Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Отчет о лабораторной работе № 1

по дисциплине «Программирование на языке JAVA»

Вариант № 4

Выполнили: ст-ты гр. 19ВВ1

Васильев Л.Р

Проверили:

Юрова О.В.

Карамышева Н.С.

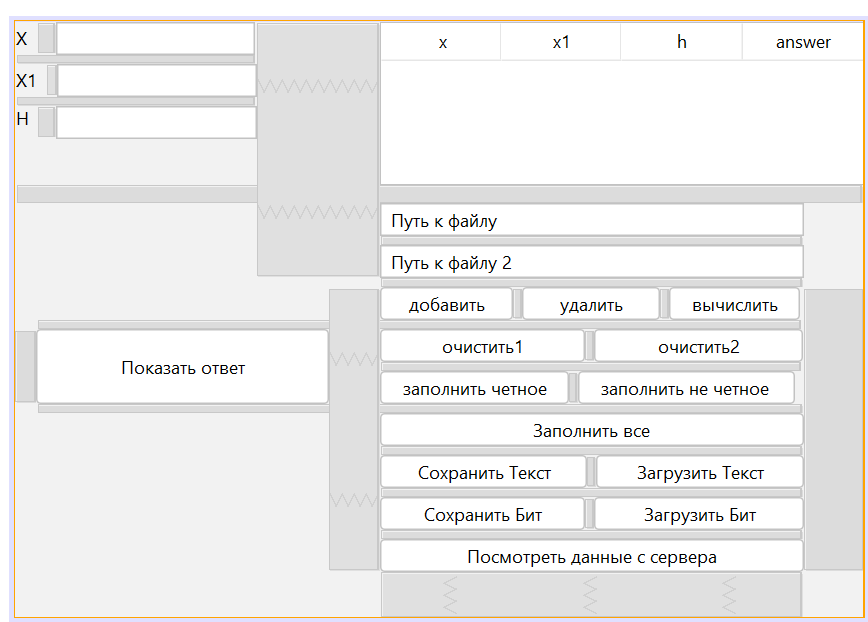
2022

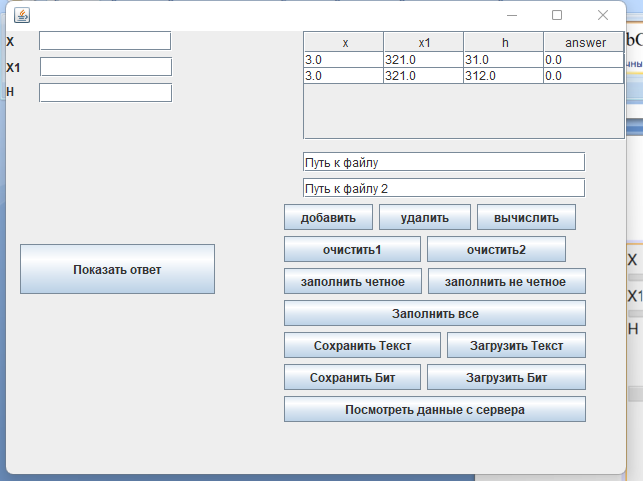
**Цель работы:** научиться разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.

**Задание на лабораторную работу:** вычислить определенный интеграл функции в соответствии с вариантом задания (Приложение 1). Разработать приложение, обладающее графическим интерфейсом с использованием языка Java и библиотеки Swing. Приложение должно содержать 3 поля ввода (JTextField), доступных для редактирования, и соответственно таблицу (JTable) с четырьмя колонками: нижняя граница интегрирования, верхняя граница интегрирования, шаг интегрирования и результат вычисления. Кроме того, должны присутствовать 3 кнопки (JButton): добавить, удалить, вычислить. Для добавления/удаления строки и вычисления значения определенного интеграла для функции в соответствии с вариантом задания (Приложение 1) и параметров выделенной строки таблицы. Результат должен выводиться в четвертой колонке, которая не доступна для редактирования.



**Решение**

****

****

Результат работы программы

**Листинг**

**Lab2.java**

**Добавить**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**DataLab table;**

**if(x\_field.getText().equals("") || x1\_field.getText().equals("") || h\_field.getText().equals("")){**

**JOptionPane.showMessageDialog(this, "Please Enter All Data!");**

**}else {**

**try{**

**float x = Float.parseFloat(x\_field.getText());**

**float x1 = Float.parseFloat(x1\_field.getText());**

**float h = Float.parseFloat(h\_field.getText());**

**DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();**

**int nRow = dtm.getRowCount();**

**table = new DataLab(x, x1, h, 0);**

**if(nRow % 2 == 0){**

**if(table.getX() > 0 && table.getX1()> 0 && table.getH() > 0){**

**rec.addArr(table);**

**dtm.addRow(new Object[] {Float.toString(table.getX()), Float.toString(table.getX1()), Float.toString(table.getH()), null});**

**recByte.addArr(new TableData(x, x1, h, 0));**

**}**

**}else{**

**if(table.getX() > 0 && table.getX1()> 0 && table.getH() > 0){**

**rec.addArr1(table);**

**dtm.addRow(new Object[] {Float.toString(table.getX()), Float.toString(table.getX1()), Float.toString(table.getH()), null});**

**recByte.addArr(new TableData(x, x1, h, 0));**

**}**

**}**

**x\_field.setText("");**

**x1\_field.setText("");**

**h\_field.setText("");**

**}catch(NumberFormatException e1){**

**JOptionPane.showMessageDialog(this, "Введите число !");**

**x\_field.setText("");**

**x1\_field.setText("");**

**h\_field.setText("");**

**}**

**}**

**}**

**Удалить**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();**

**int nRow = dtm.getRowCount();**

**if(nRow > 0){**

**dtm.removeRow(nRow-1);**

**if(rec.arr.size() != 0 && rec.arr1.size() != 0){**

**if((nRow-1) % 2 == 0){**

**rec.arr.remove(rec.arr.size() - 1);**

**}else{**

**rec.arr1.remove(rec.arr1.size() - 1);**

**}**

**}else if(rec.arr1.size() != 0 && rec.arr.size() == 0){**

**rec.arr1.remove(rec.arr1.size() - 1);**

**}else if(rec.arr1.size() == 0 && rec.arr.size() != 0){**

**rec.arr.remove(rec.arr.size() - 1);**

**}**

**}else{**

**JOptionPane.showMessageDialog(this, "Таблица пустая");**

**}**

**}**

**Вычислить**

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {**

**MyThread[] thr = new MyThread[4];**

**DataLab data = null;**

**DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();**

**int nRow = dtm.getRowCount();**

**float[] answer = new float[nRow];**

**float x = 0;**

**float x1 = 0;**

**float h = 0;**

**int n = 0;**

**int n1 = 0;**

**rec.clearArr1();**

**rec.clearArr();**

**for (int i = 0 ; i < nRow ; i++){**

**x = Float.parseFloat((String) dtm.getValueAt(i, 0));**

**x1 = Float.parseFloat((String) dtm.getValueAt(i, 1));**

**h = Float.parseFloat((String) dtm.getValueAt(i, 2));**

**data = new DataLab(x, x1, h, 0);**

**if(i % 5 == 0){**

**try {**

**sendRequest(x, x1, h, i);**

**} catch (IOException ex) {**

**Logger.getLogger(lab2.class.getName()).log(Level.SEVERE, null, ex);**

**}**

**}else if(i % 5 == 1){**

**if(i % 2 == 0){**

**thr[0] = new MyThread(x, x1, h, i,rec);**

**thr[0].start();**

**}else{**

**thr[0] = new MyThread(x, x1, h, i,rec);**

**thr[0].start();**

**}**

**}else if(i % 5 == 2){**

**if(i % 2 == 0){**

**thr[1] = new MyThread(x, x1, h, i,rec);**

**thr[1].start();**

**}else{**

**thr[1] = new MyThread(x, x1, h, i,rec);**

**thr[1].start();**

**}**

**}else if(i % 5 == 3){**

**if(i % 2 == 0){**

**thr[2] = new MyThread(x, x1, h, i,rec);**

**thr[2].start();**

**}else{**

**thr[2] = new MyThread(x, x1, h, i,rec);**

**thr[2].start();**

**}**

**}else if(i % 5 == 4){**

**if(i % 2 == 0){**

**thr[3] = new MyThread(x, x1, h, i,rec);**

**thr[3].start();**

**}else{**

**thr[3] = new MyThread(x, x1, h, i,rec);**

**thr[3].start();**

**}**

**}**

**}**

**}**

**Вычисление**

**public float getAnswer(){**

**float a = 0;**

**float b = 0;**

**float answer = 0;**

**float heightCount = 0;**

**heightCount = (this.x1 - this.x)/this.h;**

**a = x;**

**b = x + h;**

**for(int j = 0; j<(int)heightCount;j++){**

**answer = (float) (answer + (Math.tan(a/2) + Math.tan(b/2)) \* h);**

**a = a + h;**

**b = b + h;**

**}**

**return answer;**

**}**

**Вывод:** научились разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing. Выявили, что размер шага влияет на точность результата, а именно: чем меньше шаг, тем точнее результат