

Command Design Pattern

Problem:

The objective of this exercise is to implement the command design pattern.

Steps:

1. First, we'll create our **command interface**:

```
1  //Aimable Mugwaneza and Romerico David
2  interface Command{
3      public void execute();
4  }
```

2. Now let's create two **concrete commands**. One will turn on the lights, another turns off lights:

```
17  class LightsOnCommand implements Command{
18      Light light;
19
20      public LightsOnCommand(Light light){
21          this.light = light;
22      }
23
24      public void execute(){
25          light.on();
26      }
27  }
28
29  class LightsOffCommand implements Command{
30      Light light;
31
32      public LightsOffCommand(Light light){
33          this.light = light;
34      }
35
36      public void execute(){
37          light.off();
38      }
39  }
40  }
41
```

3. Light is our **receiver** class, so let's set that up now:

Name: Aimable M and David R

```
6   class Light{
7
8       public void on(){
9           System.out.println(x:"Light on");
10      }
11
12      public void off(){
13          System.out.println(x:"Light off");
14      }
15  }
16
```

4. Following the step 2 and 3, create commands for the GarageDoorOpenCommand and GarageDoorCloseCommand, and the receiver GarageDoor

Concrete Door Commands

D

```
54
55   class GarageDoorOpenCommand implements Command{
56       GarageDoor door;
57
58       public GarageDoorOpenCommand(GarageDoor door){
59           this.door = door;
60       }
61
62       public void execute(){
63           door.open();
64       }
65   }
66
67   class GarageDoorCloseCommand implements Command{
68       GarageDoor door;
69
70       public GarageDoorCloseCommand(GarageDoor door){
71           this.door = door;
72       }
73
74       public void execute(){
75           door.close();
76       }
77   }
78
79   }
80
```

Name: Aimable M and David R

GarageDoorReceiver

```
44 class GarageDoor{
45
46     public void open(){
47         System.out.println(x:"Door open");
48     }
49
50     public void close(){
51         System.out.println(x:"Door closed");
52     }
53 }
54
```

5. Our **invoker** in this case is the remote control.

```
81 class RemoteControl {
82     private Command command;
83
84     public RemoteControl() {
85     }
86
87     public void setCommand(Command command) {
88         this.command = command;
89     }
90
91     public void pressButton() {
92         command.execute();
93     }
94 }
95
```

6. set up a **client** to use the invoker

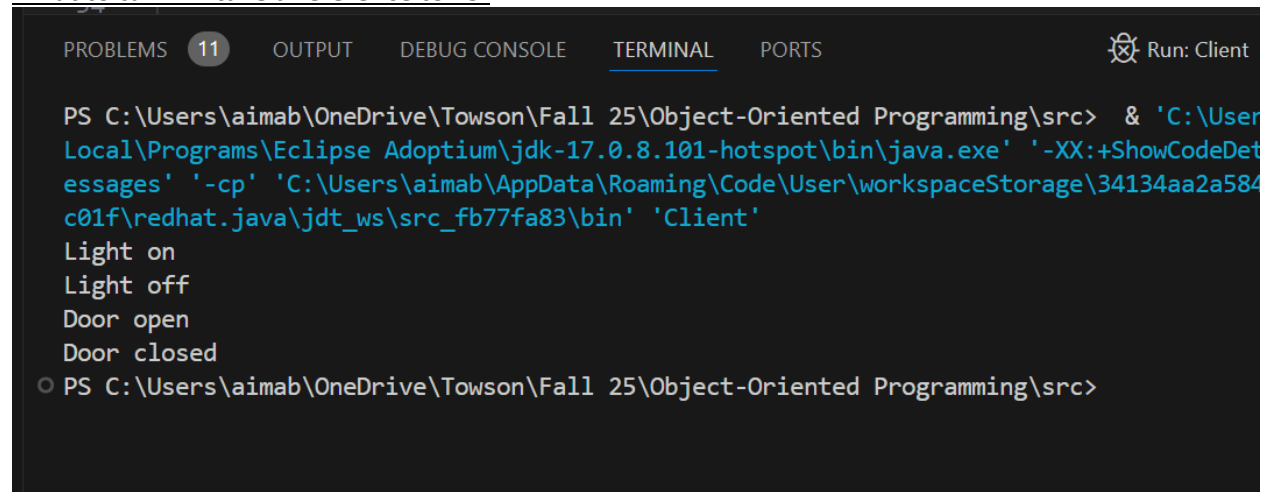
```
//Client
public class Client{
    public static void main(String[] args) {
        RemoteControl control = new RemoteControl();
        Light light = new Light();
        Command lightsOn = new LightsOnCommand(light);
        Command lightsOff = new LightsOffCommand(light);
        GarageDoor garageDoor = new GarageDoor ();
        Command garageDoorOpen = new GarageDoorOpenCommand (garageDoor);
        Command garageDoorClose = new GarageDoorCloseCommand (garageDoor);

        //switch on
        control.setCommand(lightsOn);
        control.pressButton();
        //switch off
    }
}
```

Name: Aimable M and David R

```
control.setCommand(lightsOff);  
control.pressButton();  
//switch on  
control.setCommand(garageDoorOpen);  
control.pressButton();  
//switch off  
control.setCommand(garageDoorClose);  
control.pressButton();  
  
}  
}
```

What to turn in: take a reference to E5.



The screenshot shows the Eclipse IDE's terminal window. The top bar includes tabs for PROBLEMS (11), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. A 'Run: Client' button is visible on the right. The terminal content shows a command prompt session where a Java program is executed. The output of the program is as follows:

```
PS C:\Users\aimab\OneDrive\Towson\Fall 25\Object-Oriented Programming\src> & 'C:\User  
Local\Programs\Eclipse Adoptium\jdk-17.0.8.101-hotspot\bin\java.exe' '-XX:+ShowCodeDet  
essages' '-cp' 'C:\Users\aimab\AppData\Roaming\Code\User\workspaceStorage\34134aa2a584  
c01f\redhat.java\jdt_ws\src_fb77fa83\bin' 'Client'  
Light on  
Light off  
Door open  
Door closed  
PS C:\Users\aimab\OneDrive\Towson\Fall 25\Object-Oriented Programming\src>
```