

Praktikum 9

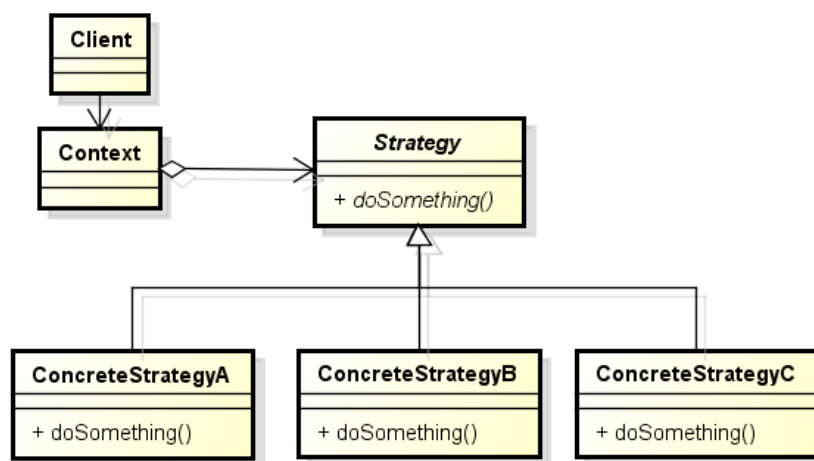
Analisis dan Desain Berorientasi Objek

Strategy Design Pattern

Seperti yang telah dijelaskan di kelas, Strategy Design Pattern bertujuan untuk mengenkapsulasi beberapa algoritma, memungkinkan algoritma-algoritma tersebut saling menggantikan untuk memenuhi sebuah tujuan. Tujuan lain dari Strategy Design Pattern adalah menangkap abstraksi dalam sebuah interface dan menyembunyikan implementasinya pada kelas turunannya.

Struktur dari Strategy Design Pattern

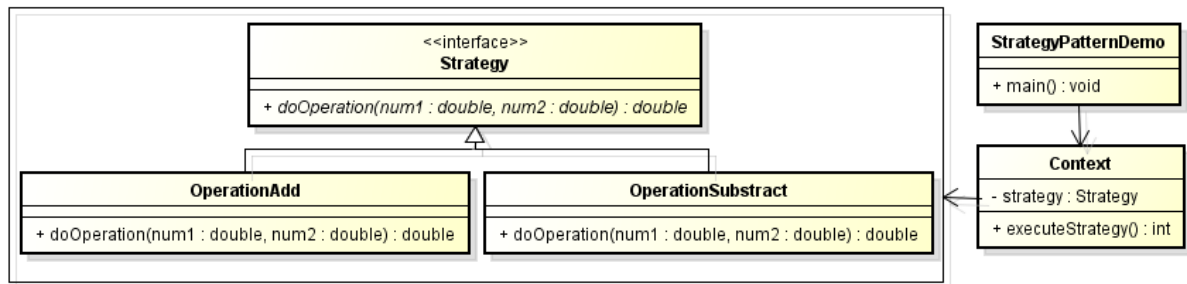
Berikut ini adalah struktur class diagram untuk Strategy Design Pattern:



- **Strategy** adalah interface yang mencakup abstraksi dari algoritma-algoritma yang akan diimplementasikan. Context menggunakan interface ini untuk memanggil algoritma yang diimplementasikan pada ConcreteStrategy.
- **ConcreteStrategy** adalah pengimplementasian konkret dari sebuah algoritma.
- **Context** berguna untuk menentukan referensi untuk sebuah objek Strategy
- **Context** mengirimkan request dari client pada Strategy. Pada umumnya, client mengirimkan sebuah objek ConcreteStrategy pada Context.

P0901xxyyy

Saat ini kita akan mencoba mengimplementasikan Strategy Design Pattern untuk sebuah kasus sederhana, yaitu perhitungan matematika yang sangat sederhana. Kita akan membuat sebuah method `doOperation` implementasi dari operation dapat berbeda-beda, yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Operan dari operasi-operasi tersebut adalah bilangan bulat. Berikut adalah class diagram untuk kasus ini:



Interface Strategy menjadi pemegang kontrak untuk kelas-kelas yang mengimplementasikan interface tersebut. Kelas OperationAdd dan OperationSubtract menjadi pengimplementasi kongkret untuk masing-masing operasi. Pada main method pada kelas StrategyPatternDemo, ditentukan Operation yang akan diimplementasikan sesuai kebutuhan user. Kelas StrategyPatternDemo tidak berurusan secara langsung dengan interface atau kelas-kelas Operation, melainkan melalui Context, karena Context-lah yang akan menentukan implementasi yang dijalankan.

Buatlah interface Strategy:

```

public interface Strategy {
    public double doOperation(double num1, double num2);
}

```

Buatlah class OperationAdd sebagai implementasi kongkrit dari method doOperation:

```

public class OperationAdd implements Strategy{
    @Override
    public double doOperation(double num1, double num2){
        return (num1+num2);
    }
}

```

Implementasikan juga class OperationSubtract.

Buatlah class Context sebagai berikut:

```

public class Context {
    private Strategy strategy;
    public Context(Strategy strategy){
        this.strategy = strategy;
    }
    public double executeStrategy(double num1, double num2){
        return strategy.doOperation(num1, num2);
    }
}

```

Tambahkan juga class untuk mengimplementasikan perkalian dan pembagian. Buatlah main class yang menerima input dua buah bilangan dan mencoba mengeksekusi berbagai operasi yang ada. Perhatikan bahwa dengan menggunakan strategy pattern, Anda dapat menambahkan operasi dengan mudah, tanpa banyak berpengaruh pada bagian lain dalam program Anda.

Sebagian dari output program adalah sebagai berikut:

```

10.0 + 5.0 = 15.0
10.0 - 5.0 = 5.0

```

Perbayaran Kuliah (P0902xyyy)

Womniv adalah sebuah universitas yang terkenal di negeri wombat. Tim developer Womniv sedang kebingungan untuk membuat sistem penghitung tagihan para mahasiswa wombat. Berikut adalah ketentuan yang harus mereka ikuti:

1. Terdapat empat jenis sks
 - a. sksKelas dengan biaya Rp 210.000,- per sks
 - b. sksPraktikum dengan biaya Rp 50.000,- per sks
 - c. sksResponsi dengan biaya Rp 45.000,- per sks
 - d. sksStudio dengan biaya Rp 210.000,- per sks
2. Terdapat tiga metode pembayaran
 - a. Pembayaran Tunai
Pembayaran tunai adalah pembayaran yang paling sederhana karena tidak mendapat diskon apapun. Perhitungan tagihan hanya dilakukan dengan menjumlahkan seluruh biaya sks.
 - b. Pembayaran dengan transfer/e-banking
Wombat yang mendaftar denganc ara ini akan mendapat diskon sebesar 5% dari seluruh tagihannya.
 - c. Pembayaran dengan credit card
Wombat yang membayar dengan metode ini, mendapat diskon sebesar 10% untuk sksKelas dan 3% untuk sksStudio.

Rektor Womniv mengundang Anda untuk menjadi developer sistem tersebut. Beberapa hal yang perlu diperhatikan adalah sistem tersebut menggunakan User Interface (UI, bukan cosole) dan menerima input berupa jumlah masing-masing elemen sks yang ada (4 jenis sks). Sistem tersebut juga memberikan tiga pilihan jenis pembayaran dengan radio button. Output yang dihasilkan dari sistem tersebut adalah tagihan yang harus dibayarkan oleh seekor mahasiswa Wombat.

Anda dipersilakan merancang UI untuk sistem tersebut. Selain source code Anda, kumpulkan juga class diagram yang Anda gunakan untuk sistem tersebut.