一.观察者模式（Observer Pattern）的定义

定义对象间一种一对多的依赖关系，使得每当一个对象改变状态，所有依赖于它的对象都会得到通知并被自动更新。

二..观察者模式的四个角色

1. 抽象主题（Subject）角色

该角色又称为“被观察者”，可以增加和删除观察者对象。

2.抽象观察者（Observer）角色

该角色为所有的具体观察者定义一个接口，在得到主题的通知时更新自己。

3.具体主题（Concrete Subject）角色

该角色又称为“具体被观察者”，他将有关状态存入具体观察者对象，在具体主题的内部状态改变时，給所有登记过的观察者发出通知。

4. 具体观察者（Concrete Observer）角色

该角色实现抽象观察者所要求的更新接口，以便使自身的状态与主题的状态相协调。

三.观察者模式的优缺点

观察者模式中的优点

1.观察者和被观察者之间是抽象耦合。被观察者并不全部知道所有的具体观察者，他只知道它们有一个共同的接口。比如现实场景中，卢康知道自己的朋友圈信息被看到，并不知道所有看他信息的人。

2. 观察者模式支持广播通信。

观察者模式的缺点

1.如果一个主题有很多个直接或间接的观察者，则通知所有的观察者会花费很多的时间，而且它的开发和调试都比较复杂。

2.如果在主题之间有循环依赖的话，被观察者会触发它们进行循环调用，导致系统崩溃。

3.如果对观察者的通知是通过另外的线程进行异步投递的话，系统必须保证投递的顺序执行。

4.观察者模式没有提供相应的机制使观察者知道所观察的对象是怎么发生变化的，只知道结果变化了。

四.观察者模式的注意事项

观察者模式使用注意的事项：

1. 广播链问题。一个观察者既可以有双重身份，它既可以是观察者，也可以是被观察者，逻辑复杂的时候，可维护性就比较低了。所以一般一个观察者模式中最多出现一个双重身份的对象就够了，较易控制。

2.异步处理的问题。异步处理就要考虑线程安全和队列的问题。