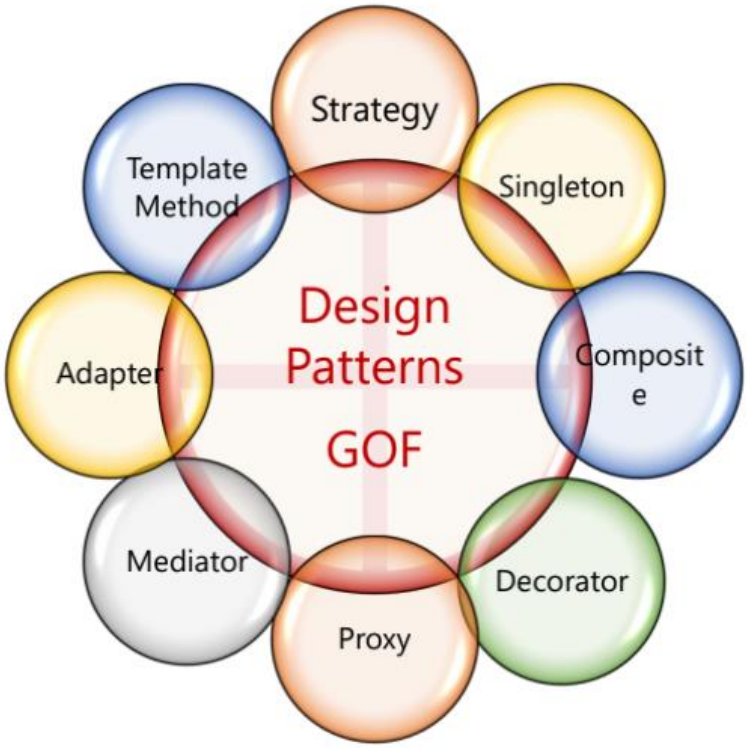
جـامعـة الحسن الثاني بالدار البيضاء
UNIVERSITÉ HASSAN II DE CASABLANCA

ENSET
ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

Observer Pattern



Réalisé par : REGOUG Zakia

GLSID3

Année universitaire : 2023-2024

Catégorie :

Comportement

Objectif :

Définir une relation entre les objets de type un a plusieurs, de façon que, lorsqu'un objet change d'état, tous ce qui ont dépendent en soient informés et en soient mise à jour automatiquement.

Diagramme de Séquence :

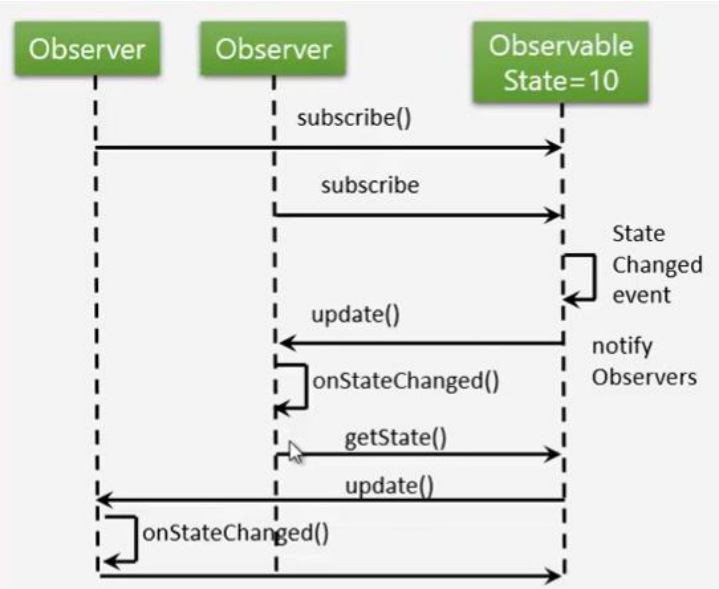
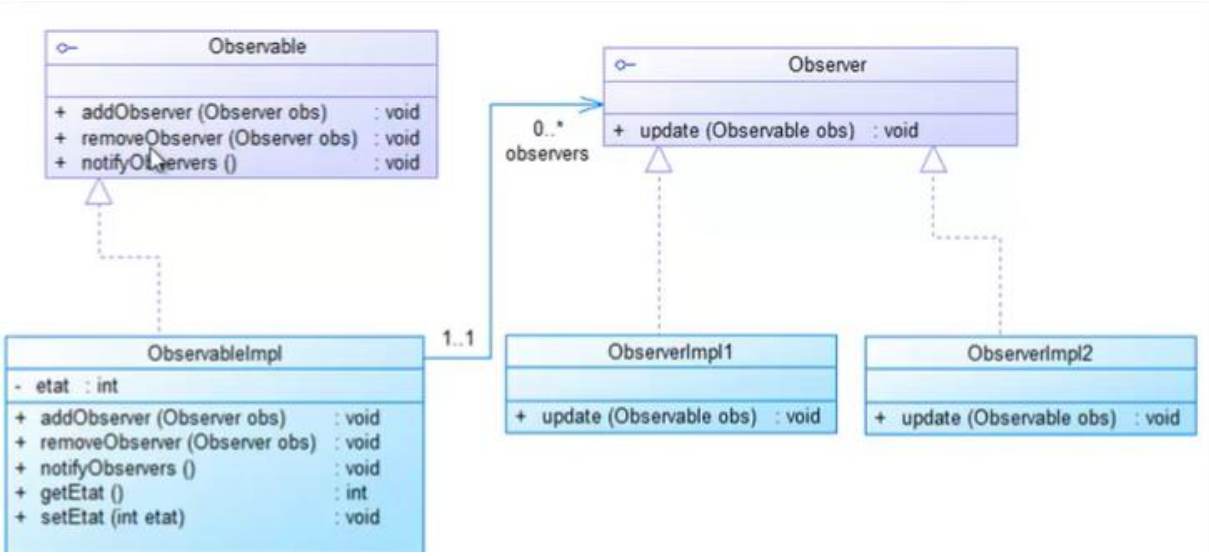


Diagramme de Classe :



Interface Observable :

```
public interface Observable {
    public void subscribe(Observer o);
    public void unsubscribe(Observer o);
    public void notifyObservers();
}
```

Implémentation observable :

```
import java.util.ArrayList;
import java.util.List;

public class ObservableImpl implements Observable {
    private int state=10;
    private List<Observer> observers=new ArrayList<>();
    @Override
    public void subscribe(Observer o) {
        this.observers.add(o);
    }

    @Override
    public void unsubscribe(Observer o) {
        this.observers.remove(o);
    }

    @Override
    public void notifyObservers() {
        for(Observer o:observers){
            o.update(this);
        }
    }

    public void setState(int state) {
        this.state = state;
        this.notifyObservers();
    }

    public int getState() {
        return state;
    }
}
```

Interface Observer :

```
public interface Observer {
    public void update(Observable observable);// update recoie la reference
de l'objet notifie
}
```

Implémentation observer 1 :

```
public class ObserverImpl implements Observer {
    @Override
    public void update(Observable observable) {
        int state=((ObservableImpl)observable).getState();
        double res=state*state+9;
        System.out.println("********** Observer Impl 1 *********");
```

```
        System.out.println("Nouvelle  mise a jour state : "+state);
        System.out.println("Resultat : "+res);
    }
}
```

Implémentation observer 2 :

```
public class ObserverImpl2 implements Observer {
    private int counter=0;
    @Override
    public void update(Observable observable) {
        int state=((ObservableImpl)observable).getState();
        if(state%2==0) ++counter;
        System.out.println("********** Observer Impl 2 *********");
        System.out.println("Nouvelle  mise a jour state :
"+(state%2==0?"Pair":"Impair"));
        System.out.println("Compteur : "+counter);
    }
}
```

Main

```
public class Main {
    public static void main(String[] args) {
       ObservableImpl observable=new ObservableImpl();
       Observer o1=new ObserverImpl();
       Observer o2=new ObserverImpl2();
       Observer o3=new ObserverImpl();

       observable.subscribe(o1);
        observable.subscribe(o2);
        observable.subscribe(o3);
       observable.setState(44);

        observable.setState(39);
        observable.setState(24);
    }
}
```

Affichage :