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 Братчиков З.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***

{  
  
 private final *JTextField* textField1;  
 private final *JTextField* textField2;  
  
 public z\_1() {  
 setTitle("Запись в файл");  
 setSize(300, 250);  
 setDefaultCloseOperation(*JFrame*.EXIT\_ON\_CLOSE);  
  
 *// Создание метки и полей ввода  
 JLabel* label1 = new JLabel("Текст (макс. 25 символов):");  
 textField1 = new JTextField(25);  
 *JLabel* label2 = new JLabel("Путь к файлу:");  
 textField2 = new JTextField(25);  
  
 *// Создание кнопки  
 JButton* button = new JButton("Записать в файл");  
  
 *// Добавление обработчика события нажатия на кнопку* button.addActionListener(*e* -> {  
 *String* text = textField1.getText();  
 *String* path = textField2.getText();  
 writeToFile(text, path);  
 });  
  
 *// Создание панели и добавление компонентов на неё  
 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();  
 *System*.out.println("Данные успешно записаны в файл.");  
 }else {  
 *System*.out.println("Ошибка при записи в файл: ");  
 }  
 } catch (*IOException e*) {  
 *System*.out.println("Ошибка при записи в файл: " + *e*.getMessage());  
 }  
 }  
  
 public static void main(*String*[] *args*) {  
 new z\_1();  
 }  
}

***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 |  |

**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 *z\_2* extends *JFrame* {  
  
 public *JTextField* lastNameField;  
 public final *JTextField* firstNameField;  
 public *JTextField* middleNameField;  
 public *JTextField* birthDateField;  
 public *JTextField* groupField;  
  
 public z\_2() {  
 setTitle("Форма ввода данных");  
 setSize(300, 250);  
 setDefaultCloseOperation(*JFrame*.EXIT\_ON\_CLOSE);  
  
 *// Создание меток и полей ввода  
 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);  
  
 *// Создание кнопок  
 JButton* saveButton = new JButton("Записать");  
 *JButton* loadButton = new JButton("Загрузить");  
  
 *// Добавление обработчика события нажатия на кнопку "Записать"* saveButton.addActionListener(*e* -> saveDataToFile());  
  
 *// Добавление обработчика события нажатия на кнопку "Загрузить"* loadButton.addActionListener(*e* -> loadDataFromFile());  
  
 *// Создание панели и добавление компонентов на неё  
 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);  
  
 *// Добавление панели на окно* 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("data.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 записаны в файл.");  
 } catch (*IOException e*) {  
 *JOptionPane*.*showMessageDialog*(this, "Ошибка при записи в файл: " + *e*.getMessage());  
 }  
 }  
  
 private void loadDataFromFile() {  
 try (*BufferedReader* reader = new BufferedReader(new FileReader("data.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.");  
 } catch (*IOException e*) {  
 *JOptionPane*.*showMessageDialog*(this, "Ошибка при загрузке из файла: " + *e*.getMessage());  
 }  
 }  
  
 public static void main(*String*[] *args*) {  
 new z\_2();  
 }  
}

***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.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 *z\_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, 600);  
 *primaryStage*.setScene(scene);  
 *primaryStage*.show();  
 }  
  
 private void plotGraph() {  
 double from = *Double*.*parseDouble*(fromTextField.getText());  
 double to = *Double*.*parseDouble*(toTextField.getText());  
  
 *// Создание серии данных* XYChart.*Series*<Number, Number> series = new XYChart.Series<>();  
 series.setName("График функции");  
  
 *// Заполнение серии данных значениями функции на интервале* for (double x = from; x <= to; x += 0.1) {  
 double y = *Math*.*cos*(x) - *Math*.*sin*(*Math*.*pow*(x, 2));  
 series.getData().add(new XYChart.Data<>(x, y));  
 }  
 *// Очистка предыдущих серий данных и добавление новой серии на график* lineChart.getData().clear();  
 lineChart.getData().add(series);  
 }  
  
 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 |  |

**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.TextArea*;  
import *javafx.scene.layout.VBox*;  
import *javafx.stage.Stage*;  
  
import *java.util.Random*;  
  
public class *z\_4* extends Application {  
  
 private *TextArea* resultTextArea;  
  
 *@Override* public void start(*Stage primaryStage*) {  
 *primaryStage*.setTitle("Генератор случайных чисел");  
  
 *// Создание кнопки  
 Button* generateButton = new Button("Генерировать");  
 generateButton.setOnAction(*e* -> generateNumbers());  
  
 *// Создание текстового поля для отображения результатов* resultTextArea = new TextArea();  
 resultTextArea.setEditable(false);  
  
 *// Создание главной панели  
 VBox* mainPane = new VBox(10);  
 mainPane.setPadding(new Insets(10));  
 mainPane.getChildren().addAll(generateButton, resultTextArea);  
  
 *Scene* scene = new Scene(mainPane, 300, 200);  
 *primaryStage*.setScene(scene);  
 *primaryStage*.show();  
 }  
  
 private void generateNumbers() {  
 *Random* random = new Random();  
 *StringBuilder* sb = new StringBuilder();  
  
 *// Генерация 7 случайных чисел* for (int i = 0; i < 7; i++) {  
 int randomNumber = random.nextInt();  
 sb.append(randomNumber).append("\n");  
 }  
  
 *// Отображение результатов в текстовом поле* resultTextArea.setText(sb.toString());  
 }  
  
 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.layout.VBox*;  
import *javafx.stage.Stage*;  
  
public class *z\_5* extends Application {  
  
 private *LineChart*<Number, Number> lineChart;  
  
 public z\_5() {  
 }  
  
 *@Override* public void start(*Stage primaryStage*) {  
 *primaryStage*.setTitle("График попадания точки в область");  
  
 *// Создание осей* final *NumberAxis* xAxis = new NumberAxis();  
 final *NumberAxis* yAxis = new NumberAxis();  
 lineChart = new LineChart<>(xAxis, yAxis);  
  
 *// Создание главной панели  
 VBox* mainPane = new VBox(20);  
 mainPane.setPadding(new Insets(20));  
 mainPane.getChildren().add(lineChart);  
  
 *// Построение графика* plotGraph();  
  
 *Scene* scene = new Scene(mainPane, 600, 600);  
 *primaryStage*.setScene(scene);  
 *primaryStage*.show();  
 }  
  
 private void plotGraph() {  
 *// Создание серии данных 1* XYChart.*Series*<Number, Number> series1 = new XYChart.Series<>();  
 series1.setName("Серия 1");  
  
 for (double x = -3; x <= -1; x += 0.01) {  
 double y = *Math*.*sqrt*(3 - *Math*.*pow*(x, 2) - 2 \* x) + 1;  
 series1.getData().add(new XYChart.Data<>(x, y));  
 }  
 *// Создание серии данных 2* XYChart.*Series*<Number, Number> series2 = new XYChart.Series<>();  
 series2.setName("Серия 2");  
 for (double y = 3; y <= 5; y += 0.01) {  
 double x = -1;  
 series2.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//3* XYChart.*Series*<Number, Number> series3 = new XYChart.Series<>();  
 series3.setName("Серия 3");  
 for (double x = -1; x <= 2; x += 0.01) {  
 double y = 5;  
 series3.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//4* XYChart.*Series*<Number, Number> series4 = new XYChart.Series<>();  
 series4.setName("Серия 4");  
 for (double x = 2; x >= 0; x -= 0.01) {  
 double y = 1.5 \* x + 2;  
 series4.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//5* XYChart.*Series*<Number, Number> series5 = new XYChart.Series<>();  
 series5.setName("Серия 5");  
 for (double x = 0; x <= 2; x += 0.01) {  
 double y = 2 - x;  
 series5.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//6* XYChart.*Series*<Number, Number> series6 = new XYChart.Series<>();  
 series6.setName("Серия 6");  
 for (double x = 2; x >= -2; x -= 0.01) {  
 double y = -0.25 \* x + 0.5;  
 series6.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//7* XYChart.*Series*<Number, Number> series7 = new XYChart.Series<>();  
 series7.setName("Серия 7");  
 for (double x = -2; x <= 0; x += 0.01) {  
 double y = -x - 1;  
 series7.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//8* XYChart.*Series*<Number, Number> series8 = new XYChart.Series<>();  
 series8.setName("Серия 8");  
 for (double x = 0; x <= 1; x += 0.01) {  
 double y = -3 \* x - 1;  
 series8.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//9* XYChart.*Series*<Number, Number> series9 = new XYChart.Series<>();  
 series9.setName("Серия 9");  
 for (double x = 1; x >= -1; x -= 0.01) {  
 double y = -1.5 \* x - 2.5;  
 series9.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *//10* XYChart.*Series*<Number, Number> series10 = new XYChart.Series<>();  
 series10.setName("Серия 10");  
 for (double x = -1; x >= -3; x -= 0.01) {  
 double y = -*Math*.*sqrt*(3 - *Math*.*pow*(x, 2) - 2 \* x) + 1;  
 series10.getData().add(new XYChart.Data<>(x, y));  
 }  
 double x0 = -3;  
 double y0 = *Math*.*sqrt*(3 - *Math*.*pow*(x0, 2) - 2 \* x0) + 1;  
 series10.getData().add(new XYChart.Data<>(x0, y0));  
 *//11* XYChart.*Series*<Number, Number> series11 = new XYChart.Series<>();  
 series11.setName("Серия 11");  
 for (double x = 2; x <= 3; x += 0.01) {  
 double y = 3\*x-5;  
 series11.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//12* XYChart.*Series*<Number, Number> series12 = new XYChart.Series<>();  
 series12.setName("Серия 12");  
 for (double x = 3; x <= 4; x += 0.01) {  
 double y = -3\*x+13;  
 series12.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//13* XYChart.*Series*<Number, Number> series13 = new XYChart.Series<>();  
 series13.setName("Серия 13");  
 for (double x = 6; x <= 7; x += 0.01) {  
 double y = -2\*x+11;  
 series13.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//14* XYChart.*Series*<Number, Number> series14 = new XYChart.Series<>();  
 series14.setName("Серия 14");  
 for (double x = 5; x <= 7; x += 0.01) {  
 double y = x-10;  
 series14.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//15* XYChart.*Series*<Number, Number> series15 = new XYChart.Series<>();  
 series15.setName("Серия 15");  
 for (double x = 3; x <= 5; x += 0.01) {  
 double y = -0.5\*x-2.5;  
 series15.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//16* XYChart.*Series*<Number, Number> series16 = new XYChart.Series<>();  
 series16.setName("Серия 16");  
 for (double x = 3; x <= 4; x += 0.01) {  
 double y = x-7;  
 series16.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//17* XYChart.*Series*<Number, Number> series17 = new XYChart.Series<>();  
 series17.setName("Серия 17");  
 for (double x = 2; x <= 3; x += 0.01) {  
 double y = -x+1;  
 series17.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//18* XYChart.*Series*<Number, Number> series18 = new XYChart.Series<>();  
 series18.setName("Серия 18");  
 for (double x = 3; x <=4; x += 0.01) {  
 double y = 2\*x-8;  
 series18.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//19* XYChart.*Series*<Number, Number> series19 = new XYChart.Series<>();  
 series19.setName("Серия 19");  
 for (double x = 2; x <= 4; x += 0.01) {  
 double y = -0.5\*x+2;  
 series19.getData().add(new XYChart.Data<>(x, y));  
 }  
 *//20* XYChart.*Series*<Number, Number> series20 = new XYChart.Series<>();  
 series20.setName("Серия 20");  
 for (double x = 4; x <= 6; x += 0.01) {  
 double y = *Math*.*sqrt*(-12 - *Math*.*pow*(x, 2) + 8 \* x) - 1;  
 series20.getData().add(new XYChart.Data<>(x, y));  
 }  
 *// Создание серии данных 21* XYChart.*Series*<Number, Number> series21 = new XYChart.Series<>();  
 series21.setName("Серия 21");  
 for (double x = 2; x <= 4; x += 0.01) {  
 double y = -*Math*.*sqrt*(-12 - *Math*.*pow*(x, 2) + 8 \* x) - 1;  
 series21.getData().add(new XYChart.Data<>(x, y));  
 }  
  
 *// Очистка предыдущих серий данных и добавление новых серий на график* lineChart.getData().clear();  
 lineChart.setCreateSymbols(false); *// Отключение отображения точек данных* lineChart.getData().addAll(  
 series1, series2, series3, series4, series5, series6, series7, series8, series9, series10,  
 series11, series12, series13, series14, series15, series16, series17, series18, series19, series20, series21  
 );  
 }  
  
 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 в задании Лабораторной работы №5 |
| 1 |  |  |