

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

**Лабораторная работа №7
По дисциплине: «ССП»
Вариант - 12**

Выполнил:
Студент 3 курса
Группы ПО-8
Иванюк М.С.
Проверил:
Крощенко А.А

Брест, 2024

Лабораторная работа №7

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

Задание 1: Задать составление строки из символов, появляющихся из разных углов апплета и выстраивающихся друг за другом. Процесс должен циклически повторяться.

Код программы:

```
package org.example.lab7_2;

import javafx.animation.*;
import javafx.application.Application;
import javafx.geometry.Bounds;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.Pane;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.scene.shape.Line;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.util.Duration;

public class Main extends Application {

    private String characters;
    private int currentIndex = 0;

    @Override
    public void start(Stage primaryStage) {
        Pane root = new Pane();
        Scene scene = new Scene(root, 400, 400);

        Text centerText = new Text();
        root.getChildren().add(centerText);

        TextField textField = new TextField();
        textField.setPromptText("Enter characters");
        Button startButton = new Button("Start Animation");

        VBox inputBox = new VBox(10);
        inputBox.getChildren().addAll(textField, startButton);
        inputBox.setLayoutX((scene.getWidth() - 150) / 2);
        inputBox.setLayoutY(20);

        root.getChildren().add(inputBox);

        startButton.setOnAction(event -> {
            if (!textField.getText().isEmpty()) {
                characters = textField.getText().toUpperCase();
            }
            currentIndex = 0;
            centerText.setText("");

            Timeline timeline = new Timeline(new
                KeyFrame(Duration.seconds(1), e -> {
                    if (currentIndex >= characters.length()) {
```

```

        currentIndex = 0;
    }

    Text cornerText = new
Text(Character.toString(characters.charAt(currentIndex++)));
    root.getChildren().add(cornerText);

    double x = Math.random() < 0.5 ? 0 : scene.getWidth();
    double y = Math.random() < 0.5 ? 0 : scene.getHeight();

    cornerText.setX(x);
    cornerText.setY(y);

    PathTransition transition = new
PathTransition(Duration.seconds(1), new Line(x, y, scene.getWidth() / 2,
scene.getHeight() / 2), cornerText);
    transition.setOnFinished(e2 -> {
        root.getChildren().remove(cornerText);
        centerText.setText(centerText.getText() +
cornerText.getText());

        Bounds textBounds = centerText.getBoundsInLocal();
        centerText.setX(scene.getWidth() / 2 -
textBounds.getWidth() / 2);
        centerText.setY(scene.getHeight() / 2 -
textBounds.getHeight() / 2);
    });
    transition.play();
    }));

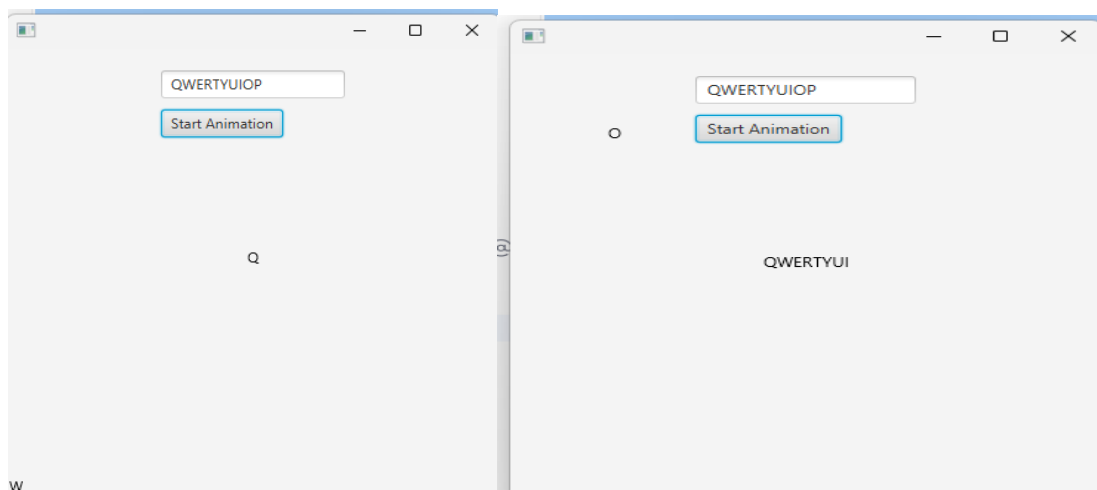
    timeline.setCycleCount(Animation.INDEFINITE);
    timeline.play();
    });

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

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

```

Результат работы программы:



Задание 2: Кривая Гильберта.

Код программы:

```
package org.example.lab7_1;

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class HilbertCurve extends Application {

    private final int lx = 10, ly = 10;
    private int X = 50, Y = 50;
    TextField orderField;
    @Override
    public void start(Stage primaryStage) {
        Canvas canvas = new Canvas(800, 800);
        GraphicsContext gc = canvas.getGraphicsContext2D();

        Button drawButton = new Button("Draw");
        orderField = new TextField("1");
        orderField.setPromptText("Введите порядок кривой Гильберта:");

        drawButton.setOnAction(e -> draw(gc, orderField.getText()));

        GridPane grid = new GridPane();
        grid.setAlignment(Pos.TOP_CENTER);
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(20));

        grid.add(new Label("Введите порядок кривой Гильберта:"), 0, 0);
        grid.add(orderField, 1, 0);
        grid.add(drawButton, 1, 1);

        StackPane root = new StackPane(canvas, grid);
        StackPane.setAlignment(grid, Pos.TOP_CENTER);

        Scene scene = new Scene(root, 800, 800);

        primaryStage.setTitle("Hilbert Curve");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private void draw(GraphicsContext gc, String input) {
        try {
            gc.clearRect(0, 0, 800, 800);
        }
    }
}
```

```

        int order = Integer.parseInt(input);
        //      int initialX = 50;
        //      int initialY = 50;
        X = 50;
        Y = 50;
        a(order, gc);
    } catch (NumberFormatException e) {
        Alert alert = new Alert(Alert.AlertType.WARNING);
        alert.setTitle("Warning");
        alert.setHeaderText("Invalid Input");
        alert.setContentText("Please enter a valid integer value.");
        alert.showAndWait();
        orderField.clear();
    }
}

private void drawPart(GraphicsContext gc, int lx, int ly) {
    gc.setStroke(Color.VIOLET);
    gc.strokeLine(X, Y, X + lx, Y + ly);
    X += lx;
    Y += ly;
}

private void a(int i, GraphicsContext gc) {
    if (i > 0) {
        d(i - 1, gc);
        drawPart(gc, lx, 0);
        a(i - 1, gc);
        drawPart(gc, 0, ly);
        a(i - 1, gc);
        drawPart(gc, -lx, 0);
        c(i - 1, gc);
    }
}

private void b(int i, GraphicsContext gc) {
    if (i > 0) {
        c(i - 1, gc);
        drawPart(gc, -lx, 0);
        b(i - 1, gc);
        drawPart(gc, 0, -ly);
        b(i - 1, gc);
        drawPart(gc, lx, 0);
        d(i - 1, gc);
    }
}

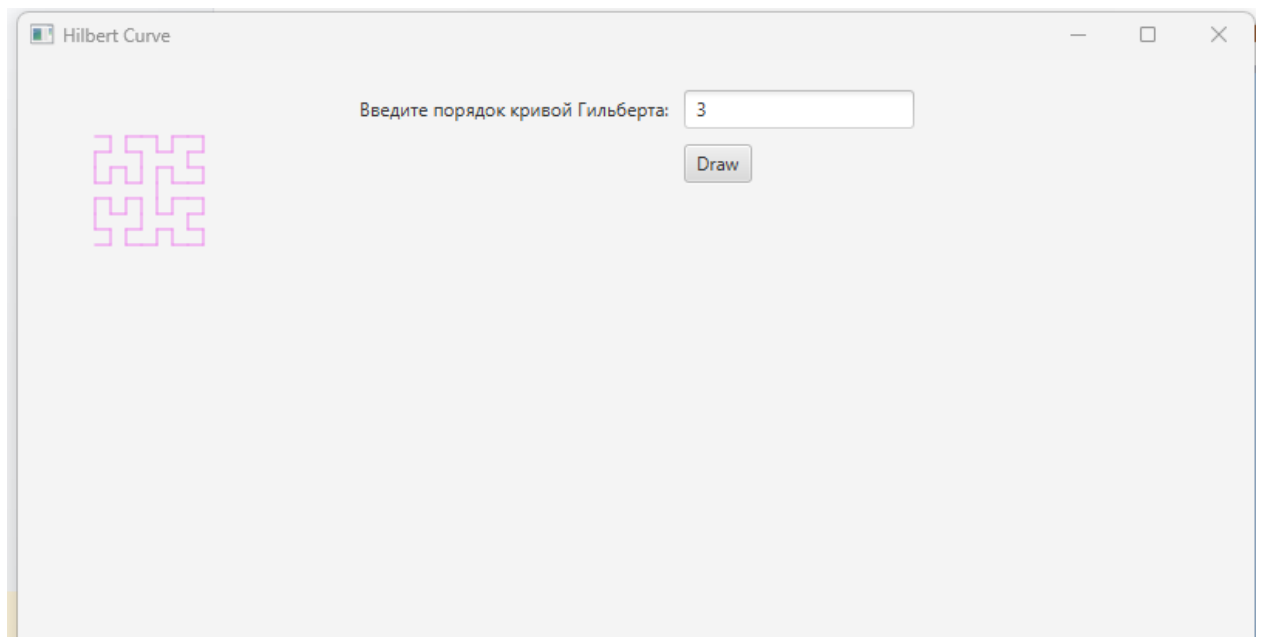
private void c(int i, GraphicsContext gc) {
    if (i > 0) {
        b(i - 1, gc);
        drawPart(gc, 0, -ly);
        c(i - 1, gc);
        drawPart(gc, -lx, 0);
        c(i - 1, gc);
        drawPart(gc, 0, ly);
        a(i - 1, gc);
    }
}

private void d(int i, GraphicsContext gc) {
    if (i > 0) {
        a(i - 1, gc);
        drawPart(gc, 0, ly);
        d(i - 1, gc);
    }
}

```

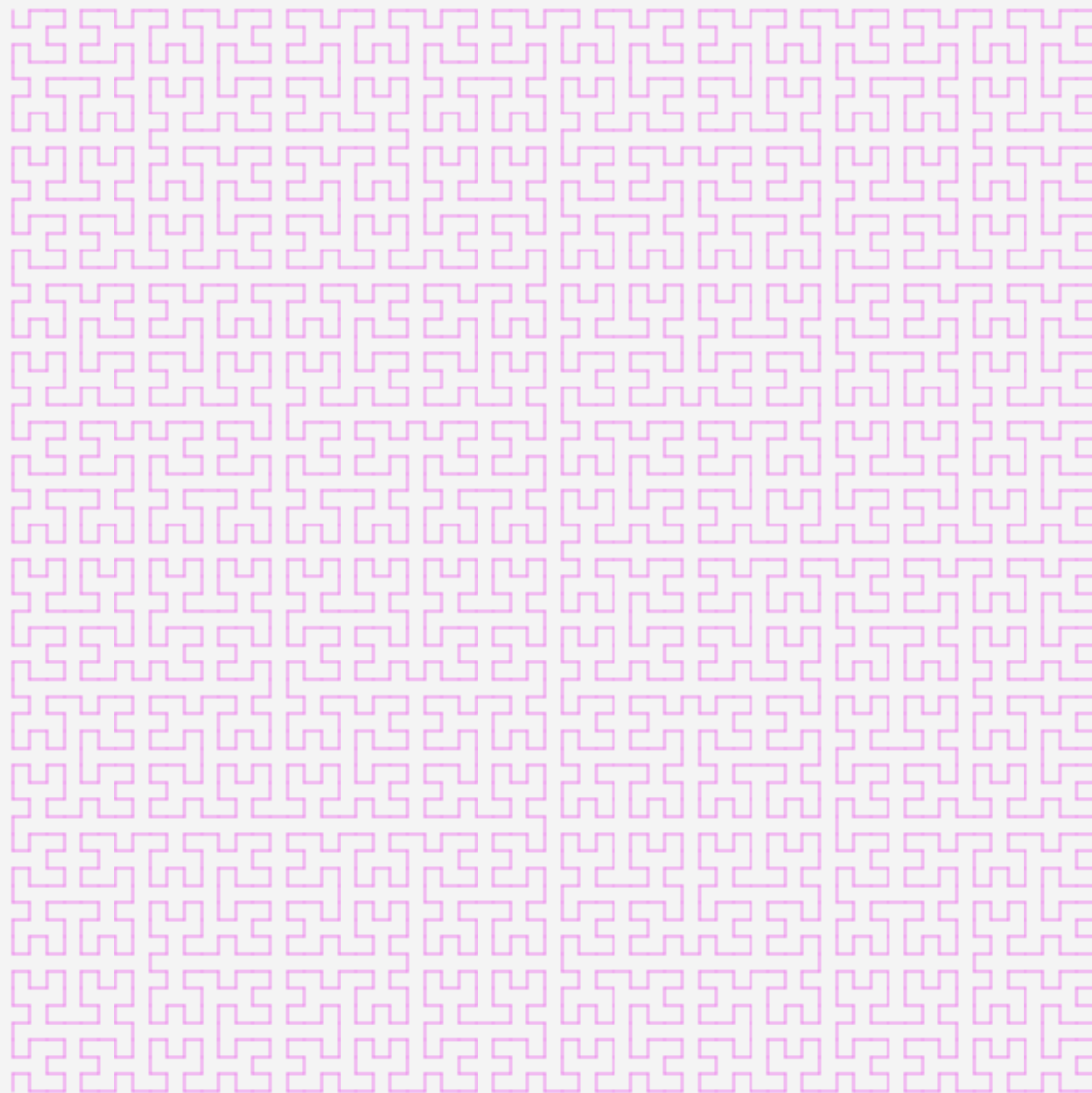
```
        drawPart(gc, lx, 0);  
        d(i - 1, gc);  
        drawPart(gc, 0, -ly);  
        b(i - 1, gc);  
    }  
}  
  
public static void main(String[] args) {  
    launch(args);  
}  
}
```

Результат работы программы:



Введите порядок кривой Гильберта:

Draw



Введите порядок кривой Гильберта:

Draw



Вывод: в ходе выполнения лабораторной работы я освоил возможности языка программирования Java в построении графических приложений .