Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Графические интерфейсы

Отчет о лабораторной работе №1

по дисциплине «Разработка кроссплатформенных приложений»

Выполнили: ст-ты гр. 21ВОЭ1

Кокорин А. А.

Тишаев М. Р.

Приняли: к.т.н., доцент каф. ВТ

Юрова О. В.

к.т.н., доцент каф. ВТ

Карамышева Н. С.

2024

1 Цель работы

Научиться разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.

2 Формулировка задания

Вычислить определенный интеграл функции. 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 sin(*x*) 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.

3 Графическое описание алгоритма

Алгоритм вычисления интеграла в виде схемы программы представлен на рисунке 1. Алгоритм начинается с ввода *row* – количество строк в таблице (блок 2). Затем происходит проверка условия *row* = 0 (блок 3). Если условие истинно, то программа завершает свою работу. В противном случае осуществляется ввод: *a* –нижняя граница интегрирования, *b* – нижняя граница интегрирования, *h* – шаг интегрирования (блок 4). Далее происходит проверка условия *a* < *b* и ост((*a* - *b*) / *h*) = 0 (блок 5).

Если условие истинно, то с помощью цикла происходит вычисление определенного интеграла (блоки 6 - 8) и вывод результата вычисления (блок 9). В противном случае программа завершает свою работу.

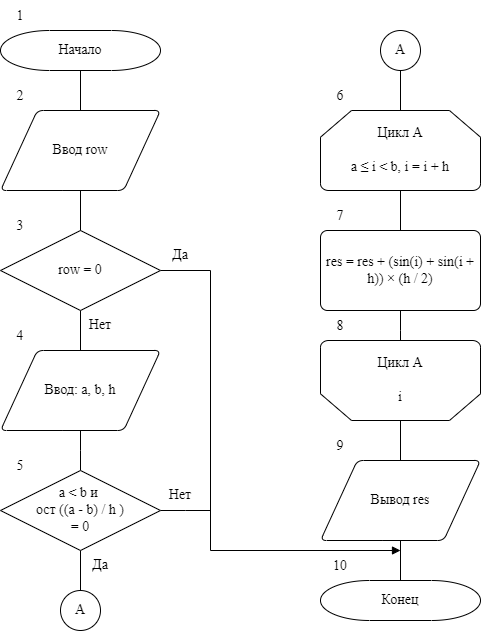


Рисунок 1 – Схема программы для вычисления интеграла

4 Описание программы

Текст программы приведен в приложении А.

На Форме расположены следующие объекты:

три однострочных редактора: *jTextFieldLow* (для ввода нижней границы интегрирования), *jTextFieldTop* (для ввода верхней границы интегрирования), *jTextFieldStep* (для ввода шага интегрирования);

три объекта метки: *jLabel1*, *jLabel2*, *jLabel3*. В свойстве *text* этих объектов помещены пояснения для пользователя программы:

*jLabel1.text* — «Нижняя граница»;

*jLabel2.text* — «Верхняя граница»;

*jLabel3.text* — «Шаг»;

одна таблица — *jTable1* (для ввода исходных данных);

кнопки *jButtonAdd* для создания обработчика события с целью добавления строки с введенными данными в таблицу, *jButtonDel* для создания обработчика события с целью удаления строки из таблицы, *jButtonFind* для создания обработчика события с целью вычисления интеграла и его вывода в таблицу. В свойстве *text* этих объектов помещены пояснения для пользователя программы:

*ButtonAdd.text* — «Добавить»;

*ButtonDel.text* — «Удалить»;

*ButtonFind.text* — «Вычислить»;

Программа состоит из трех обработчиков событий: обработчик события клика на кнопку формы *jButtonAdd* — *jButtonAddActionPerformed*, обработчик события клика на кнопку формы *jButtonDel* — *jButtonDelActionPerformed*, обработчик события клика на кнопку формы *jButtonFind* — *jButtonFindActionPerformed.* В методе *jButtonFindActionPerformed* описаны:

переменная целого типа *integer:*

*row* – количество строк в таблице.

переменные вещественного типа *double:*

*a* – нижняя граница интегрирования;

*b* – верхняя граница интегрирования;

*h* – шаг интегрирования;

*res* – результат вычисления интеграла.

5 Программа и методика испытаний

Для проверки правильности работы программы подготовлен вручную тестовый набор данных в таблице 1.

Таблица 1 – Тестовый набор данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | h | Результат работы программы | Результат ручного просчета |
| 2 | 4 | 0.2 | 0,1047 | 0,1047 |
| 0 | 3 | 0.3 | 0,0950 | 0,0950 |
| 100 | 100000 | 0.01 | -19,9084 | -19,9084 |
| 24 | 200 | 0.02 | 106,1827 | 106,1827 |

При работе программы с тестовыми данными получены результаты, приведенные в Приложении Б. Результат, приведенный в таблице 1, совпадают с результатом работы программы на рисунке Б.2, Б.3, Б.4, Б.5. Таким образом, можно сделать вывод, что программа работает правильно.

Вывод

Результатом выполнения данной лабораторной работы является получение навыков разработки приложений, обладающих графическим интерфейсом пользователя, с использованием библиотеки Swing и написание программы, обладающей графическим интерфейсом, для вычисления определенного интеграла.

ТЕКСТ ПРОГРАММЫ

Приложение А

(обязательное)

package my.javaapplication1;

import javax.swing.table.DefaultTableModel;

public class NewJFrame extends javax.swing.JFrame {

public NewJFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButtonAdd = new javax.swing.JButton();

jButtonDel = new javax.swing.JButton();

jButtonFind = new javax.swing.JButton();

jTextFieldLow = new javax.swing.JTextField();

jTextFieldTop = new javax.swing.JTextField();

jTextFieldStep = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButtonAdd.setText("Добавить");

jButtonAdd.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonAddActionPerformed(evt);

}

});

jButtonDel.setText("Удалить");

jButtonDel.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonDelActionPerformed(evt);

}

});

jButtonFind.setText("Вычислить");

jButtonFind.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonFindActionPerformed(evt);

}

});

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Нижняя граница", "Верхняя граница", "Шаг", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.getTableHeader().setReorderingAllowed(false);

jScrollPane1.setViewportView(jTable1);

if (jTable1.getColumnModel().getColumnCount() > 0) {

jTable1.getColumnModel().getColumn(0).setResizable(false);

jTable1.getColumnModel().getColumn(1).setResizable(false);

jTable1.getColumnModel().getColumn(2).setResizable(false);

jTable1.getColumnModel().getColumn(3).setResizable(false);

}

jLabel1.setText("Нижняя граница");

jLabel2.setText("Верхняя граница");

jLabel3.setText("Шаг");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButtonFind, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButtonAdd, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButtonDel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 94, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addGap(0, 0, Short.MAX\_VALUE))

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jTextFieldStep)

.addComponent(jTextFieldTop)

.addComponent(jTextFieldLow))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 551, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonAdd)

.addComponent(jTextFieldLow, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel2)

.addGap(6, 6, 6)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonDel)

.addComponent(jTextFieldTop, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel3)

.addGap(3, 3, 3)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonFind)

.addComponent(jTextFieldStep, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 176, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

pack();

}// </editor-fold>

private void jButtonAddActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

TModel.addRow(new Object[] {jTextFieldLow.getText(), jTextFieldTop.getText(), jTextFieldStep.getText()});

}

private void jButtonDelActionPerformed(java.awt.event.ActionEvent evt) {

if (jTable1.getSelectedRow() != -1) {

DefaultTableModel TModel = (DefaultTableModel) jTable1.getModel();

TModel.removeRow(jTable1.getSelectedRow());

}

}

private void jButtonFindActionPerformed(java.awt.event.ActionEvent evt) {

int row = jTable1.getSelectedRow();

if (row != -1) {

double h, a, b, res = 0;

a = Double.parseDouble(jTable1.getValueAt(row, 0).toString());

b = Double.parseDouble(jTable1.getValueAt(row, 1).toString());

h = Double.parseDouble(jTable1.getValueAt(row, 2).toString());

if ((a < b) && ((b - a) % h < 0.0001 || (b - a) % h > h - 0.0001)) {

for (double i = a; i < b; i += h)

res += ((Math.sin(i \* Math.PI / 180) + Math.sin((i + h) \* Math.PI / 180)) \* (h / 2));

jTable1.setValueAt(String.format("%.4f",res), row, 3);

}

}

}

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new NewJFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButtonAdd;

private javax.swing.JButton jButtonDel;

private javax.swing.JButton jButtonFind;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextFieldLow;

private javax.swing.JTextField jTextFieldStep;

private javax.swing.JTextField jTextFieldTop;

// End of variables declaration

}

РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Приложение Б

(обязательное)

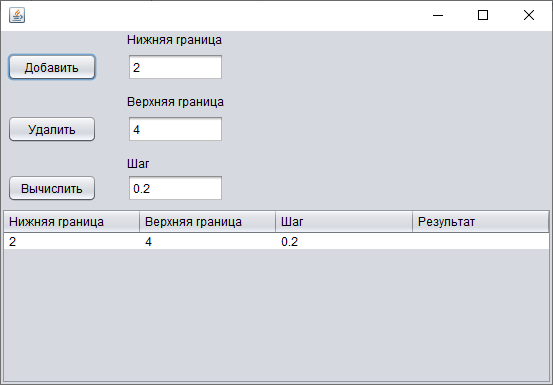


Рисунок Б.1 – Добавление строки с исходными данными в таблицу

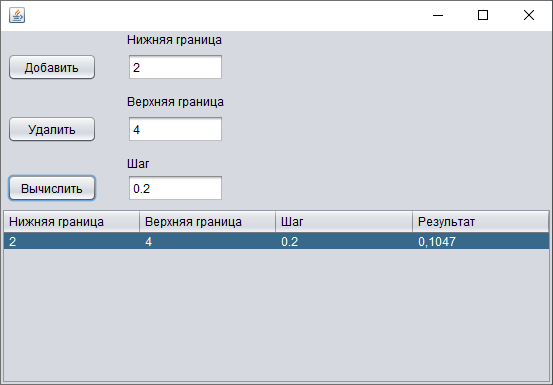


Рисунок Б.2 – Вычисление и вывод результата

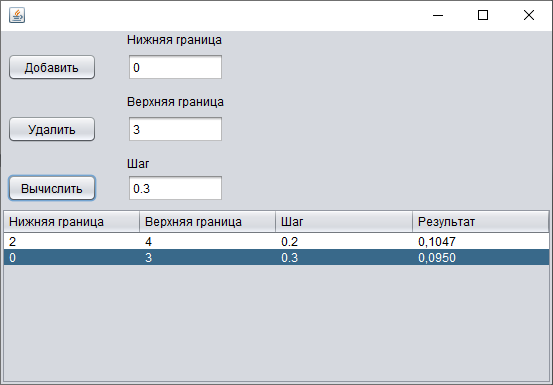


Рисунок Б.3 – Вычисление и вывод результата

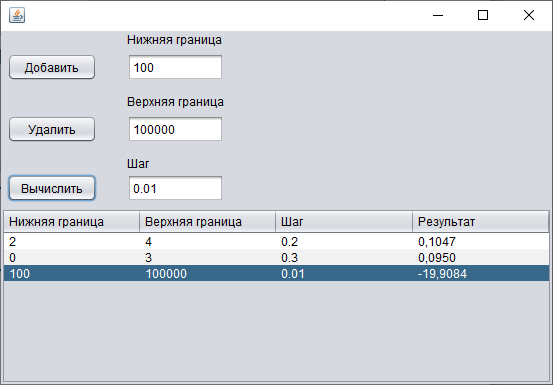


Рисунок Б.4 – Вычисление и вывод результата

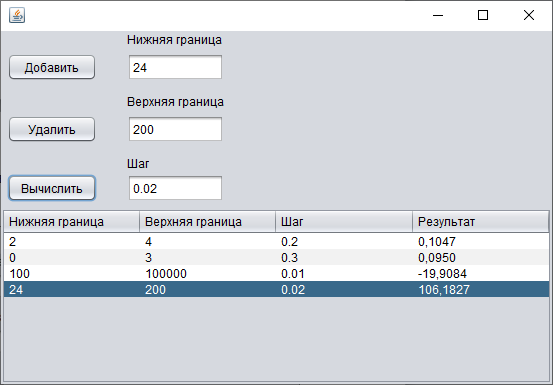


Рисунок Б.5 – Вычисление и вывод результата

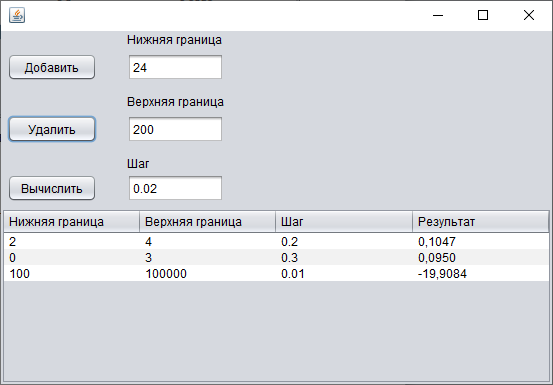


Рисунок Б.6 – Удаление строки из таблицы