

```

//Command
//public
interface Command{
    public void execute();
}

//Now let's create two concrete commands. One will turn on the lights,
another turns off lights:
//Concrete Command
//public
class LightOnCommand implements Command{
    //reference to the light
    Light light;
    public LightOnCommand(Light light)
    {
        this.light = light;
        System.out.println("\n*****Light On *****");
    }
    public void execute(){
        light.switchOn();
    }
}

//Concrete Command
//public
class LightOffCommand implements Command
{
    //reference to the light
    Light light;
    public LightOffCommand(Light light)
    {
        this.light = light;
        System.out.println("\n*****Light Off *****");
    }
    public void execute(){
        light.switchOff();
    }
}

//Light is our receiver class, so let's set that up now:

//Receiver
//Receiver
//public
class Light
{
    private boolean on;
    public void switchOn()
    {
        on = true;
    }
}

```

```
    public void switchOff()
    {
        on = false;
    }
}
```

//Our invoker in this case is the remote control.

```
//Invoker
//public
class RemoteControl{
    private Command command;
    public void setCommand(Command command)
    {
        this.command = command;
    }
    public void pressButton(){
        command.execute();
    }
}
```

//Finally we'll set up a client to use the invoker

```
//Client
//public
class Client
{
    public static void main(String[] args)
    {
        RemoteControl control = new RemoteControl();
        Light light = new Light();
        Command lightsOn = new LightOnCommand(light);
        Command lightsOff = new LightOffCommand(light);

        //switch on
        control.setCommand(lightsOn);
        control.pressButton();

        //switch off
        control.setCommand(lightsOff);
        control.pressButton();
        System.out.println("File execute -----");
    }
}
```