Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Отчет о лабораторной работе № 2

по дисциплине «Разработка к-п приложений»

Вариант № 8

Выполнили: ст-ты гр.19ВВ1

Нырков М.

Буданов Д.

Проверили:

Юрова О.В.

Карамышева Н.С.

2022

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

**Задание на лабораторную работу:** 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.

**Решение:**

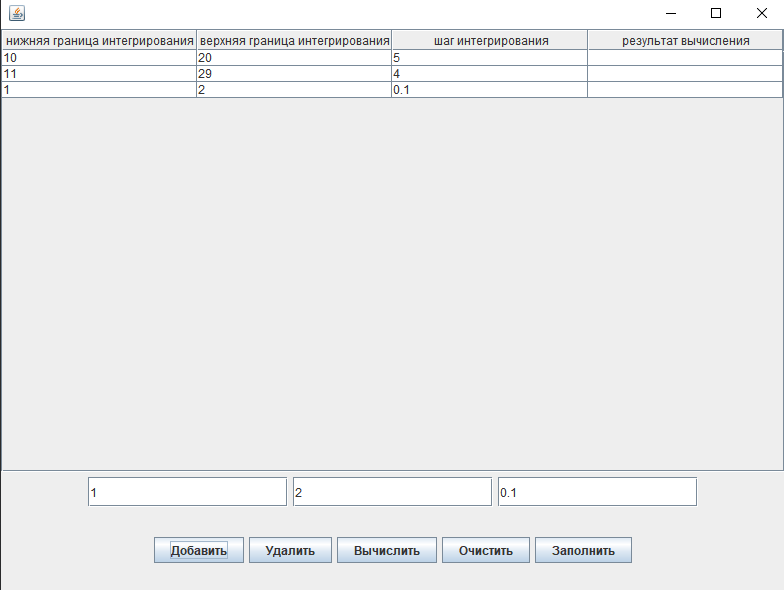


Рисунок 1 - Заполнили таблицу данными

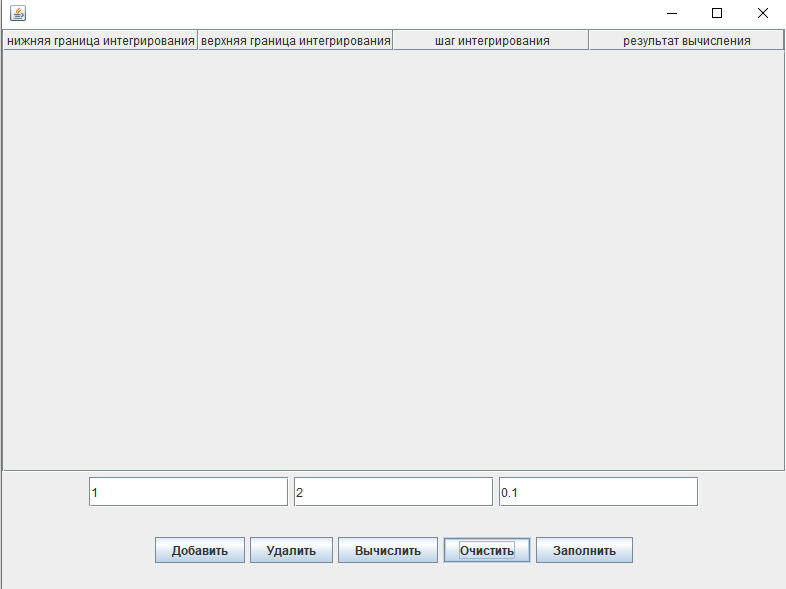


Рисунок 2 - Очистили таблицу с помощью кнопки «Очистить»

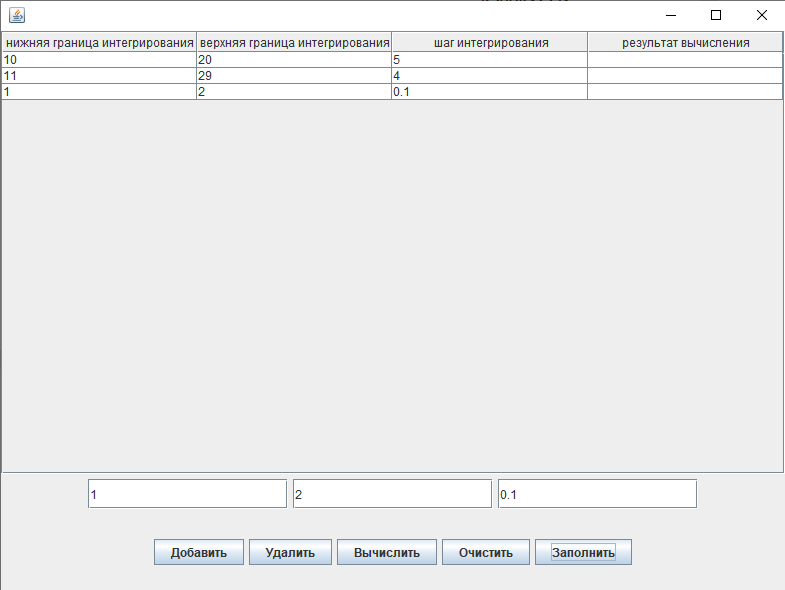


Рисунок 3 - Заполнили таблицу с помощью кнопки «Заполнить»

**Вывод:** изучили библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

**Листинг Main.java**

package com.company;  
  
  
import java.util.concurrent.Delayed;  
  
public class Main {  
  
 public static void main(String[] args) {  
 System.out.println("Hello World!");  
 HOPE w = new HOPE();  
 w.setVisible(true);  
 }  
}

**Листинг RecIntegral.java**

package com.company;  
  
public class RecIntegral {  
 public String start = "";  
 public String end = "";  
 public String step = "";  
 public String result = "";  
  
 public RecIntegral(){  
  
 }  
  
 public RecIntegral(String \_start, String \_end, String \_step){  
 start = \_start;  
 end = \_end;  
 step = \_step;  
 }  
}

**Листинг HOPE.java**

package com.company;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
import java.util.LinkedList;  
import java.util.Vector;  
  
public class HOPE extends JFrame{  
 private DefaultTableModel tableModel;  
 private JTable table1;  
  
 private JTextField textField1;  
 private JTextField textField2;  
 private JTextField textField3;  
 private JButton add;  
 private JButton remove;  
 private JButton calc;  
 private JButton clear;  
 private JButton fill;  
  
 private double E = 2.7182818284590452353602874713527;  
 private ArrayList<RecIntegral> ne\_chet = new ArrayList<>();  
 private LinkedList<RecIntegral> chet = new LinkedList<>();  
  
 Object[] columnNames = {"нижняя граница интегрирования",  
"верхняя граница интегрирования",  
"шаг интегрирования",  
"результат вычисления"};  
  
 public HOPE(){  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
  
 CreateTextModel();  
 CreateTextField();  
 CreateButton();  
  
 CreateForm();  
 }  
  
 private void CreateTextModel(){  
 tableModel = new DefaultTableModel();  
 tableModel.setColumnIdentifiers(columnNames);  
  
 table1 = new JTable(tableModel);  
 }  
  
 private void CreateTextField(){  
 textField1 = new JTextField();  
 textField1.setPreferredSize(new Dimension(200, 30));  
 textField2 = new JTextField();  
 textField2.setPreferredSize(new Dimension(200, 30));  
 textField3 = new JTextField();  
 textField3.setPreferredSize(new Dimension(200, 30));  
 }  
  
 private void CreateButton(){  
 add = new JButton("Добавить");  
 add.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 System.out.println(idx);  
  
 if (ne\_chet.size() >= chet.size()){  
 chet.add(new RecIntegral(textField1.getText(),  
 textField2.getText(), textField3.getText()));  
  
 tableModel.insertRow(tableModel.getRowCount(), new String[] {chet.getLast().start,  
 chet.getLast().end, chet.getLast().step, chet.getLast().result});  
 }else{  
 ne\_chet.add(new RecIntegral(textField1.getText(),  
 textField2.getText(), textField3.getText()));  
  
 tableModel.insertRow(tableModel.getRowCount(), new String[] {ne\_chet.get(ne\_chet.size() - 1).start,  
 ne\_chet.get(ne\_chet.size() - 1).end, ne\_chet.get(ne\_chet.size() - 1).step,  
 ne\_chet.get(ne\_chet.size() - 1).result});  
 }  
 }  
 });  
  
 remove = new JButton("Удалить");  
 remove.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1) {  
 return;  
 }  
 tableModel.removeRow(idx);  
 if(idx % 2 == 0){  
 chet.remove(idx / 2);  
 }else{  
 ne\_chet.remove(idx / 2);  
 }  
 }  
 });  
  
 calc = new JButton("Вычислить");  
 calc.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int idx = table1.getSelectedRow();  
 if(idx == -1){  
 return;  
 }  
 double[] data;  
 try {  
 data = new double[]{Double.valueOf((String) tableModel.getValueAt(idx, 0)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 1)),  
 Double.valueOf((String) tableModel.getValueAt(idx, 2))};  
 }catch(Throwable t){  
 tableModel.setValueAt("NULL", idx, 3);  
 return;  
 }  
 double start, end, step, result;  
 System.out.println();  
 start = data[0];  
 end = data[1];  
 step = data[2];  
 result = 0.0;  
  
 while(start < end){  
 if(start + step > end){  
 step = end - start;  
 }  
 result += 0.5 \* (Math.pow(E, -start) + Math.pow(E, -(start + step))) \* step;  
 start += step;  
 }  
 tableModel.setValueAt(result, idx, 3);  
 }  
 });  
  
 clear = new JButton("Очистить");  
 clear.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 while(tableModel.getRowCount() > 0){  
 tableModel.removeRow(0);  
 }  
 }  
 });  
  
 fill = new JButton("Заполнить");  
 fill.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 int i = 0;  
 while(i <= chet.size() + ne\_chet.size()){  
 if(i % 2 == 0) {  
 if ((i / 2) < chet.size()) {  
 tableModel.insertRow(i, new String[]{chet.get(i / 2).start,  
 chet.get(i / 2).end, chet.get(i / 2).step, chet.get(i / 2).result});  
 }  
 }else {  
 if((i / 2) < ne\_chet.size() ) {  
 tableModel.insertRow(i, new String[]{ne\_chet.get(i / 2).start,  
 ne\_chet.get(i / 2).end, ne\_chet.get(i / 2).step,  
 ne\_chet.get(i / 2).result});  
 }  
 }  
 i++;  
 }  
 }  
 });  
 }  
  
 private void CreateForm(){  
 Box contents = new Box(BoxLayout.Y\_AXIS);  
 contents.add(new JScrollPane(table1));  
  
 JPanel text = new JPanel();  
 text.add(textField1);  
 text.add(textField2);  
 text.add(textField3);  
 contents.add(text);  
  
 JPanel button = new JPanel();  
 button.add(add);  
 button.add(remove);  
 button.add(calc);  
 button.add(clear);  
 button.add(fill);  
 contents.add(button);  
  
 getContentPane().add(contents);  
  
 setSize(800, 600);  
 setVisible(true);  
 }  
}