

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Программирование»
Тема: Итерационные вычисления суммы
с вещественными числами

Студент гр. 4335

Иванов Г.Д.

Руководитель

Калмычков В.А.

Санкт-Петербург

2025

Формулировка задания:

Написание программы с использованием циклов (итерационные алгоритмы с переходным коэффициентом q)

Дано:

Формула для слагаемого частичной суммы:

$$u_i(x) = \frac{(-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)}}{(2i+1)!}$$

Найти:

Значение частичной суммы $S(\varepsilon, x) = \sum_{i=1}^n u_i(x)$, где:

$$\begin{cases} |u_{n-1}(x)| > \varepsilon \\ |u_n(x)| \leq \varepsilon \\ n \leq 1000 \end{cases}$$

Способ решения:

Для решения задачи воспользуемся полученной в «Практическом задании №2» формулой с переходным коэффициентом:

$$u_{i+1}(x) = -\frac{2ix^2}{2i+3} * u_i(x)$$
$$u_1(x) = -\frac{x^3}{3}$$

Особенности компьютерной реализации:

При реализации программ версий 2 и 3 использовались различные варианты работы с массивами – объявление ограниченного массива в области локальной памяти (стэк/stack) и выделение динамического массива в области динамической памяти (куча/heap) соответственно. Разница между этими двумя подходами представлена в таблице 1:

Версия программы	2	3
Тип памяти	Локальная (стек / stack)	Динамическая (куча / heap)
Способ выделения	Объявление локальной переменной	Динамическое выделение через оператор new []
Количество памяти	Константное, определено на этапе компиляции	Динамическое, определено на этапе исполнения
Обращение к элементам массива	Индексный оператор обращения (p[i])	Разыменование указателя (* (p+i))
Способ очистки памяти	Выход за границы области видимости (выход из функции main)	Освобождение памяти оператором delete []

Таблица 1. Особенности подходов к использованию различных типов памяти

В версии программы 2 при каждом запуске будет задействовано одинаковое количество оперативной памяти, вне зависимости от введенных пользователем параметров. В версии 3 – будет выделено столько памяти, сколько требуется для конкретного вычисления с конкретными введенными параметрами.

Организация интерфейса пользователя:

Интерфейс программы состоит из нескольких макетов ввода/вывода, с небольшими отличиями между 2 и 3 версией программы.

1) МО1:

1.1) Версия 2:

=====

Лабораторная работа №2 по дисциплине "Программирование"

Автор: Иванов Григорий Денисович

Группа: 4335 (подгруппа 1)

Версия: 2

Период выполнения работы: 16.01.2025 - 17.01.2025

=====

Описание:

В этой программе будет проведен расчёт частичной суммы

$S_n(x) = \text{SUM}(u_1(x), u_2(x), \dots, u_n(x))$, номер последнего слагаемого определяется в соответствии со значением числа $\epsilon > 0$.

Формула для слагаемого $u_i(x)$:

$$u_i(x) = (-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)} / (2i+1)!$$

Упрощенная формула (формула с переходным коэффициентом):

$$u_{i+1}(x) = (-2x^{2*i}) / (2i+3) * u_i(x) \setminus n''$$

При вычислениях используются объявленные массивы с определенным числом элементов

1.2) Версия 3:

=====

Лабораторная работа №2 по дисциплине "Программирование"

Автор: Иванов Григорий Денисович

Группа: 4335 (подгруппа 1)

Версия: 3

Период выполнения работы: 16.01.2025 - 17.01.2025

=====

Описание:

В этой программе будет проведен расчёт частичной суммы

$S_n(x) = \text{SUM}(u_1(x), u_2(x), \dots, u_n(x))$, номер последнего слагаемого определяется в соответствии со значением числа $e > 0$.

Формула для слагаемого $u_i(x)$:

$$u_i(x) = (-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)} / (2i+1)!$$

Упрощенная формула (формула с переходным коэффициентом):

$$u_{i+1}(x) = (-2x^{2*i}) / (2i+3) * u_i(x)$$

При вычислении используются инструменты для динамического выделения необходимой памяти

2) MI2:

Введите X из диапазона (-1, 1): <вещественное число в обычном или научном представлении>

3) MI3:

Введите e из диапазона (0, 1): <вещественное число в обычном или научном представлении>

3) MO4:

i	$u_i(x)$	$s_i(x)$
1 <число с точностью до 10 знаков>	<число с точностью до 10 знаков>	
2 <число с точностью до 10 знаков>	<число с точностью до 10 знаков>	
...		
...		
n-1 <число с точностью до 10 знаков>	<число с точностью до 10 знаков>	
n <число с точностью до 10 знаков>	<число с точностью до 10 знаков>	

4) MO5:

Результат: <число с точностью до 10 знаков>

Для реализации макетов использовались такие модификаторы потока вывода как:

- 1) `std::setw` – для указания ширины выводимого слова

- 2) `std::setprecision` – для указания максимального количества знаков после запятой

Описание алгоритма работы программы:

Версия 2:

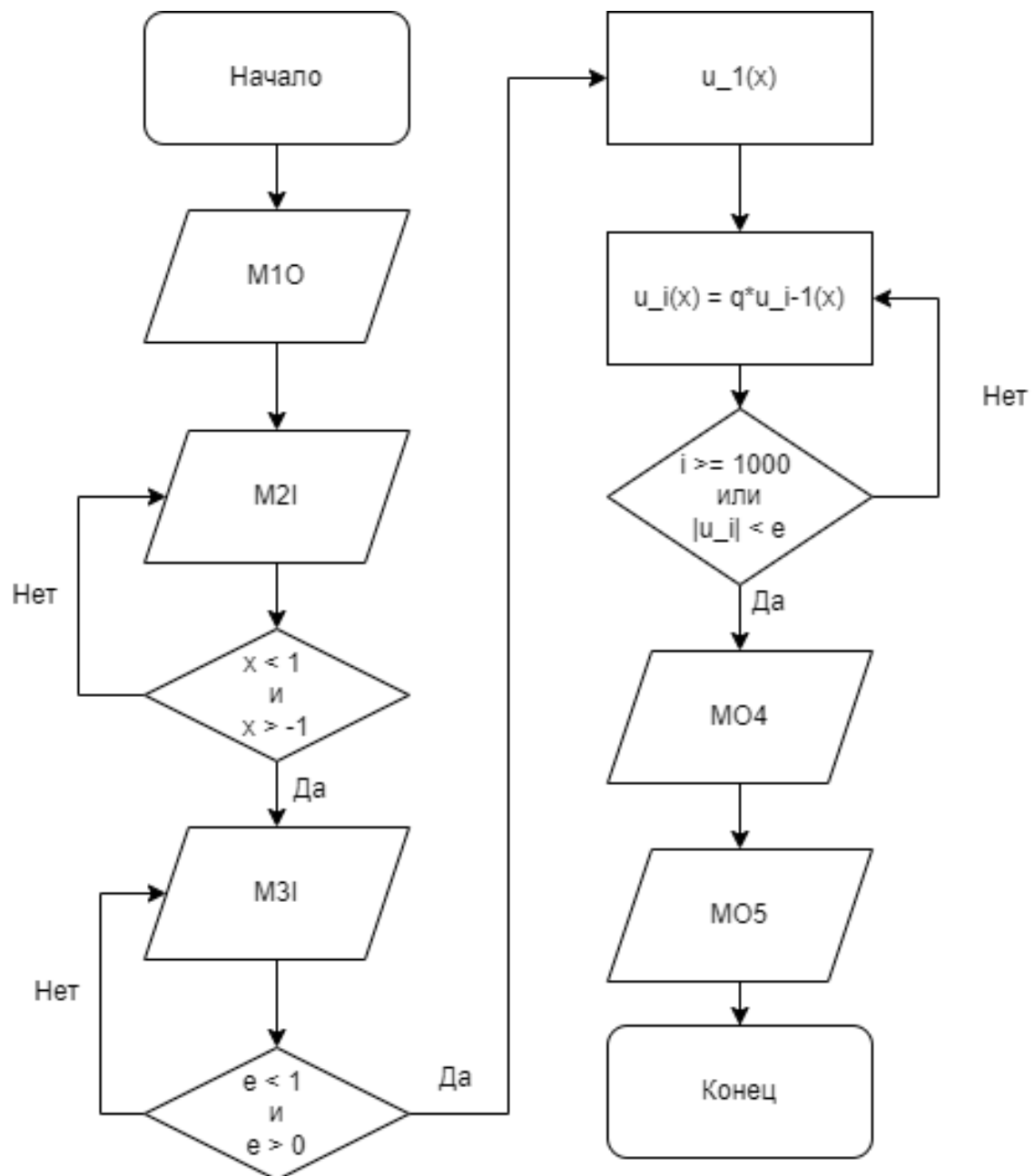


Рисунок 1. Алгоритм работы программы версии 2.

Версия 3:

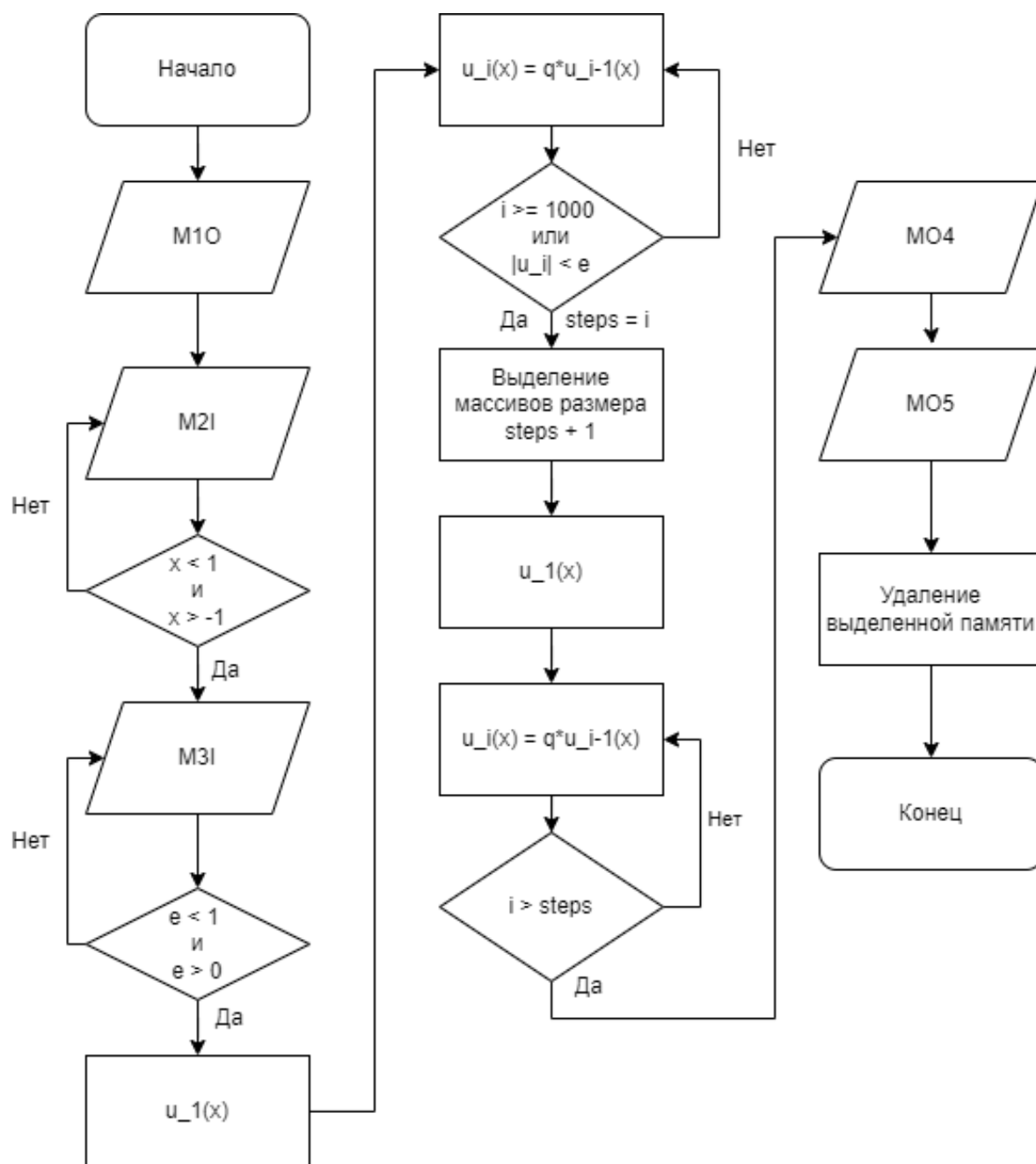


Рисунок 2. Алгоритм работы программы версии 3.

Текст программы:

Версия 2:

```
#include <iostream>
#include <iomanip>

namespace {
const char credits [] = "=====\\n"
    "Лабораторная работа №2 по дисциплине \\\"Программирование\\\"\\n"
    "Автор: Иванов Григорий Денисович\\n"
    "Группа: 4335 (подгруппа 1)\\n"
    "Версия: 2\\n"
    "Период выполнения работы: 16.01.2025 - 17.01.2025\\n"
    "=====\\n";

const char description [] = "Описание:\\n"
    "В этой программе будет проведен расчёт частичной суммы\\n"
    "S_n(x) = SUM (u_1(x), u_2(x), ... u_n(x)), номер последнего слагаемого\\n"
    "определяется в соответствии со значением числа e>0.\\n"
    "Формула для слагаемого u_i(x):\\n"
    "  u_i(x)=(-1)^i * 2^(2i-1) * (i-1)! * i! * x^(2i+1) / (2i+1)!\\n"
    "Упрощенная формула (формула с переходным коэффициентом):\\n"
    "  u_{i+1}(x) = (-2x^2*i) / (2i+3) * u_i(x)\\n"
    "При вычислениях используются объявленные массивы c\\n"
    "определенным числом элементов\\n";
}

int main()
{
    double x, ep;
    double tmps[1001], sums[1001];
    unsigned int i {1}, steps;

    std::cout << credits << std::endl;
    std::cout << description << std::endl;

    do
    {
        std::cout << "Введите X из диапазона (-1, 1):";
        std::cin >> x;
    }while(x <= -1 || x >= 1);

    do
```

```

{
    std::cout << "Введите e из диапазона (0, 1):";
    std::cin >> ep;
}while(ep <= 0 || ep >= 1);

tmps[0] = -x*x*x / 3;
sums[0] = tmps[0];

for(;;i++)
{
    tmps[i] = tmps[i-1] * -2 * x * x * i;
    tmps[i] /= 2 * i + 3;
    sums[i] = sums[i-1] + tmps[i];

    if(i >= 1000 || (tmps[i] < 0? -tmps[i] : tmps[i]) < ep)
        break;
}
steps = i;

std::cout << "|" << std::setw(10) << "i"    //"Шаг"
    << "|" << std::setw(20) << "u_i(x)" //"Слагаемое"
    << "|" << std::setw(20) << "s_i(x)" //"Сумма"
    << "|" << std::endl;

for(i = 0; i < steps; i++)
{
    std::cout << "|" << std::setw(10) << i + 1
        << "|" << std::setw(20) << std::setprecision(10) << tmps[i]
        << "|" << std::setw(20) << std::setprecision(10) << sums[i]
        << "|" << std::endl;
}
std::cout << "Результат: " << sums[steps-1] << std::endl;
return 0;
}

```

Версия 3:

```

#include <iostream>
#include <iomanip>

namespace {

```



```

const char credits [] = "=====\\n"
    "Лабораторная работа №2 по дисциплине \\\"Программирование\\\"\\n"
    "Автор: Иванов Григорий Денисович\\n"
    "Группа: 4335 (подгруппа 1)\\n"
    "Версия: 3\\n"
    "Период выполнения работы: 16.01.2025 - 17.01.2025\\n"
    "=====\\n";

const char description [] = "Описание:\\n"
    "В этой программе будет проведен расчёт частичной суммы\\n"
    "S_n(x) = SUM (u_1(x), u_2(x), ... u_n(x)), номер последнего слагаемого\\n"
    "определяется в соответствии со значением числа e>0.\\n"
    "Формула для слагаемого u_i(x):\\n"
    "  u_i(x)=(-1)^i * 2^(2i-1) * (i-1)! * i! * x^(2i+1) / (2i+1)!\\n"
    "Упрощенная формула (формула с переходным коэффициентом):\\n"
    "  u_{i+1}(x) = (-2x^2*i) / (2i+3) * u_i(x)\\n"
    "При вычислении используются инструменты для динамического\\n"
    "  выделения необходимой памяти\\n";
}

int main()
{
    double x, tmp, ep;
    double * tmps { nullptr }, * sums { nullptr };
    unsigned int i {1}, steps;

    std::cout << credits << std::endl;
    std::cout << description << std::endl;

    do
    {
        std::cout << "Введите X из диапазона (-1, 1):";
        std::cin >> x;
    }while(x <= -1 || x >= 1);

    do
    {
        std::cout << "Введите e из диапазона (0, 1):";
        std::cin >> ep;
    }while(ep <= 0 || ep >= 1);

    tmp = -x*x*x / 3;

```

```

for(;;i++)
{
    tmp *= -2 * x * x * i;
    tmp /= 2 * i + 3;

    if(i >= 1000 || (tmp < 0? -tmp : tmp) < ep)
        break;
}
steps = i;
tmps = new double[steps + 1];
sums = new double[steps + 1];

*tmps = -x*x*x / 3;
*sums = *tmps;

for(i = 1; i <= steps; i++)
{
    *(tmps + i) = *(tmps + i - 1) * -2 * x * x * i;
    *(tmps + i) /= 2 * i + 3;
    *(sums + i) = *(sums + i - 1) + *(tmps + i);
}

std::cout << "|" << std::setw(10) << "i"    //"Шаг"
    << "|" << std::setw(20) << "u_i(x)" //"Слагаемое"
    << "|" << std::setw(20) << "s_i(x)" //"Сумма"
    << "|" << std::endl;

for(i = 0; i < steps; i++)
{
    std::cout << "|" << std::setw(10) << i + 1
        << "|" << std::setw(20) << std::setprecision(10) << *(tmps + i)
        << "|" << std::setw(20) << std::setprecision(10) << *(sums + i)
        << "|" << std::endl;
}

std::cout << "Результат: " << *(sums + steps - 1) << std::endl;
delete [] tmps;
delete [] sums;

return 0;
}

```

Результат работы программы:

Версия 2:

```
=====
Лабораторная работа №2 по дисциплине "Программирование"
Автор: Иванов Григорий Денисович
Группа: 4335 (подгруппа 1)
Версия: 2
Период выполнения работы: 16.01.2025 - 17.01.2025
=====
Описание:
В этой программе будет проведен расчёт частичной суммы
 $S_n(x) = \sum (u_1(x), u_2(x), \dots u_n(x))$ , номер последнего слагаемого
определяется в соответствии со значением числа  $\epsilon > 0$ .
Формула для слагаемого  $u_i(x)$ :
 $u_i(x) = (-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)} / (2i+1)!$ 
Упрощенная формула (формула с переходным коэффициентом):
 $u_{i+1}(x) = (-2x^{2i}) / (2i+3) * u_i(x)$ 
При вычислениях используются объявленные массивы с
определенным числом элементов

Введите X из диапазона (-1, 1):0.98
Введите e из диапазона (0, 1):0.001
| i | u_i(x) | s_i(x) |
|---|---|---|
| 1 | -0.3137306667 | -0.3137306667 |
| 2 | 0.1205227729 | -0.1932078938 |
| 3 | -0.06614289777 | -0.2593507915 |
| 4 | 0.04234909268 | -0.2170016989 |
| 5 | -0.02957968626 | -0.2465813851 |
| 6 | 0.02185256207 | -0.224728823 |
| 7 | -0.01678976049 | -0.2415185835 |
| 8 | 0.01327931786 | -0.2282392657 |
| 9 | -0.01073975315 | -0.2389790188 |
| 10 | 0.008840964797 | -0.230138054 |
| 11 | -0.007383358775 | -0.2375214128 |
| 12 | 0.006240060435 | -0.2312813524 |
| 13 | -0.00532707026 | -0.2366084226 |
| 14 | 0.004586864662 | -0.232021558 |
| 15 | -0.003978912742 | -0.2360004707 |
| 16 | 0.003473952543 | -0.2325265182 |
| 17 | -0.003050408249 | -0.2355769264 |
| 18 | 0.002692075968 | -0.2328848504 |
| 19 | -0.00238658747 | -0.2352714379 |
| 20 | 0.002124365538 | -0.2331470724 |
| 21 | -0.001897898291 | -0.2350449707 |
| 22 | 0.001701225417 | -0.2333437453 |
| 23 | -0.001529568153 | -0.2348733134 |
| 24 | 0.001379058647 | -0.2334942548 |
| 25 | -0.001246539223 | -0.234740794 |
| 26 | 0.001129411575 | -0.2336113824 |
| 27 | -0.001025522138 | -0.2346369045 |
Результат: -0.2346369045
press RETURN to close this window
```

Версия 3:

```
=====
Лабораторная работа №2 по дисциплине "Программирование"
Автор: Иванов Григорий Денисович
Группа: 4335 (подгруппа 1)
Версия: 3
Период выполнения работы: 16.01.2025 - 17.01.2025
=====
Описание:
В этой программе будет проведен расчёт частичной суммы
 $S_n(x) = \sum (u_1(x), u_2(x), \dots, u_n(x))$ , номер последнего слагаемого
определяется в соответствии со значением числа  $\epsilon > 0$ .
Формула для слагаемого  $u_i(x)$ :
 $u_i(x) = (-1)^i * 2^{(2i-1)} * (i-1)! * i! * x^{(2i+1)} / (2i+1)!$ 
Упрощенная формула (формула с переходным коэффициентом):
 $u_{i+1}(x) = (-2x^{2*i}) / (2i+3) * u_i(x)$ 
При вычислении используются инструменты для динамического
выделения необходимой памяти

Введите X из диапазона (-1, 1):0.5
Введите e из диапазона (0, 1):0.0001
| i | u_i(x) | s_i(x) |
| 1 | -0.04166666667 | -0.04166666667 |
| 2 | 0.004166666667 | -0.0375 |
| 3 | -0.0005952380952 | -0.0380952381 |
Результат: -0.0380952381
```

Выводы:

В этой лабораторной работе мы научились работать с различными видами циклов, освоили два разных способа работы с массивами (локальное определение и динамическое выделение), а также воспользовались модификаторами стандартного потока вывода c++. В ходе работы мы провели реализацию итерационного алгоритма нахождения частичной суммы математического выражения.