**Implementing the Command Pattern**

**CommandPatternExample:**

interface Command {

    void execute();

}

class Light {

    void turnOn() {

        System.out.println("Light has turned ON");

    }

    void turnOff() {

        System.out.println("Light has turned OFF");

    }

}

class LightOnCommand implements Command {

    private Light light;

    public LightOnCommand(Light light) {

        this.light = light;

    }

    public void execute() {

        light.turnOn();

    }

}

class LightOffCommand implements Command {

    private Light light;

    public LightOffCommand(Light light) {

        this.light = light;

    }

    public void execute() {

        light.turnOff();

    }

}

class RemoteControl {

    private Command command;

    public void setCommand(Command command) {

        this.command = command;

    }

    public void Execute() {

        command.execute();

    }

}

public class CommandPatternExample {

    public static void main(String[] args) {

        Light light = new Light();

        Command lightOn = new LightOnCommand(light);

        Command lightOff = new LightOffCommand(light);

        RemoteControl remote = new RemoteControl();

        remote.setCommand(lightOn);

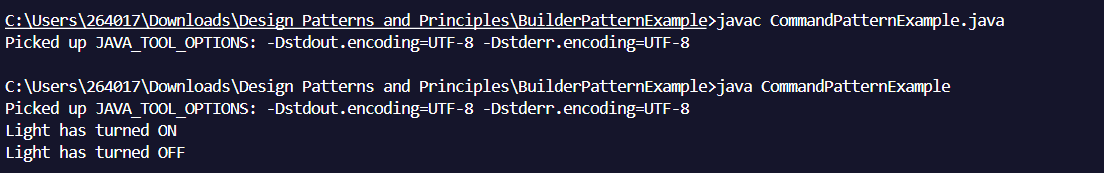
        remote.Execute();

        remote.setCommand(lightOff);

        remote.Execute();

    } }

**Output:**

****