

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

ОТЧЁТ
по лабораторной работе №7

Выполнил
студент 3 курса
группы ПО-9
Ничингер Кирилл Александрович

Проверил:
Крощенко А. А.

Брест 2024

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

Вариант 4

Задание 1: изобразить разносторонний треугольник, вращающийся в плоскости апплета вокруг своего центра тяжести.

Код программы:

```
public class RotatingTriangle extends Application {
    private ArrayList<Double> triangleSides = new ArrayList<>(List.of(300., 200.,
150.));
    private static List<Double> triangleVertexPositions;
    ObservableList<Double> trianglePoints;
    Polygon triangle;
    private static double windowWidth, windowHeight;
    private static Point rotationCenter = new Point();
    double maxDistToVertex;

    @Override
    public void init() throws Exception {
        //validate triangle sides
        if (triangleSides.size() != 3)
            throw new IllegalArgumentException("Provided not 3 triangle sides");
        for (int i = 0; i < triangleSides.size(); i++){
            if (triangleSides.get(i) >= triangleSides.get((i+1)%3)
                + triangleSides.get((i+2)%3)){
                throw new IllegalArgumentException("Wrong triangle sides");
            }
        }

        //convert triangle sides to vertex positions
        double a = triangleSides.get(0);
        double b = triangleSides.get(1);
        double c = triangleSides.get(2);

        //A coords
        double x1 = 0.0;
        double y1 = 0.0;

        //B coords
        double x2 = c;
        double y2 = 0.0;

        //C coords
        double cosCAB = (b * b + c * c - a * a) / (2 * b * c);
        double x3 = b * cosCAB;
        double y3 = Math.sqrt(b * b - x3 * x3);

        triangleVertexPositions = List.of(x1,y1,x2,y2,x3,y3);

        //Find center of mass coord
        rotationCenter.setX(Math.abs((x1 + x2 + x3) / 3));
        rotationCenter.setY(Math.abs((y1 + y2 + y3) / 3));
    }
}
```

```

        maxDistToVertex = 0;
        for (int i = 0; i < 3; i++) {
            maxDistToVertex = Math.max(maxDistToVertex,
                Math.sqrt(
                    Math.pow(rotationCenter.getX() -
                        triangleVertexPositions.get(i * 2), 2)
                    + Math.pow(rotationCenter.getY() -
                        triangleVertexPositions.get(i * 2 + 1), 2)
                )
            );
        }
        windowHeight = windowWidth = maxDistToVertex * 2;

        super.init();
    }

    @Override
    public void start(Stage primaryStage) {
        triangle = new Polygon();
        triangle.getPoints().addAll(triangleVertexPositions);
        trianglePoints = triangle.getPoints();
        triangle.setFill(Color.RED);

        triangle.setLayoutY(maxDistToVertex - rotationCenter.getY());
        triangle.setLayoutX(maxDistToVertex - rotationCenter.getX());

        Group root = new Group(triangle);

        Scene scene = new Scene(root, windowHeight, windowHeight);

        primaryStage.setScene(scene);
        primaryStage.show();

        Timer timer = new Timer();
        timer.scheduleAtFixedRate(new TimerTask() {
            @Override
            public void run() {
                rotateTriangle();
            }
        }, 0, 10);
    }

    private void rotateTriangle() {
        for (int i = 0; i < trianglePoints.size(); i += 2) {
            Point trianglePoint = new Point(trianglePoints.get(i),
                trianglePoints.get(i+1));

            trianglePoint.rotateAroundPoint(rotationCenter, 0.01);
            trianglePoints.set(i, trianglePoint.getX());
            trianglePoints.set(i+1, trianglePoint.getY());
        }
    }

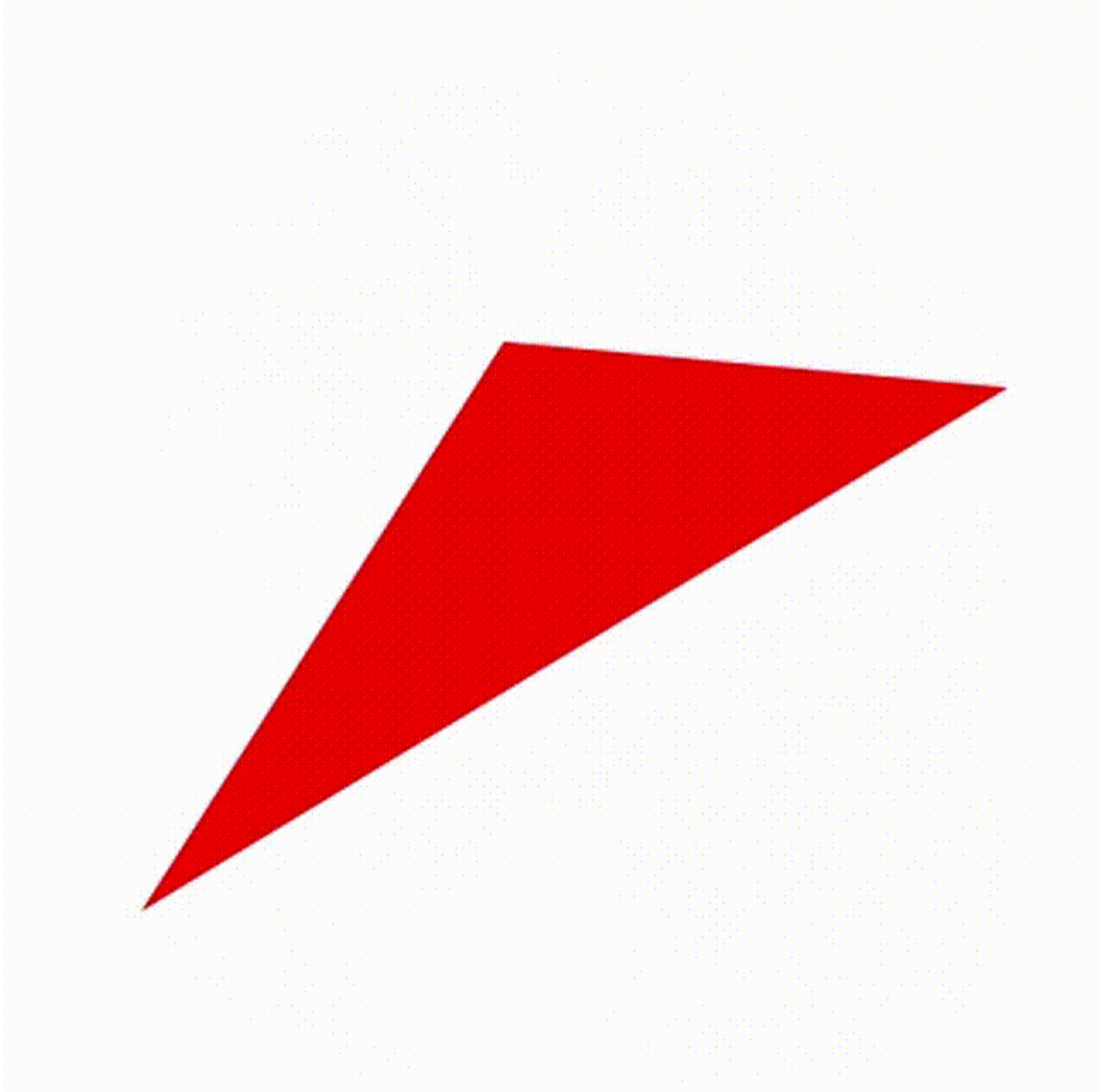
    public static void main(String[] args) {
        launch(args);
    }
}

```

Входные данные:

```
// стороны треугольника  
private ArrayList<Double> triangleSides = new ArrayList<>(List.of(300., 200., 150.));
```

Результат работы программы:



Задание 2: Ковер Серпинского.

Код программы:

```
public class SierpinskiCarpet extends Application {
    private static double windowSize = 729;
    private static int depth = 5;
    private static Group root;
    private static List<Color> colors = new ArrayList<>(
        List.of(
            Color.RED, Color.AQUAMARINE, Color.ALICEBLUE, Color.ANTIQUEWHITE,
            Color.BISQUE, Color.AZURE, Color.WHEAT, Color.CORNFLOWERBLUE,
            Color.TAN, Color.SLATEGREY, Color.TOMATO, Color.SILVER,
            Color.VIOLET, Color.PINK, Color.SALMON, Color.OLIVE
        )
    );

    @Override
    public void start(Stage primaryStage) {
        root = new Group();

        drawRectangles(depth, windowSize / 2, windowSize / 2);

        Scene scene = new Scene(root, windowSize, windowSize);

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

    private void drawRectangles(int curDepth, double x, double y) {
        if (curDepth > 0) {
            int rectSize = (int) (windowSize / Math.pow(3, depth - curDepth + 1));
            Rectangle rect = new Rectangle(x - rectSize / 2, y - rectSize / 2,
                rectSize, rectSize);
            rect.setFill(colors.get((int) (Math.random() * colors.size())));
            root.getChildren().add(rect);
            curDepth--;

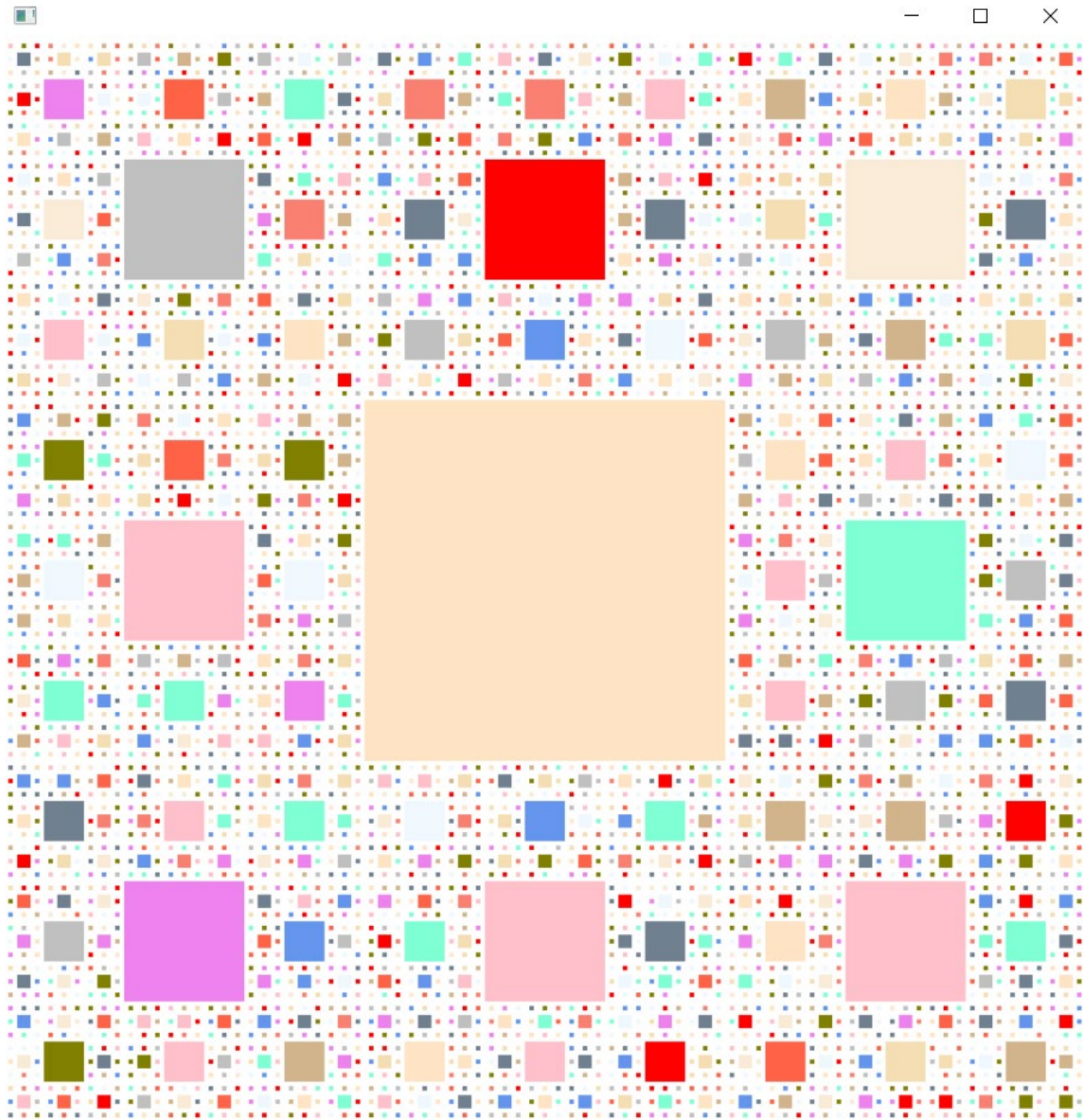
            drawRectangles(curDepth, x + rectSize, y + rectSize);
            drawRectangles(curDepth, x - rectSize, y + rectSize);
            drawRectangles(curDepth, x + rectSize, y - rectSize);
            drawRectangles(curDepth, x - rectSize, y - rectSize);
            drawRectangles(curDepth, x + rectSize, y);
            drawRectangles(curDepth, x - rectSize, y);
            drawRectangles(curDepth, x, y + rectSize);
            drawRectangles(curDepth, x, y - rectSize);
        }
    }

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

Входные данные:

```
//размер окна  
private static double windowSize = 729;  
// глубина построения фрактала  
private static int depth = 5;
```

Результат работы программы:



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