Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Программирование на языке Java»

на тему «Работа с коллекциями объектов»

Выполнили: студенты группы 22ВВП1

Коннов А. Д.

Митрошин Ю. Е.

Приняли:

Юрова О. В.

Карамышева Н. С.

Пенза 2025

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

Вариант 7:

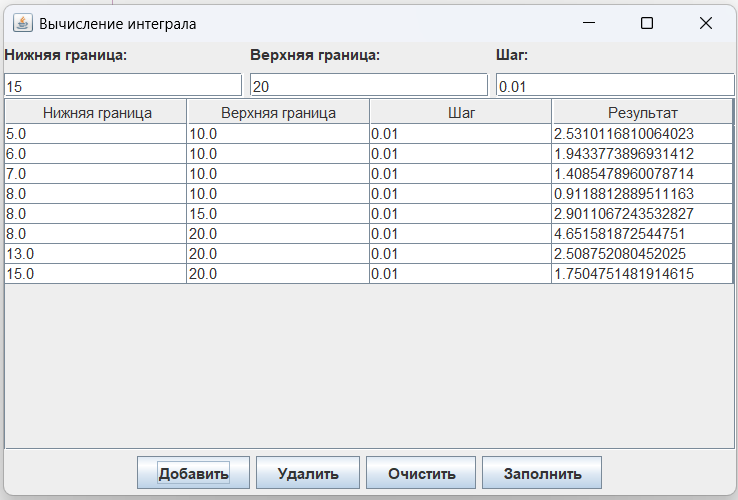
Класс-коллекция: ArrayList.

Изображение выглядит как снимок экрана, линия, Прямоугольник, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Результат работы программы:

Меню приложения:



Очистка данных:

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Заполнение таблицы:

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Вывод: в ходе лабораторной работы мы изучили библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

Приложение А:

Листинг

Файл Mavenproject1.java

package com.mycompany.mavenproject1;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.util.ArrayList;

public class Mavenproject1 extends JFrame {

private JTextField lowerBoundField, upperBoundField, stepField;

private JTable table;

private DefaultTableModel tableModel;

private ArrayList<RecIntegral> records = new ArrayList<>();

public Mavenproject1() {

setTitle("Вычисление интеграла");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(500, 400);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel(new GridLayout(2, 3, 5, 5));

lowerBoundField = new JTextField();

upperBoundField = new JTextField();

stepField = new JTextField();

inputPanel.add(new JLabel("Нижняя граница:"));

inputPanel.add(new JLabel("Верхняя граница:"));

inputPanel.add(new JLabel("Шаг:"));

inputPanel.add(lowerBoundField);

inputPanel.add(upperBoundField);

inputPanel.add(stepField);

add(inputPanel, BorderLayout.NORTH);

tableModel = new DefaultTableModel(new String[]{"Нижняя граница", "Верхняя граница", "Шаг", "Результат"}, 0);

table = new JTable(tableModel);

add(new JScrollPane(table), BorderLayout.CENTER);

JPanel buttonPanel = new JPanel();

JButton addButton = new JButton("Добавить");

JButton removeButton = new JButton("Удалить");

JButton clearButton = new JButton("Очистить");

JButton fillButton = new JButton("Заполнить");

buttonPanel.add(addButton);

buttonPanel.add(removeButton);

buttonPanel.add(clearButton);

buttonPanel.add(fillButton);

add(buttonPanel, BorderLayout.SOUTH);

addButton.addActionListener(e -> addRow());

removeButton.addActionListener(e -> removeRow());

clearButton.addActionListener(e -> clearTable());

fillButton.addActionListener(e -> fillTable());

setVisible(true);

}

private void addRow() {

try {

double lower = Double.parseDouble(lowerBoundField.getText());

double upper = Double.parseDouble(upperBoundField.getText());

double step = Double.parseDouble(stepField.getText());

if (lower >= upper || step <= 0) {

JOptionPane.showMessageDialog(this, "Некорректные значения!", "Ошибка", JOptionPane.ERROR\_MESSAGE);

return;

}

double result = integrate(lower, upper, step);

RecIntegral record = new RecIntegral(lower, upper, step, result);

records.add(record);

tableModel.addRow(new Object[]{record.getLowerBound(), record.getUpperBound(), record.getStep(), record.getResult()});

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Введите числовые значения!", "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

}

private void removeRow() {

int selectedRow = table.getSelectedRow();

if (selectedRow != -1) {

records.remove(selectedRow);

tableModel.removeRow(selectedRow);

} else {

JOptionPane.showMessageDialog(this, "Выберите строку для удаления!", "Ошибка", JOptionPane.WARNING\_MESSAGE);

}

}

private void clearTable() {

tableModel.setRowCount(0);

}

private void fillTable() {

tableModel.setRowCount(0);

for (RecIntegral record : records) {

tableModel.addRow(new Object[]{record.getLowerBound(), record.getUpperBound(), record.getStep(), record.getResult()});

}

}

private double integrate(double lower, double upper, double step) {

double sum = 0.0;

for (double x = lower; x < upper; x += step) {

double newX = Math.min(x + step, upper);

double f1 = function(x);

double f2 = function(newX);

sum += (f1 + f2) \* (newX - x) / 2.0;

}

return sum;

}

private double function(double x) {

return 1.0 / Math.log(x);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(Mavenproject1::new);

}

}

Файл RecInregral.java:

package com.mycompany.mavenproject1;

public class RecIntegral {

private double lowerBound;

private double upperBound;

private double step;

private double result;

public RecIntegral(double lowerBound, double upperBound, double step, double result) {

this.lowerBound = lowerBound;

this.upperBound = upperBound;

this.step = step;

this.result = result;

}

public double getLowerBound() {

return lowerBound;

}

public double getUpperBound() {

return upperBound;

}

public double getStep() {

return step;

}

public double getResult() {

return result;

}

public void setLowerBound(double lowerBound) {

this.lowerBound = lowerBound;

}

public void setUpperBound(double upperBound) {

this.upperBound = upperBound;

}

public void setStep(double step) {

this.step = step;

}

public void setResult(double result) {

this.result = result;

}

}