

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

**Лабораторная работа №7**

По дисциплине: «СПП»

Вариант 11

**Выполнил:**

Студент 3 курса

Группы ПО-9

Лебедович В.А.

**Проверил:**

Крощенко А.А

**Брест 2024**

## Лабораторная работа №7

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

### Задание 1:

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

11) Изобразить прямоугольник, вращающийся в плоскости фрейма вокруг одной из своих вершин.

### Код программы:

#### Main.java

```
package org.example.lab7_1;

import javafx.animation.AnimationTimer;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.paint.Color;
import javafx.scene.transform.Rotate;
import javafx.stage.Stage;
import javafx.scene.shape.Rectangle;

public class Main extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Group root = new Group();
        Scene scene = new Scene(root, 500, 500);

        Rectangle rect = new Rectangle(0, 0, 200, 15);
        rect.setTranslateX(250);
        rect.setTranslateY(250);
        rect.setFill(Color.BLACK);

        root.getChildren().add(rect);

        final TextField widthField = new TextField("200");
        widthField.setTranslateX(20);
        widthField.setTranslateY(20);
        root.getChildren().add(widthField);

        final TextField heightField = new TextField("15");
        heightField.setTranslateX(20);
        heightField.setTranslateY(50);
        root.getChildren().add(heightField);

        Button applyButton = new Button("Apply");
        applyButton.setTranslateX(20);
        applyButton.setTranslateY(80);
        applyButton.setOnAction(event -> {
            try {
```

```

        double rectWidth = Double.parseDouble(widthField.getText());
        double rectHeight = Double.parseDouble(heightField.getText());
        rect.setWidth(rectWidth);
        rect.setHeight(rectHeight);

        scene.getRoot().requestFocus();
    } catch (NumberFormatException e) {
    }
});
root.getChildren().add(applyButton);

final Rotate rotate = new Rotate();
rect.getTransforms().add(rotate);

AnimationTimer timer = new AnimationTimer() {
    @Override
    public void handle(long now) {
        stage.getScene().setOnKeyPressed(e -> {
            rotate.setPivotX(rect.getX());
            rotate.setPivotY(rect.getY());
            if (e.getCode() == KeyCode.LEFT) {
                rotate.setAngle(rotate.getAngle() - 5);
            } else if (e.getCode() == KeyCode.RIGHT) {
                rotate.setAngle(rotate.getAngle() + 5);
            }
        });
    }
};
timer.start();

stage.setScene(scene);
stage.show();
}

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

```

**Результат работы программы:**



**Задание 2:** Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала.

11) Множество Жюлиа

**Код программы:**

**JuliaSet.java**

```
package org.example.lab7 2;  
  
import javafx.application.Application;  
import javafx.scene.Scene;
```

```

import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class JuliaSet extends Application {

    private static final int WIDTH = 800;
    private static final int HEIGHT = 800;

    private double cRe = -0.7;
    private double cIm = 0.27015;
    private int maxIterations = 1000;

    @Override
    public void start(Stage primaryStage) {
        Canvas canvas = new Canvas(WIDTH, HEIGHT);
        GraphicsContext gc = canvas.getGraphicsContext2D();

        redrawJuliaSet(gc);

        TextField cReField = new TextField(Double.toString(cRe));
        TextField cImField = new TextField(Double.toString(cIm));
        TextField maxIterationsField = new TextField(Integer.toString(maxIterations));
        Button drawButton = new Button("Draw");
        drawButton.setOnAction(event -> {
            try {
                cRe = Double.parseDouble(cReField.getText());
                cIm = Double.parseDouble(cImField.getText());
                maxIterations = Integer.parseInt(maxIterationsField.getText());
                redrawJuliaSet(gc);
            } catch (NumberFormatException e) {
                System.out.println("Ошибка: Некорректный ввод");
            }
        });

        VBox vbox = new VBox();
        vbox.getChildren().addAll(
            new HBox(new Label("c Real: "), cReField),
            new HBox(new Label("c Imaginary: "), cImField),
            new HBox(new Label("Max Iterations: "), maxIterationsField),
            drawButton
        );

        StackPane root = new StackPane();
        root.getChildren().addAll(canvas, vbox);

        Scene scene = new Scene(root, WIDTH, HEIGHT);

        primaryStage.setTitle("Julia Set");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private void redrawJuliaSet(GraphicsContext gc) {
        for (int x = 0; x < WIDTH; x++) {
            for (int y = 0; y < HEIGHT; y++) {
                double z_re = -2.0 + (x * (4.0 / WIDTH));
                double z_im = -2.0 + (y * (4.0 / HEIGHT));

                int iterations = 0;
            }
        }
    }
}

```

```

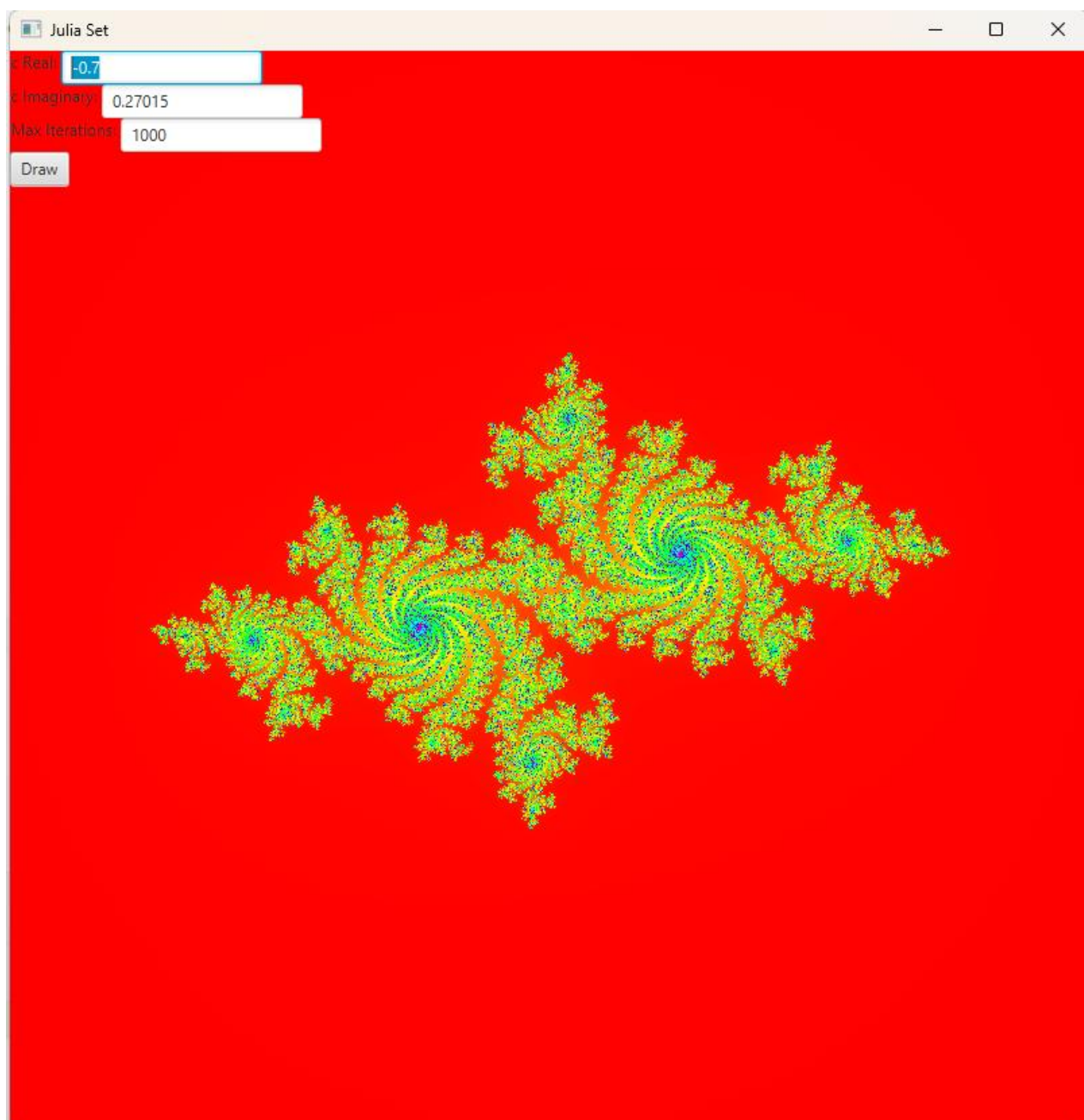
        while (iterations < maxIterations && (z_re * z_re + z_im * z_im) < 4) {
            double new_re = z_re * z_re - z_im * z_im + cRe;
            double new_im = 2 * z_re * z_im + cIm;
            z_re = new_re;
            z_im = new_im;
            iterations++;
        }

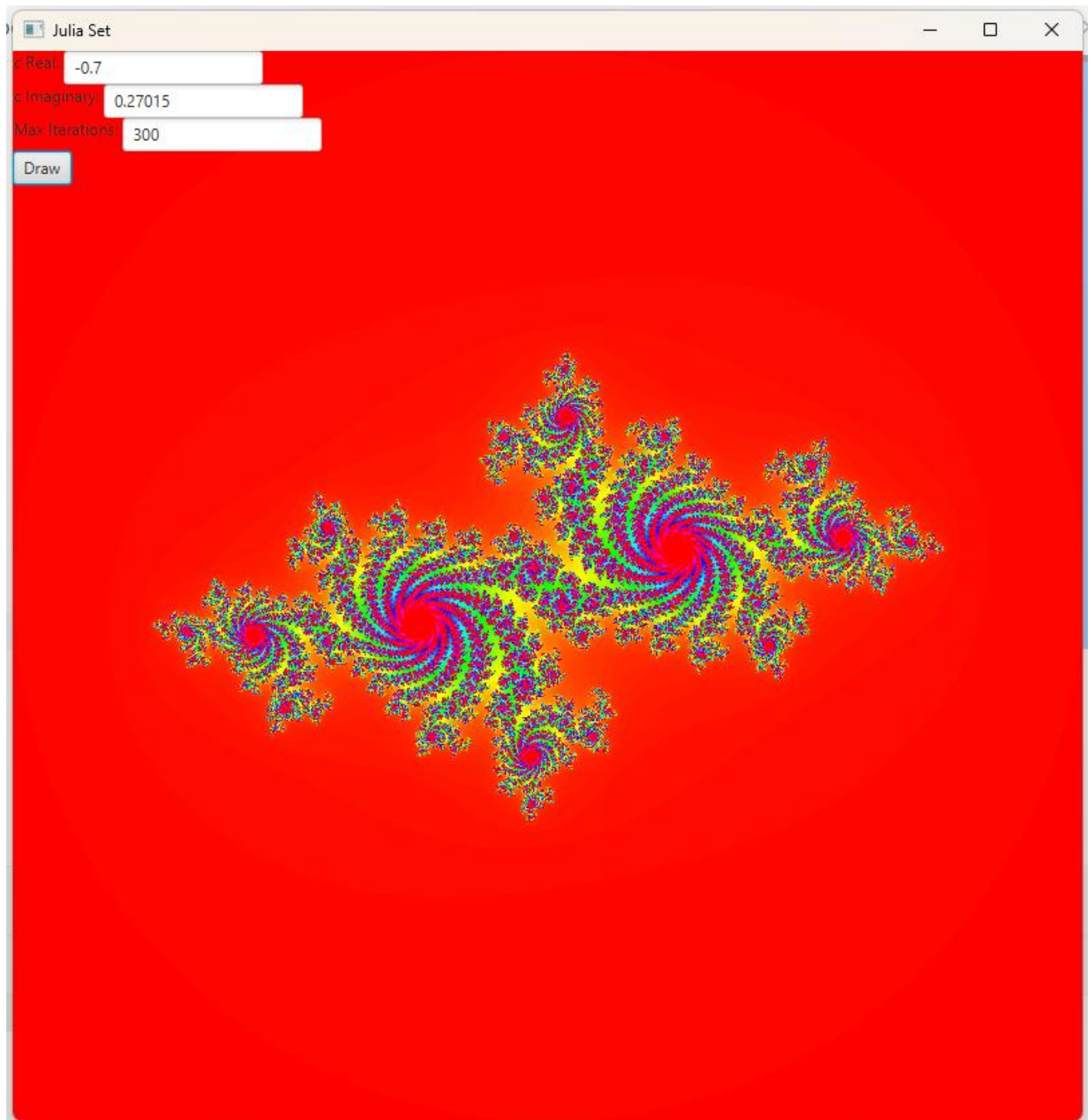
        double t = (double) iterations / maxIterations;
        Color color = Color.hsb(360 * t, 1.0, 1.0);
        gc.setFill(color);
        gc.fillRect(x, y, 1, 1);
    }
}

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

```

## Результаты работы программы:





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