Пензенский Государственный Университет

Кафедра «Вычислительная техника»

Отчет о выполнении работы по дисциплине

«Программирование на языке Java»

Лабораторная работа №2

Выполнили: ст. группы 20ВВП1

Александров В.С.

Сафронов Д.В.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза, 2023

**Цель работы**: изучить библиотеку стандартных коллекций Java Collections

Framework, позволяющую хранить различные структуры данных.

**Заданиe:**

Модифицировать приложение из предыдущей лабораторной работы,

реализовав хранение данных таблицы с использованием библиотеки

коллекций. Для этого реализовать класс RecIntegral, способный хранить одну запись таблицы. Для нечетных вариантов в качестве класса-коллекции выбрать ArrayList, для четных - LinkedList. Кроме того, добавить пару кнопок: очистить / заполнить, которые будут очищать таблицу и заполнять ее данными из коллекции соответственно

**Ход работы:**

1. Добавили класс **RecIntegral с**одержащий класс-коллекцию **ArrayList.**
2. В классе **RecIntegral** реализованы методы для получения и записи данных таблицы.
3. Объект класса **RecIntegral** способен хранить одну строку таблицы нашего приложения.
4. Переписали код из прошлой лабораторной работы, для работы с объектами класса в качестве хранилища данных.

**Листинг**:

**Файл формы**

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.event.\*;

public class GUIForm extends JDialog {

private int realColCnt;

private JPanel contentPane;

private JButton buttonOK;

private JButton buttonCancel;

private JTextField textField1;

private JTextField textField2;

private JTextField textField3;

private JButton добавитьButton;

private JButton удалитьButton;

private JButton вычислитьButton;

private JTable table1;

private JButton заполнитьButton;

private JButton очиститьButton;

////////////////////////////////////ПЕРЕМЕННЫЕ/////////////////////////////////////////////////

private double[] dataT = new double[4];

int num =1;

public RecIntegral[] dataListObject = new RecIntegral[10];

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

////////////////////////////////////////////////////////////////////////////////////////////////

public GUIForm() {

setContentPane(contentPane);

setModal(true);

getRootPane().setDefaultButton(buttonOK);

setTitle("Лаба 1");

createTable();

buttonOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

onOK();

}

});

buttonCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

onCancel();

}

});

добавитьButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

добавитьButton();

}

});

удалитьButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { удалитьButton();

}

});

вычислитьButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

вычислитьButton();

}

});

заполнитьButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

заполнитьButton();

}

});

очиститьButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

очиститьButton();

}

});

// call onCancel() when cross is clicked

setDefaultCloseOperation(DO\_NOTHING\_ON\_CLOSE);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

onCancel();

}

});

// call onCancel() on ESCAPE

contentPane.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

createTable();

}

private void onCancel() {

// add your code here if necessary

dispose();

}

////////////////////////////////////РАБОТА С ДАННЫМИ/////////////////////////////////////////////////

private void добавитьButton() {

dataListObject[num-1]= new RecIntegral();

dataListObject[num-1].setDataA(Double.valueOf(

textField1.getText()));

dataListObject[num-1].setDataB(Double.valueOf(

textField3.getText()));

dataListObject[num-1].setDataC(Double.valueOf(

textField2.getText()));

dataListObject[num-1].setDataD(0.0);

model.addRow(dataListObject[num-1].getDataList());

textField1.setText("");

textField2.setText("");

textField3.setText("");

}

private void удалитьButton() {

int rowNamber;

rowNamber=table1.getSelectedRow();

model.removeRow(rowNamber);

num--;

}

private void заполнитьButton() {

int countData = dataListObject[num].getCount();

num=1;

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

model.addRow(dataListObject[i].getDataList());

num++;

}

}

private void очиститьButton() {

model.setRowCount(0);

}

private void вычислитьButton() {

dataListObject[num-1].

setDataD(Trap(// сохраняем результат вычисления

dataListObject[num-1].getDataA(),

dataListObject[num-1].getDataB(),

dataListObject[num-1].getDataC()));

model.removeRow(num-1);

model.addRow(dataListObject[num-1].getDataList());

num++;

}

public void createTable(){

model.addColumn("Верхняя граница интегрирования");

model.addColumn("Нижняя граница интегрирования");

model.addColumn("Шаг интегрирования");

model.addColumn("Результат");

}

////////////////////////////////////ВЫЧИСЛЕНИЯ ИНТЕГРАЛА/////////////////////////////////////////////////

public static double InFunction(double x) //Подынтегральная функция

{

return 1/(Math.log(x));

}

public Double Trap(double a, double b, double h){

double result=0;

int n = (int)((a-b)/h);

result += (InFunction(a)+InFunction(b))/2;

for(int i = 1; i < n; i++) {

result += InFunction(b + h \* i);

}

return h\*result;

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////

public static void main(String[] args) {

GUIForm dialog = new GUIForm();

dialog.pack();

dialog.setVisible(true);

dialog.setName("laba1");

System.exit(0);

}

}

**Файл класса RecIntegral**

import java.util.ArrayList;

public class RecIntegral {

ArrayList<Double> dataList = new ArrayList<Double>();

private static int counter;

public RecIntegral(){

counter++;

}

/////////////////////////////////СЕТЫ///////////////////////////////////

public void setDataA(Double dataA) {

dataList.add(0, dataA);

}

public void setDataB(Double dataB) {

dataList.add(1, dataB);

}

public void setDataC(Double dataC) {

dataList.add(2, dataC);

}

public void setDataD(Double dataD) {

dataList.add(3, dataD);

}

//////////////////////////////ГЕТЫ////////////////////////////////////////

public Double[] getDataList() {

Double[] dblArray = new Double[dataList.size()];

dblArray = dataList.toArray(dblArray);

return dblArray;

}

public Double getDataA() {

return dataList.get(0);

}

public Double getDataB() {

return dataList.get(1);

}

public Double getDataC() {

return dataList.get(2);

}

public Double getDataD() {

return dataList.get(3);

}

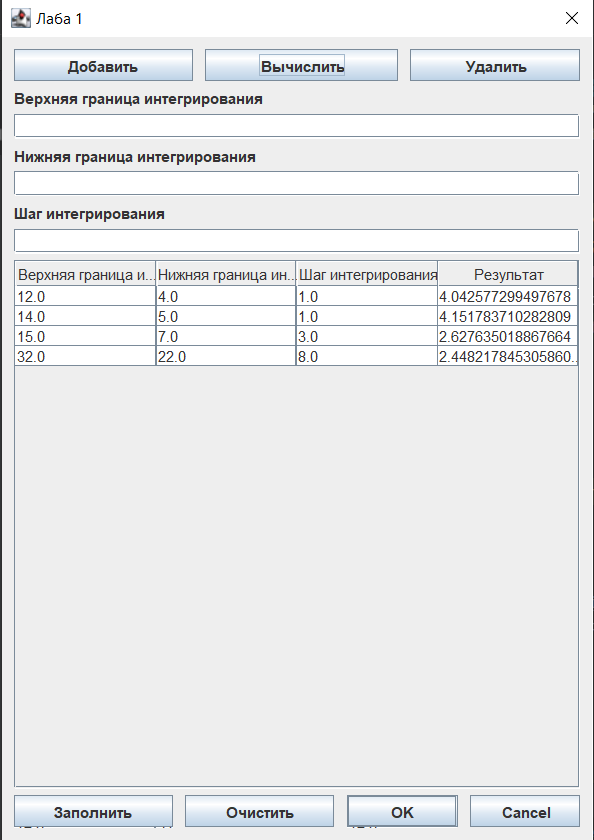
public static int getCount() {

return counter;

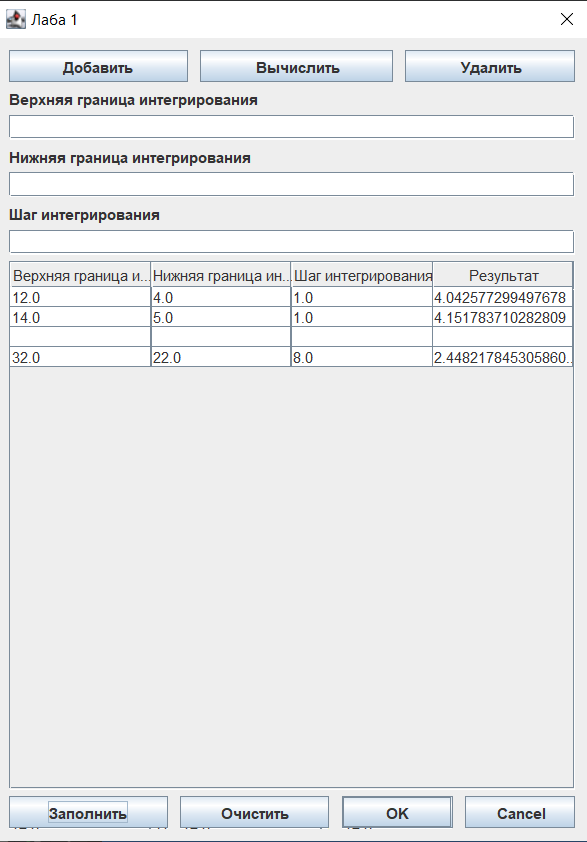
}

}

До удаления из коллекции



После удаления из коллекции



**Вывод:** изучили библиотеку стандартных коллекций Java Collections

Framework, позволяющую хранить различные структуры данных