Coding Exercise: 3.10- A little bit of Physics [1]

Chapter 3: Methods[2]

Jarrian Vince G. Gojar[3]

[1] A coding exercise for Chapter 3of the Study Guide on the course Object-Oriented Programming.
[2] This chapter covers the basics of methods in Java.
[3] https://github.com/godkingjay

Sorsogon State University - Bulan Campus

Coding Exercises

Instructions: Write a program that solves the following problems. Submit your code to the Google Drive folder provided by the instructor.

The following exercises are designed to help you practice using methods in Java. You can create separate methods for each exercise and call them from the "main" method to test your solutions.

Exercise 1: Calculate Speed

In this exercise, you will write a Java program to calculate the speed of a vehicle. The speed of a vehicle is calculated using the formula:

$$\text{Speed} = \frac{\text{Distance}}{\text{Time}} \tag{1}$$

$$\text{Distance} = \text{Final Distance} - \text{Initial Distance} \tag{2}$$

$$\text{Time} = \text{Final Time} - \text{Initial Time} \tag{3}$$

$$\text{Speed} = \frac{\text{Final Distance} - \text{Initial Distance}}{\text{Final Time} - \text{Initial Time}} \tag{4}$$

1. Create a method called "calculateSpeed" that accepts the initial distance, final distance, initial time, and final time as parameters and returns the speed of the vehicle.
2. Calculate the speed of a vehicle that travels from an initial distance of 0 meters to a final distance of 100 meters in 10 seconds.
3. Print the speed of the vehicle to the console.

   Example:
   A vehicle left from 0 meters and traveled to 100 meters in 10 seconds. The speed of the vehicle is 10 meters per second.

   $InitialDistance$: 0
   $FinalDistance$: 100
   $InitialTime$: 0
   $FinalTime$: 10

   $Speed = \frac{100-0}{10-0} = \frac{100}{10} = 10$
   $Speed = 10$ meters per second

   Output:
   The vehicle traveled from 0 meters to 100 meters in 0 seconds to 10 seconds at a speed of 10 meters per second.

```java
// CalculateSpeed.java
package com.oop.Exercises;

import java.util.Scanner;

public class CalculateSpeed {
```

```
7       public static void main(String[] args) {
8           Scanner scanner = new Scanner(System.in);
9
10          double finalDistance = 0, initialDistance = 0, finalTime = 0,
                initialTime = 0;
11
12          do {
13              if (finalDistance <= initialDistance) {
14                  System.out.println("\nFinal distance should be greater than
                        initial distance.");
15              }
16
17              System.out.print("Enter the initial distance: ");
18              initialDistance = scanner.nextDouble();
19              System.out.print("Enter the final distance: ");
20              finalDistance = scanner.nextDouble();
21          } while (finalDistance <= initialDistance);
22
23          do {
24              if (finalTime <= initialTime) {
25                  System.out.println("\ndFinal time should be greater than
                        initial time.");
26              }
27
28              System.out.print("Enter the initial time: ");
29              initialTime = scanner.nextDouble();
30              System.out.print("Enter the final time: ");
31              finalTime = scanner.nextDouble();
32          } while (finalTime <= initialTime);
33
34          double speed = calculateSpeed(initialDistance, finalDistance,
                initialTime, finalTime);
35
36          String output = String.format(
37                  "The vehicle traveled from %.2f meters to %.2f meters in %.2f
                        seconds to %.2f seconds at a speed of %.2f meters per
                        second.",
38                  initialDistance, finalDistance, initialTime, finalTime,
                        speed);
39
40          System.out.println("\n" + output);
41
42          scanner.close();
43      }
44
45      public static double calculateSpeed(double initialDistance, double
            finalDistance, double initialTime,
46              double finalTime) {
47          // Calculate the speed of the vehicle
48      }
49  }
```

Code 1: Initial Code for Exercise 1

Complete the code in Code 1 by implementing the "calculateSpeed" method to calculate the speed of the vehicle. Test the program by entering the initial distance, final distance, initial time, and final time from the user and printing the speed of the vehicle to the console.

Exercise 2: Arithmetic Sequence

In this exercise, you will write a Java program to calculate the sum of an arithmetic sequence using recursion. An arithmetic sequence is a sequence of numbers in which the difference between consecutive terms is constant. The sum of an arithmetic sequence is calculated using the formula:

$$\text{Sum} = \frac{n}{2} \times (2a + (n-1)d) \tag{5}$$

Where:

- n is the number of terms in the sequence.
- a is the first term in the sequence.
- d is the common difference between consecutive terms.
- $2a + (n-1)d$ is the sum of the first and last terms in the sequence.
- $\frac{n}{2} \times (2a + (n-1)d)$ is the sum of the arithmetic sequence.

1. Create a method called "calculateArithmeticSequence" that accepts the first term, common difference, number of terms, and current term as parameters and returns the sum of the arithmetic sequence.

2. Calculate the sum of an arithmetic sequence with the first term 1, common difference 2, and number of terms 5.

   Base Case: If the current term is equal to the number of terms, return the current term.
   Recursive Case: Calculate the sum of the arithmetic sequence using recursion.

3. Print the sum of the arithmetic sequence to the console.

   Example:
   If the first term is 1, the common difference is 2, and the number of terms is 5, then the sum of the arithmetic sequence is 25.

   $Sequence : 1, 3, 5, 7, 9$
   $Sum : 1 + 3 + 5 + 7 + 9 = 25$

```java
// ArithmeticSequence.java
package com.oop.Exercises;

import java.util.Scanner;

public class ArithmeticSequence {
```

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

        System.out.print("Enter the first term of the arithmetic sequence: ");
        double firstTerm = scanner.nextDouble();
        System.out.print("Enter the common difference of the arithmetic sequence: ");
        double commonDifference = scanner.nextDouble();
        System.out.print("Enter the number of terms in the arithmetic sequence: ");
        int numberOfTerms = scanner.nextInt();

        double sum = calculateArithmeticSequence(firstTerm, commonDifference, numberOfTerms, 1);

        String output = String.format(
                "The sum of the arithmetic sequence with the first term %.2f, common difference %.2f, and %d terms is %.2f.",
                firstTerm, commonDifference, numberOfTerms, sum);

        System.out.println("\n" + output);

        scanner.close();
    }

    public static double calculateArithmeticSequence(double firstTerm, double commonDifference, int numberOfTerms,
            int currentTerm) {
        // Calculate the sum of the arithmetic sequence using recursion
    }
}
```

Code 2: Initial Code for Exercise 2

Complete the code in Code 2 by implementing the "calculateArithmeticSequence" method to calculate the sum of the arithmetic sequence using recursion. Test the program by entering the first term, common difference, and number of terms from the user and printing the sum of the arithmetic sequence to the console.

Submission of Coding Exercises

Instructions:

1. Go to the Google Drive folder provided by the instructor:

   For BSCS 2-1:
   https://drive.google.com/drive/folders/1c56xFCJgFh6FWQQ4iZ-UuKKcWioF8pgs?usp=sharing

For BSCS 2-2:
https://drive.google.com/drive/folders/1jANc3o6atOYbHyoJZ6b-j-nDlTknEiu-?usp=sharing

2. Inside the folder, create another folder for your group with the following format:

   Group Number - LastName1_FirstName1, LastName2_FirstName2
   Example: **Group 1 - Doe_John, Smith_Jane**

3. Inside the sub-folder, create another folder with the name:

   Chapter 3- Coding Exercise 3.10- A little bit of Physics

4. Inside the folder, upload the file of your submission.

   Fill in the template provided in the following link and upload it inside the folder:
   https://docs.google.com/document/d/1sctvVLgpPSVnXN82k6LsOPSPApKp2rV0/edit?usp=drive_link&ouid=112709378145681657270&rtpof=true&sd=true

5. The activity must be submitted **on or before October 18, 2024**.

6. Late submissions will not be accepted.