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»

Выполнили:

студент группы 20ВВП1

Беляков А. М.

Проверили:

Юрова О.В.

Карамышева Н.С.

Пенза 2023

### Цель работы

Научится разрабатывать приложения, обладающие графическим интерфейса пользователя, с использование библиотеки Swing.

### Задание

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.

Функция для интегрирования: cos(x)

**Листинг:**

import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.Arrays;  
import java.util.Vector;  
  
public class Application extends JFrame {  
 private JTextField stepTextField;  
 private JTextField minTextField;  
 private JTextField maxTextField;  
 private JButton addButton;  
 private JButton deleteButton;  
 private JButton calculateButton;  
 private JTextField result;  
 private JTable table;  
 private JPanel rootPanel;  
  
 private final DefaultTableModel defaultTableModel;  
 private static final String[] *tableHeader* = {"step", "min", "max", "result"};  
  
 public Application() {  
 super("Lab\_1");  
 setContentPane(rootPanel);  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 setVisible(true);  
 setSize(800, 600);  
 defaultTableModel = (DefaultTableModel) table.getModel();  
 Arrays.*stream*(*tableHeader*).forEach(defaultTableModel::addColumn);  
  
 addButton.addActionListener(new AddButtonActionListener());  
 deleteButton.addActionListener(new DeleteButtonActionListener());  
 calculateButton.addActionListener(new CalculateButtonActionListener());  
  
 }  
  
 private void createUIComponents() {  
 table = new JTable(){  
 @Override  
 public boolean isCellEditable(int row,int column){  
 return column != 3;  
 }  
 };  
 }  
  
 private class AddButtonActionListener implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String[] row = new String[3];  
 if ((!stepTextField.getText().isEmpty() && stepTextField.getText().matches("-?\\d\*+(\\.\\d+)?")) &&  
 (!minTextField.getText().isEmpty() && minTextField.getText().matches("-?\\d\*+(\\.\\d+)?")) &&  
 (!maxTextField.getText().isEmpty() && maxTextField.getText().matches("-?\\d\*+(\\.\\d+)?"))) {  
 row[0] = stepTextField.getText();  
 stepTextField.setText("");  
 row[1] = minTextField.getText();  
 minTextField.setText("");  
 row[2] = maxTextField.getText();  
 maxTextField.setText("");  
 defaultTableModel.addRow(row);  
 }  
 }  
 }  
  
 private class DeleteButtonActionListener implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow != -1) {  
 defaultTableModel.removeRow(selectedRow);  
 }  
 }  
 }  
 private class CalculateButtonActionListener implements ActionListener {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow == -1) {  
 return;  
 }  
 Vector args = defaultTableModel.getDataVector().get(selectedRow);  
 double step = Double.*parseDouble*((String) args.get(0));  
 double min = Double.*parseDouble*((String) args.get(1));  
 double max = Double.*parseDouble*((String) args.get(2));  
 double inResult = 0;  
  
 for (double i = min; i <= max - step; i += step) {  
 if (i != 0) {  
 inResult += (Math.*cos*(i) + Math.*cos*(i + step)) \* step / 2;  
 defaultTableModel.setValueAt(inResult, selectedRow, 3);  
 }  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 new Application();  
 }  
  
}

**Ручной расчет**

****

Рисунок 1.

**Результат работы программы:**

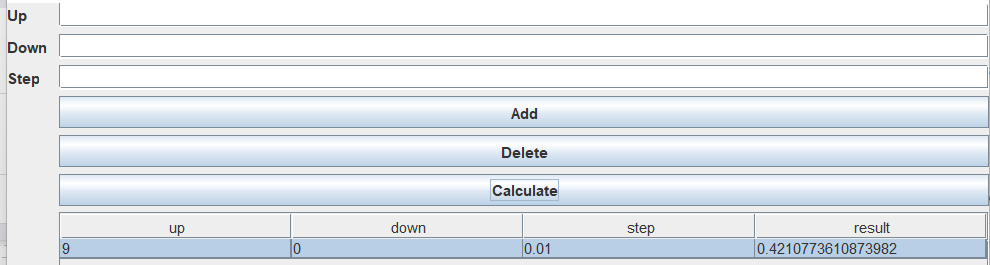


Рисунок 2.

Добавление строки:

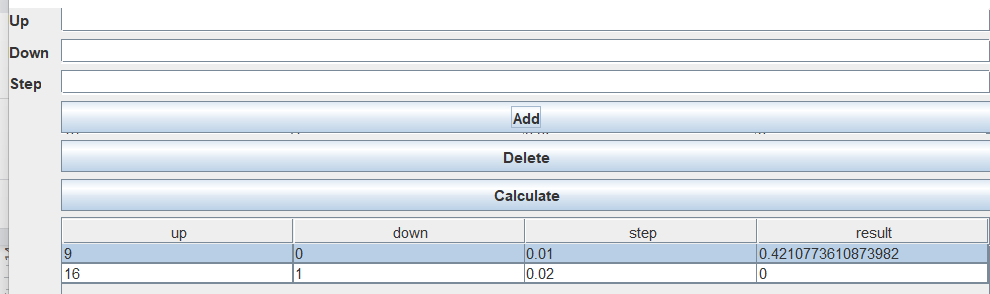


Рисунок 3.

Подсчет строки:

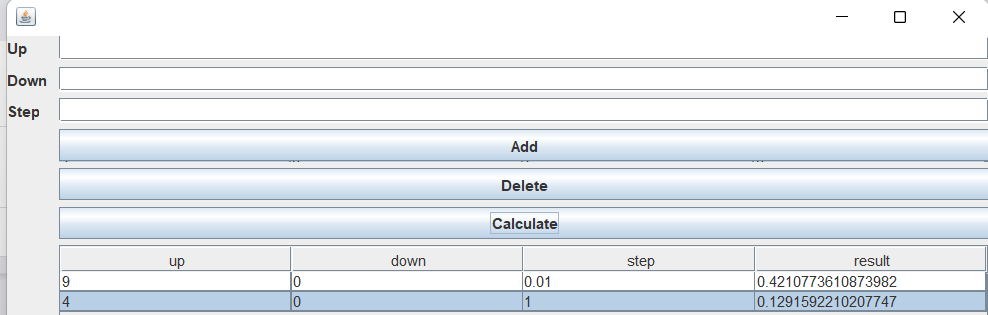


Рисунок 4.

Удаление строки:

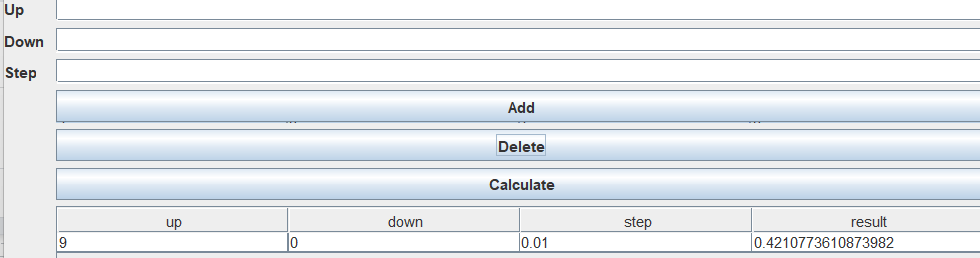


Рисунок 5.

### Вывод

Научились разрабатывать приложения, обладающие графическим интерфейса пользователя, с использование библиотеки Swing.