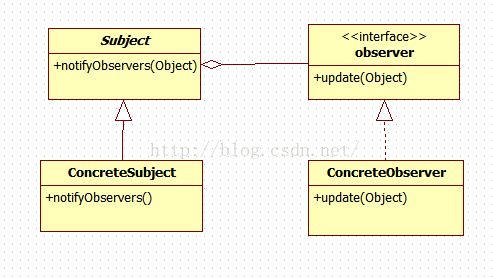
## 观察者模式

1. 定义与使用场景

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，则所有依赖于它的对象都会得到通知并被自动更新。

使用场景：关联行为的场景；事件多级触发场景；跨系统的消息交换场景，比如：消息队列、事件总线的处理机制。

1. UML类图



Subject：抽象主题，被观察对象的角色（Observable），抽象主题角色把所有观察者对象的引用保存到一个集合中，每个主题都可以有任意数量的观察者，抽象主题提供一个接口，可以增加和删除观察者对象。

ConcreteSubject：具体主题，该角色将有关状态存入具体观察者对象，在具体主题的内部状态发生改变时，给所有注册过的观察者发出通知，具体主题角色又叫做具体被观察者（ConcreteObservable）。

Observer：抽象观察者，该角色是观察者的抽象类，它定义了一个更新接口，使得在得到主题的更改通知时更新自己。

ConcreteObserver：具体的观察者，该角色实现抽象观察者角色所定义的更新接口，以便在主题的状态发生变化时更新自身状态。

1. 简单实现

Observer和Observable是JDK中的内置类型，可见观察者模式是非常重要的。

Observer：抽象观察者的角色

Observable：抽象主题的角色

1. 源码实现
2. 实战总结