

实现命令

嗯，我们已经尝试为SimpleRemoteControl 实现LightOnCommand。我们可以把同样的代码插入到这里，一切都运行得很漂亮。Off命令也没有什么不同；事实上，LightOffCommand 看起来像这样：

```
public class LightOffCommand implements Command {
    Light light;

    public LightOffCommand(Light light) {
        this.light = light;
    }

    public void execute() {
        light.off();
    }
}
```

LightOffCommand的运行方式和LightOnCommand一样，除了绑定接收者到不同的动作：off()方法。

我们来尝试一些更有挑战性的事情。为音响编写开和关命令怎么样？OK，关容易，只要把Stereo绑定到StereoOffCommand的off()方法。On有点复杂。假设我们要编写一个StereoOnWithCDCCommand……

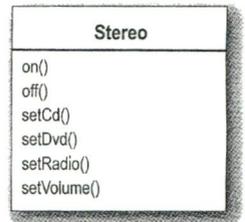
```
public class StereoOnWithCDCCommand implements Command {
    Stereo stereo;

    public StereoOnWithCDCCommand(Stereo stereo) {
        this.stereo = stereo;
    }

    public void execute() {
        stereo.on();
        stereo.setCD();
        stereo.setVolume(11);
    }
}
```

就像LightOnCommand一样，我们传入要控制的音响的实例，把它存在一个实例变量中。

要履行这个请求，我们需要调用音响的三个方法：首先，打开，然后设置成播放CD，最后把音量设置为11。为什么是11？嗯，总比10好些，对吧？



还不错。看看剩下的厂商类，现在你肯定可以搞定剩下的Command类。