

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《研磨设计模式》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 本节课程概览

### n 学习观察者模式

#### 一：初识观察者模式

包括：定义、结构、参考实现

#### 二：体会观察者模式

包括：场景问题、使用模式的解决方案

#### 三：理解观察者模式

包括：认识观察者模式、推模型和拉模型、Java中的观察者模式、Swing中的观察者模式、观察者模式的优缺点

#### 四：思考观察者模式

包括：观察者模式的本质、何时选用、简单变形使用示例（区别对待观察者）

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

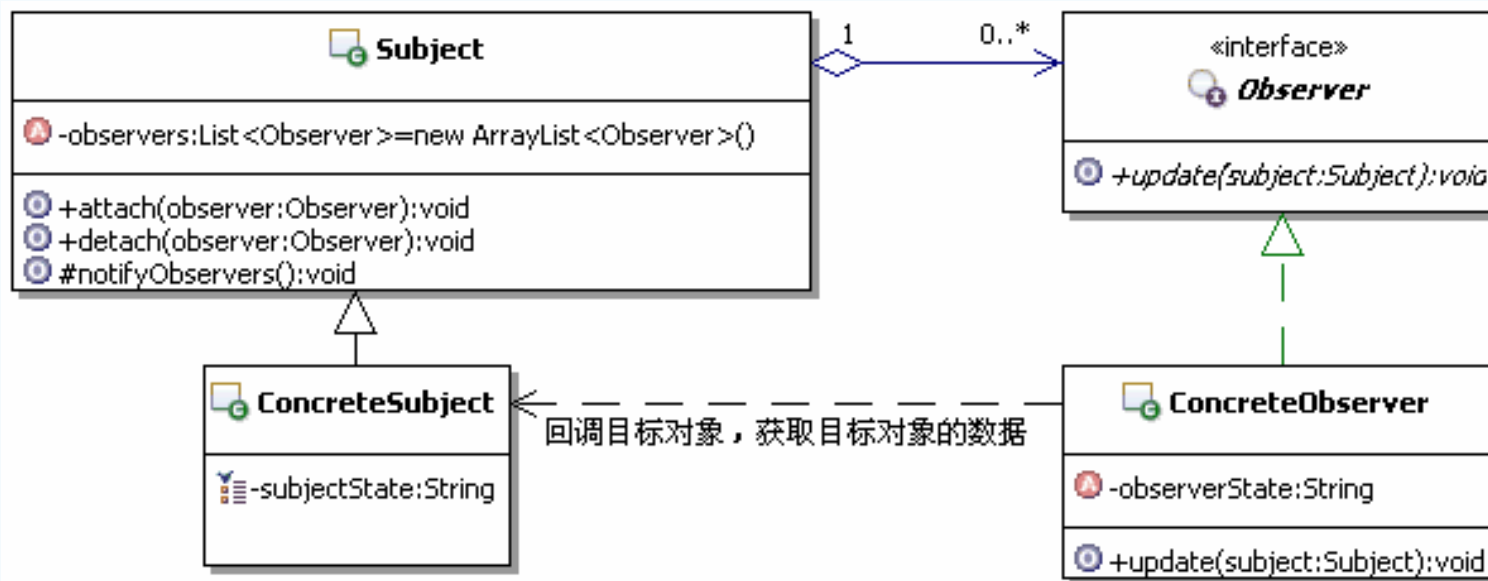
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识观察者模式

### n 定义

定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

### n 结构和说明



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识观察者模式

Subject:

目标对象，通常具有如下功能：

- (1) 一个目标可以被多个观察者观察
- (2) 目标提供对观察者注册和退订的维护
- (3) 当目标的状态发生变化时，目标负责通知所有注册的、有效的观察者

Observer:

定义观察者的接口，提供目标通知时对应的更新方法，这个更新方法进行相应的业务处理，可以在这个方法里面回调目标对象，以获取目标对象的数据。

ConcreteSubject:

具体的目标实现对象，用来维护目标状态，当目标对象的状态发生改变时，通知所有注册有效的观察者，让观察者执行相应的处理。

ConcreteObserver:

观察者的具体实现对象，用来接收目标的通知，并进行相应的后续处理，比如更新自身的状态以保持和目标的相应状态一致。

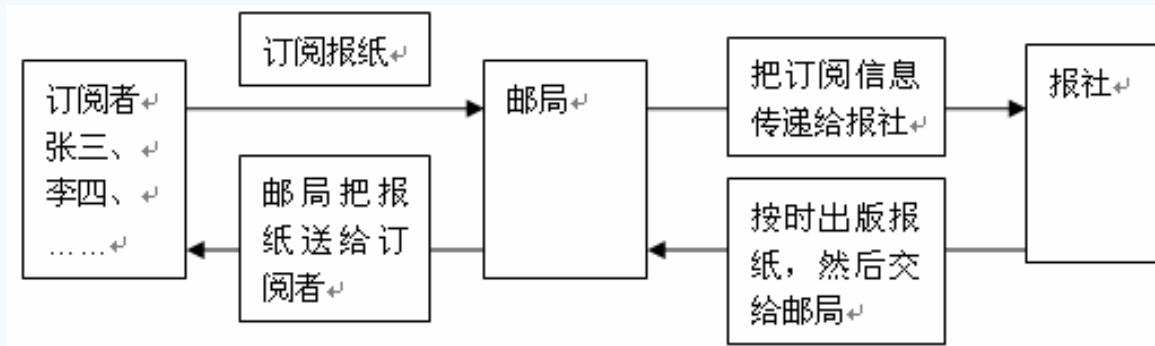
做最好的在线学习社区

网 址：<http://sishuok.com>

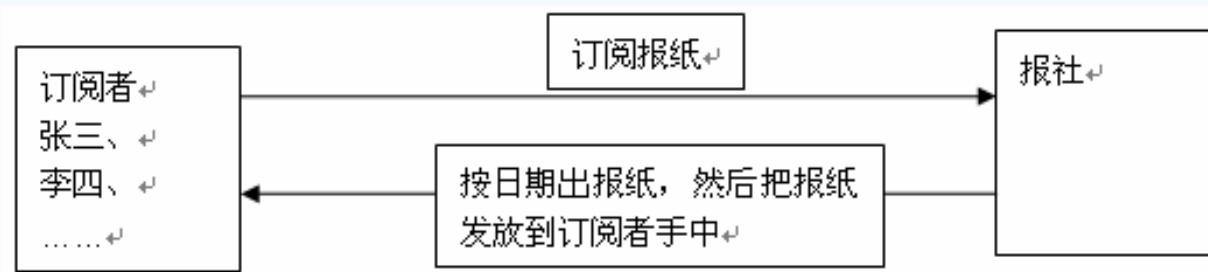
咨询QQ：2371651507

## 体会观察者模式

### n 订阅报纸的过程



在整个过程中，邮局只不过起到一个中转的作用，为了简单，我们去掉邮局，让订阅者直接和报社交互



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会观察者模式

### n 订阅报纸的问题

在上述过程中，订阅者在完成订阅后，最关心的问题就是何时能收到新出的报纸。幸好在现实生活中，报纸都是定期出版，这样发放到订阅者手中也基本上有一个大致的时间范围，差不多到时间了，订阅者就会看看邮箱，查收新的报纸。

要是报纸出版的时间不固定呢？

那订阅者就麻烦了，如果订阅者想要第一时间阅读到新报纸，恐怕只能天天守着邮箱了，这未免也太痛苦了吧。

继续引申一下，用类来描述上述的过程，描述如下：

订阅者类向出版者类订阅报纸，很明显不会只有一个订阅者订阅报纸，订阅者类可以有很多；当出版者类出版新报纸的时候，多个订阅者类如何知道呢？还有订阅者类如何得到新报纸的内容呢？

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 体会观察者模式

把上面的问题对比描述一下：

具体描述↵	对应的抽象描述↵
当报社有新报纸出版的时候↵	当出版者类的状态发生改变的时候↵
多个订阅报纸的人员↵	多个订阅者类↵
如何知道?↵	如何能得到通知?↵
订阅报纸的人员需要得到新报纸的内容，要看这些新内容↵	订阅者类会相应进行什么样的处理或改变↵

进一步抽象描述这个问题：当一个对象的状态发生改变的时候，如何让依赖于它的所有对象得到通知，并进行相应的处理呢？

做最好的在线学习社区

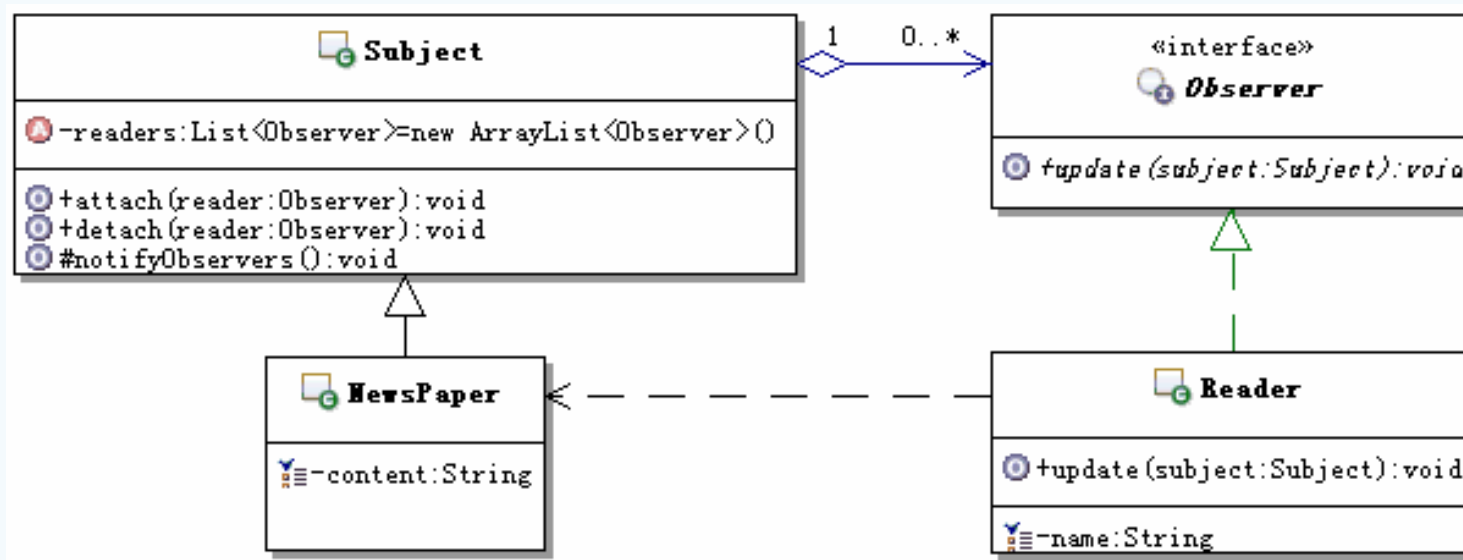
网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会观察者模式

### n 使用模式的解决方案



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 理解观察者模式

### n 认识观察者模式

#### 1: 目标和观察者之间的关系

按照模式的定义，目标和观察者之间是典型的一对多的关系。

但是要注意，如果观察者只有一个，也是可以的，这样就变相实现了目标和观察者之间一对一的关系，这也使得在处理一个对象的状态变化会影响到另一个对象的时候，也可以考虑使用观察者模式。

同样的，一个观察者也可以观察多个目标，如果观察者为多个目标定义的通知更新方法都是update方法的话，这会带来麻烦，因为需要接收多个目标的通知，如果是一个update的方法，那就需要在方法内部区分，到底这个更新的通知来自于哪一个目标，不同的目标有不同的后续操作。

一般情况下，观察者应该为不同的观察者目标，定义不同的回调方法，这样实现最简单，不需要在update方法内部进行区分。

## 理解观察者模式

### 2: 单向依赖

在观察者模式中，观察者和目标是单向依赖的，只有观察者依赖于目标，而目标是不会依赖于观察者的。

它们之间联系的主动权掌握在目标手中，只有目标知道什么时候需要通知观察者，在整个过程中，观察者始终是被动的，被动的等待目标的通知，等待目标传值给它。

对目标而言，所有的观察者都是一样的，目标会一视同仁的对待。当然也可以通过在目标里面进行控制，实现有区别对待观察者，比如某些状态变化，只需要通知部分观察者，但那是属于稍微变形的用法了，不属于标准的、原始的观察者模式了。

## 理解观察者模式

### 3: 基本的实现说明

具体的目标实现对象要能维护观察者的注册信息，最简单的实现方案就如同前面的例子那样，采用一个集合来保存观察者的注册信息。

具体的目标实现对象需要维护引起通知的状态，一般情况下是目标自身的状态，变形使用的情况下，也可以是别的对象的状态。

具体的观察者实现对象需要能接收目标的通知，能够接收目标传递的数据，或者是能够主动去获取目标的数据，并进行后续处理。

如果是一个观察者观察多个目标，那么在观察者的更新方法里面，需要去判断是来自哪一个目标的通知。一种简单的解决方案就是扩展update方法，比如在方法里面多传递一个参数进行区分等；还有一种更简单的方法，那就是干脆定义不同的回调方法。

## 理解观察者模式

### 4: 命名建议

- (1) 观察者模式又被称为发布-订阅模式
- (2) 目标接口的定义，建议在名称后面跟Subject
- (3) 观察者接口的定义，建议在名称后面跟Observer
- (4) 观察者接口的更新方法，建议名称为update，当然方法的参数可以根据需要定义，参数个数不限、参数类型不限

### 5: 触发通知的时机

一般情况下，是在完成了状态维护后触发，因为通知会传递数据，不能够先通知后改数据，这很容易出问题，会导致观察者和目标对象的状态不一致。

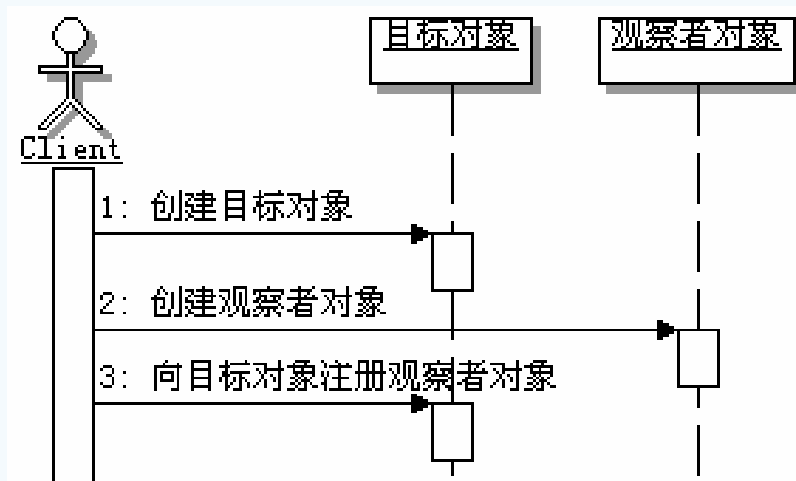
### 6: 相互观察

A对象的状态变化会引起C对象的联动操作，反过来，C对象的状态变化也会引起A对象的联动操作。对于出现这种状况，要特别小心处理，因为可能会出现死循环的情况。

## 理解观察者模式

### 7: 观察者模式的调用顺序示意图

在使用观察者模式时，会很明显的分成两个阶段，第一个阶段是准备阶段，也就是维护目标和观察者关系的阶段，这个阶段的调用顺序如图



做最好的在线学习社区

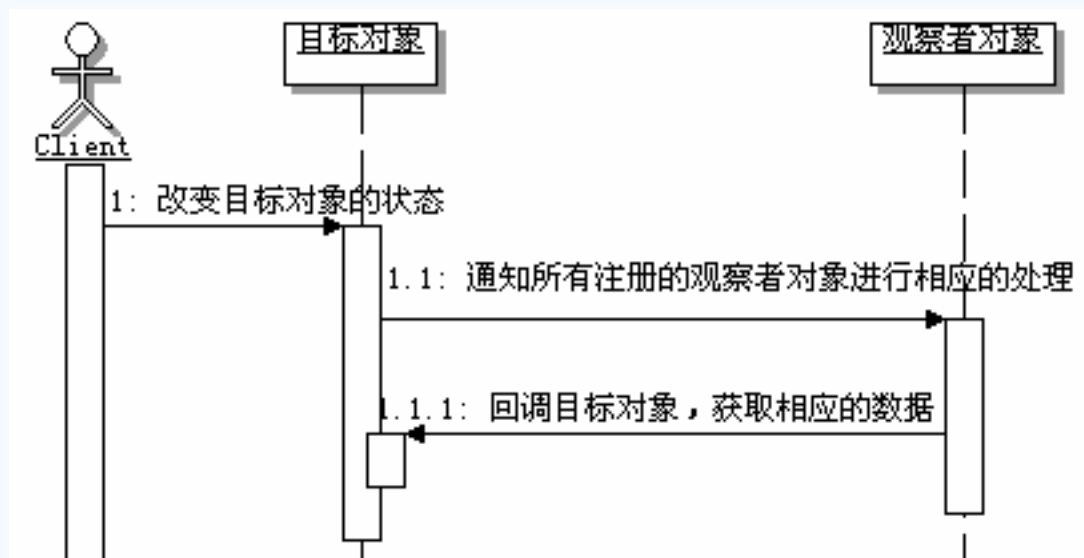
网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解观察者模式

接下来就是实际的运行阶段了，这个阶段的调用顺序如图



### 8: 通知的顺序

从理论上说，当目标对象的状态变化后通知所有观察者的时候，顺序是不确定的，因此观察者实现的功能，绝对不要依赖于通知的顺序，也就是说，多个观察者之间的功能是平行的，相互不应该有先后的依赖关系。

**做最好的在线学习社区**

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 理解观察者模式

### n 推模型和拉模型

推模型：目标对象主动向观察者推送目标的详细信息，不管观察者是否需要，推送的信息通常是目标对象的全部或部分数据，相当于是在广播通信。

拉模型：目标对象在通知观察者的时候，只传递少量信息，如果观察者需要更具体的信息，由观察者主动到目标对象中获取，相当于是观察者从目标对象中拉数据。

一般这种模型的实现中，会把目标对象自身通过update方法传递给观察者，这样在观察者需要获取数据的时候，就可以通过这个引用来获取了

## 理解观察者模式

### n 关于两种模型的比较

两种实现模型，在开发的时候，究竟应该使用哪一种，还是应该具体问题具体分析。这里，只是把两种模型进行一个简单的比较。

- 1: 推模型是假定目标对象知道观察者需要的数据；而拉模型是目标对象不知道观察者具体需要什么数据，没有办法的情况下，干脆把自身传给观察者，让观察者自己去按需取值。
- 2: 推模型可能会使得观察者对象难以复用，因为观察者定义的update方法是按需而定义的，可能无法兼顾没有考虑到的使用情况。这就意味着出现新情况的时候，就可能需要提供新的update方法，或者是干脆重新实现观察者。

而拉模型就不会造成这样的情况，因为拉模型下，update方法的参数是目标对象本身，这基本上是目标对象能传递的最大数据集合了，基本上可以适应各种情况的需要。

## 理解观察者模式

### n Java中的观察者模式

在java.util包里面有一个类Observable，它实现了大部分我们需要的目标的功能；还有一个接口Observer，它里面定义了update的方法，就是观察者的接口。

### n Swing中的观察者模式

Swing中到处都是观察者模式的身影，比如大家熟悉的事件处理，就是典型的观察者模式的应用。（说明一下：早期的Swing事件处理用的是职责链）

Swing组件是被观察的目标，而每个实现监听器的类就是观察者，监听器的接口就是观察者的接口，在调用addXXXListener方法的时候就相当于注册观察者。

当组件被点击，状态发生改变的时候，就会产生相应的通知，会调用注册的观察者的方法，就是我们所实现的监听器的方法。

从这里还可以学一招：如何处理一个观察者观察多个目标对象

## 理解观察者模式

### n 观察者模式的优缺点

- 1: 观察者模式实现了观察者和目标之间的抽象耦合
- 2: 观察者模式实现了动态联动
- 3: 观察者模式支持广播通信
- 4: 观察者模式可能会引起无谓的操作

## 思考观察者模式

### n 观察者模式的本质

观察者模式的本质是：触发联动

### n 何时选用观察者模式

- 1: 当一个抽象模型有两个方面，其中一个方面的操作依赖于另一个方面的状态变化，那么就可以选用观察者模式。
- 2: 如果在更改一个对象的时候，需要同时连带改变其它的对象，而且不知道究竟应该有多少对象需要被连带改变，这种情况可以选用观察者模式，被更改的那一个对象很明显就相当于是目标对象，而需要连带修改的多个其它对象，就作为多个观察者对象了。
- 3: 当一个对象必须通知其它的对象，但是你又希望这个对象和其它被它通知的对象是松散耦合的，也就是说这个对象其实不想知道具体被通知的对象，这种情况可以选用观察者模式，这个对象就相当于是目标对象，而被它通知的对象就是观察者对象了。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 理解观察者模式

### n 简单变形示例——区别对待观察者

#### 1: 范例需求

这是一个实际系统的简化需求：在一个水质监测系统中有这样一个功能，当水中的杂质为正常的时候，只是通知监测人员做记录；当为轻度污染的时候，除了通知监测人员做记录外，还要通知预警人员，判断是否需要预警；当为中度或者高度污染的时候，除了通知监测人员做记录外，还要通知预警人员，判断是否需要预警，同时还要通知监测部门领导做相应的处理。

#### 2: 解决思路和范例代码

分析上述需求就会发现，对于水质污染这件事情，有可能会涉及到监测员、预警人员、监测部门领导，根据不同的水质污染情况涉及到不同的人员，也就是说，监测员、预警人员、监测部门领导三者是平行的，职责都是处理水质污染，但是处理的范围不一样。

因此很容易套用观察者模式，如果把水质污染的记录当作被观察的目标的话，那么监测员、预警人员和监测部门领导就都是观察者了。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507



## 理解观察者模式

前面学过的观察者模式，当目标通知观察者的时候是全部都通知，但是现在这个需求是不同的情况来让不同的人处理，怎么办呢？

解决的方式通常有两种，一种是目标可以通知，但是观察者不做任何操作；另外一种是在目标里面进行判断，干脆就不通知了。两种实现方式各有千秋，这里选择后面一种方式来示例，这种方式能够统一逻辑控制，并进行观察者的统一分派，有利于业务控制和今后的扩展。