



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 2

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”

Виконав
студент III курсу
групи КП-82

Шило Андрій Володимирович
(прізвище, ім'я, по батькові)

варіант № 20

Зарахована
“ ____ ” “ ____ ” 20__ р.
викладачем

Шкурат Оксаною Сергіївною
(прізвище, ім'я, по батькові)

Варіант завдання

Завдання: За допомогою Java2D намалювати картинку з лабораторної роботи No1 (за варіантом).

1. Хоча б 1 стандартний примітив, та хоча б 1 фігуру, побудовану по точкам (ламаною).
2. Хоча б 1 фігуру залити градієнтною фарбою за вибором (в цьому випадку колір може не співпадати з варіантом із лабораторної роботи No 1).
3. На достатній відстані від побудованого малюнку намалювати прямокутну рамку, всередині якої відбуватиметься анімація. Тип лінії рамки задано за варіантом.
4. Виконати анімацію малюнку, за варіантом. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов'язкова, якщо не передбачено варіантом.

Варіант: 20



Лістинг коду програми

```
package sample;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import java.awt.geom.Point2D;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Main extends JPanel implements ActionListener {
    Timer timer;

    private static int maxWidth = 800;
    private static int maxHeight = 800;

    private double angle = 0;

    private double scale = 0.1;
    private double delta = 0.01;

    public Main() {
        timer = new Timer(100, this);
        timer.start();
    }

    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        RenderingHints rh = new RenderingHints(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        rh.put(RenderingHints.KEY_RENDERING, RenderingHints.VALUE_RENDER_QUALITY);
        g2d.setRenderingHints(rh);

        g2d.setBackground(new Color(0, 128, 255));
        g2d.clearRect(0, 0, maxWidth, maxHeight);

        g2d.setColor(new Color(255, 0, 0));
        BasicStroke bs1 = new BasicStroke(16, BasicStroke.CAP_ROUND,
            BasicStroke.JOIN_MITER);
        g2d.setStroke(bs1);
        g2d.drawRect(0, 0, maxWidth, maxHeight);

        g2d.translate(maxWidth / 2, maxHeight / 2);
        double centerX = 1;
        double centerY = 1;
        g2d.rotate(angle, centerX, centerY);
        g2d.scale(scale, scale);

        g2d.setColor(new Color(0, 128, 0));
        GeneralPath tr1 = drawTriangle(0, 40, 200, 60);
        g2d.fill(tr1);
        GeneralPath tr2 = drawTriangle(0, 120, 200, 60.1);
        g2d.fill(tr2);
        GeneralPath tr3 = drawTriangle(0, 200, 200, 60.15);
        g2d.fill(tr3);

        g2d.setColor(new Color(128, 64, 0));

        g2d.fillRect(-30, 400, 60, 100);

        GradientPaint gp = new GradientPaint(1, 1, Color.YELLOW, 2, 2, Color.BLUE,
true);
        g2d.setPaint(gp);

        g2d.fillRect(0, 140, 20, 20);
        g2d.fillRect(0, 240, 20, 20);
    }

    private GeneralPath drawTriangle(double x1, double y1, double x2, double y2) {
        GeneralPath gp = new GeneralPath();
        gp.moveTo(0, 0);
        gp.lineTo(x1, y1);
        gp.lineTo(x2, y2);
        gp.closePath();
        return gp;
    }
}
```

```

g2d.fillRect(0, 340, 20, 20);
g2d.fillRect(50, 210, 20, 20);
g2d.fillRect(-60, 210, 20, 20);
g2d.fillRect(-60, 310, 20, 20);
g2d.fillRect(60, 310, 20, 20);

}

public void actionPerformed(ActionEvent e) {

    if (scale < 0.01) {
        delta = -delta;
    } else if (scale > 0.99) {
        delta = -delta;
    }
    angle -= 0.1;
    scale += delta;
    repaint();
}

GeneralPath drawTriangle(int x, int y, double length, double angle){
    Point2D[] points = new Point2D[] {
        new Point2D.Double(x,y),
        new Point2D.Double(x-(length * Math.tan(angle)), y+length),
        new Point2D.Double(x+(length * Math.tan(angle)), y+length)
    };
    GeneralPath tr = new GeneralPath();
    tr.moveTo(points[0].getX(), points[0].getY());
    tr.lineTo(points[1].getX(), points[1].getY());
    tr.lineTo(points[2].getX(), points[2].getY());
    tr.closePath();
    return tr;
}

public static void main(String[] args) {
    JFrame frame = new JFrame("lab2");
    frame.add(new Main());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(maxWidth, maxHeight);
    frame.setResizable(false);
    frame.setVisible(true);
    Dimension size = frame.getSize();
    Insets insets = frame.getInsets();
    maxWidth = size.width - insets.left - insets.right - 1;
    maxHeight = size.height - insets.top - insets.bottom - 1;
}
}

```

Результат

