**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 9: Implementing the Command Pattern**

Scenario: You are developing a home automation system where commands can be issued to turn devices on or off. Use the Command Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named CommandPatternExample.
2. **Define Command Interface:**
   * Create an interface Command with a method execute().
3. **Implement Concrete Commands:**
   * Create classes LightOnCommand, LightOffCommand that implement Command.
4. **Implement Invoker Class:**
   * Create a class RemoteControl that holds a reference to a Command and a method to execute the command.
5. **Implement Receiver Class:**
   * Create a class Light with methods to turn on and off.
6. **Test the Command Implementation:**
   * Create a test class to demonstrate issuing commands using the RemoteControl.

**DESCRIPTION:**

This program demonstrates the Command Pattern in a home automation system.

* Command is the interface with the execute() method.
* LightOnCommand and LightOffCommand are concrete commands.
* Light is the receiver with actions turnOn() and turnOff().
* RemoteControl is the invoker that triggers commands.
* The client sets and executes commands through the remote.

**PROGRAM:**

public class CommandPatternExample {

public static void main(String[] args) {

Light livingRoomLight = new Light();

Command lightOn = new LightOnCommand(livingRoomLight);

Command lightOff = new LightOffCommand(livingRoomLight);

RemoteControl remote = new RemoteControl();

remote.setCommand(lightOn);

remote.pressButton();

remote.setCommand(lightOff);

remote.pressButton();

}

}

interface Command {

void execute();

}

class Light {

public void turnOn() {

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

}

public void turnOff() {

System.out.println("Light is 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 pressButton() {

if (command != null) {

command.execute();

} else {

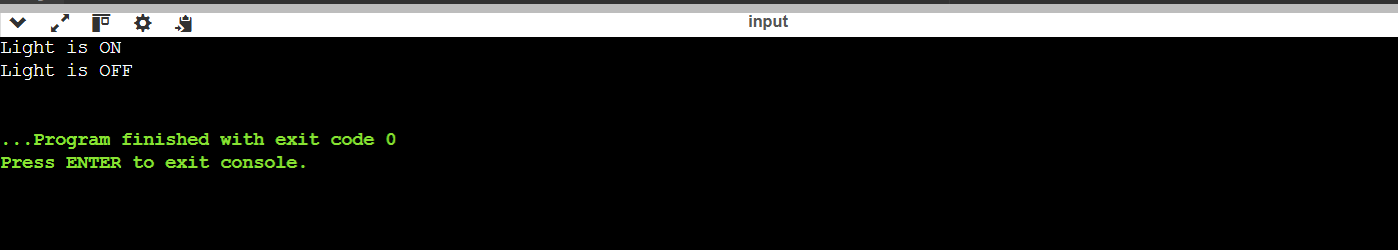
System.out.println("No command set.");

}

}

}

**OUTPUT:**

****