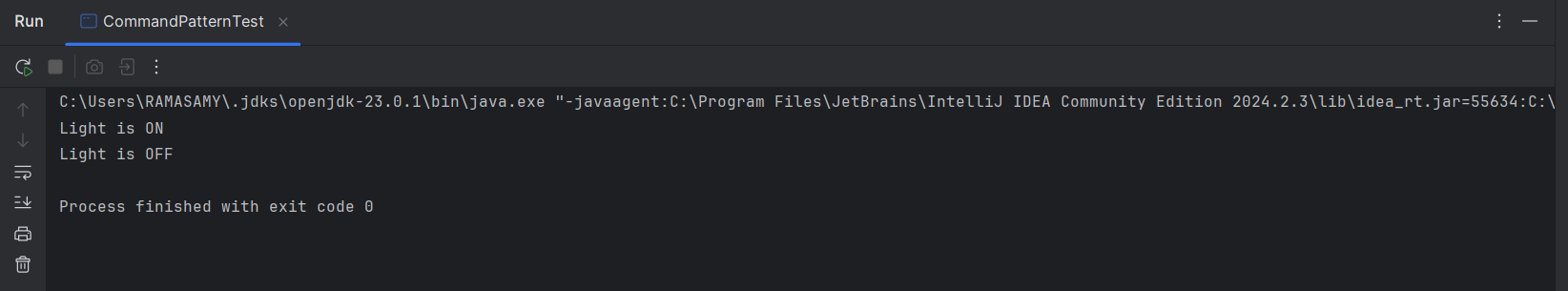
**Exercise 9: Implementing the Command Pattern**

**CODE:**

**CommandPatternTest.java**

interface Command {  
 void execute();  
}  
  
class Light {  
 public void on() {  
 System.*out*.println("Light is ON");  
 }  
 public void off() {  
 System.*out*.println("Light is OFF");  
 }  
}  
  
class LightOnCommand implements Command {  
 private Light light;  
 public LightOnCommand(Light light) {  
 this.light = light;  
 }  
 public void execute() {  
 light.on();  
 }  
}  
  
class LightOffCommand implements Command {  
 private Light light;  
 public LightOffCommand(Light light) {  
 this.light = light;  
 }  
 public void execute() {  
 light.off();  
 }  
}  
  
class RemoteControl {  
 private Command command;  
 public void setCommand(Command command) {  
 this.command = command;  
 }  
 public void pressButton() {  
 command.execute();  
 }  
}  
  
class CommandPatternTest {  
 public static void main(String[] args) {  
 Light light = new Light();  
 RemoteControl remote = new RemoteControl();  
  
 remote.setCommand(new LightOnCommand(light));  
 remote.pressButton();  
  
 remote.setCommand(new LightOffCommand(light));  
 remote.pressButton();  
 }  
}

**OUTPUT:**

****