

Светофор.

1. Прописываем класс Light

Добавляем классы из библиотеки awt:

```
import java.awt.Color;
import java.awt.geom.Path2D;
import java.awt.Graphics2D;
```

Создаем класс Light:

```
6 usages  divan*
6 public class Light {
7     // класс лайгхт
8
9 }
10 //
```

Прописываем классу поля:

```
6 usages  divan*
6 public class Light {
7     1usage
    private final Color color;
8     1usage
    private final int y;
9 }
```

Прописываем конструктор:

```
3 usages  divan
10 public Light(Color color, int y) {
11     this.color = color;
12     this.y = y;
13 }
```

Прописываем метод drawLight()

```
3 usages  divan
15 @ public void drawLight(Graphics2D g2d) {
16     g2d.setColor(Color.WHITE); // Устанавливает цвет для фона светофора
17     g2d.fillOval(315, -20, 100, 100); // Рисует белый фон светофора
18     g2d.setColor(Color.DARK_GRAY); // Устанавливает цвет для обводки светофора
19     g2d.fillOval(315, -8, 100, 100); // Рисует обводку светофора
20     g2d.setColor(color); // Устанавливает цвет для сигнала светофора (красный, желтый, зеленый)
21     g2d.fillOval(315, y, 90, 90); // Рисует сигнал светофора на соответствующей вертикальной позиции
22 }
```

И метод drawEar()

```
3 usages  divan*
24 @ public void drawEar(Graphics2D g2d) {
25     int x = 440;
26     int x2 = 290;
27
28     g2d.setColor(Color.DARK_GRAY); // Устанавливает цвет для рисования
29     // right ear
30     Path2D.Double ear = new Path2D.Double();
31
32     ear.moveTo(x, y + 59); // Устанавливает начальную точку правого треугольника
33     ear.lineTo(x, y + 4); // Рисует верхнюю границу правого треугольника
34     ear.quadTo(x, y - 1, x2: x + 5, y2: y - 1); // Рисует кривую верхнюю часть правого треугольника
35     ear.lineTo(x + 60, y - 1); // Рисует горизонтальную линию верхней части правого треугольника
36     ear.quadTo(x + 65, y - 1, x2: x + 65, y2: y - 6); // Рисует кривую нижнюю часть правого треугольника
37     ear.lineTo(x + 5, y + 64); // Рисует нижнюю границу правого треугольника
38     ear.quadTo(x, y + 64, x, y2: y + 59); // Рисует кривую нижнюю часть правого треугольника
39     g2d.fill(ear); // Заполняет правого треугольника указанным цветом
40
41     // left ear
42     Path2D.Double ear1 = new Path2D.Double();
43     ear1.moveTo(x2, y + 59); // Устанавливает начальную точку левого треугольника
44     ear1.lineTo(x2, y + 4); // Рисует верхнюю границу левого треугольника
45     ear1.quadTo(x2, y - 1, x2: x2 - 5, y2: y - 1); // Рисует кривую верхнюю часть левого треугольника
46     ear1.lineTo(x2 + 60, y - 1); // Рисует горизонтальную линию верхней части левого треугольника
47     ear1.quadTo(x2 + 65, y - 1, x2: x2 - 65, y2: y + 6); // Рисует кривую нижнюю часть левого треугольника
48     ear1.lineTo(x2 - 5, y + 64); // Рисует нижнюю границу левого треугольника
49     ear1.quadTo(x2, y + 64, x2, y2: y + 59); // Рисует кривую нижнюю часть левого треугольника
50     g2d.fill(ear1); // Заполняет левый треугольник указанным цветом
51 }
```

2. Прописываем класс DrawingCanvas.

Подключаем библиотеки:

```
import javax.swing.JComponent;
import java.awt.Graphics;
import java.awt.geom.Path2D;
import java.awt.Graphics2D;
import java.awt.Color;
```

Создаём класс DrawingCanvas

```
public class DrawingCanvas extends JComponent {
    |
}
```

Прописываем классу поля:

```

2 usages
9     private final int weight;
2 usages
10    private final int height;

```

Создаем объекты класса Light:

```

10    private final int height;
2 usages
11    private final Light red = new Light(Color.RED, y: 270);
2 usages
12    private final Light yellow = new Light(Color.YELLOW, y: 380);
2 usages
13    private final Light green = new Light(Color.GREEN, y: 490);
14

```

Прописываем конструктор:

```

14
15    public DrawingCanvas(int weight, int height) {
16        this.weight = weight;
17        this.height = height;
18    }
19
20    1 usage  1 divan

```

Переопределяем метод paintComponent()

```

20    @Override
21    protected void paintComponent(Graphics graphics) {
22        super.paintComponent(graphics);
23        Graphics2D graphics2D = (Graphics2D) graphics;
24        // body
25        graphics2D.setColor(Color.DARK_GRAY); // Устанавливает цвет для тела светофора
26        // fillRoundRect Используется для рисования закрашенного (заполненного) прямоугольника с закругленными углами
27        graphics2D.fillRoundRect(300, 240, 130, 360, 25, 25);
28        graphics2D.fillRoundRect(335, 603, 60, 65, 15, 15);
29        // lights
30        red.drawLight(graphics2D); // рисуем красный круг
31        yellow.drawLight(graphics2D); // рисуем жёлтый круг
32        green.drawLight(graphics2D); // рисуем зелёный круг
33        // hat
34        Path2D.Double hat = new Path2D.Double();
35        hat.moveTo(320, 230); // используется для создания пути
36        hat.lineTo(410, 230); // используется для рисования линии от текущей позиции до указанной точки
37        hat.quadTo(410.5, 225.5, 410, 225); // используется для рисования квадратичной кривой Безье
38        hat.curveTo(390, 215, 330, 215, 320, 225); // используется для рисования кубической кривой Безье
39        hat.quadTo(315.5, 225.5, 320, 230); // используется для рисования квадратичной кривой Безье
40        graphics2D.setColor(Color.DARK_GRAY); // указываем цвет
41        graphics2D.fill(hat); // Заполняет указанным цветом
42        // sideTriangles
43        red.drawEar(graphics2D); // рисуем уши от красного круга
44        yellow.drawEar(graphics2D); // рисуем уши от жёлтого круга
45        green.drawEar(graphics2D); // рисуем уши от зелёного круга
46    }
47

```

3. Класс Main:

```
1 package pz_21;
2
3 import javax.swing.*;
4
5  divan *
6 public class Main {
7     divan *
8     public static void main(String[] args) {
9         int weight = 900, height = 800;
10        JFrame jFrame = new JFrame();
11        DrawingCanvas drawingCanvas = new DrawingCanvas(weight, height);
12        // устанавливает размер окна в пикселях
13        jFrame.setSize(weight, height);
14        // устанавливает заголовок в верхней части окна
15        jFrame.setTitle("Светофор");
16        // добавляем в окно элементы
17        jFrame.add(drawingCanvas);
18        // для установки реакции на закрытие окна JFrame
19        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20        // окно становится видимым и пользователь может взаимодействовать с ним
21        jFrame.setVisible(true);
22        // метод программной перерисовки repaint() для стандартных компонентов
23        drawingCanvas.repaint();
24    }
25 }
```