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 20ВВП2:

Горожанин Я. А.

Скирдова В. М.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2023

**Цель работы:** приобретение навыки разработки приложений, обладающих графическим интерфейсом пользователя с использованием с использованием языка Java и библиотеки Swing.

**Лабораторное задание:**

Вычислить определенный интеграл функции в соответствии с вариантом задания. 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 и параметров выделенной строки таблицы.

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

|  |  |
| --- | --- |
| Вариант | Функция |
| 5 |  |

**Листинг:**

package com.company;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.event.\*;

public class Form extends JDialog {

private JPanel rootPanel;

private JButton buttonOK;

private JButton buttonCancel;

private JButton addButton;

private JButton delButton;

private JButton calkButton;

private JTextField input1;

private JTextField input2;

private JTextField input3;

private JTable table1;

public Form() {

setContentPane(rootPanel);

setModal(true);

getRootPane().setDefaultButton(buttonOK);

createTable();

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

double limUp, limDown, step;

String str\_limUp, str\_limDown, str\_step;

str\_limUp = input1.getText();

str\_limDown = input2.getText();

str\_step = input3.getText();

try {

limUp = Double.*parseDouble*(str\_limUp);

limDown = Double.*parseDouble*(str\_limDown);

step = Double.*parseDouble*(str\_step);

} catch (Exception e) {

ShowMsg("Введено некорректное значение");

return;

}

DefaultTableModel model = (DefaultTableModel) table1.getModel();

model.addRow(new Object[]{str\_limUp, str\_limDown, str\_step});

input1.setText("");

input2.setText("");

input3.setText("");

UpdateWindow();

}

});

delButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int SelectedRow = table1.getSelectedRow();

int RowCount = table1.getRowCount();

if (SelectedRow == -1) {

ShowMsg("Не выбрана строка в таблице ");

return;

}

DefaultTableModel model = (DefaultTableModel) table1.getModel();

model.removeRow(SelectedRow);

table1.setModel(model);

if (SelectedRow == RowCount - 1) {

table1.changeSelection(SelectedRow - 1, 0, false, false);

} else {

table1.changeSelection(SelectedRow, 0, false, false);

}

UpdateWindow();

}

});

calkButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int row = table1.getSelectedRow();

DefaultTableModel model = (DefaultTableModel) table1.getModel();

double limUp = Double.*parseDouble*((String) model.getValueAt(row, 0));

double limDown = Double.*parseDouble*((String) model.getValueAt(row, 1));

double Step = Double.*parseDouble*((String) model.getValueAt(row, 2));

double sum = 0;

while (limDown + Step < limUp) {

sum += ((Math.*exp*(-limDown) + Math.*exp*(-(limDown + Step))) / 2) \* Step;

limDown += Step;

}

sum += ((Math.*exp*(-limDown) + Math.*exp*(-limUp)) / 2) \* Step;

model.setValueAt(sum, row, 3);

UpdateWindow();

}

});

buttonOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

onOK();

}

});

buttonCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

onCancel();

}

});

*// call onCancel() when cross is clicked*

setDefaultCloseOperation(*DO\_NOTHING\_ON\_CLOSE*);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

onCancel();

}

});

*// call onCancel() on ESCAPE*

rootPanel.registerKeyboardAction(new ActionListener() {

public void actionPerformed(ActionEvent e) {

onCancel();

}

}, KeyStroke.*getKeyStroke*(KeyEvent.*VK\_ESCAPE*, 0), JComponent.*WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT*);

}

private void onOK() {

*// add your code here*

dispose();

}

private void onCancel() {

*// add your code here if necessary*

dispose();

}

private void createTable() {

table1.setModel(new DefaultTableModel(

null,

new String[]{

"Верхняя граница интегрирования ", "Нижняя граница интегрирования",

"Шаг интегрирования", "Результат"}

) {

@Override

public boolean isCellEditable(int row, int column) {

return column != 3;

}

});

}

private void ShowMsg(String s) {

this.setVisible(true);

JOptionPane.*showMessageDialog*(null, s);

this.setVisible(true);

}

private void UpdateWindow() {

this.setVisible(true);

}

public static void main(String[] args) {

Form dialog = new Form();

dialog.pack();

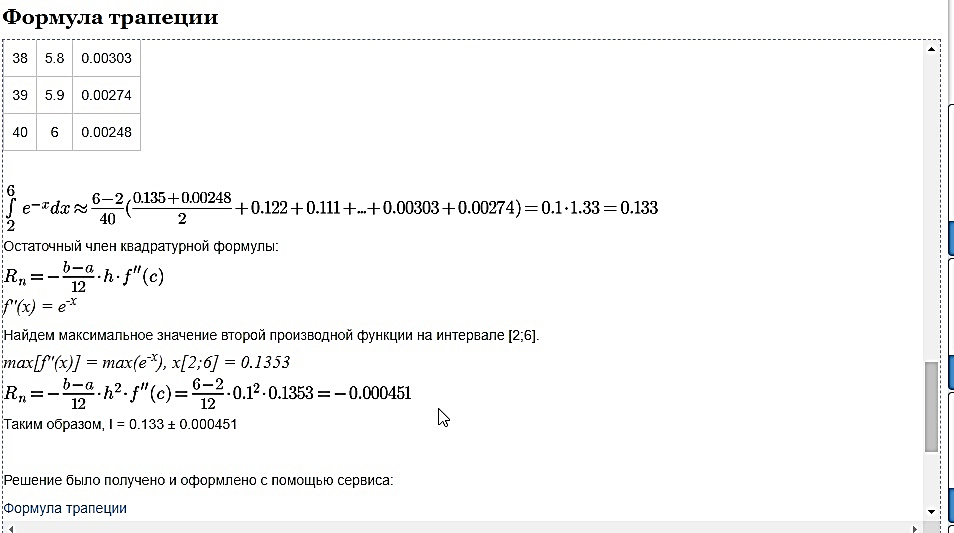
dialog.setVisible(true);

System.*exit*(0);

}

}

**Ручной просчет:**



**Результат выполнения программы:**

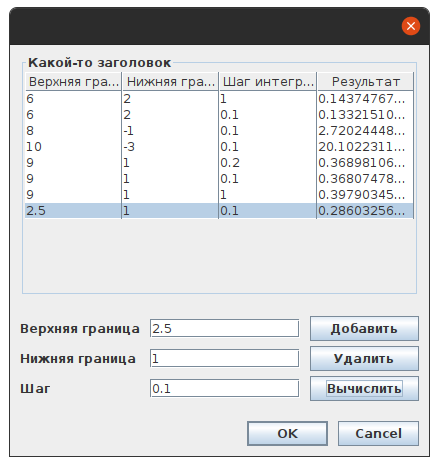


Рисунок 1 — Результаты работы программы

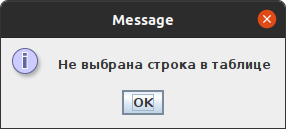


Рисунок 2 — Обработка исключений

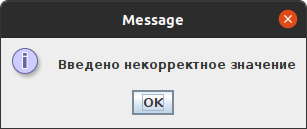


Рисунок 3 — Обработка исключений

**Вывод:** в ходе выполнения данной лабораторной работы были приобретены навыки разработки приложений, обладающих графическим интерфейсом пользователя с использованием с использованием языка Java и библиотеки Swing.