



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”  
тема “Побудова та анімація зображень за допомогою Java2D”

Виконав(ла)  
студент(ка) III курсу  
групи КП-83  
Матіюк Дарина Андріївна

Зарахована  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20\_\_ р.  
Шкурат Оксаною Сергіївною

варіант №12

Київ 2021

### 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 (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.

**Варіант:**

12	1,8	JOIN ROUND
----	-----	------------

**Типи анімації:**

1. Рух по колу проти годинникової стрілки
8. Обертання навколо кута малюнка проти годинникової стрілки

## Лістинг коду програми

### Main.java

```
public class Main extends JPanel implements ActionListener {
    Timer timer;
    private static int maxWidth = 800;
    private static int maxHeight = 800;
    private double angle, rotAlpha = (Math.random() / 360) * 2 * Math.PI +
    (-Math.PI);
    private final double v = 1;
    private final int CX = 1;
    private final int CY = 1;
    private double sx = 1;
    private double tx = 1;
    private double sy = 1;
    private double ty = 1;

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

    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(255, 128, 64));
        g2d.clearRect(0, 0, maxWidth + 1, maxHeight + 1);
        g2d.setColor(new Color(0,128,128));
        BasicStroke bs = new BasicStroke(15, BasicStroke.CAP_ROUND,
        BasicStroke.JOIN_ROUND);
        g2d.setStroke(bs);
        g2d.drawRect(20, 20, maxWidth - 35, maxHeight - 35);
        g2d.translate(maxWidth / 2, maxHeight / 2);
        g2d.rotate(angle, CX + 50, CY + 50);
        g2d.translate(tx, ty);
        GradientPaint gp = new GradientPaint(5, 25, Color.yellow, 20, 2,
        Color.orange, true);
        g2d.setPaint(gp);
        double sunBody[][] = {
            {CX - 150, CY - 50},
            {CX - 100, CY - 120},
            {CX, CY - 150},
            {CX + 140, CY - 50},
            {CX + 130, CY + 40},
            {CX + 10, CY + 100},
            {CX - 110, CY + 50}
        };
        GeneralPath polyBody = new GeneralPath();
        polyBody.moveTo(sunBody[0][0], sunBody[0][1]);
        for (int k = 1; k < sunBody.length; k++) {
            polyBody.lineTo(sunBody[k][0], sunBody[k][1]);
        }
        polyBody.closePath();
        g2d.fill(polyBody);
    }
}
```

```

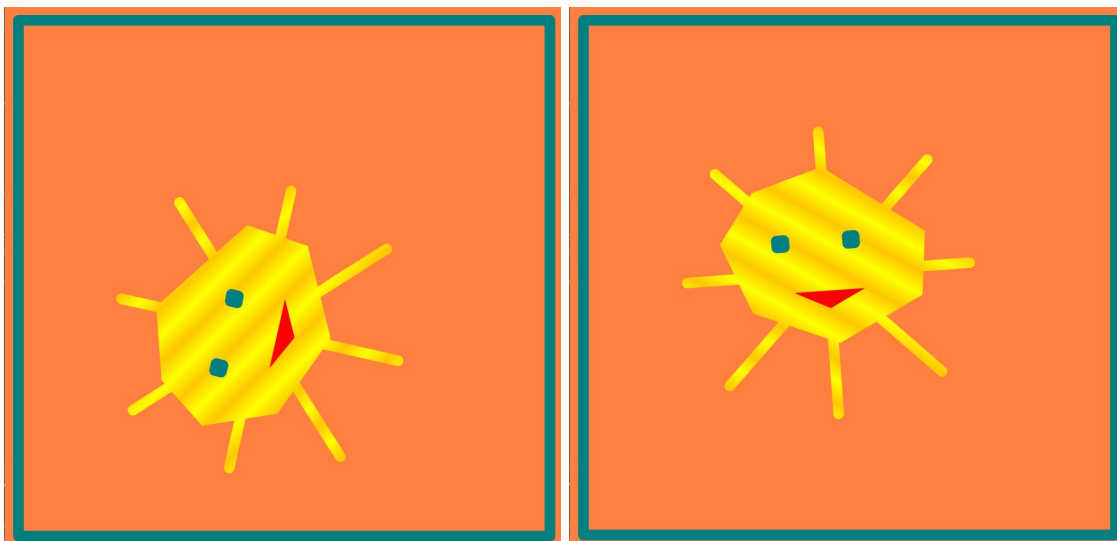
g2d.drawLine(CX, CY, CX + 200, CY);
g2d.drawLine(CX, CY, CX + 150, CY + 150);
g2d.drawLine(CX, CY, CX, CY + 200);
g2d.drawLine(CX, CY, CX + 150, CY - 150);
g2d.drawLine(CX, CY, CX - 200, CY);
g2d.drawLine(CX, CY, CX, CY - 200);
g2d.drawLine(CX, CY, CX - 150, CY + 150);
g2d.drawLine(CX, CY, CX - 150, CY - 150);
g2d.setColor(new Color(0,128,128));
g2d.drawRect(CX - 70, CY - 50, 10, 10);
g2d.drawRect(CX + 30, CY - 50, 10, 10);
g2d.setColor(new Color(255,0,0));
double sunMouth[][] = {
    {CX - 50, CY + 25},
    {CX + 50, CY + 25},
    {CX, CY + 50}
};
GeneralPath polyMouth = new GeneralPath();
polyMouth.moveTo(sunMouth[0][0], sunMouth[0][1]);
for (int k = 1; k < sunMouth.length; k++) {
    polyMouth.lineTo(sunMouth[k][0], sunMouth[k][1]);
}
polyMouth.closePath();
g2d.fill(polyMouth);
}

public static void main(String[] args) {
    JFrame frame = new JFrame("lab2");
    frame.add(new Main());
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(maxWidth, maxHeight);
    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;
}

public void actionPerformed(ActionEvent e) {
    angle -= 0.01;
    tx = (sx - (tx / 2) + Math.cos(rotAlpha) * 40);
    ty = (sy - (ty / 2) + Math.sin(rotAlpha) * 40);
    rotAlpha -= 0.033;
    repaint();
}
}

```

## Результат



*Рис. 1-2. Результати роботи програми*