

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

ОТЧЁТ
по лабораторной работе №7

Выполнил:

студент группы ПО-8
Вейгандт И.О.

Проверил:
Крощенко А.А.

Брест 2024

Вариант 5

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

Задание 1:

Изобразить в окне приложения (апплета) отрезок, вращающийся в плоскости экрана вокруг одной из своих концевых точек. Цвет прямой должен изменяться при переходе от одного положения к другому.

```
import javafx.animation.AnimationTimer;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class RotatingLineApplet extends Application {
    private static final int WIDTH = 400;
    private static final int HEIGHT = 400;
    private static final Color START_COLOR = Color.RED;
    private static final Color END_COLOR = Color.BLUE;
    private static final double ROTATION_SPEED = 0.01;

    private double angle = 0.0;
    private Color currentColor = START_COLOR;

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

        Group root = new Group();
```

```

root.getChildren().add(canvas);

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

scene.setFill(Color.WHITE);

primaryStage.setTitle("Rotating Line Applet");
primaryStage.setScene(scene);
primaryStage.show();

    AnimationTimer animationTimer = new
AnimationTimer() {
        @Override
        public void handle(long now) {

            gc.clearRect(0, 0, WIDTH, HEIGHT);

            double x1 = WIDTH / 2;
            double y1 = HEIGHT / 2;
            double x2 = x1 + 100 * Math.cos(angle);
            double y2 = y1 + 100 * Math.sin(angle);

            gc.setStroke(currentColor);
            gc.setLineWidth(2);
            gc.strokeLine(x1, y1, x2, y2);

            angle += ROTATION_SPEED;

            currentColor =
interpolateColor(START_COLOR, END_COLOR, angle % (2 *
Math.PI) / (2 * Math.PI));
        }
    };

```

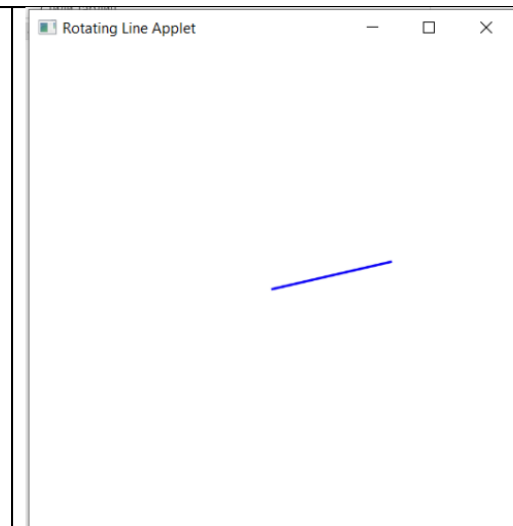
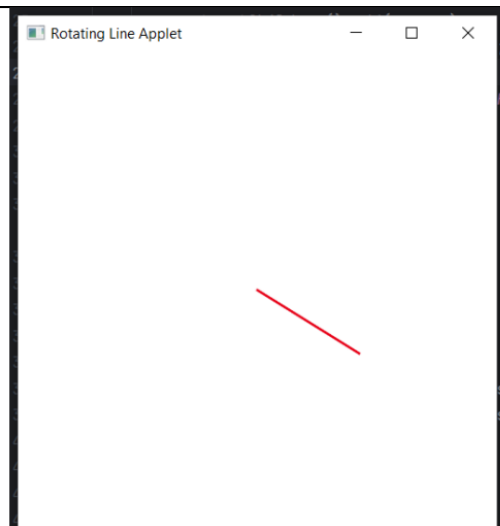
```

        animationTimer.start();
    }

    private Color interpolateColor(Color startColor, Color
endColor, double t) {
        double r = startColor.getRed() + t *
(endColor.getRed() - startColor.getRed());
        double g = startColor.getGreen() + t *
(endColor.getGreen() - startColor.getGreen());
        double b = startColor.getBlue() + t *
(endColor.getBlue() - startColor.getBlue());
        double opacity = startColor.getOpacity() + t *
(endColor.getOpacity() - startColor.getOpacity());
        return new Color(r, g, b, opacity);
    }

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

```



Задание 2:

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

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class PythagorasTree extends Application {

    private static final int WIDTH = 800;
    private static final int HEIGHT = 600;

    private GraphicsContext gc;

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

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Pythagoras Tree");
        Group root = new Group();
        Canvas canvas = new Canvas(WIDTH, HEIGHT);
        root.getChildren().add(canvas);
        gc = canvas.getGraphicsContext2D();
        drawPythagorasTree(400, 600, 200, -Math.PI / 2,
12);
```

```

        Scene scene = new Scene(root, WIDTH, HEIGHT);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private void drawPythagorasTree(double x, double y,
double size, double angle, int depth) {
        if (depth == 0) {
            return;
        }
        double x2 = x + Math.cos(angle) * size;
        double y2 = y + Math.sin(angle) * size;

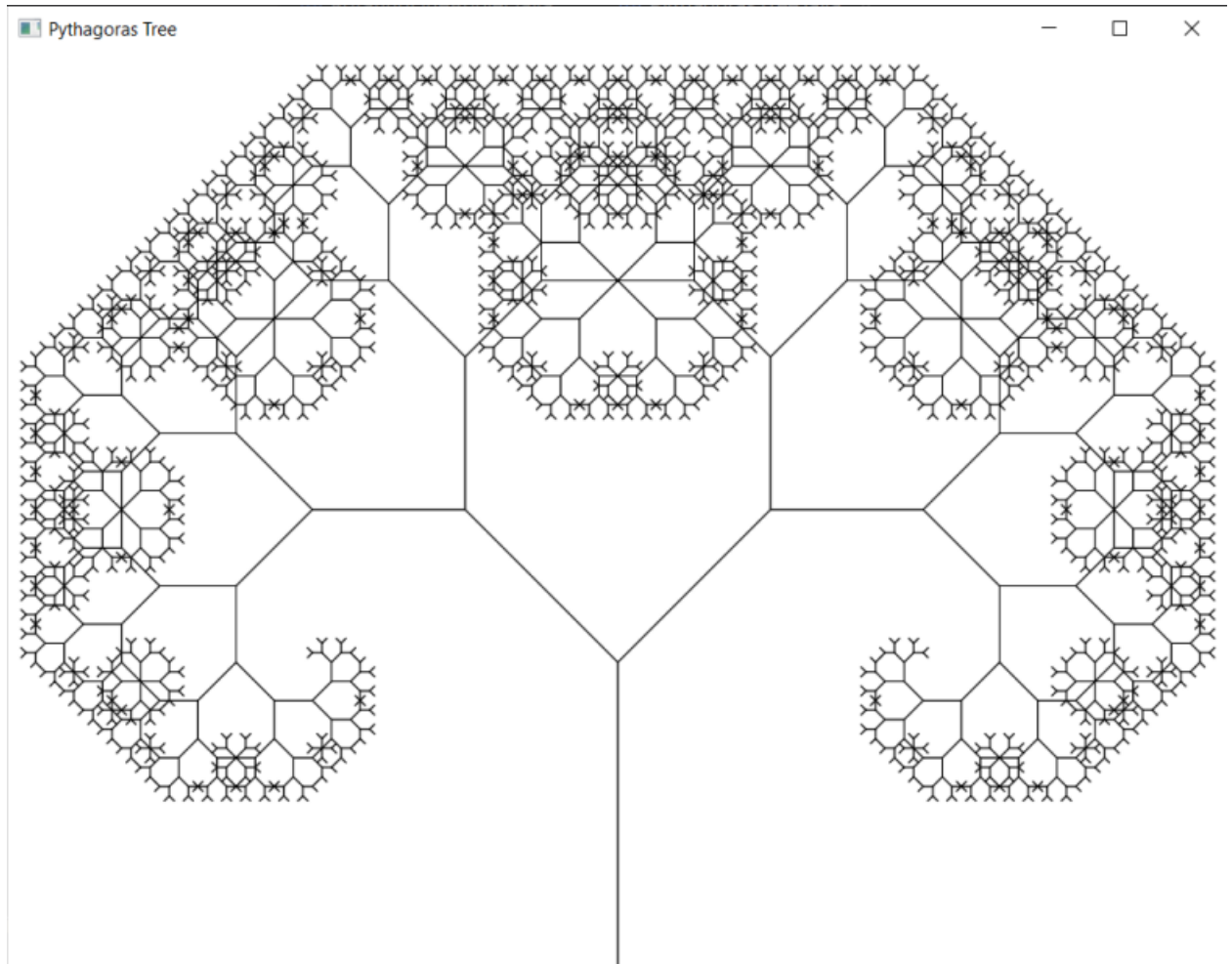
        gc.setStroke(Color.BLACK);
        gc.strokeLine(x, y, x2, y2);
        double angle1 = angle - Math.PI / 4;
        double angle2 = angle + Math.PI / 4;

        double size1 = size / Math.sqrt(2);

        drawPythagorasTree(x2, y2, size1, angle1, depth -
1);
        drawPythagorasTree(x2, y2, size1, angle2, depth -
1);
    }

```

}



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