Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 11

тема «GUI»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Синьковский Г.С.

Проверил: Нетбай Георгий Владимирович

Пермь, 2023

**Задание 1**

* 1. ***Постановка задачи***

Создать окно с двумя полями ввода и кнопкой. У первого поля установить ограничение по количеству вводимых символов 25 символов. Второе поле использовать для указания пути к файлу. При нажатии на кнопку должна происходить запись данных из поля ввода в файл путь к которому указывается во втором поле ввода.

***1.2. Решение задачи, код программы***

import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.BufferedWriter;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class Zadanie\_1 extends JFrame implements ActionListener {  
  
 private JTextField textField1;  
 private JTextField textField2;  
  
 public Zadanie\_1() {  
 setTitle("Запись в файл");  
 setSize(350, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JLabel label1 = new JLabel("Введите текст (до 25 символов) для записи в файл:");  
 textField1 = new JTextField(25);  
 JLabel label2 = new JLabel("Введите путь к файлу:");  
 textField2 = new JTextField(25);  
  
 JButton button = new JButton("Записать");  
  
 button.addActionListener(this);  
  
 JPanel panel = new JPanel();  
 panel.add(label1);  
 panel.add(textField1);  
 panel.add(label2);  
 panel.add(textField2);  
 panel.add(button);  
  
 // Добавление панели на окно  
 getContentPane().add(panel);  
  
 setVisible(true);  
 }  
  
 private void writeToFile(String text, String path) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(path))) {  
 if (text.length() <= 25) {  
 writer.write(text);  
 writer.flush();  
 JOptionPane.*showMessageDialog*(this, "Данные успешно сохранены в файл.", "Успех", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(this, "Ошибка: Длина текста должна быть не более 25 символов.", "Ошибка", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 } catch (IOException e) {  
 JOptionPane.*showMessageDialog*(this, "Ошибка при сохранении в файл: " + e.getMessage(), "Ошибка", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(Zadanie\_1::new);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String text = textField1.getText();  
 String path = textField2.getText();  
 writeToFile(text, path);  
 }  
}

***1.3. Тестирование работы программы с проверкой***

В таблице 1 представлено тестирование работы программы с проверкой решения задачи на языке Java.

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

**Задание 2**

***2.1. Постановка задачи***

1. Создать программу с графическим интерфейсом, которая предлагает два варианта использования
   1. Отображает поля ввода имени, фамилии, отчества, даты рождения, учебной группы. При нажатии на кнопку «Записать» сохраняет данные в файл
   2. При нажатии кнопки «Загрузить» извлекает из файла данные и отображает в полях ввода.

***2.2. Решение задачи, код программы***

import java.io.\*;  
import javax.swing.\*;  
  
public class Zadanie\_2 extends JFrame {  
  
 private JTextField lastNameField;  
 private JTextField firstNameField;  
 private JTextField middleNameField;  
 private JTextField birthDateField;  
 private JTextField groupField;  
  
 public Zadanie\_2() {  
 setTitle("Форма ввода данных");  
 setSize(320, 250);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JLabel lastNameLabel = new JLabel("Фамилия:");  
 lastNameField = new JTextField(20);  
 JLabel firstNameLabel = new JLabel("Имя:");  
 firstNameField = new JTextField(20);  
 JLabel middleNameLabel = new JLabel("Отчество:");  
 middleNameField = new JTextField(20);  
 JLabel birthDateLabel = new JLabel("Дата рождения:");  
 birthDateField = new JTextField(20);  
 JLabel groupLabel = new JLabel("Учебная группа:");  
 groupField = new JTextField(20);  
  
 JButton saveButton = new JButton("Записать");  
 JButton loadButton = new JButton("Загрузить");  
  
 saveButton.addActionListener(e -> saveDataToFile());  
  
 loadButton.addActionListener(e -> loadDataFromFile());  
  
 JPanel panel = new JPanel();  
 panel.add(lastNameLabel);  
 panel.add(lastNameField);  
 panel.add(firstNameLabel);  
 panel.add(firstNameField);  
 panel.add(middleNameLabel);  
 panel.add(middleNameField);  
 panel.add(birthDateLabel);  
 panel.add(birthDateField);  
 panel.add(groupLabel);  
 panel.add(groupField);  
 panel.add(saveButton);  
 panel.add(loadButton);  
  
 getContentPane().add(panel);  
  
 setVisible(true);  
 }  
  
 private void saveDataToFile() {  
 String lastName = lastNameField.getText();  
 String firstName = firstNameField.getText();  
 String middleName = middleNameField.getText();  
 String birthDate = birthDateField.getText();  
 String group = groupField.getText();  
  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter("2.txt"))) {  
 writer.write(lastName + "\n");  
 writer.write(firstName + "\n");  
 writer.write(middleName + "\n");  
 writer.write(birthDate + "\n");  
 writer.write(group + "\n");  
 writer.flush();  
 JOptionPane.*showMessageDialog*(this, "Данные успешно сохранены в файл.", "Успех", JOptionPane.*INFORMATION\_MESSAGE*);  
 } catch (IOException e) {  
 JOptionPane.*showMessageDialog*(this, "Ошибка при сохранении в файл: " + e.getMessage(), "Ошибка", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
  
 private void loadDataFromFile() {  
 try (BufferedReader reader = new BufferedReader(new FileReader("2.txt"))) {  
 String lastName = reader.readLine();  
 String firstName = reader.readLine();  
 String middleName = reader.readLine();  
 String birthDate = reader.readLine();  
 String group = reader.readLine();  
  
 lastNameField.setText(lastName);  
 firstNameField.setText(firstName);  
 middleNameField.setText(middleName);  
 birthDateField.setText(birthDate);  
 groupField.setText(group);  
  
 JOptionPane.*showMessageDialog*(this, "Данные успешно загружены из файла.", "Успех", JOptionPane.*INFORMATION\_MESSAGE*);  
 } catch (IOException e) {  
 JOptionPane.*showMessageDialog*(this, "Ошибка при загрузке из файла: " + e.getMessage(), "Ошибка", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(Zadanie\_2::new);  
 }  
}

***2.3. Тестирование работы программы с проверкой***

Далее в таблице 2 представлено тестирование работы программы с проверкой решения задач на языке Java

Таблица 2

Тестирование работы программы

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

**Задание 3**

***3.1. Постановка задачи***

Создать программу, которая запрашивает у пользователя интервал и строит на нем график функции из задания №10 лабораторной работы «Циклы» вашего варианта.

***3.2. Решение задачи, код программы***

import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.chart.LineChart;  
import javafx.scene.chart.NumberAxis;  
import javafx.scene.chart.XYChart;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
public class Zadanie\_3 extends Application {  
  
 private TextField fromTextField;  
 private TextField toTextField;  
 private Button plotButton;  
 private LineChart<Number, Number> lineChart;  
  
 @Override  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("График функции");  
  
 // Создание осей  
 final NumberAxis xAxis = new NumberAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 lineChart = new LineChart<>(xAxis, yAxis);  
  
 // Создание текстовых полей и кнопки  
 Label fromLabel = new Label("От:");  
 Label toLabel = new Label("До:");  
 fromTextField = new TextField();  
 toTextField = new TextField();  
 plotButton = new Button("Построить график");  
  
 // Обработка события нажатия на кнопку  
 plotButton.setOnAction(e -> plotGraph());  
  
 // Создание панели с элементами управления  
 HBox inputPane = new HBox(10);  
 inputPane.setAlignment(Pos.*CENTER*);  
 inputPane.getChildren().addAll(fromLabel, fromTextField, toLabel, toTextField, plotButton);  
  
 // Создание главной панели  
 VBox mainPane = new VBox(10);  
 mainPane.setPadding(new Insets(10));  
 mainPane.getChildren().addAll(inputPane, lineChart);  
  
 Scene scene = new Scene(mainPane, 800, 500);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 private void plotGraph() {  
 try {  
 double from = Double.*parseDouble*(fromTextField.getText());  
 double to = Double.*parseDouble*(toTextField.getText());  
  
 if (from > to) {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("Ошибка");  
 alert.setHeaderText(null);  
 alert.setContentText("Недопустимый интервал: значение \"От\" должно быть меньше значения \"До\".");  
 alert.showAndWait();  
 return;  
 }  
  
 XYChart.Series<Number, Number> series = new XYChart.Series<>();  
 series.setName("График функции");  
  
 for (double x = from; x <= to; x += 0.1) {  
 double y = 20 \* Math.*pow*(x, 2) - 5 \* Math.*pow*((Math.*sin*(x) + x), 2);  
 series.getData().add(new XYChart.Data<>(x, y));  
 }  
 lineChart.getData().clear();  
 lineChart.setCreateSymbols(false);  
 lineChart.getData().add(series);  
  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("График успешно построен.");  
 alert.showAndWait();  
 } catch (NumberFormatException ex) {  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("Ошибка");  
 alert.setHeaderText(null);  
 alert.setContentText("Некорректный ввод данных.");  
 alert.showAndWait();  
 }  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

***3.3. Тестирование работы программы с проверкой***

Далее в таблице 3 представлено тестирование работы программы с проверкой решения задач на языке Java.

Таблица 3

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |

**Задание 4**

***4.1. Постановка задачи***

Создать программу, которая при нажатии кнопки вычисляет 7 случайных чисел и отображает их одно под другим в текстовом поле ввода. Поле ввода должно быть неизменяемо для пользователя!

***4.2. Решение задачи, код программы***

import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextArea;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.GridPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Random;  
  
public class Zadanie\_4 extends Application {  
  
 private TextField fromTextField;  
 private TextField toTextField;  
 private TextArea resultTextArea;  
  
 @Override  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("Генератор случайных чисел");  
  
 Label fromLabel = new Label("От:");  
 Label toLabel = new Label("До:");  
 fromTextField = new TextField();  
 toTextField = new TextField();  
  
 Button generateButton = new Button("Генерировать");  
 generateButton.setOnAction(e -> generateNumbers());  
  
 resultTextArea = new TextArea();  
 resultTextArea.setEditable(false);  
  
 GridPane inputPane = new GridPane();  
 inputPane.setHgap(10);  
 inputPane.setVgap(10);  
 inputPane.setPadding(new Insets(10));  
 inputPane.addRow(0, fromLabel, fromTextField);  
 inputPane.addRow(1, toLabel, toTextField);  
  
 VBox mainPane = new VBox(10);  
 mainPane.setPadding(new Insets(10));  
 mainPane.getChildren().addAll(inputPane, generateButton, resultTextArea);  
  
 Scene scene = new Scene(mainPane, 300, 300);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 private void generateNumbers() {  
 try {  
 int from = Integer.*parseInt*(fromTextField.getText());  
 int to = Integer.*parseInt*(toTextField.getText());  
  
 List<Integer> randomNumbers = new ArrayList<>();  
 Random random = new Random();  
  
 for (int i = 0; i < 7; i++) {  
 int randomNumber = random.nextInt(to - from + 1) + from;  
 randomNumbers.add(randomNumber);  
 }  
  
 StringBuilder sb = new StringBuilder();  
 for (int number : randomNumbers) {  
 sb.append(number).append("\n");  
 }  
 resultTextArea.setText(sb.toString());  
 } catch (NumberFormatException ex) {  
 resultTextArea.setText("Ошибка: некорректный ввод интервала.");  
 }  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

***4.3. Тестирование работы программы с проверкой***

Далее в таблице 4 представлено тестирование работы программы с проверкой решения задач на языке Java.

Таблица 4

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

**Задание 5**

***5.1. Постановка задачи***

Построить график фигуры попадания точки в область из задания №5 лабораторной работы «Ветвящиеся алгоритмы» Вашего варианта.

***5.2. Решение задачи, код программы***

import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.scene.Scene;  
import javafx.scene.chart.LineChart;  
import javafx.scene.chart.NumberAxis;  
import javafx.scene.chart.XYChart;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
public class Zadanie\_5 extends Application {  
  
 private LineChart<Number, Number> lineChart;  
 private TextField xTextField;  
 private TextField yTextField;  
 private Label resultLabel;  
  
 public Zadanie\_5() {  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("График попадания точки в область");  
  
 final NumberAxis xAxis = new NumberAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 lineChart = new LineChart<>(xAxis, yAxis);  
  
 VBox mainPane = new VBox(20);  
 mainPane.setPadding(new Insets(20));  
  
 HBox inputPane = new HBox(10);  
 inputPane.setPadding(new Insets(10));  
  
 xTextField = new TextField();  
 yTextField = new TextField();  
 Button checkButton = new Button("Проверить");  
 checkButton.setOnAction(e -> checkPoint());  
  
 inputPane.getChildren().addAll(  
 new Label("X: "), xTextField,  
 new Label("Y: "), yTextField,  
 checkButton  
 );  
  
 resultLabel = new Label();  
  
 mainPane.getChildren().addAll(lineChart, inputPane, resultLabel);  
  
 plotGraph();  
  
 Scene scene = new Scene(mainPane, 600, 600);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 private void plotGraph() {  
  
 XYChart.Series<Number, Number> series1 = new XYChart.Series<>();  
 series1.setName("Серия 1");  
 for (double x = 0; x <= 1; x += 0.01) {  
 double y=3\*x-1;  
 series1.getData().add(new XYChart.Data<>(x, y));  
 }  
 XYChart.Series<Number, Number> series2 = new XYChart.Series<>();  
 series2.setName("Серия 2");  
 for (double x = -1; x <= 0; x += 0.01) {  
 double y=-2\*x-1;  
 series2.getData().add(new XYChart.Data<>(x, y));  
 }  
 XYChart.Series<Number, Number> series3 = new XYChart.Series<>();  
 series3.setName("Серия 3");  
 for (double x = -1; x <= 1; x += 0.01) {  
 double y=-1.5\*x+3.5;  
 series3.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series4 = new XYChart.Series<>();  
 series4.setName("Серия 4");  
 for (double x = -3; x <= -1; x += 0.01) {  
 double y=Math.*sqrt*(4-Math.*pow*((x+1),2))+3;  
 series4.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series5 = new XYChart.Series<>();  
 series5.setName("Серия 5");  
 for (double x = -3; x <= -1; x += 0.01) {  
 double y=-Math.*sqrt*(4-Math.*pow*((x+1),2))+3;  
 series5.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series6 = new XYChart.Series<>();  
 series6.setName("Серия 6");  
 for (double x = 2; x <= 7; x += 0.01) {  
 double y=-0.2\*x+1.4;  
 series6.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series7 = new XYChart.Series<>();  
 series7.setName("Серия 7");  
 for (double x = 2; x <= 3; x += 0.01) {  
 double y=-x+3;  
 series7.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series8 = new XYChart.Series<>();  
 series8.setName("Серия 8");  
 for (double x = 6; x <= 7; x += 0.01) {  
 double y=x-7;  
 series8.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series9 = new XYChart.Series<>();  
 series9.setName("Серия 9");  
 for (double x = 2; x <= 3; x += 0.01) {  
 double y=x-3;  
 series9.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 XYChart.Series<Number, Number> series10 = new XYChart.Series<>();  
 series10.setName("Серия 10");  
 for (double x = 2; x <= 6; x += 0.01) {  
 double y = -Math.*sqrt*(4 - Math.*pow*(x-4, 2)) - 1;  
 series10.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 lineChart.getData().clear();  
 lineChart.setCreateSymbols(false);  
 lineChart.getData().addAll(  
 series1, series2, series3, series4, series5, series6, series7, series8, series9, series10  
 );  
 }  
 private void checkPoint() {  
 double x = Double.*parseDouble*(xTextField.getText());  
 double y = Double.*parseDouble*(yTextField.getText());  
  
 boolean isInside = checkInsideArea(x, y);  
 resultLabel.setText(isInside ? "Точка попадает в область" : "Точка не попадает в область");  
 }  
  
 private boolean checkInsideArea(double x, double y) {  
 // Условие для попадания точки в область  
 return (x >= -1 && y >= 2.5 \* x + 3.5 && y <= -1.5 \* x + 3.5)  
 || (x <= 0 && y <= 2.5 \* x + 3.5 && y >= -2 \* x - 1)  
 || (x >= 0 && y <= -1.5 \* x + 3.5 && y >= 3 \* x - 1)  
 || (y >= -1 && y <= x - 3 && y <= -1f / 3 \* x + 1)  
 || (y <= 0 && y >= x - 7 && y >= -1f / 3 \* x + 1)  
 || (y >= 0 && y >= -x + 3 && y <= - 0.2 \* x + 1.4)  
 || (Math.*pow*((x + 1), 2) + Math.*pow*((y - 3), 2) <= 4 && x <= -1)  
 || (Math.*pow*((x - 4), 2) + Math.*pow*((y + 1), 2) <= 4 && y <= -1);  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

***5.3. Тестирование работы программы с проверкой***

Далее в таблице 5 представлено тестирование работы программы с проверкой решения задач на языке Java.

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | График из лабы 5 |
| 1 |  |  |
| 2 |  |