МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики
Інженерія програмного забезпечення

ЛАБОРАТОРНА РОБОТА №2
з дисципліни *"Математичні та алгоритмічні основи комп'ютерної графіки"*

Виконав:
студент 3-го курсу, групи КП-83, спеціальності 121 – Інженерія програмного забезпечення *Медведєв Ілля Анатолійович*

Зарахована:
Шкурат О. С.

Київ – 2021

Мета: Ознайомитися з можливостями побудови зображень та їх анімації у Java2D

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 13:

Тип анімації:

Рух по квадрату проти годинникової стрілки та зміна прозорості,

Текст коду програм

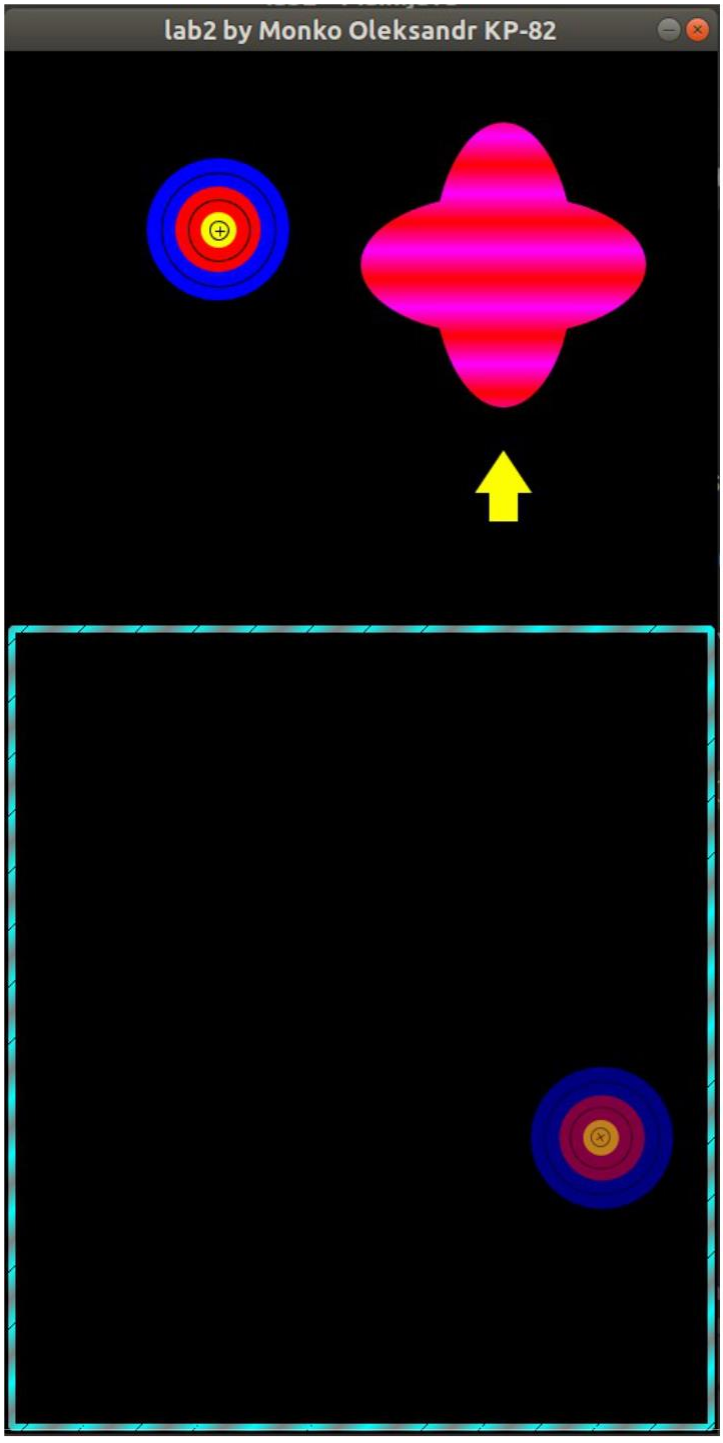
App.java

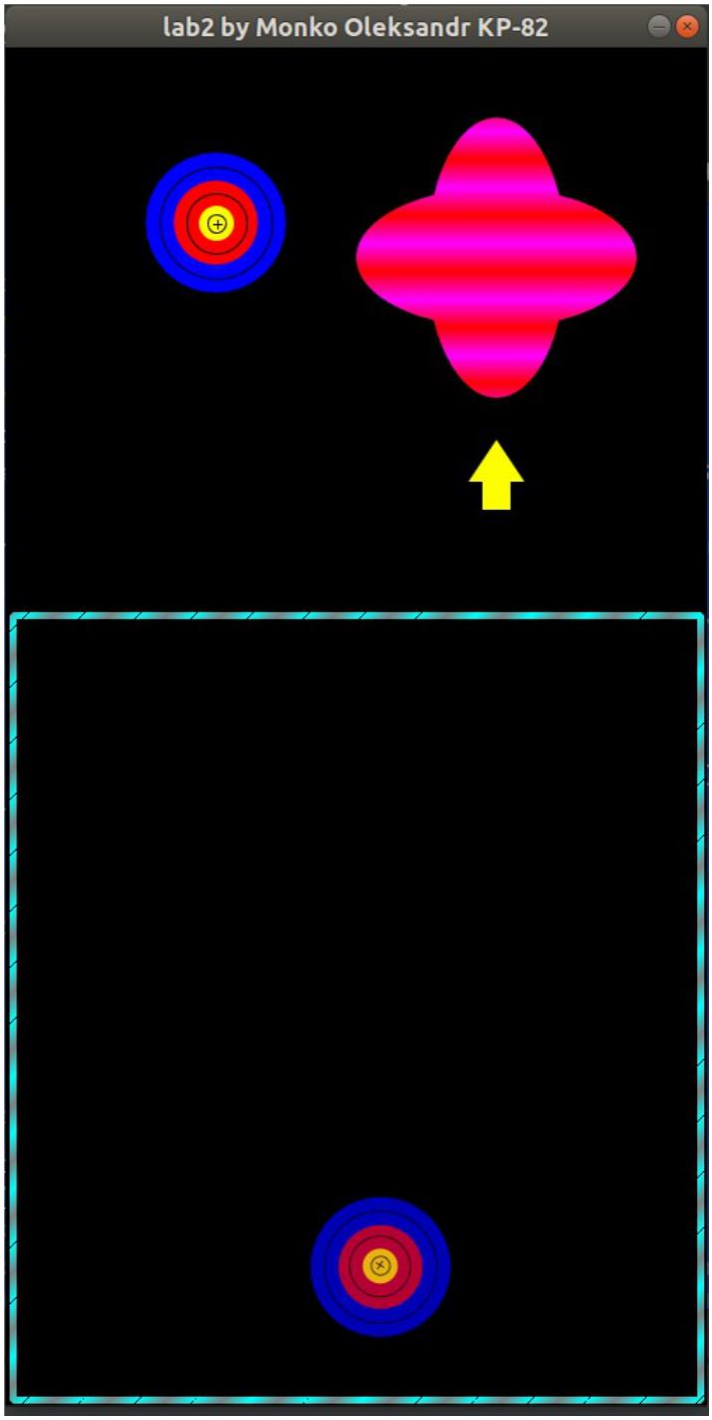
```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.*;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;
@SuppressWarnings("serial")
class App extends JPanel implements ActionListener {

    Timer timer;

    private static int maxWidth;
    private static int maxHeight;

    private static int alpha = 0;
    private static boolean alphaInc = true;
    private static double angle = 0;

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

    public void paint(Graphics g) {
        Graphics2D graphics2d = (Graphics2D) g;
        RenderingHints hints = new RenderingHints(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        hints.put(RenderingHints.KEY_RENDERING, RenderingHints.VALUE_RENDER_QUALITY);
        graphics2d.setRenderingHints(hints);


        fillBlackBackground(graphics2d);

        drawPicture(graphics2d);
        drawComplex(graphics2d);
        drawPrimitive(graphics2d);
        drawAnimation(graphics2d);
    }

    private void drawPicture(Graphics2D g) {
        drawPicture(g, 255);
    }

    private void drawPicture(Graphics2D g, int alpha) {
        drawBlueCircle(g, alpha);
        drawRedCircle(g, alpha);
        drawYellowCircle(g, alpha);
        drawBlackCircles(g, alpha);
        drawTarget(g, alpha);
    }

    private void drawBlueCircle(Graphics2D g, int alpha) { g.setColor(new
        Color(Color.BLUE.getRed(), Color.BLUE.getGreen(),
Color.BLUE.getBlue(), alpha));
        g.fillOval(100, 75, 100, 100);
    }
```

```
    private void drawRedCircle(Graphics2D g, int alpha) {
        g.setColor(new Color(Color.RED.getRed(), Color.RED.getGreen(),
Color.RED.getBlue(), alpha));
        g.fillOval(120, 95, 60, 60);
    }

    private void drawYellowCircle(Graphics2D g, int alpha) {
        g.setColor(new Color(Color.YELLOW.getRed(), Color.YELLOW.getGreen(),
Color.YELLOW.getBlue(), alpha));
        g.fillOval(Math.round(150 - 12.5f), Math.round(125 - 12.5f), 25, 25);
    }

    private void drawBlackCircles(Graphics2D g, int alpha) {
        int[] radiuses = new int[] { 80, Math.round(25 + 17.5f), Math.round(12.5f) };
        for (int i = 0; i < radiuses.length; i++) {
            g.setColor(new Color(Color.BLACK.getRed(),
Color.BLACK.getGreen(), Color.BLACK.getBlue(), alpha));
            g.drawOval(150 - radiuses[i] / 2, 125 - radiuses[i] / 2,
radiuses[i], radiuses[i]);
        }
    }

    private void drawTarget(Graphics2D g, int alpha) {
        g.setColor(new Color(Color.BLACK.getRed(),
Color.BLACK.getGreen(), Color.BLACK.getBlue(), alpha));
        g.drawLine(150 - 2, 125 + 1, 150 + 4, 125 + 1);
        g.drawLine(150 + 1, 125 - 2, 150 + 1, 125 + 4);
    }

    private void drawPrimitive(Graphics2D g) {
        g.setColor(Color.WHITE);
        GradientPaint gp = new GradientPaint(0, 0, Color.RED, 0, 20, Color.MAGENTA,
true);
        g.setPaint(gp);
        g.fillOval(250, 100, 200, 100);
        g.fillOval(300, 50, 100, 200);
    }

    private void drawComplex(Graphics2D g) {
        g.translate(20, 300);
        int[][] points = { { 0, 60 }, { 20, 30 }, { 40, 60 }, { 30, 60 }, { 30, 80 },
{ 10, 80 }, { 10, 60 },
                {0,60}};
        int[] offset = {310, -50};

        g.setColor(Color.YELLOW);

        GeneralPath house = new GeneralPath();
        house.moveTo(points[0][0] + offset[0], points[0][1] + offset[1]);

        for (int i = 1; i < points.length; i++) {
            house.lineTo(points[i][0] + offset[0], points[i][1] + offset[1]);
        }

        house.closePath();
        g.fill(house);
        g.translate(-20, -300);
    }

    private void drawAnimation(Graphics2D g) {
        int offsetX = 0;
        int offsetY = 400;
        int borderWidth = 5;
        Stroke defaultStroke = g.getStroke();
        BasicStroke bs = new BasicStroke(borderWidth, BasicStroke.CAP_BUTT,
```

```
 BasicStroke.JOIN_BEVEL);
        g.setStroke(bs);
        GradientPaint gp = new GradientPaint(20, 20, Color.gray, 10, 10, Color.cyan,
true);
        g.setPaint(gp);
        g.drawRect(offsetX + borderWidth, offsetY + borderWidth, maxWidth - offsetX -
10, maxHeight - offsetY - 10);
        g.setStroke(defaultStroke);
        g.translate(offsetX, offsetY);

        g.rotate(angle, (maxWidth - offsetX - 1) / 2, (maxHeight - offsetY - 1) / 2);
        drawPicture(g, alpha);
    }

    private void fillBlackBackground(Graphics2D g)
        { g.setBackground(Color.BLACK);
        g.clearRect(0, 0, maxWidth, maxHeight);
    }

    private static void setSizes(JFrame frame) {
        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) {
        angle += 0.02;
        if (alpha >= 255) {
            alphaInc = false;
        } else if (alpha <= 0) {
            alphaInc = true;
        }

        if (alphaInc) {
            alpha++;
        } else {
            alpha--;
        }
        repaint();
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("lab2 by Monko Oleksandr KP-82");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 1000);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);

        frame.add(new App());
        frame.setVisible(true);

        setSizes(frame);
    }

}
```

Результати роботи програми





Висновки

Виконавши дану лабораторну роботу, я навчився працювати з бібліотекою JavaFX та ознайомився з можливостями побудови зображень та їх анімації у Java2D.