MODUL 9

OBJECT & DRIVER CLASS

Adam Arthur Faizal

M3119001

TI A

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

SURAKARTA

2020

MODUL 9. OBJECT & DRIVER CLASS

Capaian Pembelajaran Praktikum:

- Create an Object class
- Create a Driver clasS

Tools:

- Java Development Kit (JDK)
- Eclipse

Terminologi:

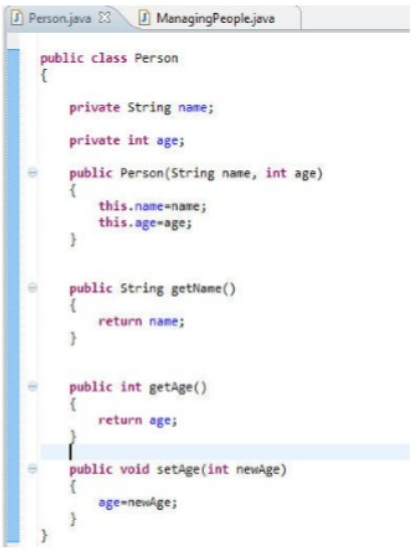
Isikan terminology yang sesuai untuk definisi dibawah ini:

[Packages] A group of related Java classes.
[Code Block] Sections of code that are enclosed inside a set of curly braces. {}
[Upper camel case] First letter uppercase and the first letter of each internal word capitalized. Example: SavingsAccount
[Constant] A named value that does not change.
[Lower camel case] First letter lowercase and the first letter of each internal word capitalized. Example: studentFirstName
[Driver class] A class that contains a main method.
[Import statement] A code statement in a Java class file that includes java code from another package or class.
[Programmer-created object class] A class that defines instances of objects to be used in another class.
[Java comments] Code that is preceded by //. Comments are used to clarify programming logic. Comments are ignored by the compiler.
[Java Keywords] A word that has a special function in the Java language, and cannot be used as names for classes, methods, or variables.
[Java API] The library of Java classes available to import into a programmercreated class.
[Object class] The outline of an object, including class variables, constructors, and methods.
[Constructor] A special kind of method that is a template for an object.
[Arguments] Values that are sent into a method or constructor to be used in a calculation or substituted with values from the class.
[Variables] Values, such as numbers, characters, or booleans. References to objects, such as a BankAccount object.
[Acces modifiers] Keywords used to specify the accessibility of a class (or type) and its members. Ex: public, private, protected, default
[Method] A block of code inside a class that is used to change or access information about the class.

TRY IT / SOLVE IT:

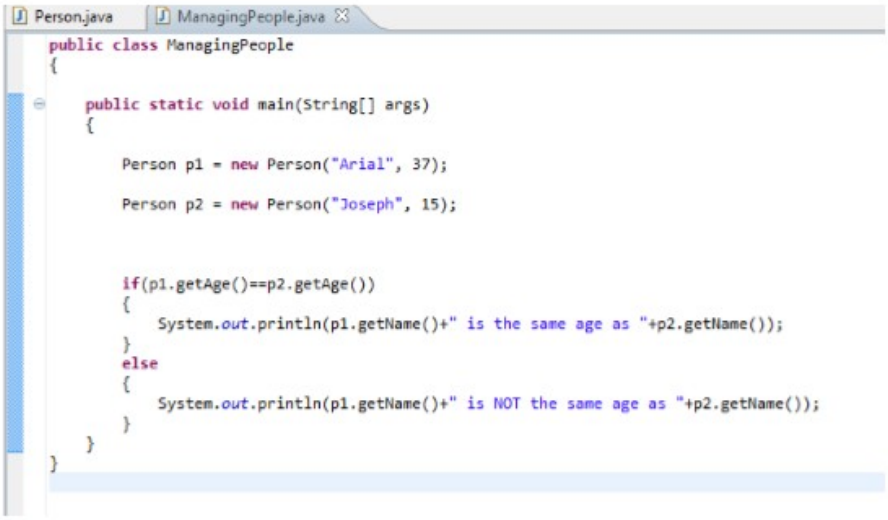
1. Sebutkan komponen secara urut dalam membuat sebuah class di java.
    a. Access Modifiers
    b. Class Declaration
    c. Field/Variables
    d. Constructor
    e. Methods

2. Buatlah proyek baru, namai dengan Universitas. Kemudian buat package dengan nama person. Setelah itu buatlah object class Person.java (didalam package person) seperti gambar di bawah ini. Berikan penjelasan untuk source code yang anda tuliskan.
    a. Kita membuat Object class bernama Person. Lalu terdapat beberapa class variables didalamnya, diantaranya ada String name dan int age. Lalu di dalam class tersebut juga terdapat constructor berparameter, dimana saat kelas lain menginstansiasi object dari kelas ini, maka statement yang ada di dalam constructor tersebut akan dijalankan. Selain constructor, juga terdapat method getter and setter yang berfungsi untuk mengambil dan mengubah value dari class variables tersebut jika sedang diakses dari kelas yang lain.




```java
package com.latihan2;

/**
 * @author adam
 *
 */
public class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    /**
     * @return the name
     */
    public String getName() {
        return this.name;
    }

    /**
     * @param name the name to set
     */
    public void setName(String name) {
        this.name = name;
    }

    /**
     * @return the age
     */
    public int getAge() {
        return this.age;
    }

    /**
     * @param age the age to set
     */
    public void setAge(int age) {
        this.age = age;
    }
}
```

3. Buat driver kelas, namai dengan ManagingPeople.java seperti gambar di bawah ini. Kemudian compile dan jalankan. Berikan penjelasan untuk source code yang anda tuliskan.
    a. Kita menginstansiasi 2 object baru di kelas ManagingPeople sekaligus memberikan argument pada Constructor nya karena sebelumnya kita membuat Constructor berparameter pada kelas Person. Kali ini kita membuat nilai argument nya berbeda, setelah itu kita buat if else yang mengecek age dari tiap2 object, jika age nya sama, maka akan mengeluarkan output sedemikian rupa, sebaliknya jika age nya tidak sama, maka juga akan mengeluarkan output yang berbeda.




```
package com.latihan2;

/**
 * @author adam
 *
 */
public class ManagingPeople {

    public static void main(String[] args) {
        Person p1 = new Person("Arial", 37);
        Person p2 = new Person("Joseph", 15);

        if (p1.getAge() == p2.getAge()) {
            System.out.printf("%s is the same age as %s \n", p1.getName(), p2.getName());
        } else {
            System.out.printf("%s is not the same age as %s \n", p1.getName(), p2.getName());
        }
    }
}
```

4. Sebutkan setidaknya 20 keyword di java.
    a. **abstract:** Java abstract keyword is used to declare abstract class. Abstract class can provide the implementation of interface. It can have abstract and non-abstract methods.
    b. **boolean:** Java boolean keyword is used to declare a variable as a boolean type. It can hold True and False values only.

c. **break:** Java break keyword is used to break loop or switch statement. It breaks the current flow of the program at specified condition.

d. **byte:** Java byte keyword is used to declare a variable that can hold an 8-bit data values.

e. **case:** Java case keyword is used to with the switch statements to mark blocks of text.

f. **catch:** Java catch keyword is used to catch the exceptions generated by try statements. It must be used after the try block only.

g. **char:** Java char keyword is used to declare a variable that can hold unsigned 16-bit Unicode characters

h. **class:** Java class keyword is used to declare a class.

i. **continue:** Java continue keyword is used to continue the loop. It continues the current flow of the program and skips the remaining code at the specified condition.

j. **default:** Java default keyword is used to specify the default block of code in a switch statement.

k. **do:** Java do keyword is used in control statement to declare a loop. It can iterate a part of the program several times.

l. **double:** Java double keyword is used to declare a variable that can hold a 64-bit floating-point numbers.

m. **else:** Java else keyword is used to indicate the alternative branches in an if statement.

n. **enum:** Java enum keyword is used to define a fixed set of constants. Enum constructors are always private or default.

o. **extends:** Java extends keyword is used to indicate that a class is derived from another class or interface.

p. **final:** Java final keyword is used to indicate that a variable holds a constant value. It is applied with a variable. It is used to restrict the user.

q. **finally:** Java finally keyword indicates a block of code in a try-catch structure. This block is always executed whether exception is handled or not.

r. **float:** Java float keyword is used to declare a variable that can hold a 32-bit floating-point number.

s. **for:** Java for keyword is used to start a for loop. It is used to execute a set of instructions/functions repeatedly when some conditions become true. If the number of iteration is fixed, it is recommended to use for loop.

t. **if:** Java if keyword tests the condition. It executes the if block if condition is true.

LATIHAN:

5. Buatlah sebuah object class Student yang memiliki variabel studentID, name, NIM, IPK. Kemudian buatlah method setter dan getter nya. (anda bisa melihat modul/slide)

```java
/**
 * @author adam
 *
 */
public class Student {
    private int studentId;
    private String nama;
    private String nim;
    private int umur;
    private double ipk;

    public Student(int studentId, String nama, String nim, int umur, double ipk) {
        this.studentId = studentId;
        this.nama = nama;
        this.nim = nim;
        this.umur = umur;
        this.ipk = ipk;
    }

    /**
     * @return the studentId
     */
    public int getStudentId() {
        return this.studentId;
    }

    /**
     * @param studentId the studentId to set
     */
    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }

    /**
     * @return the nama
     */
    public String getNama() {
        return this.nama;
    }

    /**
     * @param nama the nama to set
     */
    public void setNama(String nama) {
        this.nama = nama;
    }

    /**
     * @return the nim
     */
    public String getNim() {
        return this.nim;
    }

    /**
     * @param nim the nim to set
     */
    public void setNim(String nim) {
        this.nim = nim;
    }

    /**
     * @return the umur
     */
    public int getUmur() {
        return this.umur;
    }

    /**
     * @param umur the umur to set
     */
    public void setUmur(int umur) {
        this.umur = umur;
    }

    /**
     * @return the ipk
     */
    public double getIpk() {
        return this.ipk;
    }

    /**
     * @param ipk the ipk to set
     */
    public void setIpk(double ipk) {
        this.ipk = ipk;
    }
}
```

6. Buatlah driver class untuk menguji object class yang anda buat al, membuat objek, memanggil method dan mencetak data objek Student. (anda bisa melihat modul/slide)

```java
package com.latihan2;

/**
 * @author adam
 *
 */
public class UjiStudents {

    public static void main(String[] args) {
        // studentId, nama, nim, umur, ipk
        Student student1 = new Student(1, "Adam Arthur Faizal", "M3119001", 17, 3.67);
        Student student2 = new Student(2, "Mbah Putih Mulyosugito", "M3119000", 107, -3.67);

        // Student 1
        System.out.printf("Halo, namaku %s \n", student1.getNama());
        System.out.printf("Umurku baru %d \n", student1.getUmur());
        System.out.printf("Kemarin aku dapet IPK %f \n\n", student1.getIpk());

        // Student 2
        System.out.printf("Halo, namaku %s \n", student2.getNama());
        System.out.printf("Umurku baru %d \n", student2.getUmur());
        System.out.printf("Kemarin aku dapet IPK %f \n\n", student2.getIpk());

    }
}
```

7. Jalankan program yang anda buat! Tuliskan hasilnya!

```
Halo, namaku Adam Arthur Faizal
Umurku baru 17
Kemarin aku dapet IPK 3.670000

Halo, namaku Mbah Putih Mulyosugito
Umurku baru 107
Kemarin aku dapet IPK -3.670000
```

8. Catat hasil yang anda peroleh dari step nomor 5 & 6, kemudian berikan penjelasan bilamana perlu.
    a. Saya membuat kelas Student, yang didalamnya terdapat beberapa class variables, yaitu studentId, nama, nim, umur, ipk. Lalu juga terdapat constructor berparameter untuk memberikan nilai pada masing-masing class variables yang sudah di deklarasikan sebelumnya. Tak lupa method getter and setter untuk masing-masing class variables agar mereka semua dapat diambil dan dimodifikasi valuenya dari kelas lain.

Setelah sesi praktikum SELESAI, laporan praktikum harus dikirim/diupload ke google classroom pada hari yang berikutnya.