W7_COMPUTER PROGRAMMINING 2019 SPRING

W7 Further class concepts, polymorphism, abstract class, composition, inheritance, indirect referencing, lambda variables

We will use the class IO

```
 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));}

   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)
  {     return Double.parseDouble(JOptionPane.showInputDialog(s));}

  public static int Iinput(String s)
  {     return Integer.parseInt(JOptionPane.showInputDialog(s));}

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

EX 1 Composition

```
 //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

Abstract class f_x.java

```
//abstract class
abstract public class f_x
{abstract public double func(double x);}
```

```
class f1 extends f_x
{ public double func(double x)
  {return x*x-2.3*x-2.0;}
}

public class W7E2
{
  public static double bisection(f_x 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);
  }
}
```

EX3

Interface if_x.java

```
interface if_x
{public double func(double x);}
```

```
class f1 implements if_x
{ public double func(double x)
  {return x*x-2.3*x-2.0;}
}

public class W7E3
{
  public static double bisection(if_x 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);
  }
}
```

EX4 Lambda variables

```
 public class W7E4
{
  public static double bisection(if_x 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=");
    if_x f=x->x*x-2.3*x-2.0;
    double x0=bisection(f,a,b);
    String s="x0="+x0;
    IO.print(s);
  }
}
```

EX5 Lambda variables

```
interface if_xy
{public double func(double x[]);}
```

```
public class W7E5
{
  public static void main(String arg[])
  { //root of a function
    double x=IO.Dinput("x=");
    double y=IO.Dinput("y=");
    if_xy f=(double z[])->z[0]*z[0]+z[1]*z[1];
    double z[]={x,y};
    String s="x="+z[0]+"y="+z[1]+"\nx*x+y*y = "+f.func(z);
    IO.print(s);
  }
}
```

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 adress your file name should be "group"+"week#"+studentname+studentid#.pdf

A W1_turhan_coban_0101333.pdf

B W3_ali_veli_02335646.pdf

W7HW1 : abstract class y_x is given calculate function value and derivative of **y(x)= x*x-2.3*x-2.0;**

```
//abstract class y_x.java
abstract public class y_x
{abstract public double func(double x);
 //derivative of func
 public double dfunc(double x)
 {double h=0.0001;
  double dy=(func(x+h)-func(x-h))/(2.0*h);
  return dy;
 }
}
```

```
 class y1 extends y_x
{ public double func(double x)
  {return x*x-2.3*x-2.0;}
}
public class W7HW1
{
  public static void main(String arg[])
  { //
              double x=IO.Dinput("x=");
              y1 y=new y1();
……….
    String s=……
    IO.print(s);
  }
}
```

W7HW2 Interface iy_x.java is given calculate function value and derivative of **y(x)= x*x-2.3*x-2.0;**

```
@FunctionalInterface
interface iy_x
{  public double func(double x);
//first order derivative
default double dfunc(double x)
{double h=1.0e-3;
 double dy=(func(x+h)-func(x-h))/(2.0*h);
 return dy;
 }
}
```

```
class y1 implements iy_x
{ public double func(double x)
  {return x*x-2.3*x-2.0;}
}
public class W7HW2
{
  public static void main(String arg[])
  { //
              double x=IO.Dinput("x=");
              y1 y=new y1();
……..
    String s=……..
    IO.print(s);
  }
}
```

W7HW3 Interface iy_x.java is given calculate function value and derivative of **y(x)= x*x-2.3*x-2.0** by using lambda variables

W7HW4 Interface iz_xy is given as

```
@FunctionalInterface
interface iz_xy
{public double func(double x[]);
default double[] dfunc(double x[])
{double h=1.0e-3;
int n=x.length;
double dy[]=new double[n];
double z1[]=new double[n];
```

```
double z2[]=new double[n];
for(int i=0;i<n;i++)
{ for(int j=0;j<n;j++) {z1[i]=x[i];z2[i]=x[i];}
 z1[i]=x[i]+h;
 z2[i]=x[i]-h;
 dy[i]=(func(z1)-func(z2))/(2.0*h);
 z1[i]=x[i];z2[i]=x[i];
}
return dy;
}
}
```

```
public class W7HW4
{
  public static void main(String arg[])
  { //
              double x=IO.Dinput("x=");
              double y=IO.Dinput("y=");
              iz_xy f2=(double z[])->z[0]*z[0]+z[1]*z[1];
              double z[]={x,y};
              double dz[]=f2.dfunc(z);
             String s="x="+z[0]+"y="+z[1]+"\nz(x,y)="+f2.func(z)+"dz/dx="+dz[0]+"dz/dy="+dz[1];
            IO.print(s);
  }
}
```

Calculate

$Z=2x+3xy+3.3*x^2-1.23y^2$

z=2.0*x[0]+3.0*x[0]*x[1]+3.3*x[0]*x[0]-1.23*x[1]*x[1]

function value and its derivatives by using lambda variables