



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-83

Палій Дмитро Володимирович
(прізвище, ім'я, по батькові)

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

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

варіант № 16

Варіант завдання

Завдання: За допомогою Java 2D намалювати картинку з лабораторної роботи №1.

Додатково виконати:

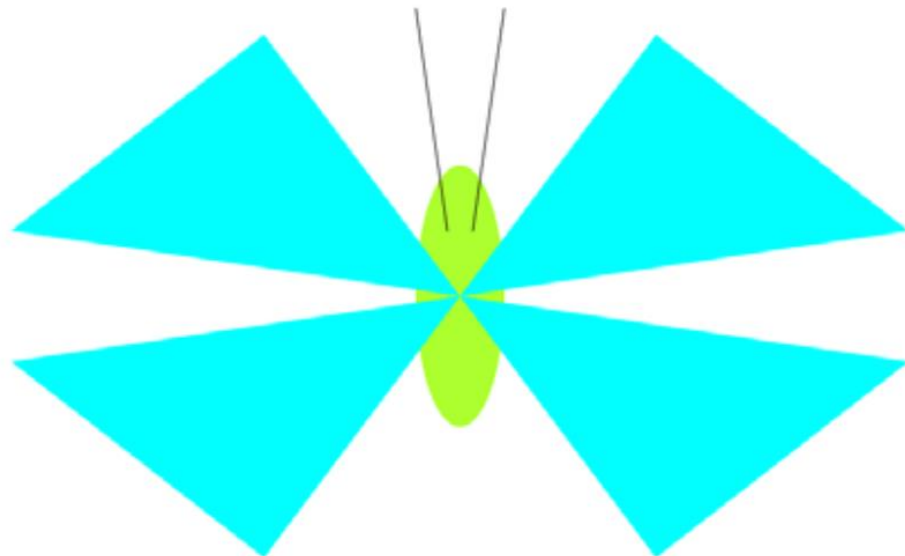
1. Хоча б 1 стандартний примітив, та хоча б 1 фігуру, побудовану по точкам (ламаную).
2. Хоча б 1 фігуру залити градієнтною фарбою за вибором (в цьому випадку колір може не співпадати з варіантом із лабораторної роботи № 1).
3. На достатній відстані від побудованого малюнку намалювати прямокутну рамку,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

Типи анімації:

4. Рух по квадрату за годинниковою стрілкою
10. Масштабування

Варіант:

16.



№	Типи анімації	Тип лінії рамки
---	---------------	-----------------

16	4, 10	JOIN_BEVEL
----	-------	------------

Лістинг коду програми

```
package com.company;

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 Moth extends JPanel implements ActionListener {
    int padding = 50;
    int r = 100;
    double dv = 5;
    double scale = 1;
    double delta = 0.02;
    double points[][] = {
        {0, 0}, {160, -160}, {300, -60},
```

```

        {0, 0}, {-160, -160}, {-300, -60},
        {0, 0}, {160, 160}, {300, 60},
        {0, 0}, {-160, 160}, {-300, 60}
    };

    Timer timer;

    private double dx = dv;
    private double dy = 0;
    private double tx = -r;
    private double ty = -r;
    private static int maxWidth;
    private static int maxHeight;

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

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

        g2d.setBackground(Color.black);
        g2d.setColor(Color.RED);
        g2d.setStroke(new BasicStroke(10, BasicStroke.CAP_BUTT,
BasicStroke.JOIN_BEVEL));
        g2d.clearRect(0, 0, maxWidth + 1, maxHeight + 1);

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

        g2d.drawRect(padding, padding, maxWidth - 2 * padding, maxHeight - 2 *
padding);

        g2d.translate(maxWidth/2, maxHeight/2);
        g2d.translate(tx, ty);

        // Lab 2 picture
        g2d.setColor(Color.decode("#C7EA46"));
        g2d.fillOval(-25, -100, 50, 200);

        g2d.setColor(Color.GREEN);
        g2d.setStroke(new BasicStroke(5, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_BEVEL));
        g2d.drawLine(-10, -60, -30, -160);
    }

```

```

g2d.drawLine(10, -60, 30, -160);

RadialGradientPaint gp = new RadialGradientPaint(new Point2D.Double(0,0),
    25,
    new Point2D.Double(0, 0),
    new float[] { 0.0f, 1.0f },
    new Color[] { Color.yellow, Color.blue },
    RadialGradientPaint.CycleMethod.REFLECT);

g2d.setPaint(gp);
GeneralPath wings = new GeneralPath();
wings.moveTo(points[0][0], points[0][1]);
for (int k = 1; k < points.length; k++)
    wings.lineTo(points[k][0], points[k][1]);
wings.closePath();
g2d.scale(scale, 0.99);
g2d.fill(wings);
//
}

public static void main(String[] args) {
    JFrame frame = new JFrame("lab2");
    frame.add(new Moth());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1280, 720);
    frame.setResizable(false);
    frame.setLocationRelativeTo(null);
    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;

}

// Цей метод буде викликано щоразу, як спрацює таймер
public void actionPerformed(ActionEvent e) {
    if ( scale < 0.33 || scale > 0.99) {
        delta = -delta;
    }

    if ( tx < -r ) {
        tx = -r;
        dx = 0;
        dy = -dv;
    } else if ( tx > r ) {
        tx = r;
        dx = 0;

```

```

        dy = dv;
    }

    if ( ty < -r ) {
        ty = -r;
        dy = 0;
        dx = dv;
    } else if ( ty > r ) {
        ty = r;
        dy = 0;
        dx = -dv;
    }
    scale += delta;
    tx += dx;
    ty += dy;

    repaint();
}
}

```

Результат

