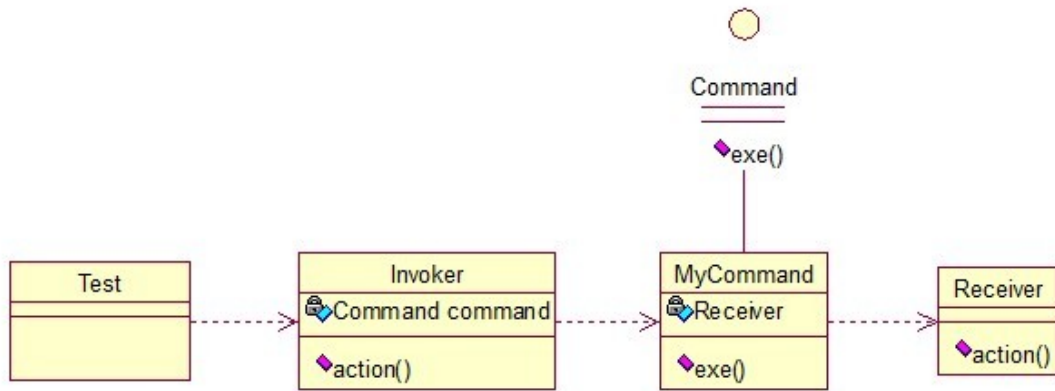


命令模式很好理解，举个例子，司令员下令让士兵去干件事情，从整个事情的角度来考虑，司令员的作用是，发出口令，口令经过传递，传到了士兵耳朵里，士兵去执行。这个过程好在，三者相互解耦，任何一方都不用去依赖其他人，只需要做好自己的事儿就行，司令员要的是结果，不会去关注到底士兵是怎么实现的。我们看看关系图：



Invoker是调用者（司令员），Receiver是被调用者（士兵），MyCommand是命令，实现了Command接口，持有接收对象，看实现代码：

[java] [view plaincopy](#)

```
public interface Command {

    public void exe();

}
```

[java] [view plaincopy](#)

```
public class MyCommand implements Command {

    private Receiver receiver;

    public MyCommand(Receiver receiver) {

        this.receiver = receiver;

    }

    @Override

    public void exe() {

        receiver.action();

    }

}
```

```
}
```

[java] [view plaincopy](#)

```
public class Receiver {  
    public void action(){  
        System.out.println("command received!");  
    }  
}
```

[java] [view plaincopy](#)

```
public class Invoker {  
  
    private Command command;  
  
    public Invoker(Command command) {  
        this.command = command;  
    }  
  
    public void action(){  
        command.exe();  
    }  
}
```

[java] [view plaincopy](#)

```
public class Test {  
  
    public static void main(String[] args) {  
        Receiver receiver = new Receiver();  
        Command cmd = new MyCommand(receiver);  
        Invoker invoker = new Invoker(cmd);  
        invoker.action();  
    }  
}
```

输出：command received!