**Java lab-08**

**1).**

package com.mycompany.bankobjects;

public class BankObjects

{

public static void main(String[] args)

{

CheckingAccount c1=new CheckingAccount();

c1.setbalance(200000.00f);

c1.calculateInterest();

System.out.println("Interest for checking account:"+c1.calculateInterest());

SavingAccount s1=new SavingAccount();

s1.setbalance(300000.00f);

s1.calculateInterest();

System.out.println("Interest for saving account:"+s1.calculateInterest());

}

}

package com.mycompany.bankobjects;

public abstract class BankAccount

{

//data

private int accountNumber;

private float balance;

//getter and setter

//getter

public void setaccountNumber( int newAcNumber)

{

this.accountNumber=newAcNumber;

}

public void setbalance( float newbalance)

{

this.balance=newbalance;

}

//setter

public int getaccountNumber()

{

return accountNumber;

}

public float getbalance()

{

return balance;

}

public abstract float calculateInterest();

}

package com.mycompany.bankobjects;

public class CheckingAccount extends BankAccount

{

@Override

public float calculateInterest()

{

return (getbalance()\*2)/100;

}

}

package com.mycompany.bankobjects;

public class SavingAccount extends BankAccount

{

@Override

public float calculateInterest()

{

return (getbalance()\*12)/100;

}

}

**2).**

// Shape interface

interface Shape {

double calculateArea();

double calculatePerimeter();

}

// Circle class implementing the Shape interface

class Circle implements Shape {

private double radius;

// Constructor

public Circle(double radius) {

this.radius = radius;

}

// Getter for radius

public double getRadius() {

return radius;

}

// Setter for radius

public void setRadius(double radius) {

this.radius = radius;

}

// Implementing calculateArea method for Circle

@Override

public double calculateArea() {

return Math.PI \* radius \* radius;

}

// Implementing calculatePerimeter method for Circle

@Override

public double calculatePerimeter() {

return 2 \* Math.PI \* radius;

}

}

// Rectangle class implementing the Shape interface

class Rectangle implements Shape {

private double length;

private double width;

// Constructor

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

// Getters for length and width

public double getLength() {

return length;

}

public double getWidth() {

return width;

}

// Setters for length and width

public void setLength(double length) {

this.length = length;

}

public void setWidth(double width) {

this.width = width;

}

// Implementing calculateArea method for Rectangle

@Override

public double calculateArea() {

return length \* width;

}

// Implementing calculatePerimeter method for Rectangle

@Override

public double calculatePerimeter() {

return 2 \* (length + width);

}

}

// Triangle class implementing the Shape interface

class Triangle implements Shape {

private double base;

private double height;

private double side1;

private double side2;

private double side3;

// Constructor

public Triangle(double base, double height, double side1, double side2, double side3) {

this.base = base;

this.height = height;

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

// Getters for base, height, and sides

public double getBase() {

return base;

}

public double getHeight() {

return height;

}

public double getSide1() {

return side1;

}

public double getSide2() {

return side2;

}

public double getSide3() {

return side3;

}

// Setters for base, height, and sides

public void setBase(double base) {

this.base = base;

}

public void setHeight(double height) {

this.height = height;

}

public void setSide1(double side1) {

this.side1 = side1;

}

public void setSide2(double side2) {

this.side2 = side2;

}

public void setSide3(double side3) {

this.side3 = side3;

}

// Implementing calculateArea method for Triangle

@Override

public double calculateArea() {

return 0.5 \* base \* height;

}

// Implementing calculatePerimeter method for Triangle

@Override

public double calculatePerimeter() {

return side1 + side2 + side3;

}

}

public class TestShapes {

public static void main(String[] args) {

// Test Circle

Circle circle = new Circle(5.0);

System.out.println("Circle:");

System.out.println("Radius: " + circle.getRadius());

System.out.println("Area: " + circle.calculateArea());

System.out.println("Perimeter: " + circle.calculatePerimeter());

System.out.println();

// Test Rectangle

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Rectangle:");

System.out.println("Length: " + rectangle.getLength());

System.out.println("Width: " + rectangle.getWidth());

System.out.println("Area: " + rectangle.calculateArea());

System.out.println("Perimeter: " + rectangle.calculatePerimeter());

System.out.println();

// Test Triangle

Triangle triangle = new Triangle(3.0, 4.0, 5.0, 4.0, 3.0);

System.out.println("Triangle:");

System.out.println("Base: " + triangle.getBase());

System.out.println("Height: " + triangle.getHeight());

System.out.println("Side1: " + triangle.getSide1());

System.out.println("Side2: " + triangle.getSide2());

System.out.println("Side3: " + triangle.getSide3());

System.out.println("Area: " + triangle.calculateArea());

System.out.println("Perimeter: " + triangle.calculatePerimeter());

}

}