

# 定义命令模式

在经过对象村餐厅的学习之后，你已经实现了部分的遥控器 API，而且在这个过程中，你也对命令模式内的类和对象是如何互动的理解得很清楚了。现在，我们就来定义命令模式，并敲定所有的细节。

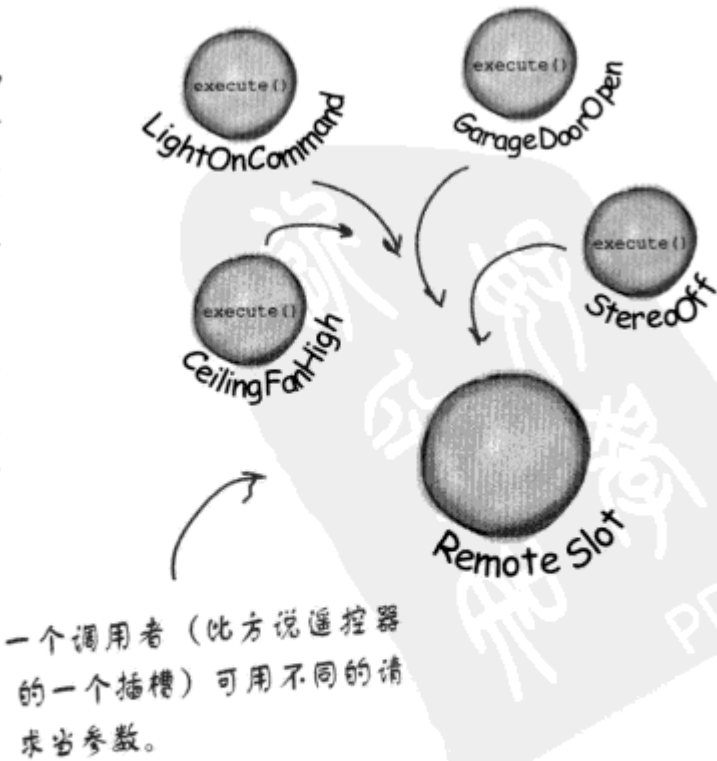
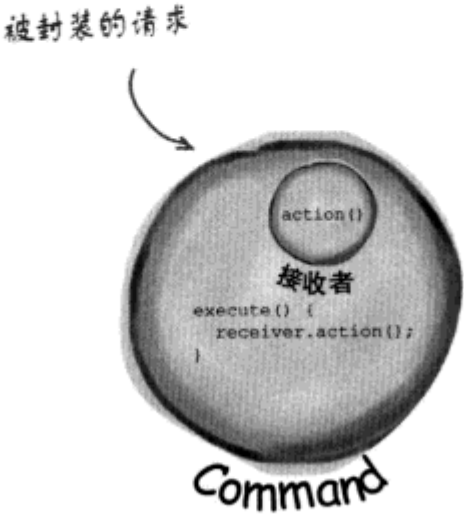
先从正式的定义开始：

**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。命令模式也支持可撤销的操作。

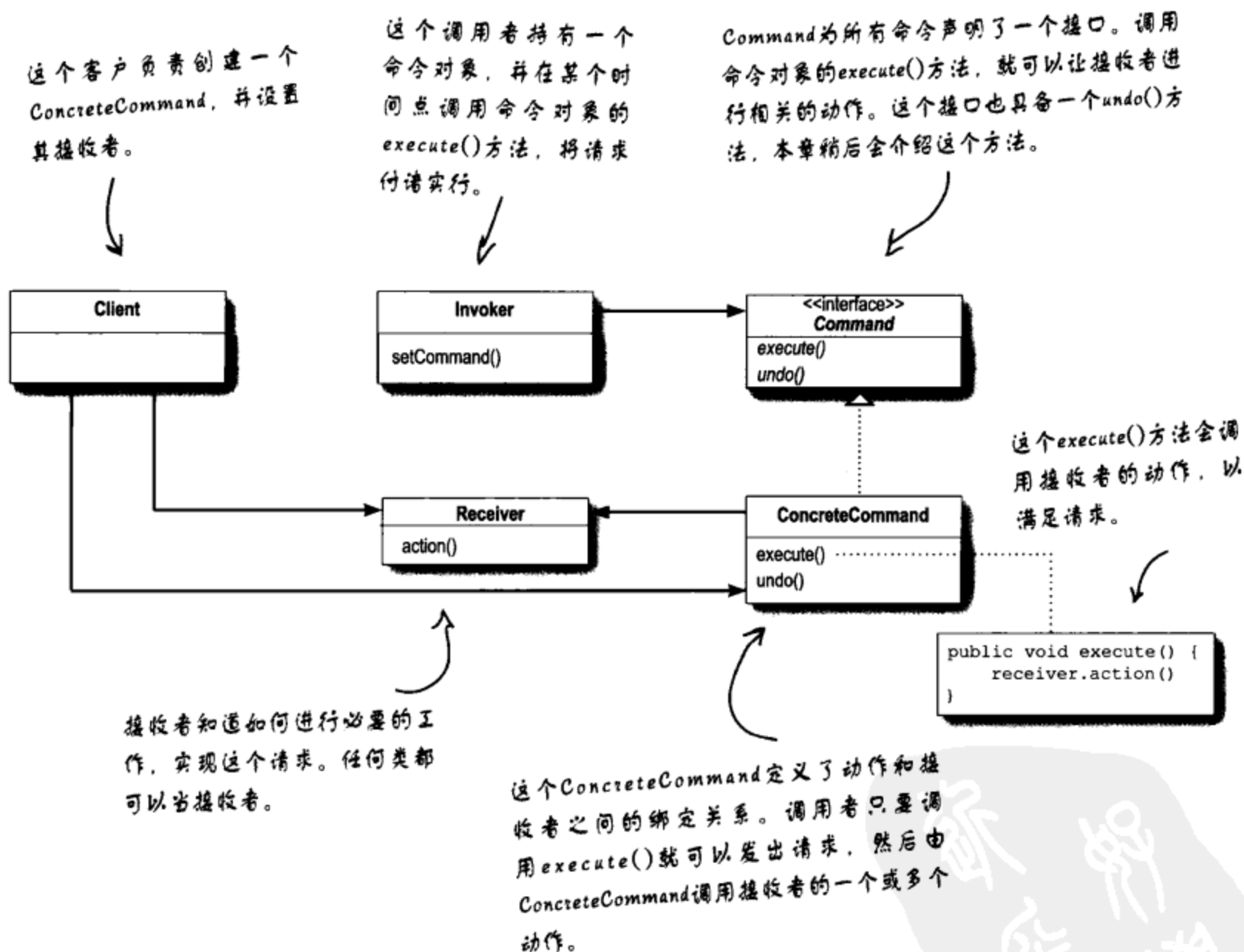
现在，仔细看这个定义。我们知道一个命令对象通过在特定接收者上绑定一组动作来封装一个请求。要达到这一点，命令对象将动作和接收者包进对象中。这个对象只暴露出一个 `execute()` 方法，当此方法被调用的时候，接收者就会进行这些动作。从外面来看，其他对象不知道究竟哪个接收者进行了哪些动作，只知道如果调用 `execute()` 方法，请求的目的就能达到。

我们也看到了利用命令来参数化对象的一些例子。再回到餐厅，一整天下来，女招待参数化许多订单。在简单遥控器中，我们先用一个“打开电灯”命令加载按钮插槽，稍后又将命令替换成为另一个“打开车库门”命令。就和女招待一样，遥控器插槽根本不在乎所拥有的是什么命令对象，只要该命令对象实现了 `Command` 接口就可以了。

我们还未说到使用命令模式来实现“队列、日志和支持撤销操作”。别担心，这是基本命令模式相当直接的扩展，很快我们就会看到这些内容。一旦有了足够的基础，也可以轻易地持所谓的 `Meta Command Pattern`。`Meta Command Pattern` 可以创建命令的宏，以便一次执行多个命令。



## 定义命令模式： 类图



命令模式的设计如何支持请求调用者和请求接收者之间的解耦？