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

Куимов Н.А.

Комраков А.М.

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. Затем происходит проверка условия *stroka !=-1* (блок 3). Если условие ложно, то программа завершает свою работу. В противном случае осуществляется ввод: *nzg* –нижняя граница интегрирования, *vzg* – верхняя граница интегрирования, *shg* – шаг интегрирования (блок 4). Далее происходит цикл res += (((1 / i) + (1 / (i + shg)))) \* (shg / 2) (блок 5).

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

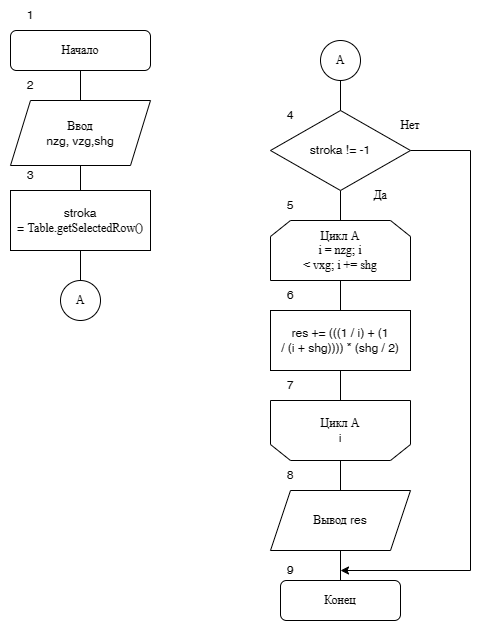


Рисунок 1 – Схема программы для вычисления интеграла

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:*

*stroka*– количество строк в таблице.

переменные вещественного типа *double:*

*nzg* – нижняя граница интегрирования;

*vzg* – верхняя граница интегрирования;

*shg* – 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 |
| 5 | 15 | 1 | 1,101562… | 1,101562… |
| 6 | 32 | 1 | 1,676235… | 1,676235… |
| 6 | 12 | 4 | 0 | 0 |
| 10 | 20 | 4 | 0 | 0 |

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. 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)

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package my.numberaddition;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author student

\*/

public class ContactEditorUI extends javax.swing.JFrame {

/\*\*

\* Creates new form ContactEditorUI

\*/

public ContactEditorUI() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

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

private void initComponents() {

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

Table = new javax.swing.JTable();

jButton\_Add = new javax.swing.JButton();

jButton\_Del = new javax.swing.JButton();

jButton\_Res = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButtonClearData = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

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

jTextField2.addActionListener(new java.awt.event.ActionListener() {

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

jTextField2ActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"Нижняя граница интегрирования ", "Верхняя граница интегрирования", "Шаг интегрирования", "Результат вычисления"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(Table);

jButton\_Add.setText("Добавить");

jButton\_Add.addActionListener(new java.awt.event.ActionListener() {

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

jButton\_AddActionPerformed(evt);

}

});

jButton\_Del.setText("Удалить");

jButton\_Del.addActionListener(new java.awt.event.ActionListener() {

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

jButton\_DelActionPerformed(evt);

}

});

jButton\_Res.setText("Вычислить");

jButton\_Res.addActionListener(new java.awt.event.ActionListener() {

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

jButton\_ResActionPerformed(evt);

}

});

jLabel1.setText("Верхняя граница");

jLabel2.setText("Шаг интегрирования");

jLabel3.setText("Нижнняя граница");

jButtonClearData.setText("Очистить");

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

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

jButtonClearDataActionPerformed(evt);

}

});

jButton2.setText("Заполнить");

jButton2.addActionListener(new java.awt.event.ActionListener() {

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

jButton2ActionPerformed(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(319, 319, 319)

.addComponent(jButton\_Add, javax.swing.GroupLayout.PREFERRED\_SIZE, 93, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(40, 40, 40)

.addComponent(jButton\_Del, javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(36, 36, 36)

.addComponent(jButton\_Res)

.addGap(32, 32, 32)

.addComponent(jButtonClearData, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 107, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(88, 88, 88)

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

.addComponent(jLabel2)

.addGroup(layout.createSequentialGroup()

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

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

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

.addComponent(jTextField2)

.addComponent(jTextField1)

.addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT\_SIZE, 72, Short.MAX\_VALUE))

.addComponent(jLabel1))

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(47, 47, 47)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 892, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addGap(0, 111, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(125, 125, 125)

.addComponent(jLabel2)

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

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 141, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1)

.addGap(1, 1, 1)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel3)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(43, 43, 43)

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

.addComponent(jButton\_Add, javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton\_Del, javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton\_Res, javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButtonClearData, javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(332, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton\_AddActionPerformed(java.awt.event.ActionEvent evt) {

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

TModel.addRow (new Object[] {jTextField3.getText(), jTextField2.getText(), jTextField1.getText()});

}

private void jButton\_DelActionPerformed(java.awt.event.ActionEvent evt) {

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

if (Table.getSelectedRow()!=-1)

TModel.removeRow(Table.getSelectedRow());

}

private void jButton\_ResActionPerformed(java.awt.event.ActionEvent evt) {

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

double shg, vxg, nzg, res = 0;

int stroka;

stroka = Table.getSelectedRow();

if (stroka != -1) {

nzg = Double.parseDouble(Table.getValueAt(stroka, 0).toString());

vxg = Double.parseDouble(Table.getValueAt(stroka, 1).toString());

shg = Double.parseDouble(Table.getValueAt(stroka, 2).toString());

for (double i = nzg; i < vxg; i += shg)

{

res += (((1 / i) + (1 / (i + shg)))) \* (shg / 2);

}

}

Table.setValueAt(res, stroka, 3);

}

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

// TODO add your handling code here:

DefaultTableModel model = (DefaultTableModel)Table.getModel();

model.setRowCount(0);

}

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

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

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(ContactEditorUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new ContactEditorUI().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JTable Table;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButtonClearData;

private javax.swing.JButton jButton\_Add;

private javax.swing.JButton jButton\_Del;

private javax.swing.JButton jButton\_Res;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

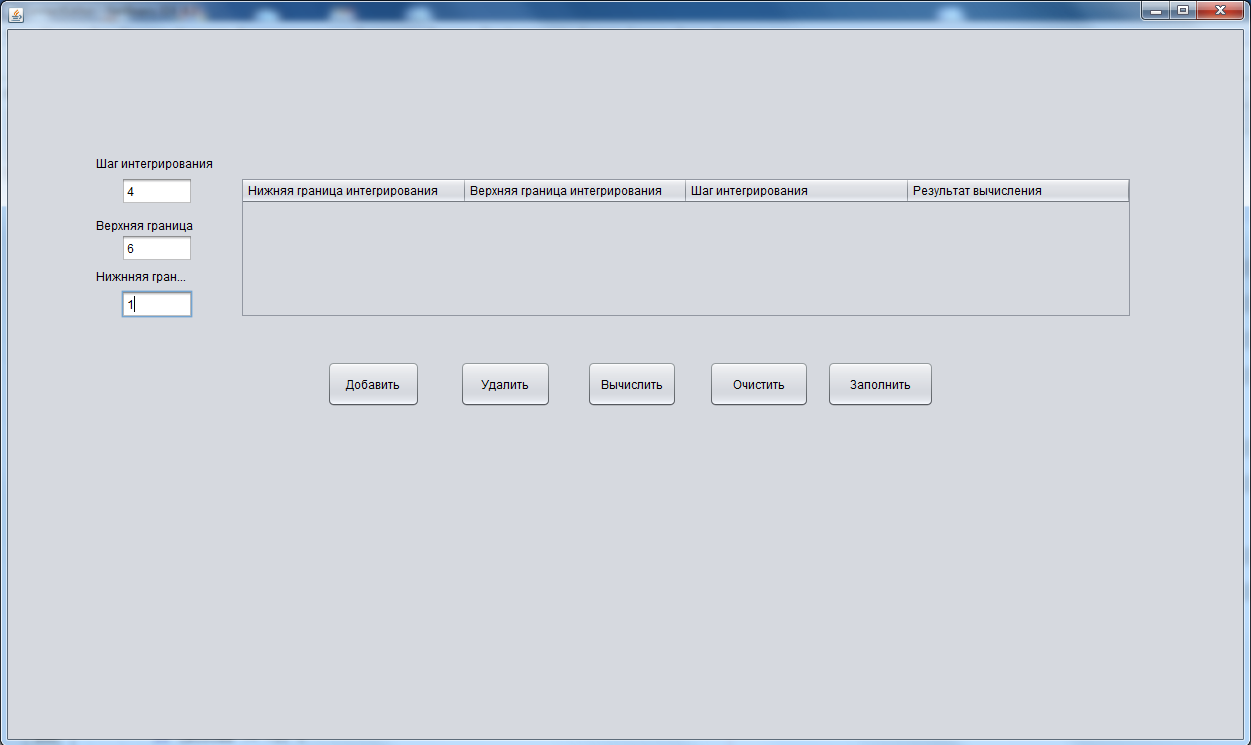
// End of variables declaration

}

РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Приложение Б

(обязательное)

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

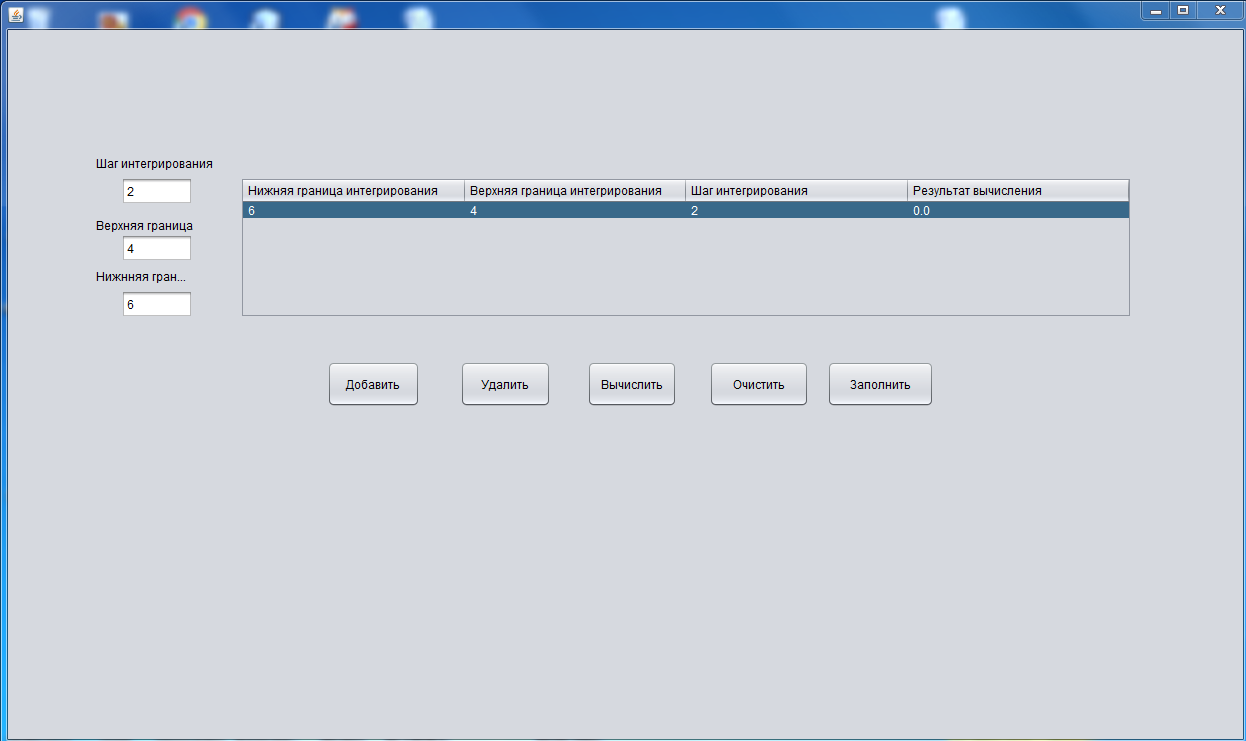


Рисунок Б.2 – Вычисление и вывод результата

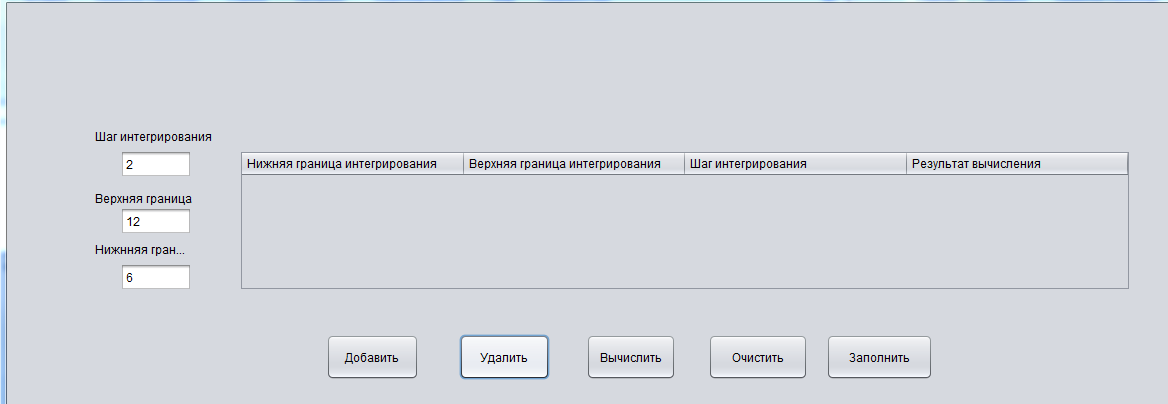


Рисунок Б.3 – Удаление строки из таблицы