

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

C#面向对象设计模式纵横谈

15. Command 命令（行为型模式）

李建忠

jianzhong.lee@gmail.com

设计模式论坛:

forum.softcompass.com

上海祝成科技 高级培训讲师

www.softcompass.com

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 (Motivation)

在软件构建过程中，“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——比如需要对行为进行“记录、撤销/重做（undo/redo）、事务”等处理，这种无法抵御变化的紧耦合是不合适的。

在这种情况下，如何将“行为请求者”与“行为实现者”解耦？将一组行为抽象为对象，可以实现二者之间的松耦合。

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，以及支持可撤销的操作。

——《设计模式》GoF

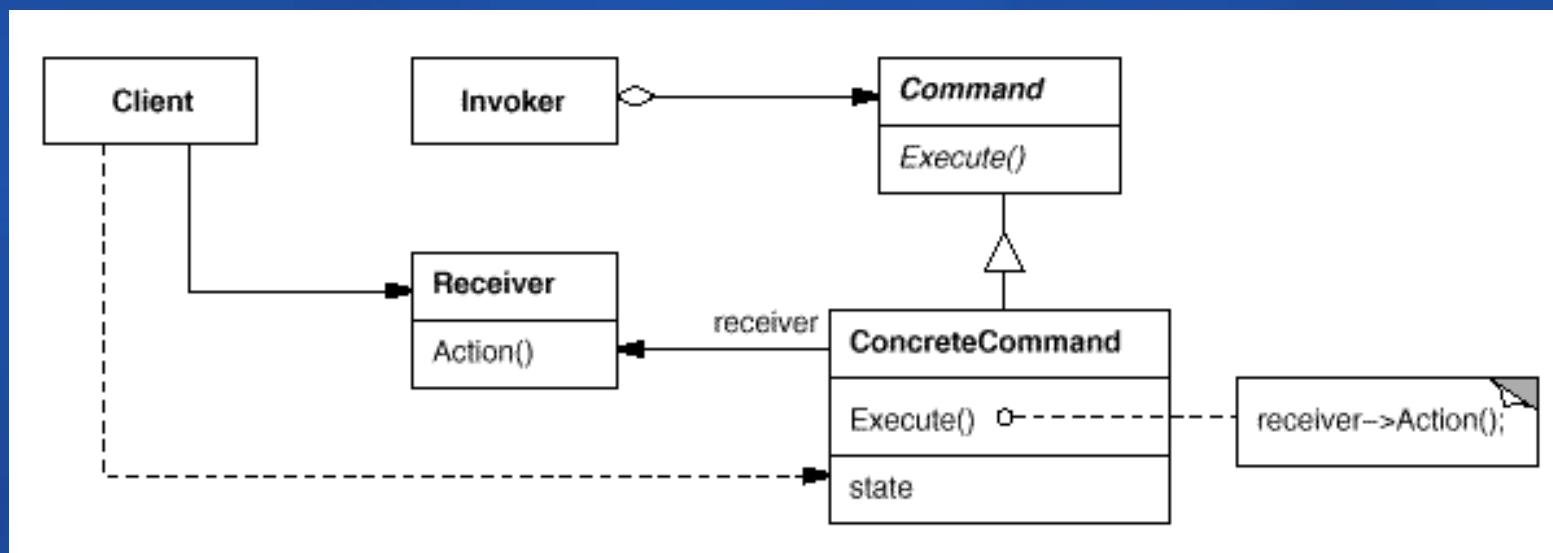
例说Command应用

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Codes in VS.NET

结构 (Structure)



Command模式的几个要点

- Command模式的根本目的在于将“行为请求者”与“行为实现者”解耦，在面向对象语言中，常见的实现手段是“将行为抽象为对象”。
- 实现Command接口的具体命令对象ConcreteCommand有时候根据需要可能会保存一些额外的状态信息。
- 通过使用Composite模式，可以将多个“命令”封装为一个“复合命令”MacroCommand。
- Command模式与C#中的Delegate有些类似。但两者定义行为接口的规范有所区别：Command以面向对象中的“接口-实现”来定义行为接口规范，更严格，更符合抽象原则；Delegate以函数签名来定义行为接口规范，更灵活，但抽象能力比较弱。

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

.NET架构中的Command

Codes in VS.NET

推荐资源


- 《设计模式：可复用面向对象软件的基础》 GoF
- 《面向对象分析与设计》 Grady Booch
- 《敏捷软件开发：原则、模式与实践》 Robert C. Martin
- 《重构：改善既有代码的设计》 Martin Fowler
- 《Refactoring to Patterns》 Joshua Kerievsky

获取更多MSDN资源

- **MSDN中文网站**
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.mspx>



Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts