Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 5

тема «GUI в Java»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Михайлов Александр

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

Оглавление

Оглавление

[Вариант 14 4](#_Toc136898360)

[Задание 1 4](#_Toc136898361)

[Постановка задачи 4](#_Toc136898362)

[Решение задачи 4](#_Toc136898363)

[Проверка результата 5](#_Toc136898364)

[Задание 2 6](#_Toc136898365)

[Постановка задачи 6](#_Toc136898366)

[Решение задачи 6](#_Toc136898367)

[Проверка результата 8](#_Toc136898368)

[Задание 3 9](#_Toc136898369)

[Постановка задачи 9](#_Toc136898370)

[Решение задачи 9](#_Toc136898371)

[Проверка результата 11](#_Toc136898372)

[Задание 4 12](#_Toc136898373)

[Постановка задачи 12](#_Toc136898374)

[Решение задачи 12](#_Toc136898375)

[Проверка результата 13](#_Toc136898376)

[Задание 5 14](#_Toc136898377)

[Постановка задачи 14](#_Toc136898378)

[Решение задачи 14](#_Toc136898379)

[Проверка результата 17](#_Toc136898380)

# Вариант 14

# Задание 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.

## Решение задачи

Код программы:

import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.FileWriter;  
import java.io.IOException;  
  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JTextField;  
  
public class Main extends JFrame {  
  
 private static final long *serialVersionUID* = 1L;  
  
 private JLabel label1, label2;  
 private JTextField textField1, textField2;  
 private JButton button;  
  
 public Main() {  
 setTitle("Запись данных в файл");  
 setSize(400, 200);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(null);  
  
 label1 = new JLabel("Введите текст (максимум 25 символов):");  
 label1.setBounds(20, 20, 250, 20);  
 add(label1);  
  
 textField1 = new JTextField();  
 textField1.setBounds(20, 50, 250, 20);  
 textField1.setDocument(new JTextFieldLimit(25));  
 add(textField1);  
  
 label2 = new JLabel("Укажите путь к файлу:");  
 label2.setBounds(20, 80, 250, 20);  
 add(label2);  
  
 textField2 = new JTextField();  
 textField2.setBounds(20, 110, 250, 20);  
 add(textField2);  
  
 button = new JButton("Записать в файл");  
 button.setBounds(100, 140, 150, 30);  
 button.addActionListener(new ActionListener() {  
   
 public void actionPerformed(ActionEvent e) {  
 String text = textField1.getText();  
 String path = textField2.getText();  
  
 try {  
 FileWriter writer = new FileWriter(path);  
 writer.write(text);  
 writer.close();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
 });  
 add(button);  
  
 setVisible(true);  
 }  
  
 public static void main(String[] args) {  
 new Main();  
 }  
  
}  
  
class JTextFieldLimit extends javax.swing.text.PlainDocument {  
 private static final long *serialVersionUID* = 1L;  
 private int limit;  
  
 JTextFieldLimit(int limit) {  
 super();  
 this.limit = limit;  
 }  
  
 public void insertString(int offset, String str, javax.swing.text.AttributeSet attr) throws javax.swing.text.BadLocationException {  
 if (str == null)  
 return;  
  
 if ((getLength() + str.length()) <= limit) {  
 super.insertString(offset, str, attr);  
 }  
 }  
}

## Проверка результата

|  |
| --- |
|  |

# Задание 2

## Постановка задачи

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.

## Решение задачи

Код программы:

import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JTextField;  
  
public class Main extends JFrame {  
  
 private static final long *serialVersionUID* = 1L;  
  
 private JLabel label1, label2, label3, label4, label5;  
 private JTextField textField1, textField2, textField3, textField4, textField5;  
 private JButton buttonSave, buttonLoad;  
  
 public Main() {  
 setTitle("Работа с файлом");  
 setSize(400, 300);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(null);  
  
 label1 = new JLabel("Имя:");  
 label1.setBounds(20, 20, 100, 20);  
 add(label1);  
  
 textField1 = new JTextField();  
 textField1.setBounds(120, 20, 250, 20);  
 add(textField1);  
  
 label2 = new JLabel("Фамилия:");  
 label2.setBounds(20, 50, 100, 20);  
 add(label2);  
  
 textField2 = new JTextField();  
 textField2.setBounds(120, 50, 250, 20);  
 add(textField2);  
  
 label3 = new JLabel("Отчество:");  
 label3.setBounds(20, 80, 100, 20);  
 add(label3);  
  
 textField3 = new JTextField();  
 textField3.setBounds(120, 80, 250, 20);  
 add(textField3);  
  
 label4 = new JLabel("Дата рождения:");  
 label4.setBounds(20, 110, 100, 20);  
 add(label4);  
  
 textField4 = new JTextField();  
 textField4.setBounds(120, 110, 250, 20);  
 add(textField4);  
  
 label5 = new JLabel("Учебная группа:");  
 label5.setBounds(20, 140, 100, 20);  
 add(label5);  
  
 textField5 = new JTextField();  
 textField5.setBounds(120, 140, 250, 20);  
 add(textField5);  
  
 buttonSave = new JButton("Записать в файл");  
 buttonSave.setBounds(100, 180, 150, 30);  
 buttonSave.addActionListener(new ActionListener() {  
  
 public void actionPerformed(ActionEvent e) {  
 String name = textField1.getText();  
 String surname = textField2.getText();  
 String patronymic = textField3.getText();  
 String birthDate = textField4.getText();  
 String group = textField5.getText();  
  
 try {  
 FileWriter writer = new FileWriter("C:\\Users\\mikha\\OneDrive\\Рабочий стол\\Новый текстовый документ.txt");  
 writer.write(name + "\n");  
 writer.write(surname + "\n");  
 writer.write(patronymic + "\n");  
 writer.write(birthDate + "\n");  
 writer.write(group + "\n");  
 writer.close();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
 });  
 add(buttonSave);  
  
 buttonLoad = new JButton("Загрузить из файла");  
 buttonLoad.setBounds(100, 220, 150, 30);  
 buttonLoad.addActionListener(new ActionListener() {  
  
 public void actionPerformed(ActionEvent e) {  
 try {  
 BufferedReader reader = new BufferedReader(new FileReader("C:\\Users\\mikha\\OneDrive\\Рабочий стол\\Новый текстовый документ.txt"));  
 textField1.setText(reader.readLine());  
 textField2.setText(reader.readLine());  
 textField3.setText(reader.readLine());  
 textField4.setText(reader.readLine());  
 textField5.setText(reader.readLine());  
 reader.close();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
 });  
 add(buttonLoad);  
  
 setVisible(true);  
 }  
  
 public static void main(String[] args) {  
 new Main();  
 }  
  
}

## Проверка результата

|  |
| --- |
|  |

# Задание 3

## Постановка задачи

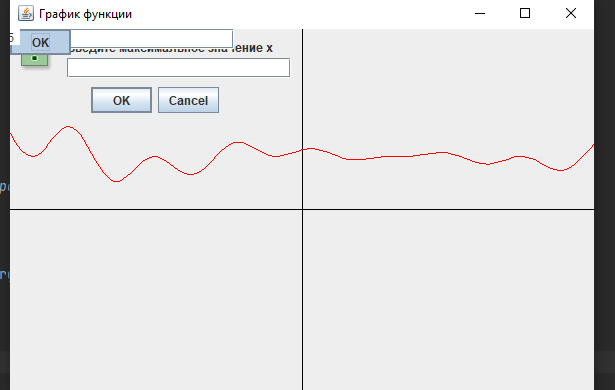
Создать программу, которая запрашивает у пользователя интервал и строит на нем график функции из задания №10 лабораторной работы «Циклы» вашего варианта.

## Решение задачи

Код программы:

import java.awt.Color;  
import java.awt.Graphics;  
import javax.swing.JFrame;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
  
public class Main extends JFrame {  
  
 private static final long *serialVersionUID* = 1L;  
  
 private JPanel panel;  
  
 public Main() {  
 setTitle("График функции");  
 setSize(600, 400);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 panel = new JPanel() {  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 int width = getWidth();  
 int height = getHeight();  
 double xMin = Double.*parseDouble*(JOptionPane.*showInputDialog*("Введите минимальное значение x"));  
 double xMax = Double.*parseDouble*(JOptionPane.*showInputDialog*("Введите максимальное значение x"));  
 double yMin = -20; // минимальное значение y  
 double yMax = 20; // максимальное значение y  
 double step = 0.1; // шаг  
  
 g.setColor(Color.*BLACK*);  
 g.drawLine(0, height / 2, width, height / 2); // ось X  
 g.drawLine(width / 2, 0, width / 2, height); // ось Y  
  
 g.setColor(Color.*RED*);  
 double xPrev = xMin;  
 double yPrev = f(xPrev);  
 for (double x = xMin + step; x <= xMax; x += step) {  
 double y = f(x);  
 int x1 = (int) ((xPrev - xMin) / (xMax - xMin) \* width);  
 int y1 = (int) ((yPrev - yMin) / (yMax - yMin) \* height);  
 int x2 = (int) ((x - xMin) / (xMax - xMin) \* width);  
 int y2 = (int) ((y - yMin) / (yMax - yMin) \* height);  
 g.drawLine(x1, height - y1, x2, height - y2);  
 xPrev = x;  
 yPrev = y;  
 }  
 }  
 };  
 add(panel);  
  
 setVisible(true);  
 }  
  
 private double f(double x) {  
 return x \* Math.*cos*(x) \* Math.*pow*(Math.*sin*(x), 2) + 6;  
 }  
  
 public static void main(String[] args) {  
 new Main();  
 }  
  
}

## Проверка результата



# Задание 4

## Постановка задачи

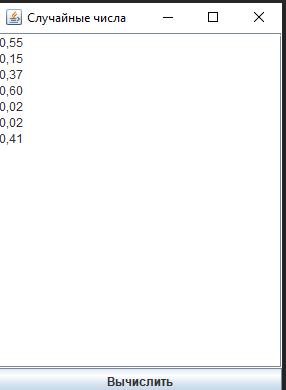
Создать программу, которая при нажатии кнопки вычисляет 7 случайных чисел и отображает их одно под другим в текстовом поле ввода. Поле ввода должно быть неизменяемо для пользователя!

## Решение задачи

Код программы:

import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
  
public class Main extends JFrame implements ActionListener {  
  
 private static final long *serialVersionUID* = 1L;  
  
 private JTextArea textArea;  
 private JButton button;  
  
 public Main() {  
 setTitle("Случайные числа");  
 setSize(300, 400);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 textArea = new JTextArea();  
 textArea.setEditable(false);  
 JScrollPane scrollPane = new JScrollPane(textArea);  
 add(scrollPane);  
  
 button = new JButton("Вычислить");  
 button.addActionListener(this);  
 add(button, "South");  
  
 setVisible(true);  
 }  
   
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == button) {  
 for (int i = 0; i < 7; i++) {  
 double random = Math.*random*();  
 String str = String.*format*("%.2f", random);  
 textArea.append(str + "\n");  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 new Main();  
 }  
  
}

## Проверка результата



# Задание 5

## Постановка задачи

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

## Решение задачи

Код программы:

package com.example.main1;  
  
import javafx.application.Application;  
import javafx.scene.Scene;  
import javafx.scene.chart.NumberAxis;  
import javafx.scene.chart.ScatterChart;  
import javafx.scene.chart.XYChart;  
import javafx.stage.Stage;  
  
public class Main extends Application {  
  
  
 public void start(Stage stage) {  
 stage.setTitle("Фигура попадания точки в область");  
  
 // Ось X  
 final NumberAxis xAxis = new NumberAxis(-10, 10, 1);  
 xAxis.setLabel("X");  
  
 // Ось Y  
 final NumberAxis yAxis = new NumberAxis(-10, 10, 1);  
 yAxis.setLabel("Y");  
  
 // График  
 final ScatterChart<Number, Number> chart = new ScatterChart<>(xAxis, yAxis);  
 chart.setTitle("Фигура попадания точки в область");  
  
 // Добавляем данные на график  
 XYChart.Series series1 = new XYChart.Series();  
 for (double x = -10; x <= 10; x += 0.1) {  
 for (double y = -10; y <= 10; y += 0.1) {  
 if (*fig1*(x, y) || *fig2*(x, y) || *fig3*(x, y) || *fig4*(x, y) || *fig5*(x, y) || *fig6*(x, y) || *fig7*(x, y) || *fig8*(x, y) || *fig9*(x, y) || *fig10*(x, y) || *fig11*(x, y)) {  
 series1.getData().add(new XYChart.Data(x, y));  
 }  
 }  
 }  
  
 // Добавляем данные на график  
 chart.getData().addAll(series1);  
  
 Scene scene = new Scene(chart, 800, 600);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static double k(double x1, double y1, double x2, double y2) {  
 return (y1 - y2) / (x1 - x2);  
 }  
  
 public static double b(double x1, double y1, double x2, double y2) {  
 return y1 - x1 \* *k*(x1, y1, x2, y2);  
 }  
  
 public static boolean fig1(double x, double y) {  
 if (y >= *k*(-6, -4, -2, -6) \* x + *b*(-6, -4, -2, -6) && y <= *k*(-6, -4, -4, 0) \* x + *b*(-6, -4, -4, 0) && y <= *k*(-4, 0, -2, -6) \* x + *b*(-4, 0, -2, -6)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig2(double x, double y) {  
 if (y >= *k*(-6, -4, -4, 0) \* x + *b*(-6, -4, -4, 0) && y <= *k*(-6, -4, -5, 2) \* x + *b*(-6, -4, -5, 2) && y <= *k*(-5, 2, -4, 0) \* x + *b*(-5, 2, -4, 0)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig3(double x, double y) {  
 if (y >= *k*(-4, 0, 0, -1) \* x + *b*(-4, 0, 0, -1) && y >= *k*(0, -1, 1, 4) \* x + *b*(0, -1, 1, 4) && y <= *k*(-4, 0, 1, 4) \* x + *b*(-4, 0, 1, 4)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig4(double x, double y) {  
 if (y >= *k*(-4, 0, 1, 4) \* x + *b*(-4, 0, 1, 4) && y >= *k*(-5, 2, -4, 0) \* x + *b*(-5, 2, -4, 0) && y <= *k*(-5, 2, 1, 4) \* x + *b*(-5, 2, 1, 4)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig5(double x, double y) {  
 if (y >= *k*(-5, 2, 1, 4) \* x + *b*(-5, 2, 1, 4) && y >= *k*(-5, 2, -7, 5) \* x + *b*(-5, 2, -7, 5) && y <= *k*(-7, 5, 1, 4) \* x + *b*(-7, 5, 1, 4)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig6(double x, double y) {  
 if (y >= *k*(-1, -4, 1, -7) \* x + *b*(-1, -4, 1, -7) && y <= *k*(-1, -4, 1, -2) \* x + *b*(-1, -4, 1, -2) && x <= 1) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig7(double x, double y) {  
 if (y >= *k*(1, -7, 3, -4) \* x + *b*(1, -7, 3, -4) && y <= *k*(1, -2, 3, -4) \* x + *b*(1, -2, 3, -4) && x >= 1) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig8(double x, double y) {  
 if (y >= *k*(1, -2, 3, -4) \* x + *b*(1, -2, 3, -4) && y <= *k*(1, -2, 3, -1) \* x + *b*(1, -2, 3, -1) && x <= 3) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig9(double x, double y) {  
 if (y >= *k*(3, -4, 6, -3) \* x + *b*(3, -4, 6, -3) && y <= *k*(3, -1, 6, -3) \* x + *b*(3, -1, 6, -3) && x >= 3) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig10(double x, double y) {  
 if (y >= *k*(3, -1, 6, -3) \* x + *b*(3, -1, 6, -3) && y <= *k*(5, 2, 6, -3) \* x + *b*(5, 2, 6, -3) && y <= *k*(3, -1, 5, 2) \* x + *b*(3, -1, 5, 2)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean fig11(double x, double y) {  
 if (y >= *k*(5, 2, 6, -3) \* x + *b*(5, 2, 6, -3) && y <= *k*(5, 2, 7, 4) \* x + *b*(5, 2, 7, 4) && y >= *k*(7, 4, 6, -3) \* x + *b*(7, 4, 6, -3)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

## Проверка результата

