

**Московский Авиационный Институт**

Институт №3  
Программная инженерия  
Кафедра 304

**Отчёт по лабораторной работе  
по учебной дисциплине "Информационные технологии"  
на тему  
"Вычисление суммы бесконечного ряда"**

Выполнили: студенты группы МЗО-111Б-22

Яковченко Н.Р.

Деккер Сергей, Сергей, Серёга, извини

Москва, 2022 г.

# Содержание

<b>1</b>	<b>Задание</b>	<b>2</b>
<b>2</b>	<b>Схема алгоритма</b>	<b>3</b>
<b>3</b>	<b>Текст программы</b>	<b>4</b>
<b>4</b>	<b>Тесты</b>	<b>5</b>
4.1	Некорректные тесты . . . . .	5
4.2	Корректные тесты . . . . .	5

# 1. Задание

---

Кафедра 304

Курс: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Задание: **Вычисление суммы бесконечного числового ряда**

---

Вариант №11

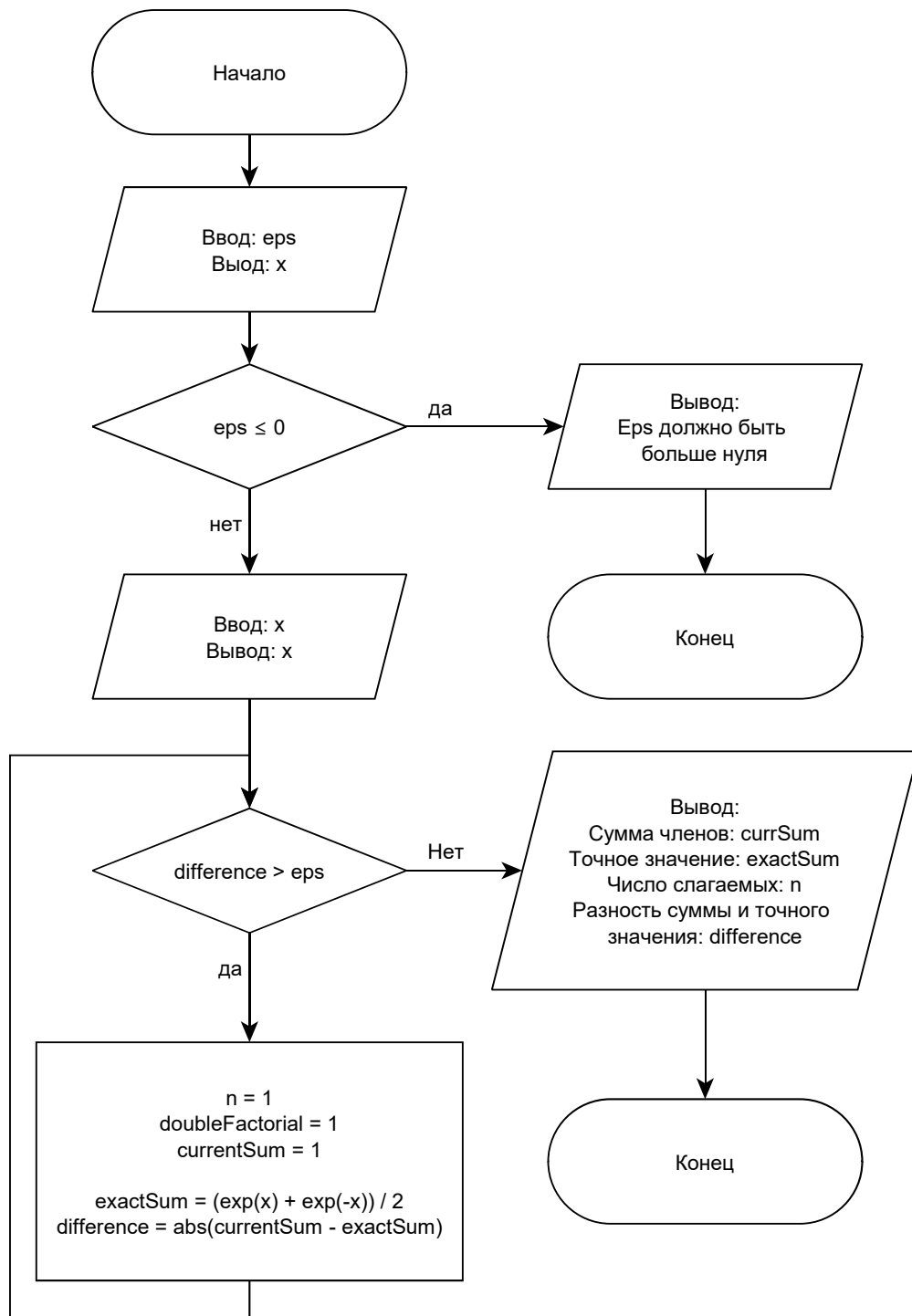
Определить с заданной точностью сумму членов бесконечного степенного ряда:

$$1 + \sum_{n=1} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots$$

Предусмотреть ввод точности и печать: количества просуммированных элементов, суммы, разности суммы и точного значения, которое равно:

$$\frac{e^x + e^{-x}}{2} = ch(x)$$

## 2. Схема алгоритма



### 3. Текст программы

```
#include <iostream>
#include <cmath>
using namespace std;

int main() // Начало программы
{
    setlocale(LC_ALL, "rus"); // Подключение русского языка

    int n; // Число итераций
    int doubleFactorial; // Текущее значение удвоенного факториала
    double difference; // Разность текущей и точной сумм
    double currentSum; // Сумма
    double exactSum; // Точное значение суммы
    double x; // Переменная
    double eps; // Точность
    double numerator; // Числитель

    cout << "Введите точность: ";
    cin >> eps; // Ввод точности
    cout << eps << endl; // Эхо-печать

    if (eps <= 0) // Валидация входящих данных
    {
        // Вывод сообщения об ошибке
        cout << "Заданная точность должна быть больше нуля" << endl;
        // Завершение работы программы в случае некорректности введенных данных
        return 1;
    }

    cout << "Введите X: ";
    cin >> x; // Ввод переменной x
    cout << x << endl; // Эхо-печать

    // Инициализация переменных
    n = 1;
    doubleFactorial = 1;
    currentSum = 1;
    numerator = 1;

    exactSum = (exp(x) + exp(-x)) / 2; // Подсчет точной суммы
    difference = abs(currentSum - exactSum); // Подсчет разности

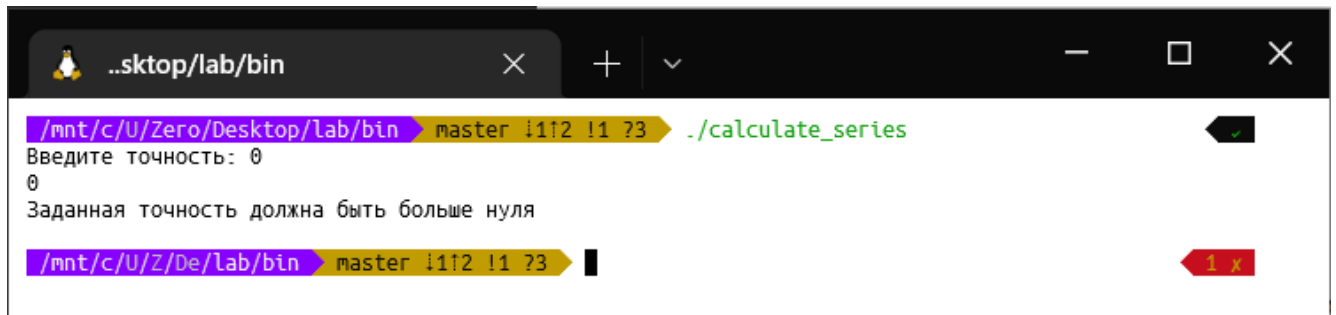
    while (difference > eps) // Начало цикла
    {
        // Вычисление суммы ряда
        doubleFactorial = doubleFactorial * (2 * n) * (2 * n - 1);
        numerator = numerator * x * x;
        currentSum += numerator / doubleFactorial;
        n += 1;
        difference = abs(currentSum - exactSum);
    } // Конец цикла

    // Вывод значений переменных
    cout << "Сумма членов: " << currentSum << endl;
    cout << "Точное значение: " << exactSum << endl;
    cout << "Число слагаемых: " << n << endl;
    cout << "Разность суммы и точного значения: " << difference << endl;
    return 0; // Возврат значения 0
}
```

## 4. Тесты

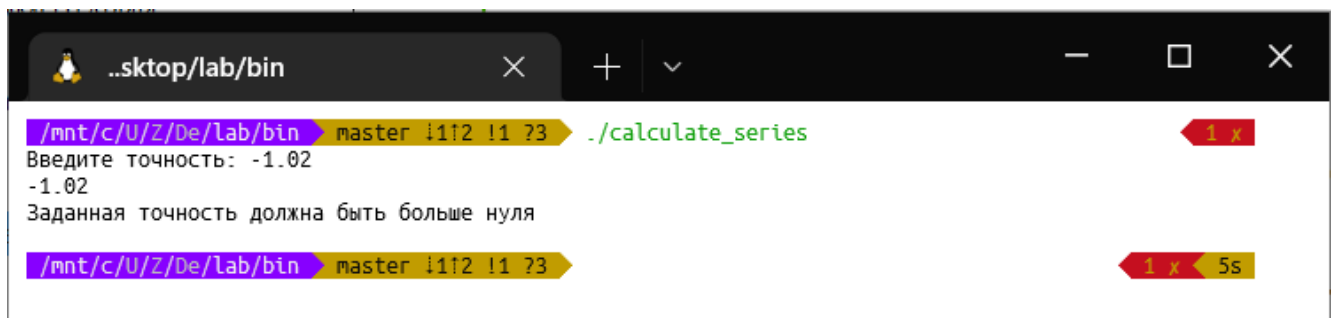
### 4.1. Некорректные тесты

**Тест 1** Цель: проверить работу программы на границе некорректной области  
Исходные данные:  $\epsilon_{ps} = 0$   
Ожидаемый результат: Заданная точность должна быть больше нуля  
Полученный результат:



Вывод по тесту: Полученный результат совпал с ожидаемым. Тест ошибок не выявил.

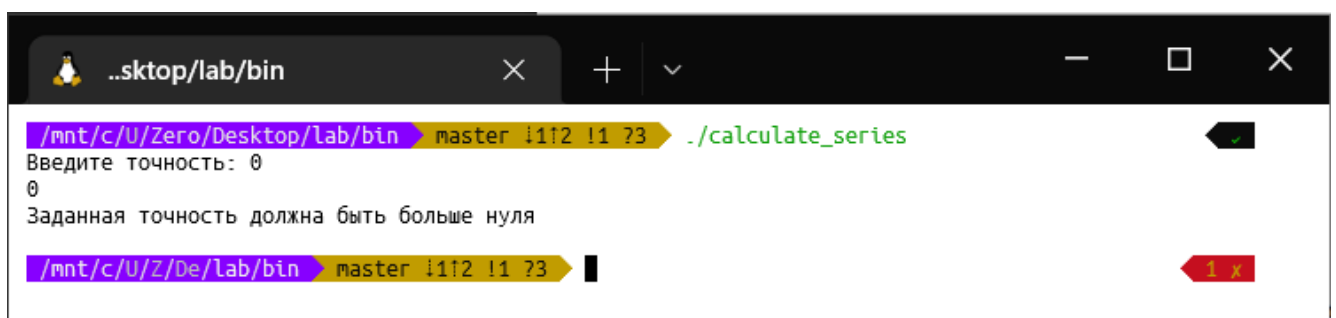
**Тест 2** Цель: проверить работу программы на границе некорректной области  
Исходные данные:  $\epsilon_{ps} = -1.02$   
Ожидаемый результат: Заданная точность должна быть больше нуля  
Полученный результат:



Вывод по тесту: Полученный результат совпал с ожидаемым. Тест ошибок не выявил.

### 4.2. Корректные тесты

**Тест 1** Цель: проверить работу программы на границе некорректной области  
Исходные данные:  $\epsilon_{ps} = 0$   
Ожидаемый результат: Заданная точность должна быть больше нуля  
Полученный результат:



Вывод по тесту: Полученный результат совпал с ожидаемым. Тест ошибок не выявил.