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 Радостев К.П.

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.awt.event.KeyAdapter;  
import java.awt.event.KeyEvent;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class Zad1 implements ActionListener {  
 private static JLabel *userLabel*;  
 private static JTextField *userText*;  
 private static JLabel *fileLabel*;  
 private static JTextField *fileText*;  
 private static JButton *button*;  
 private static JLabel *res*;  
 public static JFrame *frame*;  
 public static JPanel *panel*;  
  
 public static void main(String[] args) {  
 *frame* = new JFrame();  
 *panel* = new JPanel();  
  
 *frame*.setSize(350, 250);  
 *frame*.setTitle("Задание №1");  
 *frame*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 *frame*.add(*panel*);  
 *panel*.setLayout(null);  
  
 *userLabel* = new JLabel("Текст") ;  
 *userLabel*.setBounds(10, 20, 80, 25);  
 *panel*.add(*userLabel*);  
  
 *userText* = new JTextField();  
 *userText*.addKeyListener(new KeyAdapter() {  
 @Override  
 public void keyTyped(KeyEvent e) {  
 if (*userText*.getText().length() >= 25 ) // ограничение 25 символов  
 e.consume();  
 }  
 });  
 *userText*.setBounds(100, 20, 165, 25);  
 *panel*.add(*userText*);  
  
 *fileLabel* = new JLabel("Путь к файлу");  
 *fileLabel*.setBounds(10, 50, 80, 25);  
 *panel*.add(*fileLabel*);  
  
 *fileText* = new JTextField();  
 *fileText*.setBounds(100, 50, 165, 25);  
 *panel*.add(*fileText*);  
  
 *button* = new JButton("Запись");  
 *button*.setBounds(10, 80, 80, 25);  
 *button*.addActionListener(new Zad1());  
 *panel*.add(*button*);  
  
 *res* = new JLabel("Максимум 25 символов");  
 *res*.setBounds(10, 110, 300, 25);  
 *panel*.add(*res*);  
  
 *frame*.setVisible(true);  
 } // GUI  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String text = *userText*.getText();  
 String filehome = *fileText*.getText();  
  
 try(FileWriter writer = new FileWriter(filehome, false))  
 {  
 // запись всей строки  
 writer.write(text);  
 // запись по символам  
 writer.flush();  
 *res*.setText("Текст записан!");  
 }  
 catch(IOException ex){  
  
 *res*.setText(ex.getMessage());  
 }  
 } // Запись в файл  
  
}

***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 javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.KeyAdapter;  
import java.awt.event.KeyEvent;   
import java.io.FileWriter;  
import java.io.IOException;  
import java.nio.charset.StandardCharsets;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.nio.file.StandardOpenOption;  
import java.util.Arrays;  
import java.util.List;  
  
public class Zad2 {  
 private static JLabel *name1Label*; // имя  
 private static JLabel *name2Label*; // фамилия  
 private static JLabel *name3Label*; // отчество  
 private static JLabel *wasBornLabel*; // дата рождения  
 private static JLabel *classLabel*; // учебная группа  
 private static JTextField *name1Text*;  
 private static JTextField *name2Text*;  
 private static JTextField *name3Text*;  
 private static JTextField *wasBornText*;  
 private static JTextField *classText*;  
 private static JTextField *feedBackText*;  
 private static JLabel *feedBackLabel*;  
 private static JButton *write*;  
 private static JButton *read*;  
  
 public static JFrame *frame*;  
 public static JPanel *panel*;  
  
 public static void main(String[] args) {  
 *frame* = new JFrame();  
 *panel* = new JPanel();  
  
 *frame*.setSize(445, 218);  
 *frame*.setTitle("Задание №2");  
 *frame*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 *frame*.add(*panel*);  
 *panel*.setLayout(null);  
  
 *name1Label* = new JLabel("Имя");  
 *name1Label*.setBounds(10, 20, 80, 25);  
 *panel*.add(*name1Label*);  
  
 *name2Label* = new JLabel("Фамилия");  
 *name2Label*.setBounds(10, 50, 80, 25);  
 *panel*.add(*name2Label*);  
  
 *name3Label* = new JLabel("Отчество");  
 *name3Label*.setBounds(10, 80, 80, 25);  
 *panel*.add(*name3Label*);  
  
 *wasBornLabel* = new JLabel("Дата рождения");  
 *wasBornLabel*.setBounds(10, 110, 80, 25);  
 *panel*.add(*wasBornLabel*);  
  
 *classLabel* = new JLabel("Уч. группа");  
 *classLabel*.setBounds(10, 140, 80, 25);  
 *panel*.add(*classLabel*);  
  
  
 *name1Text* = new JTextField();  
 *name1Text*.setBounds(100, 20, 165, 25);  
 *panel*.add(*name1Text*);  
  
 *name2Text* = new JTextField();  
 *name2Text*.setBounds(100, 50, 165, 25);  
 *panel*.add(*name2Text*);  
  
 *name3Text* = new JTextField();  
 *name3Text*.setBounds(100, 80, 165, 25);  
 *panel*.add(*name3Text*);  
  
 *wasBornText* = new JTextField();  
 *wasBornText*.setBounds(100, 110, 165, 25);  
 *panel*.add(*wasBornText*);  
  
 *classText* = new JTextField();  
 *classText*.setBounds(100, 140, 165, 25);  
 *panel*.add(*classText*);  
  
 *write* = new JButton("Write");  
 *write*.setBounds(270, 20, 70, 25);  
 *panel*.add(*write*);  
  
 *read* = new JButton("Read");  
 *read*.setBounds(350, 20, 70, 25);  
 *panel*.add(*read*);  
  
 *feedBackText* = new JTextField();  
 *feedBackText*.addKeyListener(new KeyAdapter() {  
 @Override  
 public void keyTyped(KeyEvent e) {  
 if (*feedBackText*.getText().length() >= 0 ) // ограничение 25 символов  
 e.consume();  
 }  
 });  
 *feedBackText*.setBounds(270, 80, 150, 85);  
 *panel*.add(*feedBackText*);  
  
 *feedBackLabel* = new JLabel("Окно взаимодействия:");  
 *feedBackLabel*.setBounds(270, 50, 150, 25);  
 *panel*.add(*feedBackLabel*);  
  
 *write*.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 String filehome = "D:/home.txt";  
 try (FileWriter writer = new FileWriter(filehome, false)) {  
 // запись всей строки  
 List<String> lines = Arrays.*asList*(*name1Text*.getText(), *name2Text*.getText(),  
 *name3Text*.getText(), *wasBornText*.getText(), *classText*.getText());  
 Files.*write*(Paths.*get*(filehome), lines, StandardOpenOption.*CREATE*);  
 *feedBackText*.setText(" Данные записаны!");  
 } catch (IOException ex) {  
  
 *feedBackText*.setText(" Файл не найден!");  
 }  
 } // Запись в файл}  
 }); // Запись в файл  
 *read*.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 Path path = Paths.*get*("D:/gg.txt");  
 try {  
 List<String> lines = Files.*readAllLines*(path, StandardCharsets.*UTF\_8*);  
 *name1Text*.setText(lines.get(0)); *name2Text*.setText(lines.get(1)); *name3Text*.setText(lines.get(2));  
 *wasBornText*.setText(lines.get(3)); *classText*.setText(lines.get(4));  
 *feedBackText*.setText(" Данные прочитаны!");  
 }  
 catch (IOException q) {  
 *feedBackText*.setText(" Файл не найден!");  
 }  
 }}); // Чтение из файла  
  
 *frame*.setVisible(true);  
 }  
}

***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.Node;  
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 Zad3 extends Application {  
 private TextField from;  
 private TextField to;  
 private Button button;  
 private LineChart<Number, Number> lineChart;  
  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("Задание №3");  
 NumberAxis xAxis = new NumberAxis();  
 NumberAxis yAxis = new NumberAxis();  
 this.lineChart = new LineChart(xAxis, yAxis);  
 Label fromLabel = new Label("От:");  
 Label toLabel = new Label("До:");  
 this.from = new TextField();  
 this.to = new TextField();  
 this.button = new Button("Построить");  
 this.button.setOnAction((e) -> {  
 this.plotGraph();  
 });  
 HBox inputPane = new HBox(10.0);  
 inputPane.setAlignment(Pos.*CENTER*);  
 inputPane.getChildren().addAll(new Node[]{fromLabel, this.from, toLabel, this.to, this.button});  
 VBox mainPane = new VBox(10.0);  
 mainPane.setPadding(new Insets(10.0));  
 mainPane.getChildren().addAll(new Node[]{inputPane, this.lineChart});  
 Scene scene = new Scene(mainPane, 500, 500);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 private void plotGraph() {  
 double from = Double.*parseDouble*(this.from.getText());  
 double to = Double.*parseDouble*(this.to.getText());  
 XYChart.Series<Number, Number> series = new XYChart.Series();  
 series.setName("График функции");  
  
 for(double x = from; x <= to; x += 0.1) {  
 double y = Math.*log*(Math.*abs*(Math.*pow*(x, 2)-5\*x)+5\*Math.*sin*(x));  
 series.getData().add(new XYChart.Data(x, y));  
 }  
  
 this.lineChart.getData().clear();  
 this.lineChart.setCreateSymbols(false);  
 this.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 | desmos |

**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 java.util.Random;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.scene.Node;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextArea;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
public class Zad4 extends Application {  
 private TextArea resultTextArea;  
  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("Задание №4");  
 Button generateButton = new Button("Рандом");  
 generateButton.setOnAction((e) -> {  
 this.generateNumbers();  
 });  
 this.resultTextArea = new TextArea();  
 this.resultTextArea.setEditable(false);  
 VBox mainPane = new VBox(12.0);  
 mainPane.setPadding(new Insets(10.0));  
 mainPane.getChildren().addAll(new Node[]{generateButton, this.resultTextArea});  
 Scene scene = new Scene(mainPane, 300.0, 200.0);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 private void generateNumbers() {  
 Random random = new Random();  
 StringBuilder sb = new StringBuilder();  
  
 for(int i = 0; i < 7; ++i) {  
 int randomNumber = random.nextInt();  
 sb.append(randomNumber).append("\n");  
 }  
  
 this.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.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 Zad5 extends Application {  
  
 private LineChart<Number, Number> lineChart;  
 private TextField xText;  
 private TextField yText;  
 private Label result;  
  
 @Override  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("Задание №5");  
  
 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));  
  
 xText = new TextField();  
 yText = new TextField();  
 Button checkButton = new Button("Проверить");  
 checkButton.setOnAction(e -> checkPoint());  
  
 inputPane.getChildren().addAll(  
 new Label("X: "), xText,  
 new Label("Y: "), yText,  
 checkButton  
 );  
  
 result = new Label();  
  
 mainPane.getChildren().addAll(lineChart, inputPane, result);  
  
 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*(xText.getText());  
 double y = Double.*parseDouble*(yText.getText());  
  
 boolean isInside = checkInsideArea(x, y);  
 result.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 |  |  |