**Пермский национальный исследовательский политехнический университет**

**Кафедра информационные технологии и автоматизированные системы**

Лабораторная работа №1

**Методы решения нелинейных уравнений**

Вариант: 12

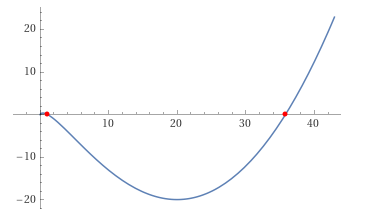
|  |  |  |
| --- | --- | --- |
|  |  | Выполнил:  студент группы: ИВТ-24-2б  Шишкин Максим Григорьевич  Проверил:  доцент кафедры ИТАС  О.А. Полякова |

Пермь 2024

**Постановка задачи**

Решить уравнение 0,1x^2 - x ln x = 0 тремя методами: методом итерации , методом Ньютона и методом половинного деления, на отрезке [1;2] , с заданной точностью 0.001.

**Геометрическая интерпретация**



**Анализ задач**

**Метод Ньютона** - это итерационный метод для нахождения корней уравнений, который использует производную функции для улучшения приближений. Формула для обновления приближения xₙ₊₁ выглядит следующим образом:

xₙ₊₁ = xₙ - f(xₙ) / f'(xₙ),

где f(x) = 0.1x² - x ln x.

1. Определение функции и её производной:

• Функция: f(x) = 0.1x² - x ln x

• Производная: f'(x) = 0.2x - (ln x + 1)

1. Проверка наличия корня на отрезке: Для применения метода Ньютона необходимо убедиться, что функция меняет знак на заданном отрезке [1; 2]:

• f(1) = 0.1(1²) - 1 ln(1) = 0.1

• f(2) = 0.1(2²) - 2 ln(2) ≈ 0.4 - 1.386 ≈ -0.986

Так как f(1) > 0 и f(2) < 0, по теореме Больцано на отрезке [1; 2] действительно существует корень.

1. Применение метода Ньютона: Итерации будут продолжаться до тех пор, пока абсолютное значение разности между двумя последовательными приближениями не станет меньше заданного эпсилон (0.001).

**Метод итераций** - это итерационный метод для нахождения корней уравнений, который использует предыдущее приближение для вычисления следующего. Формула для обновления приближения выглядит следующим образом:

xₙ₊₁ = g(xₙ),

где g(x) — это функция, преобразованная из исходного уравнения, чтобы получить форму итерации.

1. Определение функции и её преобразование:

• Исходная функция:

f(x) = 0.1x² - x ln x

• Для применения метода итераций необходимо выразить x через f(x). Можно использовать, например, следующую формулу:

x = 0.1x² / ln x

что можно переписать как:

g(x) = 0.1x² / ln x

1. Проверка наличия корня на отрезке:

Для применения метода итераций необходимо убедиться, что функция меняет знак на заданном отрезке [1; 2]:

• Вычисляем значения функции в границах отрезка:

• f(1) = 0.1(1²) - 1 ln(1) = 0.1

• f(2) = 0.1(2²) - 2 ln(2) ≈ 0.4 - 1.386 ≈ -0.986

Так как f(1) > 0 и f(2) < 0, по теореме Больцано на отрезке [1; 2] действительно существует корень.

1. Применение метода итераций:

Итерации будут продолжаться до тех пор, пока абсолютное значение разности между двумя последовательными приближениями не станет меньше заданного эпсилон (0.001):

• Начальное приближение x₀ выбираем, например, равным 1.5.

• Затем вычисляем последовательные значения:

xₙ₊₁ = g(xₙ)

**Метод половинного деления -** это итерационный метод для нахождения корней уравнений, основанный на делении отрезка пополам и выборе подотрезка, в котором функция меняет знак. Метод гарантирует сходимость при условии, что функция непрерывна и меняет знак на рассматриваемом отрезке.

1. Определение функции:

• Функция:

f(x) = 0.1x² - x ln x

1. Проверка наличия корня на отрезке:

Для применения метода половинного деления необходимо убедиться, что функция меняет знак на заданном отрезке [1; 2]:

• Вычисляем значения функции в границах отрезка:

• f(1) = 0.1(1²) - 1 ln(1) = 0.1

• f(2) = 0.1(2²) - 2 ln(2) ≈ 0.4 - 1.386 ≈ -0.986

Так как f(1) > 0 и f(2) < 0, по теореме Больцано на отрезке [1; 2] действительно существует корень.

1. Применение метода половинного деления:

Итерации будут продолжаться до тех пор, пока длина интервала не станет меньше заданного эпсилон (0.001). Процесс включает следующие шаги:

• На каждой итерации вычисляем середину отрезка:

c = a + b / 2

где a и b — текущие границы отрезка.

• Затем вычисляем значение функции в середине:

f(c)

• Если f(c) = 0, то найден корень.

• Если f(a) ⋅ f(c) < 0, то корень находится в отрезке [a; c], и обновляем границы:

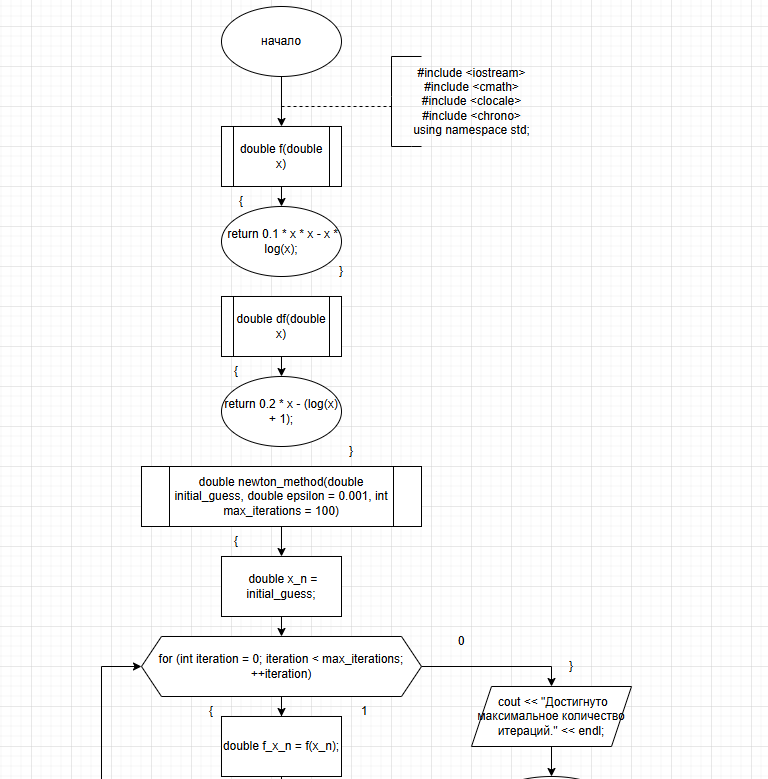
b = c

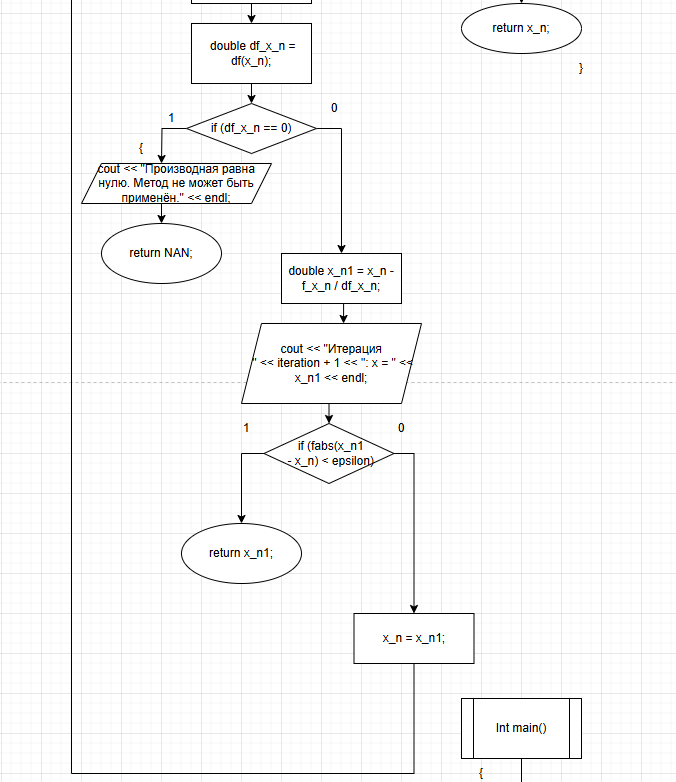
• Если f(c) ⋅ f(b) < 0, то корень находится в отрезке [c; b], и обновляем границы:

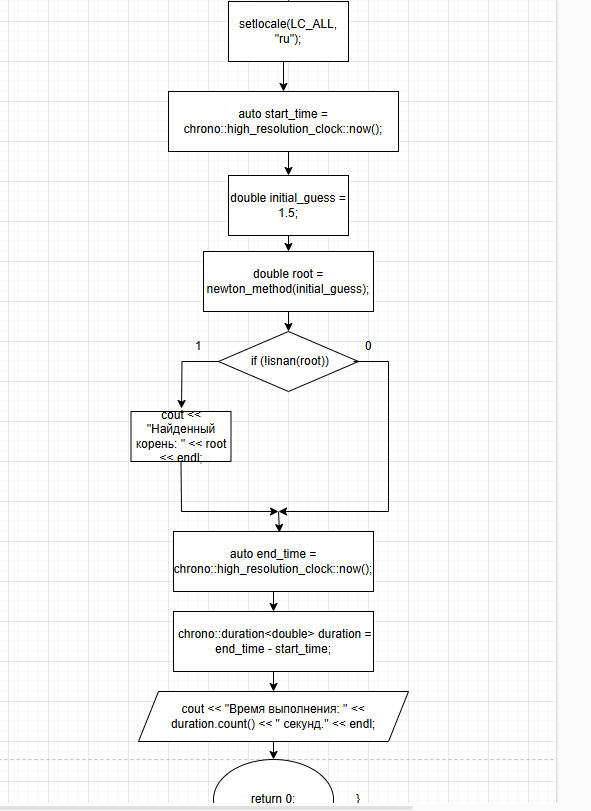
a = c

**Блок схемы**

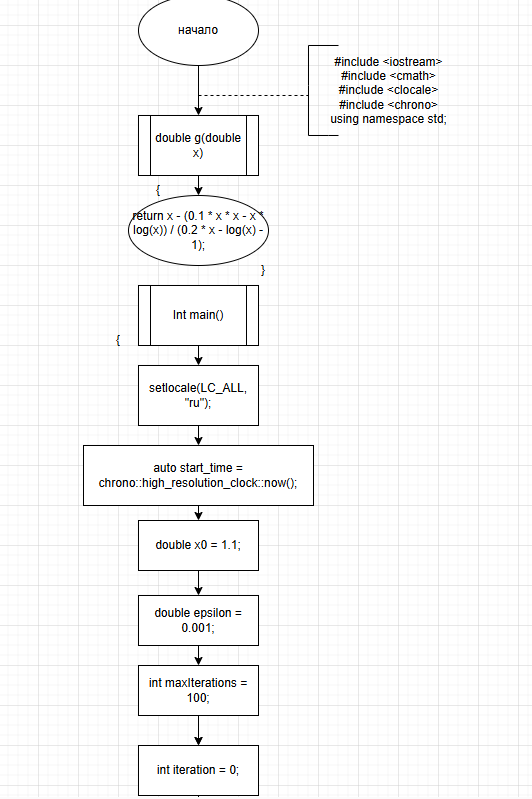
**Метод Ньютона**

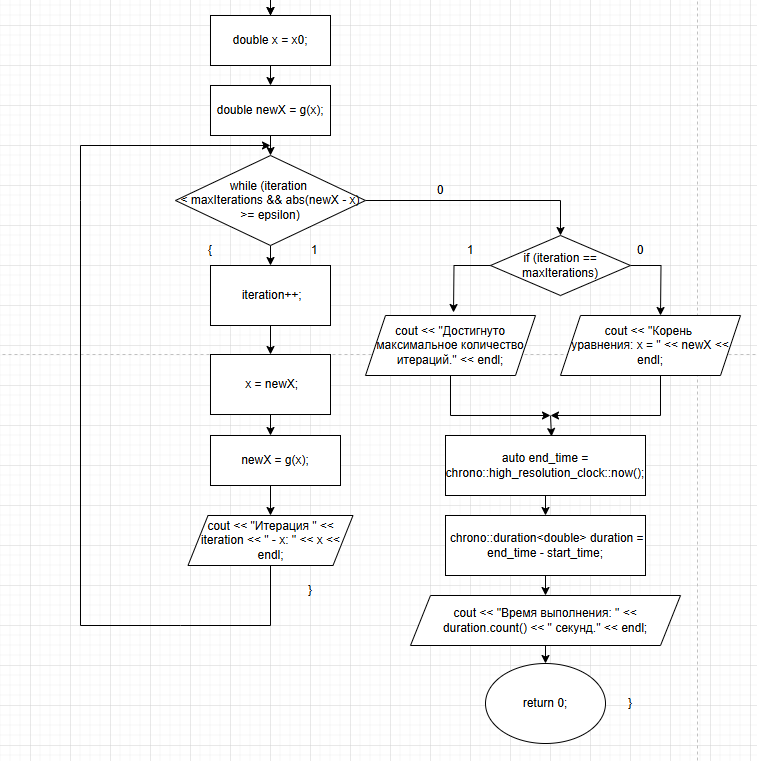




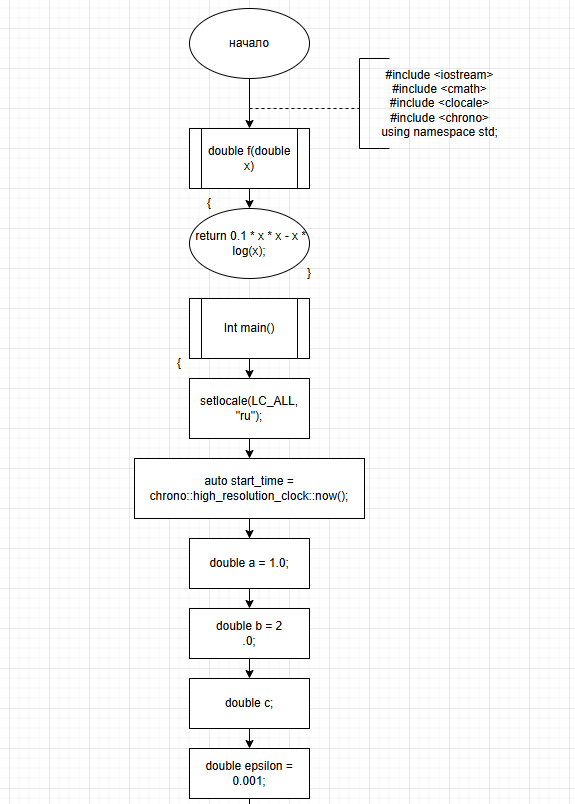


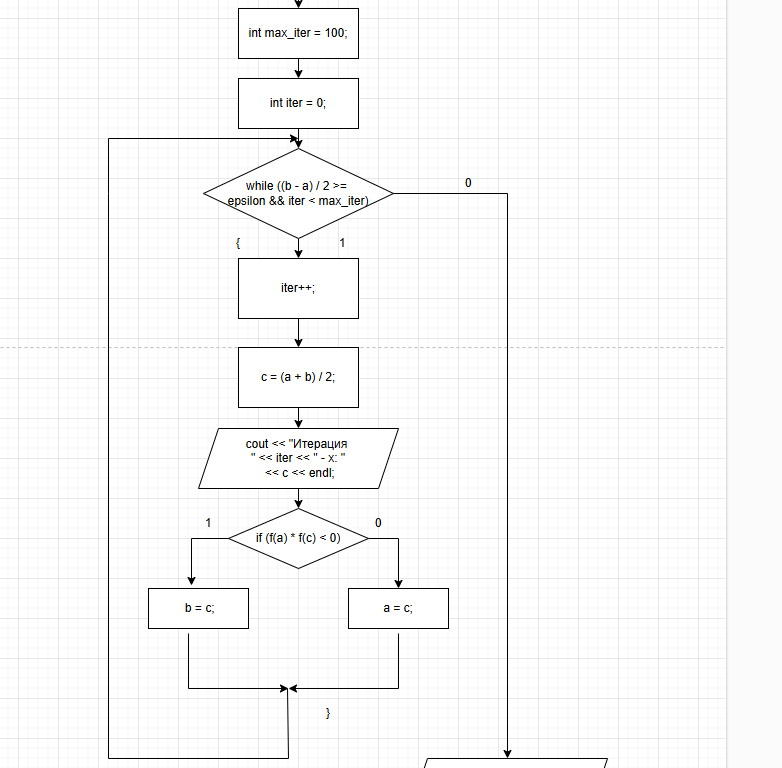
**Метод итераций**

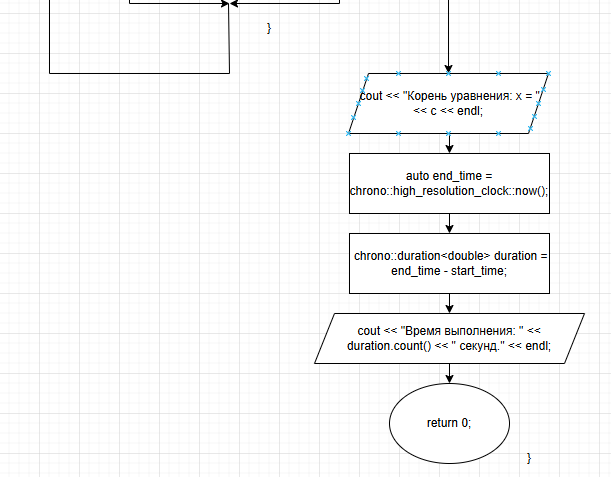




**Метод половинного деления**

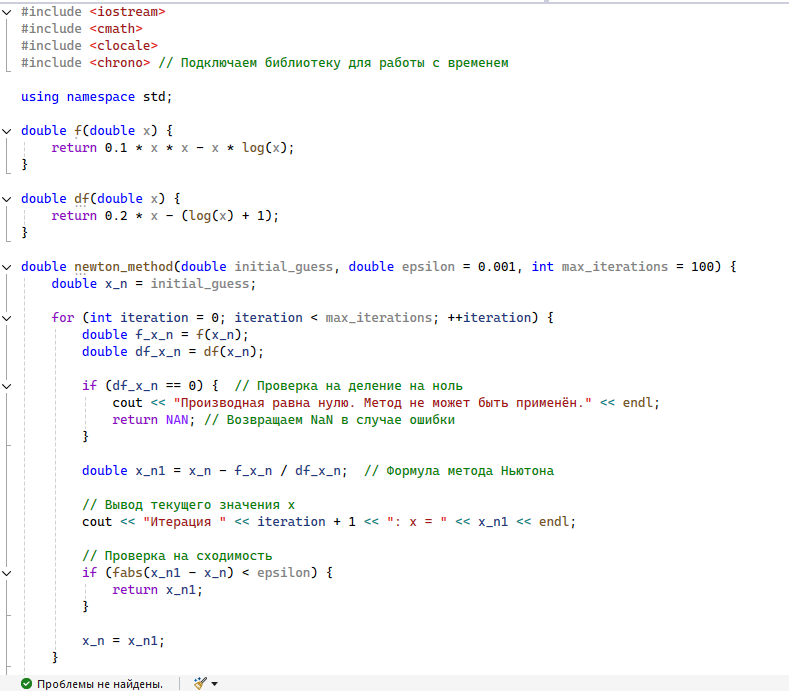


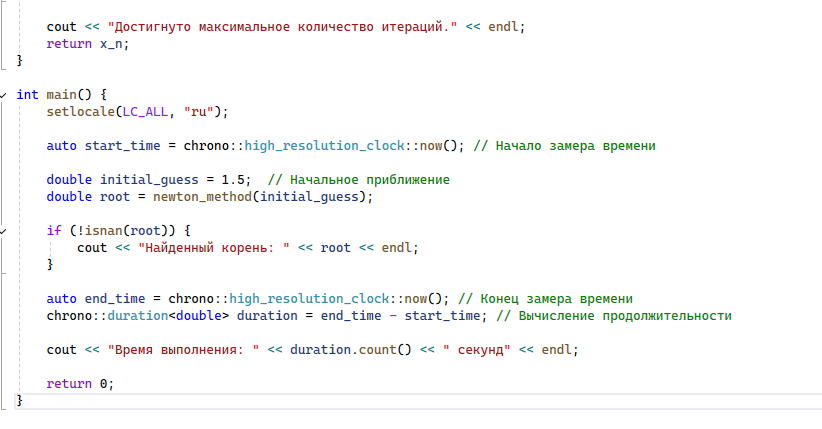


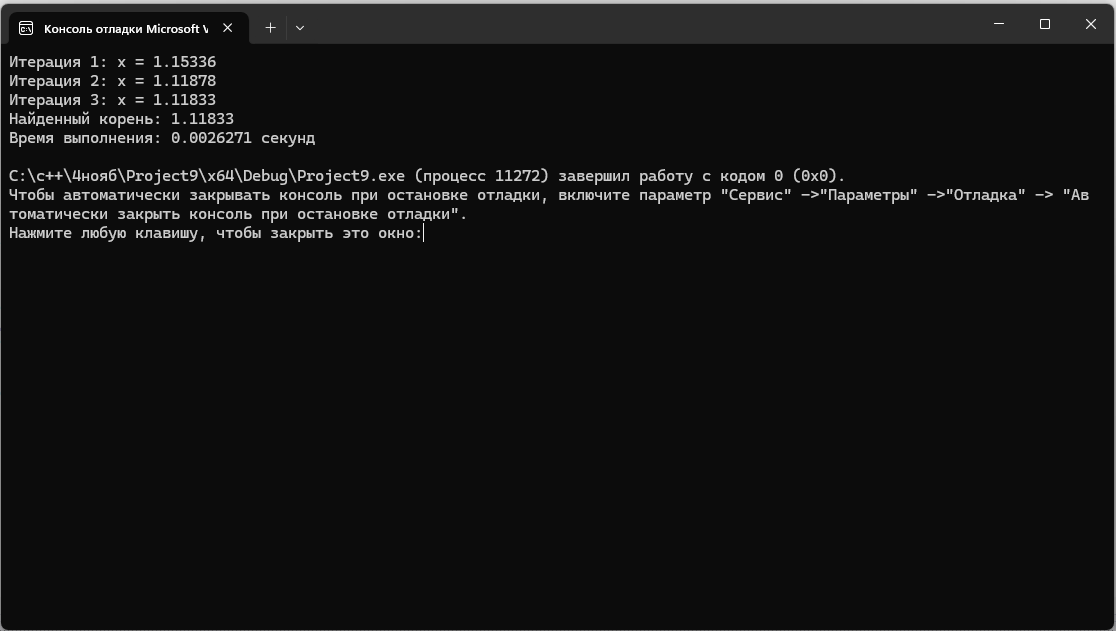


**Код на С++**

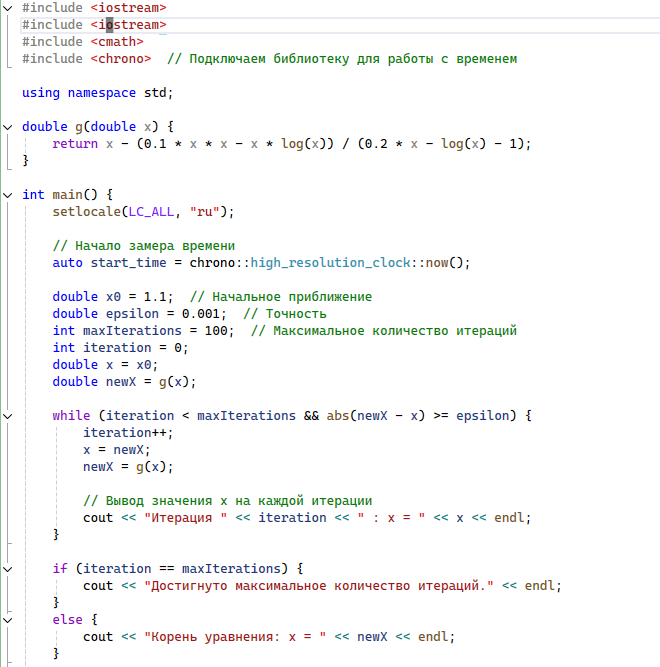
**Ньютон**



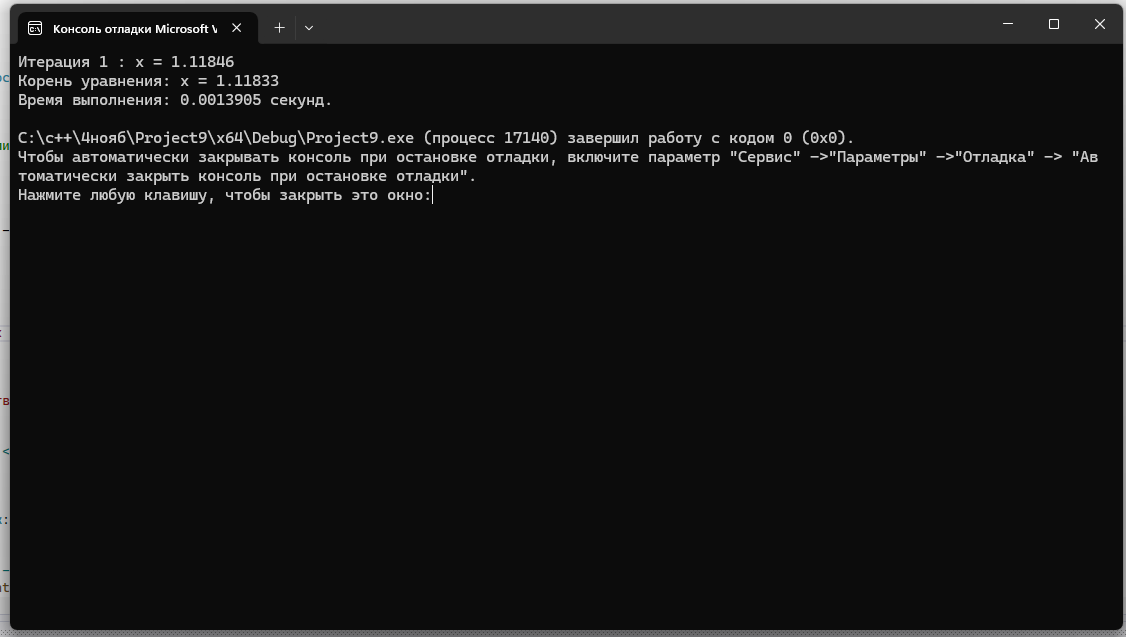




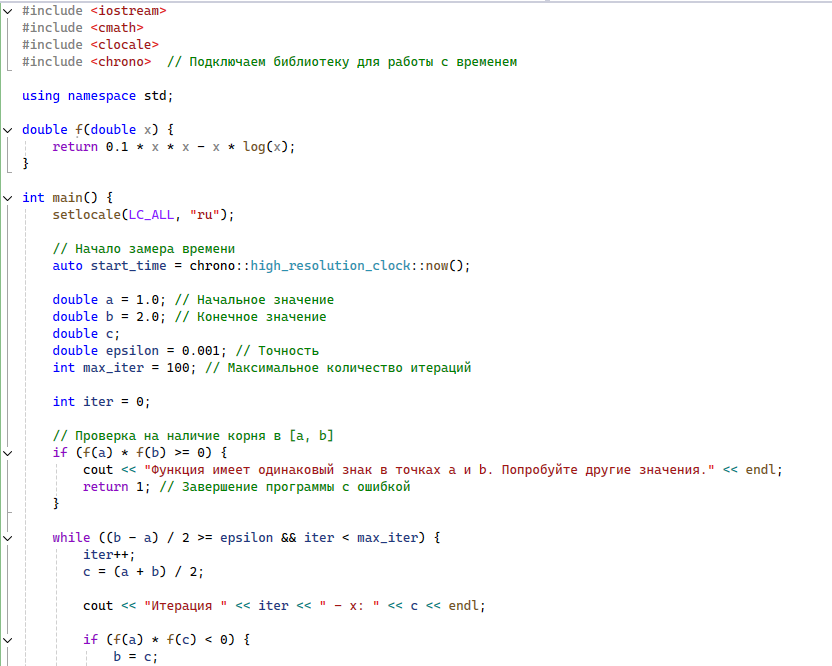
**Метод итераций**

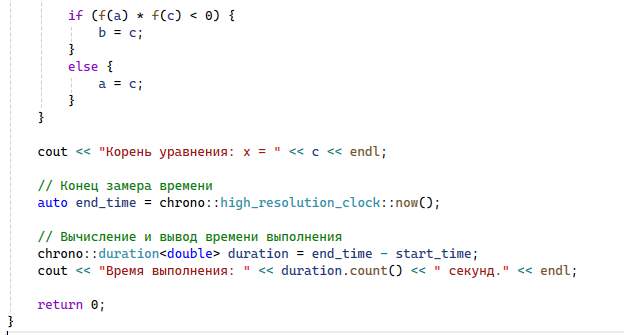


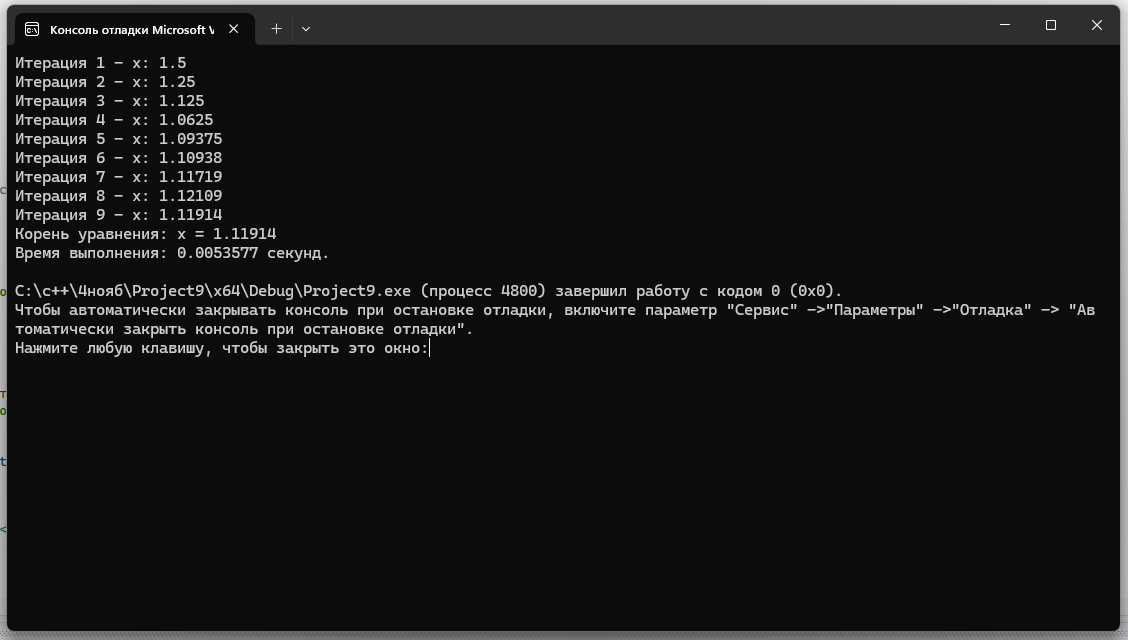




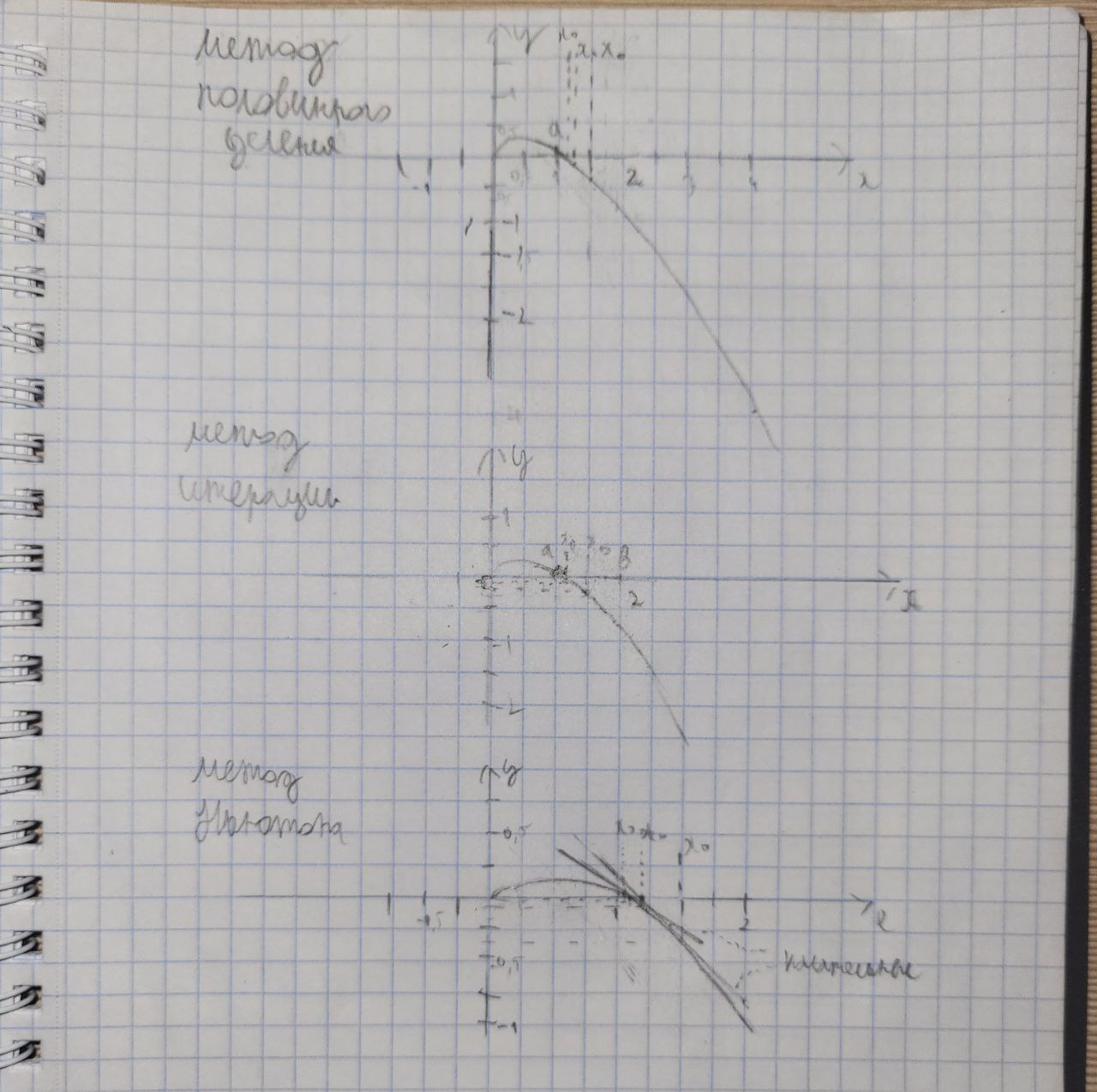
**Метод половинного деления**





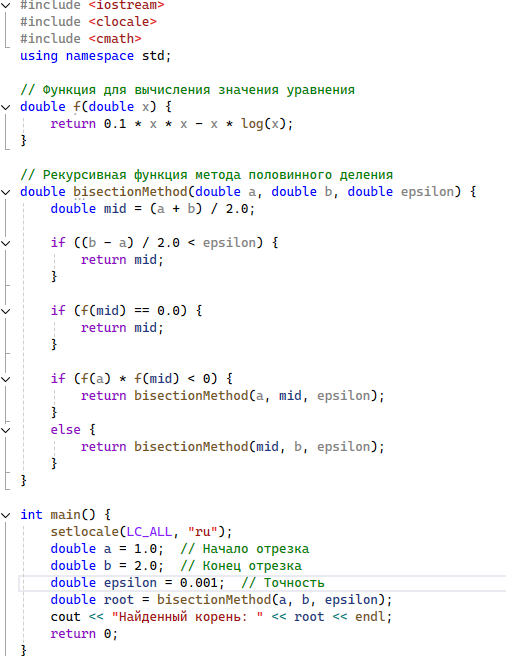


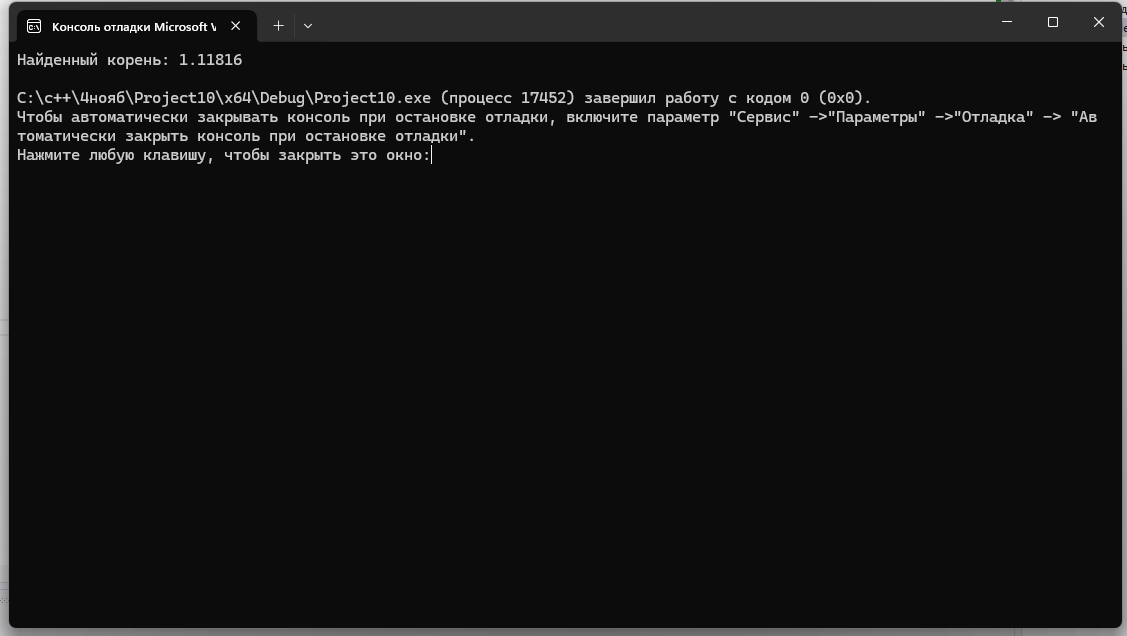
**Графическое решение**



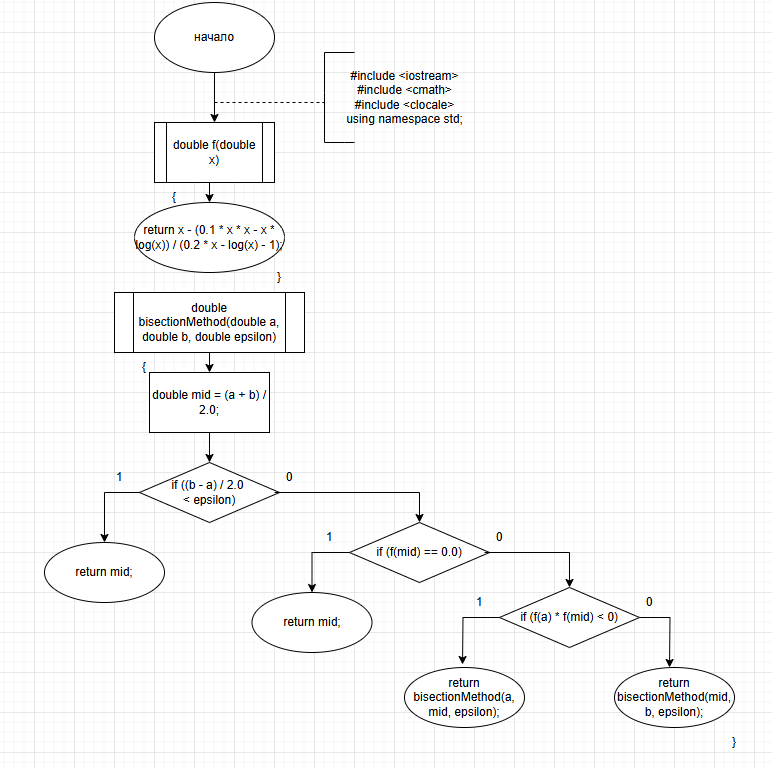
**Рекурсия**

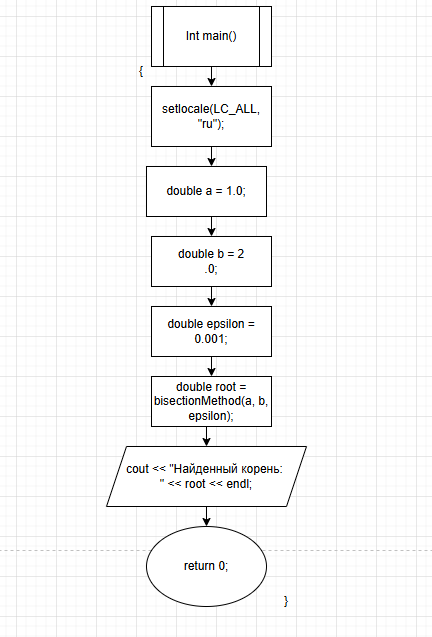
**Код на С++**





**Блок схема**





Ссылка на GitHub <https://github.com/MAKSPOWERO/-.git>