



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-81

Дикий Ілля
(прізвище, ім'я, по батькові)

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

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

варіант № 4

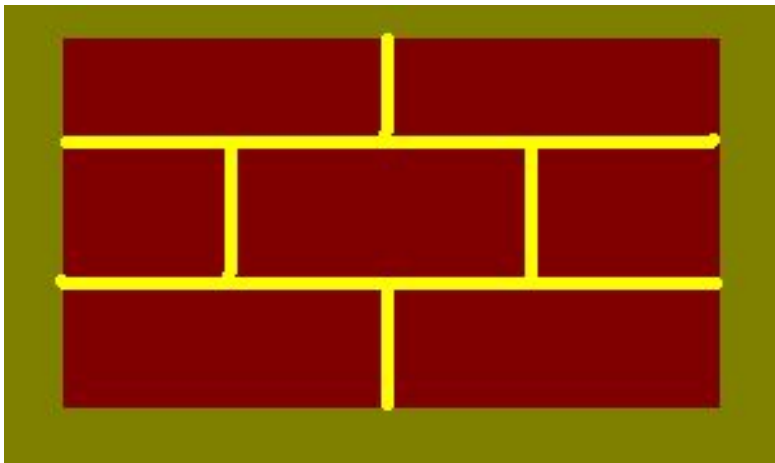
Київ 2021

Завдання

За допомогою Java 2D намалювати картинку з лабораторної роботи No1 (за варіантом).

Додатково виконати:

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 **JOIN_BEVEL**.
4. Виконати анімацію малюнку, а саме масштабування та рух по квадрату проти годинникової стрілки. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов'язкова, якщо не передбачено варіантом.



Лістинг коду програми

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import javax.swing.*;

@SuppressWarnings("serial")
public class FirstTask extends JPanel implements ActionListener {
    private static int maxWidth;
    private static int maxHeight;
    Timer timer;
    private double scale = 1;
    private double delta = 0.01;
    private double dx = 1;
    private double tx = 300;
    private double dy = 1;
    private double ty = 200;
    private double edge = 400;
    private double sx = 300;
    private double sy = 200;

    public FirstTask() {
        timer = new Timer(10, this);
        timer.start();
    }

    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;

        // створюємо фон
        g2d.setBackground(new Color(128, 128, 0));
        g2d.clearRect(0, 0, maxWidth, maxHeight);
        // малюємо рамку
        g2d.setColor(Color.ORANGE);
        BasicStroke bs = new BasicStroke(12, BasicStroke.CAP_ROUND,
            BasicStroke.JOIN_BEVEL);
        g2d.setStroke(bs);
        g2d.drawRect(20, 20, 960, 760);
    }
}
```

```

g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);

// застосовуємо анімацію
g2d.translate(tx, ty);
g2d.scale(scale, scale);
// малюнок з першої лабораторної
g2d.setBackground(new Color(139, 0, 0));
g2d.clearRect(-250, -150, 500, 300);

g2d.setColor(Color.YELLOW);
g2d.setStroke(new BasicStroke(10));
g2d.drawLine(-250, 50, 250, 50);
g2d.drawLine(-250, -50, 250, -50);
g2d.drawLine(-100, -50, -100, 50);
g2d.drawLine(100, -50, 100, 50);
g2d.drawLine(0, 50, 0, 150);
g2d.drawLine(0, -50, 0, -150);

// додаткова фігура побудована ламаною та залита градієнтом
GradientPaint gp = new GradientPaint(0, 0,
    Color.PINK, 50, 50, Color.ORANGE, true);
g2d.setPaint(gp);

double points[][] = {
    {-90, -40}, {0, -25}, {90, -40}, {90, 40}, {0, 25}, {-90 ,
40}, {-90, -40}
};

GeneralPath cRect = new GeneralPath();
cRect.moveTo(points[0][0], points[0][1]);
for (int i = 1; i < points.length; i++)
    cRect.lineTo(points[i][0], points[i][1]);
cRect.closePath();
g2d.fill(cRect);
}

public static void main(String[] args) {
    JFrame frame = new JFrame("First task");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1000, 850);
}

```

```

        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new FirstTask());
        frame.setVisible(true);

        Dimension size = frame.getSize();
        Insets insets = frame.getInsets();
        maxWidth = size.width - insets.left - insets.right;
        maxHeight = size.height - insets.top - insets.bottom;

    }

    public void actionPerformed(ActionEvent e) {
        if ( scale < 0.1 ) {
            delta = -delta;
        } else if (scale > 0.99) {
            delta = -delta;
        }

        if ( tx <= sx && ty < sy+edge) {
            ty += dy;
        } else if (ty >= sy+edge && tx < sx+edge) {
            tx += dx;
        } else if (ty > sy && tx >= sx+edge){
            ty -= dy;
        } else if (tx > sx && ty >= sy) {
            tx -= dx;
        }

        scale += delta;

        repaint();
    }
}

```

Результат

