



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: «Побудова та анімація зображень за допомогою Java2D»

Виконав  
студент III курсу  
каф. ПЗКС ФПМ  
групи КП-93

Інюшев Артем  
Владиславович

Перевірила

старший  
викладач

Шкурат О. С.

Київ 2022

Завдання: За допомогою Java 2D намалювати картинку з лабораторної роботи №1 (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 № 1).
3. На достатній відстані від побудованого малюнку намалювати прямокутну рамку, всередині якої відбуватиметься анімація. Тип лінії рамки задано за варіантом (JOIN\_ROUND).
4. Виконати анімацію малюнку, за варіантом. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов'язкова, якщо не передбачено варіантом.

## Варіант 9

Програмний код:

```
package com.example.lab2;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;
import javafx.scene.shape.Polyline;
import javafx.scene.text.Text;
import javafx.stage.Stage;

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

public class HelloApplication extends JPanel implements ActionListener {
    private static int maxWidth;
    private static int maxHeight;

    private int tx=550, ty=350, maxx=200, maxy=100, angle=0;
    private char dir='l';
    public static void main (String args[]) {
        JFrame frame = new JFrame("Привіт, Java 2D!");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1200, 800);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new HelloApplication());
        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;
    }
}
```

```

public HelloApplication(){
    new Timer(100, this).start();
}
public void paint(Graphics g){
    Graphics2D g2d = (Graphics2D) g;
    java.awt.RenderingHints rh = new
java.awt.RenderingHints(java.awt.RenderingHints.KEY_ANTIALIASING,
java.awt.RenderingHints.VALUE_ANTIALIAS_ON);
    rh.put(java.awt.RenderingHints.KEY_RENDERING, java.awt.RenderingHints.VALUE_RENDER_QUALITY);
    g2d.setRenderingHints(rh);

    g2d.setBackground(Color.RED);
    g2d.clearRect(0, 0, maxWidth, maxHeight);

    g2d.translate(tx, ty);
    g2d.scale(0.5, 0.5);
    g2d.rotate(Math.toRadians(angle), 600,600);

    paintFlower(g2d);

    g2d.rotate(Math.toRadians(-angle), 600,600);
    g2d.scale(2, 2);
    g2d.translate(-tx, -ty);

    g2d.setColor(Color.WHITE);
    BasicStroke bs1 = new BasicStroke(16, BasicStroke.CAP_ROUND,
        BasicStroke.JOIN_ROUND);
    g2d.setStroke(bs1);
    g2d.drawRect(8+20, 8+20, maxWidth-16-40, maxHeight-16-40);
}

private void paintFlower(Graphics2D g2d){
    g2d.setColor(Color.YELLOW);
    g2d.fillRect(320-6, 350, 12, 220);
    //g2d.setColor(Color.YELLOW);
    GradientPaint gp = new GradientPaint(5, 25,
        new Color(255,255,0), 20, 2, new Color(0,0,255), true);
    g2d.setPaint(gp);
    g2d.fill(takePolygon(new double[][]{
        {200.0, 280.0},
        {230.0, 200.0},
        {400.0, 190.0},
        {440.0, 290.0},
        {320.0, 350.0},
    })));
    g2d.setColor(Color.YELLOW);
    g2d.fill(takePolygon(new double[][]{
        {200.0, 280.0},
        {230.0, 200.0},
        {90.0, 210.0}
    })));
    g2d.fill(takePolygon( new double[][]{
        {230.0, 200.0},
        {400.0, 190.0},
        {310.0, 50.0}
    })));
    g2d.fill(takePolygon(new double[][]{
        {400.0, 190.0},
        {440.0, 290.0},
        {600.0, 195.0}
    })));
    g2d.fill(takePolygon(new double[][]{
        {440.0, 290.0},
        {320.0, 350.0},
        {500.0, 450.0}
    })));
    g2d.fill(takePolygon(new double[][]{
        {200.0, 280.0},
        {320.0, 350.0},
        {150.0, 420.0}
    })));
    g2d.setColor(new Color(0,100,0));
    //g2d.fillOval(600-6, 600-6, 12, 12);
}

```

```

    }

    private GeneralPath takePolygon(double[][] points){
        GeneralPath pol = new GeneralPath();
        pol.moveTo(points[0][0], points[0][1]);
        for (int k = 1; k < points.length; k++)
            pol.lineTo(points[k][0], points[k][1]);
        pol.closePath();
        return pol;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("timer");
        angle=angle+10;

        if (dir=='r')
            if (tx<maxx)
                tx+=10;
            else {
                tx=maxx;
                dir='d';
            }
        else if (dir=='d')
            if (ty<maxy)
                ty+=10;
            else {
                ty=maxy;
                dir='l';
            }
        else if (dir=='l')
            if (tx>0)
                tx-=10;
            else {
                tx=0;
                dir='u';
            }
        else if (dir=='u')
            if (ty>0)
                ty-=10;
            else {
                ty=0;
                dir='r';
            }

        repaint();
    }
}

```

## Результати

