Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по курсу «Программирование на языке Java»

на тему «Работа с коллекциями объектов»

Выполнили:

студенты группы 19ВВ2

Гусев Д.О.

Кубасов И.М.

Принял:

Карамышева Н.С.

Юрова О.В.

Пенза 2021

Цель работы

Изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

Лабораторное задание

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

**Кнопка “очистить”.**

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

for(int i = 0; i < jTable1.getRowCount(); i++)

{

jTable1.setValueAt(null, i, 0);

jTable1.setValueAt(null,i, 1);

jTable1.setValueAt(null,i, 2);

jTable1.setValueAt(null,i, 3);

}

}

**Кнопка “заполнить”.**

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

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

{

jTable1.setValueAt(Array.get(i).GetBottom(), i, 0);

jTable1.setValueAt(Array.get(i).GetHigh(),i, 1);

jTable1.setValueAt(Array.get(i).GetStep(),i, 2);

jTable1.setValueAt(Array.get(i).GetResult(),i, 3);

}

}

**Класс RecIntegral и объявление односвязного списка.**

LinkedList <RecIntegral> Array = new LinkedList <>();

public class RecIntegral

{

private String bottom;

private Double dwBottom;

private String high;

private Double dwHigh;

private String step;

private Double dwStep;

private String result;

private Double dwResult;

public RecIntegral()

{

bottom = "";

dwBottom = 0.0;

high = "";

dwHigh = 0.0;

step = "";

dwStep = 0.0;

result = "";

dwResult = 0.0;

}

public void SetBottom(String bottom)

{

this.bottom = bottom;

this.dwBottom = Double.parseDouble(bottom);

}

public void SetBottom(Double dwBottom)

{

this.dwBottom = dwBottom;

this.bottom = dwBottom.toString();

}

public void SetHigh(String high)

{

this.high = high;

this.dwHigh = Double.parseDouble(high);

}

public void SetHigh(Double dwHigh)

{

this.dwHigh = dwHigh;

this.high = dwHigh.toString();

}

public void SetStep(String step)

{

this.step = step;

this.dwStep = Double.parseDouble(step);

}

public void SetStep(Double dwStep)

{

this.dwStep = dwStep;

this.step = dwStep.toString();

}

public void SetResult(String result)

{

this.result = result;

this.dwResult = Double.parseDouble(result);

}

public void SetResult(Double dwResult)

{

this.dwResult = dwResult;

this.result = dwResult.toString();

}

public String GetBottom()

{

return this.bottom;

}

public Double GetdwBottom()

{

return this.dwBottom;

}

public String GetHigh()

{

return this.high;

}

public Double GetdwHigh()

{

return this.dwHigh;

}

public String GetStep()

{

return this.step;

}

public Double GetdwStep()

{

return this.dwStep;

}

public String GetResult()

{

return this.result;

}

public Double GetdwResult()

{

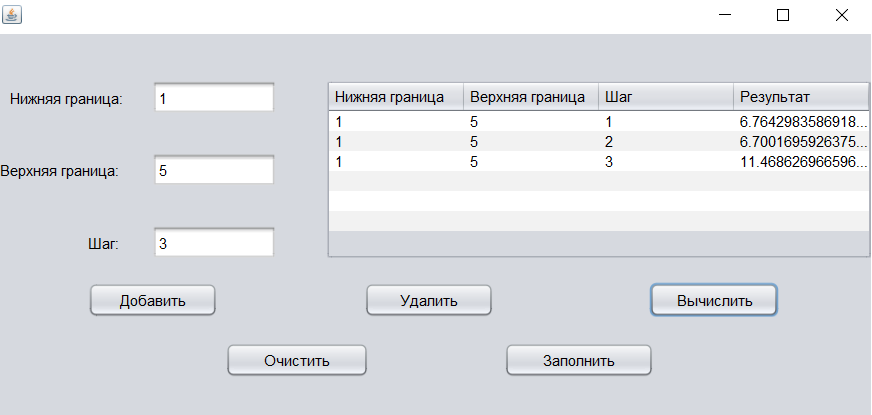
return this.dwResult;

}

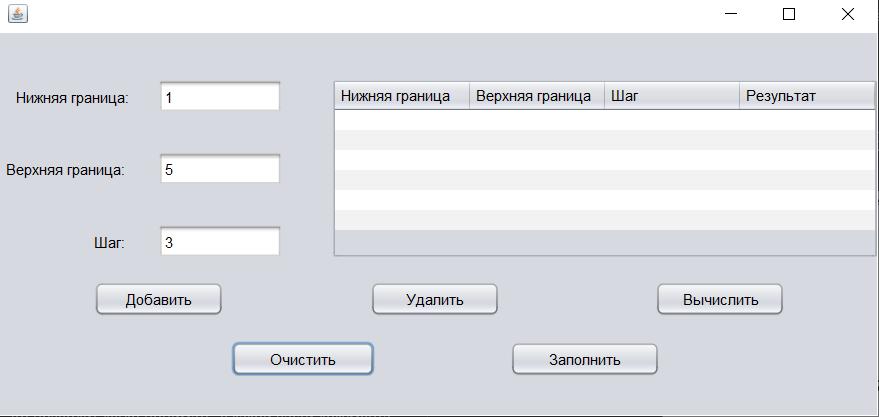
}

Результаты работы программы

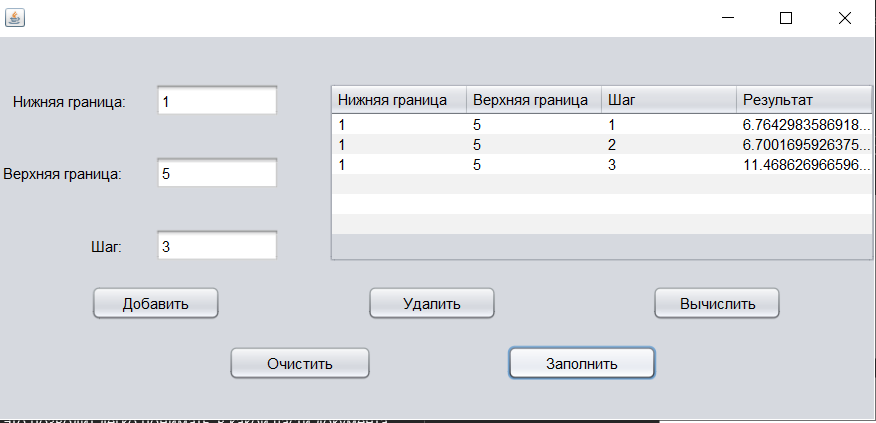
Добавили и вычислили значения.



Нажали кнопку ОЧИСТИТЬ



Нажали кнопку ЗАПОЛНИТЬ



Вывод

Изучили библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.