

DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Design Pattern

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

Pattern Adapter

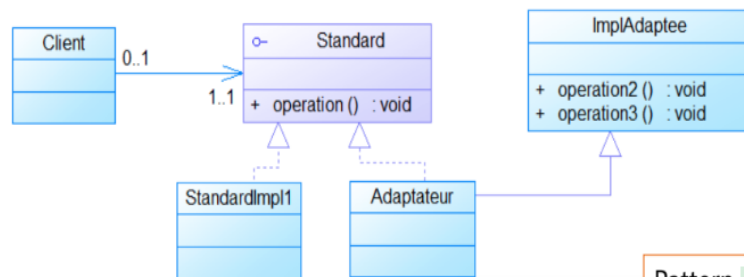
Réalisé par :

Najat ES-SAYYAD

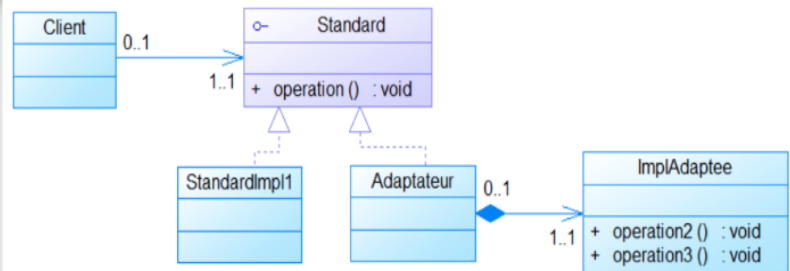
Année Universitaire : 2023-2024

Pattern Adapter

Pattern Adapter par héritage

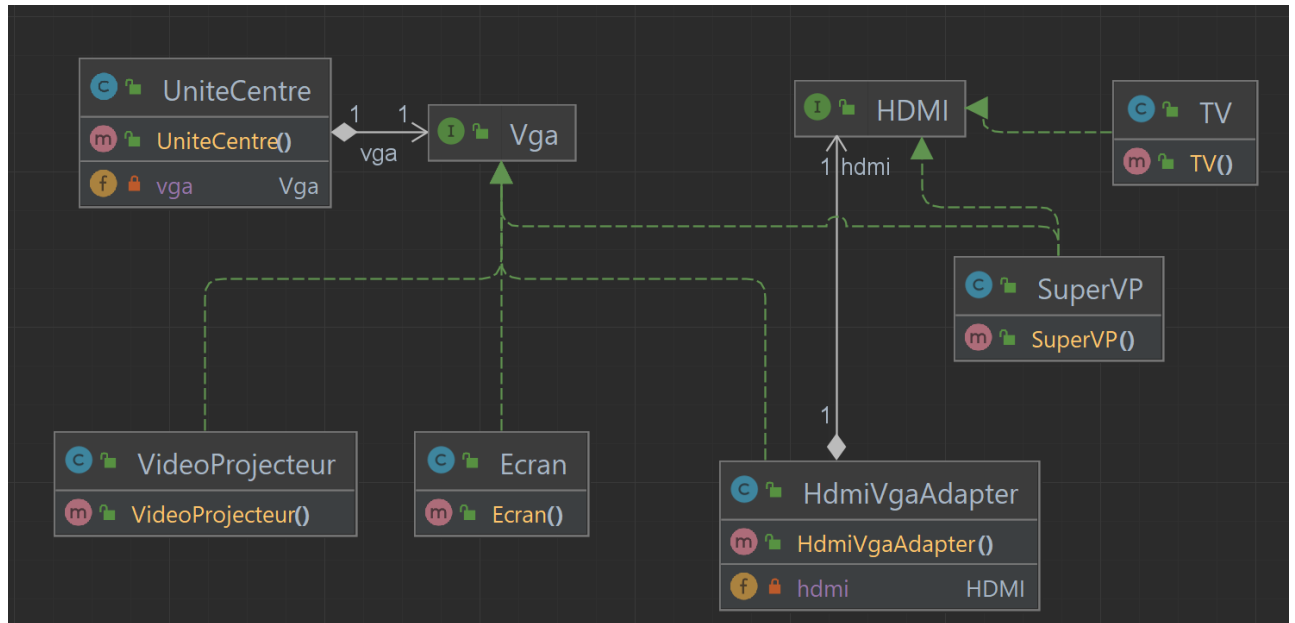


Pattern Adapter par Composition

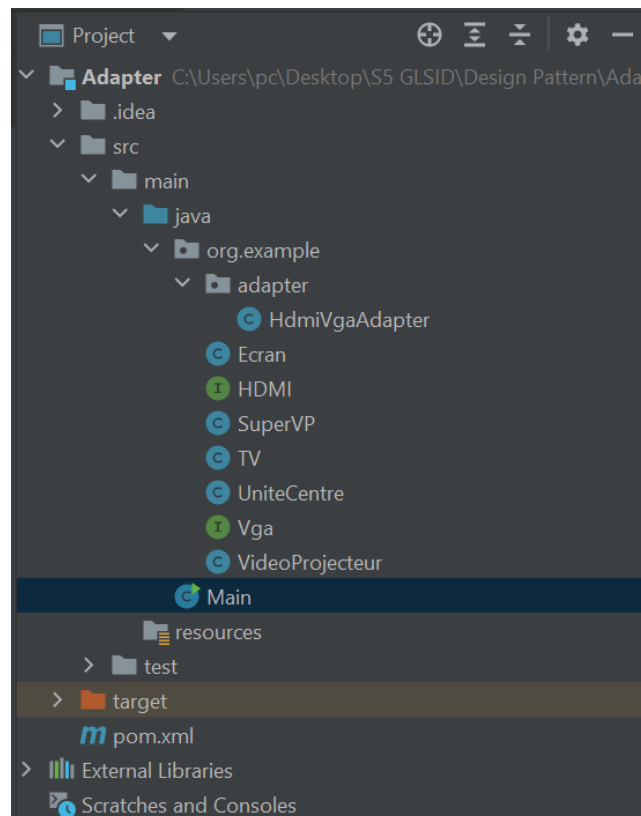


Première Partie : Pattern Adapter par Composition

Diagramme de Classe :



Structure de projet :



Classe UniteCentre :

```
1 package org.example;
2
3 3 usages
4 public class UniteCentre {
5     2 usages
6     private Vga vga;
7
8     2 usages
9     public void print(String message){
10         System.out.println("*****");
11         vga.print(message);
12         System.out.println("*****\n");
13     }
14
15     3 usages
16     public void setVga(Vga vga) { this.vga = vga; }
```

Interface Vga :

```
1 package org.example;
2
3 7 usages 4 implementations
4 public interface Vga {
5     2 usages 4 implementations
6     void print(String message);
7 }
```

La classe Ecran :

```
1 package org.example;
2
3 2 usages
4 public class Ecran implements Vga{
5     2 usages
6     @Override
7     public void print(String message) {
8         System.out.println("=====Ecran=====");
9         System.out.println(message);
10        System.out.println("=====Ecran=====\n");
11    }
12 }
```

La classe VideoProjecteur :

```
1 package org.example;
2
3 2 usages
4 public class VideoProjecteur implements Vga{
5     2 usages
6     @Override
7     public void print(String message) {
8         System.out.println(".....Video.....");
9         System.out.println(message);
10        System.out.println(".....Video.....");
11    }
12 }
```

L'interface Hdmi :

```
1 package org.example;
2
3 5 usages 2 implementations
4 public interface HDMI {
5     1 usage 2 implementations
6     void view(byte[] data);
7 }
8
```

La classe TV :

```
1 package org.example;
2
3 2 usages
4 public class TV implements HDMI{
5     1 usage
6     @Override
7     public void view(byte[] data) {
8         System.out.println("??????? TV ??????");
9         String message=new String(data);
10        System.out.println(message);
11        System.out.println("??????? TV ??????");
12    }
13 }
```

La classe HdmiVgaAdapter :

```
1      package org.example.adapter;
2
3      import org.example.HDMI;
4      import org.example.Vga;
5
6      3 usages
7      public class HdmiVgaAdapter implements Vga {
8          2 usages
9          private HDMI hdmi;
10         2 usages
11         @Override
12         public void print(String message) {
13             System.out.println("===== Adapter =====");
14             byte[] data=message.getBytes();
15             hdmi.view(data);
16             System.out.println("===== Adapter =====");
17         }
18
19         1 usage
20         public void setHdmi(HDMI hdmi) { this.hdmi=hdmi; }
21     }
```

La classe SuperVP :

```
1      package org.example;
2
3      2 usages
4      public class SuperVP implements Vga, HDMI{
5          1 usage
6          @Override
7          public void view(byte[] data) {
8              String message =new String(data);
9              System.out.println(".....super VP Hdmi..... ");
10             System.out.println(message);
11             System.out.println(".....super VP Hdmi..... ");
12         }
13
14         2 usages
15         @Override
16         public void print(String message) {
17             System.out.println(".....super VP Vga..... ");
18             System.out.println(message);
19             System.out.println(".....super VP Vga..... ");
20         }
21     }
```

Test :

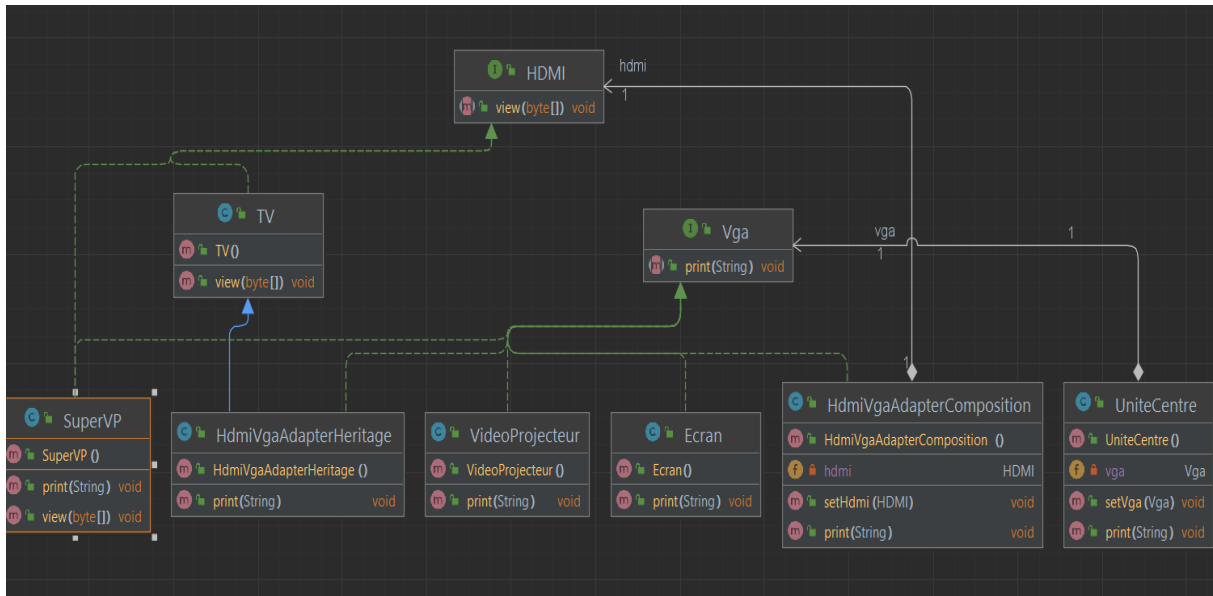
```
4  ▶ public class Main {  
    no usages  
5  ▶  public static void main(String[] args) {  
6      UniteCentre uniteCentre=new UniteCentre();  
7      uniteCentre.setVga(new Ecran());  
8      uniteCentre.print("Bonjour");  
9      uniteCentre.setVga(new VideoProjecteur());  
10     uniteCentre.print("Bonjour");  
11  
12     HdmiVgaAdapter hdmiVgaAdapter=new HdmiVgaAdapter();  
13     hdmiVgaAdapter.setHdmi(new TV());  
14     uniteCentre.setVga(hdmiVgaAdapter);  
15     hdmiVgaAdapter.print("Bonsoir GLSID Design Pattern");  
16  
17     uniteCentre.setVga(new SuperVP());  
18     uniteCentre.print("Hello");  
19  
20     HdmiVgaAdapter hdmiVgaAdapter1=new HdmiVgaAdapter();  
21     hdmiVgaAdapter1.setHdmi(new SuperVP());  
22     uniteCentre.setVga(hdmiVgaAdapter1);  
23     uniteCentre.print("Bonsoir");  
24 }  
25 }
```

Résultat :

```
*****  
=====Ecran=====  
Bonjour  
=====Ecran=====  
*****  
*****  
.....Video.....  
Bonjour  
.....Video.....  
*****  
===== Adapter =====  
???????? TV ???????  
Bonsoir GLSID Design Pattern  
???????? TV ???????  
===== Adapter =====  
*****  
.....super VP Vga.....  
Hello  
.....super VP Vga.....  
*****  
*****  
===== Adapter =====  
.....super VP Hdmi.....  
Bonsoir  
.....super VP Hdmi.....  
===== Adapter =====  
*****
```

Deuxième Partie : Pattern Adapter par Composition

Diagramme de Classe :



La classe HdmiVgaAdapterHeritage :

```
3 usages
7   public class HdmiVgaAdapterHeritage extends TV implements Vga {
8
9       3 usages
10      @Override
11      public void print(String message) {
12          System.out.println("==== Adapter =====");
13          byte[] data=message.getBytes();
14          super.view(data);
15          System.out.println("==== Adapter =====");
16      }
17  }
```

Test :

```
HdmiVgaAdapterHeritage hdmiVgaAdapterHeritage=new HdmiVgaAdapterHeritage();
uniteCentre.setVga(hdmiVgaAdapterHeritage);
hdmiVgaAdapterHeritage.print("Test par heritage");
}
```


Résultat :

```
===== Adapter =====  
???????? TV ?????????  
Test par héritage  
???????? TV ?????????  
===== Adapter =====
```