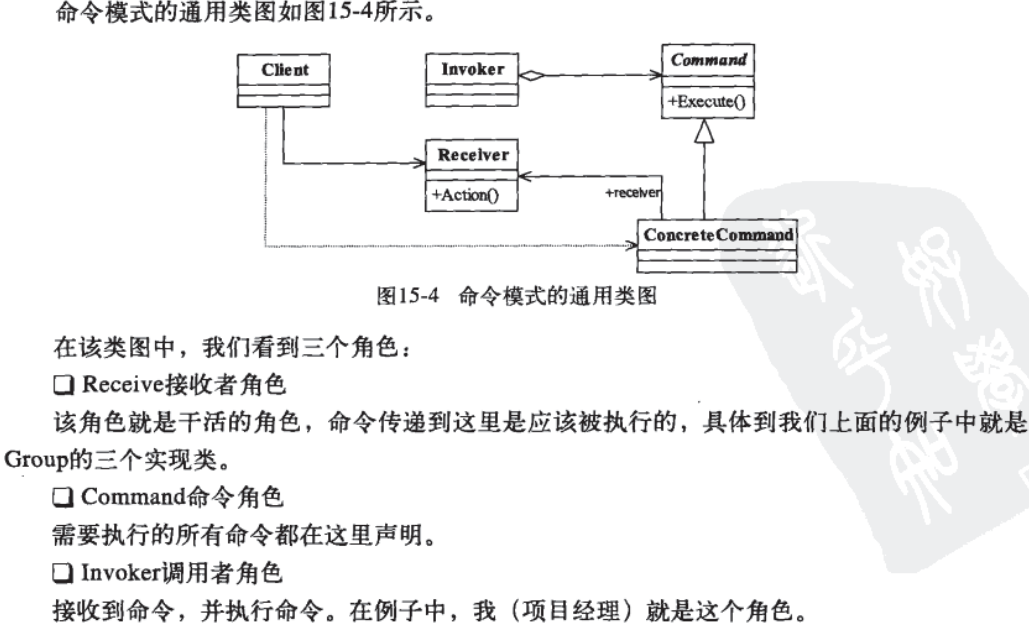
# 命令模式

1. 定义

Encapsulate a request as an object,there by letting you parameterize clients with different requests,queue or logs requests,and support undoable operations.(将一个请求封装成一个对象，从而让你使用不同的请求把客户端参数化，对请求排队或者日志记录请求，可以提供命令的恢复和撤销功能)

1. 类图



1. 模式的应用
2. 模式的优点
3. 类间解耦：调用者角色与接受者角色之间没有依赖关系，调用者实现功能时只需调用Command抽象类的execute方法即可，不需要了解到底哪几个接受者执行。
4. 可扩展性：Command子类可以非常容易的扩展，而调用者Invoker和高层次的模块Client不产生严重的耦合。
5. 命令模式结合其他模式会更加优秀：**命令模式**可以结合**责任链模式**，实现命令簇解析任务；结合模板方法模式，则可以减少Command子类的膨胀问题
6. 模式的缺点

Command子类会因为命令数的增多，膨胀的非常大，通常一个命令对应一个Command子类。

1. 实际应用场景

认为是命令的场景。

1. GUI开发，一个按钮的点击是一个命令
2. 模拟DOS命令；
3. 触发-反馈机制的处理。