Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине: "Разработка кроссплатформенных приложений"

на тему: "Графические интерфейсы"

Выполнили:

Студенты группы 21ВВB1

Кирьянов В.Е.

Аляев А.О.

Приняли

К.т.н. Юрова О.В.

К.т.н. Карамышева Н.С.

Пенза 2024

Название работы: «Графические интерфейсы».

**Цель работы:** научиться разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.

Задание на лабораторную работу:

Вычислить определенный интеграл функции √x. 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 и параметров выделенной строки таблицы. Результат должен выводиться в четвертой колонке, которая не доступна для редактирования.

Ход работы:

В классе Main реализованы функции для расчёта интеграла √x.

Определённый интеграл рассчитывается методом трапеций, путём вычисления суммы площадей. Каждая площадь зависит от заданного значения шага. Основная функция calculation рассчитывает значение интеграла. В подфункции square рассчитывается значение площади от текущего шага.

В классе NewForm определяется графическая составляющая приложения: кнопки: удалить, добавить, рассчитать, текстовые поля: шаг, верхний и нижний пределы, таблица. В классе есть функции прослушки кнопок, которые выполняются при нажатии, выполняя соответствующие действия. В функции main определяется frame приложения. Графический интерфейс реализован с помощью Java GUI swing.

Листинг программы.

Class Main.

import static java.lang.Math.\*;

public class Main {

public static double Integral(double a, double b, double step) {

return (calculation(a,b,step));

}

public static double calculation(double a, double b, double step) {

double h = b - a;

double stepCount = 0;

if((h/step) % 1 == 0){ stepCount = h / step; }

else{ stepCount = floor(h / step); } //2 //2.5

double left = a;

double right = a + step;

double res = 0.0;

for(double i = 0; i <= stepCount ; i++) {

double a1 = integral(left);

double b1 = integral(right);

res = res + square(step,a1,b1);

left = left + step;

if(b < (right + step)){break;}

right = right + step;

}

if(b < (right + step)){

right = b;

step = b - left;

double a1 = integral(left);

double b1 = integral(right);

res = res + square(step,a1,b1);

}

return res;

}

public static double integral(double a) {

//return (double) 2 / 3 \* sqrt(pow(a,3));

return sqrt(a);

}

public static double square(double h, double a, double b){

return (double) 1 / 2 \* h \* (a + b);

//return b - a;

}

}

Class NewForm.

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class NewForm {

double a = 0; //down

double b = 0; //upp

double step = 0; //step

private JPanel panel1;

private JTextField textField1;

private JTextField textField2;

private JTextField textField3;

private JButton CalcButton;

private JButton AddButton;

private JButton DelButton;

private JTable table1;

private JLabel label1;

public NewForm() {

CalcButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

for(int row = 1; row < table1.getRowCount(); row++){

if(table1.getValueAt(row,0) == "" || table1.getValueAt(row,1) == "" || table1.getValueAt(row,2) == ""){ }

else {

step = Double.parseDouble(table1.getValueAt(row, 0).toString());

b = Double.parseDouble(table1.getValueAt(row, 1).toString());

a = Double.parseDouble(table1.getValueAt(row, 2).toString());

table1.setValueAt(Main.Integral(a, b, step), row, 3);

}

}

}

});

AddButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

table1.setValueAt(table1.getValueAt(1,0),2,0);

table1.setValueAt(table1.getValueAt(1,1),2,1);

table1.setValueAt(table1.getValueAt(1,2),2,2);

table1.setValueAt(table1.getValueAt(1,3),2,3);

table1.setValueAt(textField1.getText(),1,0);

table1.setValueAt(textField2.getText(),1,1);

table1.setValueAt(textField3.getText(),1,2);

table1.setValueAt("",1,3);

}

});

DelButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

final int row = table1.getSelectedRow();

//final int column = table1.getSelectedColumn();

for(int i = 0; i < 4; i++) {

table1.setValueAt("", row, i);

}

}

});

}

public static void main(String[] args) {

JFrame frame = new JFrame("NewForm");

frame.setLocationRelativeTo(null);

frame.setContentPane(new NewForm().panel1);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.pack();

frame.setVisible(true);

}

private void createUIComponents() {

String[] columnNames = {"","","",""};

String[][] data = {

{"Шаг", "Верхний", "Нижний", "Результат"},

{"","","",""},

{"","","",""}

};

table1 = new JTable(data, columnNames);

//table1.setBounds(30, 40, 200, 300);

table1.setModel(new DefaultTableModel(data,

columnNames

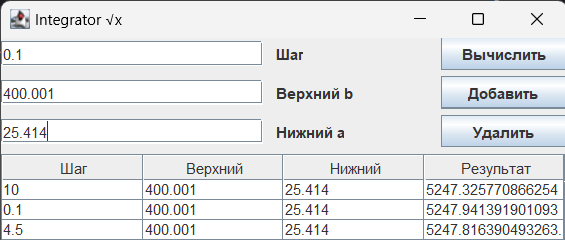
) {public boolean isCellEditable(int row, int column){return false;}}

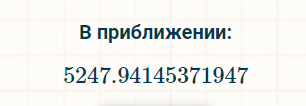
);

}

}

Результаты программы.





Вывод: научились разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing. С увеличением шага точность увеличивается.