ECOLE NORMALE SUPÉRIEURE DE
L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA

ENSET

المدرسة العليا لأساتذة التعليم التقني
المحمدية
جامعة الحسن الثاني بالدار البيضاء

DEPARTEMENT MATHEMATIQUES ET INFORMATIQUE

Design Pattern

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

Pattern Observer

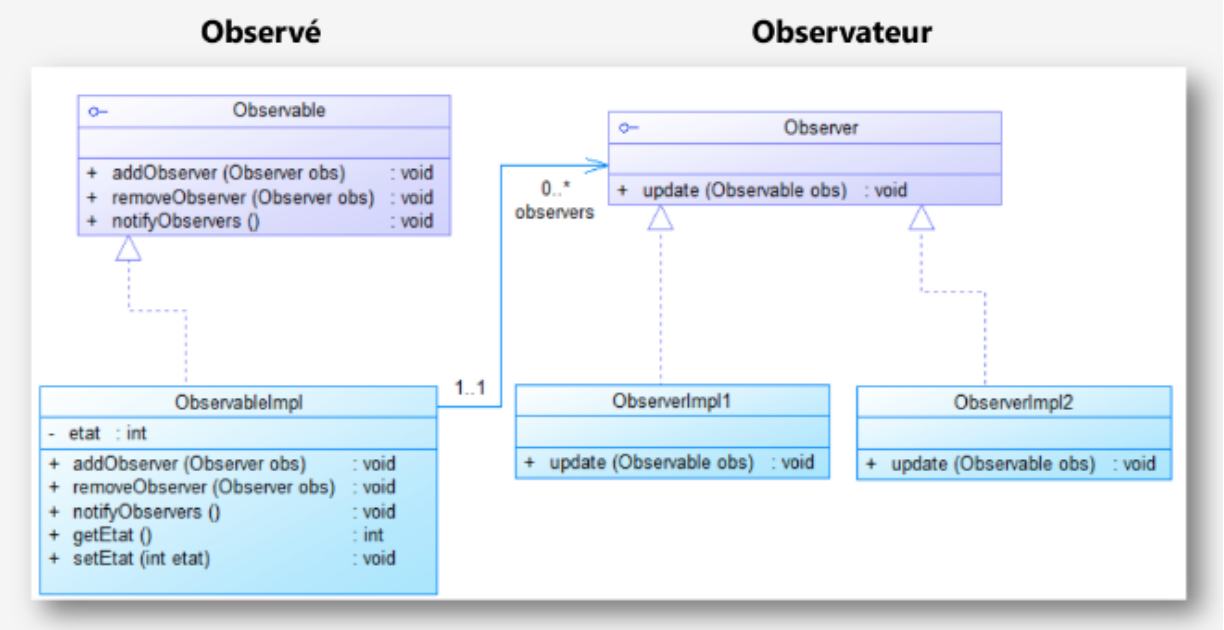
Réalisé par :

Najat ES-SAYYAD

Année Universitaire : 2023-2024

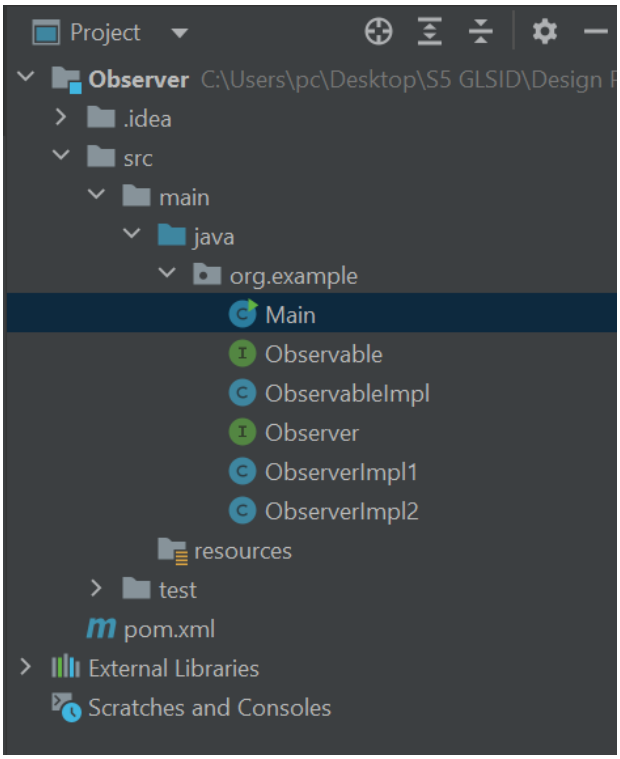
::

Pattern Observer



Pop from observable model

Structure de projet :



Interface Observer :

```java
package org.example;

12 usages  3 implementations
public interface Observer {
    1 usage  3 implementations
    void update(Observable observable);
}
```

Interface Observable :

```java
package org.example;

5 usages  1 implementation
public interface Observable {
    5 usages  1 implementation
    void subscribe(Observer o) ; // addObserver
    1 usage  1 implementation
    void unsubscribe(Observer o) ; // removeObserver
    1 usage  1 implementation
    void notifyObservers();
}
```

ObservableImpl :

```java
public class ObservableImpl implements Observable {
    2 usages
    private int state=10;
    3 usages
    private List<Observer> observers=new ArrayList<>();
    5 usages
    @Override
    public void subscribe(Observer o) { this.observers.add(o); }
    1 usage
    @Override
    public void unsubscribe(Observer o) { this.observers.remove(o); }
    1 usage
    @Override
    public void notifyObservers() {
        for (Observer o:observers){
            o.update( observable: this);
        }
    }
    3 usages
    public void setState(int state) {
        this.state = state;
        this.notifyObservers();
    }
    2 usages
    public int getState() { return state; }
```

ObserverImpl 1 :

```java
package org.example;

2 usages
public class ObserverImpl1 implements Observer {
    1 usage
    @Override
    public void update(Observable observable) {
        int state=((ObservableImpl)observable).getState();
        double res=state*state+9;
        System.out.println("****** ObserverImpl1 ******");
        System.out.println("Nouvelle mise à jour : state="+state);
        System.out.println("Résultat de calcul :"+res);
        System.out.println("**************************");

    }
}
```

ObserverImpl 2 :

```java
package org.example;

1 usage
public class ObserverImpl2 implements Observer {
    2 usages
    private int counter;
    1 usage
    @Override
    public void update(Observable observable) {
        int state=((ObservableImpl)observable).getState();
        if(state%2==0) ++counter;
        System.out.println("****** ObserverImpl2 ******");
        System.out.println("Nouvelle mise à jour : state="+state);
        System.out.println("Résultat de calcul :"+((state%2)==0?"Pair":"Impair"));
        System.out.println("Le compteur est :"+counter);
        System.out.println("**************************");

    }
}
```

Main :

```java
public class Main {
    no usages
    public static void main(String[] args) {
        ObservableImpl observable=new ObservableImpl();
        Observer o1=new ObserverImpl1();
        Observer o2=new ObserverImpl2();
        Observer o3=new ObserverImpl1();
        observable.subscribe(o1);
        observable.subscribe(o2);
        observable.subscribe(o3);
        observable.setState(22);
        observable.setState(33);
        observable.unsubscribe(o1);
        observable.subscribe(new Observer() {
            1 usage
            @Override
            public void update(Observable observable) {
                System.out.println("Observateur anonyme ======>");
            }
        });
        observable.subscribe(obs -> System.out.println("Observateur anonyme ======>"));
        observable.setState(77);
    }
}
```

Résultat :

```
****** ObserverImpl1 ******
Nouvelle mise à jour : state=22
Résultat de calcul :493.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=22
Résultat de calcul :Pair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=22
Résultat de calcul :493.0
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=33
Résultat de calcul :1098.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=33
Résultat de calcul :Impair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=33
Résultat de calcul :1098.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=77
Résultat de calcul :Impair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=77
Résultat de calcul :5938.0
**************************
Observateur anonyme ======>
Observateur anonyme ======>

Process finished with exit code 0
```

Push to observers model

Interface Observer :

```java
package org.example;

12 usages  3 implementations
public interface Observer {
    //void update(Observable observable);
    1 usage  3 implementations
    void update(int state);
}
```

ObservableImpl :

```java
public class ObservableImpl implements Observable {
    3 usages
    private int state=10;
    3 usages
    private List<Observer> observers=new ArrayList<>();
    5 usages
    @Override
    public void subscribe(Observer o) { this.observers.add(o); }
    1 usage
    @Override
    public void unsubscribe(Observer o) { this.observers.remove(o); }
    1 usage
    @Override
    public void notifyObservers() {
        for (Observer o:observers){
           // o.update(this);
            o.update(this.state);
        }}
    3 usages
    public void setState(int state) {
        this.state = state;
        this.notifyObservers();
    }
    no usages
    public int getState() { return state; }
```

ObserverImpl 1 :

```java
package org.example;

2 usages
public class ObserverImpl1 implements Observer {
    1 usage
    @Override
    public void update(int state) {
      //  int state=((ObservableImpl)observable).getState();
        double res=state*state+9;
        System.out.println("****** ObserverImpl1 ******");
        System.out.println("Nouvelle mise à jour : state="+state);
        System.out.println("Résultat de calcul :"+res);
        System.out.println("**************************");

    }
}
```

ObserverImpl 2 :

```java
package org.example;

1 usage
public class ObserverImpl2 implements Observer {
    2 usages
    private int counter;
    1 usage
    @Override
    public void update(int state) {
    //    int state=((ObservableImpl)observable).getState();
        if(state%2==0) ++counter;
        System.out.println("****** ObserverImpl2 ******");
        System.out.println("Nouvelle mise à jour : state="+state);
        System.out.println("Résultat de calcul :"+((state%2)==0?"Pair":"Impair"));
        System.out.println("Le compteur est :"+counter);
        System.out.println("**************************");
    }
}
```

Main :

```java
public class Main {
    no usages
    public static void main(String[] args) {
        ObservableImpl observable=new ObservableImpl();
        Observer o1=new ObserverImpl1();
        Observer o2=new ObserverImpl2();
        Observer o3=new ObserverImpl1();
        observable.subscribe(o1);
        observable.subscribe(o2);
        observable.subscribe(o3);
        observable.setState(22);
        observable.setState(33);
        observable.unsubscribe(o1);
//...
        observable.subscribe(new Observer() {
            1 usage
            @Override
            public void update(int state) {
                System.out.println("Observateur anonyme ======>");
            }
        });
        observable.subscribe(obs -> System.out.println("Observateur anonyme =======>"));
        observable.setState(77);
    }
}
```

Résultat :

```
****** ObserverImpl1 ******
Nouvelle mise à jour : state=22
Résultat de calcul :493.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=22
Résultat de calcul :Pair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=22
Résultat de calcul :493.0
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=33
Résultat de calcul :1098.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=33
Résultat de calcul :Impair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=33
Résultat de calcul :1098.0
**************************
****** ObserverImpl2 ******
Nouvelle mise à jour : state=77
Résultat de calcul :Impair
Le compteur est :1
**************************
****** ObserverImpl1 ******
Nouvelle mise à jour : state=77
Résultat de calcul :5938.0
**************************
Observateur anonyme ======>
Observateur anonyme ======>

Process finished with exit code 0
```