

```
1 package semaforo.trafficLight.simple;
2
3 import java.awt.Point;
18
19 public class SimpleTrafficLight implements Paintable, TrafficLight {
20
21     private Point position;
22     private Dimension dimension;
23     private Image mask;
24     private SpotLight yellow;
25     private SpotLight green;
26     private SpotLight red;
27
28     public SimpleTrafficLight() throws IOException {
29         this.position = new Point(0,0);
30         this.dimension = new Dimension(70,180);
31         this.create();
32         this.configurePosition();
33     }
34
35     public SimpleTrafficLight(Point position, Dimension dimension) throws IOException {
36         this.position = position;
37         this.dimension = dimension;
38         this.create();
39         this.configurePosition();
40     }
41
42     public void setPosition(Point position) {
43         this.position = position;
44     }
45
46     public Point getPosition() {
47         return this.position;
48     }
49
50     public void setDimension(Dimension dimension) {
51         this.dimension = dimension;
52     }
53
54     public Dimension getDimension() {
55         return this.dimension;
56     }
57
58     private String currentRelativePath() {
59
60         return "/" +
61             this.getClass()
62                 .getPackageName()
63                 .toString()
64                 .replace('.', '/') +
65             "/";
66     }
67
68     private SpotLight createSpot(String color) throws IOException {
69
70         final String path = currentRelativePath() + "img/";
71         URL url;
72
73         url = getClass().getResource(path + color + "On.png");
74         Image maskOn = ImageIO.read(url);
75
76     }
```

```
77         url = getClass().getResource(path + color + "Off.png");
78         Image maskOff = ImageIO.read(url);
79
80         SpotLight spot = new SpotLight(maskOn, maskOff);
81
82         return spot;
83
84     }
85
86     public void create() throws IOException {
87
88         this.green = createSpot("green");
89         this.green.setLight(new E27LightBulb());
90
91         this.yellow = createSpot("yellow");
92         this.yellow.setLight(new E27LightBulb());
93
94         this.red = createSpot("red");
95         this.red.setLight(new E27LightBulb());
96
97         String path = this.currentRelativePath();
98         URL url = this.getClass().getResource(path + "img/trafficLight.png");
99         this.mask = ImageIO.read(url);
100
101     }
102
103     private void configurePosition() {
104
105         final int WIDTH = (this.dimension.width - 20);
106         final int HEIGHT = ((this.dimension.height - 30) / 3);
107         final Dimension dimension = new Dimension(WIDTH, HEIGHT);
108
109         int xLeft = this.position.x + 10;
110         int yTop = this.position.x + 10;
111         this.green.setPosition(xLeft, yTop);
112         this.green.setDimension(dimension);
113
114         yTop = (yTop + 5 + HEIGHT);
115         this.yellow.setPosition(xLeft, yTop);
116         this.yellow.setDimension(dimension);
117
118         yTop = (yTop + 5 + HEIGHT);
119         this.red.setPosition(xLeft, yTop);
120         this.red.setDimension(dimension);
121
122     }
123
124     public void paint(Graphics g) {
125
126         synchronized(g) {
127             int xLeft = this.position.x;
128             int yTop = this.position.y;
129             int width = this.dimension.width;
130             int height = this.dimension.height;
131
132             g.drawImage(mask, xLeft, yTop, width, height, null);
133
134             this.green.paint(g);
135             this.yellow.paint(g);
136             this.red.paint(g);
137
138         }
```

```
139     )
140   }
141
142   @Override
143   public TurnOnOff spotGreen() {
144     return green;
145   }
146
147   @Override
148   public TurnOnOff spotYellow() {
149     return yellow;
150   }
151
152   @Override
153   public TurnOnOff spotRed() {
154     return red;
155   }
156 }
157
```