**CommandPattern Example**

**Command.java**

package com.command;

public interface Command {

    void execute();

}

**Light.java**

package com.command;

public class Light {

    public void turnOn() {

        System.out.println("Light is ON.");

    }

    public void turnOff() {

        System.out.println("Light is OFF.");

    }

}

**LightOffCommand.java**

package com.command;

public class LightOffCommand implements Command {

    private Light light;

    public LightOffCommand(Light light) {

        this.light = light;

    }

    @Override

    public void execute() {

        light.turnOff();

    }

}

**LightOnCommand.java**

package com.command;

public class LightOnCommand implements Command {

    private Light light;

    public LightOnCommand(Light light) {

        this.light = light;

    }

    @Override

    public void execute() {

        light.turnOn();

    }

}

**Main.java**

package com.command;

public class Main {

    public static void main(String[] args) {

        Light livingRoomLight = new Light();

        // Commands

        Command lightOn = new LightOnCommand(livingRoomLight);

        Command lightOff = new LightOffCommand(livingRoomLight);

        // Invoker

        RemoteControl remote = new RemoteControl();

        // Turn on the light

        remote.setCommand(lightOn);

        remote.pressButton();

        // Turn off the light

        remote.setCommand(lightOff);

        remote.pressButton();

    }

}

**RemoteControl.java**

package com.command;

public class RemoteControl {

    private Command command;

    public void setCommand(Command command) {

        this.command = command;

    }

    public void pressButton() {

        if (command != null) {

            command.execute();

        } else {

            System.out.println("No command assigned to remote.");

        }

    }

}

