

## W8\_COMPUTER PROGRAMMING 2019 SPRING

### W8 Error control , Exception handling

#### We will use the class IO

```
import java.util.*;
import javax.swing.*;
import java.awt.Font;

class IO
{
    static Scanner input = new Scanner( System.in );
    //change font and size for JOptionPane class, //example font "Arial"
    //example size 14
    public static void setOptionPane(String font,int size)
    { UIManager.put("OptionPane.messageFont", new Font(font, Font.PLAIN, size)); }
    //array input
    public static double[] Dinput(int n)
    { double c[]=new double[n];
      for(int i=0;i<n;i++)
      { c[i]=Dinput("a["+i+"] = "); }
      return c;
    }

    public static double[][] Dinput(int n,int m)
    { double c[][]=new double[n][m];
      for(int i=0;i<n;i++)
      { for(int j=0;j<m;j++)
        { c[i][j]=Dinput("a["+i+" "+j+"] = "); }
      }
      return c;
    }

    public static int[] Iinput(int n)
    { int c[]=new int[n];
      for(int i=0;i<n;i++)
      { c[i]=Iinput("a["+i+"] = "); }
      return c;
    }

    public static int[][] Iinput(int n,int m)
    { int c[][]=new int[n][m];
      for(int i=0;i<n;i++)
      { for(int j=0;j<m;j++)
        { c[i][j]=Iinput("a["+i+" "+j+"] = "); }
      }
      return c;
    }

    public static String[] input(int n)
    { String c[]=new String[n];
      for(int i=0;i<n;i++)
      { c[i]=input("a["+i+"] = "); }
      return c;
    }

    public static String[][] input(int n,int m)
    { String c[][]=new String[n][m];
      for(int i=0;i<n;i++)
      { for(int j=0;j<m;j++)
        { c[i][j]=input("a["+i+" "+j+"] = "); }
      }
      return c;
    }

    public static String toString(double a[],int n)
    { Locale us=new Locale("us");
      String s1="[";
      for(int i=0;i<a.length;i++)
      { s1+=String.format(us, "%"+n+"f",a[i]); }
      s1+="]n";
      return s1;
    }

    public static String toString(int a[],int n)
    { String s1="[";
      for(int i=0;i<a.length;i++)
      { s1+=String.format("%"+n+"d",a[i]); }
      s1+="]n";
      return s1;
    }
}
```

```

public static String toString(String a[],int n)
{ String s1="[";
for(int i=0;i<a.length;i++)
  { s1+=String.format("% +n+ "s",a[i]);}
s1+="]n";
return s1;
}

public static String toString(double a[],int n)
{ String s1="";
for(int i=0;i<a.length;i++)
  { s1+=toString(a[i],n);}
return s1;
}

public static String toString(int a[],int n)
{ String s1="";
for(int i=0;i<a.length;i++)
  { s1+=toString(a[i],n);}
return s1;
}

public static String toString(String a[],int n)
{ String s1="";
for(int i=0;i<a.length;i++)
  { s1+=toString(a[i],n);}
return s1;
}

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

public static void Cprint(String s)
{System.out.print(s);}

public static void Cprintln(String s)
{System.out.println(s);}

public static double DCinput(String s)
{  System.out.print(s);
return Double.parseDouble(input.next());}

public static int ICinput(String s)
{ Cprint(s);return input.nextInt();}

public static String Cinput(String s)
{ Cprint(s);return input.next();}

public static double Dinput(String s)
{ double x=0;
  try{
    x=Double.parseDouble(JOptionPane.showInputDialog(s));
  } catch(NumberFormatException e) {System.out.println("number format exception");}
return x;
}
public static int Iinput(String s)
{ int x=0;
  try{
    x=Integer.parseInt(JOptionPane.showInputDialog(s));
  } catch(NumberFormatException e) {System.out.println("number format exception");}
return x;
}

public static String input(String s)
{ return JOptionPane.showInputDialog(s);}
}

```

### EX 1 try to enter a string in input

```

//Composition
class f1
{ public double func(double x)
  {return x*x-2.3*x-2.0;}
}

```

```

public class W7E1
{
    public static double bisection(f1 f,double a,double b)
    {double b1=1.1*b;
    double r=(a+b)/2.0;
    double eps=1.0e-8;
    int nmax=100;
    int i=0;
    while(Math.abs(f.func(r))>eps && i<nmax)
    {if(f.func(a)*f.func(r)<0) b=r;
    else a=r;
    r=(a+b)/2.0;;
    i++;
    }
    if(i>=nmax) r=bisection(f,a,b1);
    return r;
    }
    public static void main(String arg[])
    { //root of a function
    double a=IO.Dinput("a=");
    double b=IO.Dinput("a=");
    f1 f=new f1();
    double x0=bisection(f,a,b);
    String s="x0="+x0;
    IO.print(s);
    }
}

```

### EX2 try to give 0 as x2

```

public class dividebyzeroException extends ArithmeticException
{ private static final long serialVersionUID = 985786L;
  public dividebyzeroException() { super("number divided by zero "); }
}

```

```

import javax.swing.*;

public class W8E2
{
    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 arg[])
    {
        int x1,x2;
        double x3=0;
        x1=IO.Iinput("x1=");
        x2=IO.Iinput("x2=");
        try{
            x3=divide(x1,x2);
        } catch(dividebyzeroException e1) {System.out.println(e1.toString());x3=1.0/0.0;}
        IO.print("x3="+x3);
    }
}

```

### EX3 negative or zero box size is not excepted

```

public class zero_or_negative_sizeException extends ArithmeticException
{ private static final long serialVersionUID = 9875958L;
  public zero_or_negative_sizeException() { super("the given dimension is zero or negative "); }
}

```

```

public class box
{ double width,length,height;
  String bcolor;
  public box(double widthi,double lengthi,double heighti,String bc) throws zero_or_negative_sizeException
  { if(widthi<=0 ||lengthi<=0 || heighti<=0) throw new zero_or_negative_sizeException();
    else{ width=widthi;length=lengthi;height=heighti;bcolor=bc; }
  }
  public double area()
  {return width*length+width*height+length*height;}
}

```

```

public double volume()
{return length*width*height;}
public String toString()
{String s="width = "+width+" m length = "+length+" m height = "+height+" m area = "+area()+" m"+'\u00B2'+ " volume = "+volume()+" m"+'\u00B3'+ " color = "+bcolor;
return s;
}
}

```

### Try to input 0 or negative number for width, length or height

```

import javax.swing.JOptionPane;
class H8E3
{
public static void main(String args[])
{
double width=IO.Dinput("width = ");
double length=IO.Dinput("length = ");
double height=IO.Dinput("height = ");
try{
box firstbox= new box(width,length,height,"blue");
String s="First box : \n"+firstbox+"\n";
IO.print(s);
} catch(zero_or_negative_sizeException e1)
{ JOptionPane.showMessageDialog(null,e1.toString(),"ERROR",JOptionPane.ERROR_MESSAGE);}
System.exit(0);
}
}

```

### EX4

```

public class companyisbrokenException extends Exception
{ private static final long serialVersionUID = 4783855L;
public companyisbrokenException() { super("the company is broken"); }
}

```

```

public class company
{ public String name;
public double capital,balance,profit;
public int process;
//this method establish the company

//constructor method
public company(String is,double Ri)
{name=is;
capital=Ri;
balance=Ri;
profit=0;
process=0;
}

public void buy(double x)
{balance-=x;
process++;
if(balance<=0)
{System.out.println("company is broken "+process);}
}

public void sell(double x)
{balance+=x;
profit=balance-capital;
process++;
}

public String account() throws companyisbrokenException
{ String s="-----"+name+"-----\n";
if(balance<=0) throw new companyisbrokenException();
//s+="company is broken \n";
else
s+=" capital = "+capital+" TL\n";
s+=" balance = "+balance+" TL\n";
s+=" profit = "+profit+" TL\n";
s+=" number of processes = "+process+" \n";
return s;
}
}

```

```
}  
}
```

```
import javax.swing.*;  
public class H8E4  
{ public static void main(String args[])  
{  
    company T=new company("Defne Holding",1.0e6);  
    for(int i=0;i<100;i++) {T.buy(100);T.sell(400);}  
    String s="";  
    String s1="",s2="";  
    company A=new company("Ali Limited",100.0);  
    for(int i=0;i<100;i++) {A.buy(100);A.sell(110);}  
    try{  
        s1=T.acount();  
        s+=s1;  
        s2=A.acount();  
        s+=s2;  
    }catch(companyisbrokenException e){System.out.println(e.toString());}  
    JOptionPane.showMessageDialog(null,s,"company class test",JOptionPane.PLAIN_MESSAGE);  
}  
}
```

## HOMEWORK EXERCISES

Homework exercises will be done at home and will bring to next Thursday class printed no late exercises will be excepted. Each code should include student name id#, code plus results should be given. Homeworks will be accepted in written format plus a computer copy in pdf format will be sent to [computer\\_programming@turhancoban.com](mailto:computer_programming@turhancoban.com) adress your file name should be "group"+"week#"+"studentname+studentid#.pdf

A W1\_turhan\_coban\_0101333.pdf

B W3\_ali\_veli\_02335646.pdf

**W7HW1** : class square\_roots are given also zero\_or\_negative\_sizeException is given. Apply this exception so that if delta is less then zero program gives exception

```
public static double[] square_roots(double d[]) throws zero_or_negative_sizeException  
{}
```

```
public class zero_or_negative_sizeException extends ArithmeticException  
{ private static final long serialVersionUID = 9875958L;  
    public zero_or_negative_sizeException() { super("the given dimension is zero or negative "); }  
}
```

```
public class square_roots  
{  
    public static double[] square_roots(double d[])  
    {  
        double x[]=new double[2];  
        double a=d[2];  
        double b=d[1];  
        double c=d[0];  
        double delta=b*b-4*a*c;  
        if(delta>=0)  
        {  
            x[0]=(-b+Math.sqrt(delta))/(2*a);  
            x[1]=(-b-Math.sqrt(delta))/(2*a);  
        }  
        else  
        {IO.print("roots are complex");  
        }  
        return x;  
    }  
    public static void main(String arg[])  
    {
```

```

double d[]={1,-2,1};
double x[]=square_roots(d);
IO.print(IO.toString(x,10));
}
}

```

```

public class W8HW1
{
public static double[] square_roots(double d[]) throws zero_or_negative_sizeException
{
double x[]=new double[2];
double a=d[2];
double b=d[1];
double c=d[0];
double delta=b*b-4*a*c;
if(delta>=0)
{
x[0]=(-b+Math.sqrt(delta))/(2*a);
x[1]=(-b-Math.sqrt(delta))/(2*a);
}
else
{..... }
return x;
}
public static void main(String arg[])
{
double d[]={11,2,1};
try{
double x[]=square_roots(d);
IO.print(IO.toString(x,10));
}
catch(.....){.....}
}
}

```

## W8HW2

```

public class minuslogarithmException extends ArithmeticException
{
public minuslogarithmException() { super("logarithm of a minus number "); }
}

```

Complete the progra so that it will give **minuslogarithmException**

```

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

public class H8HW2
{
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;
try{
    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);
} catch(.....) {.....}
} // end method main
} // end class

```

### W8HW3

Rational number class is given

```

import javax.swing.JOptionPane; // giriş çıkış
// class rational numbers
public class rational {

    public int numerator;
    public int denominator;

    // constructor methods
    public rational(int numerator,int ndenominator) throws dividebyzeroException
    {
        numerator=numerator;
        if(ndenominator==0) throw new dividebyzeroException();
        denominator=ndenominator;
    }
    public rational(rational c ) throws dividebyzeroException
    {
        numerator=c.numerator;
        if(c.denominator==0) throw new dividebyzeroException();
        denominator=c.denominator;
    }
    //calculation methods
    public int leastcommondivisor()
    { // least common divisor
        int n=numerator;
        int m=denominator;
        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 simplify()
    { //simplify the rational number by using least common divisor
        int isaret=1;
        if(numerator<0)
        { isaret=-isaret;
          numerator=-numerator;
        }
        if(denominator<0)
        { isaret=-isaret;
          denominator=-denominator;
        }
        if(denominator==0) throw new dividebyzeroException();
        int ebob=leastcommondivisor();
        ebob=Math.abs(ebob);
        numerator=isaret*numerator/ebob;
        denominator=denominator/ebob;
    }
    public double toDouble()
    { //double number equivalent of the rational numbers
        return ((double)numerator/(double)denominator);
    }

    public void add(rational sag)

```

```

//addition of rational numbers
numerator = numerator*sag.denominator + sag.numerator*denominator;
denominator = denominator*sag.denominator;
simplify();
}
public void subtract(rational sag)
{//subtraction of rational numbers
numerator = numerator*sag.denominator - sag.numerator*denominator;
denominator = denominator*sag.denominator;
simplify();
}
public void multiply(rational sag )
{
numerator = numerator*sag.numerator;
denominator = denominator*sag.denominator;
simplify();
}
public void divide(rational sag )
{
numerator = numerator*sag.denominator;
denominator = denominator*sag.numerator;
simplify();
}
public String toString()
{// yazima hazır rational formda String değişkenini iletir.
String b="";
if(Math.abs(denominator)!=1)
{
b=b+"( "+numerator+" / "+denominator+" )";
}
else
{
b=b+numerator+" ";
}
return b;
}
};
//end of class rational

```

Write a main program and add two rational numbers

```

rational r1=new rational(1,3);
rational r2=new rational(1,0);
r2.add(r1);
IO.print(""+r2);

```

inside a try and catch block, then try

```

rational r1=new rational(1,3);
rational r2=new rational(1,7);
r2.add(r1);
IO.print(""+r2);

```