

ГУАП

КАФЕДРА № 42

ЗАЩИЩЕН С ОЦЕНКОЙ _____
ПРЕПОДАВАТЕЛЬ

К.т.н. Доцент
должность, уч. степень, звание

подпись, дата

В. А. Ушаков

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ НА JAVA

Вариант 5

по курсу: КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № _____ 4128

подпись, дата

В.А.Воробьев

инициалы, фамилия

Санкт-Петербург 2023

Цель работы: изучение процедурной модели программирования, изучение основ работы со строками и файлами на языке JAVA.

Задание:

Разработать программу на языке JAVA, использующую графический пользовательский интерфейс для ввода и вывода информации. Программа должна вычислять значение заданной функции путем разложения в ряд Маклорена с заданной точностью и с использованием стандартной функции класса Math. Аргумент функции и точность должны задаваться пользователем.

При разработке необходимо использовать следующие общие элементы интерфейса.

1. Метку JLabel с формулой разложения заданной функции в ряд Маклорена.
2. Метки JLabel для обозначения элементов ввода данных.
3. Метки JLabel для вывода результатов вычислений (одну для значения, рассчитанного с разложением в ряд Маклорена и одну для эталонного значения, рассчитанного с помощью стандартной функции).
4. Кнопку JButton для инициализации вычислений.
5. Кнопку JButton для сброса входных данных на значения по умолчанию.
6. Кнопку JButton для выхода из программы.
7. Элементы ввода данных должны выбираться исходя из заданного варианта.

Также при разработке программы предусмотреть альтернативный метод ввода исходных данных:

Вместо ввода одного аргумента организовать ввод диапазона аргументов. Т.е. использовать три заданных типа элементов ввода для задания: нижней границы диапазона, верхней границы диапазона и шага изменения.

Результаты вычислений выводить в таблицу JTable в три столбца: аргумент функции, значение, рассчитанное разложением в ряд Маклорена, эталонное значение.

Номер в журнале: 5

Группа 4128

Функция с разложением в ряд Маклорена представлена на рисунке 1.

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}, \quad |x| < 1$$

Рисунок 1 – Функция по варианту

Элементы ввода значений: JSpinner

Обработка строк: JList

Результаты работы программы:

Основной задачей было создание функционального приложения, способного взаимодействовать с пользователем через графический интерфейс. Мы настроили окно приложения, загрузили изображение, реализовали элементы управления, такие как выпадающий список и спиннер, и обеспечили их взаимодействие с программой. Также были добавлены кнопки для выполнения дополнительных действий. Исходный код доступен Приложении и на GitHub (URL - https://github.com/vladcto/suai-labs/tree/355629db65b3539b04f89332b026364acbd0016d/5_semester/%D0%9A%D0%9F/src/lab2).

Главным результатом является успешная реализация программы, способной выполнять необходимые вычисления и обновлять результаты на графическом интерфейсе. Графический интерфейс представлен на рисунках 2-5.

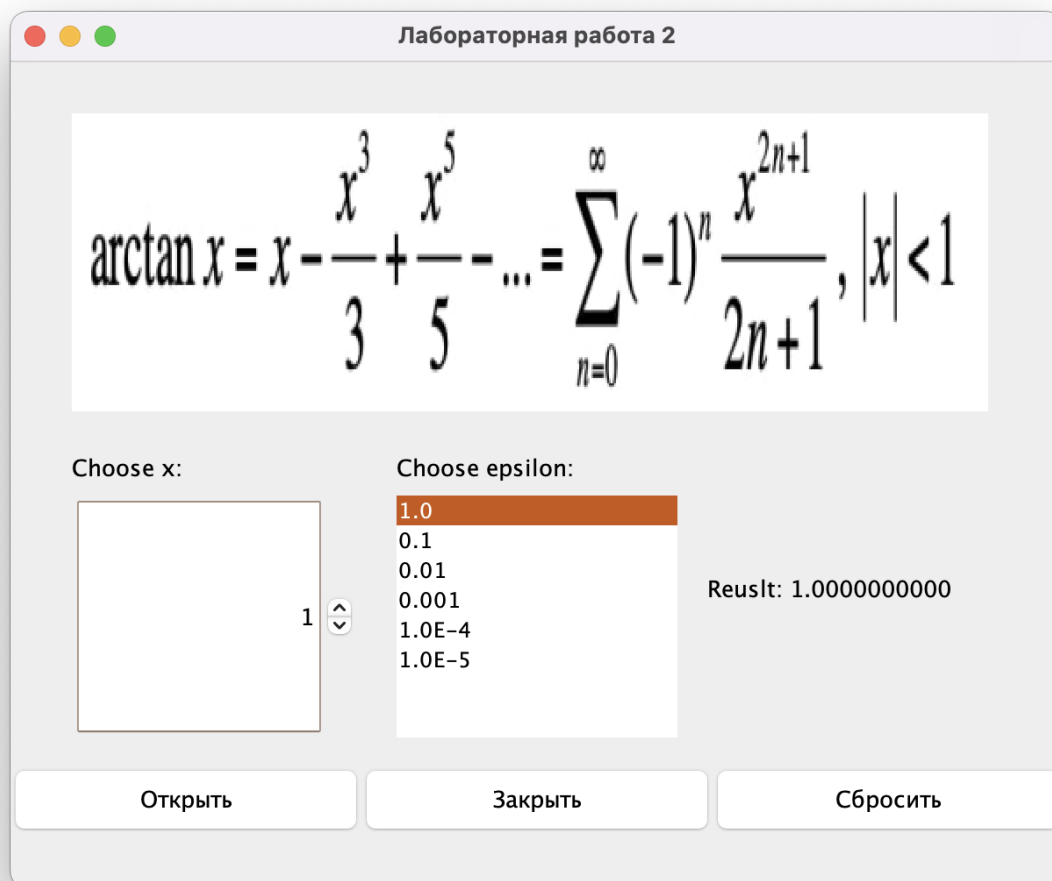


Рисунок 2 – Интерфейс программы

Лабораторная работа 2

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}, |x| < 1$$

Choose x:

0.5

Choose epsilon:

1.0
0.1
0.01
0.001
1.0E-4
1.0E-5

Result: 0.4636398866

Открыть Закрыть Сбросить

Рисунок 3 – Расчет значения

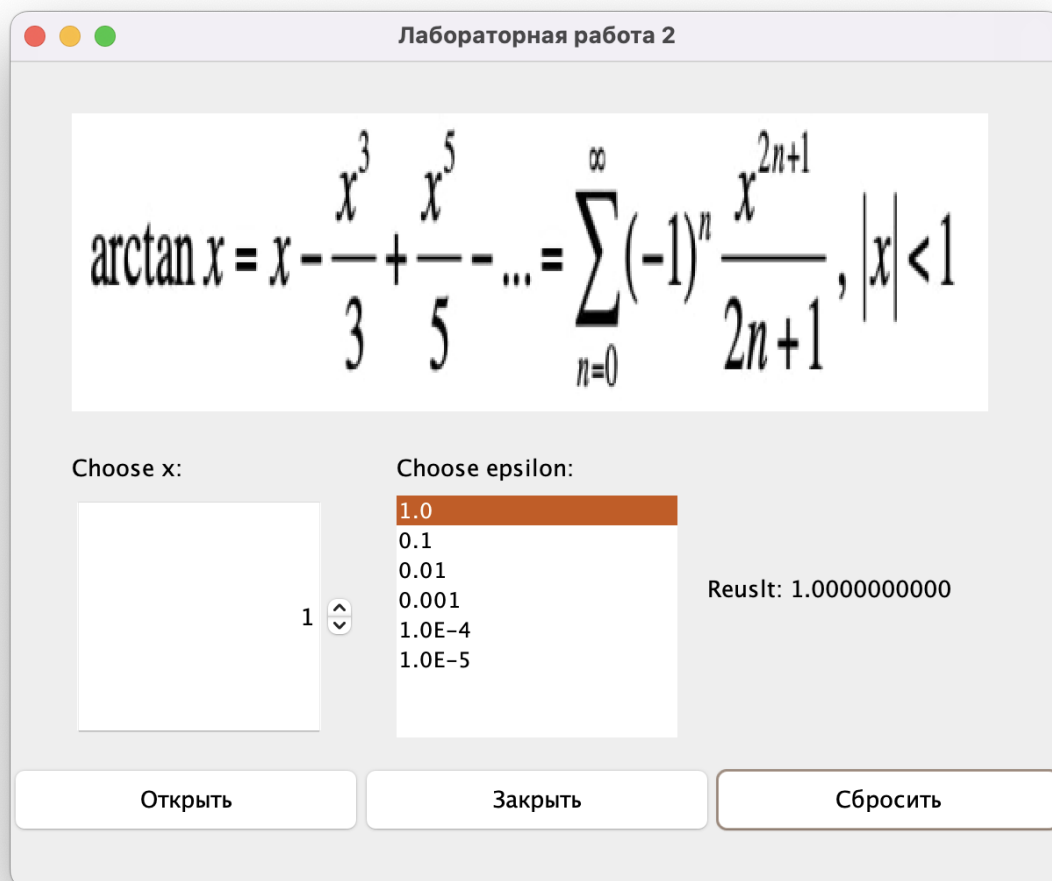


Рисунок 4 – Сброс значения

Начало отрезка:

0.5

Конец отрезка:

1

Шаг:

0.1

Точность:

0.001

Считаем

X	Результат	Эталон
0.5	0.46368427579365074	0.4636476090008061
0.6	0.5403428427802598	0.5404195002705842
0.7	0.6109472310914681	0.6107259643892086
0.7999999999999999	0.6744634216951987	0.6747409422235526
0.8999999999999999	0.7332202611827637	0.7328151017865066
0.9999999999999999	0.7858971648964463	0.7853981633974483

Рисунок 5 – Разложение в диапазоне

Вывод:

В ходе выполнения лабораторной работы были успешно изучены основы процедурной модели программирования и работы со строками и файлами на языке JAVA. Разработанная программа демонстрирует использование графического интерфейса для взаимодействия с пользователем и реализует необходимый функционал. Лабораторная работа позволила более глубоко ознакомиться с языком программирования JAVA и его особенностями.

Таким образом эта лабораторная работа позволила овладеть навыками создания интерфейса и освоить его различные элементы, а также научиться вычислять значение функции при помощи ряда Маклорена.

ПРИЛОЖЕНИЕ А

TableMath.java

```
package lab2;
```

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import java.awt.*;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
public class TableMath extends JFrame {
```

```
    private JTextField startXField, endXField, stepField, precisionField;
```

```
    private JTable resultTable;
```

```
    public TableMath() {
```

```
        setSize(600, 400);
```

```
        initComponents();
```

```
    }
```

```
    private void initComponents() {
```

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

```
inputPanel.add(new JLabel("Start X:"));
```

```
startXField = new JTextField();
```

```
inputPanel.add(startXField);
```

```
inputPanel.add(new JLabel("End X:"));
```

```
endXField = new JTextField();
```

```
inputPanel.add(endXField);
```

```
inputPanel.add(new JLabel("Step:"));
```

```
stepField = new JTextField();
```

```
inputPanel.add(stepField);
```

```
inputPanel.add(new JLabel("Precision:"));
```

```
precisionField = new JTextField();
```

```
inputPanel.add(precisionField);
```

```
inputPanel.add(new JLabel());
```

```
JButton calculateButton = new JButton("Calculate");
```

```
calculateButton.addActionListener((x) -> calculateAndDisplayResults());
```

```
inputPanel.add(calculateButton);
```

```

        resultTable = new JTable(new DefaultTableModel(new Object[]{"X",
"Result", "Reference"}, 0));

        JScrollPane tableScrollPane = new JScrollPane(resultTable);

        add(inputPanel, BorderLayout.NORTH);
        add(tableScrollPane, BorderLayout.CENTER);
    }

    private void calculateAndDisplayResults() {
        DefaultTableModel model = (DefaultTableModel) resultTable.getModel();
        model.setRowCount(0);

        double startX = Double.parseDouble(startXField.getText());
        double endX = Double.parseDouble(endXField.getText());
        double step = Double.parseDouble(stepField.getText());
        double precision = Double.parseDouble(precisionField.getText());

        for (double x = startX; x <= endX; x += step) {
            double result = MyMath.calculateFunctionValue(x, precision);
            double referenceResult = Math.atan(x);

            model.addRow(new Object[]{x, result, referenceResult});
        }
    }

```

```
    }  
}  
}
```

MyMath.java

```
package lab2;
```

```
public final class MyMath {
```

```
    private MyMath() {  
    }  
}
```

```
static double _macloren(double x, int n) {  
    return Math.pow(-1, n) * Math.pow(x, 2 * n + 1) / (2 * n + 1);  
}
```

```
public static double calculateFunctionValue(double x, double precision) {
```

```
    double result = 0;
```

```
    int n = 0;
```

```
    double term = Double.POSITIVE_INFINITY;
```

```
    while (Math.abs(term) > precision) {
```

```
        term = _macloren(x, n);
```

```

        result += term;

        n++;
    }

    return result;
}
}

```

Main.java

```

package lab2;

import java.awt.Image;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.*.*;

public class Main {

    static final double SELECTED_VALUE = 1;

    static final double SELECTED_EPSILON = 1;

```

```

static final int LABEL_HEIGHT = 32;

static final int CONTAINER_HEIGHT = 170 - LABEL_HEIGHT;

static final int CONTAINER_OFFSET = 216;


static final String imagePath =
"/Users/razrab-ytka/Documents/Projects/SUAI_homework/5_semester/KП/assets/f
ormula_1.jpg";

static final JFrame frame = new JFrame("Лабораторная работа 2");

static final JPanel panel = new JPanel(null);

static final JLabel resLabel =

    new JLabel();

static final JSpinner valueSpinner = new JSpinner();

static final Double[] epsilonListValues = {1.0, 0.1, 0.01, 1e-3, 1e-4, 1e-5};

static final JList<Double> epsilonList = new JList<>();

static final String resText = "Reuslt: %.10f";

static final JButton openTable = new JButton("Открыть");

static final JButton close = new JButton("Закрыть");

static final JButton reset = new JButton("Сбросить");


static double selectedValue = SELECTED_VALUE;

static double selectedEpsilon = SELECTED_EPSILON;

```

```

public static void main(String[] args) throws IOException {

    setupFrame();

    setupImage();

    setupResLabel();

    setupSpinner();

    setupEpsilonList();

    setupButtons();

    frame.setVisible(true);

    frame.add(panel);
}

static void setupFrame() {

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    frame.setSize(600, 500);
}

static void setupImage() throws IOException {

    final var width = 522;

    final var height = 170;

    BufferedImage myPicture = ImageIO.read(new File(imagePath));

    Image dimg = myPicture.getScaledInstance(width, height,
Image.SCALE_SMOOTH);

    final var imageLabel = new JLabel(new ImageIcon(dimg));

```



```
imageLabel.setBounds(35, 30, width, height);  
panel.add(imageLabel);  
}
```

```
static void setupEpsilonList() {  
    final var label = new JLabel("Choose epsilon:");  
    label.setBounds(220, CONTAINER_OFFSET, 160, LABEL_HEIGHT);  
    panel.add(label);  
  
    epsilonList.setListData(epsilonListValues);  
    epsilonList.setSelectedIndex(0);  
    selectedValue = epsilonListValues[0];  
    epsilonList.addListSelectionListener((x) -> {  
        selectedEpsilon = epsilonList.getSelectedValue();  
        updateResult();  
    });  
    epsilonList.setBounds(220, CONTAINER_OFFSET + LABEL_HEIGHT,  
160, CONTAINER_HEIGHT);  
    panel.add(epsilonList);  
}
```

```
static void setupResLabel() {  
    resLabel.setBounds(397, CONTAINER_OFFSET, 160, 170);
```

```

    panel.add(resLabel);

    updateResult();
}

static void setupSpinner() {

    final var label = new JLabel("Choose x:");

    label.setBounds(35, CONTAINER_OFFSET, 160, LABEL_HEIGHT);

    panel.add(label);

    valueSpinner.setModel(

        new SpinnerNumberModel(selectedValue, -1, 1, 0.1)

    );

    valueSpinner.addChangeListener((x) -> {

        selectedValue = (double) valueSpinner.getValue();

        updateResult();

    });

    valueSpinner.setBounds(35, CONTAINER_OFFSET + LABEL_HEIGHT,
160, CONTAINER_HEIGHT);

    panel.add(valueSpinner);

}

static void setupButtons() {

```

```
final var y = CONTAINER_OFFSET + LABEL_HEIGHT +  
CONTAINER_HEIGHT + 16;
```

```
openTable.addActionListener((x) -> (new TableMath()).setVisible(true));  
openTable.setBounds(0, y, 600 / 3, 40);  
panel.add(openTable);
```

```
close.addActionListener((x) -> System.exit(0));  
close.setBounds(600 / 3, y, 600 / 3, 40);  
panel.add(close);
```

```
reset.addActionListener((x) -> {  
    selectedValue = SELECTED_VALUE;  
    valueSpinner.setValue(selectedValue);  
    epsilonList.setSelectedIndex(0);  
    selectedEpsilon = epsilonList.getSelectedValue();  
});  
reset.setBounds(600 / 3 * 2, y, 600 / 3, 40);  
panel.add(reset);  
}
```

```
static void updateResult() {
```

```
        final var newValue = MyMath.calculateFunctionValue(selectedValue,  
selectedEpsilon);  
  
        resLabel.setText(String.format(resText, newValue));  
  
    }  
}
```