**PEMOGRAMAN BERORIENTASI OBJEK I**



Intan Pratiwi 140810160022

Syifa Fauziyah NI 140810160026

M. Fahmi Irfananda 140810160028

Ibnu Ahsani 140810160054

Kelas B

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJAJARAN

2017

# SOAL A

1. Apakah yang dimaksud dengan overloading?

Overloading adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda. Contoh penggunaan overloading dilihat dibawah ini:

Gambar(Titik t1) 1 parameter titik, untuk menggambar titik

Gambar(Titik t1, Titik t2) 2 parameter titik, untuk menggambar garis

Gambar(Titik t1,Titik t2,Titik t3)3 parameter titik, untuk menggambar segitiga

Gambar(Titik t1,Titik t2,Titik t3,Titik t4)4 parameter titik, untuk menggambar persegi empat

Overloading ini dapat terjadi pada class yang sama atau pada suatu parent class dan subclass-nya. Overloading mempunyai ciri-ciri sebagai berikut:

1. Nama method harus sama

2. Daftar parameter harus berbeda

3. Return type boleh sama, juga boleh berbeda

1. Apakah yang dimaksud dengan overriding

Overriding adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. Overriding mempunyai ciri-ciri sebagai berikut :

1. Nama method harus sama

2. Daftar parameter harus sama

3. Return type harus sama

Berikut ini contoh terjadinya overriding dimana method Info() pada class Child meng-override method Info() pada class parent:

class Parent {

public void Info() {

System.out.println(“Ini class Parent”);

}

}

class Child extends Parent {

public void Info() {

System.out.println(“Ini class Child”);

}

}

Method yang terkena override (overridden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (overriding method).

1. Jelaskan aturan tentang overridden

Method yang terkena override (overridden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (overriding method).

1. Apakah yang dimaksud dengan polimorfisme?

Polymorphism (polimorfisme) adalah kemampuan untuk mempunyai beberapa bentuk class yang berbeda. Polimorfisme ini terjadi pada saat suatu obyek bertipe parent class, akan tetapi pemanggilan constructornya melalui subclass. Misalnya deklarasi pernyataan berikut ini:

Employee employee=new Manager();

dimana Manager() adalah kontruktor pada class Manager yang merupakan subclass dari class Employee.

1. Jelaskan proses terjadinya Virtual Method Invocation

Virtual Method Invocation (VMI) bisa terjadi jika terjadi polimorfisme dan overriding. Pada saat obyek yang sudah dibuat tersebut memanggil overridden method pada parent class, kompiler Java akan melakukan invocation (pemanggilan) terhadap overriding method pada subclass, dimana yang seharusnya dipanggil adalah overridden method.

Berikut contoh terjadinya VMI:

class Parent { int x = 5;

public void Info() {

System.out.println("Ini class Parent");

}

}

class Child extends Parent {

int x = 10; public void Info() {

System.out.println("Ini class Child");

}

}

public class Tes {

public static void main(String args[]) {

Parent tes=new Child();

System.out.println("Nilai x = " + tes.x);

tes.Info();

}

}

Hasil dari running program diatas adalah sebagai berikut:



1. Apakah yang dimaksud dengan polymorphic arguments?

Polymorphic arguments adalah tipe suatu parameter yang menerima suatu nilai yang bertipe subclass-nya. Berikut contoh dari polymorphics arguments:

class Pegawai {

…

}

class Manajer extends Pegawai {

…

}

public class Tes {

public static void Proses(Pegawai peg) {

…

}

public static void main(String args[]) {

Manajer man = new Manajer();

Proses(man); }

}

1. Apakah kegunaan kata kunci instanceof?

Pernyataan instanceof sangat berguna untuk mengetahui tipe asal dari suatu polymorphic arguments.

1. Apakah perbedaan Abstract dan Interface?

**Class Abstract**

Metode abstrak adalah metode hanya dengan tanda (i.e., nama method, daftar argumen dan jenis return) tanpa implementasi (i.e., method body). Anda menggunakan kata kunci abstract untuk menyatakan method abstrak.

Catatan:

* Metode abstrak tidak dapat dideklarasikan final, sebagai metode akhir tidak dapat di override. Metode abstrak, di sisi lain, harus di-override dalam keturunan sebelum dapat digunakan.
* Metode abstrak tidak dapat private (akan menghasilkan kesalahan saat kompilasi). Hal ini karena metode private tidak terlihat terhadap subclass dan dengan demikian tidak dapat diganti.

**Class Interface**

Sebuah Interface adalah kontrak (atau protokol, atau pemahaman umum) dari apa yang kelas dapat lakukan. Ketika kelas mengimplementasikan interface tertentu, berarti kelas menyediakan implementasi untuk semua metode abstrak yang dideklarasikan dalam interface. Interface mendefinisikan satu set perilaku umum. Kelas mengimplementasikan interface setuju untuk perilaku dan menyediakan implementasi mereka sendiri untuk perilaku. Java tidak mendukung multiple inheritance (layaknya C ++)

**Abstract vs Interface**

* Desain yang lebih baik: Interface atau Abstract? Tidak ada jawaban yang jelas.
* Gunakan superclass abstrak jika ada hirarki kelas yang jelas. Abstract dapat berisi implementasi parsial (seperti contoh variabel dan metode). Interface tidak dapat berisi implementasi apapun, tetapi hanya mendefinisikan perilaku.
* Sebagai contoh, Thread pada Java dapat dibangun dengan menggunakan interface Runnable atau superclass Thread.

|  |  |  |
| --- | --- | --- |
| Komponen | Interface | Absctract |
| Attribute | Hanya berupa konstanta | Bebas memiliki tipe data apa saja |
| Method | Hanya berupa deklarasi | Boleh berupa deklarasi, boleh berupa method lengkap |
| Syntax | Seluruhnya absctract (berupa deklarasi) | Sebagian abstract |

# SOAL B

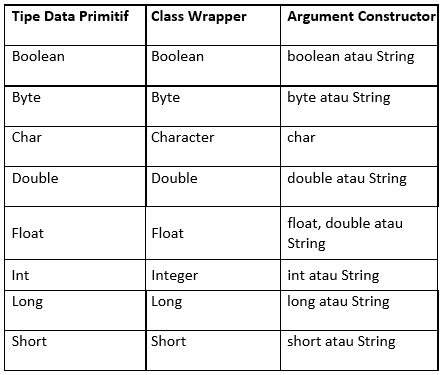
## Apakah yang anda ketahui tentang Inner Class

Inner Class adalah kelas yang disisipkan di dalam kelas yang lain. Fungsi kelas sisipan ini adalah mendukung suatu proses yang akan dijalankan oleh kelas utamanya. Inner Class bersifat tersarang terhadap kelas – kelas utamanya, seperti halnya blok penyeleksian (if, for) yang tersarang pada blok penyeleksian lainnya atau method yang tersarang pada method lainnya. Analogi Inner Class.

Inner Class dapat dianalogikan sebagi hubungan antara manusia dan paru – paru. Setiap manusia pasti bernafas dengan menggunakan paru – paru. Dalam hal ini berarti kinerja dari paru – paru turut mendukung/menetukan kinerja dari manusia. Dalam bahasa pemrograman berorientasi objek manusia dapat dikatakan sebagai outer class dan paru – paru sebagai inner class-nya

## Apakah yang anda ketahui tentang Wrapper

Class Wraper adalah tipe data class untuk tipe data primitive. Class ini membungkus tipe data primitive agar dapat digunakan sebagai objek.



## Apakah yang anda ketahui tentang Enum

Enum adalah tipe data yang constant