



Tutorial Week 10

Interface, Polymorphism, Enumeration

Interface: Shape

Penjelasan:

- Interface adalah kumpulan metode yang tidak memiliki implementasi.
- Interface Shape digunakan untuk menetapkan kontrak bahwa setiap bentuk geometri harus memiliki metode `calculateArea()`.

Kegunaan:

- Interface digunakan untuk mendefinisikan kontrak atau aturan yang harus diikuti oleh kelas yang mengimplementasikannya.
- Ini memungkinkan adanya polimorfisme, di mana objek dari kelas yang berbeda dapat diakses dan digunakan melalui antarmuka yang sama.

Alasan Penggunaan:

- Dengan menggunakan interface, kita dapat memastikan bahwa setiap bentuk geometri memiliki metode `calculateArea()`, tanpa perlu peduli tentang implementasi spesifik dari setiap bentuk.

Enumeration: ShapeType

Penjelasan:

- Enumeration adalah kumpulan nilai konstan yang dapat digunakan untuk mewakili kumpulan pilihan yang mungkin.
- Enumeration ShapeType digunakan untuk mewakili jenis bentuk yang tersedia dengan nilai-nilai konstan.

Kegunaan:

- Enumeration digunakan untuk menyediakan himpunan nilai yang terbatas dan terdefinisi dengan jelas.
- Ini membantu dalam membatasi pilihan menjadi nilai-nilai yang valid dan meningkatkan kejelasan dan keamanan kode.

Alasan Penggunaan:

- Dengan menggunakan enumeration ShapeType, kita dapat dengan jelas menentukan jenis-jenis bentuk yang tersedia dan memastikan bahwa hanya nilai-nilai yang valid yang dapat digunakan sebagai jenis bentuk.

Polymorphism

Penjelasan:

- Polimorfisme memungkinkan sebuah objek untuk memiliki banyak bentuk (poli = banyak, morph = bentuk).
- Di dalam Shape Area Calculator, polimorfisme memungkinkan kita untuk menghitung luas dari berbagai bentuk geometri tanpa peduli jenis spesifik bentuk yang digunakan.

Kegunaan:

- Polimorfisme memungkinkan kita untuk menggunakan objek dari berbagai kelas yang berbeda dengan cara yang sama melalui antarmuka yang sama.
- Ini mempermudah pengembangan, pemeliharaan, dan perluasan kode, karena kita dapat menangani berbagai objek secara generik tanpa harus mengetahui jenis spesifiknya.

Polymorphism

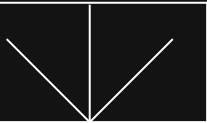
Alasan Penggunaan:

Dalam Shape Area Calculator, kita ingin menghitung luas dari berbagai bentuk geometri seperti lingkaran, persegi panjang, dan segitiga. Dengan menggunakan polimorfisme, kita dapat dengan mudah menghitung luas dari setiap bentuk geometri tanpa perlu mengetahui jenis spesifik bentuk yang digunakan.

driver

ShapeCalculatorDriver.java

`ShapeCalculatorDriver.java` adalah kelas utama yang berisi method `main()` untuk menjalankan program "Shape Area Calculator".



model

Shape.java

Interface untuk bentuk geometri

Circle.java

Implementasi lingkaran

Rectangle.java

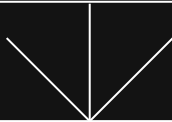
Implementasi persegi panjang

Triangle.java

Implementasi segitiga

ShapeType.java

Enumeration untuk jenis bentuk



Interface

Kumpulan `Method` yang tidak memiliki implementasi

Ini adalah antarmuka(interface) Java yang memuat metode `calculateArea()`, yang harus diimplementasikan oleh kelas-kelas yang mewarisi antarmuka ini untuk menghitung luas bentuk geometris.

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


Contoh Pengimplementasian Interface Shape

```
public class Rectangle implements Shape {  
    private double length;  
    private double width;  
  
    public Rectangle(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
  
    @Override  
    public double calculateArea() {  
        return length * width;  
    }  
}
```

Kelas Rectangle mengimplementasikan antarmuka Shape, yang berarti kelas ini harus menyediakan implementasi untuk metode `calculateArea()` yang didefinisikan dalam antarmuka Shape.



Enumeration

Enumerasi Java yang mendefinisikan tipe-tipe bentuk geometri seperti lingkaran, persegi panjang, dan segitiga

```
public enum ShapeType
{
    CIRCLE,
    RECTANGLE,
    TRIANGLE
}
```

Contoh Penggunaan Enumeration ShapeType



Pada kode di sebelah kanan menggunakan struktur switch untuk memilih tipe bentuk (ShapeType) berdasarkan nilai pilihan (choice) yang diberikan.

Sedangkan pada kode di bawah ini menggunakan struktur if-else if untuk menentukan jenis bentuk geometri berdasarkan shapeType yang telah dipilih, dan kemudian memanggil fungsi `calculateArea` dengan parameter sesuai dengan shapeType.

```
switch (choice) {  
    case 0:  
        shapeType = ShapeType.CIRCLE;  
        break;  
    case 1:  
        shapeType = ShapeType.RECTANGLE;  
        break;  
    case 2:  
        shapeType = ShapeType.TRIANGLE;  
        break;  
}
```

```
if (shapeType == ShapeType.CIRCLE) {  
    System.out.println("Luas Lingkaran: " + calculateArea(circle));  
} else if (shapeType == ShapeType.RECTANGLE) {  
    System.out.println("Luas Persegi Panjang: " + calculateArea(rectangle));  
} else if (shapeType == ShapeType.TRIANGLE) {  
    System.out.println("Luas Segitiga: " + calculateArea(triangle));  
}
```

Polymorphism

```
System.out.println("Luas Persegi Panjang: " + calculateArea(rectangle));
```

```
private static double calculateArea(Shape shape) {  
    return shape.calculateArea();  
}
```

Method `calculateArea(Shape shape)` merupakan contoh polimorfisme dalam Java, di mana metode tersebut dapat menerima objek dari kelas apa pun yang mengimplementasikan Interface `Shape`, sehingga memungkinkan penggunaan Polymorphism untuk memanggil metode `calculateArea()` yang sesuai dengan jenis bentuk geometri yang diberikan sebagai argumen.

Latihan

Buatlah program yang mirip dengan code yang telah diberikan dengan study case berbeda.

Contoh: Vehicle Management System

Link Program yang dibahas:

<https://github.com/samuelsihotang1/Shape-Area-Calculator-Java>

