

DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Design Pattern

Filière :
« Génie du Logiciel et des Systèmes Informatiques Distribués »
GLSID

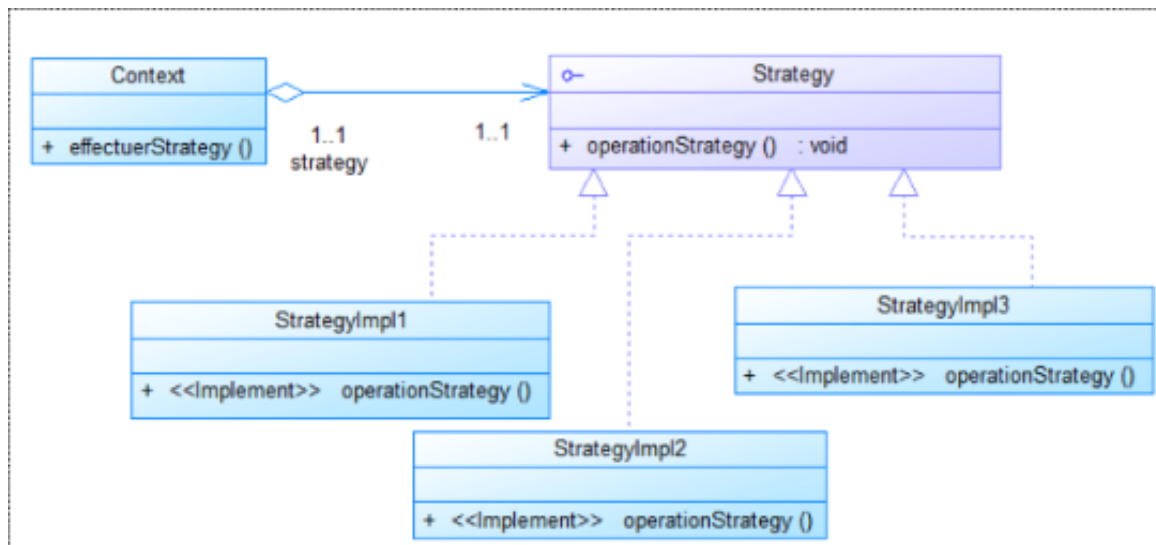
Pattern Strategy

Réalisé par :

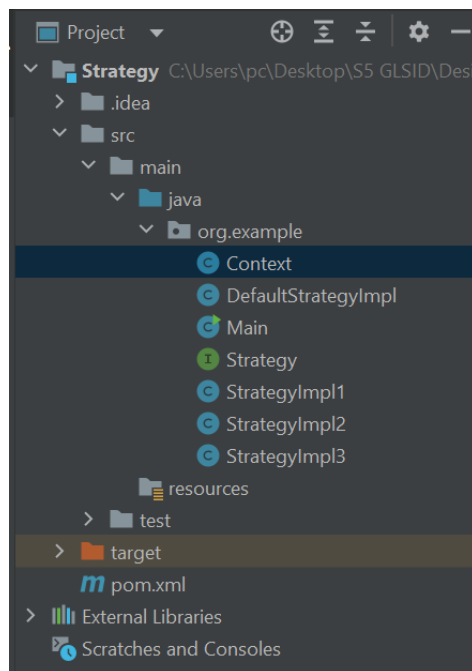
Najat ES-SAYYAD

Année Universitaire : 2023-2024

Pattern Strategy



Structure de projet :



Classe Context :

```
1 package org.example;
2
3 2 usages
4 public class Context {
5     2 usages
6     private Strategy strategy=new DefaultStrategyImpl();
7     4 usages
8     public void effectuerOperation(){
9         System.out.println("*****");
10        strategy.operationStrategy();
11        System.out.println("+++++++\n");
12    }
13
14    3 usages
15    public void setStrategy(Strategy strategy) {
16        this.strategy = strategy;
17    }
18 }
```

Interface Strategy :

```
1 package org.example;
2
3 6 usages 4 implementations
4 public interface Strategy {
5     2 usages 4 implementations
6     void operationStrategy();
7 }
8 }
```

Implémentation 1 :

```
1 package org.example;
2
3 public class StrategyImpl1 implements Strategy{
4     @Override
5     public void operationStrategy() {
6         System.out.println(" .....Strategy 1..... ");
7     }
8 }
```

Implémentation 2 :

```
1 package org.example;
2
3 public class StrategyImpl2 implements Strategy{
4     @Override
5     public void operationStrategy() {
6         System.out.println(" .....Strategy 2..... ");
7     }
8 }
```

Implémentation 3 :

```
1 package org.example;
2
3 public class StrategyImpl3 extends DefaultStrategyImpl implements Strategy{
4     @Override
5     public void operationStrategy() {
6         super.operationStrategy();
7         System.out.println(" .....Strategy 3..... ");
8     }
9 }
```

Implémentation par défaut :

```
1 package org.example;
2
3 public class DefaultStrategyImpl implements Strategy{
4     @Override
5     public void operationStrategy() {
6         System.out.println(" .....Default Strategy..... ");
7     }
8 }
```

Classe Main La façon 1 :

```
1 package org.example;
2
3 import java.util.Scanner;
4
5 no usages
6 public class Main {
7     no usages
8     public static void main(String[] args) {
9         Context context=new Context() ;
10        context.effectuerOperation();
11        context.setStrategy(new StrategyImpl1());
12        context.effectuerOperation();
13        context.setStrategy(new StrategyImpl2());
14        context.effectuerOperation();
15        context.setStrategy(new StrategyImpl3());
16        context.effectuerOperation();
17    }
```

Résultat :

```
↑ "C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent
↓ *****
☰ .....Default Strategy.....
☑ ++++++
☒ *****
☒ .....Strategy 1.....
☒ ++++++
*****
.....Strategy 2.....
++++++
*****
.....Default Strategy.....
.....Strategy 3.....
++++++

Process finished with exit code 0
```

Classe Main La façon 2 :

```
1 package org.example;
2
3 import java.util.Scanner;
4
5 no usages
6 public class Main {
7     no usages
8     public static void main(String[] args) throws Exception {
9         Context context=new Context() ;
10        Scanner scanner=new Scanner(System.in);
11        while (true){
12            System.out.println("Quelle stratégie ? : ");
13            String str=scanner.nextLine();
14            Strategy strategy=(Strategy) Class.forName( className: "org.example.StrategyImpl"+str).newInstance();
15            context.setStrategy(strategy);
16            context.effectuerOperation();
17        }
18    }
19 }
```

Résultat :

```
↑ "C:\Program Files\Java\jdk-17\bin\java.exe" "-jav
↓
Quelle stratégie ? :
1
*****
.....Strategy 1.....
+++++++

Quelle stratégie ? :
2
*****
.....Strategy 2.....
+++++++

Quelle stratégie ? :
3
*****
.....Default Strategy.....
.....Strategy 3.....
+++++++

Quelle stratégie ? :
```