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”

Варіант 7

Виконав
студент 3 курсу
групи КП-81
Каснер Максим

Київ 2021

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 7 : типи анімації - 3(Рух по квадрату проти годинникової стрілки), 5(Обертання навколо центру малюнка за годинниковою стрілкою); тип лінії рамки – JOIN_BEVEL

Код програми

Picture1.java

```
import java.awt.*;
import java.awt.geom.GeneralPath;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Picture1 extends JPanel{

    private static int maxWidth;
    private static int maxHeight;

    private void paintBug(Graphics2D g2d) {
```

```
        int[] x1 = {100, 190, 140, 50};
        int[] y1 = {20, 55, 80 ,60};
        GeneralPath gp1 = new GeneralPath();
        gp1.moveTo(x1[0], y1[0]);
        for (int i = 1; i < x1.length; i++) {
            gp1.lineTo(x1[i], y1[i]);
        }
        gp1.closePath();
        g2d.setColor(new Color(0,255,1 ));
        g2d.fill(gp1);

        int[] x2 = x1.clone();
        int[] y2 = y1.clone();
        x2[0] = x2[3] + 30;
        x2[1] = x2[2] + 15;
        y2[0] = y2[3] + 60;
        y2[1] = y2[2] + 30;

        GeneralPath gp2 = new GeneralPath();
        gp2.moveTo(x2[0], y2[0]);
        for (int i = 1; i < x1.length; i++) {
            gp2.lineTo(x2[i], y2[i]);
        }
        gp2.closePath();
        g2d.fill(gp2);
        g2d.draw(gp2);

        GeneralPath gp3 = new GeneralPath();
        gp3.moveTo(x1[1] - 10, y1[1] + 17);
        gp3.lineTo(x1[2] + 10, y2[2]);
        gp3.lineTo(x2[1] + 7, y2[1] - 10);
        gp3.closePath();
        g2d.setColor(Color.yellow);
        g2d.fill(gp3);

        g2d.setColor(new Color(4, 126, 6));
        g2d.fillRect(80, 80 , 7, 7);
        g2d.fillRect(90, 50 , 7, 7);

        g2d.setStroke(new BasicStroke(3));
        g2d.setColor(Color.black);
        g2d.drawLine(x1[2], y1[2], x1[3], y1[3]);

        g2d.setStroke(new BasicStroke(5));
        g2d.drawLine(63, 90, 30, 100);
        g2d.drawLine(63, 47, 30, 15);
```

```
    }

    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        RenderingHints rh = new RenderingHints(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
        rh.put(RenderingHints.KEY_RENDERING,
                RenderingHints.VALUE_RENDER_QUALITY);
        g2d.setRenderingHints(rh);

        g2d.setBackground(new Color(0, 128, 129));
        g2d.clearRect(0, 0, maxWidth, maxHeight);

        paintBug(g2d);
    }

    public static void main(String[] args) {

        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 500);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new Picture1());
        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;
    }
}
```

Picture2.java

```
import java.awt.*;
import java.awt.geom.GeneralPath;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Picture2 extends JPanel{

    private static int maxWidth;
    private static int maxHeight;
```

```
    private void paintFence(Graphics2D g2d) {

        GeneralPath gp1 = new GeneralPath();
        gp1.moveTo(50, 250);
        g2d.setColor(Color.black);
        for(int i = 0; i < 8; i++) {
            gp1.lineTo(50 + 50 * i, 100);
            gp1.lineTo(75 + 50 * i, 50);
            gp1.lineTo(100 + 50 * i, 100);
            gp1.lineTo(100 + 50 * i, 250);
            gp1.moveTo(100 + 50 * i, 250);
        }
        gp1.closePath();
        GradientPaint gradientPaint = new GradientPaint(5 ,25, Color.yellow, 20, 7,
Color.lightGray, true);
        g2d.setPaint(gradientPaint);
        g2d.fill(gp1);

        g2d.setColor(Color.black);
        g2d.setStroke(new BasicStroke(2));
        for(int i = 0; i < 8; i++) {
            g2d.drawLine(50 + 50 * i, 250, 50 + 50 * i, 100);
            g2d.drawLine(100 + 50 * i, 250, 100 + 50 * i, 100);
            g2d.drawLine(50 + 50 * i, 100, 75 + 50 * i, 50);
            g2d.drawLine(75 + 50 * i, 50, 100 + 50 * i, 100);
        }
    }

    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        RenderingHints rh = new RenderingHints(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
        rh.put(RenderingHints.KEY_RENDERING,
                RenderingHints.VALUE_RENDER_QUALITY);
        g2d.setRenderingHints(rh);

        g2d.setBackground(new Color(120, 108, 129));
        g2d.clearRect(0, 0, maxWidth, maxHeight);

        paintFence(g2d);
    }

    public static void main(String[] args) {

        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 500);
```

```
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new Picture2());
        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;
    }
}
```

Picture3.java

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

public class Picture3 extends JPanel implements ActionListener {

    private static int maxWidth;
    private static int maxHeight;
    Timer timer;

    private int dx = 0;
    private int dy = 1;

    private double angle = 0;
    private int rectWidth = 20;
    private int rectHeight = 20;
    private int rectStartX = 90;
    private int rectStartY = 90;

    private static int frameWidth;
    private static int frameHeight;
    private int frameStartX = 100;
    private int frameStartY = 100;

    public Picture3() {
        timer = new Timer(10, this);
        timer.start();
    }

    private void paintPic(Graphics2D g2d) {
```

```
        g2d.setStroke(new BasicStroke(3, BasicStroke.CAP_ROUND, BasicStroke.JOIN_BEVEL));
        Rectangle frame = new Rectangle(frameStartX, frameStartY, frameWidth, frameHeight);
        g2d.draw(frame);

        g2d.setColor(Color.orange);

        Rectangle rectangle = new Rectangle(rectStartX, rectStartY, rectWidth, rectHeight);
        g2d.fill(rectangle);


        g2d.rotate(angle, maxWidth/2, maxHeight/2);
        g2d.fillRoundRect(maxWidth/2, maxHeight/2, 30, 40, 5, 5);
    }

    public void paint(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;
        RenderingHints rh = new RenderingHints(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
        rh.put(RenderingHints.KEY_RENDERING,
                RenderingHints.VALUE_RENDER_QUALITY);
        g2d.setRenderingHints(rh);

        g2d.setBackground(new Color(120, 108, 129));
        g2d.clearRect(0, 0, maxWidth, maxHeight);

        paintPic(g2d);
    }

    public static void main(String[] args) {

        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 500);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new Picture3());
        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;
        frameWidth = maxWidth - 100 * 2;
        frameHeight = maxHeight - 100 * 2;
    }
```

```
    @Override
    public void actionPerformed(ActionEvent e) {

        if(rectStartX <= frameStartX - rectWidth / 2 ) {
            if(rectStartY <= frameStartY - rectHeight / 2) {
                dx = 0;
                dy = 1;
            }
            else if(rectStartY >= frameStartY + frameHeight - rectHeight / 2) {
                dx = 1;
                dy = 0;
            }
        }
        else if(rectStartX >= frameStartX + frameWidth - rectWidth / 2) {
            if(rectStartY >= frameStartY + frameHeight - rectHeight / 2) {
                dx = 0;
                dy = -1;
            }
            else if(rectStartY <= frameStartY - rectHeight / 2) {
                dx = -1;
                dy = 0;
            }
        }

        rectStartX += dx;
        rectStartY += dy;
        angle += 0.01;
        repaint();
    }
}
```

Результати роботи програми :