Министерство науки и высшего образования Российской федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Программирование на языке Java»

на тему «Графические интерфейсы»»

Вариант 5

Выполнили

студенты группы 21ВВП2:

Лукина Е.Ю.

Ванюшин И.А.

Тарасов Н.А.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2024

**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. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

|  |  |
| --- | --- |
| Номер варианта | Функция |
| 5 |  |

**Ход работы**

Написали класс для вычисления определенного интеграла методом трапеций.

Ниже представлен метод класса IntegralCalculator, возвращающий значение определенного интеграла.

public double F()

{

double result = 0;

double lBound = a, rBound = a + e;

while (rBound < b)

{

result += e \* (Math.exp(-lBound) + Math.exp(-rBound)) / 2;

lBound += e;

rBound += e;

}

if (lBound < b) result += e \* (Math.exp(-lBound) + Math.exp(-b)) / 2;

return result;

}

Сделали графическое оформление для приложения, позволяющее сохранять результаты вычислений в таблице, очищать таблицу, а также задавать пользовательские параметры пределов вычислений и шага интегрирования.

**Листинг**

IntegralCalculator.java

package lab1;

/\*\*

\*

\* @author User

\*/

public class IntegralCalculator {

public IntegralCalculator()

{

a = 0;

b = 1;

e = 0.1e+6;

}

public IntegralCalculator(double \_a, double \_b, double \_e)

{

a = \_a;

b = \_b;

if (b < a)

{

double tmp = a;

a = b;

b = tmp;

}

e = \_e;

}

private double e, a, b;

public double E() { return e; }

public double A() { return a; }

public double B() { return b; }

public void SetE(double e) {

if (e == 0) return;

this.e = Math.abs(e);

}

public void SetArea(double \_a, double \_b) {

a = \_a; b = \_b;

if (b < a)

{

double tmp = a;

a = b;

b = tmp;

}

}

public double F()

{

double result = 0;

double lBound = a, rBound = a + e;

while (rBound < b)

{

result += e \* (Math.exp(-lBound) + Math.exp(-rBound)) / 2;

lBound += e;

rBound += e;

}

if (lBound < b) result += e \* (Math.exp(-lBound) + Math.exp(-b)) / 2;

return result;

}

}

MyWin.java

package lab1;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

public class MyWin extends JFrame implements ActionListener

{

// Some window's elements

private final JTextField jtfA;

private final JTextField jtfB;

private final JTextField jtfE;

private final DefaultTableModel dtResultsModel;

private final JTable jtResults;

// Класс для вычисления определенного интеграла

private final IntegralCalculator ic;

// Classes serial number

private static final long serialVersionUID = 1L;

public MyWin()

{

// Initialization block

ic = new IntegralCalculator(0, 1, 1e-6);

jtfA = new JTextField();

jtfB = new JTextField();

jtfE = new JTextField();

//----------------==Table Filling==------------------------------

Object[] columnNames = new String[]{"L limit", "R limit", "Step", "Result"};

Object[][] data = new String[][]{};

dtResultsModel = new DefaultTableModel(data, columnNames);

dtResultsModel.setColumnIdentifiers(columnNames);

jtResults = new JTable(dtResultsModel);

JScrollPane sp = new JScrollPane(jtResults);

Container c = getContentPane(); // Client rect

c.setLayout(null); // НЕНАВИЖУ КОМПАНОВЩИКИ

//----------------==Left Panel's Filling==-----------------------

// Creates a left sided panel

JPanel jpLeftPanel = new JPanel();

jpLeftPanel.setLayout(null); // АГРЕССИЯ И ЗУБЫ СКРИПЯТ

jpLeftPanel.setBounds(0, 0, 200, 480); // Resizes left panel

// Creates buttons.

// The one that Adds new records & the other that Clears all inputs.

JButton jbtnAdd = new JButton("Add");

JButton jbtnClear = new JButton("Clear");

// Buttons's resize.

jbtnAdd.setBounds(10, 310, 180, 50);

jbtnClear.setBounds(10, 370, 180, 50);

// Adds action listeners for the objects.

jbtnAdd.addActionListener(this);

jbtnClear.addActionListener(this);

jtfA.setBounds(10, 10, 180, 50);

jtfB.setBounds(10, 70, 180, 50);

jtfE.setBounds(10, 130, 180, 50);

jtfA.setText(Double.toString(ic.A()));

jtfB.setText(Double.toString(ic.B()));

jtfE.setText(Double.toString(ic.E()));

sp.setBounds(200, 0, 427, 480);

// Fills panel with early created buttons & txtFields.

jpLeftPanel.add(jtfA);

jpLeftPanel.add(jtfB);

jpLeftPanel.add(jtfE);

jpLeftPanel.add(jbtnAdd);

jpLeftPanel.add(jbtnClear);

//----------------==Client rect's Filling==-----------------------

// Adds left panel to actually left side of client rect.

c.add(jpLeftPanel);

c.add(sp);

//----------------==Setting window's properties==-----------------

// Window's properties.

setTitle("Integral Calculator"); // Window's title. Obviously.

// Sets preffered size to window.

setPreferredSize(new Dimension(640, 480));

// Exit application on Exit button.

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

pack(); // Setting preffered sizes.

setVisible(true); // Makes window visible.

}

@Override

public void actionPerformed(ActionEvent e) {

// "Calculate" case. Calculates the integral with setted parameters.

if (e.getActionCommand().equals("Add"))

{

ic.SetArea(Double.parseDouble(jtfA.getText()), Double.parseDouble(jtfB.getText()));

ic.SetE(Double.parseDouble(jtfE.getText()));

jtfE.setText(Double.toString(ic.E()));

dtResultsModel.addRow(new String[]{String.valueOf(ic.A()), String.valueOf(ic.B()), String.valueOf(ic.E()), String.valueOf(ic.F())});

}

else if (e.getActionCommand().equals("Clear"))

{

dtResultsModel.setRowCount(0);

}

else throw new UnsupportedOperationException("Not supported yet."); // Unprocessed action case.

}

// запуск оконного приложения

public static void main(String args[]) {

new MyWin();

}

}

**Результат работы программы**

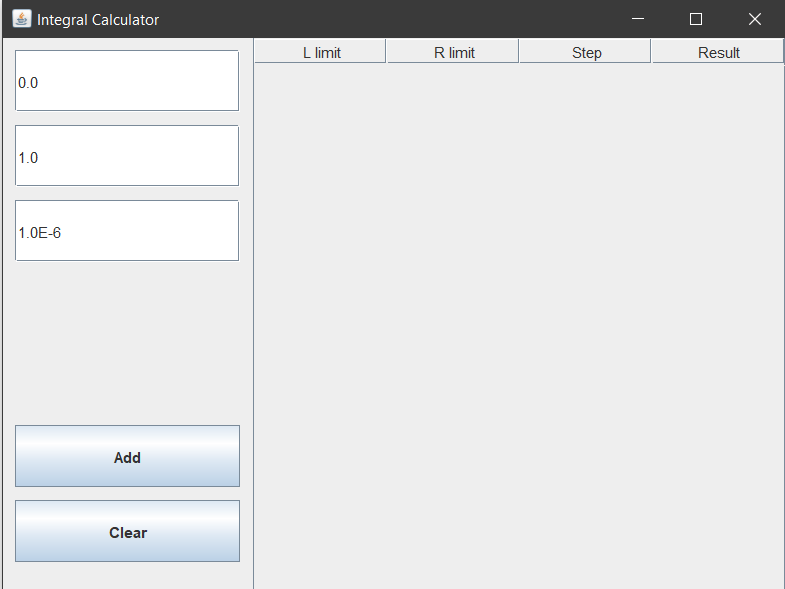
Результаты работы программы представлены на рисунках 1-3. 

Рис.1 – Состояние программы после запуска

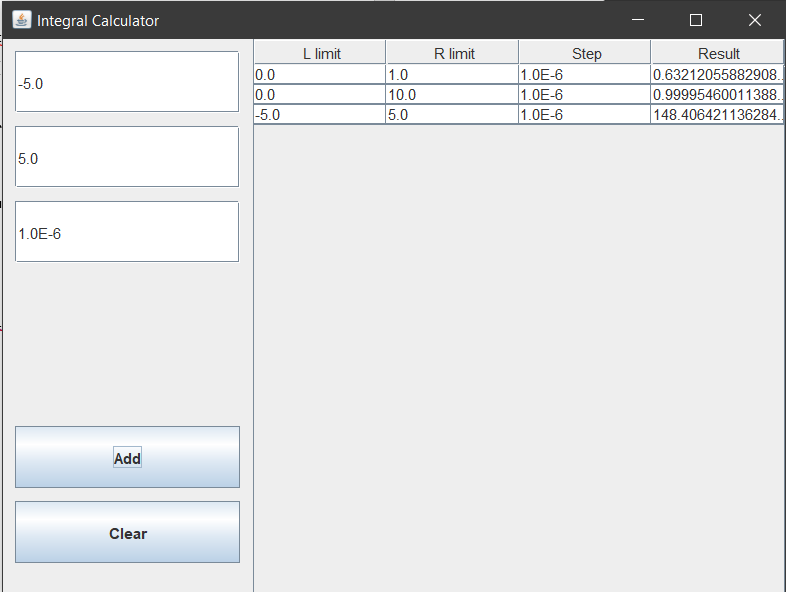


Рис.2 – Состояние программы после добавления нескольких записей в таблицу

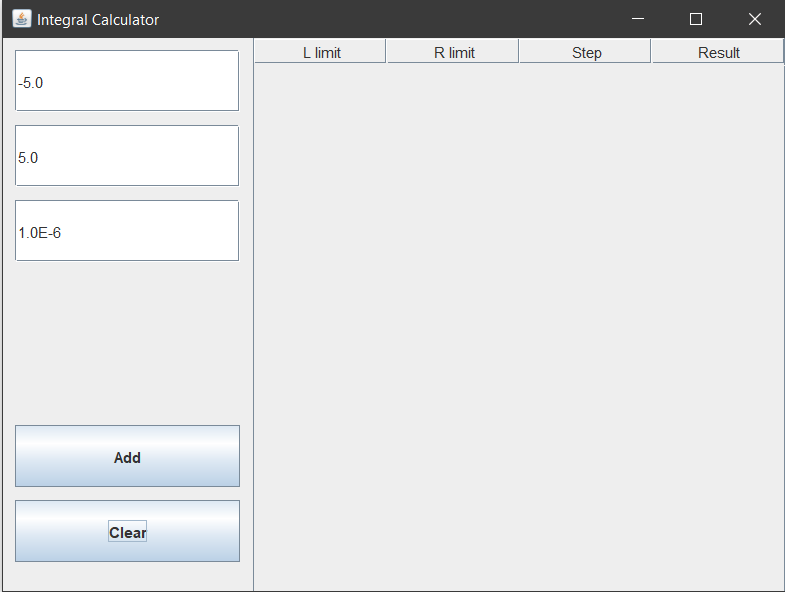
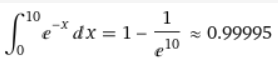
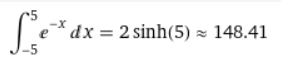


Рис.3 – Состояние программы после очистки таблицы

**Математический расчет**

****

****



Результат программы совпал с ожидаемым, следовательно, программа составлена верно.

**Вывод**

В результате выполнения методических указаний к лабораторной работе были получены навыки работы с графической оболочкой Java swing.