**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №**

по дисциплине «Основы алгоритмизации и программирования»

**Тема:**

Решение нелинейных уравнений 3 - мя способами.

Выполнила:

Студентка группы ИВТ-19-1б

Еремеева Полина Алексеевна

Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2020

**Постановка задачи**

Решить нелинейные уравнения тремя способами:

1. Метод итераций
2. Метод Ньютона
3. Метод половинного деления

**Анализ задачи**

1. Метод итераций

Есть функция. Хо - любая точка на отрезке (а:б). Присваиваем Х1 = f(X0), a f(X0) = Yxo

Х1 = f(Xo) = Yxo

Х2 = f(X1) = Yx1

Х3 = f(X2) = Yx2 …

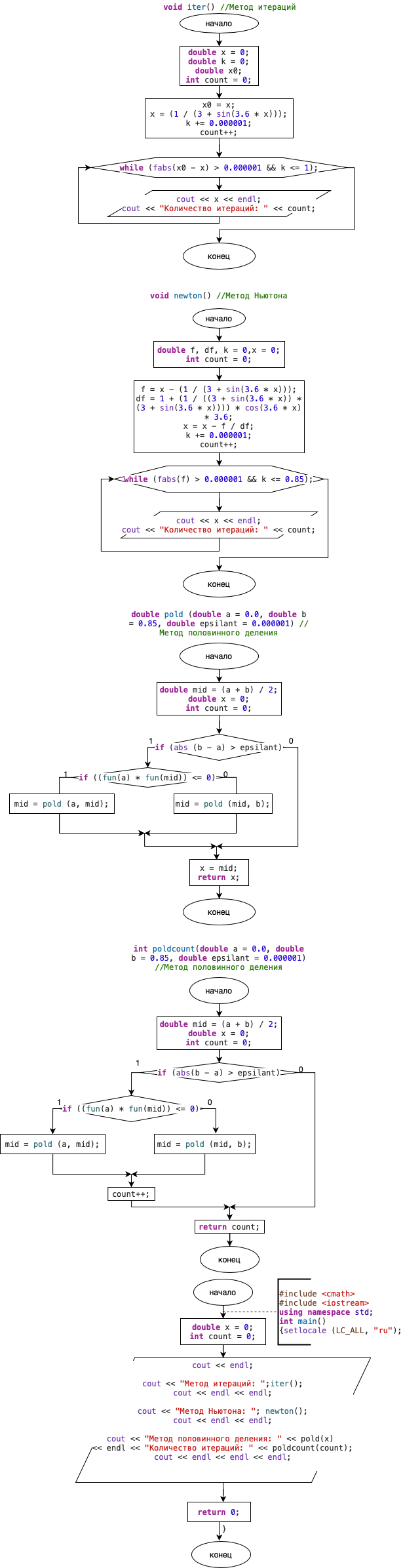
И так далее пока |Х - Хо| > E.

1. Метод Ньютона

Есть функция. Известен интервал (а:б), на которой функция имеет корень. На интервале (а:б) случайным образом выбриваться точка Хо. В этой точке к функции строится касательная и точка пересечения касательной с осью Ох дает следующую точку приближённого корня. Задана некоторая точность Е - эпсила Е = 10 ^ -6. Разность между двумя соседними корнями по абсолютной величине, должна быть меньше чем |Х - Хо| > E. Это означает, что две точки Хо и Х1 попадают в одну и туже точку. То есть любую из них мы можем считать корнем.

3. Метод половинного деления

Есть функция. Хо - делит отрезок (а:б) пополам и отбрасывает ненужный отрезок пока |Х - Хо| > E. Отбрасывание идет по признаку сравнения знаков. Если знаки на концах интервала одинаковые, мы отбрасываем его.

**Блок - схема**

**Код**

#include <cmath>

#include <iostream>

**using** **namespace** std;

**void** iter() //Метод итераций

{

**double** x = 0;

**double** k = 0;

**double** x0;

**int** count = 0;

**do**

{

x0 = x;

x = (1 / (3 + sin(3.6 \* x)));

k += 0.000001;

count++;

}

**while** (fabs(x0 - x) > 0.000001 && k <= 1);

cout << x << endl;

cout << "Количество итераций: " << count;

}

**void** newton() //Метод Ньютона

{

**double** f, df, k = 0, x = 0;

**int** count = 0;

**do**

{

f = x - (1 / (3 + sin(3.6 \* x)));

df = 1 + (1 / ((3 + sin(3.6 \* x)) \* (3 + sin(3.6 \* x)))) \* cos(3.6 \* x) \* 3.6;

x = x - f / df;

k += 0.000001;

count++;

}

**while** (fabs(f) > 0.000001 && k <= 0.85);

cout << x << endl;

cout << "Количество итераций: " << count;

}

**double** fun (**double** x)

{

**return** x - (1 / (3 + sin(3.6 \* x)));

}

**double** pold (**double** a = 0.0, **double** b = 0.85, **double** epsilant = 0.000001) //Метод половинного деления

{

**double** mid = (a + b) / 2;

**double** x = 0;

**int** count = 0;

**if** (abs (b - a) > epsilant)

{

**if** ((fun(a) \* fun(mid)) <= 0)

{

mid = pold (a, mid);

}

**else**

{

mid = pold (mid, b);

}

}

x = mid;

**return** x;

}

**int** poldcount(**double** a = 0.0, **double** b = 0.85, **double** epsilant = 0.000001) //Метод половинного деления

{

**double** mid = (a + b) / 2;

**double** x = 0;

**int** count = 0;

**if** (abs(b - a) > epsilant)

{

**if** ((fun(a) \* fun(mid)) <= 0)

{

mid = pold (a, mid);

}

**else**

{

mid = pold (mid, b);

}

count++;

}

**return** count;

}

**int** main()

{

setlocale (LC\_ALL, "ru");

**double** x = 0;

**int** count = 0;

cout << endl;

cout << "Метод итераций: ";

iter();

cout << endl << endl;

cout << "Метод Ньютона: ";

newton();

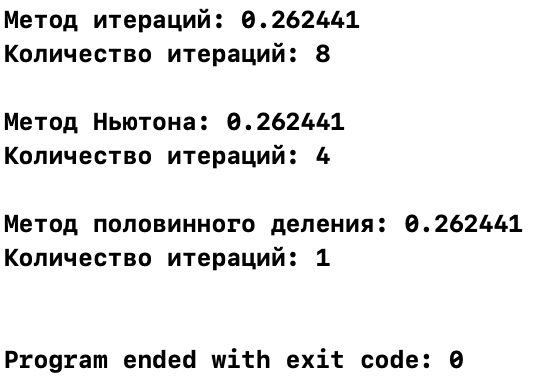
cout << endl << endl;

cout << "Метод половинного деления: " << pold(x) << endl << "Количество итераций: " << poldcount(count);

cout << endl << endl << endl;

**return** 0;

}

**Тестирование**