Министерство образования Российской Федерации

Пензенский государственный университет

**Кафедра «Вычислительная техника»**

**ОТЧЁТ**

По лабораторной работе №1

По дисциплине:

«Программирование на языке JAVA»

## На тему:

## «Графические интерфейсы»

Выполнили:

студенты группы 20ВВП1

Верховский М. В.

Лукин В. Д.

Приняли:

к.т.н., доцент, Юрова О.В.

к.т.н., доцент, Карамышева Н.С.

**Пенза 2023**

**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 №4. 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 №4 и параметров выделенной строки таблицы.

Результат должен выводиться в четвертой колонке, которая не доступна для редактирования.

**Ход выполнения лабораторной работы:**

**Листинг**

import javax.swing.\***;**import javax.swing.table.AbstractTableModel**;**import javax.swing.table.DefaultTableModel**;**import java.awt.event.ActionEvent**;**import java.awt.event.ActionListener**;**import java.util.Vector**;**public class Table {  
  
 private JButton AddButton**;** private JButton DelButton**;** private JButton CalcButton**;** private JTextField Down**;** private JTextField Up**;** private JTextField Step**;** private JTable MyTable**;** private DefaultTableModel MyModel**;** private JPanel MyPanel**;** public Table(){  
 //Убираем возможность редактирования столбца с результатом  
 MyTable.setModel(new DefaultTableModel() {  
  
 @Override  
 public boolean isCellEditable(int rowIndex**,** int columnIndex) { // Возвращает true, если ячейка в rowIndex и columnIndex доступна для редактирования  
 return columnIndex !=**3;** }  
 })**;** // Чтобы добавить и удалить строки из таблицы, нужно использовать DefaultTableModel  
 MyModel = (DefaultTableModel) MyTable.getModel()**;** // создание столбцов  
 MyModel.addColumn("Верх")**;** MyModel.addColumn("Низ")**;** MyModel.addColumn("Шаг")**;** MyModel.addColumn("Результат")**;** AddButton.addActionListener(new ActionListener() { // ActionListener хранит в себе метод, который активируется при нажатии кнопки  
 @Override  
 public void actionPerformed(ActionEvent e) { // actionPerformed - метод, который активируется при нажатии кнопки  
 String Up = Table.this.Up.getText()**;** // getText возвращает текст, содержащийся в текстовом поле Up  
 String Down = Table.this.Down.getText()**;** String Step = Table.this.Step.getText()**;** MyModel.addRow(new String[]{Up**,** Down**,** Step})**;** // addRow добавляет строку  
  
 }  
 })**;** CalcButton.addActionListener(new ActionListener() { // ActionListener хранит в себе метод, который активируется при нажатии кнопки  
 @Override  
 public void actionPerformed(ActionEvent e) { // actionPerformed - метод, который активируется при нажатии кнопки  
 int indexRow = MyTable.getSelectedRow()**;** // Метод getSelectedRow() возвращает индекс первой выбранной строки  
  
 if (indexRow == -**1**) {  
 return**;** }  
  
 double area = **0;** // площадь = интеграл  
 Vector data = MyModel.getDataVector().get(indexRow)**;**// в data записываем значения выбранной строки (значения могут меняться)  
 double a = Double.*parseDouble*((String) data.get(**0**))**;**//получаем значение из 0 столбца выбранной строки в формате double  
 double b = Double.*parseDouble*((String) data.get(**1**))**;**//parseDouble возвращает новое double-инициализированное значение, представленное указанным String  
 double h = Double.*parseDouble*((String) data.get(**2**))**;** for (int i = **0;** i < (b - a) / h**;** i++) {  
 if (a+(i+**1**)\*h<=b){  
 area += h \* (**0.5** \* (Math.*tan*(a+i\*h) + Math.*tan*(a+(i+**1**)\*h)))**;** }  
 else {  
 area += h \* (**0.5** \* (Math.*tan*(a+i\*h) + Math.*tan*(b)))**;** }  
  
 }  
  
 MyModel.setValueAt(area**,** indexRow**, 3**)**;** // установка значения интеграла в строку с индексом в столбце №3  
  
 }  
 })**;** DelButton.addActionListener(new ActionListener() { // ActionListener хранит в себе метод, который активируется при нажатии кнопки  
 @Override  
 public void actionPerformed(ActionEvent e) { // actionPerformed - метод, который активируется при нажатии кнопки  
 int indexRow = MyTable.getSelectedRow()**;** // Метод getSelectedRow() возвращает индекс первой выбранной строки  
 if (indexRow != -**1**){  
 MyModel.removeRow(indexRow)**;** // Метод removeRow() имеет параметр ind - индекс строки, которая будет удалена  
 }  
 }  
 })**;** }  
  
  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Интегральная гипербола кибернетики")**;** // создаём окно с рамкой  
 frame.setVisible(true)**;** // показывает или скрывает это окно в зависимости от значения параметра  
 frame.setContentPane(new Table().MyPanel)**;**//задаёт содержимое окна с рамкой  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*)**;**//Устанавливает операцию при закрытии окна закончить работу приложения,  
 // которая будет выполняться по умолчанию, когда пользователь инициирует «закрытие» этого окна  
 frame.setSize(**450,600**)**;** // размер окна  
 frame.setLocation(**120,100**)**;** // расположение окна  
  
  
  
 }  
  
  
}

**Результаты выполнения программы**

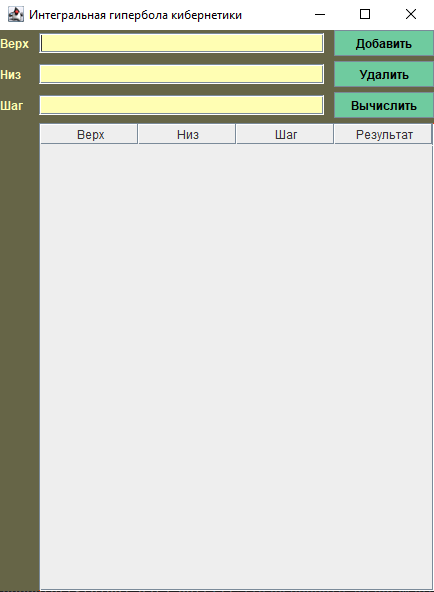


Рисунок №1 - Запуск программы

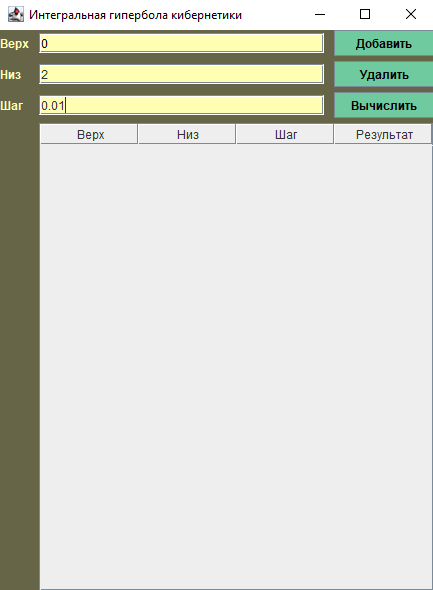


Рисунок №2 - Ввод данных для интеграла

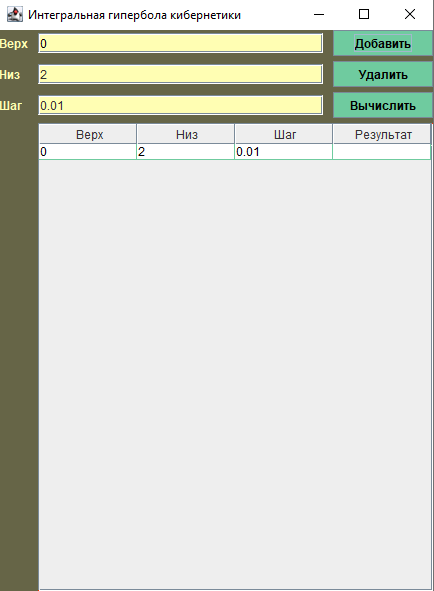


Рисунок №3 - Добавление строки с данными в таблицу

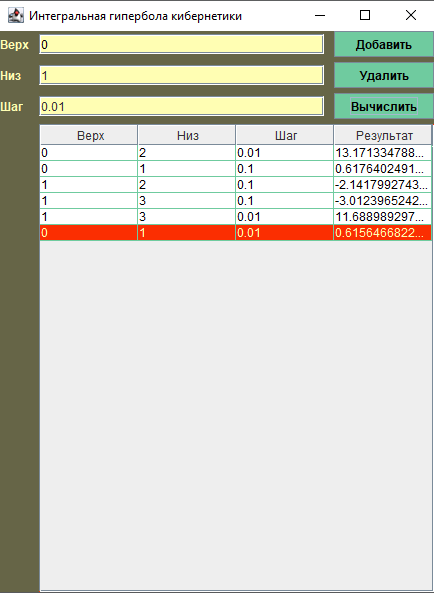


Рисунок №4 - Вычисление определённого интеграла

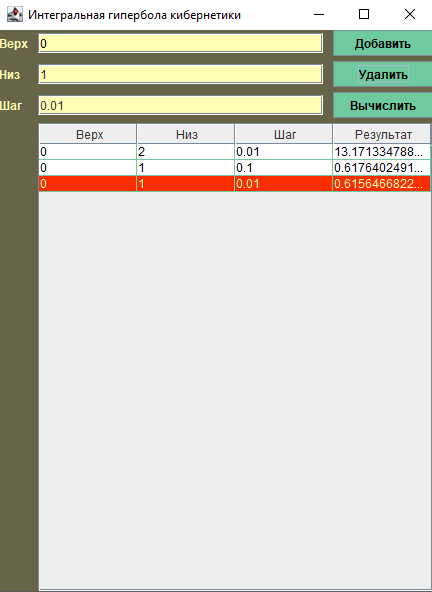


Рисунок №5 - Удаление строк с данными из таблицы

**Проверка работы программы**

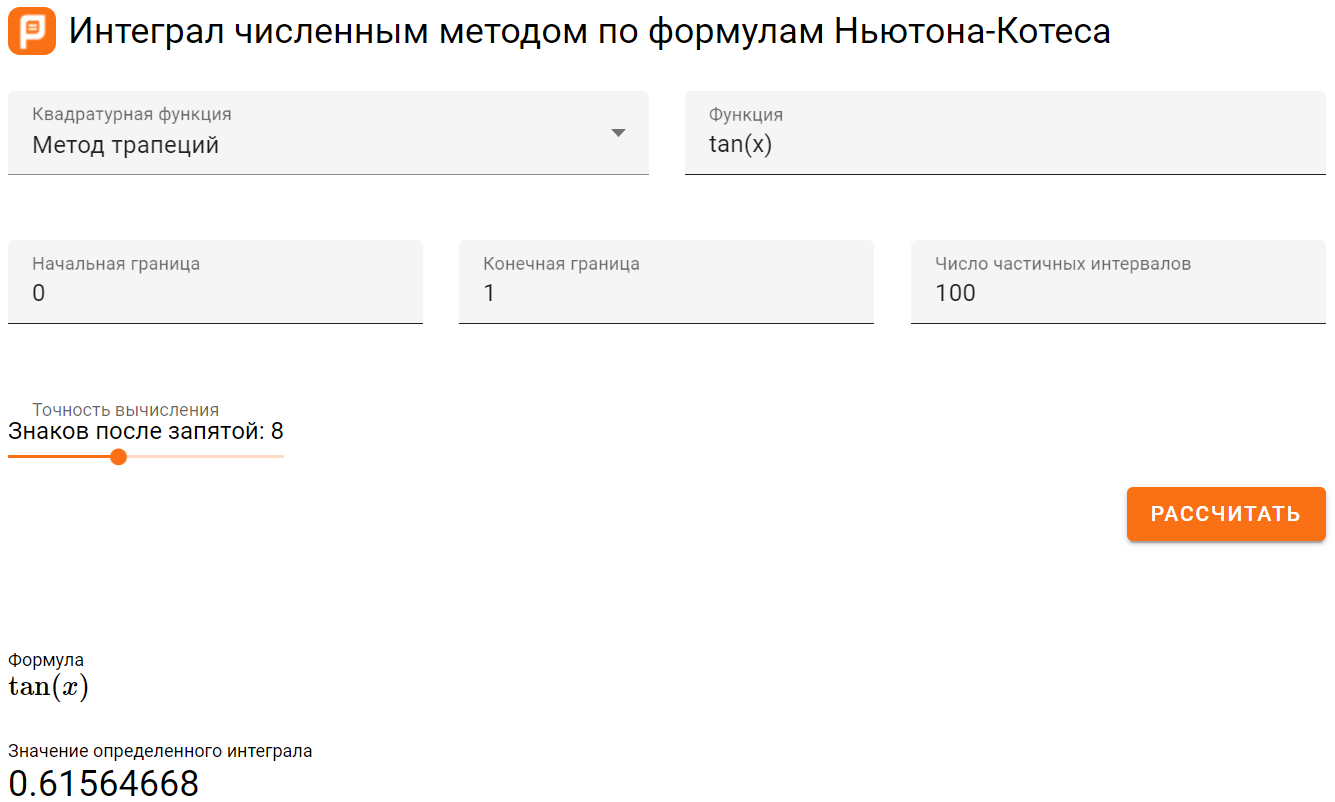


Рисунок №6 – Проверка вычисления определённого интеграла (Верх = 0 , Низ = 1, Шаг = 0.01)

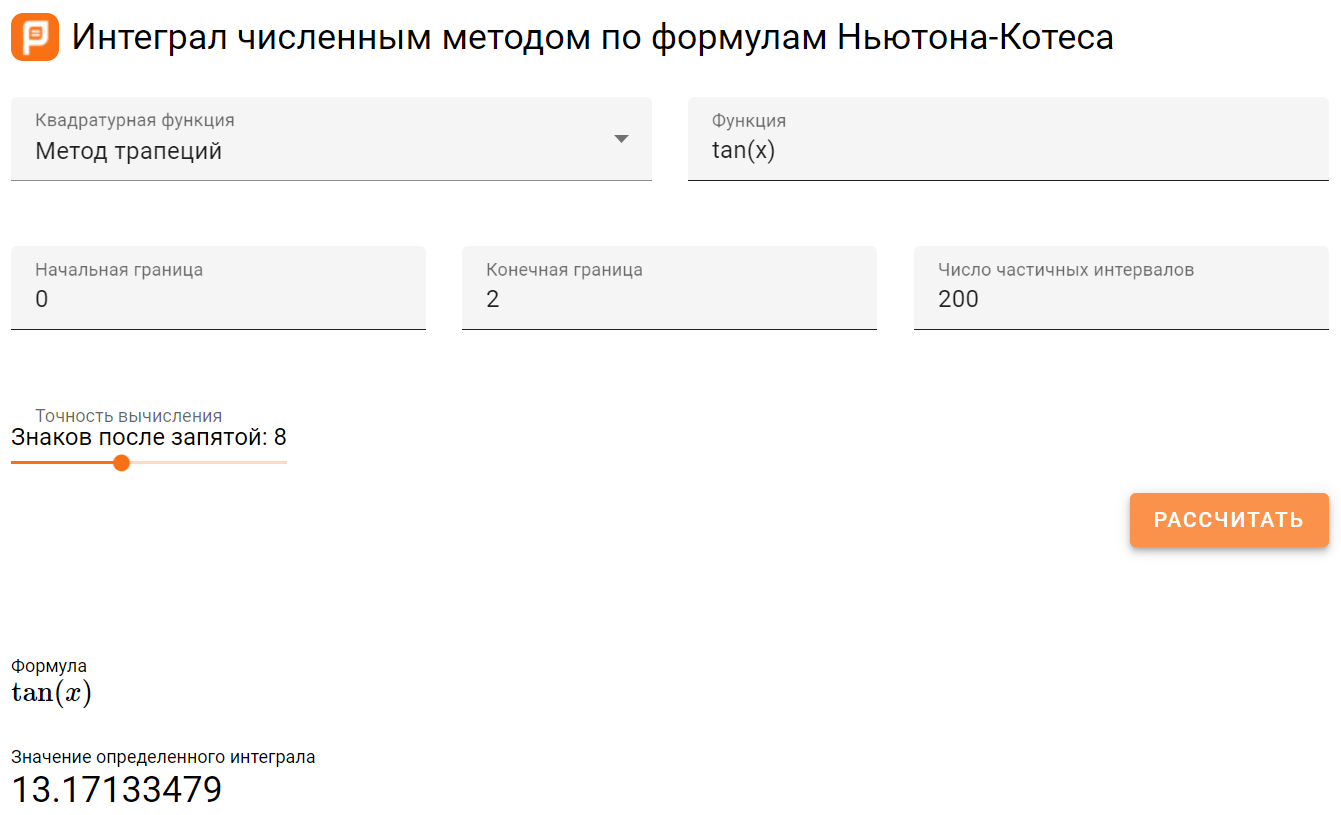


Рисунок №7 – Проверка вычисления определённого интеграла (Верх = 0 , Низ = 2, Шаг = 0.01)

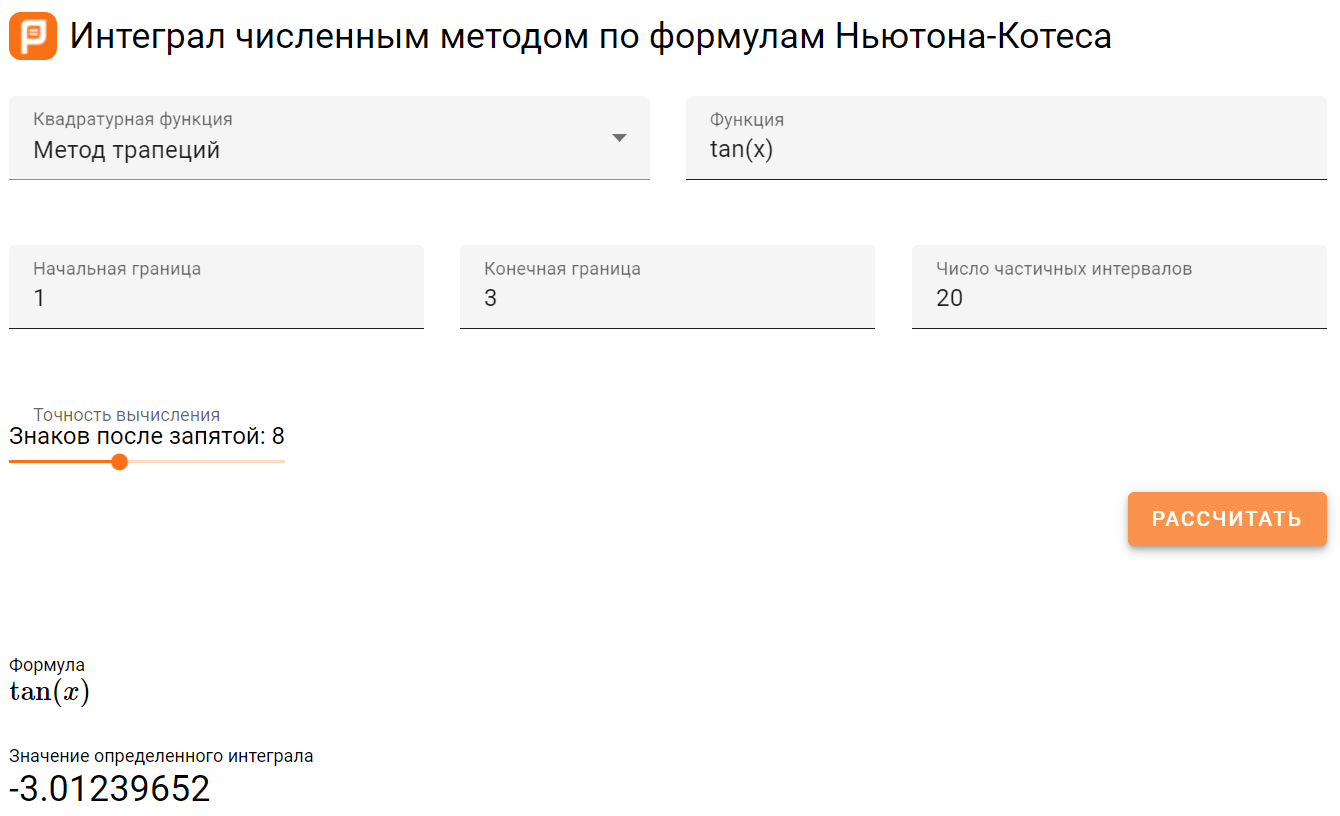


Рисунок №8 – Проверка вычисления определённого интеграла (Верх = 1 , Низ = 3, Шаг = 0.1)

**Вывод:** Мы научились разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.