Exercise 9: Smart Light Control using Command Pattern

Approach and Understanding:

* We are building a **smart light control system** where the user can easily turn the light

**ON** or **OFF** by typing commands.

We use the **Command Pattern** here, which simply means we create buttons (commands) that can control the light**.**

**Code**

import java.util.Scanner; interface Command {

void execute();

}

class Light {

public void on() {

System.out.println("The light is now ON.");

}

public void off() {

System.out.println("The light is now OFF.");

}

}

class LightOn implements Command { private Light light;

public LightOn(Light light) { this.light = light;

}

public void execute() { light.on();

}

}

class LightOff implements Command { private Light light;

public LightOff(Light light) { this.light = light;

}

public void execute() { light.off();

}

}

class Remote {

private Command command;

public void set(Command command) { this.command = command;

}

public void press() {

if (command != null) { command.execute();

}

}

}

public class HomeApp {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in); Light light = new Light();

Remote remote = new Remote();

while (true) {

System.out.print("\nEnter 'on' to turn on, 'off' to turn off, or 'stop' to exit: "); String input = scanner.nextLine().trim();

if (input.equalsIgnoreCase("stop")) { break;

} else if (input.equalsIgnoreCase("on")) { remote.set(new LightOn(light));

} else if (input.equalsIgnoreCase("off")) { remote.set(new LightOff(light));

} else {

System.out.println("Invalid input. Please try again."); continue;

}

remote.press();

}

scanner.close();

System.out.println("Goodbye!");

}

}

**OUTPUT:**

