

Introduction to Object-Oriented Programming Concepts

Name : Ahmad Hayyin B(01)

Class : SIB1G

NIM : 2341760199

1. Competence

After taking this experimental material, students are able to get to know and understand the concept of programming with a **structural paradigm (structured programming)**

2. Introduction

The fundamental differences between procedural programming and Object Oriented Programming (OOP) are:

- Structured programming: programs are broken down into sub-programs in the form of **functions**. The characteristics of the object (what it has **and** what it does) are represented in independent (unbound) variables and functions
- PBO: The program is broken down into **objects**, where the object wraps **attributes** and **methods**.

Here is an example of structural programming:

```
bycycle.java > bycycle > main(String[])
1  public class bycycle
2  {
3      Run | Debug
4      public static void main(String[] args)
5      {
6          String merk, merk2;
7          int speed, speed2, gear, gear2;
8          merk = "Polygon";
9          speed = 10;
10         gear = 1;
11
12         merk2 = "Wiim Cycle";
13         speed2 = 15;
14         gear2 = 3;
15
16         speed = increaseSpeed(speed, increment:10);
17         speed2 = reduceSpeed(speed2, decrement:5);
18
19
20         System.out.println("merk = " + merk);
21         System.out.println("speed = " + speed);
22
23         System.out.println("merk = " + merk2);
24         System.out.println("speed = " + speed2);
25
26     }
27
28     public static int increaseSpeed(int speed, int increment)
29     {
30         speed += increment;
31         return speed;
32     }
33     public static int reduceSpeed(int speed, int decrement)
34     {
35         speed -= decrement;
36         return speed;
37     }
38 }
```

Based on these examples, it can be seen that in the structural programming paradigm:

1. The characteristics/status/values of bicycle objects (things owned by bicycles) in the real world are represented or stored in the program as **independent or unrelated variables**.

The first bike, its characteristics are stored in the variables of merk, speed, and gear

The second bicycle has its characteristics stored in the merk2, speed2, and gear2

If there is a third bike, it will likely be stored in the merk3, speed3, and gear3

In effect, **there is no mechanism that guarantees** that the brand variables2, speed2, and gear2 are interconnected

2. The procedure/behavior/process of a bicycle (things that a bicycle can do) in the real world is represented as a function that can be called/executed, i.e. increaseSpeed() and reduceSpeed()

But this method **does not guarantee** that the two functions **can only be called by a bicycle-type object**, it is possible that a seat-type object can call this function.

3. Practicum Assignment

3.1. Task 1

Take the following steps so that the practicum assignments carried out are systematic:

- a. Define 1 category/class of objects. You can use a new object type or one of the objects from the PBO Theory task. For example: Bicycle
- b. Make observations of the object to determine
 - 3 variables/states/traits/states/values that can be had
 - 2 functions/behaviors/procedures/behaviors/processes that the object can perform
- c. Implement 10 objects of this type into a program with a **structural programming paradigm** (as in the bicycle example above)
 - Declare and initialize a variable for each characteristic/state/value of an object as a variable
 - Create a function of every procedure/behavior/process that can be performed by the object and then try to call the function/method

My answer :

```
J SoundSystem_01.java 9+ X J Kalkulator_01.java 1
C: > Users > HP > Downloads > J SoundSystem_01.java > SoundSystem_01 > main(String[])
1 public class SoundSystem_01 {
    Run | Debug
2     public static void main(String[] args) {
3         // Deklarasi dan inisialisasi variabel untuk 10 objek Sound System
4         String merk1 = "Sony";
5         int volume1 = 50;
6         int bassLevel1 = 70;
7
8         String merk2 = "JBL";
9         int volume2 = 60;
10        int bassLevel2 = 80;
11
12        String merk3 = "Bose";
13        int volume3 = 45;
14        int bassLevel3 = 75;
15
16        String merk4 = "Yamaha";
17        int volume4 = 55;
18        int bassLevel4 = 65;
19
20        String merk5 = "RDW";
21        int volume5 = 40;
22        int bassLevel5 = 60;
23
24        String merk6 = "Ashley";
25        int volume6 = 65;
26        int bassLevel6 = 85;
27
```

```
String merk8 = "Cannon";
int volume8 = 50;
int bassLevel8 = 70;

String merk9 = "RCF";
int volume9 = 55;
int bassLevel9 = 75;

String merk10 = "Lavoce";
int volume10 = 60;
int bassLevel10 = 80;

// Penanganan function untuk meningkatkan volume dan menurunkan bass
volume1 = increaseVolume(volume1, increment:10);
bassLevel1 = decreaseBassLevel(bassLevel1, decrement:5);

volume2 = increaseVolume(volume2, increment:15);
bassLevel2 = decreaseBassLevel(bassLevel2, decrement:10);

// Output perubahan nilai variabel
System.out.println("Sound System 1 - Merk: " + merk1 + ", Volume: " + volume1 + ", Bass Level: " + bassLevel1);
System.out.println("Sound System 2 - Merk: " + merk2 + ", Volume: " + volume2 + ", Bass Level: " + bassLevel2);
System.out.println("Sound System 2 - Merk: " + merk3 + ", Volume: " + volume3 + ", Bass Level: " + bassLevel3);
```

```
public static int increaseVolume(int volume, int increment) {
    volume += increment;
    return volume;
}

public static int decreaseBassLevel(int bassLevel, int decrement) {
    bassLevel -= decrement;
    return bassLevel;
}
```

```
ocal\Temp\vscodesws_ed951\jdt_ws\jdt.ls-java-project\bin' 'SoundSystem_01'
Sound System 1 - Merk: Sony, Volume: 60, Bass Level: 65
Sound System 2 - Merk: JBL, Volume: 75, Bass Level: 70
Sound System 2 - Merk: Bose, Volume: 75, Bass Level: 65
```

3.2. Task 2

Create a simple calculator program with a **structural programming paradigm** that can accept the input of number1, operator, and number2 and display the results to the console/screen

My answer :

```
SoundSystem_01.java 9+    J Kalkulator_01.java X
> Users > HP > Downloads > J Kalkulator_01.java > ...
1  import java.util.Scanner;
2
3  public class Kalkulator_01 {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          System.out.print("Masukkan angka pertama: ");
8          double angka1 = scanner.nextDouble();
9
10         System.out.print("Masukkan operator (+, -, *, /): ");
11         char operator = scanner.next().charAt(0);
12
13         System.out.print("Masukkan angka kedua: ");
14         double angka2 = scanner.nextDouble();
15
16         double hasil = hitung(angka1, operator, angka2);
17
18         // Menampilkan hasil/ output
19         System.out.println("Hasil: " + hasil);
20
21         scanner.close();
22     }
23
24     public static double hitung(double angka1, char operator, double angka2) {
25         double hasil = 0;
26
27         switch (operator) {
28             case '+':
29                 hasil = angka1 + angka2;
30                 break;
```

```
                case '-':
                    hasil = angka1 - angka2;
                    break;
                case '*':
                    hasil = angka1 * angka2;
                    break;
                case '/':
                    if (angka2 != 0) {
                        hasil = angka1 / angka2;
                    } else {
                        System.out.println("Error: Pembagian dengan nol tidak diperbolehkan.");
                    }
                    break;
                default:
                    System.out.println("Error: Operator tidak valid.");
                    break;
            }
        }
        return hasil;
    }
}
```

```
_ws\jdt.ls-java-project\bin' 'Kalkulator_01'
Masukkan angka pertama: 15
Masukkan operator (+, -, *, /): /
Masukkan angka kedua: 3
Hasil: 5.0
```

4. Question

Write down your analysis, is programming with a structured paradigm suitable for practicum assignments 1 and 2? Explain!

My answer :

1. Practical Assignment 1: Sound System

Analysis:

- **Advantages:** This code is simple and easy to follow for small number of objects and easy to implement.
- **Disadvantages:** As the number of sound systems increases, managing them with individual variables becomes impractical. Adding new sound systems or making major changes will require repeated code.

2. Practical Assignment 2: Simple Calculator

Analysis:

- **Advantages:** Structural programming is suitable for Calculator practical assignments because it is simple and easy to implement.
- **Disadvantages:** If a calculator needs to handle more complex features, such as additional operations, scientific calculations, or storing a calculation history, maintaining it with a structural approach can become difficult.

Conclusion:

Structural programming is suitable for simple problems like calculators, but OOP is better for more complex problems like managing many objects..

3. Direction

1. Work on the Java program code via Netbean or Visual studio Code.
2. Go to the lecturer's desk for an oral test of the results of the practical work assignment (about 5 minutes) after finishing you can go home first or continue working on the report.
3. Report in PDF according to the practical steps and screenshots of the program code, (the original program code for each practical should be kept safe by yourself if the quiz or exam time is asked)
4. Collect the report in LMS in PDF format with a deadline of 23.59 WIB on the same day as the practical, outside of that hour it will not be corrected.

--- Good Luck----