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
Симонюк Володимир Павлович

Зарахована
“____” “_____” 20___ р.
Шкурат Оксаною Сергіївною

варіант №18

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 (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:

18	5, 10	JOIN_ROUND

Типи анімації:

5. Обертання навколо центру малюнка за годинниковою стрілкою
10. Масштабування

Лістинг коду програми

Main.java

```
public class Main extends JPanel implements ActionListener {
    Timer timer;
    private static int maxWidth;
    private static int maxHeight;

    private static int vw = 800;
    private static int vh = 800;

    private double rotationAngle = 0;
    private double scale = 1;
    private double delta = 0.01;
    private double movingAngle = 0;

    Color windowBg = new Color(85,107,47);
    Color clockOutline = new Color(255,215,0);
    Color clockArrows = new Color(0,0,0);
    Color clockBg = new Color(245,245,245);

    int cx = 0;
    int cy = 0;
```

```
    int rOuter = 150;
    int rInner = 140;
    int rHourMark = 15;
    int rPath = 200;

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

        Bootstrap(g2d);
        DrawBorder(g2d);

        g2d.translate(maxWidth / 2, maxHeight / 2);
        g2d.scale(scale, scale);

        DrawClocksBase(g2d);
        DrawHourMarks(g2d);

        g2d.rotate(rotationAngle * 0.1, cx, cy);
        DrawHourArrow(g2d);

        g2d.rotate(rotationAngle, cx, cy);
        DrawMinuteArrow(g2d);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Lab2");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(vw, vh);
        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 Main() {
        timer = new Timer(20, this);
        timer.start();
    }

    public void actionPerformed(ActionEvent e) {
        if (scale < 0.7) {
            delta = -delta;
        } else if (scale > 1) {
            delta = -delta;
        }
        cx = (int)(rPath * Math.cos(movingAngle));
        cy = (int)(rPath * Math.sin(movingAngle));
        rotationAngle += 0.1;
        movingAngle += 0.1;
        scale += delta;
        repaint();
    }

    public void Bootstrap(Graphics2D g2d) {
        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(this.windowBg);
        g2d.clearRect(0, 0, maxWidth, maxHeight);
    };

    public void DrawBorder(Graphics2D g2d) {
        GradientPaint gp = new GradientPaint(0, 0, Color.MAGENTA, 800, 800,
Color.CYAN);
        g2d.setPaint(gp);
        BasicStroke bs1 = new BasicStroke(20, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_ROUND);
        g2d.setStroke(bs1);
        g2d.drawRect(20, 20, 740, 720);
    };

    public void DrawClocksBase(Graphics2D g2d) {
        drawCircle(g2d, cx, cy, rOuter, clockOutline);
        drawCircle(g2d, cx, cy, rInner, clockBg);
    };

    public void DrawHourMarks(Graphics2D g2d) {
        int ro = rInner - rHourMark - 10;
        drawCircle(g2d, cx, cy - ro, rHourMark, clockOutline);
        drawCircle(g2d, cx + ro, cy, rHourMark, clockOutline);
        drawCircle(g2d, cx, cy + ro, rHourMark, clockOutline);
        drawCircle(g2d, cx - ro, cy, rHourMark, clockOutline);
    };

    public void DrawHourArrow(Graphics2D g2d) {
        double[][] arrow = {
                {cx - 20, cy},
                {cx, cy - 15},
                {cx + rInner - 40, cy},
                {cx, cy - 10},
                {cx - 15, cy},
                {cx, cy + 10},
                {cx + rInner - 50, cy},
                {cx, cy + 15}
        };
        GeneralPath gp = new GeneralPath();
        gp.moveTo(arrow[0][0], arrow[0][1]);
        for (int k = 1; k < arrow.length; k++)
            gp.lineTo(arrow[k][0], arrow[k][1]);
        gp.closePath();

        g2d.setColor(clockArrows);
        g2d.fill(gp);
    };

    public void DrawMinuteArrow(Graphics2D g2d) {
        double[][] arrow = {
                {cx, cy + 10},
                {cx + 10, cy},
                {cx, cy - rInner + 10},
                {cx - 10, cy}
        };
        GeneralPath gp = new GeneralPath();
```

```
        gp.moveTo(arrow[0][0], arrow[0][1]);
        for (int k = 1; k < arrow.length; k++)
            gp.lineTo(arrow[k][0], arrow[k][1]);
        gp.closePath();

        g2d.setColor(clockArrows);
        g2d.fill(gp);
    };

    public void drawCircle(Graphics2D g2d, int cx, int cy, int radius, Color
color) {
        g2d.setColor(color);
        g2d.fillOval(cx - radius, cy - radius, radius * 2, radius * 2);
    }
}
```

Результат

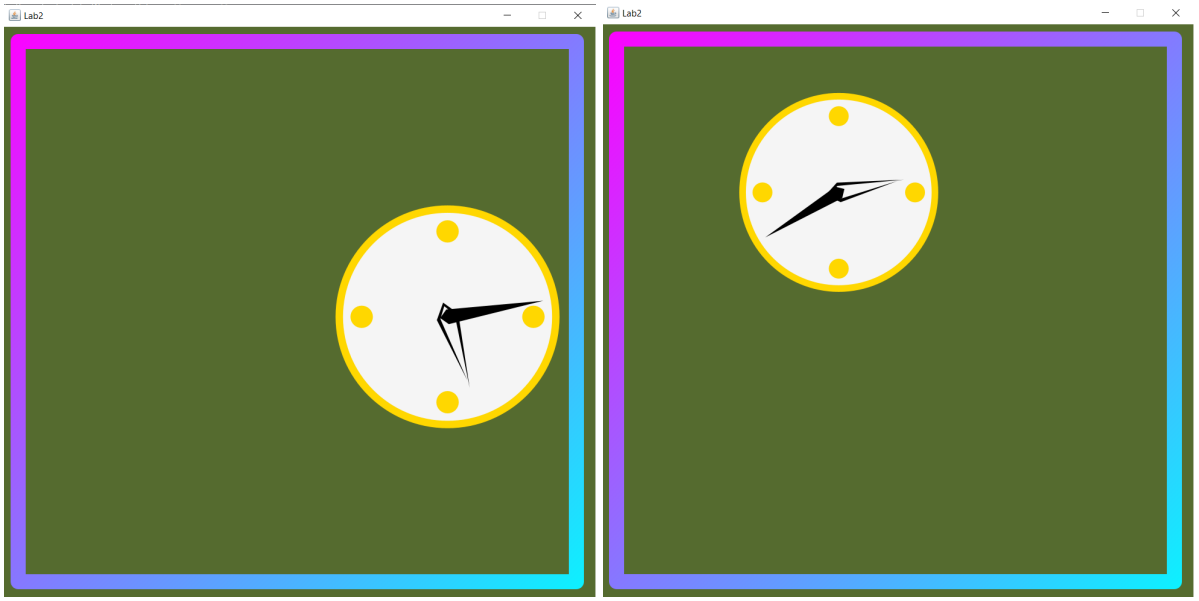

Рис. 1-2. Результати роботи програми