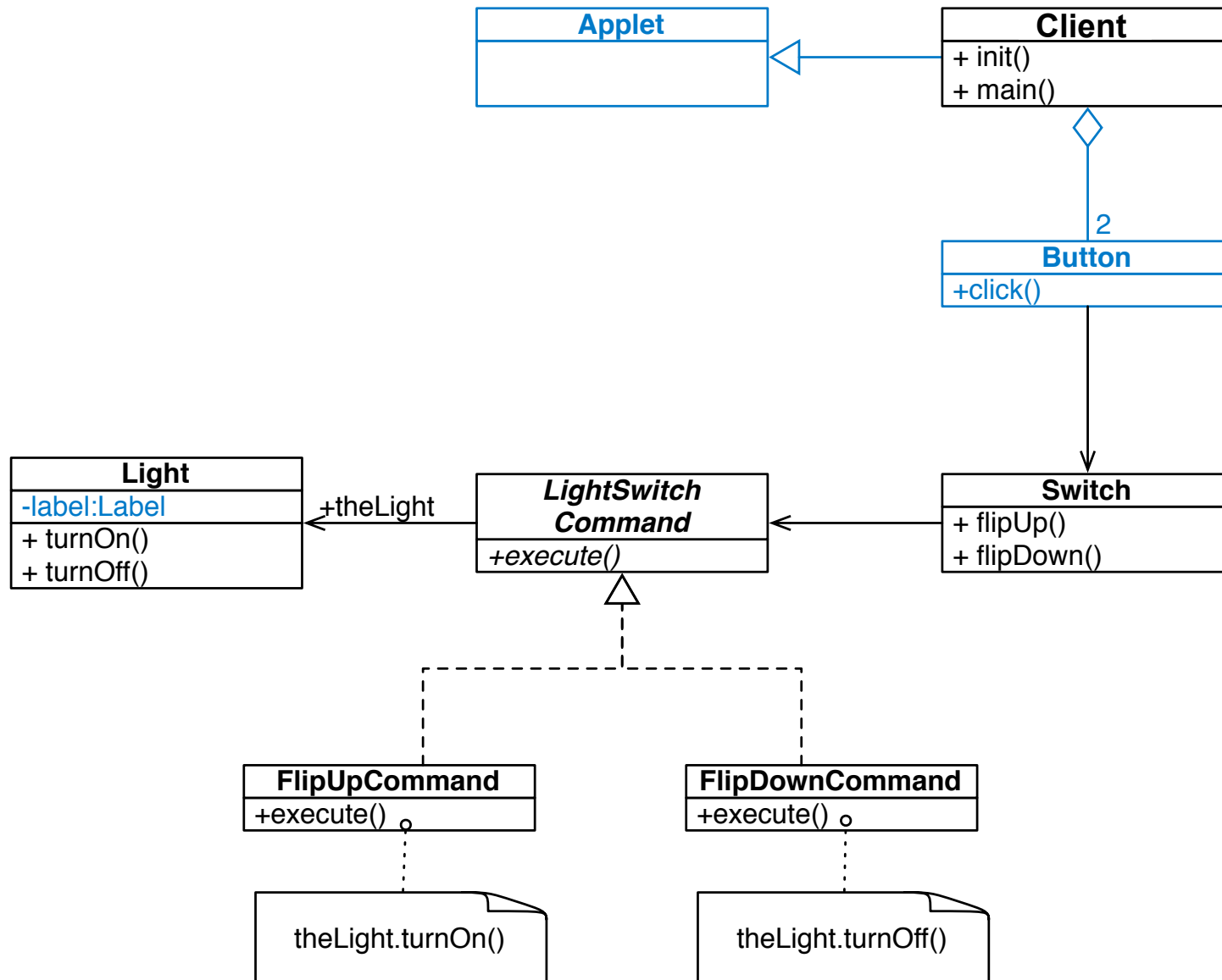# Solution to Task #1

Command Pattern

Solution to Task #1

Solution to Task #1: Client class => Applet

```java
package lightswitch;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener
```

AWT imports

```java
@SuppressWarnings("serial")
public class Client extends java.applet.Applet {
```

Client Applet

```java
    Light lamp;
    LightSwitchCommand switchUp;
    LightSwitchCommand switchDown;
    Switch lightSwitch;
    Button btnUp;
    Button btnDown;
    Label lightLabel;
```

Buttons Declarations

Solution to Task #1: Client class - Initialization

```
public void init() {
    lightLabel= new Label();
    lamp = new Light(lightLabel);
    switchUp = new FlipUpCommand(lamp);
    switchDown = new FlipDownCommand(lamp);
    lightSwitch = new Switch(switchUp, switchDown);
```

```
setLayout(new GridLayout(3,2)); //Use Grid Layout
btnUp = new Button();
btnDown = new Button();
btnUp.setLabel("switch up");
btnDown.setLabel("switch down");
add(btnUp);
add(btnDown);
```

Button initialization

```
add(lamp);
```

Add the lamp to the UI

Solution to Task #1: Client class - Delegation

```java
btnUp.addActionListener(new ActionListener() {
  @Override
  public void actionPerformed(ActionEvent e) {
    lightSwitch.flipUp();
  }
});
btnDown.addActionListener(new ActionListener() {
  @Override
  public void actionPerformed(ActionEvent e) {
    lightSwitch.flipDown();
  }
});
}
```

Connect buttons with invoker

Solution to Task #1: Light class - Receiver

```java
public class Light extends java.awt.Label {
  private final Label lightLabel;


  public Light(Label lightLabel) {
    this.lightLabel = lightLabel;
  }
  public void turnOn() {
        System.out.println("The light is on");
        lightLabel.setBackground(new java.awt.Color(255, 255, 0));
  }


  public void turnOff() {
        System.out.println("The light is off");
        lightLabel.setBackground(new java.awt.Color(0, 0, 0));
  }
}
```

Reflect the status change in the UI