Lab sheet 8

BankAcc Class

```
package com.mycompany.bankmain;

abstract class BankAcc

{

    private String accountNumber;        private double
balance;        public BankAcc(String accountNumber,
double balance)

    {

        this.accountNumber = accountNumber;

        this.balance = balance;

    }

    public String getAccountNumber() {
return accountNumber;

    }

    public void setAccountNumber(String accountNumber)
{     this.accountNumber = accountNumber;

    }

    public double getBalance() {
return balance;

    }

    public void setBalance(double balance) {
this.balance = balance;

    }

    public abstract double calculateInterest();

}
```

SavingsAcc Class

```java
package com.mycompany.bankmain;

class SavingsAcc extends BankAcc

{

  public SavingsAcc(String accountNumber, double balance)

  {

    super(accountNumber, balance);

  }


  @Override    public double
calculateInterest()

  {

    return getBalance() * 0.12;

  }
}



return getBalance() * 0.02;

  }

}
```

CheckingAcc Class

```java
package com.mycompany.bankmain;

class CheckingAcc extends BankAcc

{

  public CheckingAcc(String accountNumber, double balance)

  {

    super(accountNumber, balance);

  }

  @Override    public double
calculateInterest()

  {
```

BankMain

```java
package com.mycompany.bankmain;

public class BankMain
{
    public static void main(String[] args)
    {
        double checkingBalance = 1000000;        double
savingsBalance = 20000000;
CheckingAcc checkingAcc = new CheckingAcc("CHK12345", checkingBalance);        ,
SavingsAcc savingsAcc = new SavingsAcc("SAV67890", savingsBalance);
double checkingInterest checkingAcc.calculateInterest();
double savingsInterest = savingsAcc.calculateInterest();


 System.out.printf("Interest earned in the checking account: $%.2f%n", checkingInterest);
 System.out.printf("Interest earned in the savings account: $%.2f%n", savingsInterest);
    }
}
```

Interest earned in the checking account: $20000.00

Interest   earned   in   the   savings   account:

$2400000.00

Exercise 2

Circle class

```java
class Circle implements Shape
{
    private double radius;

    public Circle(double radius)
    {
        this.radius = radius;
    }

    public double getRadius()
    {
        return radius;
    }
```

```java
   public void setRadius(double radius)
   {
      this.radius = radius;
   }

   @Override    public double calculateArea()
   {
      return Math.PI * radius * radius;
   }

   @Override    public double calculatePerimeter()
   {
      return 2 * Math.PI * radius;
   }
}
```

Shape Interface

```java
interface Shape
{
   double calculateArea();
   double calculatePerimeter();
}
```

Rectangle Class

```java
class Rectangle implements Shape
{
   private double length;
private double width;

   public Rectangle(double length, double width)
   {
      this.length = length;
      this.width = width;
   }

   public double getLength()
   {
      return length;
   }

   public void setLength(double length)
   {
      this.length = length;
   }

   public double getWidth()
   {
      return width;
   }

   public void setWidth(double width)
   {
      this.width = width;
   }

   @Override
   public double calculateArea()
   {
      return length * width;
   }

   @Override
   public double calculatePerimeter()
   {
      return 2 * (length + width);
   }
}
```

Triangle class

```java
class Triangle implements Shape
{
    private double side1;
private double side2;
private double side3;

    public Triangle(double side1, double side2, double side3)
    {
        this.side1 = side1;
this.side2 = side2;
this.side3 = side3;
    }

    public double getSide1()
    {
        return side1;
    }

    public void setSide1(double side1)
    {
        this.side1 = side1;
    }

    public double getSide2()
    {
        return side2;
    }

    public void setSide2(double side2)
    {
        this.side2 = side2;
    }

    public double getSide3()
    {
        return side3;
    }

    public void setSide3(double side3)
    {
        this.side3 = side3;
    }

    @Override
    public double calculateArea()
    {
```

```java
        double s = (side1 + side2 + side3) / 2;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }

    @Override
    public double calculatePerimeter()
    {
        return side1 + side2 + side3;
    }
}
```