

Алгоритмизация и программирование

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 2. ФУНКЦИИ.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ.

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

$$x \in (-1, 1)$$

1. Напишите функцию для вычисления значения предложенной суммы в заданной точке (x) с указанной абсолютной погрешностью вычисления (`absError`) и максимальным числом слагаемых (`numberMax`).

Замечания:

- *Абсолютная погрешность (`absError`) представляет собой модуль первого отбрасываемого члена ряда суммы.*
- *Если номер слагаемого суммы больше `numberMax` и точность не достигнута, или x не попадает в заданный интервал $(-1,1)$, функция должна инициировать исключение.*
- *Нельзя использовать функции возведения в степень, вычисления факториала (см. другие задания).*

2. Напишите программу вывода таблицы вычисленных значений функции на указанном интервале (интервал, шаг и точность ввести из потока `cin`) и значений, полученных с использованием стандартных функций C++.

Для стандартных функций добавьте директиву препроцессора

```
#include <cmath>
```

Сделайте заголовок таблицы и «шапку».

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

```
double a = 1.0;      // числитель дроби
```

```
double b = 1.0;      // знаменатель дроби
```

```
double powerX = 1.0; // степень x
```

```
double term = 1.0;   // очередное слагаемое
```

```
double i = 1.0;      // номер слагаемого
```

```
double summa = 1.0;  // итоговая сумма
```

```
double a = 1.0;      // числитель дроби
double b = 1.0;      // знаменатель дроби
double powerX = 1.0; // степень x
double term = 1.0;   // очередное слагаемое
double i = 1.0;      // номер слагаемого
double summa = 1.0;  // итоговая сумма
```

```
    a *= (i + i - 1.0); // числитель
```

```
    b *= (i + i);       // знаменатель
```

```
    powerX *= -x;       // для чередования знаков слагаемых (-x)
```

```
    term = a / b * powerX;
```

```
    // term *= (- (i * 2.0 - 1.0) / (i * 2.0) * x);
```

```
    summa += term;
```

```
    i += 1.0;
```

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

```
while (i <= nTermsMax and abs(term) > absError) {
```

```
    a *= (i + i - 1.0);    // числитель
```

```
    b *= (i + i);          // знаменатель
```

```
    powerX *= -x;          // для чередования знаков слагаемых (-x)
```

```
    term = a / b * powerX;
```

```
    // term *= (- (i * 2.0 - 1.0) / (i * 2.0) * x);
```

```
    summa += term;
```

```
    i += 1.0;
```

```
}
```

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

```
while (i <= nTermsMax and abs(term) > absError) {
```

```
// . . .
```

```
}
```

```
if (i > nTermsMax) {
```

```
    throw EXCEEDED_NUMBER_OF_TERMS;
```

```
}
```

$$\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$$

```
double function(double x, int nTermsMax = 10, double absError = 1.0E-6)
{
    if (x <= -1.0 or x >= 1.0)          //???
        throw INVALID_RANGE;

    if (nTermsMax < N_TERMS_MIN)        // N_TERMS_MIN - const...
        throw INVALID_MAX_NUMBER;

    if ( (absError > ABS_ERROR or absError < 0.0 ) ) //ABS_ERROR - const
        throw INVALID_CALCULATION_ERROR;
```

Для стандартных функций и констант добавьте директиву препроцессора -

#include <cmath>

```
#define _USE_MATH_DEFINES // Всегда перед #include <cmath>
```

```
#include <cmath>
```

```
int main()
```

```
{
```

```
    double x = M_PI;
```

```
    // . . .
```

```
    return 0;
```

```
}
```

Структура проекта

1. **Файлы заголовков (тип .h)** - прототипы используемых функций
2. **Исходные файлы(тип .cpp)** - реализация функций. Имя функции совпадает с именем файла, начинается с маленькой буквы.
3. Функция **main** – операторы ввода/вывода, вызов функций, проверка данных.

```
// include "header.h"
```

