Practical 08

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.

Answers

// BankAccount Class:

```
public abstract class BankAccount {
    private int accountNumber;
    private double balance;
    // Getter and Setter for accountNumber
    public int getAccountNumber() {
        return accountNumber;
    }

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

    // Getter and Setter for balance
    public double getBalance() {
        return balance;
    }

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

    // Abstract method to calculate interest
    public abstract double calculateInterest();
}
```

// SavingsAccount Class

```
public class SavingsAccount extends BankAccount {
    private final double interestRate = 0.12; // 12%

    @Override
    public double calculateInterest() {
        return getBalance() * interestRate;
    }
}
// CheckingAccount Class
public class CheckingAccount extends BankAccount {
    private final double interestRate = 0.02; // 2%
```

```
  @Override
  public double calculateInterest() {
    return getBalance() * interestRate;
  }
}
```

// calculate the interest for a person with 1 million in his checking and 20 million in his saving account:

```
public class TestBankAccounts {
  public static void main(String[] args) {
    SavingsAccount savingsAccount = new SavingsAccount();
    savingsAccount.setBalance(20000000); // 20 million

    CheckingAccount checkingAccount = new CheckingAccount();
    checkingAccount.setBalance(1000000); // 1 million

    double savingInterest = savingsAccount.calculateInterest();
    double checkingInterest = checkingAccount.calculateInterest();

    System.out.println("Interest for Savings Account: " + savingInterest);
    System.out.println("Interest for Checking Account: " + checkingInterest);
  }
}
```

OUTPUT

```
Interest for Savings Account: 2400000

Interest for Checking Account: 20000
```

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.

Answers

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


class Circle implements Shape {
  private double radius;


  public Circle(double radius) {
    this.radius = radius;
  }
  public double calculateArea() {
    return Math.PI * radius * radius;
  }
  public double calculatePerimeter() {
    return 2 * Math.PI * radius;
  }
}
class Rectangle implements Shape {
  private double width;
  private double height;

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

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

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

public class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the radius of the circle: ");
    double radius = scanner.nextDouble();

    System.out.print("Enter the width and height of the rectangle: ");
    double width = scanner.nextDouble();
    double height = scanner.nextDouble();

    System.out.print("Enter the side lengths of the triangle: ");
    double side1 = scanner.nextDouble();
    double side2 = scanner.nextDouble();
    double side3 = scanner.nextDouble();

    Circle circle = new Circle(radius);
    Rectangle rectangle = new Rectangle(width, height);
    Triangle triangle = new Triangle(side1, side2, side3);
  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());

 }

}
```