Name : N D A Pinsara

ID: 28532

LAB 08

01.

```
abstract class BankAccount {

    private int accountNumber;
    private double balance;
    private double savingInterest;
    private double checkingInterest;

    public BankAccount(int accountNumber, double balance, double savingInterest, double checkingInterest) {
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.savingInterest = savingInterest;
        this.checkingInterest = checkingInterest;
    }

    public int getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }
```

```
    public double getBalance() {
        return balance;
    }

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

    public abstract double calculateInterest();
}

class SavingsAccount extends BankAccount {

    public SavingsAccount(int accountNumber, double balance, double savingInterest, double checkingInterest) {
        super(accountNumber, balance, savingInterest, checkingInterest);
    }

    @Override
    public double calculateInterest() {
        return balance * savingInterest;
    }
}

class CheckingAccount extends BankAccount {

    public CheckingAccount(int accountNumber, double balance, double savingInterest, double checkingInterest) {
        super(accountNumber, balance, savingInterest, checkingInterest);
```

```
    }

    @Override
    public double calculateInterest() {
        return balance * checkingInterest;
    }
}

public class Main {

    public static void main(String[] args) {
        SavingsAccount savingsAccount = new SavingsAccount(123456, 1000000, 0.12, 0.02);
        CheckingAccount checkingAccount = new CheckingAccount(654321, 2000000, 0.12, 0.02);

        System.out.println("Interest for savings account: " + savingsAccount.calculateInterest());
        System.out.println("Interest for checking account: " + checkingAccount.calculateInterest());
    }
}
```

02.

```
interface Shape {

    double calculateArea();

    double calculatePerimeter();
```

```
}

class Circle implements Shape {

    private double radius;

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

    public double getRadius() {
        return radius;
    }

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

    @Override
    public double calculateArea() {
        return Math.PI * Math.pow(radius, 2);
    }

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

```
class Rectangle implements Shape {

    private double length;
    private double breadth;

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

    public double getLength() {
        return length;
    }

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

    public double getBreadth() {
        return breadth;
    }

    public void setBreadth(double breadth) {
        this.breadth = breadth;
    }

    @Override
    public double calculateArea() {
        return length * breadth;
```

```
    }

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

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

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

public class Main {
```

```
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        Rectangle rectangle = new Rectangle(10, 5);
        Triangle triangle = new Triangle(5, 10, 12);

        System.out.println("The area of the circle is: " + circle.calculateArea());
        System.out.println("The perimeter of the circle is: " + circle.calculatePerimeter());

        System.out.println("The area of the rectangle is: " + rectangle.calculateArea());
        System.out.println("The perimeter of the rectangle is: " + rectangle.calculatePerimeter());

        System.out.println("The area of the triangle is: " + triangle.calculateArea());
        System.out.println("The perimeter of the triangle is: " + triangle.calculatePerimeter());
    }
}
```