



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

студент III курсу

групи КП-83

Василець Данило Андрійович

*(прізвище, ім'я, по батькові)*

Зарахована

“ \_\_\_\_ ” “ \_\_\_\_ ” 20\_\_ р.

викладачем

Шкурат Оксаною Сергіївною

*(прізвище, ім'я, по батькові)*

варіант № 2

## Варіант завдання

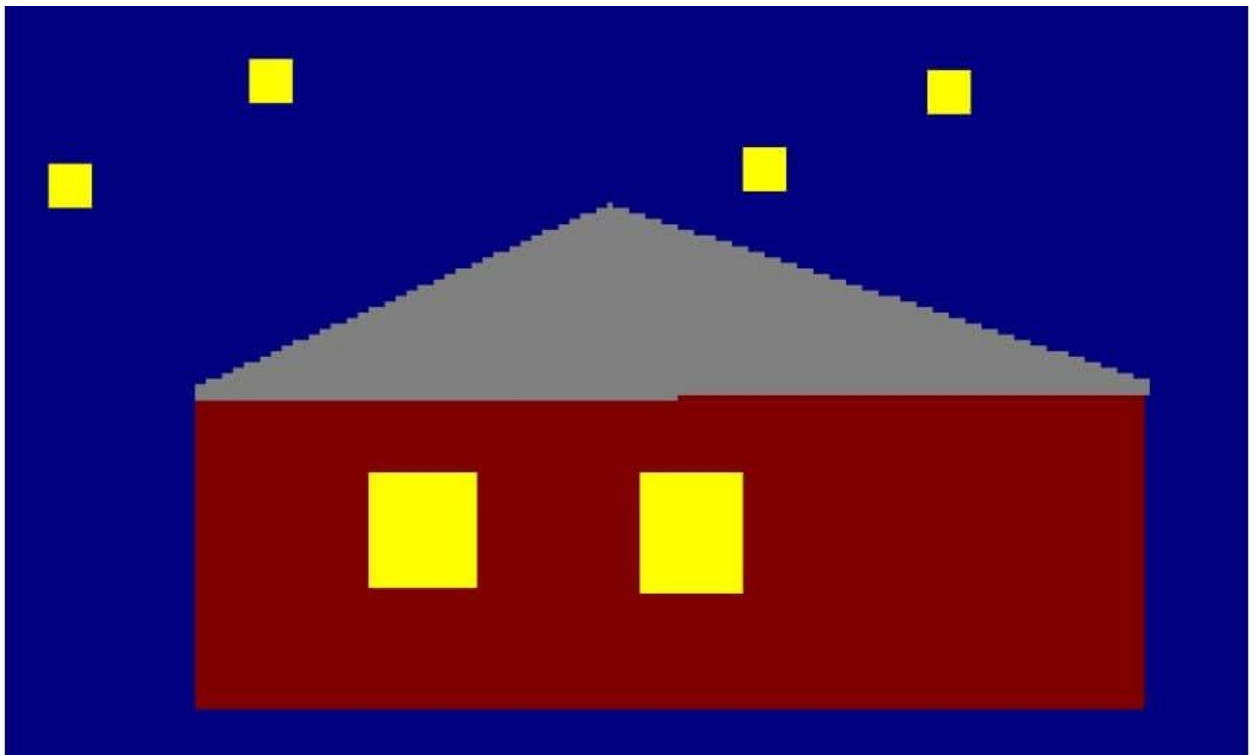
### Завдання:

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 (ламанною).
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:



2	1, 10	JOIN_MITER
---	-------	------------

## Лістинг коду програми

```
public class MyPanel extends JPanel implements ActionListener {
    private static int maxWidth;
    private static int maxHeight;

    Timer timer;

    private double angle = 0;

    private double scale = 0.01;
    private double delta = 0.01;

    private double dx = 1;
    private double tx = 0;
    private double dy = 1;
    private double ty = 6;

    public MyPanel() {
        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(1, 0, 128));
        g2d.clearRect(0, 0, maxWidth, maxHeight);

        /////box
        g2d.setPaint(new Color(128, 0, 0));
        g2d.fillRect(50, 500, 800, maxHeight - 400 - 200);
        /////roof
        g2d.setPaint(new Color(128, 128, 128));
        double[][] points = {
            {50, 500}, {400, 350}, {850, 500}, {50, 500}
        };
        GeneralPath roof = new GeneralPath();
        g2d.translate(0, 0);
        roof.moveTo(points[0][0], points[0][1]);
        for (int k = 1; k < points.length; k++)
            roof.lineTo(points[k][0], points[k][1]);
        roof.closePath();
        g2d.fill(roof);
        /////windows
        GradientPaint gp = new GradientPaint(5, 25,
            Color.MAGENTA, 20, 2, Color.BLUE, true);
        g2d.setPaint(gp);
        g2d.fillRect(150, 550, 100, 100);
        GradientPaint gp2 = new GradientPaint(5, 25,
            Color.WHITE, 20, 2, Color.BLACK, true);
        g2d.setPaint(gp2);
    }
}
```

```

g2d.fillRect(450, 550, 100, 125);
///stars
g2d.setPaint(gp2);
g2d.fillRect(50, 350, 35, 35);
g2d.fillRect(200, 250, 35, 35);
g2d.fillRect(550, 350, 35, 35);
g2d.fillRect(700, 250, 35, 35);

g2d.setPaint(Color.YELLOW);
BasicStroke bs1 = new BasicStroke(16, BasicStroke.CAP_ROUND,
    BasicStroke.JOIN_MITER);
g2d.setStroke(bs1);
g2d.drawRect(maxWidth - 500, 25, 470, 400);

g2d.translate(maxWidth - 500 + 470/2, 25 + 400/2);
double[][] points2 = {
    {0, 17}, {15, 15}, {20, 2}, {25, 15},
    {40, 17}, {30, 25}, {32, 38}, {20, 30},
    {8, 38}, {10, 25}, {0, 17}
};
GeneralPath star = new GeneralPath();
star.moveTo(points2[0][0], points2[0][1]);
for (int k = 1; k < points2.length; k++)
    star.lineTo(points2[k][0], points2[k][1]);
star.closePath();
g2d.scale(scale, scale);
g2d.rotate(angle);
g2d.fill(star);
}

public static void main(String[] args) {
    JFrame frame = new JFrame("Lab2");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1300, 900);
    frame.setLocationRelativeTo(null);
    frame.setResizable(false);
    frame.add(new MyPanel());

    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;
}

@Override
public void actionPerformed(ActionEvent e) {
    if (scale < 0.01) {
        delta = -delta;
    } else if (scale > 1) {
        delta = -delta;
    }
    if (tx < -maxWidth / 3) {
        dx = -dx;
    } else if (tx > maxWidth / 3) {
        dx = -dx;
    }
    if (ty < -maxHeight / 3) {
        dy = -dy;
    } else if (ty > maxHeight / 3) {
        dy = -dy;
    }
    scale += delta;
    angle -= 0.2;
    tx += dx;

```

```
        ty += dy;  
        repaint();  
    }  
}
```

## Результат

