

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 学习命令模式

一：初识命令模式

包括：定义、结构、参考实现

二：体会命令模式

包括：场景问题、使用模式的解决方案

三：理解命令模式

包括：认识命令模式、参数化配置、可撤销的操作、
宏命令、队列请求、日志请求、命令模式的优缺点

四：思考命令模式

包括：命令模式的本质、何时选用、退化的命令模式

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

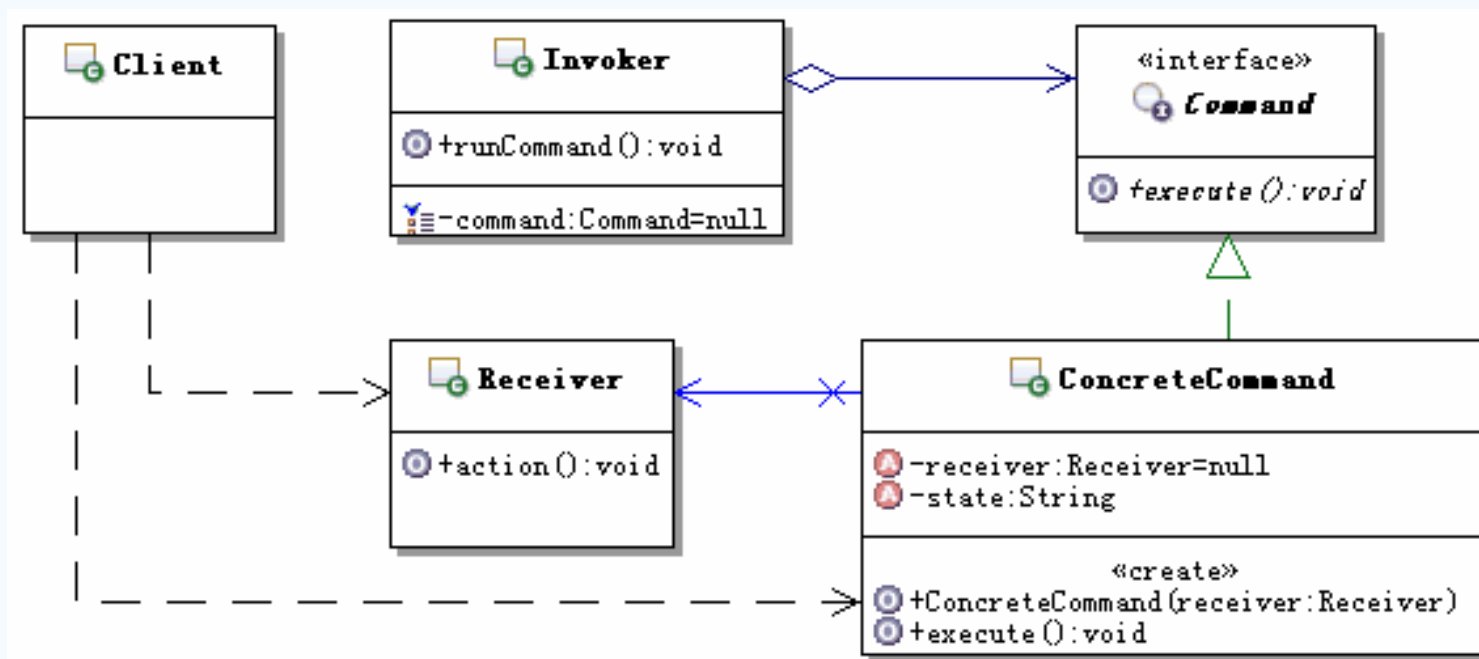
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

初识命令模式

n 定义

将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。

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

初识命令模式

Command: 定义命令的接口，声明执行的方法。

ConcreteCommand:

命令接口实现对象，是“虚”的实现；通常会持有接收者，并调用接收者的功能来完成命令要执行的操作。

Receiver:

接收者，真正执行命令的对象。任何类都可能成为一个接收者，只要它能够实现命令要求实现的相应功能。

Invoker:

要求命令对象执行请求，通常会持有命令对象，可以持有很多的命令对象。这个是客户端真正触发命令并要求命令执行相应操作的地方，也就是说相当于使用命令对象的入口。

Client: 创建具体的命令对象，并且设置命令对象的接收者。注意这个不是我们常规意义上的客户端，而是在组装命令对象和接收者，或许，把这个Client称为装配者会更好理解，因为真正使用命令的客户端是从Invoker来触发执行。

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

体会命令模式

n 如何开机

当我们按下启动按钮过后呢？谁来处理？如何处理？都经历了怎样的过程，才让电脑真正的启动起来，供我们使用呢？

如果现在要求用软件把开机的过程表现出来，该如何实现？

首先把开机的过程总结一下，主要就这么几个步骤：首先加载电源，然后是设备检查，再然后是装载系统，最后电脑就正常启动了。可是谁来完成这些过程？如何完成？

不能让使用电脑的客户——就是我们来做这些工作吧，真正完成这些工作的是主板，那么客户和主板如何发生联系呢？现实中，是用连接线把按钮连接到主板上的，这样当客户按下按钮的时候，就相当于发命令给主板，让主板去完成后续的工作。

另外，从客户的角度来看，开机就是按下按钮，不管什么样的主板都是一样的，也就是说，客户只管发出命令，谁接收命令，谁实现命令，如何实现，客户是不关心的。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

体会命令模式

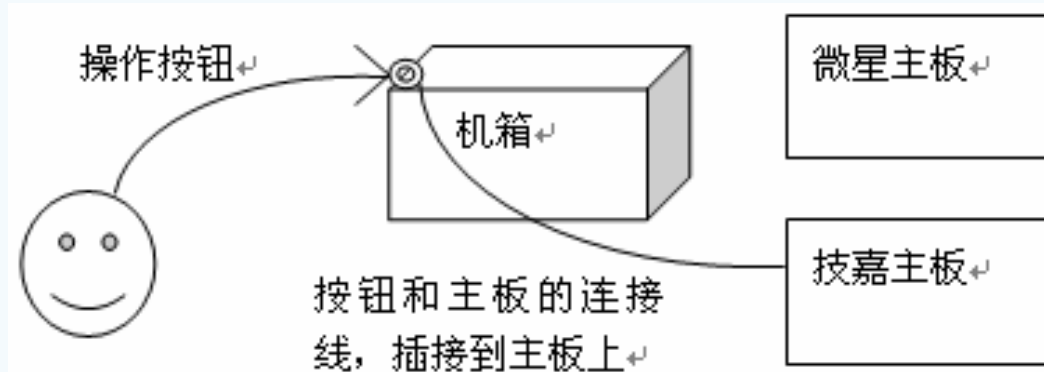
n 有何问题

把上面的问题抽象描述一下：客户端只是想要发出命令或者请求，不关心请求的真正接收者是谁，也不关心具体如何实现，而且同一个请求的动作可以有不同的请求内容，当然具体的处理功能也不一样，请问该怎么实现？

n 使用模式的解决方案

n 使用模式解决问题的思路

看看实际的电脑是如何处理上面描述的这个问题的



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

体会命令模式

n 要用程序来解决上面提出的问题，一种自然的方案就是来模拟上述解决思路

在命令模式中，会定义一个命令的接口，用来约束所有的命令对象，每个命令实现对象是对客户端某个请求的封装，对应于机箱上的按钮，一个机箱上可以有很多按钮，也就相当于会有多个具体的命令实现对象。

在命令模式中，命令对象并不知道如何处理命令，会有相应的接收者对象来真正执行命令。就像电脑的例子，机箱上的按钮并不知道如何处理功能，而是把这个请求转发给主板，由主板来执行真正的功能，这个主板就相当于命令模式的接收者。

在命令模式中，命令对象和接收者对象的关系，并不是与生俱来的，需要有一个装配的过程，命令模式中的Client对象就来实现这样的功能。这就相当于在电脑的例子中，有了机箱上的按钮，也有了主板，还需要有一个连接线把这个按钮连接到主板上才行。

命令模式还会提供一个Invoker对象来持有命令对象，就像电脑的例子，机箱上会有多个按钮，这个机箱就相当于命令模式的Invoker对象。这样一来，命令模式的客户端就可以通过Invoker来触发并要求执行相应的命令了，这也相当于真正的客户是按下机箱上的按钮来操作电脑一样。

做最好的在线学习社区

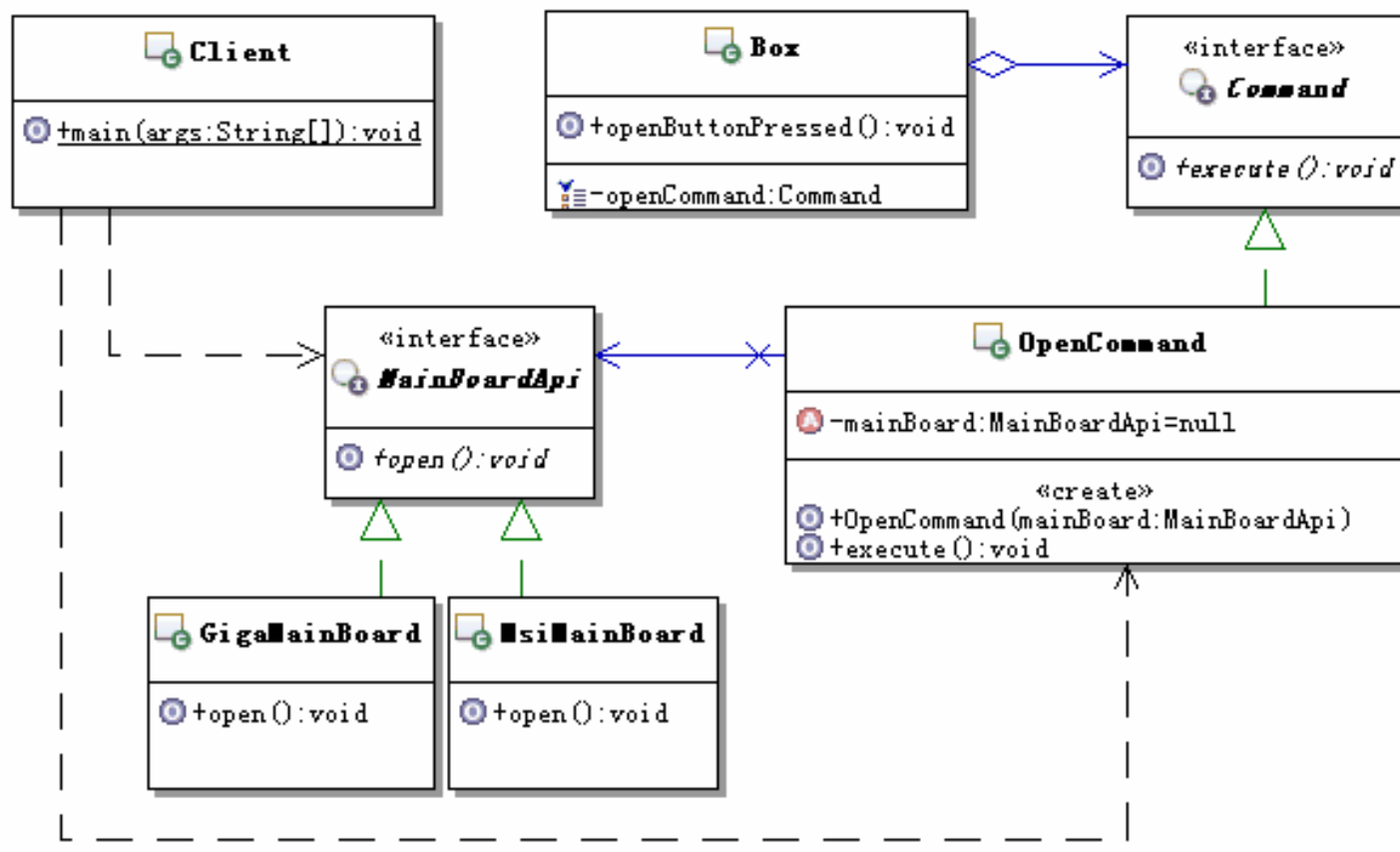
网 址：<http://sishuok.com>

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

体会命令模式

n 使用模式的解决方案的类图



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

理解命令模式

n 认识命令模式

1: 命令模式的关键

命令模式的关键之处就是把请求封装成为对象，也就是命令对象，并定义了统一的执行操作的接口，这个命令对象可以被存储、转发、记录、处理、撤销等，整个命令模式都是围绕这个对象在进行。

2: 命令模式的组装和调用

在命令模式中经常会有一个命令的组装者，用它来维护命令的“虚”实现和真实实现之间的关系。如果是超级智能的命令，也就是说命令对象自己完全实现好了，不需要接收者，那就是命令模式的退化，不需要接收者，自然也不需要组装者了。

而真正的用户就是具体化请求的内容，然后提交请求进行触发就好了。真正的用户会通过Invoker来触发命令。

在实际开发中，Client和Invoker可以融合在一起，由客户在使用命令模式时，先进行命令对象和接收者的组装，组装完成后，再调用命令执行请求。

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

理解命令模式

3: 命令模式的接收者

接收者可以是任意的类，对它没有什么特殊要求，这个对象知道如何真正执行命令的操作，执行时是从command的实现类里面转调过来。

一个接收者对象可以处理多个命令，接收者和命令之间没有约定的对应关系。接收者提供的方法个数、名称、功能和命令中的可以不一样，只要能够通过调用接收者的方法来实现命令对应的功能就可以了。

4: 智能命令

在标准的命令模式里面，命令的实现类是没有真正实现命令要求的功能的，真正执行命令的功能的是接收者。

如果命令的实现对象比较智能，它自己就能真实地实现命令要求的功能，而不再需要调用接收者，那么这种情况就称为智能命令。

也可以有半智能的命令，命令对象知道部分实现，其它的还是需要调用接收者来完成，也就是说命令的功能由命令对象和接收者共同来完成。

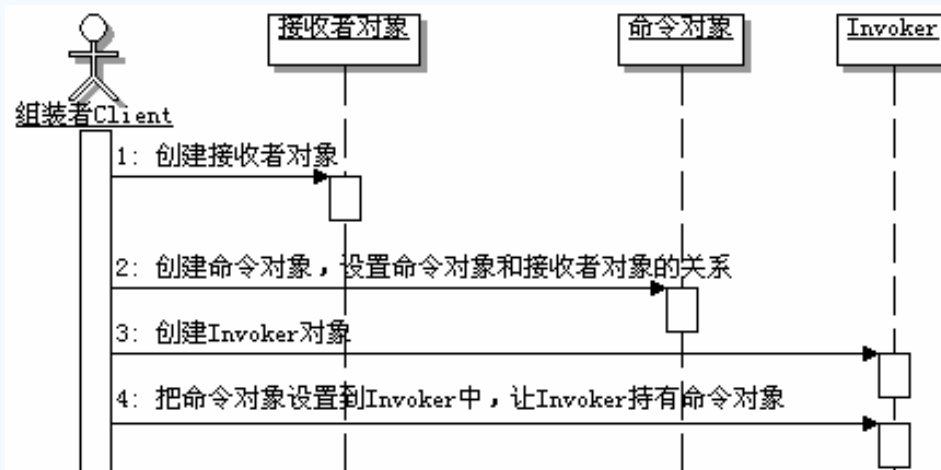
理解命令模式

5: 发起请求的对象和真正实现的对象是解耦的

请求究竟由谁处理，如何处理，发起请求的对象是不知道的，也就是发起请求的对象和真正实现的对象是解耦的。发起请求的对象只管发出命令，其它的就不管了。

6: 命令模式的调用顺序示意图

使用命令模式的过程分成两个阶段，一个阶段是组装命令对象和接收者对象的过程，另外一个阶段是触发调用Invoker，来让命令真正执行的过程。



做最好的在线学习社区

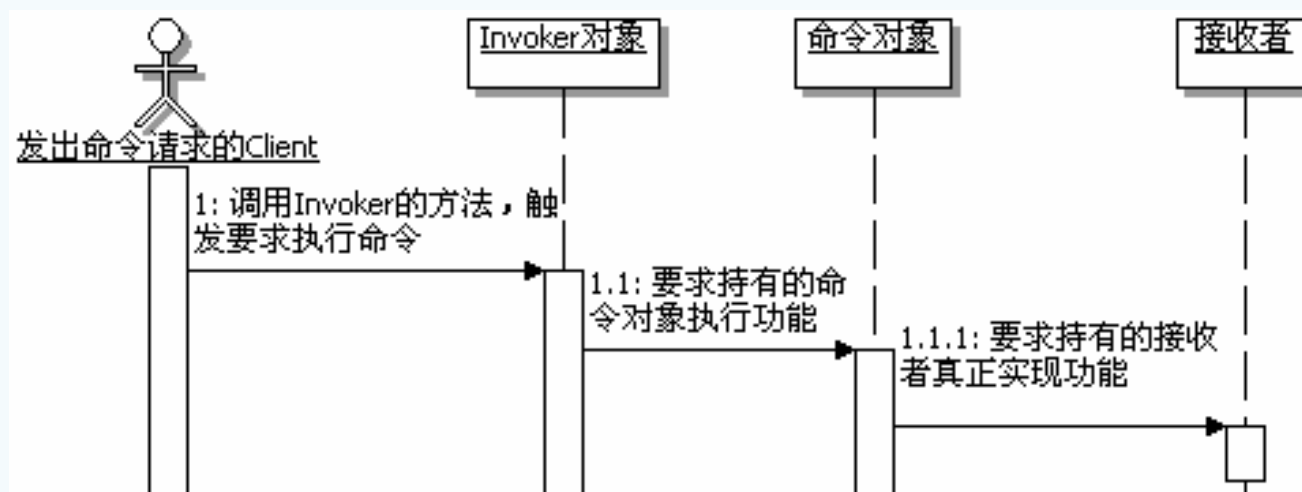
网 址: <http://sishuok.com>

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

理解命令模式

再看看真正执行命令时的调用顺序示意图



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

理解命令模式

n 参数化配置

所谓命令模式的参数化配置，指的是：可以用不同的命令对象，去参数化配置客户的请求。

像前面描述的那样：客户按下一个按钮，到底是开机还是重启，那要看参数化配置的是哪一个具体的按钮对象，如果参数化的是开机的命令对象，那就执行开机的功能，如果参数化的是重启的命令对象，那就执行重启的功能。虽然按下的是同一个按钮，相当于是同一个请求，但是为请求配置不同的按钮对象，那就会执行不同的功能。

理解命令模式

n 可撤销的操作

可撤销操作的意思就是：放弃该操作，回到未执行该操作前的状态。这是一个非常重要的功能，几乎所有GUI应用里都有撤消操作的功能。GUI的菜单是命令模式最典型的应用之一，所以你总是能在菜单上找到撤销这样的菜单项。

既然这么常用，那该如何实现呢？

有两种基本的思路来实现可撤销的操作，一种是补偿式，又称反操作式：比如被撤销的操作是加的功能，那撤消的实现就变成减的功能；同理被撤销的操作是打开的功能，那么撤销的实现就变成关闭的功能。

另外一种方式是存储恢复式，意思就是把操作前的状态记录下来，然后要撤销操作的时候就直接恢复回去就可以了。

这里先讲第一种方式，就是补偿式或者反操作式，第二种方式放到备忘录模式中去讲解

理解命令模式

n 范例需求

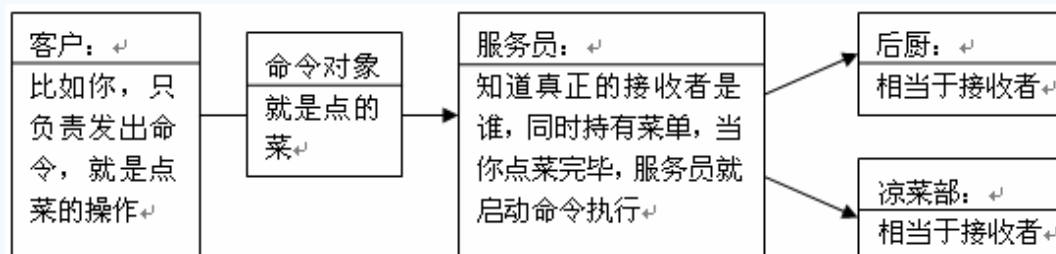
考虑一个计算器的功能，最简单的那种，只能实现加减法运算，现在要让这个计算器支持可撤销的操作。

理解命令模式

n 宏命令

简单点说就是包含多个命令的命令，是一个命令的组合。

举个例子来说吧，设想一下你去饭店吃饭的过程：



1: 宏命令在哪里?

现实中是当你你点完菜，说“点完了”的时候，服务员才会启动命令的执行，请注意，这个时候执行的就不是一个命令了，而是执行一堆命令。

描述这一堆命令的就是菜单，如果把菜单也抽象成为一个命令，就相当于一个大的命令，当客户说“点完了”的时候，就相当于触发这个大的命令，意思就是执行菜单这个命令就可以了，这个菜单命令包含多个命令对象，一个命令对象就相当于一道菜。那么这个菜单就相当于我们说的宏命令。

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

理解命令模式

2：如何实现宏命令

宏命令从本质上讲类似于一个命令，基本上把它当命令对象进行处理。但是它跟普通的命令对象又有些不一样，就是宏命令包含有多个普通的命令对象，执行一个宏命令，简单点说，就是执行宏命令里面所包含的所有命令对象，有点打包执行的意味。

理解命令模式

n 队列请求

所谓队列请求，就是对命令对象进行排队，组成工作队列，然后依次取出命令对象来执行。多用多线程或者线程池来进行命令队列的处理，当然也可以不用多线程，就是一个线程，一个命令一个命令的循环处理，就是慢点。

n 日志请求

所谓日志请求，就是把请求的历史记录保存下来，一般是采用永久存储的方式。如果运行请求的过程中，系统崩溃了，那么在系统再次运行时，就可以从保存的历史记录里面获取日志请求，并重新执行命令。

日志请求的实现有两种方案，一种就是直接使用Java中的序列化方法，另外一种就是在命令对象里面添加上存储和装载的方法，其实就是让命令对象自己实现类似序列化的功能。当然要简单就直接使用Java中的序列化。

理解命令模式

- n 命令模式的优缺点
 - 1: 更松散的耦合
 - 2: 更动态的控制
 - 3: 能很自然的复合命令
 - 4: 更好的扩展性

思考命令模式

n 命令模式的本质

命令模式的本质是：**封装请求**

n 何时选用命令模式

- 1: 如果需要抽象出需要执行的动作，并参数化这些对象，可以选用命令模式，把这些需要执行的动作抽象成为命令，然后实现命令的参数化配置
- 2: 如果需要在不同的时刻指定、排列和执行请求，可以选用命令模式，把这些请求封装成为命令对象，然后实现把请求队列化
- 3: 如果需要在支持取消操作，可以选用命令模式，通过管理命令对象，能很容易的实现命令的恢复和重做的功能
- 4: 如果需要在支持当系统崩溃时，能把对系统的操作功能重新执行一遍，可以选用命令模式，把这些操作功能的请求封装成命令对象，然后实现日志命令，就可以在系统恢复回来后，通过日志获取命令列表，从而重新执行一遍功能
- 5: 在需要事务的系统中，可以选用命令模式，命令模式提供了对事务进行建模的方法，命令模式有一个别名就是Transaction。

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

理解命令模式

n 退化的命令模式

前面讲到了智能命令，如果命令的实现对象超级智能，实现了命令所要求的功能，那么就不需要接收者了，既然没有了接收者，那么也就不需要组装者了。