ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ассистент |  |  | |  | Н. И. Чулочникова |
| должность, уч. степень, звание |  | | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1 |
| ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ |
| по курсу: |
| КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4326 |  |  |  | Г. С. Томчук |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. Цель работы

Целью лабораторной работы является изучение принципов функционального программирования на языке Java и их применение для решения вычислительных задач.

В рамках работы необходимо реализовать программу, вычисляющую значение функции

путём разложения в ряд Маклорена с заданной точностью.

Также требуется сравнить результат, полученный через разложение, со значением функции, вычисленным с помощью стандартной библиотеки Math, и проанализировать погрешность вычислений.

1. Задание

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. Ввод и вывод информации можно осуществлять через командную строку. При разработке следует придерживаться принципов функционального программирования.

Работа выполнялась по варианту № 7:

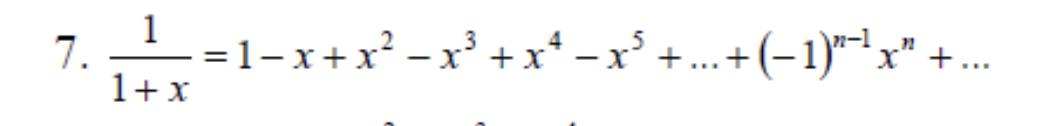


Рисунок 1 — Заданная функция

1. Листинг программы

Ниже представлен листинг полной программы, реализованной на Java.

Листинг 1 — Листинг программы

import java.text.DecimalFormat;

import java.util.Scanner;

import java.util.function.DoubleUnaryOperator;

public class MaclaurinLab {

// n-й член ряда для аргумента x

private static DoubleUnaryOperator member(double x) {

return n -> Math.pow(-1, n) \* Math.pow(x, n);

}

// Вычисление суммы ряда с заданной точностью eps

private static double maclaurin(double x, double eps) {

double currentMember;

double sum = 0.0;

int n = 0;

do {

currentMember = member(x).applyAsDouble(n);

sum += currentMember;

n++;

} while (Math.abs(currentMember) >= eps);

return sum;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

DecimalFormat df = new DecimalFormat();

double x, eps;

df.setMaximumFractionDigits(8);

// ввод x

while (true) {

System.out.print("Введите x ∈ (-1; 1): ");

if (sc.hasNextDouble()) {

x = sc.nextDouble();

if (x > -1 && x < 1)

break;

else

System.err.println("Ошибка: x должен быть в диапазоне (-1; 1).");

} else {

System.err.println("Ошибка: нужно ввести число.");

sc.next(); // очищаем неверный ввод

}

}

// ввод eps

while (true) {

System.out.print("Введите точность eps ∈ (0; 1): ");

if (sc.hasNextDouble()) {

eps = sc.nextDouble();

if (eps > 0 && eps < 1)

break;

else

System.err.println("Ошибка: eps должно быть в диапазоне (0; 1).");

} else {

System.err.println("Ошибка: нужно ввести число.");

sc.next();

}

}

double maclaurinValue = maclaurin(x, eps);

double mathValue = 1.0 / (1.0 + x);

System.out.println("Значение через разложение в ряд Маклорена: " + df.format(maclaurinValue));

System.out.println("Вычисленное значение: " + df.format(mathValue));

System.out.println("Разница: " + df.format(Math.abs(maclaurinValue - mathValue)));

sc.close();

}

}

1. Результаты работы программы

На рис. 2–4 изображены результаты исполнения программы с различными входными данными.

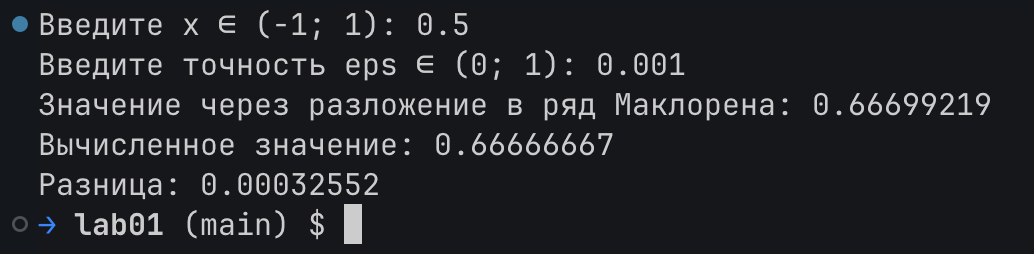


Рисунок 2 — Тестирование программы

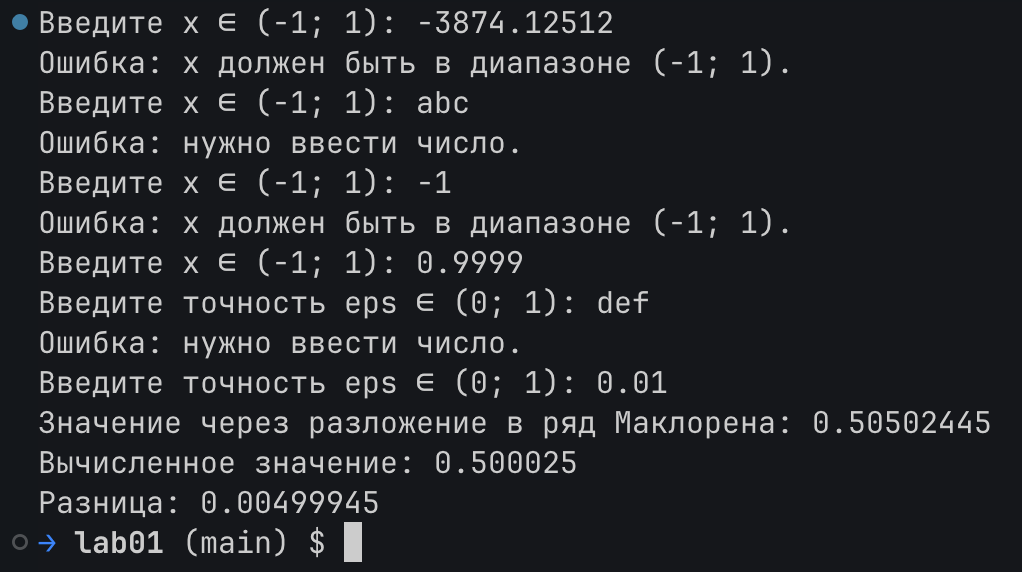


Рисунок 3 — Тестирование программы. Ввод некорректных значений

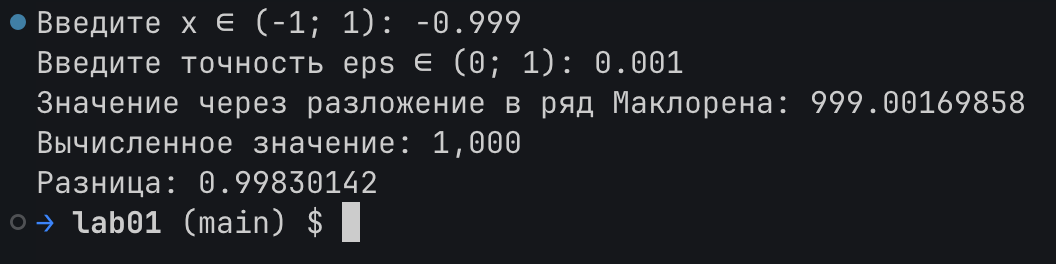


Рисунок 4 — Тестирование программы. Отрицательное значение *x*

1. Выводы

В ходе выполнения лабораторной работы:

* + - * была реализована программа на языке Java, вычисляющая значение функции через разложение в ряд Маклорена;
      * выполнено сравнение результатов с использованием стандартной функции Math, что подтвердило корректность реализации;
      * реализована проверка корректности ввода данных и обработка ошибок пользователя;
      * изучены и применены элементы функционального стиля программирования (лямбда-выражения, функциональные интерфейсы).

Программа работает стабильно и обеспечивает вычисления с заданной точностью, демонстрируя практическое применение функционального подхода в Java.