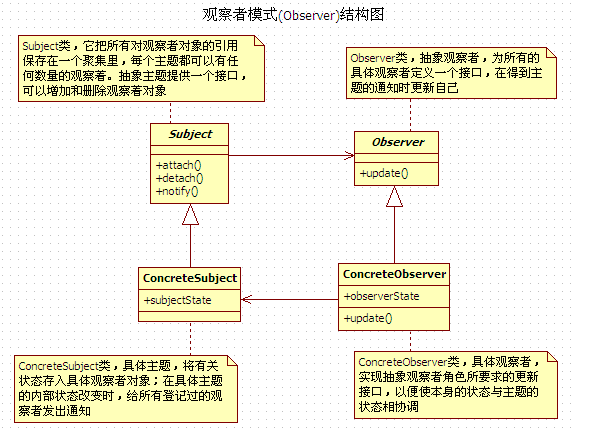
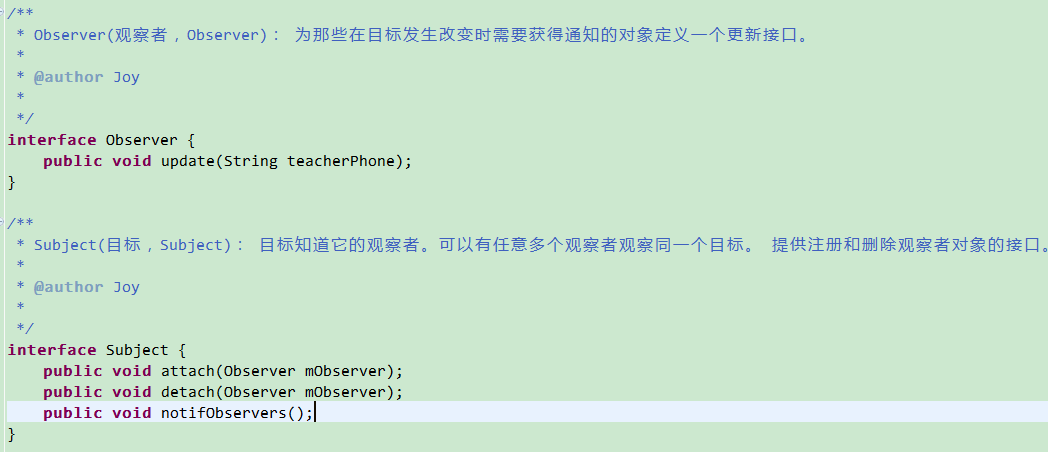
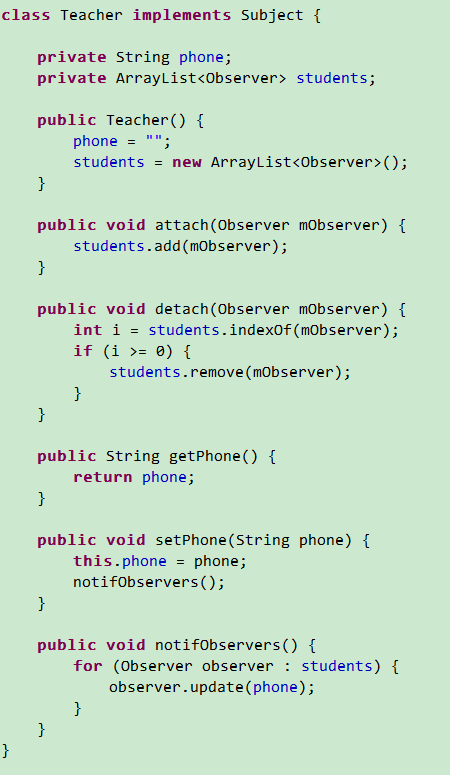
1. 观察者模式：定义了一种**一对多**的依赖关系，让多个观察者对象同时监听某一主题对象。这个主题对象在状态发生变化时，会通知所有观察者对象，使它们能够自动更新自己。观察者模式又叫发布-订阅(Publish/Subscribe)模式。
2. 类图

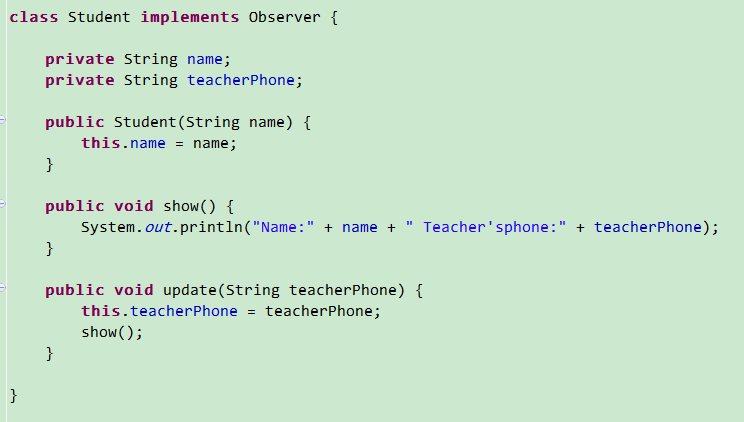


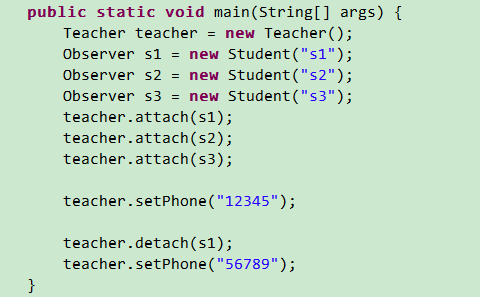
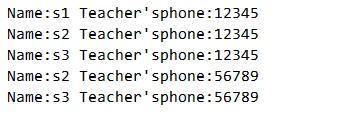
* 1. Subject类，它把所有对观察者对象的引用保存在一个聚集里，每个主题都可以有任何数量的观察者。抽象主题提供一个接口，可以增加和删除观察着对象。
  2. Observer类，抽象观察者，为所有的具体观察者定义一个接口，在得到主题的通知时更新自己。
  3. ConcreteSubject类，具体主题，将有关状态存入具体观察者对象；在具体主题的内部状态改变时，给所有登记过的观察者发出通知。
  4. ConcreteObserver类，具体观察者，实现抽象观察者角色所要求的更新接口，以便使本身的状态与主题的状态相协调。

1. 举例1







1. 可以使用JDK自带java.util.Observable和java.util.Observer实现观察者模式

