Austin Loza

Perm 9715590

CS 56 Winter

February 10, 2017

Homework Assignment 03

Q1.~~~~~~

```
public class SavingsAccount {
   public static double annualInterestRate;
   private double savingsBalance;

   public SavingsAccount(double savings){
      savingsBalance = savings;

   }

   public double calculateMonthlyInterest(){
      double x;
      x = savingsBalance * (annualInterestRate/12);
      savingsBalance += x ;
      return savingsBalance;
   }

   public static void modifyInterestRate(double percent){
      annualInterestRate = percent * 0.01;
   }

   public static void main(String[] args){
      SavingsAccount saver1, saver2;
      saver1 = new SavingsAccount(2000);
      saver2 = new SavingsAccount(3000);

      SavingsAccount.modifyInterestRate(4);

      for(int i = 0; i < 12; i++){
         System.out.println("Saver 1 balance:");
         System.out.println(saver1.calculateMonthlyInterest());
         System.out.println("Saver 2 balance:");
         System.out.println(saver2.calculateMonthlyInterest());

      }
```

```
        System.out.println("Next month's saver balance after the interest rate increases to
5%:");

        SavingsAccount.modifyInterestRate(5);

        System.out.println("Saver 1 balance:");
        System.out.println(saver1.calculateMonthlyInterest());
        System.out.println("Saver 2 balance:");
        System.out.println(saver2.calculateMonthlyInterest());

    }
}
```

This outputs:

```
Saver 1 balance:
2006.6666666666667
Saver 2 balance:
3010.0
Saver 1 balance:
2013.3555555555556
Saver 2 balance:
3020.0333333333333
Saver 1 balance:
2020.066740740741
Saver 2 balance:
3030.100111111111
Saver 1 balance:
2026.80029654321
Saver 2 balance:
3040.2004448148145
Saver 1 balance:
2033.5562975316873
Saver 2 balance:
3050.3344462975306
Saver 1 balance:
2040.3348185234595
Saver 2 balance:
3060.502227785189
Saver 1 balance:
2047.1359345852043
Saver 2 balance:
3070.7039018778064
Saver 1 balance:
```

```
2053.9597210338216
Saver 2 balance:
3080.9395815507323
Saver 1 balance:
2060.8062534372675
Saver 2 balance:
3091.2093801559013
Saver 1 balance:
2067.6756076153915
Saver 2 balance:
3101.5134114230877
Saver 1 balance:
2074.567859640776
Saver 2 balance:
3111.8517894611646
Saver 1 balance:
2081.4830858395785
Saver 2 balance:
3122.2246287593684
Next month's saver balance after the interest rate increases to 5%:
Saver 1 balance:
2090.155932030577
Saver 2 balance:
3135.2338980458658
```

Q2. ~~~~~~

```
public class Complex {
    private float realPart, imaginaryPart;

    public Complex(){
        realPart = 0;
        imaginaryPart = 0;
    }

    public Complex(float realP, float imagP){
        realPart = realP;
        imaginaryPart = imagP;
    }

    public float real(){
        return realPart;
    }
```

```
    public float imag(){
        return imaginaryPart;
    }

    public Complex add(Complex other){
        Complex out;
        float x,y;
        x = realPart + other.real();
        y = imaginaryPart + other.imag();
        out = new Complex(x,y);
        return out;
    }

    public Complex mult(Complex other){
        Complex out;
        float x, y;
        x = realPart * other.real();
        x = x - (imaginaryPart * other.imag());
        y = realPart * other.imag() + other.real() * imaginaryPart;
        out = new Complex(x, y);
        return out;
    }

    public void Print(){
        System.out.printf("(%f,%f)", realPart, imaginaryPart);
    }

    public static void Print(Complex x ){
        System.out.printf("(%f,%f)", x.real(), x.imag());
    }

    public static void main(String[] args){
        Complex a, b;
        a = new Complex(1,2);
        b = new Complex(3,4);
        Print(a.add(b));
        a.Print();
        b.Print();
        Print(b.mult(a));
        b.Print();

    }


}
```

This outputs:

(4.000000,6.000000)(1.000000,2.000000)(3.000000,4.000000)(-5.000000,10.000000)
(3.000000,4.000000)

Q3.~~~~~~

```
public class Point {
   private double x, y;
   public Point(){
      x= 0;
      y = 0;
   }

   public void copy(Point dup){
      x = dup.getX();
      y = dup.getY();
   }

   public double getX(){
      return x;
   }

   public double getY(){
      return y;
   }

   public void setX(double set){
      x = set;
   }

   public void setY(double set){
      y = set;
   }

   public double rise(Point other){
      double rise;
      rise = Math.abs(y-other.getY());
      return rise;
   }

   public double run(Point other){
```

```
        double run;
        run =  Math.abs(x - other.getX());
        return run;
    }

    public double distance(Point other){
        double rise, run, dist;
        rise = Math.abs(y-other.getY());
        run =  Math.abs(x - other.getX());
        dist = Math.pow(rise, 2) + Math.pow(run, 2);
        dist = Math.sqrt(dist);
        return dist;
    }
}


public class Quadrilateral extends Point {
    protected Point a,b,c,d;
    //   a------b
    //   |      |
    //   |      |
    //   c------d
    protected double qheight, qlength;

    public Quadrilateral(){
        qheight = 0;
        qlength = 0;
    }
    public Quadrilateral(Point a_s, Point b_s, Point c_s, Point d_s){
        qheight = a.rise(c);
        qlength = a.run(b);
    }

    public double getQHeight(){
        qheight = a.rise(c);
        return qheight;
    }

    public double getQLength(){
        qlength = c.run(d);
        return qlength;
    }



}
```

```
}


public class Trapezoid extends Quadrilateral {
   public double getRunTop(){
      return a.run(b);
   }
   public double getRunBase(){
      return c.run(d);
   }

   public double area(){
      double area;
      area = getRunTop() + getRunBase();
      area /= 2;
      area *= qheight;
      return area;
   }
}

public class Parallelogram extends Quadrilateral{
   public double area(){
      return qlength * qheight;
   }
}

public class Rectangle extends Parallelogram {


}

public class Square extends Rectangle {

}
```

Q4.~~~~~~

```
public class PieceWorker extends Employee {
   private double wages;
   private int pieces;
   private Employee[] array;

   public double earnings(){
      double earned;
      earned = wages*pieces;
      return earned;
   }
```

```
    public String toString() {
        String returnMe;
        returnMe = "PieceWorker ";
        double x = earnings();
        returnMe += x;
        return returnMe;

    }

}
```

Q5.~~~~~~

```
int[] a = { 5, 4, 3, 2, 1};
MyReport t = new MyReport();
System.out.printf( "The total is %d\n", t.sum(a, a.length) ); //Prints 15
System.out.printf( "One value is %d\n", a[1]); //Prints 0
System.out.printf( "The count is %d\n", a.length); // Prints 5
```

The output:

```
15
0
5
```

Q6.~~~~~~

```
int secondLargest(int[] x){
        int hi = Integer.MIN_VALUE;
        int secHi = Integer.MIN_VALUE;
        for(int num: x){
            if(num > hi){
                secHi = hi;
                hi = num;
            }
            else if (num > secHi){
                secHi = num;

            }
            return secHi;
        }
```

For an array [4,2,6,7,8,3] the method returns 7
For an array [3,8,9,15,2,3] the method returns 9
For an array [1,2,3,3] the method returns 2

Q7.~~~~~~

Constructors are called. First, the constructor for A is called, then the constructor for B is called.

Q8.~~~~~~

```
public class FlyingDuck{
	public FlyingDuck(){
		System.out.println(“A new flying duck.”);
	}
}

public class MallardDuck extends FlyingDuck{
	public MallardDuck(){
		System.out.println(“A new mallard duck is created.”);
	}
}
```

Q9.~~~~~~

```
public class ClassA{

	public static type m(){
		ClassB B = new ClassB();
		B.method1();
	}


}

public class ClassB{
	public static type method1(){
		C.method2();
	}
}

public class ClassC{
	public static type method2(){}
}
```

Q10.~~~~~~

```
public class ClassA{
	public static type M1(){}
}

public class ClassB{
	public static type m(){
		A.M1();
		C.M2();
		C.M3();
	}
}

public class ClassC{

	public static type M2(){
		D.M4();
	}
	public static type M3(){
		D.M5();
	}
}


public class ClassD{
	public static type M4(){}
	public static type M5(){}
}
```