

21、命令模式（Command Pattern）

一种行为型模式

-----钱国红、袁锋、何丽慧

COMMAND—俺有一个MM家里管得特别严，没法见面，只好借助于她弟弟在我们俩之间传送信息，她对我有什么指示，就写一张纸条让她弟弟带给我。这不，她弟弟又传送过来一个COMMAND，为了感谢他，我请他吃了碗杂酱面，哪知道他说：“我同时给我姐姐三个男朋友送COMMAND，就数你最小气，才请我吃面。”，

命令模式：命令模式把一个请求或者操作封装到一个对象中。命令模式把发出命令的责任和执行命令的责任分割开，委派给不同的对象。命令模式允许请求的一方和发送的一方独立开来，使得请求的一方不必知道接收请求的一方的接口，更不必知道请求是怎么被接收，以及操作是否执行，何时被执行以及是怎么被执行的。系统支持命令的撤消。

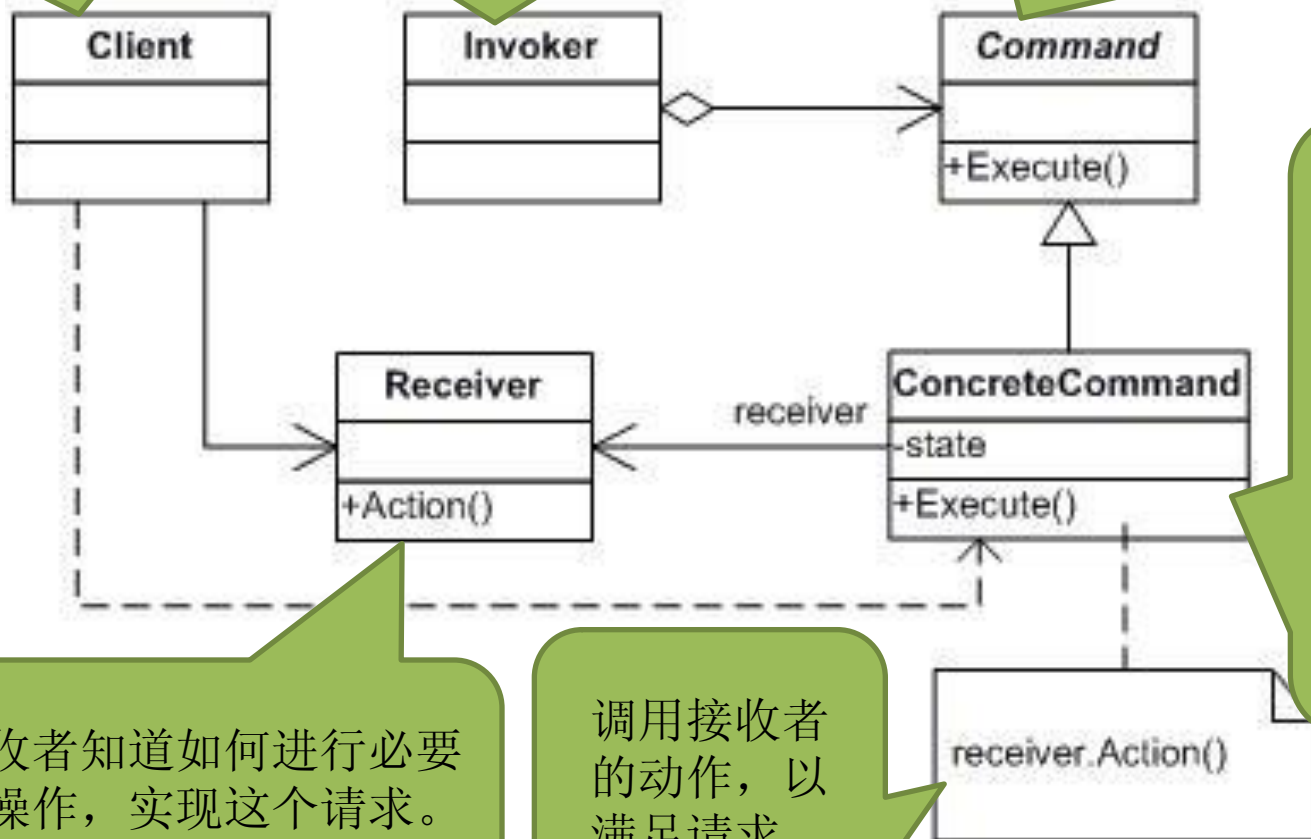
命令模式的定义

- 命令模式将“请求”封装成对象，以便使用不同的请求、队列或者日志来参数化其他对象。命令模式也支持撤销操作。
- 核心：实现了请求调用者和请求接收者之间的解耦。

创建命令，
设置其接收者。

持有一个命令对象，并
在某个时间点调用命令
对象的execute（）方
法，将请求付诸实行。

为所有命令申明了一个接口。
调用命令对象的execute（）方
法，就可以让接收者进行相关
的动作。这个接口也具备一个
undo（）方法。



定义了动作和
接收者之间的
绑定关系。调
用者只要调用
execute（）
就可以发出请
求，然后用它
调用接收者的
一个或多个动
作。

接收者知道如何进行必要
的操作，实现这个请求。
任何类都可以当接收者。

调用接收者的
动作，以
满足请求。

命令模式的适用场合

- 1. 使用命令模式作为“Call Back”在面向对象系统中的替代。
- 2. 需要在不同的时间指定请求、将请求排队。
- 3. 系统需要支持命令的撤消(undo)。
- 4. 如果一个系统要将系统中所有的数据更新到日志里，以便在系统崩溃时，可以根据日志里读回所有的数据更新命令，重新调用Execute()方法一条一条执行这些命令，从而恢复系统在崩溃前所做的数据更新。
- 5. 一个系统需要支持交易。

命令模式的优缺点

优点：

- 命令模式把请求一个操作的对象和知道怎么执行一个操作的对象分割开。
- 命令类和其他任何别的类一样，可以修改和推广。
- 可以把命令对象聚合在一起合成为合成命令。
- 由于增加新的具体命令类不影响其他的类，因此增加新的具体命令类很容易。

缺点：

- 适用命令模式会导致某些系统有过多的具体命令类。

命令模式总结

- 耦合与变化：

耦合是软件不能抵御变化灾难的根本性原因。不仅实体对象与实体对象之间存在耦合关系，实体对象与行为操作之间也存在耦合关系。
- 动机(Motivate)：

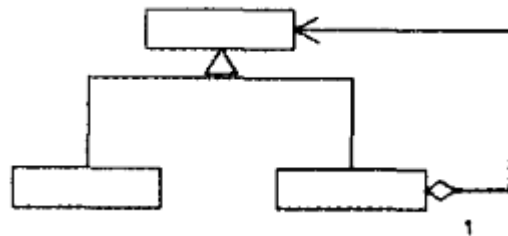
在软件系统中，“行为请求者”与“行为实现者”通常呈现一种“紧耦合”。但在某些场合，比如要对行为进行“记录、撤销/重做、事务”等处理，这种无法抵御变化的紧耦合是不合适的。
- 意图(Intent)：

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，以及支持可撤消的操作。

命令模式和其他模式的关系

- 与合成模式的关系：

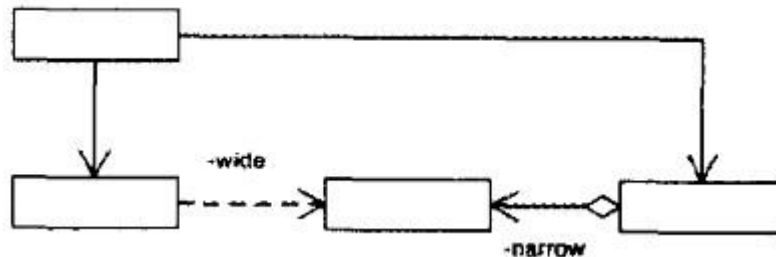
合成模式可以应用的命令类的合成上，从几个具体命令类合成宏命令类。合成模式的简略类图如下所示：



命令模式和其他模式的关系

- 与备忘录模式的关系：

如果命令需要撤销和恢复功能的话，备忘录模式可以用来存储关于命令的效果状态信息，以便在撤销命令时可以撤销命令的效果。备忘录模式的简略类图如下所示：



命令模式和其他模式的关系

- 与原始模型模式的关系：

如果命令类带有clone（）方法的话，命令就可以被复制。原始模型模式的简略类图如下图所示：

