the event-dispatch thread
private void displayMessage( final String messageToDisplay )
{
    SwingUtilities.invokeLater(
        new Runnable()
        {
            public void run() // updates displayArea
            {
                displayArea.append( messageToDisplay );
            } // end method run
        } // end anonymous inner class
    ); // end call to SwingUtilities.invokeLater
} // end method displayMessage

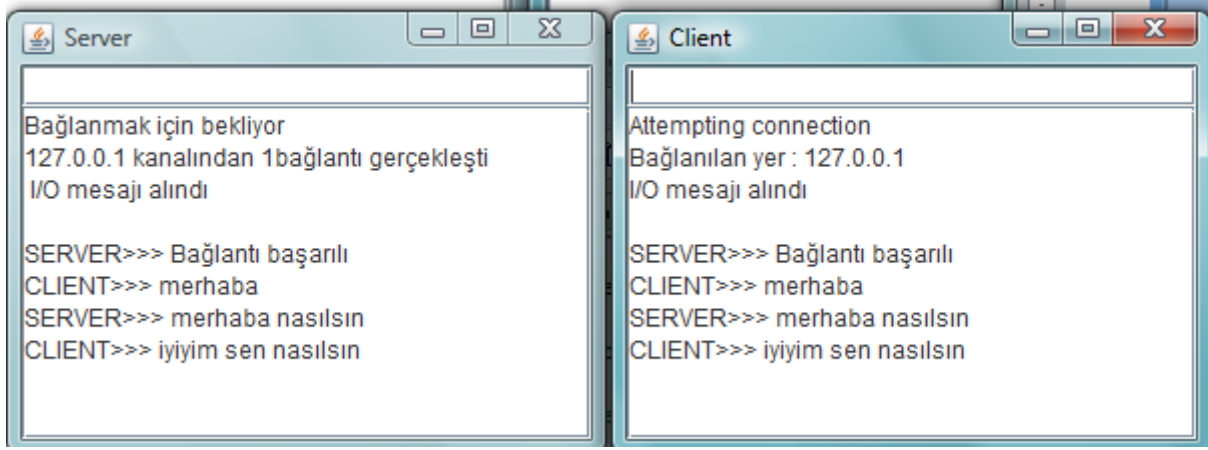
// manipulates enterField in the event-dispatch thread
private void setTextFieldEditable( final boolean editable )
{
    SwingUtilities.invokeLater(
        new Runnable()
        {
            public void run() // sets enterField's editability
            {
                enterField.setEditable( editable );
            } // end method run
        } // end anonymous inner class
    ); // end call to SwingUtilities.invokeLater
} // end method setTextFieldEditable

public static void main( String args[] )
{
    Client application; // declare client application

    // if no command line args
    if ( args.length == 0 )
        application = new Client( "127.0.0.1" ); // connect to localhost
    else
        application = new Client( args[ 0 ] ); // use args to connect

    application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    application.runClient(); // run client application
} // end main
} // end class Client

```



BÖLÜM 17 JAVA ANADİL (NATIVE LANGUAGE) PROGRAMLAMASINA GİRİŞ

C++ ve C PROGRAMLAMA DİLLERİNİN JAVA İLE BİRLİKTE KULLANIMI

17.1 TEMEL KAVRAMLAR

Java programlama dili C dilinde yazılmıştır. Bu yüzden C'den javaya geçiş oldukça kolay bir şekilde yapılabilir. Bir örnekle birden fazla programlama dilinin nasıl kullanılabildiğine bakmadan önce, bir temel kavramı iyi anlamamız gerekir. Java güvenlik açısından maksimum güvenliği verebilen bir dildir. Öte taraftan C maksimum hızlara ulaşmayı sağlayabilen bir dildir. C programlarını java programlarıyla birlikte kullanmamızın temel nedeni hız gereken yerlerde bu hızı sağlamaktır. Ancak hız sağlarken güvenlikten olan kaybımızı her zaman hesaba katmamız ve ikisi arasında bir denge bulmamız biz programcılara bırakılmıştır. Bu yüzden iki dilli programlamada oldukça dikkatli bir şekilde çalışma gereksinimi oluşturur. Bu konu C++ dilini iyi bilen programcılar içindir. Burada C programlamasına girmeyeceğiz. Eğer bu dili bilmiyorsanız, bu konuya başlamadan önce öğrenmenizi veya java anadil programlamasını hiç kullanmamanızı tavsiye ederim. C++ dilini kullanabilmek için, hele iki dili bir arada kullanabilmek için iyi bir programcı olmanız gerekiyor, eğer kendinizi güveniyorsanız, bundan sonraki kısma geçebilirsiniz.

17.2 ÖRNEK PROBLEMLE STEP STEP İKİ DİLLİ PROGRAM KODUNUN HAZIRLANMASI VE ÇALIŞTIRILMASI

Çok dilli programlama birden fazla basamağı gerektiren bir prosestir. Bir örnek problemler bu basamakları inceleyelim :C++ dili ile java dilinin beraber çalışacağı bir örnek oluşturalım. İlk step olarak java dilindeki programımızı yazalım :

Program 17.1 JavaC++.java programı

```
class JavaC++
{
    public native void cdenaktar();
    static
    {
        System.loadLibrary("jc++");
    }

    public static void main(String[] args)
    {
        new JavaC++().cdenaktar();
    }
}
```

JavaC++ programında iki temel metod görüyoruz. Birinci metod cdenaktar() metodudur. Bu metod biraz sonra C++ dilinde oluşturacağımız fonksiyonu Java'nın bir parçası olarak çağırır. İkinci metod ise main metodudur. Cdenaktar metoduna bakarken bunun java metodu olmadığını belirten native sözcüğü yer almaktadır. Metodun türü void'dir, yani hiçbir şey aktarılmıyacaktır. Programımızı, herhangi bir java programında olduğu gibi önce

java JavaC++.java

Komutunu kullanarak derleriz. İkinci stepte javah komutunu kullanarak

javah JavaC++

JavaC++.h dosyasını oluştururuz. Bu dosyanın içeriği :

Program 17.2 JavaC++.h arabağlantı programı

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class JavaC++ */

#ifdef _Included_JavaC++
```

```

#define _Included_JavaCpp
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class:   JavaCpp
 * Method:  cdenaktar
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_JavaCpp_cdenaktar
    (JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif

```

şeklinde olacaktır. Bundan sonraki stepte C++ dilinde aşağıdaki programı oluşturalım :

Program 17.3 JavaCpp_c.cpp C++ programı

```

#include <iostream.h>
#include <jni.h>
#include "JavaCpp.h"

JNIEXPORT void JNICALL
Java_JavaCpp_cdenaktar(JNIEnv *env, jobject obj)
{
    cout<<"Java-C++ programlama dillerini birlikte kullanıyoruz ";
    return;
}

```

C++ programını derlemek için elbette bir derleyiciye ihtiyaç vardır. Biz burada MS Visual C++ versiyon 6 derleyicisini kullandık, herhangi bir dll dosyası hazırlayan derleyici (MS Windows için) aynı işi görebilir. Bu derleyici ile derleme için ya direk olarak dosya

```
cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %JavaCpp_c.cpp -Fejcpp.dll
```

komutunu kullanırız, yada bu uzun komutu her sefer yazmamak için bir isim.bat dosyası yaratıp bunu kullanabiliriz. Örnekte benim java dosyalarım c:\co\java Örneğin hazırladığımız pcomp.bat dosyası

Program 17.4 pcomp.bat MSdos programı

```
cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LD %1.cpp -Fe%2.dll
```

kullanılarak yukarıdaki komut
pcomp JavaCpp_c jcpp

şeklinde verilebilir. Bu işlemleri yaptığımızda programımız çalışmaya hazır olacaktır. Programı çalıştırmak için :

java JavaCpp

komutunu kullanmamız yeterli olacaktır. Program çıktısı :

Java-C++ programlama dillerini birlikte kullanıyoruz

Şeklinde olacaktır. Burada derlemenin jcpp.dll dosyasına yapıldığını hatırlatalım, ve bu dosya java programı üzerinden C++ metodunu çalıştırırken kullandığımız ana programdı.

17.2 DEĞİŞKENLERİN VE JAVA NESLERİNİN AKTARILMASI

İlk programımızda java ve C++ arasında herhangi bir değişken aktarılması olmamış, sadece void türü fonksiyon kullanılarak belli bir yazı yazan C++ fonksiyonu çağırılmıştır. İkinci örneğimizde java programımızdan String türü değişkeni C++ programlama diline aktaracak ve kullanacağız.

İlk java programımız Showmessage.java. bu programda ekrandan girilen bir string, yahut ekrandan girilen bir string değeri yoksa programın içinde tanımlanan bir stringle birlikte C++ programına aktarılıp burada ekrana yazılmaktadır.

Program 17.5 ShowMessage.java programı

```
public class ShowMessage {
private native void ShowMessage(String msg);
static { System.loadLibrary("MsgI"); }

public static void main(String[] args) {
ShowMessage app = new ShowMessage();
if(args.length==0)
app.ShowMessage("Bu mesaj javadan C++ ya gönderildi ve orada yazıldı");
else
app.ShowMessage(args[0]);
}
}
```

bu program **javac ShowMessage.java** ve **javah ShowMessage** komutlarıyla işlendikten sonra

Program 17.6 MsgImpl.cpp programı

```
//dosya : MsgImpl.cpp
#include <jni.h>
#include <iostream.h>
#include <stdio.h>
#include "ShowMessage.h"
extern "C" JNIEXPORT void JNICALL
Java_ShowMessage_ShowMessage(JNIEnv* env,
 jobject, jstring jMsg)
{
const char* msg=env->GetStringUTFChars(jMsg,0);
printf("JNI cikti stringi : %s\n",msg) ;
env->ReleaseStringUTFChars(jMsg, msg);
}
```

c++ dilindeki MsgImpl.cpp programı hazırlanmış ve üstte hazırladığımız pcomp.bat yardımıyla

pcomp MsgImpl MsgI

komutu kullanılarak derlenmiştir. Program çıktısı

java ShowMessage “Merhaba Dünya” komutu kullanılarak :

JNI cikti stringi : Merhaba Dünya

veya

java ShowMessage komutu kullanılarak

JNI cikti stringi : Bu mesaj javadan C++ ya gönderildi ve orada yazdırıldı

Çıktısı alınabilir. Programda veri aktarımı yapmak amacıyla java tarafında metod tanımı yapılırken

private native void ShowMessage(String msg);

tanımı verilmiştir. Buradaki String msg metodun dolayısıyla C++ fonksiyonunun girdisini teşkil etmektedir. C++ tarafında mesajı alırken,

```
Java_ShowMessage_ShowMessage(JNIEnv* env,
 jobject, jstring jMsg)
{
const char* msg=env->GetStringUTFChars(jMsg,0);
```


jstring jMsg tanımıyla aktarılmış ve const char* msg=env->GetStringUTFChars(jMsg,0); tanımıyla c değişkenine dönüştürülmüştür.

Javada Stringler unicode (16 bit) olarak tanımlanmıştır. C stringleri ise 8 bitlik ascii char boyutlu değişkeni olarak tanımlanır. Bu yüzden string değişkenlerini javadan cye veya cden javaya aktarırken dönüşümler yapmak gerekir. Aynı zamanda java string değişkenleri length (boyut) değişkenini de bünyelerinde taşırlar. Cde ise boyut tanımlanmamıştır. Aynı bir değişken üzerinden tanımlanması gerekebilir. Bu dönüşümleri yapmak için bir dizi metod tanımlanmıştır.

Bu tanımların tamamı şöyledir :

```
GetStringChars
GetStringLength
GetStringUTFChars
GetStringUTFLength
NewString
NewStringUTF
ReleaseStringChars
ReleaseStringUTFChars
```

Buradaki ikinci örneğimizde java programında c++ dilindeki bir fonksiyonun (metod) yaptığı hesap sonuçlarını aktaracağız. Programımız java'da çağırılan bir C++ kare metodunu kullanarak java dilinde girilen sayının karesini hesaplayacaktır.

Program 17.7 karesinifi.java programı

```
class karesinifi {
    private native double kare(double x);
    public static void main(String args[]) {
        karesinifi p = new karesinifi();
        double x=2.0;
        double y = p.kare(x);
        System.out.println(" "+x+" kare = "+ y);
    }
    static {
        System.loadLibrary("kareI");
    }
}
```

bu programı javac karesinifi.java deyiimiyle derleyip, javah karesinifi deyiimiyle anadil kütüphanesine çevirirsek

Program 14.8 karesinifi.h programı

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class karesinifi */

#ifdef _Included_karesinifi
#define _Included_karesinifi
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class: karesinifi
 * Method: kare
 * Signature: (D)D
 */
JNIEXPORT jdouble JNICALL Java_karesinifi_kare
    (JNIEnv *, jobject, jdouble);

#ifdef __cplusplus
}
#endif
#endif
```

```
#endif
```

karesinifi.h bize java programındaki
private native double kare(double x);
metodunun c++ anadil (Native) karşılığının
JNIEXPORT jdouble JNICALL Java_karesinifi_kare
(JNIEnv *, jobject, jdouble);
olduğunu göstermektedir. Şimdi kare metodunun aslını C++ dilinde yazarsak :

Program 14.9 karesinifi.cpp C++ programı

```
#include <stdio.h>
#include <jni.h>
#include "karesinifi.h"

JNIEXPORT jdouble JNICALL Java_karesinifi_kare(JNIEnv *, jobject, jdouble x )
{
    return x*x;
}
```

burada metod isminin kare yerine **Java_karesinifi_kare** şeklini aldığını görüyoruz. Double değişkeni de yerini jdouble değişkenine bırakmış durumda. Java anadilinde programları yazarken metod isimleri Java_sınıfismi_metodismi şeklini alır. Basit java değişkenlerinin Anadil karşılıkları ise

Tablo 14.1 Java değişken tipleri ve ana dil (native) eşdeğerleri

Java değişken tipi	Anadil (Native) değişken eşdeğeri	Değişken bit değeri
Boolean	jboolean	8, işaretli
Byte	jbyte	8
Char	jchar	16, işaretli
Short	jshort	16
Int	jint	32
Long	jlong	64
Float	jfloat	32
Double	jdouble	64
Object	jobject	
Void	void	Yok

Şeklinde. C++ programı
pcomp karesinifi kareI
deyimiyle veya açık yazılışıyla
cl -Ic:\co\java\include -Ic:\co\java\include\win32 -LDkaresinifi.cpp -KareI.dll
deyimiyle derlenirse (MS Visual C++ 6), sonuç :
2.0 kare = 4.0
şeklinde çıkacaktır.

Şimdi de bu işlemin tam tersini inceleyelim. Java'da yazılmış olan kare metodunu C++ ve C dillerinde çağıralım.

Java programımız karesinifi1.java aşağıdaki gibi tanımlanmıştır.

Program 17.10 karesinifi1.java java programı

```
class karesinifi1 {
    private native void nativeMethod();
    private double kare(double x)
    {
```

```

return x*x;
}
public static void main(String args[])
{
karesinifi1 c=new karesinifi1();
c.nativeMethod();
}
static {
System.loadLibrary("kare11");
}
}

```

sınıfın alt sınıfı olan kare’de girilen sayının karesi hesaplanmaktadır. NativeMethod metodu da C (veya c++) dilleriyle bağlantı sağlamaktadır. Metodumuzu yine java karesinifi1.java ile derledikten sonra javah karesinifi1 komutuyla karesinifi1.h dosyasını oluştururuz. Bu dosya :

Program 14.11 karesinifi1.h programı

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class karesinifi1 */

#ifdef _Included_karesinifi1
#define _Included_karesinifi1
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class: karesinifi1
 * Method: nativeMethod
 * Signature: ()V
 */
JNIEXPORT void JNICALL Java_karesinifi1_nativeMethod
(JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif

```

anadil tarafında program yazılırken bize yardımcı olabilecek bir ara java programı javap de mevcuttur. Bu işlemi yapmak zorunda değiliz. Yapma sebebimizi c++ dili programını inceleyince daha iyi anlayacaksınız.

javap -s -p karesinifi1

Program 14.12 javap komutuyla oluşturulan ara kod

```

Compiled from karesinifi1.java
class karesinifi1 extends java.lang.Object {
    karesinifi1();
        /* ()V */
    private native void nativeMethod();
        /* ()V */
    private double kare(double);
        /* (D)D */
    public static void main(java.lang.String[]);
        /* ([Ljava/lang/String;)V */
    static {};
        /* ()V */
}

```

şimdi c++ koduna bakalım :

Program 14.13 karesinifi1.cpp programı

```
#include <stdio.h>
#include <iostream.h>
#include <jni.h>
#include "karesinifi1.h"

JNIEXPORT void JNICALL Java_karesinifi1_nativeMethod(JNIEnv *env, jobject obj)
{
    jclass cls=env->GetObjectClass(obj);
    jmethodID mid=env->GetMethodID(cls,"kare","(D)D");
    double x=2.0;
    double y=env->CallDoubleMethod(obj,mid,x);
    cout<<x<<"nin karesi = "<<y;
}
```

Bu programda önce java sınıfının (karesinifi1) aktarılması **jobject obj** değişkeni ve `jclass cls=env->GetObjectClass(obj);` üzerinden yapılır. Burada sınıf adresi `cls` değişkenine yüklenmiş olur. Alt metod karenin tanımı için `jmethodID mid=env->GetMethodID(cls,"kare","(D)D");` tanımı kullanılmıştır. Buradaki “(D)D” tanımı program 14.12 de bize verilmişti. Anlamı java programındaki kare metodunun double değişken girişi olduğu ve çıkışında double değişken istediğidir. “kare” deyimiyle metodun javadaki ismi verilmiştir. `cls` ise üstte tanımladığımız karesinifi1 java sınıfının adresine belirtmektedir.

`double y=env->CallDoubleMethod(obj,mid,x);` deyimiyle de metodu çağırıp sonuçlarını `y` değişkenine yüklemekteyiz.

Burada kullandığımız D işareti java değişken türünün double olduğunu belirtiyordu. Java değişken tiplerinin işaret karşılıkları Tablo 14.2 de verilmiştir.

Tablo 17.2 Java tiplerinin eşdeğer işaretleri

İşaret	Java değişken tipi
Z	boolean
B	Byte
C	Char
S	Short
I	Int
J	Long
F	Float
D	Double
L sınıf ismi	sınıf ismi
[Değişken türü	Değişken türü[]

Bu programda sınıfı çağırarak için `GetObjectClass` metodu, metodu tanımlamak içinde `GetMethodID` metodunu kullandık. `CallDoubleMethod` metodu ile de metodu çalıştırdık. Paralel işlemleri yapmak için mevcut olan metod listesi ve anlamları şöyledir :

- GetObjectClass : sınıf adresini tanımlar
- GetMethodID : dinamik metod adresini tanımlar
- GetStaticMethodID : statik metod adresini tanımlar
- Call<değişkentürü>Method : çeşitli metodları çağırır
 - CallBooleanMethod
 - CallByteMethod
 - CallCharMethod
 - CallDoubleMethod
 - CallFloatMethod

CallIntMethod
CallLongMethod
CallObjectMethod
CallShortMethod
CallVoidMethod

CallStatic< deęişkentürü >Method : statik metodları çağırır, deęişken türleri olarak yukarıdakinin aynı türleri kabul eder.

CallNonvirtual< deęişkentürü >Method
Call< deęişkentürü >MethodV
Call< deęişkentürü >MethodA

Aynı programın c dilinde yazılması küçük farklılıklar taşır. C dilindeki program :

Program 17.14 karesinif1.c programı

```
#include <stdio.h>
#include <jni.h>
#include "karesinif1.h"

JNIEXPORT void JNICALL Java_karesinif1_nativeMethod(JNIEnv *env, jobject obj)
{
    jclass cls=(*env)->GetObjectClass(env,obj);
    jmethodID mid=(*env)->GetMethodID(env,cls,"kare","(D)D");
    double x=2.0;
    double y=(*env)->CallDoubleMethod(env,obj,mid,x);
    printf("%f nin karesi = %f ",x,y);
}
```

programdan da görüldüğü gibi C++ daki :

```
jclass cls=env->GetObjectClass(obj);
komutu
jclass cls=(*env)->GetObjectClass(env,obj);
halini almıştır.
jmethodID mid=env->GetMethodID(cls,"kare","(D)D");
komutu
jmethodID mid=(*env)->GetMethodID(env,cls,"kare","(D)D");
halini almıştır.
double y=env->CallDoubleMethod(obj,mid,x);
komutu ise
double y=(*env)->CallDoubleMethod(env,obj,mid,x);
halini almıştır. Sonuç olarak env yerine referans deęişkeni (*env) gelmektedir ve env referans deęişkeni aynı zamanda metod deęişkeni olarakda çağırılmaktadır. Bunun dışında işlemler arasında bir fark yoktur.
Program java karesinif1 deyimiyle çalıştırıldığında
2nin karesi = 4
sonucunu alırız.
```

İkinci bir sınıf ulaşım örneęi olarak UseObjects programını verelim. Burada sınıf, MyJavaClass içinde bir int deęişken ve bir void metod mevcuttur. İkinci bir metod olan UseObjects metodu kullanılarak native metod changeObject'e ulaşılmakta ve bu metod c++ programında kullanılarak sonuçlar hesaplanmaktadır.

Program 17.15 UseObjects.java programı

```
//UseObjects.java
class MyJavaClass {
public int aValue;
public void divByTwo() { aValue /= 2; }
}
public class UseObjects {
private native void
changeObject(MyJavaClass obj);
static {
System.loadLibrary("UseObjImpl");
```

```

}
public static void main(String[] args) {
UseObjects app = new UseObjects();
MyJavaClass anObj = new MyJavaClass();
anObj.aValue = 2;
app.changeObject(anObj);
System.out.println("Java: " + anObj.aValue);
}
}

```

Program 17.16 javah UseObject komutu kullanılarak oluşturulan UseObjects.h programı

```

/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class UseObjects */

#ifdef _Included_UseObjects
#define _Included_UseObjects
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class: UseObjects
 * Method: changeObject
 * Signature: (LMyJavaClass;)V
 */
JNIEXPORT void JNICALL Java_UseObjects_changeObject
(JNIEnv *, jobject, jobject);

#ifdef __cplusplus
}
#endif
#endif

```

Program 17.17 UseObjImp.cpp programı

```

#include <jni.h>
extern "C" JNICALL void JNICALL
Java_UseObjects_changeObject(JNIEnv* env, jobject, jobject obj)
{
jclass cls = env->GetObjectClass(obj);
jfieldID fid = env->GetFieldID(cls, "aValue", "I");
jmethodID mid = env->GetMethodID(cls, "divByTwo", "()V");
int value = env->GetIntField(obj, fid);
printf("Native: %d\n", value);
env->SetIntField(obj, fid, 6);
env->CallVoidMethod(obj, mid);
value = env->GetIntField(obj, fid);
printf("Native: %d\n", value);
}

```

Program 17.15 de tanımlanan MyJavaClass sınıfı bir int değişken, aValue, ve bir metod, divByTwo, içermektedir. Metod int de verilen değeri 2 ye bölmekte ve sonucu aktarmaktadır. Bu programda int sınıf alt değişkenini tanımlamak için

```

jfieldID fid = env->GetFieldID(cls, "aValue", "I");
deyimini kullandık. Void aValue metodu
jmethodID mid = env->GetMethodID(cls, "divByTwo", "()V");
deyimiyle tanımlandı. Ve
env->SetIntField(obj, fid, 6);

```

tanımıyla 6 değeri atandı.

Sınıfların alt değişkenlerine değer atamak (ve okumak)için

```
GetFieldID
GetStaticFieldID
Get< deęişkentürü >Field
  GetBooleanField
  GetByteField
  GetCharField
  GetDoubleField
  GetFloatField
  GetIntField
  GetLongField
  GetObjectField
  GetShortField
Set< deęişkentürü >Field
GetStatic< deęişkentürü >Field
SetStatic< deęişkentürü >Field
```

Metodları mevcuttur.

Şimdi de Çok boyutlu deęişkenleri nasıl kullanabildięimizi inceleyelim :

Program 17.18 IntArray.java programı

```
class IntArray {
  private native int sumArray(int arr[]);
  public static void main(String args[]) {
    IntArray p = new IntArray();
    int arr[] = new int [10];
    for (int i = 0; i < 10; i++)
      arr[i] = i;
    int sum = p.sumArray(arr);
    System.out.println("sum = " + sum);
  }
  static {
    System.loadLibrary("MyImpOfIntArray");
  }
}
```

Program 14.19 IntArray.c programı

```
#include <jni.h>
#include "IntArray.h"

JNIEXPORT jint JNICALL
Java_IntArray_sumArray(JNIEnv *env, jobject obj, jintArray arr)
{
  jsize len = (*env)->GetArrayLength(env, arr);
  int i, sum = 0;
  jint *body = (*env)->GetIntArrayElements(env, arr, 0);
  for (i=0; i<len; i++) {
    sum += body[i];
  }
  (*env)->ReleaseIntArrayElements(env, arr, body, 0);
  return sum;
}
```

burada tanımlanan java programında boyutlu deęişken arr tanımlanmış, ve boyutlu deęişkenin toplamı c programında hesaplanmıştır. C metodunda boyutlu deęişkeni tanımlamak için jintArray türü arr deęişkeni kullanılmıştır. Boyutlu deęişkenin boyutu jsize tipi len deęişkenine
jsize len = (*env)->GetArrayLength(env, arr);

deyimiyle aktarılmıştır.

arr değişkeninin değeri jint tipi body boyutlu değişkene, indeksi 0dan başlamak üzere

```
jint *body = (*env)->GetIntArrayElements(env, arr, 0);
```

metoduyla aktarılır.

arr boyutlu değişkeninin değeri (değerler değişmiş olabilir) tekrar kullanılabilir üzere

```
(*env)->ReleaseIntArrayElements(env, arr, body, 0);
```

deyimi kullanılarak serbest bırakılır (javaya gönderilir). Bu işlemlerin sonucunda java programı :

```
sum = 45
```

sonucunu verecektir. JNI boyutlu değişken çağırma fonksiyonları (metodları) şöyledir :

```
GetArrayLength  
Get< deęişkentürü >ArrayElements  
  GetBooleanArrayElements  
  GetByteArrayElements  
  GetCharArrayElements  
  GetDoubleArrayElements  
  GetFloatArrayElements  
  GetIntArrayElements  
  GetLongArrayElements  
  GetShortArrayElements  
Release< deęişkentürü >ArrayElements  
Get< deęişkentürü >ArrayRegion  
Set< deęişkentürü >ArrayRegion  
GetObjectArrayElement  
SetObjectArrayElement
```

Java dilinde tanımlanmış, fakat c dilinde direk tanımlanmamış işlemlerden birisi de hata analiz işlemidir (Catch-throw exception). Ana dil programlaması kullanırken, bu işlemi de yapmak mümkündür. Aşağıdaki CatchThrow program setinde hata sisteminin kullanılmasını görüyoruz.

Program 17.20 CatchThrow.java programı

```
class CatchThrow {  
  private native void catchThrow() throws IllegalArgumentException;  
  private void callback() throws NullPointerException {  
    throw new NullPointerException("thrown in CatchThrow.callback");  
  }  
  public static void main(String args[]) {  
    CatchThrow c = new CatchThrow();  
    try {  
      c.catchThrow();  
    } catch (Exception e) {  
      System.out.println("In Java:\n " + e);  
    }  
  }  
  static {  
    System.loadLibrary("MyImpOfCatchThrow");  
  }  
}
```

Program 17.21 CatchThrow.c programı

```
#include <jni.h>  
#include "CatchThrow.h"  
  
JNIEXPORT void JNICALL  
Java_CatchThrow_catchThrow(JNIEnv *env, jobject obj)  
{  
  jclass cls = (*env)->GetObjectClass(env, obj);  
  jmethodID mid = (*env)->GetMethodID(env, cls, "callback", "()V");
```



```

jthrowable exc;
if (mid == 0) {
    return;
}
(*env)->CallVoidMethod(env, obj, mid);
exc = (*env)->ExceptionOccurred(env);
if (exc) {
    /*
Biz aslında c’de hata konusunda gerçekten bir şey yapmıyoruz,
sadece bir mesaj iletiyoruz, hata değişkenini temizliyoruz ve
yeni bir exception gönderiyoruz. */
    jclass newExcCls;

    (*env)->ExceptionDescribe(env);
    (*env)->ExceptionClear(env);

    newExcCls = (*env)->FindClass(env, "java/lang/IllegalArgumentException");
    if (newExcCls == 0) { /* Unable to find the new exception class, give up. */
        return;
    }
    (*env)->ThrowNew(env, newExcCls, "thrown from C code");
}
}

```

bu programı çalıştırdığımızda :

```

C:\co\java\prog>java CatchThrow
Exception in thread "main" java.lang.NullPointerException: thrown in CatchThrow.
callback
    at CatchThrow.callback(CatchThrow.java:4)
    at CatchThrow.catchThrow(Native Method)
    at CatchThrow.main(CatchThrow.java:9)
In Java:
    java.lang.IllegalArgumentException: thrown from C code

```

sonucunu alırız. Buradaki

In Java:

java.lang.IllegalArgumentException: thrown from C code
mesajı c tarafından oluşturulup gönderilmiştir.

JNI Hata fonksiyonları şunlardır :

ExceptionClear
ExceptionDescribe
ExceptionOccurred

Burada iki dilli programlama kavramına bir giriş yaptık. Buradaki kavramlar size biraz zor geldiye lütfen umutsuzluğa kapılmayın, java ve C dillerini çok iyi öğrendikten sonra geri dönün. İki dilli programlama her zaman programlama dünyasının en zor işlerindedir, ayrıca jbuilder, Microsoft j++ gibi paketler java anadil (native) programlamasında işleri kolaylaştıran otomatik kod oluşturma sistemleri sunarlar, bu tür programlama araçlarıyla işinizi daha kolay hale getirebilirsiniz.