**Exercise 9: Implementing the Command Pattern**

**Source Code:**

**Command.java**

public interface Command { void execute(); }

**Light.java**

public class Light {

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

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

}

**LightOnCommand.java**

public class LightOnCommand implements Command {

private final Light l;

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

public void execute() { l.on(); }

}

**LightOffCommand.java**

public class LightOffCommand implements Command {

private final Light l;

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

public void execute() { l.off(); }

}

**RemoteControl.java**

public class RemoteControl {

private Command cmd;

public void setCommand(Command c) { cmd = c; }

public void pressButton() { cmd.execute(); }

}

**CommandDemo.java**

public class CommandDemo {

public static void main(String[] a) {

Light light = new Light();

RemoteControl rc = new RemoteControl();

rc.setCommand(new LightOnCommand(light));

rc.pressButton();

rc.setCommand(new LightOffCommand(light));

rc.pressButton();

}

}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**