Exercise 01:

Create a class named "BankAccount" with private instance variables "accountNumber" and "balance." Implement encapsulation by providing public getter and setter methods for both variables. Additionally, create an abstract method called "calculateInterest" in the "BankAccount" class. Implement two subclasses, "SavingsAccount" and "CheckingAccount," that extend the "BankAccount" class and provide their own implementations of the "calculateInterest" method. Write the implementation code for the getter and setter methods in the "BankAccount" class, and the "calculateInterest" method in both the "SavingsAccount" and "CheckingAccount" classes. Assuming that the interest for saving is 12% and checking is 2% (both private variables), find out What will be the interest for a person with 1 million in his checking and 20 million in his saving account.

package com.mycompany.project04;

public abstract class BankAccount {

protected int accno;

protected double balance;

public int getAccno()

{

return accno;

}

public void setAccno(int accno)

{

this.accno=accno;

}

public double getBal()

{

return balance;

}

public void setBal(double balance)

{

this.balance=balance;

}

public abstract double calculateInterest();

}

Exercise 02:

Create an interface called "Shape" with two abstract methods: "double calculateArea()" and "double calculatePerimeter()". Implement the "Shape" interface in three classes: "Circle", "Rectangle", and "Triangle". Each class should have private instance variables relevant to its shape, and provide public getter and setter methods for these variables. Additionally, each class should define a constructor that initializes the instance variables. Write the implementation code for the "Shape" interface, the getter and setter methods in each class, and the constructors in each class.

package com.mycompany.test02;

public interface Shape {

double calculateArea();

double calculateperimeter();

}

package com.mycompany.test02;

public class Circle implements Shape {

private double radius;

public Circle(double radius)

{

this.radius=radius;

}

public double getR()

{

return radius;

}

public void setR(double radius)

{

this.radius=radius;

}

public double calculateArea()

{

return Math.PI\*radius;

}

public double calculateperimeter()

{

return 2\*Math.PI\*radius;

}

}

package com.mycompany.test02;

public class Rectangal implements Shape {

private int width;

private int length;

public Rectangal(int width,int length)

{

this.width=width;

this.length=length;

}

public int getW()

{

return width;

}

public void setW(int width)

{

this.width=width;

}

public int getL()

{

return length;

}

public void setL(int length)

{

this.length=length;

}

public double calculateArea()

{

return width\*length;

}

public double calculateperimeter()

{

return 2\*(width+length);

}

}

package com.mycompany.test02;

public class Triangle implements Shape{

private int sideA;

private int sideB;

private int sideC;

public Triangle(int sideA,int sideB,int sideC)

{

this.sideA=sideA;

this.sideB=sideB;

this.sideC=sideC;

}

public int getA()

{

return sideA;

}

public int getB()

{

return sideB;

}

public int getC()

{

return sideC;

}

public void setA(int sideA)

{

this.sideA=sideA;

}

public void setB(int sideB)

{

this.sideB=sideB;

}

public void setC(int sideC)

{

this.sideC=sideC;

}

public double calculateArea()

{

double s=(sideA+sideB+sideC)/2;

return Math.sqrt(s\*(s-sideA)\*(s-sideB)\*(s-sideC));

}

public double calculateperimeter()

{

return sideA+sideB+sideC;

}

}