IPB University
Bogor Indonesia

KOM120C -- PEMROGRAMAN

Object Oriented Programming

- Polymorphisme pada Java

Tim Pengajar Pemrograman IPB University

Polymorphisme

Ada banyak jenis *polymorphism*(**poly** = banyak, **morph** = bentuk) dalam konteks pemrograman. Secara umum merujuk pada fenomena/kemampuan sepotong kode (variabel, fungsi, object dll) untuk mempunyai beberapa bentuk/fungsi yang berbeda, tergantung konteksnya.

Contoh bentuk *polymorphism* pada OOP: method & operator overloading pada C++

Disini, kita akan secara spesifik merujuk pada *polymorphism* yang terkait dengan method pada pewarisan obyek (*object inheritance*) dan *interface implementation* pada Java

Polymorphisme

Polymorphism dalam konteks ini berarti adanya kemampuan **obyek** untuk menjalankan sebuah methode sesuai dengan konteksnya: pemanggilan fungsi secara polimorfik. Perhatikan objek *obj* di bawah ini. Kedua pemanggilan metode f(); di bawah secara polimorfik akan memiliki bentuk yang berbeda

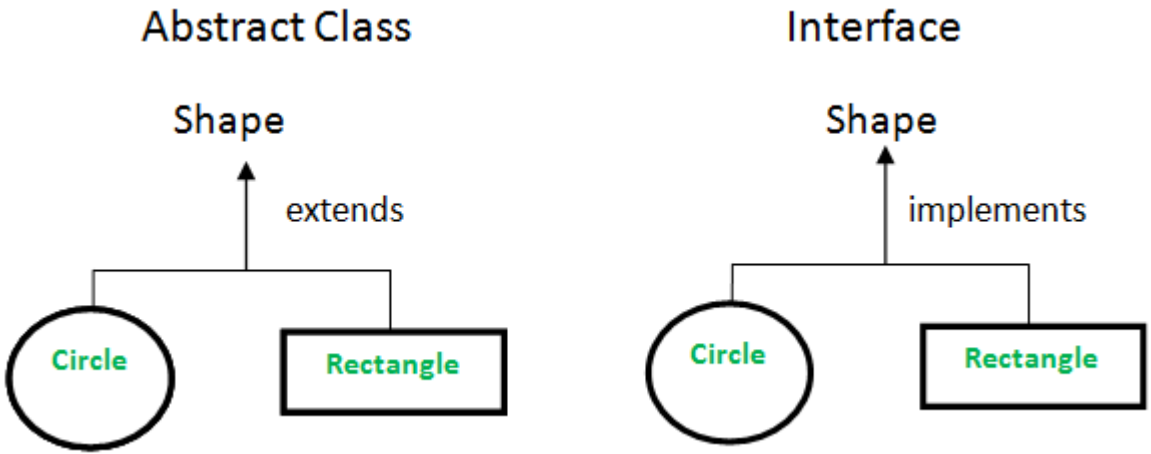
```
X obj;
obj = new Y();
obj.f()
…

obj = new Z();
obj.f()
```

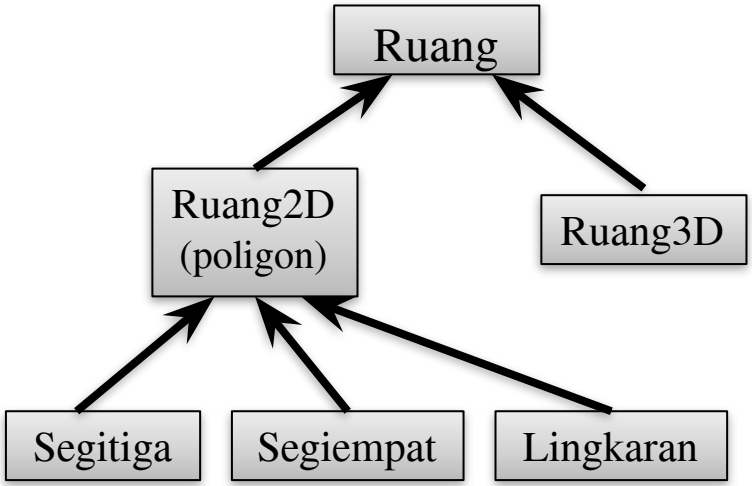
Polymorphisme

Polymorphisme pada Java memiliki dua cara yang berbeda:

- Menggunakan kelas dan metode abstrak
- Menggunakan interface

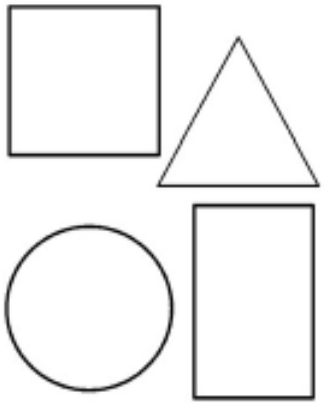


Review Abstract Class



Class Ruang mempunyai method area() yang akan di-override oleh subclasses nya.

Ruang merupakan sifat umum dari suatu bidang dua dimensi (Segitiga, Segiempat, Lingkaran) dan tiga dimensi (Bola, Kubus).



Method dalam class Ruang tidak memiliki implementasi. Class jenis ini yang disebut dengan abstract class.

Review Abstract Method

Method dalam abstract class yang tidak mempunyai implementasi dinamakan abstract method.

Only method signature, no body.

Pada C++, ditunjukkan dengan adanya method **virtual murni: virtual double area() = 0;**

Pada Java, dengan adanya deklarasi method **abstract,** Contoh:

```
public abstract class Shape {
    public abstract double area();
}
```

Polymorphism dengan Abstract Class

Sebuah **abstract class X** memiliki sebuah metode abstract **f()**. X kemudian diwariskan pada dua kelas: **Y** dan **Z**.

Jika kedua kelas turunan, **Y** dan **Z** meng-override metode **f**(), maka kita dapat mengamati adanya polimorfisme dengan membuat objek bertipe **X** namun dengan **diisi** dengan objek **Y** atau **Z** dan kemudian memanggil fungsi **f().**

Contoh Polimorfisme dengan Abstract Class

```
abstract class Shape
{
    public abstract double area();
}

class Persegi extends Shape {
    double panjang;
    double lebar;
    public Persegi(double p, double l) {
        panjang = p;
        lebar = l;
    }
    public double area (){
        return panjang * lebar;
}}
```

Contoh Polimorfisme dengan Abstract Class

```
class Lingkaran extends Shape {
    double jari;
    public static final double PI = 3.1425;
public Lingkaran(double r)    {
        jari = r;
    }
    public double area()    {
        return PI * jari * jari;
    }
}
```

Contoh Polimorfisme dengan Abstract Class

```
public class Main{
    public static void main(String[] args) {
    Shape s;
    s = new Persegi(3, 4);          System.out.
printf("Luas persegi = %.2f\n", s.area());
    …
    s = new Lingkaran(10);     System.out.printf("Luas
lingkaran = %.2f\n", s.area());
    }
}
Output:
Luas persegi = 12.00
Luas lingkaran = 314.25
```

Review Interface

Interface dalam Java didefinisikan sebagai tipe abstrak untuk menentukan perilaku dari class.

Interfaces dapat memiliki abstract methods dan variables.

Interface tidak dapat dibuat obyek namun hanya dapat diimplementasi.

Sintaks:

```
interface <nameOfInterface> {
    // declare constant fields
    // declare methods that abstract
    // by default.
}
```

Polymorphism dengan Interface

Dua buah kelas **A** dan **B** sama-sama mengimplementasikan sebuah *interface* **X**, yang memiliki sebuah metode **f()**.

Jika kita buat sebuah variabel bertipe *interface* **X** lalu mengisinya dengan **A** atau **B**, kemudian memanggil metode **f()** pada variabel tersebut, maka kita akan mengamati fenomena polimorfisme pada objek tersebut.

Contoh Polimorfisme dengan Interface

```
interface Shape
{
    public double area();
}

class Persegi implements Shape {
    double panjang;
    double lebar;
    public Persegi(double p, double l) {
        panjang = p;
        lebar = l;
    }
    public double area (){
        return panjang * lebar;
}}
```

Contoh Polimorfisme dengan Interface

```
class Lingkaran implements Shape {
    double jari;
    public static final double PI = 3.1425;
public Lingkaran(double r)    {
        jari = r;
    }
    public double area()    {
        return PI * jari * jari;
    }
}
```

Contoh Polimorfisme dengan Interface

```
public class Main{
    public static void main(String[] args) {
    Shape s;
    s = new Persegi(3, 4);          System.out.
printf("Luas persegi = %.2f\n", s.area());
    …
    s = new Lingkaran(10);     System.out.printf("Luas
lingkaran = %.2f\n", s.area());
    }
}
Output:
Luas persegi = 12.00
Luas lingkaran = 314.25
```

Polymorphisme pada kumpulan data

Salah satu pola umum yang biasa digunakan dalam penggunaan polymorphism adalah menyimpan sekumpulan objek yang berbeda-beda, namun merupakan turunan dari kelas abstract yang sama, atau yang mengimplementasikan interface yang sama dalam sebuah **collection**, lalu mengisinya dengan objek-objek dari kelas yang berbeda tersebut.

```
ArrayList<Shape> list = new ArrayList<Shape>();
list.add(new Lingkaran(10))
list.add(new Persegi(5))
…
for(int i=0; i<n; i++) sum += list[i].area();
```

Latihan Kelas

Buat program Java dengan menggunakan polymorphism dengan abstract class atau dengan interface untuk menyelesaikan permasalahan berikut:

Diberikan masukan berupa nilai *n*, diikuti *n* buah baris, masing-masing berisi dua kemungkinan:

- baris diawali angka 1 dan kemudian terdapat dua buah bilangan desimal **a** dan **b**
- baris diawali dengan angka 2, kemudian terdapat 4 buah bilangan bulat, **p q r s** yang melambangkan dua buah pecahan **a = p/q** dan **b = r/s**

Keluarkan n buah baris yang berisi pecahan hasil penjumlahan **a + b**. Jika masukan pada sebuah baris adalah pecahan desimal, maka keluarkan jumlah dalam bentuk desimal pula (dengan 2 digit di belakang koma). Jika masukan adalah pecahan rasional, keluarkan dalam bentuk pecahan rasional pula (dalam bentuk paling sederhana).

Latihan Kelas

Contoh masukan:

```
3
1 0.3 0.25
2 1 2 1 4
2 1 3 1 1
```

Contoh keluaran:

```
0.55
3 4
4 3
```