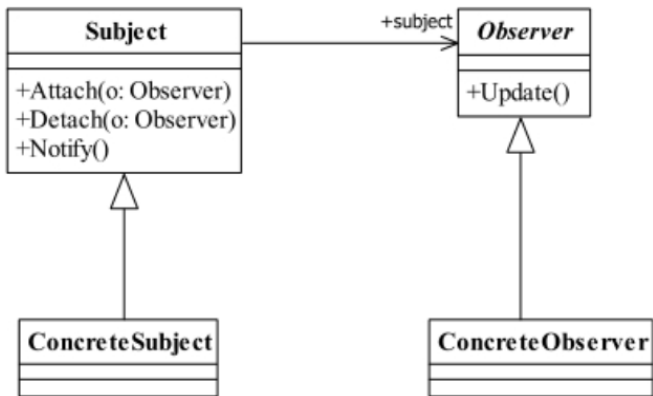


```
public abstract class Subject{
    //定义一个观察者数组
    private Vector<Observer> obsVector=new Vector<Observer>();
    //增加一个观察者
    public void addObserver(Observer o){
        this.obsVector.add(o);
    }
    //删除一个观察者
    public void delObserver(Observer o){
        this.obsVector.remove(o);
    }
    //通知所有观察者
    public void notifyObservers(){
        for(Observer o: this.obsVector){
            o.update();
        }
    }
}
```

```
public interface Observer{
    //更新方法
    public void update();
}
```



```
public class ConcreteSubject extends Subject{
    //具体的业务
    public void doSomething(){
        /// do something
        super.notifyObservers();
    }
}
```

```
public class ConcreteObserver implements Observer{
    //实现更新方法
    public void update(){
        System.out.println("接收到信息，并进行处理！")
    }
}
```

```
public class Client{
    public static void main(String[] args){
        //创建一个被观察者
        ConcreteSubject subject=new ConcreteSubject();
        //定义一个观察者
        Observer obs=new ConcreteObserver();
        //观察者观察被观察者
        subject.addObserver(obs);
        //观察者开始活动了
        subject.doSomething();
    }
}
```