

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Программирование»
Тема: Реализация итерационного алгоритма

Студент гр. 0335

Коваленко С.К.

Преподаватель

Калмычков В.А.

Санкт-Петербург

2021

Цель работы: продолжение освоения программирования циклов при реализации итерационных алгоритмов с использованием массивов для хранения рассчитываемых значений.

Дано:

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

Нахождение расчетной формулы:

$$\frac{u_{i+1}}{u_i} = \frac{4^{i+1} \cdot (1+i)^2}{(i+2)!(i+3)} \cdot x^{i+1} \cdot \frac{(i+2)!}{4^i \cdot i^2 \cdot x^i} = \frac{4x(1+i)^2}{(i+3) \cdot i^2}$$

Расчетная формула:

$$u_{i+1} = \frac{4x(1+i)^2}{(i+3) \cdot i^2} \cdot u_i$$

Найти: минимальное значение $n > 0$, для которого очередное слагаемое по модулю не меньше ε большого

Контрольный пример:

При $x=4$ и $\varepsilon=5.02177e44$

$$u_1 = \frac{4^1 \cdot 1^2}{(1+2)!} \cdot 4^1 = 2.666667e00$$

$$u_2 = 3.072000e02$$

$$u_3 = 1.179648e05$$

$$u_4 = 1.078535e08$$

$$u_5 = 1.941364e11$$

$$u_6 = 6.088116e14$$

$$u_7 = 3.054773e18$$

$$u_8 = 2.303410e22$$

$$u_9 = 2.487683e26$$

$$u_{10} = 3.704734e30$$

$$u_{11} = 7.377289e34$$

$$u_{12} = 1.915026e39$$

$$u_{13} = 6.343333e43$$

$$u_{14} = 2.632856e48$$

Частичная сумма = $2.63291981e+48$ для 14 элементов последовательности

Организация интерфейса пользователя:

При запуске:

- Вывод формул
- Запрос x
- Запрос ϵ ps большое
- Если ϵ ps не удовлетворяет условию, повторный запрос в сумме 3-ех попыток
- Если после 3 попытки ϵ ps не удовлетворяет, выход из программы
- Ведется подсчет и выводится результаты в шаблоне:
| номер итерации | очередное слагаемое | частичная сумма |
- Если очередное слагаемое выходит за пределы double диапазона программа прекращает работу
- Вывод результатов

Использовалось:

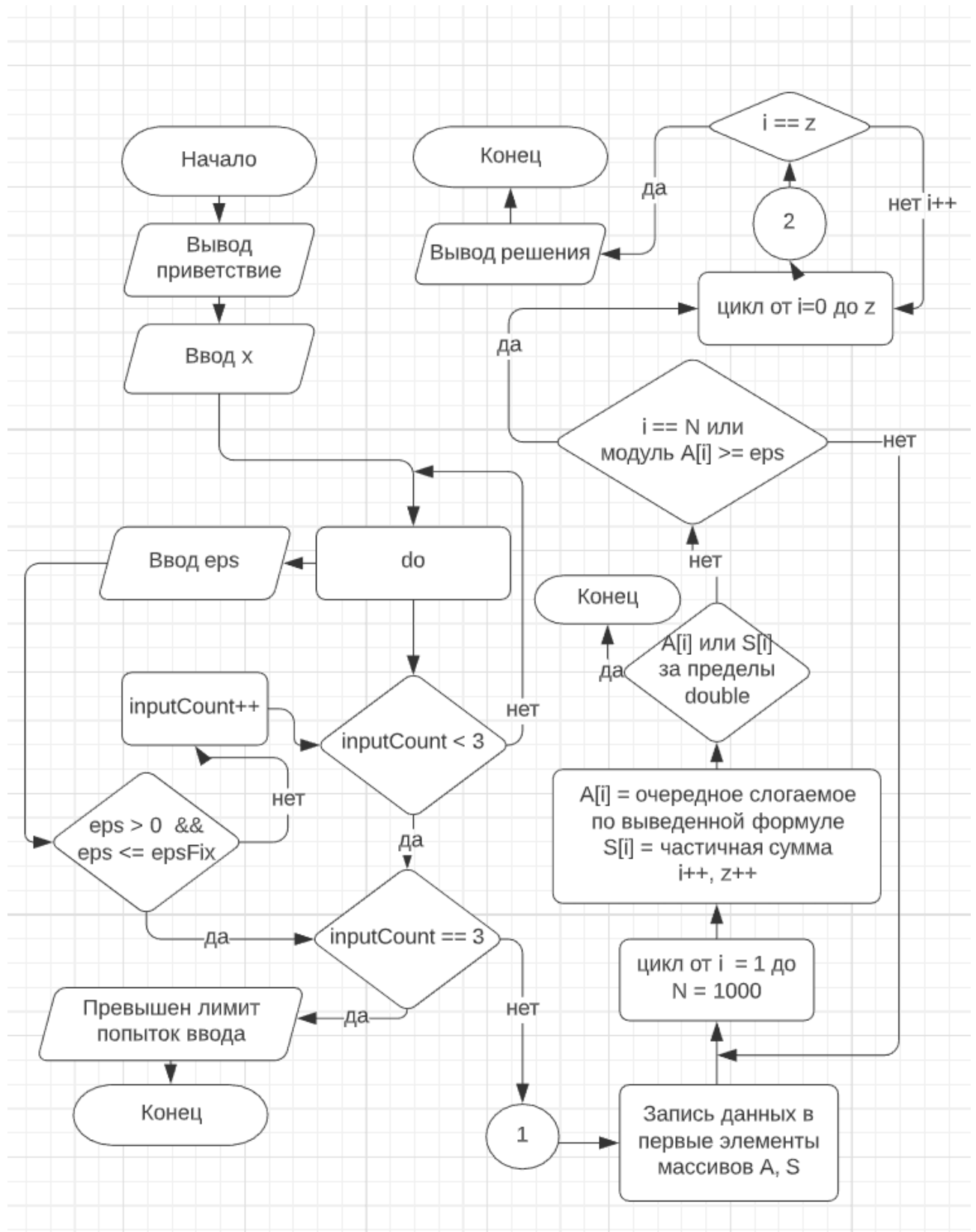
- Функции `cout`, `cin`(библиотека `iostream`)
- Функции `fabs`, `isinf`(библиотека `math.h`)
- Манипуляторы `setw`, `setprecision`, `showpoint`, `showpos`, `scientific`
- Циклы `do while`, `for`
- Условные операторы `if`, `else`
- Массивы
- Указатели

Особенности реализации:

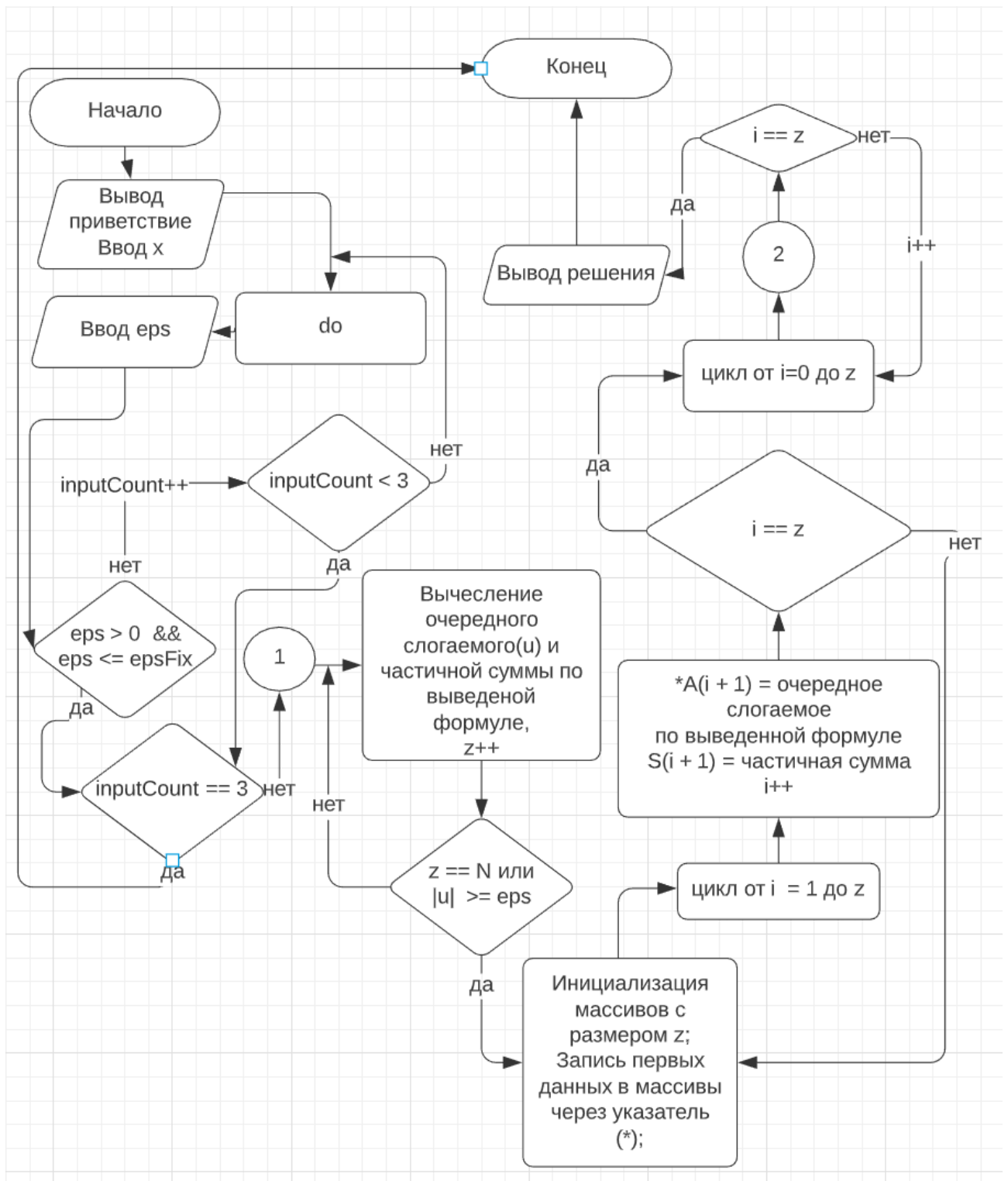
В программе №2 использовались массивы. Сначала происходит их инициализация и заполнение при помощи цикла, затем вывод содержимого на консоль при помощи цикла.

В программе №3 использовались динамические массивы с указателем на первый элемент. Сначала происходит нахождение размера циклов, затем заполнение, после вывод на консоль, при помощи циклов.

Блок-схема программы №2

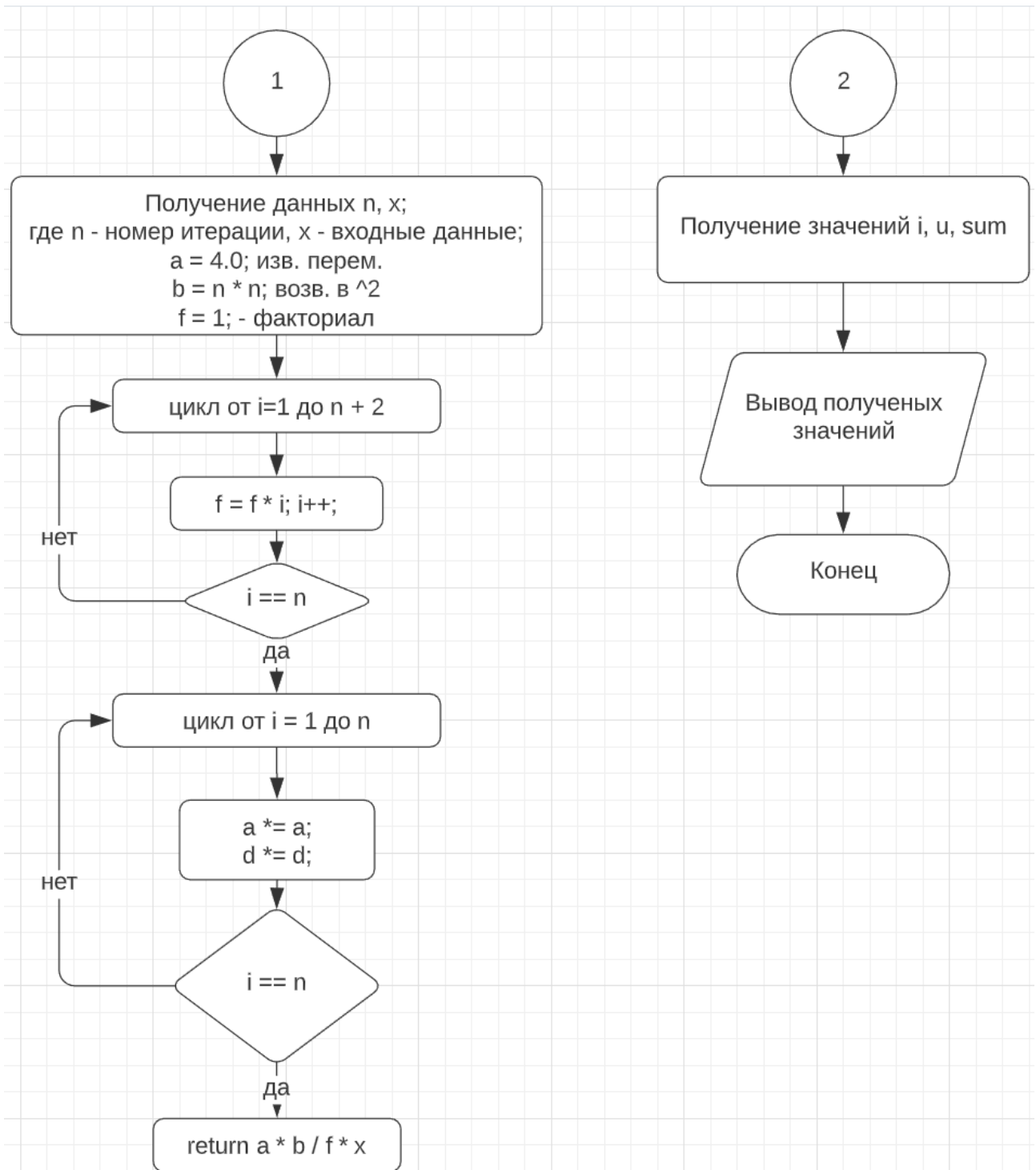


Блок-схема программы №3



Блок-схемы функций:

1. `double firstU(int n, double x)`
2. `void printIUSum(int i, double u, double sum)`



Исходный код программы №2

```
/**
 * Определить минимальное значение  $n > 0$ ,
 * для которого очередное слагаемое по модулю превышает  $\epsilon$  большое
 * при нахождении результата согласно формуле
 *
 * @author Коваленко Сергей группа 0335
 * @version 2
 * Дата начала разработки: 20.01.2021    Дата сдачи программы: 22.01.2021
 */
#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

// Метод для вычисления  $U_n$ 
double firstU(int n, double x){
    double a = 4.0;
    double b = n * n;
    int f = 1;
    for(int i = 1; i <= n + 2.0; i++) f *= i;
    double d = x;
    for(int i = 1; i < n; i++){
        a *= a;
        d *= d;
    }
    return a * b / f * d;
}

// Метод для печати в консоль значений  $n$  и  $sum$ 
void printNUSum(double n, double u, double sum){
    int i = n;
    cout << "|" << setw(5) << i << " |"
         << setw(15) << setprecision(7) << showpoint << showpos << scientific <<
    setw(14) << setprecision(6) << u << "|"
         << setw(15) << setprecision(7) << showpoint << showpos << scientific <<
    setw(14) << setprecision(50) << sum << "|" << endl;
}

//диапазон  $\epsilon$  до
const double epsFix = 5.021775e+44;
//размер массивов
const int N = 1000;
int main() {
    setlocale(LC_STYPE, "rus"); // вызов функции настройки локали

    double A[N]; //для хранения очередного слагаемого
    double S[N]; //для хранения очередной частичной суммы
    int z = 0; // количество используемых элементов в массиве
    int n = 1;
    double eps; // входные данные. Погрешность, с которой будет вычисляться
    значение ряда
    double x; // входные данные
    double sum; // частичная сумма
    double u; // очередное слагаемое

    cout << "Автор: Коваленко Сергей 0335\n"
         << "Определить минимальное значение  $n > 0$ , для которого очередное\n"
         << "слагаемое по модулю не меньше  $\epsilon$  большого при нахождении
    результата согласно формуле:" << endl;
    //Вывод формулы
    cout << " n\n"
         << "--- i      2\n"
         << "\\      4 * i      i\n";
}
```

```

        << "/" << "----- * x\n"
        << "---- (i + 2)!\n"
        << "n=1" << endl;
cout << "Расчетная формула:\n"
    << "
        << "Ui+1      4 * x * (1 + i)  2\n"
        << "----- = ----- * Ui \n"
        << " Ui      (i + 3) * i\n" << endl;
cout << "Введите x:" << endl;
cin >> x;
int inputCount = 0;
do {
    cout << "Введите eps в диапазоне 0 < eps <= " << epsFix << ":" << endl;
    cin >> eps;
    if ((eps > 0) && (eps <= epsFix)){
        break;
    } else {
        cout << "eps не соответствует" << endl;
        inputCount++;
    }
} while (inputCount < 3);
if(inputCount == 3){
    cout << "Превышен лимит попыток ввода" << endl;
    return -1;
}

A[0] = firstU(n, x); //Вычисление U1 и сохранение в массив A
S[0] = A[0]; //сохранение первой частичной суммы в массив S
z++;
//запись в массив очередных слогаемых и частичных сумм
for(int j = 1; j < N; j++){
    int i = j + 1;
    A[j] = 4.0 * x * (1.0 + i) * (1.0 + i) / (i + 3) * i * i * A[j - 1];
    S[j] = S[j - 1] + A[j];
    if(isinf(A[j])) {
        cout << "Выход за пределы диапазона double";
        return -1;
    }
    if(isinf(A[j]) ) {
        cout << "Выход за пределы диапазона double";
        return -1;
    }
    z++;
    if(fabs(A[j]) >= eps) {
        break;
    }
}
cout << "| i | u | S"
|" << endl; //шапка таблицы
// вывод на консоль таблицы
for(int i = 0; i < z; i++){
    printNUSum(i + 1, A[i], S[i]);
}

cout << "-----\n"
    << "Частичная сумма = " << noshowpos << setprecision(8) << S[z - 1] <<
" для " << z << " элементов последовательности" << endl;
return 0;
}

```

Исходный код программы №3

```
/**
 * Определить минимальное значение  $n > 0$ ,
 * для которого очередное слагаемое по модулю превышает  $\epsilon$  большое
 * при нахождении результата согласно формуле
 *
 * @author Коваленко Сергей группа 0335
 * @version 3
 * Дата начала разработки: 20.01.2021    Дата сдачи программы: 22.01.2021
 */
#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

// Метод для вычисления  $U_n$ 
double firstU(int n, double x){
    double a = 4.0;
    double b = n * n;
    int f = 1;
    for(int i = 1; i <= n + 2.0; i++) f *= i;
    double d = x;
    for(int i = 1; i < n; i++){
        a *= a;
        d *= d;
    }
    return a * b / f * d;
}

// Метод для печати в консоль значений  $n$  и  $sum$ 
void printNUSum(double n, double u, double sum){
    int i = n;
    cout << "|" << setw(5) << i << " |"
         << setw(15) << setprecision(7) << showpoint << showpos << scientific <<
    setw(14) << setprecision(6) << u << "|"
         << setw(15) << setprecision(7) << showpoint << showpos << scientific <<
    setw(14) << setprecision(50) << sum << "|" << endl;
}

//диапазон  $\epsilon$  до
const double epsFix = 5.021775e+44;
//размер массивов
const int N = 1000;
int main() {
    setlocale(LC_CTYPE, "rus"); // вызов функции настройки локали

    double *pA; //для хранения очередного слагаемого
    double *pS; //для хранения очередной частичной суммы
    int z = 0; // количество используемых элементов в массиве
    int n = 1;
    double eps; // входные данные. Погрешность, с которой будет вычисляться
    значение ряда
    double x; // входные данные
    double sum; // частичная сумма
    double u; // очередное слагаемое

    cout << "Автор: Коваленко Сергей 0335\n"
         << "Определить минимальное значение  $n > 0$ , для которого очередное\n"
         << "слагаемое по модулю не меньше  $\epsilon$  большого при нахождении
    результата согласно формуле:" << endl;
    //Вывод формулы
    cout << " n\n"
         << "--- i      2\n"
         << "\\      4 * i      i\n";
}
```

```

        << "/" << "----- * x\n"
        << "---- (i + 2)!\n"
        << "n=1" << endl;
cout << "Расчетная формула:\n"
    << "
        << "Ui+1      4 * x * (1 + i)  2\n"
        << "----- = ----- * Ui \n"
        << " Ui      (i + 3) * i\n" << endl;
cout << "Введите x:" << endl;
cin >> x;
int inputCount = 0;
do {
    cout << "Введите eps в диапазоне 0 < eps <= " << epsFix << ":" << endl;
    cin >> eps;
    if ((eps > 0) && (eps <= epsFix)){
        break;
    } else {
        cout << "eps не соответствует" << endl;
        inputCount++;
    }
} while (inputCount < 3);
if(inputCount == 3){
    cout << "Превышен лимит попыток ввода" << endl;
    return -1;
}

u = firstU(++z, x); //Вычисление U1
sum = u; //частичная сумма

while (fabs(u) < eps && z < N){
    z++;
    u *= 4.0 * x * (1.0 + z) * (1.0 + z) / (z + 3.0) * z * z;
    sum += u;
}

pA = new double[z];
pS = new double[z];

*pA = firstU(n, x);
*pS = *pA;
for(int j = 1, i = 2; j < z; j++, i++){
    *(pA + j) = 4.0 * x * (1.0 + i) * (1.0 + i) / (i + 3.0) * i * i * *(pA +
j - 1);
    *(pS + j) = *(pS + j - 1) + *(pA + j);
}
cout << " | i | u | S
|" << endl; //шапка таблицы
// вывод на консоль таблицы
for(int i = 0; i < z; i++){
    printNUSum(i + 1, *(pA + i), *(pS + i));
}

cout << "-----\n"
    << "Частичная сумма = " << noshowpos << setprecision(8) << *(pS + z -
1) << " для " << z << " элементов последовательности" << endl;
delete []pA;
delete []pS;
return 0;
}

```

Результат Программ №2 и №3

Автор: Коваленко Сергей 0335

Определить минимальное значение $n > 0$, для которого очередное

слагаемое по модулю не меньше eps большого при нахождении результата согласно формуле:

$$\frac{\sqrt[4]{i^2}}{(i+2)!} * x$$

n=1

Расчетная формула:

$$U_{i+1} = \frac{4 * x * (1+i)^2}{(i+3) * i} * U_i$$

Введите x:

4

Введите eps в диапазоне $0 < \text{eps} \leq 5.02177\text{e}+44$:

5.02177e+44

i	u	S
1	+2.666667e+00	+2.66666666666666651863693004997912794351577758789062e+00
+2	+3.072000e+02	+3.098666666666666674245789181441068649291992187500000e+02
+3	+1.179648e+05	+1.18274666666666665696538984775543212890625000000000e+05
+4	+1.078535e+08	+1.079718060952380895614624023437500000000000000000e+08
+5	+1.941364e+11	+1.9424432837752380371093750000000000000000000000e+11
+6	+6.088116e+14	+6.090058585363775000000000000000000000000000000e+14
+7	+3.054773e+18	+3.055382161308597248000000000000000000000000000e+18
+8	+2.303410e+22	+2.303715580085674390323200000000000000000000000e+22
+9	+2.487683e+26	+2.487913216777115665977835520000000000000000000e+26
+10	+3.704734e+30	+3.704982628509518171937439744000000000000000000e+30
+11	+7.377289e+34	+7.37765991302410008419330193893621760000000000000e+34
+12	+1.915026e+39	+1.915100072040514544861415294550214180864000000000e+39
+13	+6.343333e+43	+6.3435246110272452965496310479443945578299392000000e+43
+14	+2.632856e+48	+2.63291980941066342160516244027873030359537313382400e+48

Частичная сумма = 2.63291981e+48 для 14 элементов последовательности

Выводы:

В ходе лабораторной работы я освоил массивы и динамические массивы.

Научился работать с указателем и контролировать потребляемую память.