

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №7

Выполнил:
студент 3 курса
группы ПО-9
Мельничук В.М.
Проверил:
Крощенко А.А.

Брест 2024

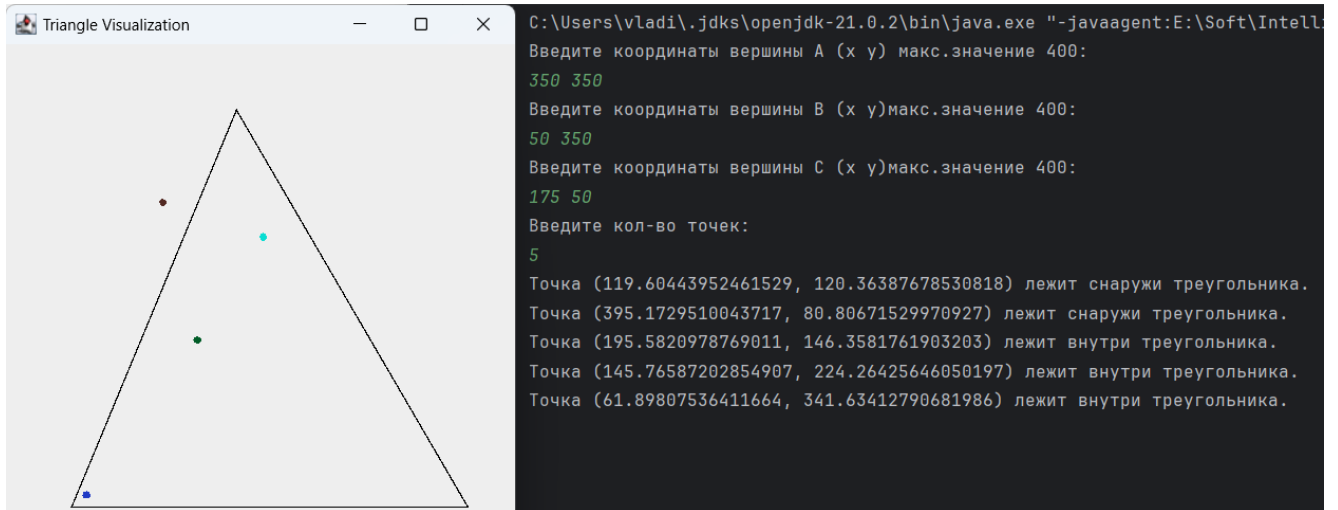
Цель работы: освоить возможности языка программирования Java в построении графических приложений.

Вариант 1

Задание 1

Создать класс Triangle и класс Point. Объявить массив из n объектов класса Point, написать функцию, определяющую, какая из точек лежит внутри, а какая – снаружи треугольника.

Выходные данные:



Код программы:

```
import java.awt.*;
import javax.swing.*;
import java.util.Random;
import java.util.Scanner;

class Point {
    private double x;
    private double y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }
}

class Triangle {
    private Point p1;
    private Point p2;
    private Point p3;

    public Triangle(Point p1, Point p2, Point p3) {
        this.p1 = p1;
        this.p2 = p2;
        this.p3 = p3;
    }

    public boolean isInside(Point point) {
        double area = calculateArea(p1, p2, p3);
        double area1 = calculateArea(point, p2, p3);
        double area2 = calculateArea(p1, point, p3);
        double area3 = calculateArea(p1, p2, point);
        return (area == area1 + area2 + area3);
    }

    private double calculateArea(Point p1, Point p2, Point p3) {
        double x1 = p1.getX(), y1 = p1.getY();
        double x2 = p2.getX(), y2 = p2.getY();
        double x3 = p3.getX(), y3 = p3.getY();
        double s = (x1*(y2+y3) + x2*(y3+y1) + x3*(y1+y2))/2;
        return s;
    }
}
```

```

private double calculateArea(Point p1, Point p2, Point p3) {
    return Math.abs((p1.getX() * (p2.getY() - p3.getY()) +
        p2.getX() * (p3.getY() - p1.getY()) +
        p3.getX() * (p1.getY() - p2.getY())) / 2.0);
}

public void draw(Graphics g) {
    int[] xPoints = {(int) p1.getX(), (int) p2.getX(), (int) p3.getX()};
    int[] yPoints = {(int) p1.getY(), (int) p2.getY(), (int) p3.getY()};
    g.drawPolygon(xPoints, yPoints, 3);
}

public void drawPoint(Graphics g, Point point, Color color) {
    g.setColor(color);
    int x = (int) point.getX();
    int y = (int) point.getY();
    g.fillOval(x - 3, y - 3, 6, 6);
}
}

class TrianglePanel extends JPanel {
    private Triangle triangle;
    private Point[] points;
    private Color[] colors;

    public TrianglePanel(Triangle triangle, Point[] points, Color[] colors) {
        this.triangle = triangle;
        this.points = points;
        this.colors = colors;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.BLACK);
        triangle.draw(g);
        for (int i = 0; i < points.length; i++) {
            triangle.drawPoint(g, points[i], colors[i]);
        }
    }
}

public class task1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите координаты вершины А (x y) макс.значение
400:");
        double x1 = scanner.nextDouble();
        double y1 = scanner.nextDouble();
        Point p1 = new Point(x1, y1);

        System.out.println("Введите координаты вершины В (x y) макс.значение
400:");
        double x2 = scanner.nextDouble();
        double y2 = scanner.nextDouble();
        Point p2 = new Point(x2, y2);

        System.out.println("Введите координаты вершины С (x y) макс.значение
400:");
        double x3 = scanner.nextDouble();
        double y3 = scanner.nextDouble();
        Point p3 = new Point(x3, y3);

        Triangle triangle = new Triangle(p1, p2, p3);

        System.out.println("Введите кол-во точек:");
        int n = scanner.nextInt();
        Point[] points = generateRandomPoints(n);
        Color[] colors = generateRandomColors(n);

        for (int i = 0; i < n; i++) {
            if (triangle.isInside(points[i])) {
                System.out.println("Точка (" + points[i].getX() + ", " +
points[i].getY() + ") лежит внутри треугольника.");
            } else {

```

```

        System.out.println("Точка (" + points[i].getX() + ", " +
points[i].getY() + ") лежит снаружи треугольника.");
    }

    JFrame frame = new JFrame("Triangle Visualization");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 400);

    TrianglePanel panel = new TrianglePanel(triangle, points, colors);
    frame.add(panel);

    frame.setVisible(true);
}

private static Point[] generateRandomPoints(int n) {
    Random random = new Random();
    Point[] points = new Point[n];
    for (int i = 0; i < n; i++) {
        double x = random.nextDouble() * 400;
        double y = random.nextDouble() * 400;
        points[i] = new Point(x, y);
    }
    return points;
}

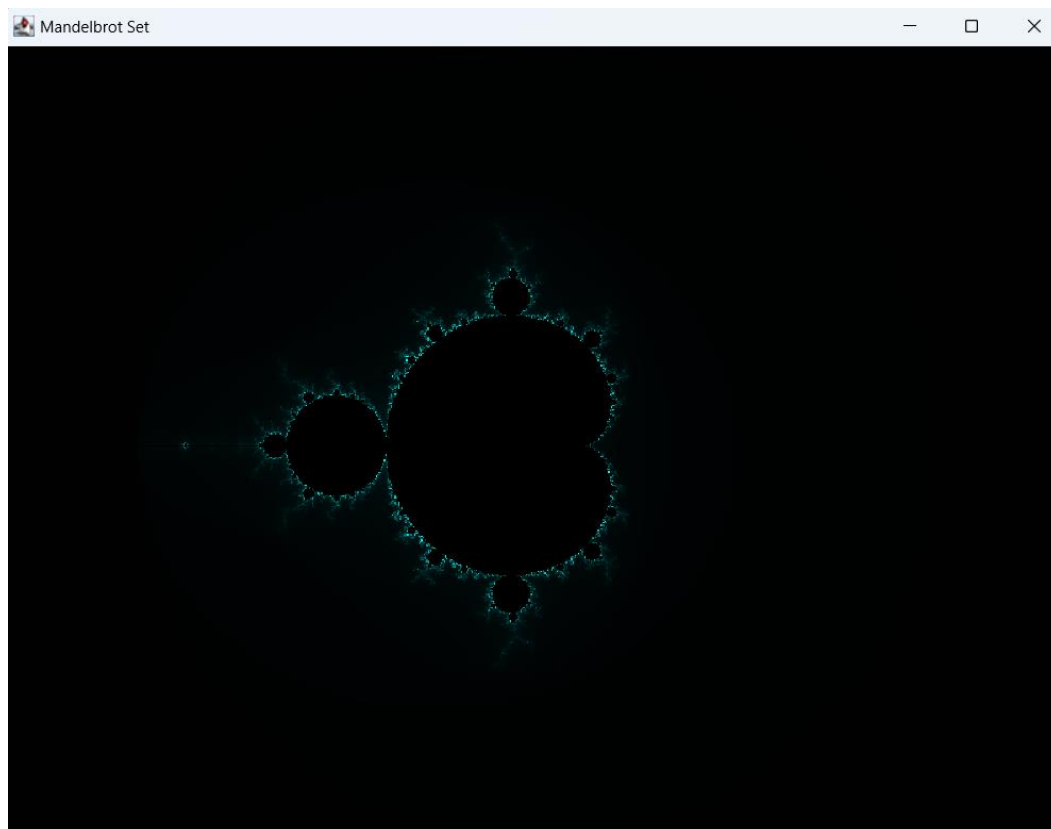
private static Color[] generateRandomColors(int n) {
    Random random = new Random();
    Color[] colors = new Color[n];
    for (int i = 0; i < n; i++) {
        int r = random.nextInt(256);
        int g = random.nextInt(256);
        int b = random.nextInt(256);
        colors[i] = new Color(r, g, b);
    }
    return colors;
}
}

```

Задание 2

Множество Мальдеброта

Выходные данные:



Код программы

```
import javax.swing.*;
import java.awt.*;
import java.awt.image.BufferedImage;

class task2 extends JPanel {
    private static final int MAX_ITER = 1000;
    private static final double ZOOM = 150;
    private static final int WIDTH = 800;
    private static final int HEIGHT = 600;

    public task2() {
        setPreferredSize(new Dimension(WIDTH, HEIGHT));
    }
    private int mandelbrot(double x, double y) {
        double zx = 0;
        double zy = 0;
        double cX = (x - WIDTH / 2) / ZOOM;
        double cY = (y - HEIGHT / 2) / ZOOM;
        int iter = 0;
        while ((zx * zx + zy * zy) < 4 && iter < MAX_ITER) {
            double tmp = zx * zx - zy * zy + cX;
            zy = 2.0 * zx * zy + cY;
            zx = tmp;
            iter++;
        }
        if (iter == MAX_ITER) return 0;
        return iter % 256;
    }

    public void paintComponent(Graphics g) {
        BufferedImage image = new BufferedImage(getWidth(), getHeight(), BufferedImage.TYPE_INT_RGB);
        for (int y = 0; y < HEIGHT; y++) {
            for (int x = 0; x < WIDTH; x++) {
                int color = mandelbrot(x, y);
                image.setRGB(x, y, color | (color << 8));
            }
        }
        g.drawImage(image, 0, 0, this);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Mandelbrot Set");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        task2 mandelbrotSet = new task2();
        frame.add(mandelbrotSet);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```