Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Отчет о лабораторной работе № 2

по дисциплине «Программирование на языке JAVA»

Вариант № 4

Выполнили: ст-ты гр. 19ВВ1

Васильев Л.Р

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:**

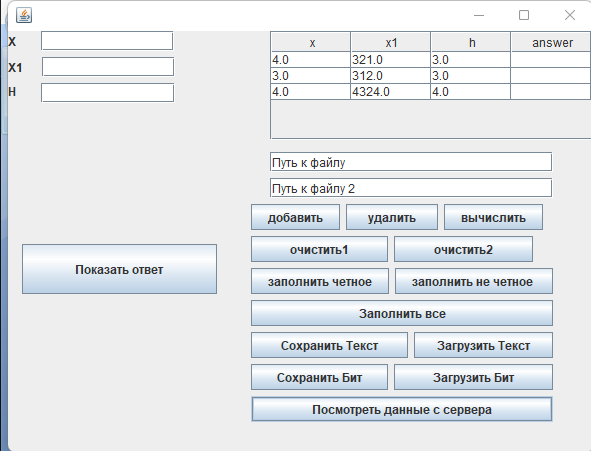


Рисунок 1 - Заполнили таблицу данными

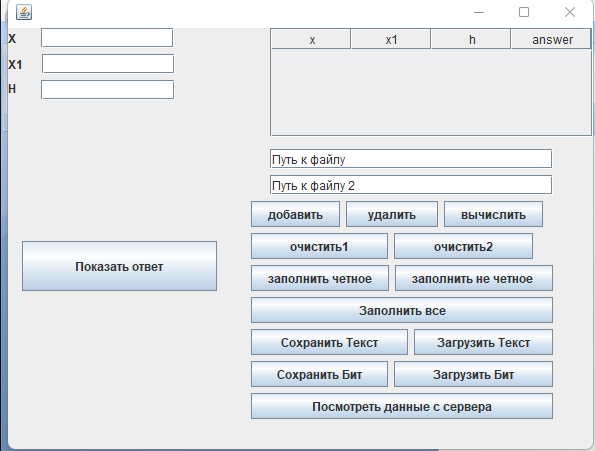
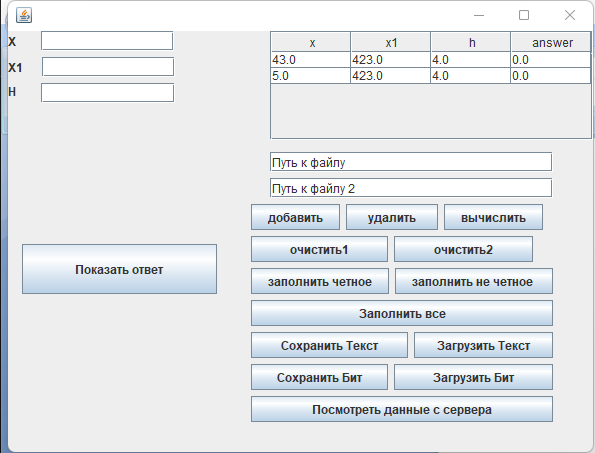


Рисунок 2 - Очистили таблицу с помощью кнопки «Очистить»



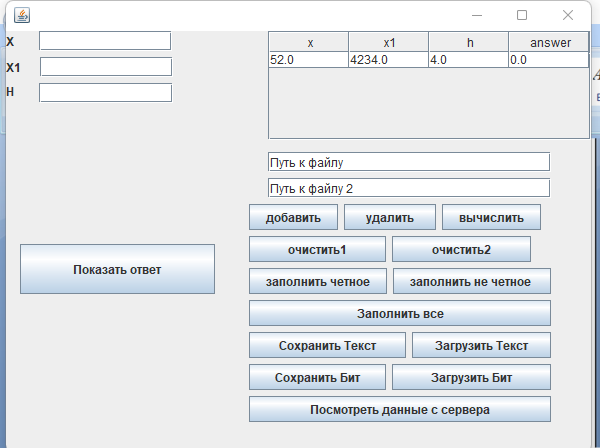


Рисунок 3 - Заполнили таблицу с помощью кнопки «Заполнить»

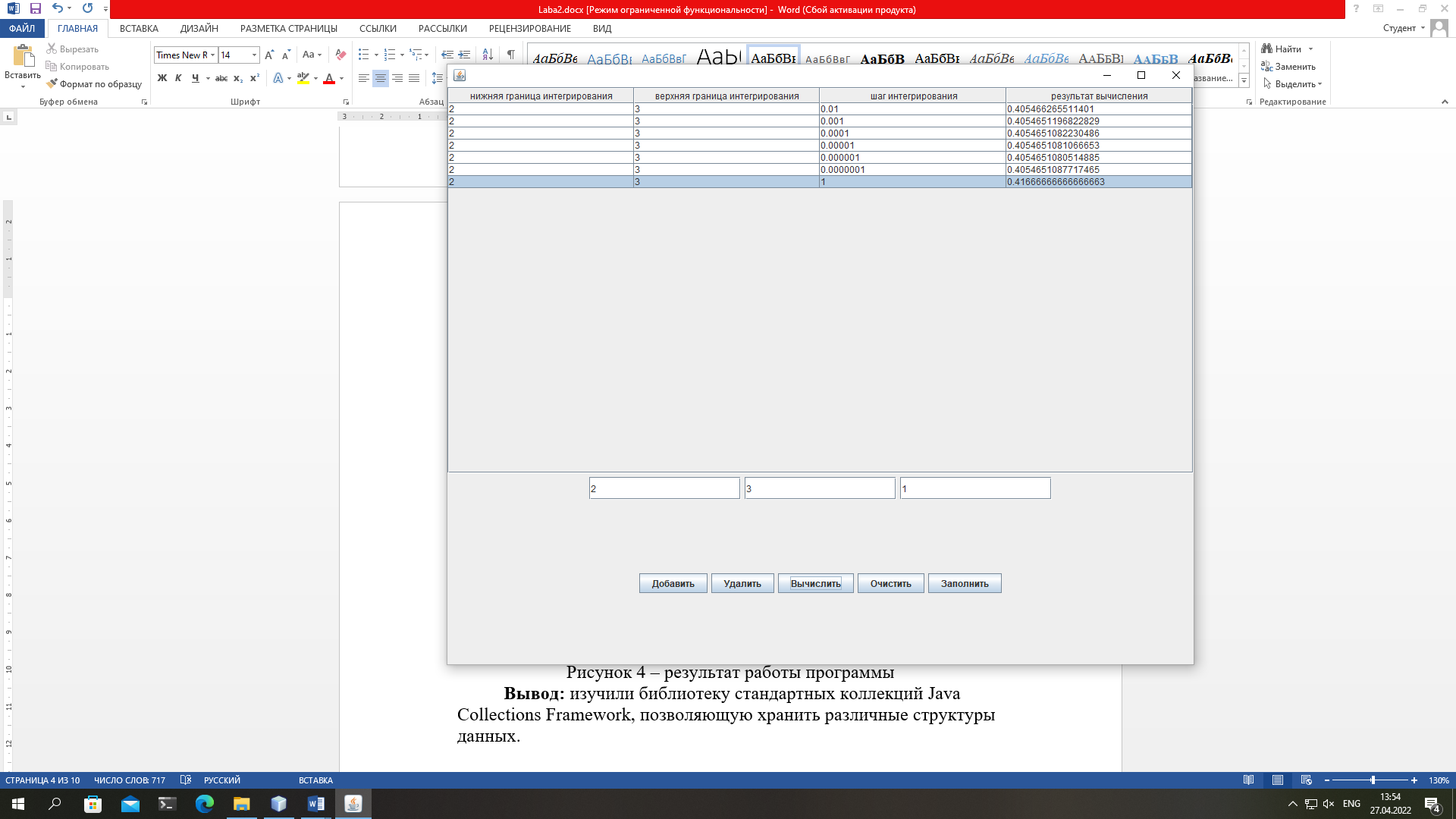


Рисунок 4 – результат работы программы

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

**Листинг RecIntegral.java**

package lab1;

import java.util.ArrayList;

import java.util.LinkedList;

import javax.swing.JOptionPane;

public class RecIntegrall {

ArrayList<DataLab> arr;

LinkedList<DataLab> arr1;

public RecIntegrall(){

arr = new ArrayList<>();

arr1 = new LinkedList<>();

}

public void addArr(DataLab pos){

arr.add(pos);

}

public void clearArr(){

arr.clear();

}

public void addArr1(DataLab pos){

arr1.add(pos);

}

public void clearArr1(){

arr1.clear();

}

}

**Листинг Sourse.java**

/\*

Заполнить

private void jButtonFillActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

if(rec.arr.size() != 0){

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

dtm.addRow(new Object[] {Float.toString(rec.arr.get(i).getX()), Float.toString(rec.arr.get(i).getX1()), Float.toString(rec.arr.get(i).getH()), Float.toString(rec.arr.get(i).getS())});

}

}else{

JOptionPane.showMessageDialog(this, "ArraySize = 0");

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

if(rec.arr1.size() != 0){

for(int i = 0; i< rec.arr1.size(); i++){

dtm.addRow(new Object[] {Float.toString(rec.arr1.get(i).getX()), Float.toString(rec.arr1.get(i).getX1()), Float.toString(rec.arr1.get(i).getH()), Float.toString(rec.arr1.get(i).getS())});

}

}else{

JOptionPane.showMessageDialog(this, "ArraySize = 0");

}

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

int allSize = rec.arr.size() + rec.arr1.size();

int n = 0;

int n1 = 0;

if(allSize != 0){

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

if(rec.arr.size() != 0 && rec.arr1.size() != 0){

if(i % 2 == 0){

dtm.addRow(new Object[] {Float.toString(rec.arr.get(n).getX()), Float.toString(rec.arr.get(n).getX1()), Float.toString(rec.arr.get(n).getH()), Float.toString(rec.arr.get(n).getS())});

n++;

}else{

dtm.addRow(new Object[] {Float.toString(rec.arr1.get(n1).getX()), Float.toString(rec.arr1.get(n1).getX1()), Float.toString(rec.arr1.get(n1).getH()), Float.toString(rec.arr1.get(n1).getS())});

n1++;

}

}else if(rec.arr1.size() != 0 && rec.arr.size() == 0){

dtm.addRow(new Object[] {Float.toString(rec.arr1.get(n1).getX()), Float.toString(rec.arr1.get(n1).getX1()), Float.toString(rec.arr1.get(n1).getH()), Float.toString(rec.arr1.get(n1).getS())});

n1++;

}else if(rec.arr1.size() == 0 && rec.arr.size() != 0){

dtm.addRow(new Object[] {Float.toString(rec.arr.get(n).getX()), Float.toString(rec.arr.get(n).getX1()), Float.toString(rec.arr.get(n).getH()), Float.toString(rec.arr.get(n).getS())});

n++;

}

}

}else{

JOptionPane.showMessageDialog(this, "ArraySize = 0");

}

}

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

int nRow = dtm.getRowCount();

int n = 0;

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

if(rec.arr1.size() != 0){

if(i % 2 == 0){

dtm.removeRow(n);

n++;

}

}else if(rec.arr1.size() == 0){

dtm.setRowCount(0);

}

}

rec.clearArr();

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

int nRow = dtm.getRowCount();

int n = 0;

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

if(rec.arr.size() != 0){

if(i % 2 != 0){

dtm.removeRow(n);

n++;

}

}else if(rec.arr.size() == 0){

dtm.setRowCount(0);

}

}

rec.clearArr1();

}