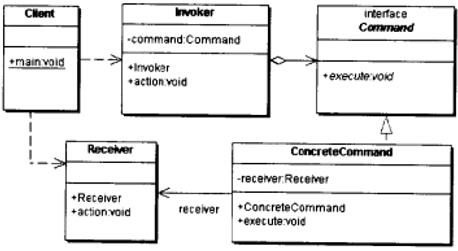
* 设计模式 – 命令模式（Command）：

把一个请求或者操作封装到一个对象中。命令模式把发出命令的责任和执行命令的责任分割开，委派给不同的对象。命令模式允许请求的一方和发送的一方独立开来，使得请求的一方不必知道接收请求的一方的接口，更不必知道请求是怎么被接收，以及操作是否执行，何时被执行以及是怎么被执行的。系统支持命令的撤消。



命令模式角色：

**1）客户(Client)角色**：创建一个具体的命令(ConcreteCommand)对象并确定其接收者。

2）**命令（Command）角色**：声明一个给所有具体命令类的抽象接口。

3）**具体命令（ConcreteCommand）角色**：定义一个接收者和行为之间的弱耦合；实现execute方法，负责调用接收者的相应操作。

4）**请求者（Invoker）角色**：负责调用命令对象执行请求。

5）**接收者（Receiver）角色**：负责具体实施和执行一个请求。

* 设计模式 – 命令模式的优点：

1：更松散的耦合。

2：更动态的控制。

3：能很自然的复合命令。

4：更好的扩展性。

在命令模式中，请求者（Invoker）不直接与接收者（Receiver）交互，即请求者（Invoker）不包含接收者（Receiver）的引用，因此彻底消除了彼此之间的耦合。

 如果增加新的具体命令和该命令的接受者，不必修改调用者的代码，调用者就可以使用新的命令对象；反之，如果增加新的调用者，不必修改现有的具体命令和接受者，新增加的调用者就可 以使用已有的具体命令。

* 设计模式 – 命令模式Demo：

