# **ObserverPattern**

## 1.定义

观察者模式也叫发布订阅模式。

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，则所有依赖于它的对象都会得到通知并被自动更新。

## 代码链接

https://github.com/qiaojiuyuan/ObserverPattern

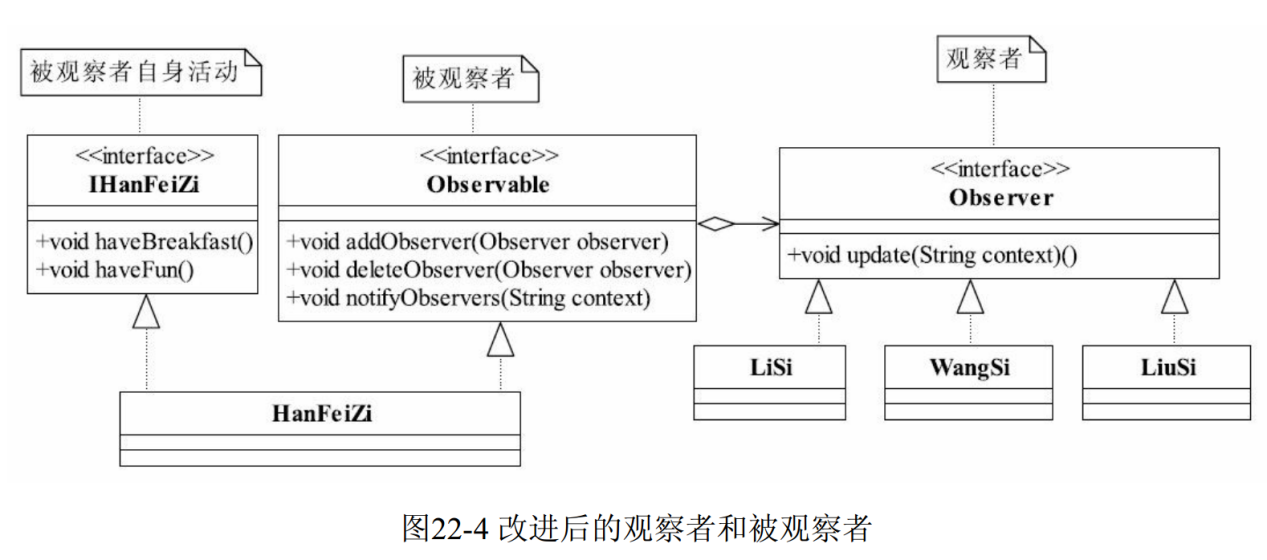
## 开发者介绍

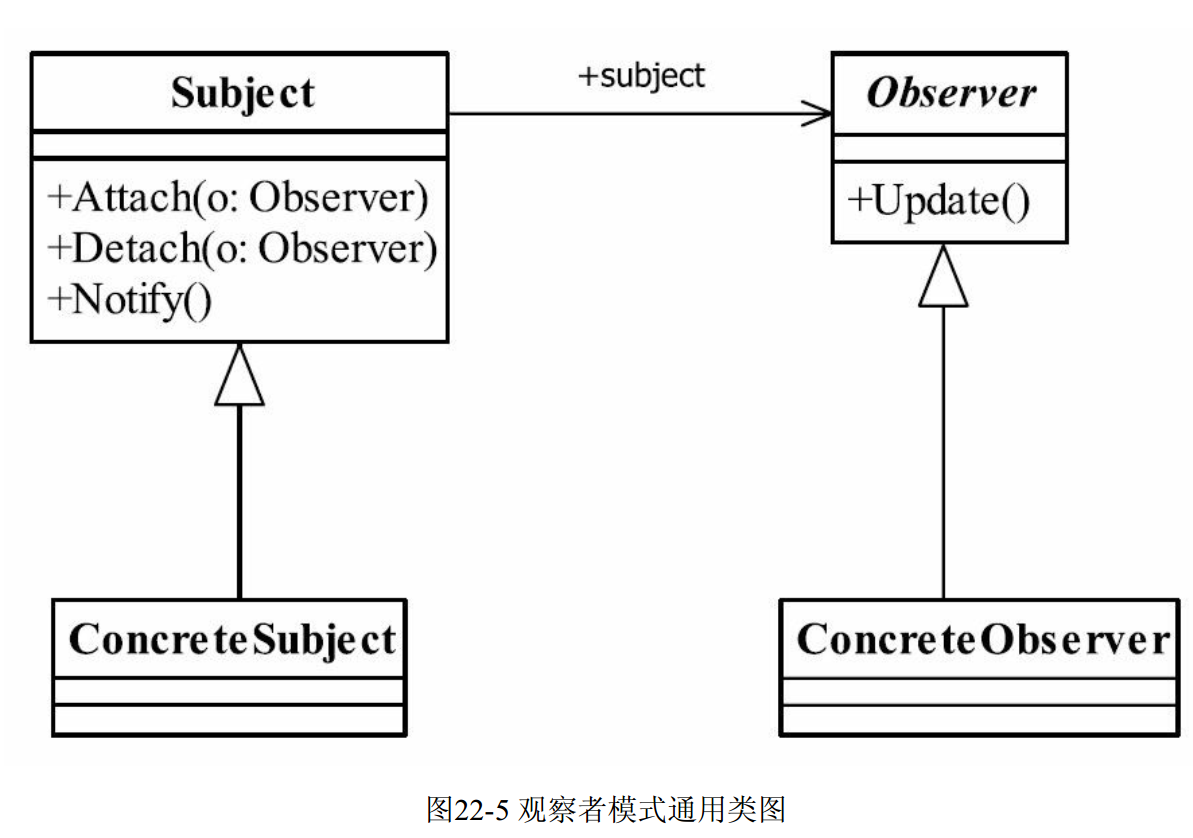
@qiaojiuyuan

## 任务

实现简单的观察者模式，建立一种对象与对象之间的依赖关系，一个对象发生改变时将自动通知其他对象，其他对象将相应做出反应。在此，发生改变的对象称为观察目标，而被通知的对象称为观察者，一个观察目标可以对应多个观察者，而且这些观察者之间没有相互联系，可以根据需要增加和删除观察者，使得系统更易于扩展。

## 5.模块关系





## 6.部分代码

1>**package** 观察者模式;

**public** **class** ClientMain {

**public** **static** **void** main(String[] args) {

Observer lisi = **new** LiSi();

Observer wangsi = **new** WangSi();

HanFeiZi hanFeiZi = **new** HanFeiZi();

hanFeiZi.addObserver(lisi);

hanFeiZi.addObserver(wangsi);

hanFeiZi.haveBreakfast();

hanFeiZi.haveFun();

}

### }

2>**package** 观察者模式;

/\*

\* 被观察者接口

\*/

**public** **interface** Observable {

/\*\*

\* 增加一个观察者

\* **@param** observable

\*/

**public** **void** addObserver(Observer observable);

/\*\*

\* 删除一个观察者

\* **@param** observable

\*/

**public** **void** deleteObserver(Observer observable);

/\*\*

\* 通知观察者

\* **@param** context

\*/

**public** **void** notifyObserviers(String context);

}

## 7.分析

此代码中韩非子是目标，李斯和王思是观察者，当韩非子的状态发生改变，李斯和王思可以监听到并且立即作出反应，即报告韩非子的状态，这是很典型的观察者模式。