# Java设计模式 命令模式

写本文是因为在看Spring Cloud的Hystrix，它采用了命令模式，本文记录一下个人的理解。

假设一个场景：

现在在家看电视，需要打开电视机、机顶盒。

这两个“开机”命令（Command），具体执行者是“接收”命令的电视、机顶盒（Receiver），而真正调用这两个命令的，不是我们人类（Client），而是遥控器（Invoker）——现实中对应有俩遥控器，本例为了方便，就假设只有一个万能遥控器。

总结以上，涉及有：

Receiver：命令接收，这是具体干活的。

Command：命令。

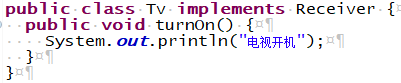
Invoker：调用者，接收、执行命令。

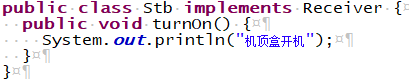
Client：客户端，发送命令。

代码如下：

Receiver:

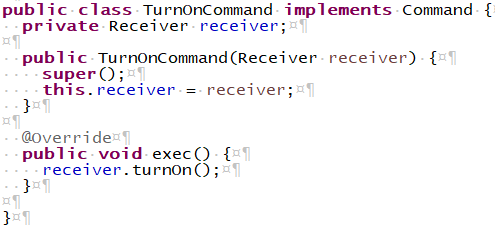






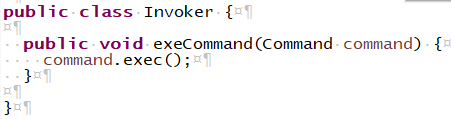
Command:



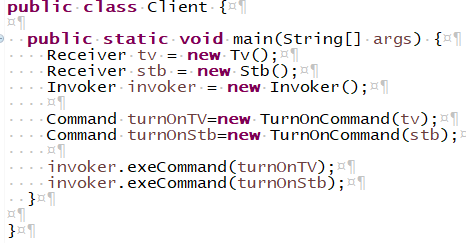


本例为了简单，就构建了一个开机命令，其实还会有很多命令，比如关机、音量大小、频道切换、图像调节等等，不再赘述。

Invoker：



Client：



结果：



简单说说优缺点。

最大的好处肯定就是解耦了，命令的调用者和接收者没有直接交互。

另一个就是命令可扩展了，就像上文所说，可以很方便地增加命令（关机、音量调节、频道切换等等），但是这也可以算成缺点，你想想我们得搞多少个具体命令类！



每一个按钮代表一个Command实现类~~~