

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
Кафедра интеллектуальных информационных технологий

## Отчет по лабораторной работе №7

По дисциплине «Современные платформы программирования»  
Специальность ПО-8

Выполнил:

Липовик И.С.

студент группы ПО-8

Проверил:

ст. преп. кафедры ИИТ,

«\_\_»\_\_\_\_\_2024 г.

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

## Вариант 15

### Задание 1. Построение графических примитивов и надписей

#### Требования к выполнению

- Реализовать соответствующие классы, указанные в задании;
- Организовать ввод параметров для создания объектов (можно использовать файлы);
- Осуществить визуализацию графических примитивов, решить поставленную задачу

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.

#### Код программы

##### Task1.java

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

class Point{
    int x;
    int y;
    Point(int x, int y){
        this.x = x;
        this.y = y;
    }
}

class Triangle extends JFrame {
    private int __windowX;
    private int __windowY;
    private Point[] points;
    public Triangle(int windowX, int windowY, Point[] _points) {
        super("circle");
        setSize(windowX, windowY);
        setVisible(true);
        setResizable(false);
        __windowX = windowX;
        __windowY = windowY;
        points=_points;
    }
    public void paint(Graphics g) {
        g.fillRect(0, 0, __windowX, __windowY);
        g.setColor(Color.BLUE);
        g.drawLine(points[0].x, points[0].y, points[1].x, points[1].y);
        g.drawLine(points[1].x, points[1].y, points[2].x, points[2].y);
        g.drawLine(points[2].x, points[2].y, points[0].x, points[0].y);
        g.setColor(Color.RED);
        for(int i=3;i<points.length;i++){
            g.fillRect(points[i].x, points[i].y, 5, 5);
        }
    }
    public void check(Point point){
        float l1=sign(point, points[0], points[1]);
        float l2=sign(point, points[1], points[2]);
        float l3=sign(point, points[2], points[0]);
        boolean hasNeg = (l1 < 0) || (l2 < 0) || (l3 < 0);
    }
}
```

```

        boolean hasPos = (l1 > 0) || (l2 > 0) || (l3 > 0);
        if(! (hasNeg && hasPos)){
            System.out.println("triangle consist point
("+point.x+", "+point.y+")");
        }
        else{
            System.out.println("triangle not consist point
("+point.x+", "+point.y+")");
        }
    }
    public float sign(Point pt, Point p1, Point p2) {
        return (pt.x - p2.x) * (p1.y - p2.y) - (p1.x - p2.x) * (pt.y - p2.y);
    }
    public void run(){
        while(true){
            repaint();
            try{
                Thread.sleep(5);
            }
            catch (InterruptedException e){
            }
            setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter width : ");
        int windowX = sc.nextInt();
        System.out.print("Enter height : ");
        int windowY = sc.nextInt();
        Point[] points = new Point[5];
        for(int i = 0; i < 5; i++){
            System.out.println("Enter point "+(i+1)+" : ");
            System.out.print("Enter x : ");
            int X = sc.nextInt();
            System.out.print("Enter y : ");
            int Y = sc.nextInt();
            points[i] = new Point(X, Y);
        }
        sc.close();
        Triangle app = new Triangle(windowX, windowY, points);
        app.check(points[3]);
        app.check(points[4]);
        app.run();
    }
}

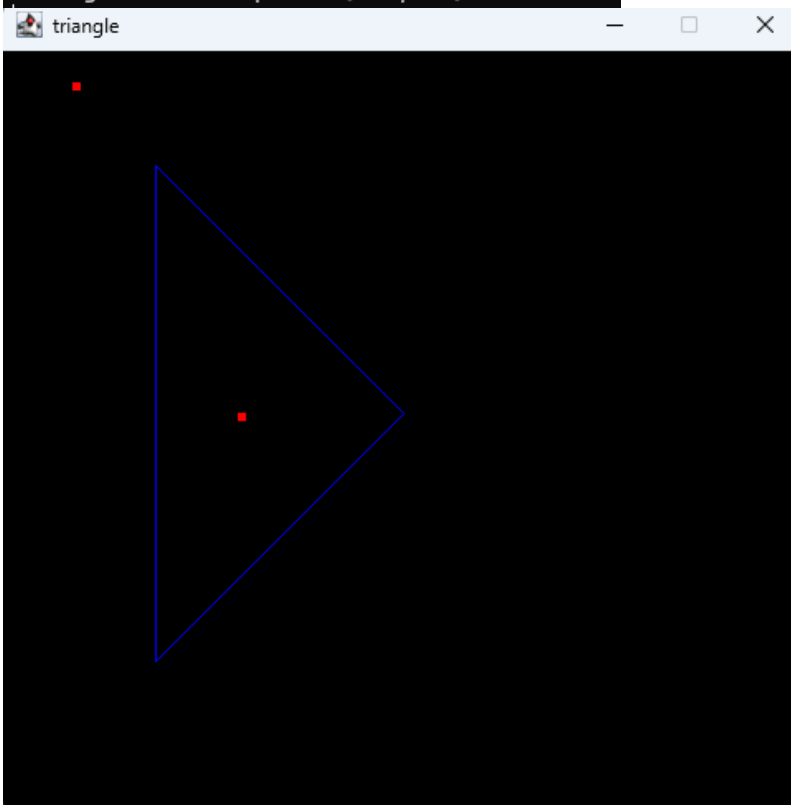
```

**Пример**

```

D:\СНП\lab7\out\production\lab7>java task1
Enter width : 500
Enter height : 500
Enter point 1 :
Enter x : 100
Enter y : 100
Enter point 2 :
Enter x : 100
Enter y : 400
Enter point 3 :
Enter x : 250
Enter y : 250
Enter point 4 :
Enter x : 50
Enter y : 50
Enter point 5 :
Enter x : 150
Enter y : 250
triangle not consist point (50,50)
triangle consist point (150,250)

```



**Задание 2. Реализовать построение заданного типа фрактала по варианту Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала**

**Множество Мальдеброта**

**Task2.java**

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Graphics;
import java.awt.Color;

public class task2 extends JPanel {

    private final int width = 800;
    private final int height = 600;
    private final int maxIterations = 1000;

```

```

private final double zoom = 300;
private final double offsetX = -0.7;
private final double offsetY = 0;

public void paintComponent(Graphics g) {
    super.paintComponent(g);

    for (int x = 0; x < width; x++) {
        for (int y = 0; y < height; y++) {
            double zx = 0;
            double zy = 0;
            double cx = (x - width / 2.0) / zoom + offsetX;
            double cy = (y - height / 2.0) / zoom + offsetY;

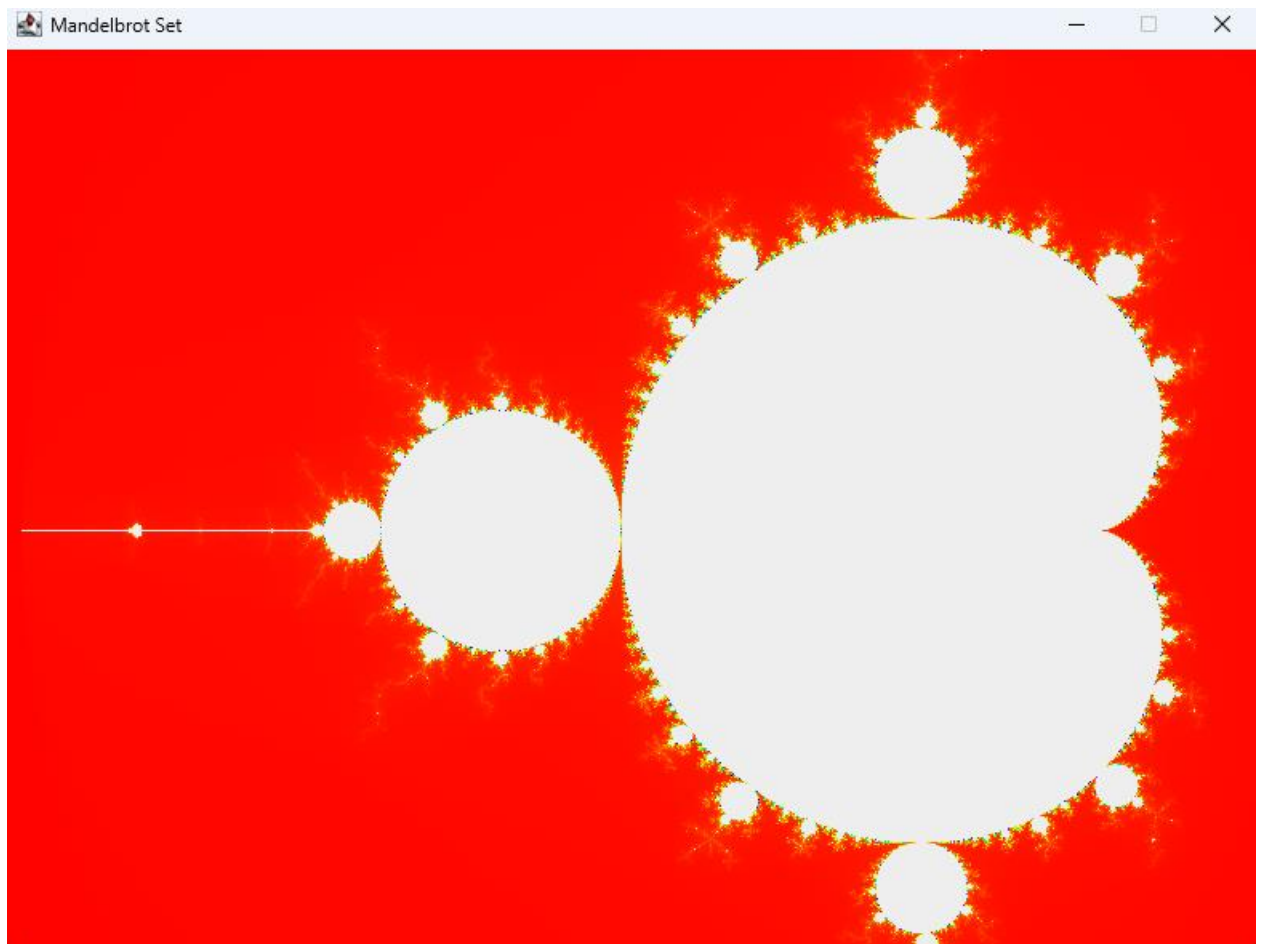
            int iter = maxIterations - 1;
            while (zx * zx + zy * zy < 4 && iter > 0) {
                double temp = zx * zx - zy * zy + cx;
                zy = 2 * zx * zy + cy;
                zx = temp;
                iter--;
            }

            if (iter > 0) {
                g.setColor(Color.getHSBColor((float) ((maxIterations -
iter) / (float) maxIterations), 1, 1));
                g.fillRect(x, y, 1, 1);
            }
        }
    }
}

public static void main(String[] args) {
    JFrame frame = new JFrame("Mandelbrot Set");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(800, 600);
    frame.setResizable(false);
    frame.add(new task2());
    frame.setVisible(true);
}
}

```

**Пример**



**Вывод:** освоили возможности языка программирования Java в построении графических приложений.