**Пермский национальный исследовательский политехнический университет**

Кафедра “Информационные технологии и автоматизированные системы”

**Лабораторная работа №6**

» По дисциплине «Основы алгоритмизации и программирования»

**Тема:**

**Решения нелинейных уравнений тремя способами**

Выполнил:

студент группы ИВТ-19-1б

Главатских Владимир Олегович

Проверила:

доцент кафедры “ИТАС”

Полякова О.А.

Пермь, 2019

**Постановка задачи**

Решить уравнение указанным в варианте методом. Уравнение передать в функцию как параметр с помощью указателя.

**Анализ задачи**

1. **Директивы:**

**#include <iostream>**

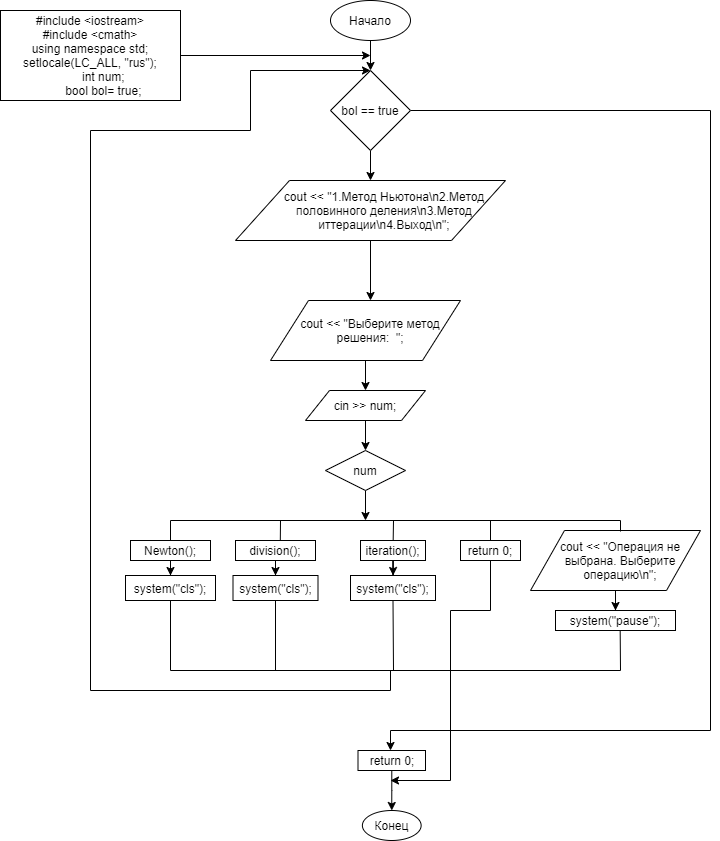
**#include <cmath>**

**using namespace std;**

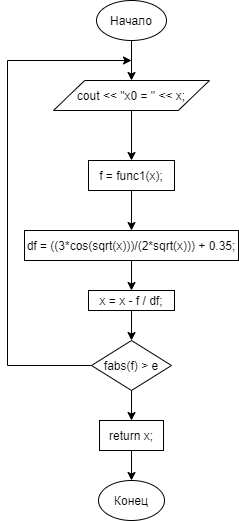
1. **Создаем функцию func, которая принимает в себя параметр х и возвращает заданное в варианте выражение.**
2. **Функция root1 отвечает за решение уравнения методом Ньютона.**
3. **Newton вызывается функцию root1 и передает в нее нужные параметры.**
4. **Функия root2 находит корень уравнения методом половинного деления и в качестве аргумента возвращает х.**
5. **division вызывает root2 и передает в нее нужные значения.**
6. **Функция root3 находит корень уравнения методом иттерации.**
7. **iteration вызывает root3 и передает в нее нужные значения.**
8. **В функции main реализовано меню, которое представляет из себя список методов решений данного нелинейного уравнения и предоставляет пользователю выбирать каким методом его решать.**

**Блок-схема**

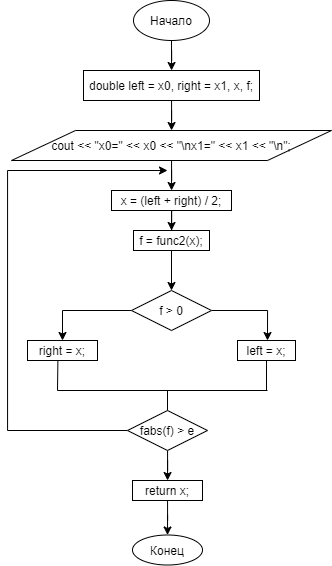
**Рис. 1 main()**



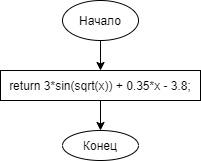
**Рис.2 root1()**



**Рис 3. root2()**



**Рис 3. func()**



**Код**

#include <iostream>

#include <cmath>

#include <clocale>

using namespace std;

double func(double x)

{

return 3 \* sin(sqrt(x)) + 0.35 \* x - 3.8;

}

double root1(double x, double e)

{

double f, df;

cout << "x0 = " << x;

do {

f = func(x);

df = ((3 \* cos(sqrt(x))) / (2 \* sqrt(x))) + 0.35;

x = x - f / df;

} while (fabs(f) > e);

return x;

}

void Newton() {

setlocale(LC\_ALL, "ru");

cout << "\nКорень выражения: " << root1(0.5, 0.000001) << endl;

system("pause");

}

double root2(double x0, double x1, double e)

{

double left = x0, right = x1, x, f;

cout << "x0=" << x0 << "\nx1=" << x1 << "\n";

do {

x = (left + right) / 2;

f = func(x);

if (f > 0) right = x;

else left = x;

} while (fabs(f) > e);

return x;

}

void division() {

setlocale(LC\_ALL, "rus");

cout << "\nКорень выражения: " << root2(2.1, 2.5, 0.000001) << endl;

system("pause");

}

double f(double x)

{

return (3.8 - 3 \* sin(pow(x, 0.5))) / (0.35);

}

int Iteration(double\* x, double eps)

{

int k = 0; double x0;

do

{

x0 = \*x;

\*x = f(x0);

k++;

} while (fabs(x0 - \*x) >= eps);

return k;

}

void iter()

{

double A=2, B=3, X;

double ep = 0.001; //Точность вычислениий.

int K;

setlocale(LC\_ALL, "");

X = (A + B) / 2;

K = Iteration(&X, ep);

cout << "\nКорень выражения: " << X << endl;

system("pause");

}

int main()

{

setlocale(LC\_ALL, "rus");

int num;

do{

cout << " ------------------------------------------------------------" << endl;

cout << "|1.Метод Ньютона | Выражение: |" << endl;

cout << "|----------------------------| |" << endl;

cout << "|2.Метод половинного деления | 3\*sin(sqrt(x)) + 0.35\*x - 3.8 |" << endl;

cout << "|----------------------------| |" << endl;

cout << "|3.Метод итерации | |" << endl;

cout << "|----------------------------| |" << endl;

cout << "|4.Выход | |" << endl;

cout << " ------------------------------------------------------------" << endl;

cout << "Выберите метод решения: ";

cin >> num;

switch (num){

case 1:

Newton();

system("cls");

break;

case 2:

division();

system("cls");

break;

case 3:

iter();

system("cls");

break;

case 4:

return 0;

default:

cout << "Операция не выбрана. Выберите операцию\n";

system("pause");

break;

}

}

while(num != 4);

return 0;

}

**Тестирование**