



MODUL PRAKTIKUM

Bekerja Dengan JAVA Class Library

Versi 1.4

Modul Praktikum Bekerja dengan Java Class Library

1. Tujuan

- menjelaskan OOP dan beberapa konsepnya
- perbedaan antara class dan object
- perbedaan antara instance variables/method dan class (static) variable/method
- menjelaskan method apa dan bagaimana memanggil method parameter
- mengidentifikasi beberapa jangkauan dari sebuah variable
- memilih tipe data primitive dan object
- membandingkan objects dan menjabarkan class dari objects.

2. Latar Belakang

Pada sesi ini, kita akan mengantarkan beberapa konsep dasar dari Object-Oriented objects, dan Programming (OOP). Selanjutnya kita akan membahas konsep dari classes dan bagaimana menggunakan class dan anggotanya. Perubahan dan pemilihan object juga akan dibahas. Sekarang, kita akan focus dalam menggunakan class yang telah dijabarkan dalam Java Class library, kita akan membahas nanti tentang bagaimana membikin class anda sendiri.

OOP berputar pada konsep dari object sebagai dasar element dari program anda. Ketika kita membandingkan dengan dunia nyata, kita dapat menemukan beberapa objek disekitar kita, seperti mobil, singa, manusia dan seterusnya. Object ini dikarakterisasi oleh sifat / atributnya dan tingkah lakunya.

Contohnya, objek sebuah mobil mempunyai sifat tipe transmisi, warna dan manufaktur. Mempunyai kelakuan berbelok, mengerem dan berakselerasi. Dengan cara yang sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari singa. Coba perhatikan table dibawah ini sebagai contoh perbandingan :

<i>Object</i>	<i>Properties</i>	<i>Behavior</i>
Car	type of transmission	turning
	manufacturer	braking
	color	accelerating
Lion	Weight	roaring

Modul Praktikum Bekerja dengan Java Class Library

<i>Object</i>	<i>Properties</i>	<i>Behavior</i>
	Color hungry or not hungry tamed or wild	sleeping hunting

3. Percobaan

Percobaan 1 Pass By Value:

```
public class TestPastByValue {  
    public static void main(String[] args) {  
        int i = 10;  
        System.out.println(i);  
        test(i);  
        System.out.println(i);  
    }  
    public static void test(int j){  
        j = 33;  
    }  
}
```

Percobaan 2 Pass by reference :

Modul Praktikum Bekerja dengan Java Class Library

```
public class TestPassByReference {
    public static void main(String[] args) {
        int []ages = {10,11,12};
        for(int i=0;i<ages.length;i++){
            System.out.println(ages[i]);
        }
        test(ages);
        for(int i=0;i<ages.length;i++){
            System.out.println(ages[i]);
        }
    }
    public static void test(int []arr){
        for(int i=0;i<arr.length;i++){
            arr[i]=i+50;
        }
    }
}
```

Percobaan 3 Perbandingan Object :

```
class EqualsTest {
    public static void main(String[] arguments) {
        String str1, str2;
        str1 = "Free the bound periodicals.";
        str2 = str1;
        System.out.println("String1: " + str1);
        System.out.println("String2: " + str2);
        System.out.println("Same object? " + (str1 == str2));
        str2 = new String(str1);
        System.out.println("String1: " + str1);
        System.out.println("String2: " + str2);
        System.out.println("Same object? " + (str1 == str2));
        System.out.println("Same value? " + str1.equals(str2));
    }
}
```

Modul Praktikum Bekerja dengan Java Class Library

Percobaan 4 Pemanggilan method menggunakan parameter:

```
public class Elevator {
    public boolean doorOpen=false;
    public int currentFloor = 1;
    public final int TOP_FLOOR = 5;
    public final int BOTTOM_FLOOR = 1;
    public void openDoor() {
        System.out.println("Opening door.");
        doorOpen = true;
        System.out.println("Door is open.");
    }
    public void closeDoor() {
        System.out.println("Closing door.");
        doorOpen = false;
        System.out.println("Door is closed.");
    }
    public void goUp() {
        System.out.println("Going up one floor.");
        currentFloor++;
        System.out.println("Floor: " + currentFloor);
    }
    public void goDown() {
        System.out.println("Going down one floor.");
        currentFloor--;
        System.out.println("Floor: " + currentFloor);
    }
    public void setFloor(int desiredFloor) {
        while (currentFloor != desiredFloor){
            if(currentFloor < desiredFloor) {
                goUp();}
            else {
                goDown();}
        }
    }
    public int getFloor(){
        return currentFloor;
    }
    public boolean checkDoorStatus(){
        return doorOpen;
    }
}
```

Modul Praktikum Bekerja dengan Java Class Library

```
public class ElevatorTest {
    public static void main(String[] args) {
        Elevator myElevator = new Elevator();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goUp();
        myElevator.goUp();
        myElevator.goUp();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goDown();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goDown();
        myElevator.setFloor(myElevator.TOP_FLOOR);
        myElevator.openDoor();
    }
}
```

Percobaan 5 Method dengan Return Value :

```
public class ElevatorTestTwo {
    public static void main(String[] args) {
        Elevator myElevator = new Elevator();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goUp();
        myElevator.goUp();
        myElevator.goUp();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goDown();
        myElevator.openDoor();
        myElevator.closeDoor();
        myElevator.goDown();
        int curFloor = myElevator.getFloor();
        System.out.println("Current Floor: " + curFloor);
        myElevator.setFloor(curFloor+1);
        myElevator.openDoor();
    }
}
```

Modul Praktikum Bekerja dengan Java Class Library

4. Latihan

4.1 Mendefinisikan Istilah

Dengan kata-kata Anda sendiri, definisikan istilah-istilah berikut ini :

1. Class
2. Object
3. Instantiate
4. Instance Variable
5. Instance Method
6. Class Variables atau static member variables
7. Constructor

4.2 Java Scavenger Hunt

Pipoy adalah suatu anggota baru dalam bahasa pemrograman Java. Dia hanya memperdengarkan bahwa telah ada APIs siap pakai dalam Java yang salah satunya dapat digunakan dalam program mereka, dan ia ingin sekali untuk mengusahakan mereka keluar. Masalahnya adalah, Pipoy tidak memiliki copy dari dokumentasi Java, dan dia juga tidak memiliki akses internet, jadi tidak ada jalan untuknya untuk menunjukkan Java APIs.

Tugas Anda adalah untuk membantu Pipoy memperhatikan APIs (Application Programming Interface). Anda harus menyebutkan class dimana seharusnya method berada, deklarasi method dan penggunaan contoh yang dinyatakan method.

Sebagai contoh, jika Pipoy ingin untuk mengetahui method yang mengkonversisebuah String ke integer, jawaban Anda seharusnya menjadi:

Class: Integer

Method Declaration: public static int parseInt(String value)

Sample Usage:

```
String      strValue = "100";

int         value = Integer.parseInt( strValue );
```

yakinkan bahwa snippet dari kode yang Anda tulis dalam contoh Anda menggunakan compiles dan memberi output jawaban yang benar, jadi tidak membingungkan Pipoy. **(Hint: Semua methods adalah dalam java.lang package).** Dalam kasus dimana Anda dapat menemukan lebih banyak methods yang dapat menyelesaikan tugas, berikan hanya satu.

Modul Praktikum Bekerja dengan Java Class Library

Sekarang mari memulai pencarian!

1. Perhatikan sebuah method yang diuji jika String pasti diakhiri suffix yang pasti. Sebagai contoh, jika diberikan string "Hello", Method harus mengembalikan nilai true suffix yang diberikan adalah "lo", dan false jika suffix yang diberikan adalah "alp".
2. Perhatikan untuk method yang mengenali character yang mewakili sebuah digit yang spesifik dalam radix khusus. Sebagai contoh, jika input digit adalah 15, dan the radix adalah 16, method akan mengembalikan Character F, sejak F adalah representasi hexadecimal untuk angka 15 (berbasis 10).
3. Perhatikan untuk method yang mengakhiri running Java Virtual Machine yang sedang berjalan
4. Perhatikan untuk method yang memperoleh lantai dari sebuah nilai double. Sebagai contoh, jika Saya input a 3.13, method harus mengembalikan nilai 3.
5. Perhatikan untuk method yang mengenali jika character yang dipakai adalah sebuah digit. Sebagai contoh, jika Saya input '3', dia akan mengembalikan nilai true.