## 2. 命令（Command）

### Intent

将命令封装成对象中，具有以下作用：

* 使用命令来参数化其它对象
* 将命令放入队列中进行排队
* 将命令的操作记录到日志中
* 支持可撤销的操作

### Class Diagram

* Command：命令
* Receiver：命令接收者，也就是命令真正的执行者
* Invoker：通过它来调用命令
* Client：可以设置命令与命令的接收者

### Implementation

设计一个遥控器，可以控制电灯开关。

public interface Command {
 void execute();
}

public class LightOnCommand implements Command {
 Light light;

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

 @Override
 public void execute() {
 light.on();
 }
}

public class LightOffCommand implements Command {
 Light light;

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

 @Override
 public void execute() {
 light.off();
 }
}

public class Light {

 public void on() {
 System.out.println("Light is on!");
 }

 public void off() {
 System.out.println("Light is off!");
 }
}

/\*\*
 \* 遥控器
 \*/
public class Invoker {
 private Command[] onCommands;
 private Command[] offCommands;
 private final int slotNum = 7;

 public Invoker() {
 this.onCommands = new Command[slotNum];
 this.offCommands = new Command[slotNum];
 }

 public void setOnCommand(Command command, int slot) {
 onCommands[slot] = command;
 }

 public void setOffCommand(Command command, int slot) {
 offCommands[slot] = command;
 }

 public void onButtonWasPushed(int slot) {
 onCommands[slot].execute();
 }

 public void offButtonWasPushed(int slot) {
 offCommands[slot].execute();
 }
}

public class Client {
 public static void main(String[] args) {
 Invoker invoker = new Invoker();
 Light light = new Light();
 Command lightOnCommand = new LightOnCommand(light);
 Command lightOffCommand = new LightOffCommand(light);
 invoker.setOnCommand(lightOnCommand, 0);
 invoker.setOffCommand(lightOffCommand, 0);
 invoker.onButtonWasPushed(0);
 invoker.offButtonWasPushed(0);
 }
}

### JDK

* [java.lang.Runnable](http://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html)
* [Netflix Hystrix](https://github.com/Netflix/Hystrix/wiki)
* [javax.swing.Action](http://docs.oracle.com/javase/8/docs/api/javax/swing/Action.html)