



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”

Виконала
студентка III курсу
групи КП-82

Лахман Ксенія
(прізвище, ім'я, по батькові)

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

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

варіант № 11

Варіант завдання

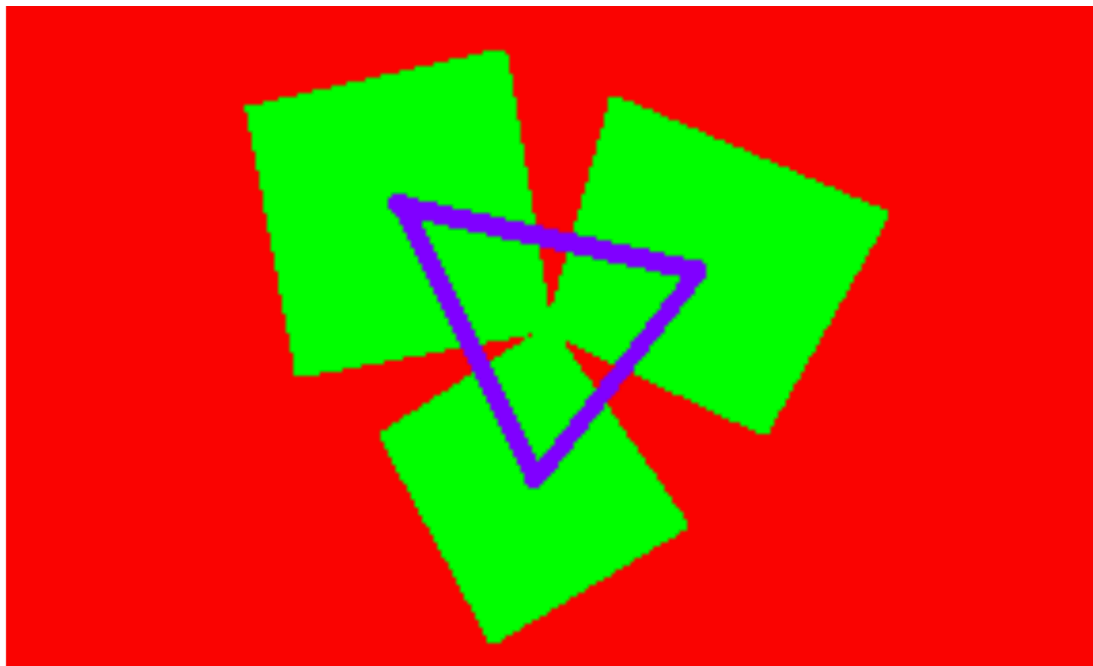
Завдання:

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.
4. Виконати анімацію малюнку, за варіантом. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов'язкова, якщо не передбачено варіантом.

Варіант:



Лістинг коду програми

```
package com.company;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import java.awt.geom.Line2D;

public class Main extends JPanel implements ActionListener {
    private static int maxWidth;
    private static int maxHeight;
    private static final int imageHeight = 190;
    private static final int imageWidth = 200;
    Timer timer;
    private float alpha = 1;
    private float delta = 0.01f;
    private int dx = 1;
    private int tx = 0;
    private int dy = 1;
    private int ty = 0;
    private static final double[][] spaceShipPoints = {
        {80.0, 0.0}, //a2
        {100.0, 85.0}, //a3

        {120.0, 10.0}, //b2
        {200.0, 40.0}, //b1
        {170.0, 110.0}, //b0
        {100.0, 85.0}, //b3

        {150.0, 140.0}, //c0
        {90.0, 190.0}, //c1
        {40.0, 130.0}, //c2
        {100.0, 85.0}, //c3

        {25.0, 100.0}, //a0
        {0.0, 20.0}, //a1
        {80.0, 0.0} //a2
    };

    private static final double[][] centerPoints = {
        {50.0, 50.0},
        {150.0, 60.0},
        {95.0, 138.0},
        {50.0, 50.0}
    };

    private static final int [][] stars = {
        {291, 46}, {146, 95}, {56, 291}, {109, 392}, {18, 62}, {478,
374}, {420, 341}, {41, 284}, {304, 320}, {210, 253}, {180, 312},
        {486, 463}, {354, 315}, {449, 60}, {64, 20}, {324, 396}, {224,
179}, {445, 282}, {185, 388}, {438, 450}, {92, 181}, {46, 283},
        {290, 251}, {130, 268}, {379, 119}, {387, 418}, {249, 430}, {12,
30}, {308, 244}, {351, 182}, {315, 466}, {34, 150}, {331, 415},
        {266, 427}, {188, 201}, {218, 112}, {111, 177}, {161, 17}, {481,
355}, {488, 122}, {495, 377}, {56, 53}, {342, 300}, {247, 110},
        {354, 388}, {33, 445}, {460, 414}, {286, 177}, {132, 454}, {487,
440}, {426, 214}, {20, 37}, {138, 376}, {110, 287}, {481, 43},
        {119, 333}, {193, 57}, {229, 365}, {372, 93}, {272, 346}, {384,
430}, {329, 368}, {103, 87}, {427, 379}, {305, 143}, {129, 271},
        {36, 201}, {471, 256}, {286, 445}, {266, 274}, {481, 167}, {38,
362}, {75, 425}, {79, 14}, {345, 14}, {268, 78}, {436, 30},
```

```

};

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

public static void main(String[] args) {
    JFrame frame = new JFrame("Lab 2");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1000, 500);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    frame.add(new Main());
    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 void paintSpaceship(Graphics2D g2d, int marginX, int marginY) {
    GradientPaint gp = new GradientPaint(5, 7, Color.GREEN, 100, 70,
    Color.BLUE, true);
    g2d.setPaint(gp);

    GeneralPath image = new GeneralPath();
    image.moveTo(marginX + spaceShipPoints[0][0], marginY +
    spaceShipPoints[0][1]);
    for (int k = 1; k < spaceShipPoints.length; k++) {
        image.lineTo(marginX + spaceShipPoints[k][0], marginY +
    spaceShipPoints[k][1]);
    }
    image.closePath();
    g2d.fill(image);
    g2d.setColor(Color.YELLOW);
    g2d.setStroke(new BasicStroke(4f));

    g2d.draw(new Line2D.Double(marginX + centerPoints[0][0], marginY +
    centerPoints[0][1], marginX + centerPoints[1][0], marginY +
    centerPoints[1][1]));
    g2d.draw(new Line2D.Double(marginX + centerPoints[1][0], marginY +
    centerPoints[1][1], marginX + centerPoints[2][0], marginY +
    centerPoints[2][1]));
    g2d.draw(new Line2D.Double(marginX + centerPoints[2][0], marginY +
    centerPoints[2][1], marginX + centerPoints[3][0], marginY +
    centerPoints[3][1]));
    g2d.fillOval(marginX + 90, marginY + 78, 20, 20);
}

public void paintStars(Graphics2D g2d) {
    for (int i = 0; i < stars.length; i++) {
        int width = getRandomNumber(7, 10);
        int height = getRandomNumber(7, 10);
        g2d.fillOval(stars[i][0], stars[i][1], width, height);
    }
}

public void paint(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;
    g2d.setBackground(Color.black);
    g2d.clearRect(0, 0, maxWidth, maxHeight);
}

```

```

        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY);
        //
        paintSpaceship(g2d, 150, 150);
        paintStars(g2d);
        //
        int margin = 5;
        g2d.setStroke(new BasicStroke(5, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_MITER));
        //
        g2d.drawRect(maxWidth/2, 0 + margin,maxWidth / 2 - margin, maxHeight
- 2*margin);
        //
        g2d.translate(maxWidth / 2, 0);
        //
        g2d.translate(tx, ty);
        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER,
alpha));
        paintSpaceship(g2d,3, 3);
    }

    private void initialMove() {dx = 0; dy = 0;}
    private void moveUp()    { dx = 0;  dy = -1; }
    private void moveDown() { dx = 0;  dy = 1; }
    private void moveLeft() { dx = -1; dy = 0; }
    private void moveRight(){ dx = 1;  dy = 0; }
    private void makeMove() { tx += dx; ty += dy;}

    @SuppressWarnings("all")
    public void actionPerformed(ActionEvent e) {
        if (alpha < 0.01f || alpha > 0.99f) {
            delta = -delta;
        }

        int minTx = 1;
        int minTy = 1;
        int maxTx = maxWidth/2 - imageWidth - 1;
        int maxTy = maxHeight - imageHeight - 1;

        initialMove();

        if (tx <= minTx || ty <= minTy) {
            moveDown();
        }
        if (ty >= maxTy) {
            moveRight();
        }
        if (tx >= maxTx) {
            moveUp();
        }
        if (ty <= minTy && tx > minTx) {
            moveLeft();
        }
        makeMove();
        alpha += delta;
        repaint();
    }

    public int getRandomNumber(int min, int max) {
        return (int) ((Math.random() * (max - min)) + min);
    }
}

```

Результат

