Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

По лабораторной работе №2

По дисциплине: «Программирование на языке Java»

На тему: «Работа с коллекциями объектов»

Выполнили:

Студент группы 20ВВП1

Кочергин Владислав

Папшев Никита

Принял:

Юрова О.В.

Карамышева Н.С.

***Пенза 2022***

**Цель работы:** изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

**Задание**

|  |  |
| --- | --- |
| Номер варианта | Коллекция |
| 9 | ArrayList |

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.

**Листинг**

**Файл Form.java**

import javax.swing.\*;  
import javax.swing.border.TitledBorder;  
import javax.swing.plaf.FontUIResource;  
import javax.swing.table.DefaultTableModel;  
import javax.swing.text.StyleContext;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.Locale;  
  
import static java.lang.Math.cos;  
  
public class Form extends JFrame {  
 private JPanel MainPanel;  
 private JTextField textField1;  
 private JButton enterButton;  
 private JTextField textField2;  
 private JTextField textField3;  
 public JTable table1;  
 private JButton removeButton;  
 private JButton calculateButton;  
 private JButton FillButton;  
 private JButton ClearButton;  
 private JButton WriteButton;  
 private JButton WriteButtonBin;  
 private JButton OpenButton;  
 private JButton OpenButtonBin;  
 private JFileChooser fileChooser;  
  
 public ArrayList<RecIntegral> Stroka = new ArrayList<>();  
  
  
 public Form() {  
 table1.setDefaultEditor(Object.class, null);  
 DefaultTableModel MyModel = (DefaultTableModel) table1.getModel();  
 fileChooser = new JFileChooser("D:\\Work2\\Work\\Java\_Proj\\FirstLab");  
  
 MyModel.addColumn("Верх");  
 MyModel.addColumn("Низ");  
 MyModel.addColumn("Шаг");  
 MyModel.addColumn("Результат");  
  
 enterButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 MyModel.addRow(new Object[]{Double.parseDouble(textField1.getText()),  
 Double.parseDouble(textField2.getText()),  
 Double.parseDouble(textField3.getText()), 0.0});  
 }  
 });  
 removeButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (table1.getSelectedRowCount() == 1) {  
 MyModel.removeRow(table1.getSelectedRow());  
 } else  
 JOptionPane.showMessageDialog(calculateButton, "Пожалуйста выберите 1 любую строку");  
 }  
 });  
 calculateButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (table1.getSelectedRowCount() == 1) {  
 int rowIndex = table1.getSelectedRow();  
 double a = Double.parseDouble(MyModel.getValueAt(rowIndex, 0).toString());  
 double b = Double.parseDouble(MyModel.getValueAt(rowIndex, 1).toString());  
 double h = Double.parseDouble(MyModel.getValueAt(rowIndex, 2).toString());  
 double s = 0;  
 int j = 0;  
 for (double i = 0; i <= (a - b) / h; i++) {  
 System.out.println("начало" + i);  
 if (b + (i + 1) \* h <= a)  
 s += (cos(b \* b + j \* h) + cos(b \* b + (j + 1) \* h)) \* h / 2;  
 else  
 s += (cos(b \* b + j \* h) + cos(a \* a)) \* h / 2;  
 j++;  
 System.out.println("готово");  
 }  
  
 table1.setValueAt((double) a, rowIndex, 0);  
 table1.setValueAt((double) b, rowIndex, 1);  
 table1.setValueAt((double) h, rowIndex, 2);  
 table1.setValueAt(s, rowIndex, 3);  
 } else  
 JOptionPane.showMessageDialog(calculateButton, "Пожалуйста выберите 1 любую строку");  
 }  
 });

ClearButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int count = table1.getRowCount();

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

double a = Double.parseDouble(table1.getValueAt(0, 0).toString());

double b = Double.parseDouble(table1.getValueAt(0, 1).toString());

double c = Double.parseDouble(table1.getValueAt(0, 2).toString());

Stroka.add(new RecIntegral(Double.parseDouble(table1.getValueAt(0, 0).toString()),

Double.parseDouble(table1.getValueAt(0, 1).toString()),

Double.parseDouble(table1.getValueAt(0, 2).toString()),

Double.parseDouble(table1.getValueAt(0, 3).toString())));

MyModel.removeRow(0);

}

}

});

FillButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

for (int i = 0; i < Stroka.size(); i++) {

double high = Stroka.get(i).high;

double low = Stroka.get(i).low;

double height = Stroka.get(i).height;

double result = Stroka.get(i).result;

//MyModel.removeRow(0);

MyModel.addRow(new Object[]{high, low, height, result});

}

Stroka.clear();

}

});

}

public static void main(String[] args) {

Form f = new Form();

f.setContentPane(f.MainPanel);

f.setTitle("Shedevr");

f.setSize(600, 300);

f.setVisible(true);

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

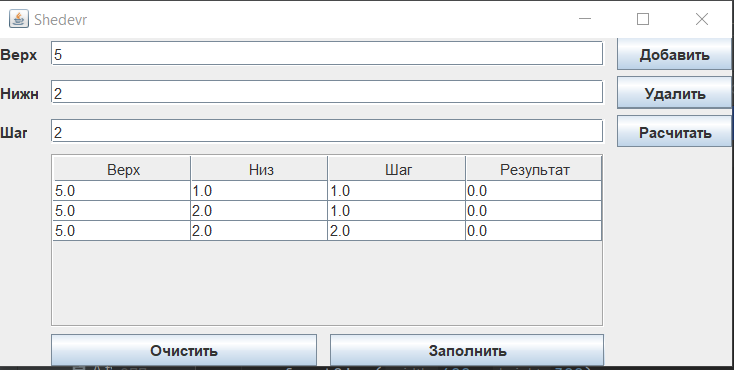
}

**Файл RecIntegral.java**

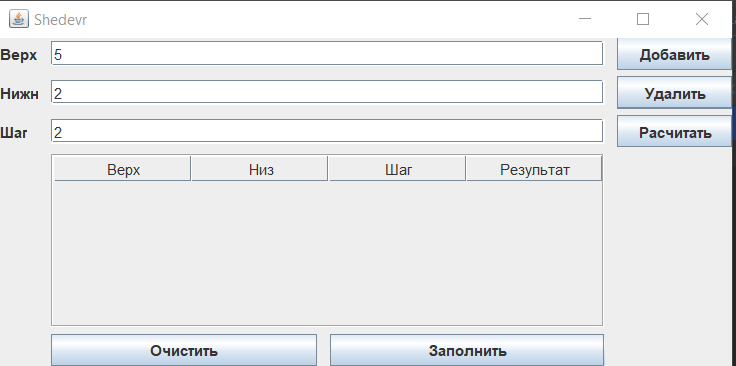
public class RecIntegral{  
  
 public double high;  
 public double low;  
 public double height;  
 public double result;  
 public RecIntegral(double high1, double low1, double height1, double result1) {  
 this.high = high1;  
 this.low = low1;  
 this.height = height1;  
 this.result = result1;  
 }  
}

**Результат работы программы**

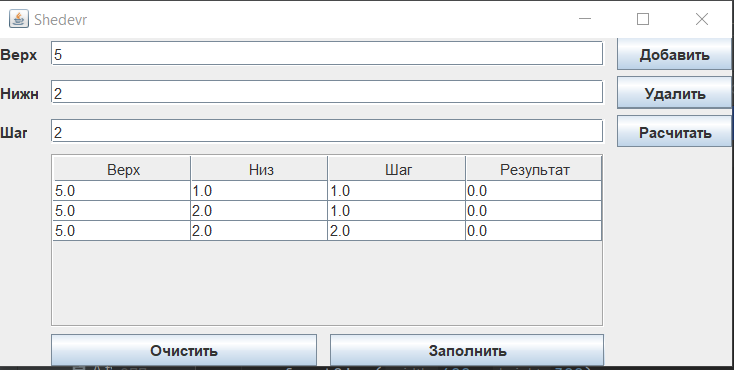
Заполнили таблицу



Нажали на кнопку «Очистить»



Нажали кнопку «Заполнить»



**Вывод:** На данной лабораторной работе нам удалось изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.