Rajalakshmi Engineering College

Name: Alvin .B
Email: 240701034@rajalakshmi.edu.in
Roll no: 240701034
Phone: 9677000577
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 7_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

John is developing a car loan calculator and has structured his program using two interfaces, Principal and InterestRate, defining methods for principal and interest rate retrieval.

The Loan class implements these interfaces, taking principal and annual interest rates as parameters. User input is solicited for these values, and the program ensures their validity before performing calculations. If input values are invalid (less than or equal to zero), an error message is displayed.

Note: Total interest = principal * interest rate * years

Input Format

The first line of input consists of a double value P, representing the principal.

The second line consists of a double value R, representing the annual interest rate.

The third line consists of an integer value N, representing the loan duration in years.

Output Format

If the input values are valid, print "Total interest paid: Rs. " followed by a double value, representing the total interest paid, rounded off to two decimal places.

If the input values are invalid (negative or zero values for principal, annual interest rate, or loan duration), print "Invalid input values!".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20000.00
0.05
5

Output: Total interest paid: Rs.5000.00

Answer

```
import java.util.Scanner;

interface Principal {
   double getPrincipal();
}

interface InterestRate {
   double getAnnualInterestRate();
}

class Loan implements Principal, InterestRate {
   private double principal;
   private double interestRate;

   public Loan(double principal, double interestRate) {
       this.principal = principal;
       this.interestRate = interestRate;
   }

   public double getPrincipal() {
       return principal;
   }

   public double getAnnualInterestRate() {
       return interestRate;
   }

   public double calculateTotalInterest(int years) {
       return principal * interestRate * years;
   }

   public boolean isValid(int years) {
       return principal > 0 && interestRate > 0 && years > 0;
   }

   public void displayTotalInterest(int years) {
       if (isValid(years)) {
           double totalInterest = calculateTotalInterest(years);
           System.out.printf("Total interest paid: Rs.%.2f\n", totalInterest);
       } else {
           System.out.println("Invalid input values!");
       }
   }
}
public class Main {
   public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);

       double carPrice = scanner.nextDouble();

       double annualInterestRate = scanner.nextDouble();

       int loanDuration = scanner.nextInt();

       if (carPrice <= 0 || annualInterestRate <= 0 || loanDuration <= 0) {
           System.out.println("Invalid input values!");
           return;
       }

       Loan carLoan = new Loan(carPrice, annualInterestRate);
       double totalInterest = carLoan.calculateTotalInterest(loanDuration);

       System.out.printf("Total interest paid: Rs.%.2f%n", totalInterest);
   }
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement:

Sam is developing a geometry application and needs a class for trapezoid calculations. Create a "Trapezoid" class implementing a "ShapeInput" interface with a method to input trapezoid dimensions.

Also, implement a "ShapeCalculator" interface with methods to compute area and perimeter. In the "Main" class, instantiate Trapezoid, gather user input, and display the calculated area and perimeter with two decimal places.

Note

Area of Trapezoid = (1/2) * (base1 + base2) * height

Perimeter of Trapezoid = base1 + base2 + side1 + side2

Input Format

The first line of input is a double-point value representing base1 of the trapezoid.

The second line of input is a double-point value representing base2 of the trapezoid.

The third line of input is a double-point value representing the height of the trapezoid.

The fourth line of input is a double-point value representing side1 of the trapezoid.

The fifth line of input is a double-point value representing side2 of the trapezoid.

Output Format

The output displays the two lines of the calculated area (double type) and perimeter (double type) of the trapezoid, each rounded to two decimal places in the following format:

"Area of the Trapezoid: <<calculated area>>".

Perimeter of the Trapezoid: <<calculated perimeter>>".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1.0
2.0
1.0
3.0
1.0

Output: Area of the Trapezoid: 1.50
Perimeter of the Trapezoid: 7.00

Answer

```
import java.util.Scanner;

import java.util.Scanner;

interface ShapeInput {
   void getInput();
}

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

```
class Trapezoid implements ShapeInput, ShapeCalculator {
   private double base1;
   private double base2;
   private double height;
   private double side1;
   private double side2;

   public void getInput() {
      Scanner sc = new Scanner(System.in);
      base1 = sc.nextDouble();
      base2 = sc.nextDouble();
      height = sc.nextDouble();
      side1 = sc.nextDouble();
      side2 = sc.nextDouble();
   }

   public double calculateArea() {
      return 0.5 * (base1 + base2) * height;
   }

   public double calculatePerimeter() {
      return base1 + base2 + side1 + side2;
   }

   public void displayResults() {
      System.out.printf("Area of the Trapezoid: %.2f\n", calculateArea());
      System.out.printf("Perimeter of the Trapezoid: %.2f\n",
calculatePerimeter());
   }
}


public class Main {
   public static void main(String[] args) {
      Trapezoid trapezoid = new Trapezoid();
      trapezoid.getInput();

      double area = trapezoid.calculateArea();
      double perimeter = trapezoid.calculatePerimeter();

      System.out.println("Area of the Trapezoid: " + String.format("%.2f", area));
      System.out.println("Perimeter of the Trapezoid: " + String.format("%.2f",
perimeter));
```

```
    }
}
```

Status : Correct **_Marks : 10/10_**

3. Problem Statement

Jeevan is developing a fitness-tracking application to monitor daily physical activity.

The application incorporates a FitnessTracker class that implements two interfaces: StepCounter for tracking the number of steps taken and CalorieCalculator for estimating total calories burned based on total steps.

Jeevan needs your help creating a program.

Note

The calorie calculation formula is: Total caloriesBurned = (total steps / 100.0) * 20.0.

Input Format

The first line of input is an integer n, representing the number of days Jeevan wants to input data.

The second line consists of space-separated integers, representing the number of steps Jeevan took on each day.

Output Format

The first line of output prints: "Total Steps: <totalSteps>", where '<totalSteps>' is the sum of steps (integer) taken over 'n' days.

The second line prints: "Calories Burned: <caloriesBurned>", where '<caloriesBurned>' is the estimated total calories (double-point number) burned based on the total steps taken rounded off to two decimal places.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
340 234 987

Output: Total Steps: 1561
Calories Burned: 312.20

Answer

```
import java.util.Scanner;

interface StepCounter {
   void countSteps(int steps);
   int getTotalSteps();
}

interface CalorieCalculator {
   double calculateCaloriesBurned(int totalSteps);
}

class FitnessTracker implements StepCounter, CalorieCalculator {
   private int totalSteps = 0;

   public void countSteps(int steps) {
      totalSteps += steps;
   }

   public int getTotalSteps() {
      return totalSteps;
   }

   public double calculateCaloriesBurned(int totalSteps) {
      return (totalSteps / 100.0) * 20.0;
   }

   public void displayResults() {
      System.out.println("Total Steps: " + totalSteps);
      System.out.printf("Calories Burned: %.2f\n",
calculateCaloriesBurned(totalSteps));
   }
}

class Main
```

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

    FitnessTracker tracker = new FitnessTracker();

    int n = scanner.nextInt();

    for (int i = 0; i < n; i++) {
      int steps = scanner.nextInt();
      tracker.countSteps(steps);
    }

    int totalSteps = tracker.getTotalSteps();
    System.out.println("Total Steps: " + totalSteps);

    double caloriesBurned = tracker.calculateCaloriesBurned(totalSteps);
    System.out.printf("Calories Burned: %.2f%n", caloriesBurned);

    scanner.close();
  }
}
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Alex and Bob are designing a control system for household appliances, and one of the appliances is a washing machine. You want to create a program to help them that models the washing machine as a motor and calculates its electricity consumption based on its capacity.

Define an interface named Motor with the following methods:

void run() double consume(double capacity)

Create a class called WashingMachine that implements the Motor interface.

In the WashingMachine class:

Implement the run() method to print "Washing machine is running."Implement a consume() method to print "Washing machine is consuming electricity."Implement the consume(double capacity) method to calculate the electricity consumption (in kWh) of the washing machine based on its capacity. The formula for electricity consumption is (capacity * 0.05).

Input Format

The input consists of a double value representing the capacity of the washing machine in kW.

Output Format

The first line of output prints "Washing machine is running."

The second line prints "Washing machine is consuming electricity."

The third line prints "Electricity consumption: X kWh" where X is a double value, rounded off to two decimal places, representing the electricity consumption.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2.5
Output: Washing machine is running.
Washing machine is consuming electricity.
Electricity consumption: 0.13 kWh

Answer

```
import java.util.Scanner;

// You are using Java
interface Motor {
   void run();
   void consume();
   double consume(double capacity);
}

class WashingMachine implements Motor {
   public void run() {
       System.out.println("Washing machine is running.");
   }

   public void consume() {
       System.out.println("Washing machine is consuming electricity.");
   }

   public double consume(double capacity) {
       double consumption = capacity * 0.05;

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

       WashingMachine washingMachine = new WashingMachine();

       double capacity = scanner.nextDouble();

       washingMachine.run();
       washingMachine.consume();


       double consumption = washingMachine.consume(capacity);
       System.out.printf("Electricity consumption: %.2f kWh" ,consumption);

       scanner.close();
   }
}
```

Status : Correct ***Marks : 10/10***