



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 6
по курсу «Языки и методы программирования»
«Программа с графическим пользовательским интерфейсом»
Вариант 3

Студент группы ИУ9-22Б Павлов И. П.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

Приобретение навыков разработки программ с графическим пользовательским интерфейсом на основе библиотеки swing.

2 Условие

В течение лабораторной работы нужно разработать программу, рисующую на экране одно из изображений, перечисленных в таблицах 1 – 7. Программа должна иметь графический пользовательский интерфейс, через который пользователь может задавать параметры изображения. Изображение должно перерисовываться автоматически при изменении любого параметра.

Окружность радиуса r , изображённая с помощью квадратов со стороной a , наклоненных на угол α .

3 Реализация класса CanvasPanel

```
import javax.swing.*;
import java.awt.*;
import java.lang.Math;
import java.util.ArrayList;

public class CanvasPanel extends JPanel {
    private Polygon hexagon;
    private Point center;
    private double angle;
    private int size;
    private ArrayList<Polygon> squares;

    public void setAngle(double angle) {
        this.angle = angle;
        repaint();
    }

    public void setSize(int size) {
        this.size = size;
        repaint();
    }

    public CanvasPanel() {
        this.center = new Point(getWidth() / 2, getHeight() / 2);
        this.angle = 0;
        this.size = 100;
        this.hexagon = createHexagon(center, size);
        this.squares = createSquares(hexagon, size);
    }
}
```

```

@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    center = new Point(getWidth() / 2, getHeight() / 2);
    hexagon = createHexagon(center, size);
    squares = createSquares(hexagon, size);
    g.setColor(Color.BLACK);
    g.drawPolygon(hexagon);
    for (Polygon square : squares) {
        g.drawPolygon(square);
    }
}

private Polygon createHexagon(Point center, int size) {
    Polygon hexagon = new Polygon();
    for (int i = 0; i < 6; i++) {
        double angle_deg = 60 * i + angle;
        double angle_rad = Math.PI / 180 * angle_deg;
        int x = (int) (center.x + size * Math.cos(angle_rad));
        int y = (int) (center.y + size * Math.sin(angle_rad));
        hexagon.addPoint(x, y);
    }
    return hexagon;
}

private ArrayList<Polygon> createSquares(Polygon hexagon, int size) {
    ArrayList<Polygon> squares = new ArrayList<>();
    for (int i = 0; i < 6; i++) {
        Point p1 = new Point(hexagon.xpoints[i], hexagon.ypoints[i]);
        Point p2 = new Point(hexagon.xpoints[(i + 1) % 6], hexagon.ypoints[(i + 1) % 6]);
        Point diff = new Point(p2.x - p1.x, p2.y - p1.y);
        double angle_rad = Math.atan2(diff.y, diff.x);
        double x_offset = size * Math.sin(angle_rad);
        double y_offset = size * Math.cos(angle_rad);
        Point sq_p1 = new Point((int) (p1.x + x_offset), (int) (p1.y - y_offset));
        Point sq_p2 = new Point((int) (p2.x + x_offset), (int) (p2.y - y_offset));
        Point sq_p3 = new Point(p2.x, p2.y);
        Point sq_p4 = new Point(p1.x, p1.y);
        Polygon square = new Polygon();
        square.addPoint(sq_p1.x, sq_p1.y);
        square.addPoint(sq_p2.x, sq_p2.y);
        square.addPoint(sq_p3.x, sq_p3.y);
        square.addPoint(sq_p4.x, sq_p4.y);
        squares.add(square);
    }
    return squares;
}
}

```

4 Реализация класса PictureForm

```

import javax.swing.*.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

```

```

public class PictureForm {
    private JPanel mainPanel;
    private JSlider rectangleSide;
    private JSlider rotationAngle;
    private CanvasPanel canvasPanel;

    public PictureForm () {
        rotationAngle.addChangeListener(new ChangeListener() {
            @Override
            public void stateChanged(ChangeEvent changeEvent) {
                double angle = rotationAngle.getValue();
                canvasPanel.setAngle(angle);
            }
        });

        rectangleSide.addChangeListener(new ChangeListener() {
            @Override
            public void stateChanged(ChangeEvent changeEvent) {
                int side = rectangleSide.getValue();
                canvasPanel.setSize(side);
            }
        });
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Окружность");
        frame.setContentPane(new PictureForm().mainPanel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}

```

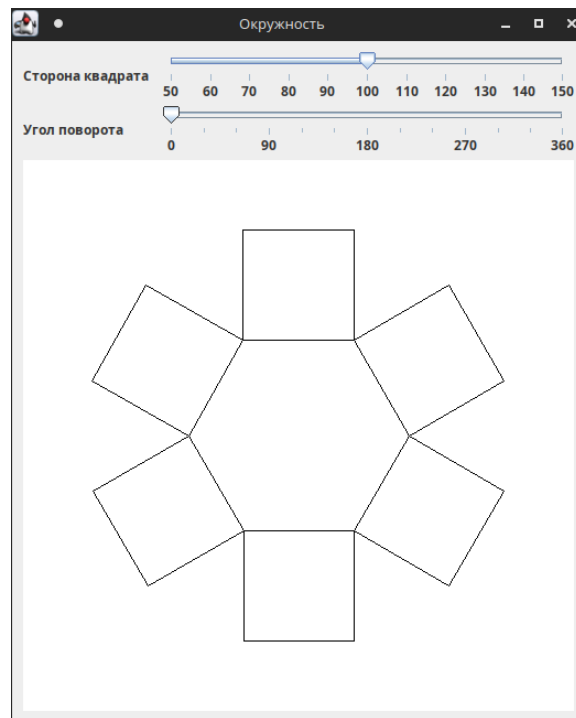


Рис. 1: Пример работы 1

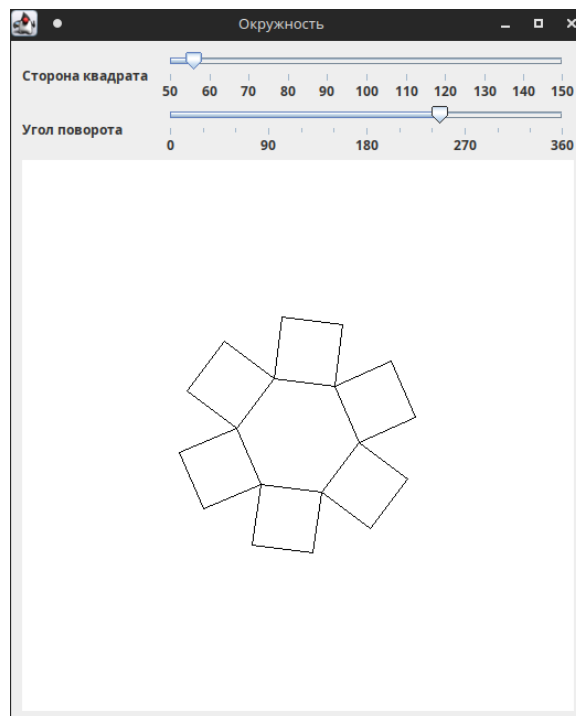


Рис. 2: Пример работы 2

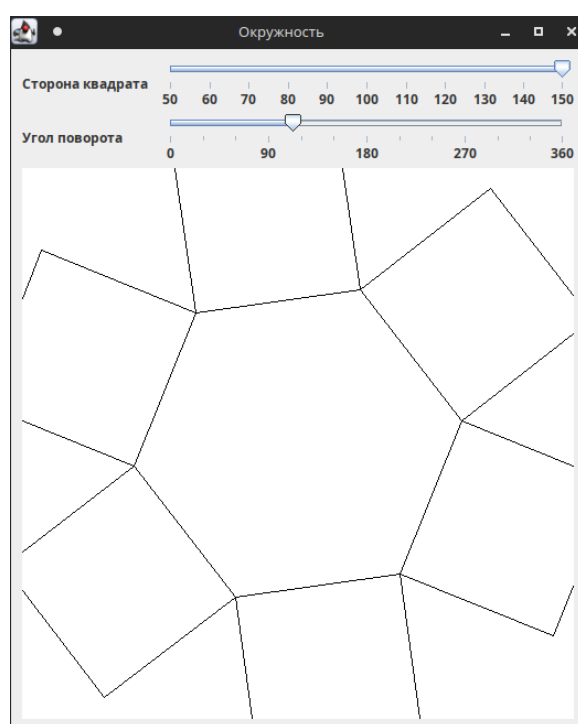


Рис. 3: Пример работы 3