

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский Государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №7**  
**по дисциплине “Современные платформы программирования”**

**Выполнил:**

Студент 3 курса

Группы ПО-8

Бондаренко К.А.

**Проверил:**

Крощенко А.А.

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

## Ход работы

### Вариант 1

**Задание 1.** 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:

Класс Точка:

```
package org.example.task1;

public class Point {
    private int X;
    private int Y;

    public Point(int X, int Y) {
        this.X = X;
        this.Y = Y;
    }

    public int getX() { return X; }

    public void setX(int x) { X = x; }

    public int getY() { return Y; }

    public void setY(int y) { Y = y; }

    @java.lang.Override
    public java.lang.String toString() {
        return "Точка{" +
            "X=" + X +
            ", Y=" + Y +
            '}';
    }
}
```

## Класс Треугольник:

```
package org.example.task1;

public class Triangle {
    private final Point[] vertices;

    public Triangle(Point pointA, Point pointB, Point pointC) {
        vertices = new Point[] { pointA, pointB, pointC };
    }

    public boolean isPointInside(Point pointO) {
        Point pointA = vertices[0], pointB = vertices[1], pointC = vertices[2];

        int OAx = pointA.getX() - pointO.getX();
        int OAy = pointA.getY() - pointO.getY();
        int OBx = pointB.getX() - pointO.getX();
        int OBy = pointB.getY() - pointO.getY();
        int OCx = pointC.getX() - pointO.getX();
        int OCy = pointC.getY() - pointO.getY();
        int ABx = pointB.getX() - pointA.getX();
        int ABY = pointB.getY() - pointA.getY();
        int BCx = pointC.getX() - pointB.getX();
        int BCy = pointC.getY() - pointB.getY();
        int CAx = pointA.getX() - pointC.getX();
        int CAy = pointA.getY() - pointC.getY();

        int OAxAB = OAx * ABY - OAy * ABx;
        int OBxBC = OBx * BCy - OBy * BCx;
        int OCxCA = OCx * CAy - OCy * CAx;

        return (OAxAB <= 0 && OBxBC <= 0 && OCxCA <= 0)
            || (OAxAB >= 0 && OBxBC >= 0 && OCxCA >= 0);
    }

    @java.lang.Override
    public java.lang.String toString() {
        return "Треугольник{" +
            "вершины=" + java.util.Arrays.toString(vertices) +
            '}';
    }

    public Point[] getVertices() {
        return vertices;
    }
}
```

## Класс контроллера:

```
package org.example.task1;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
```

```

import javafx.scene.control.TextField;
import javafx.scene.control.ComboBox;
import javafx.scene.layout.Pane;

import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Polygon;

import java.util.List;
import java.util.ArrayList;

public class HelloController {

    @FXML
    private Button add_point_button;
    @FXML
    private Button add_traingle_button;
    @FXML
    private Button remove_point_button;
    @FXML
    private Button remove_triangle_button;
    @FXML
    private Button checkPointsPosition_button;
    @FXML
    private Pane canvas;
    @FXML
    private TextField pointAX_text;
    @FXML
    private TextField pointAY_text;
    @FXML
    private TextField pointBX_text;
    @FXML
    private TextField pointBY_text;
    @FXML
    private TextField pointCX_text;
    @FXML
    private TextField pointCY_text;
    @FXML
    private TextField pointOX_text;
    @FXML
    private TextField pointOY_text;

    List<Point> points = new ArrayList<>();
    List<Triangle> triangles = new ArrayList<>();

    @FXML
    private ComboBox<String> points_comboBox;
    @FXML
    private ComboBox<String> triangles_comboBox;

    @FXML
    void addPoint(ActionEvent event) {
        Point pointO = new Point(Integer.parseInt(pointOX_text.getText()),
Integer.parseInt(pointOY_text.getText()));

```

```

points.add(pointO);
points_comboBox.getItems().add(pointO.toString());

redrawCanvas();
}

```

```

@FXML
void addTriangle(ActionEvent event) {
    Point pointA = new Point(Integer.parseInt(pointAX_text.getText()),
Integer.parseInt(pointAY_text.getText()));
    Point pointB = new Point(Integer.parseInt(pointBX_text.getText()),
Integer.parseInt(pointBY_text.getText()));
    Point pointC = new Point(Integer.parseInt(pointCX_text.getText()),
Integer.parseInt(pointCY_text.getText()));

```

```

    Triangle triangle = new Triangle(pointA, pointB, pointC);

```

```

    triangles.add(triangle);
    triangles_comboBox.getItems().add(triangle.toString());

```

```

    redrawCanvas();
}

```

```

@FXML
void removePoint(ActionEvent event) {
    int selectedIndex = points_comboBox.getSelectionModel().getSelectedIndex();
    if (selectedIndex != -1) {
        Point selectedPoint = points.get(selectedIndex);
        points_comboBox.getItems().remove(selectedIndex);
        points.remove(selectedPoint);
    }
    redrawCanvas();
}

```

```

@FXML
void removeTriangle(ActionEvent event) {
    int selectedIndex = triangles_comboBox.getSelectionModel().getSelectedIndex();
    if (selectedIndex != -1) {
        Triangle selectedTriangle = triangles.get(selectedIndex);
        triangles_comboBox.getItems().remove(selectedIndex);
        triangles.remove(selectedTriangle);
    }
    redrawCanvas();
}

```

```

private Polygon createTrianglePolygon(Triangle triangle) {
    Polygon trianglePolygon = new Polygon(
        triangle.getVertices()[0].getX(), triangle.getVertices()[0].getY(),
        triangle.getVertices()[1].getX(), triangle.getVertices()[1].getY(),
        triangle.getVertices()[2].getX(), triangle.getVertices()[2].getY()
    );
    trianglePolygon.setFill(Color.TRANSPARENT);
    trianglePolygon.setStroke(Color.BLACK);
}

```

```

        return trianglePolygon;
    }

    private void redrawCanvas() {
        canvas.getChildren().clear();
        for (Triangle triangle : triangles) {
            canvas.getChildren().add(createTrianglePolygon(triangle));
            Circle circlePointA = new Circle(triangle.getVertices()[0].getX(), triangle.getVertices()[0].getY(), 3);
            circlePointA.setFill(Color.ORANGE);
            Circle circlePointB = new Circle(triangle.getVertices()[1].getX(), triangle.getVertices()[1].getY(), 3);
            circlePointB.setFill(Color.ORANGE);
            Circle circlePointC = new Circle(triangle.getVertices()[2].getX(), triangle.getVertices()[2].getY(), 3);
            circlePointC.setFill(Color.ORANGE);

            canvas.getChildren().addAll(circlePointA, circlePointB, circlePointC);
        }

        checkPointsPosition();
    }

    @FXML
    void onClickCheckPositions(ActionEvent event) {
        checkPointsPosition();
    }

    void checkPointsPosition() {
        for (Point point : points) {
            Circle circlePoint = new Circle(point.getX(), point.getY(), 3);
            int selectedIndex = triangles_comboBox.getSelectionModel().getSelectedIndex();
            if (selectedIndex != -1) {
                Triangle selectedTriangle = triangles.get(selectedIndex);
                if (selectedTriangle.isPointInside(point)) {
                    circlePoint.setFill(Color.GREEN);
                }
                else {
                    circlePoint.setFill(Color.RED);
                }
            }
            else {
                circlePoint.setFill(Color.BLUE);
            }
            canvas.getChildren().add(circlePoint);
        }
    }
}

```

Класс с методом main:

```

package org.example.task1;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

```

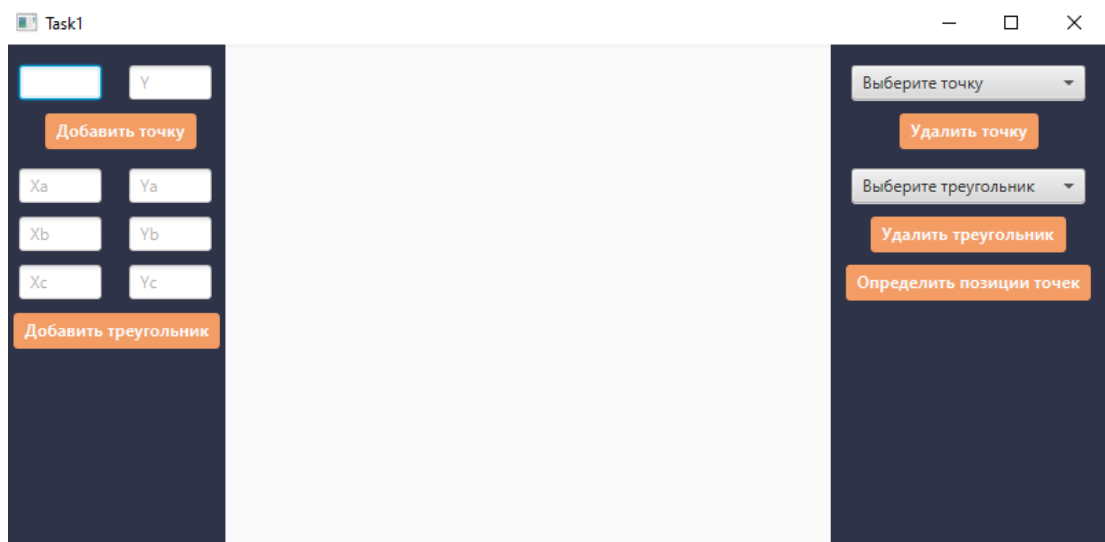
```
import java.io.IOException;

public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), 800, 400);
        stage.setTitle("Task1");
        stage.setScene(scene);
        stage.show();
    }

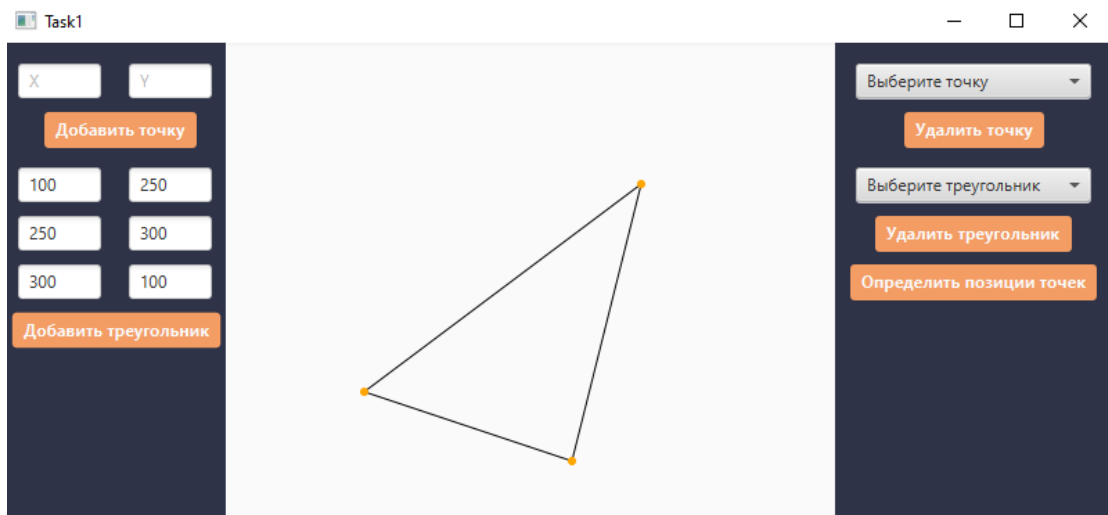
    public static void main(String[] args) {
        launch();
    }
}
```

## Результаты работы программы:

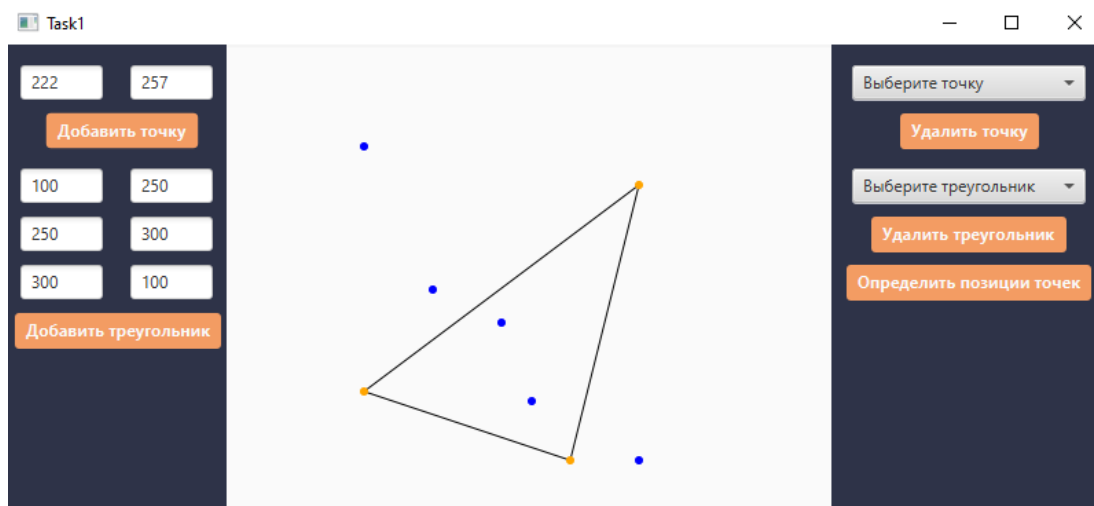
Начальное окно формы программы:



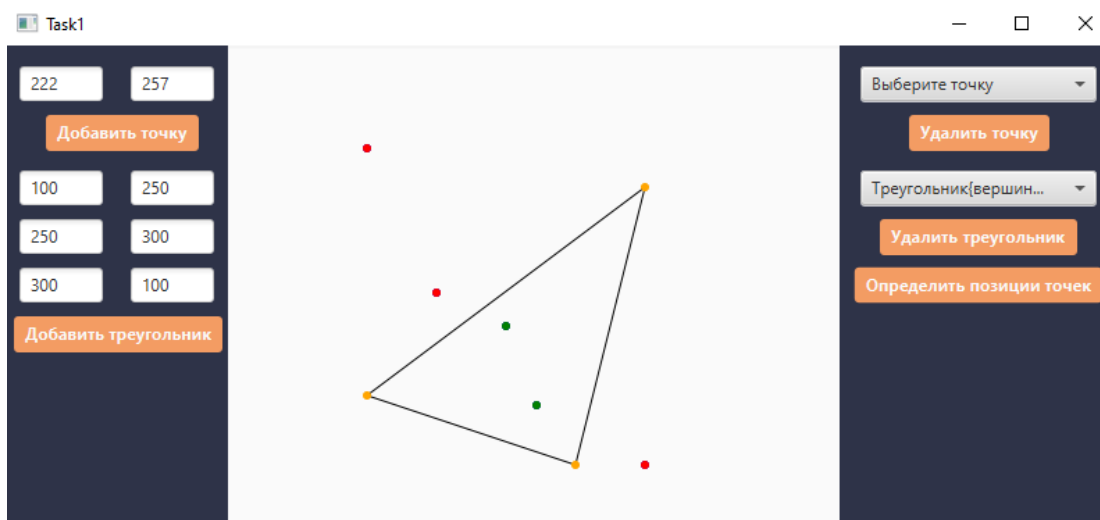
Добавили треугольник:



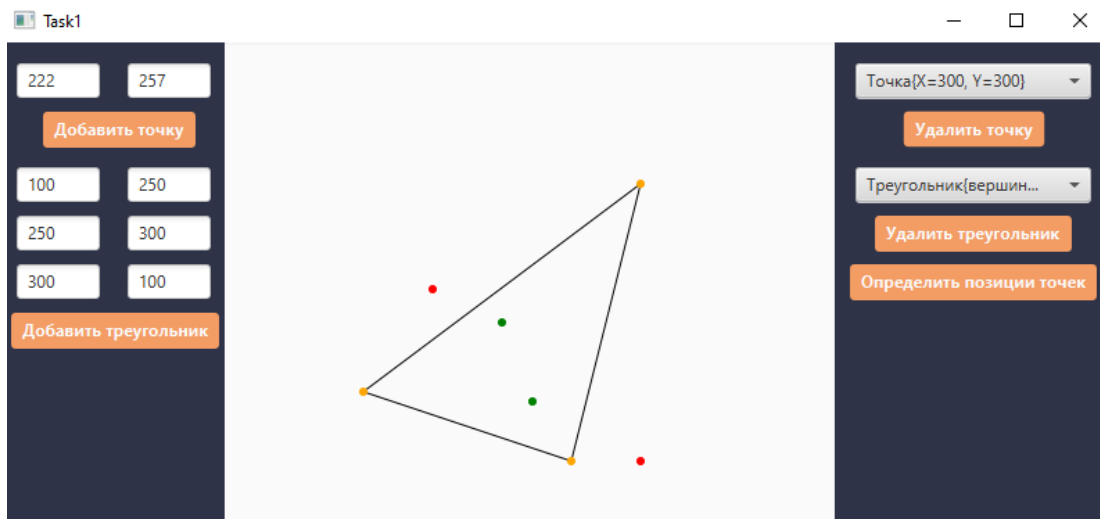
Добавили несколько точек:



Выбрали в comboBox треугольник, нажали на кнопку определения позиций точек:



Выбрали в comboBox точку, нажали на кнопку удаления точки:





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

Множество Мальдеброта

Код программы:

```
package org.example.task2;

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.stage.Stage;

import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;

import javafx.geometry.Insets;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.scene.layout.HBox;

import java.io.IOException;

public class HelloApplication extends Application {
    private Canvas canvas;
    private GraphicsContext gc;
    private TextField realStartField;
    private TextField imaginaryStartField;
    private TextField realWidthField;
    private TextField imaginaryHeightField;
    private Button drawButton;

    @Override
    public void start(Stage primaryStage) {
        canvas = new Canvas(800, 600);
        gc = canvas.getGraphicsContext2D();
        drawButton = new Button("Нарисовать множество Мальдеброта");
        drawButton.setStyle("-fx-background-color: #F39C63; -fx-text-fill: white; -fx-font-weight: bold;");
        drawButton.setAlignment(Pos.CENTER);
        drawButton.setOnAction(e -> drawMandelbrotSet());

        Label realStartLabel = new Label("Начальное значение действительной части:");
        realStartLabel.setStyle("-fx-text-fill: white; -fx-font-weight: bold;");
        realStartField = new TextField("-2.0");
        realStartField.setPrefWidth(60);

        Label imaginaryStartLabel = new Label("Начальное значение мнимой части:");
        imaginaryStartLabel.setStyle("-fx-text-fill: white; -fx-font-weight: bold;");
```

```

imaginaryStartField = new TextField("-1.25");
imaginaryStartField.setPrefWidth(60);

Label realWidthLabel = new Label("Действительная ширина:");
realWidthLabel.setStyle("-fx-text-fill: white; -fx-font-weight: bold;");
realWidthField = new TextField("3.0");
realWidthField.setPrefWidth(60);

Label imaginaryHeightLabel = new Label("Мнимая высота:");
imaginaryHeightLabel.setStyle("-fx-text-fill: white; -fx-font-weight: bold;");
imaginaryHeightField = new TextField("2.5");
imaginaryHeightField.setPrefWidth(60);

VBox root = new VBox();
root.setSpacing(10);
root.setPadding(new Insets(10));
root.setStyle("-fx-background-color: #2E3348;");

HBox buttonContainer = new HBox();
buttonContainer.setAlignment(Pos.CENTER);
buttonContainer.getChildren().add(drawButton);

root.getChildren().addAll(canvas, realStartLabel, realStartField, imaginaryStartLabel,
    imaginaryStartField,
    realWidthLabel, realWidthField, imaginaryHeightLabel, imaginaryHeightField,
    buttonContainer);

Scene scene = new Scene(root);
primaryStage.setScene(scene);
primaryStage.setTitle("Задание 2");
primaryStage.show();
}

private void drawMandelbrotSet() {
    double realStart = Double.parseDouble(realStartField.getText());
    double imaginaryStart = Double.parseDouble(imaginaryStartField.getText());
    double realWidth = Double.parseDouble(realWidthField.getText());
    double imaginaryHeight = Double.parseDouble(imaginaryHeightField.getText());

    double realEnd = realStart + realWidth;
    double imaginaryEnd = imaginaryStart + imaginaryHeight;

    double dx = (realEnd - realStart) / canvas.getWidth();
    double dy = (imaginaryEnd - imaginaryStart) / canvas.getHeight();

    for (int x = 0; x < canvas.getWidth(); x++) {
        for (int y = 0; y < canvas.getHeight(); y++) {
            double cReal = realStart + x * dx;
            double cImaginary = imaginaryStart + y * dy;
            double zReal = 0;
            double zImaginary = 0;
            int iterations = 0;

            while (zReal * zReal + zImaginary * zImaginary < 4 && iterations < 100) {

```

```

        double zRealTemp = zReal * zReal - zImaginary * zImaginary + cReal;
        zImaginary = 2 * zReal * zImaginary + cImaginary;
        zReal = zRealTemp;
        iterations++;
    }

    if (iterations == 100) {
        gc.setFill(Color.BLACK);
    } else {
        gc.setFill(Color.hsb(360 * (iterations / 100.0), 1, 1));
    }
    gc.fillRect(x, y, 1, 1);
}
}
}

public static void main(String[] args) {
    launch();
}
}

```

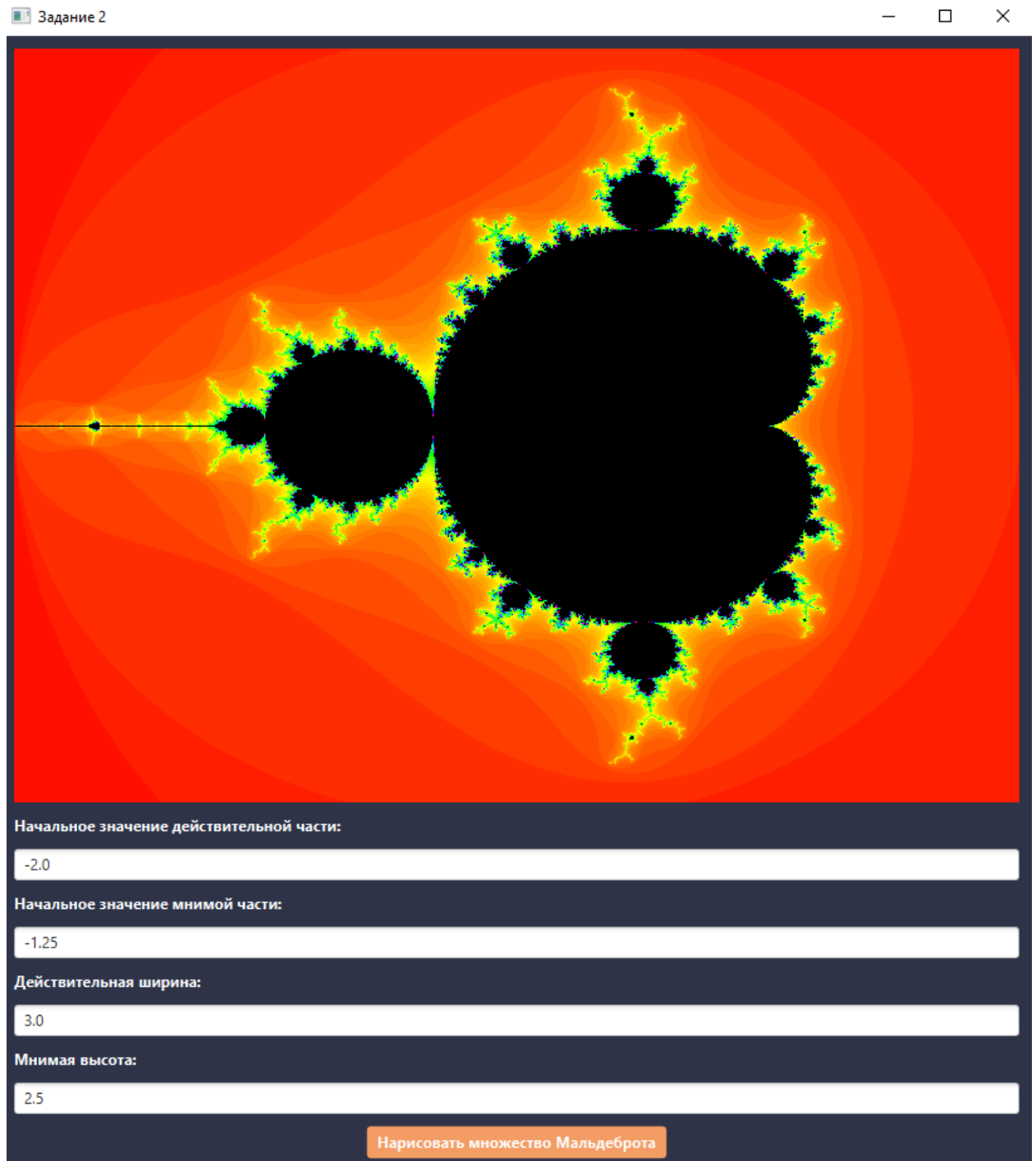
## Результаты работы программы:

Начальное окно ввода параметров:

The screenshot shows a Java Swing window titled "Задание 2" with standard Windows window controls (minimize, maximize, close). The window has a dark blue background. At the bottom, there are four input fields and a button. The labels and values are as follows:

- Начальное значение действительной части:
- Начальное значение мнимой части:
- Действительная ширина:
- Мнимая высота:
- Button: Нарисовать множество Мандельброта

Результат нажатия на кнопку отображения множества Мальдеброта:



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