Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №1.**

**«Решение нелинейных уравнений»**

**25 вариант**

Выполнили студенты гр. РИС-24-2б

Меньшиков Арсений Андреевич

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Метод Ньютона**

1. Уравнение:

Отрезок, содержащий корень: [1;2]

Точные значение: 1,3077

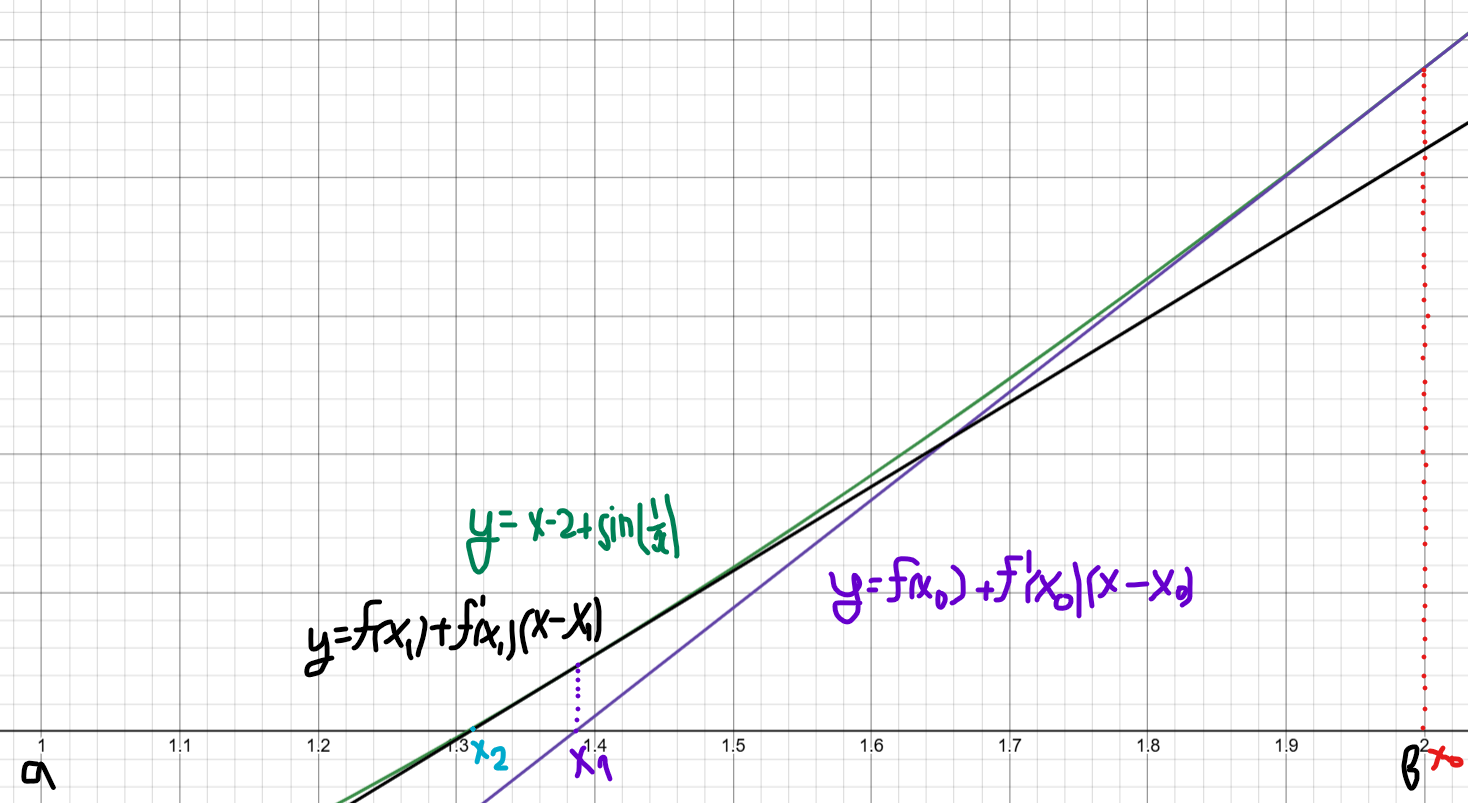
Точность: 0,00001

1. Геометрическая интерпретация метода:

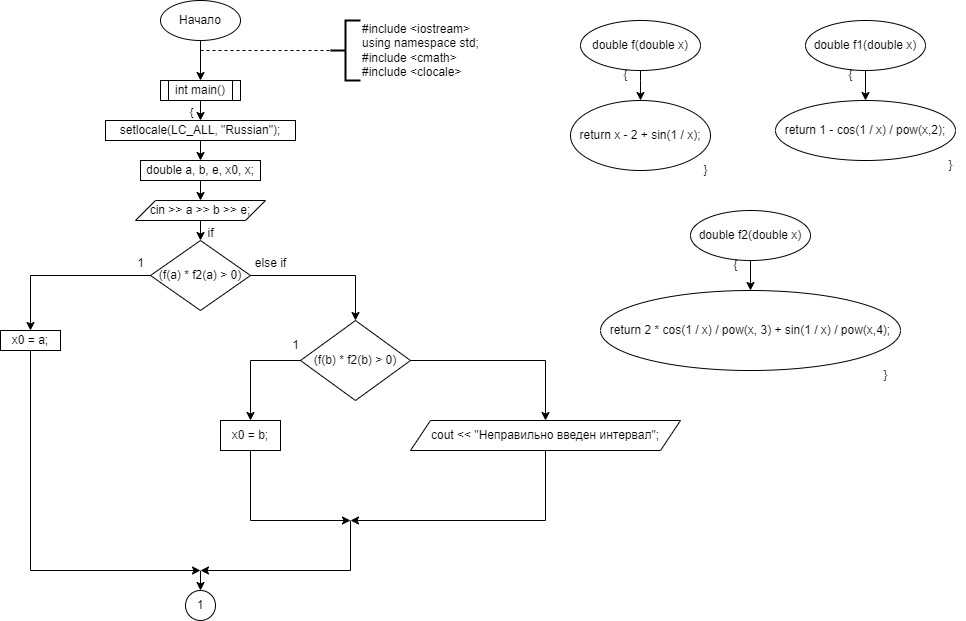
Данный метод основывается на построении касательных к графику, которые проводятся из одного из концов отрезка. В точке x1 пересечения оси X и касательной строится новая касательная к графику. Эти действия продолжаются до тех пор, пока не будет выполнена заданная точность.

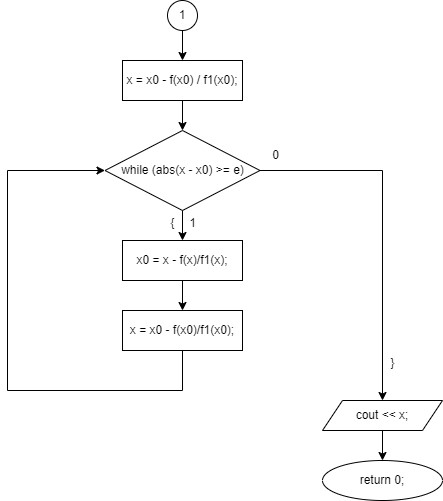
Метод применим, если:

* Известен интервал [a;b], на котором функция монотонна и непрерывна
* f(a) \* f’’(a) > 0 или f(b) \* f’’(b) > 0  
  Основа метода: Если f(a) \* f’’(a) > 0, то x0 = a, если f(b) \* f’’(b) > 0, то x0 = b. К графику проводится касательная в точке x0, уравнение пересекает ось X в точке x1. Проведем еще одну касательную из точки x1 и получим точку x2. Эти действия будут производиться до тех пор, пока , где e – точность, xn – последний найденный корень, xn-1 – предпоследний найденный корень. Формула для нахождения каждого последующего корня:

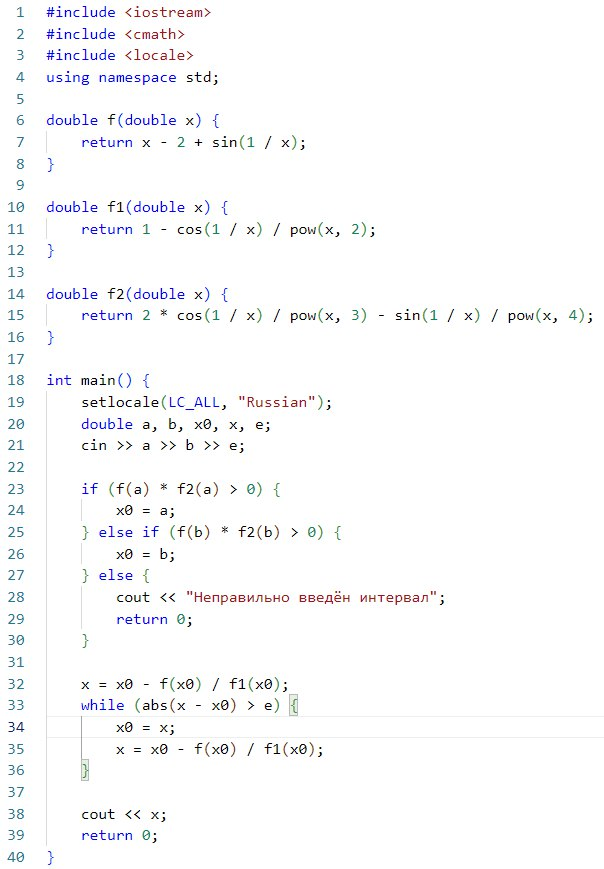


1. Анализ задачи:
2. С помощью функций будем выводить значения уравнения, первой и второй производных. Объявим и введем начало и конец отрезка, точность и объявим переменные под x0 и x1.
3. Проверяем концы отрезка a и b на условие f(a) \* f’’(a) > 0 или f(b) \* f’’(b) > 0 и выбираем нужную точку, из которой будем проводить касательную.
4. С помощью цикла проводим касательные, которые будут с каждым разом все ближе к нужному корню пересекать ось X. Проводим эти действия до тех пор, пока разница двух последних корней не станет меньше или равной точности.
5. Блок-схема:





1. Код программы:





**Метод половинного деления**

1. Уравнение:

Отрезок, содержащий корень: [1;2]

Точные значение: 1,3077

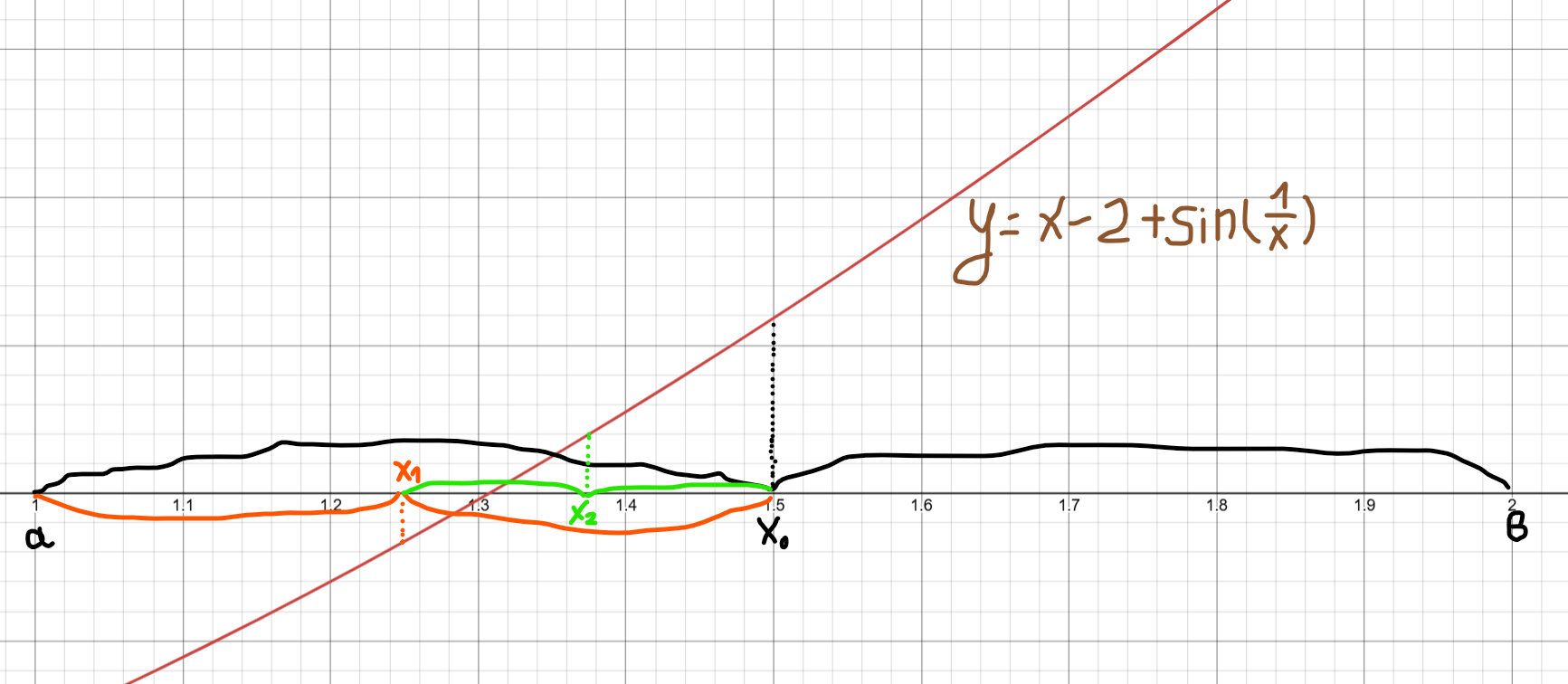
Точность: 0,00001

1. Геометрическая интерпретация метода:

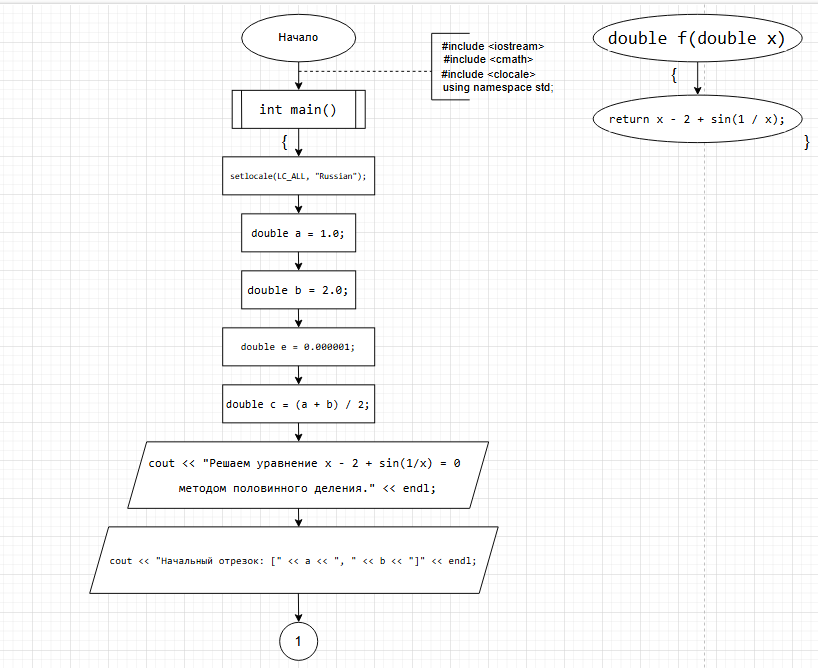
Метод применим, если:

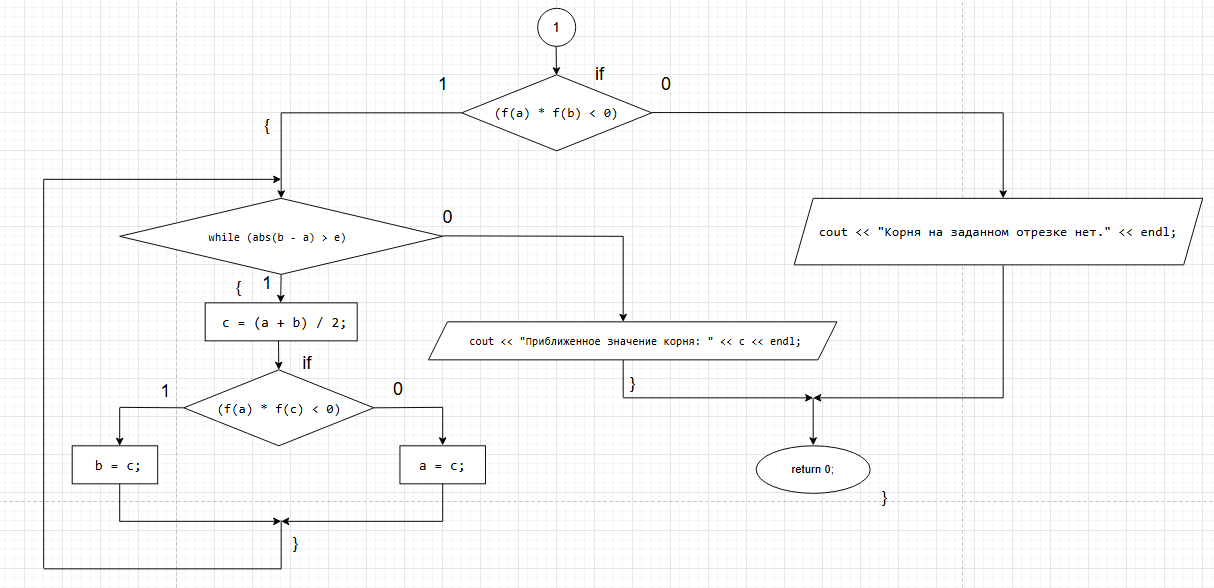
* Известен интервал [a;b], на котором функция монотонна и непрерывна.
* Выполняется условие

Основой метода является деление интервала пополам и отбрасывание той части отрезка, в которой нет корня, так как не будет выполняться одно из условий (). Данные действия будут выполняться до тех пор, пока не будет достигнута заданная точность e. ()

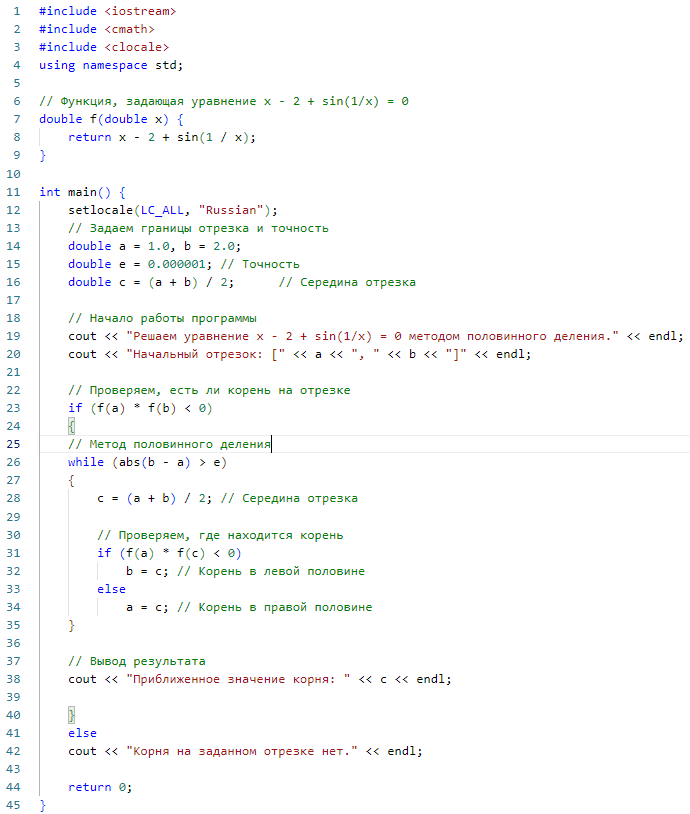


1. Анализ задачи
2. Зададим границы интервала, точность и объявим переменную, в которую будем вносить значение половины интервала.
3. Проверяем, существует ли корень на интервале, если да, то переходим к следующему шагу, иначе завершить программу.
4. Делим отрезок пополам и находим в какой половине находится корень. Проводим данные действия до тех пор, пока не дойдем до заданной точности.
5. Блок-схема





1. Код программы:





**Метод итераций**

1. Уравнение:

Отрезок, содержащий корень: [1;2]

Точные значение: 1,3077

Точность: 0,00001

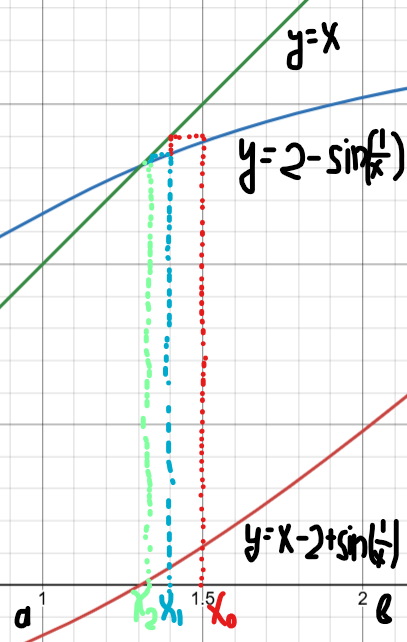
1. Геометрическая интерпретация:

Метод применим, если:

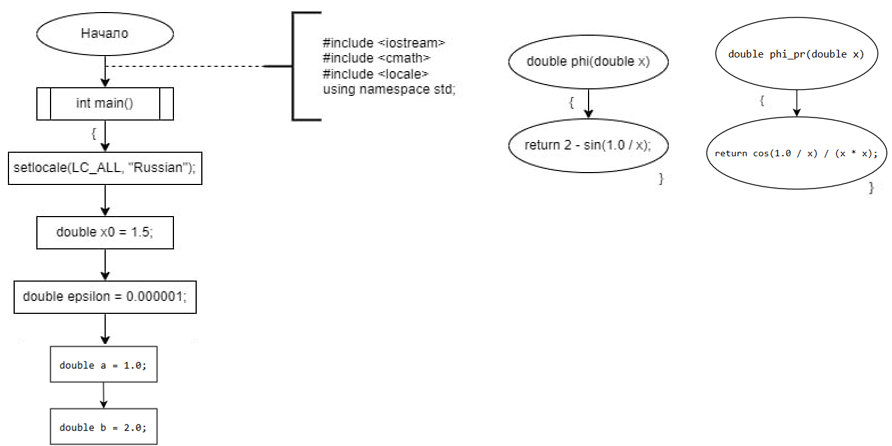
* Известен интервал [a;b], на котором есть корень.
* Выполняется условие сходимости

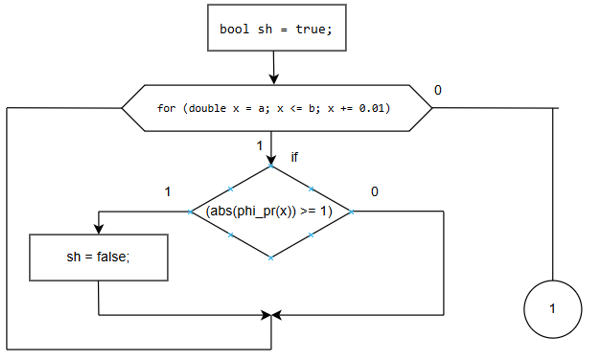
Алгоритм решения данным методом:

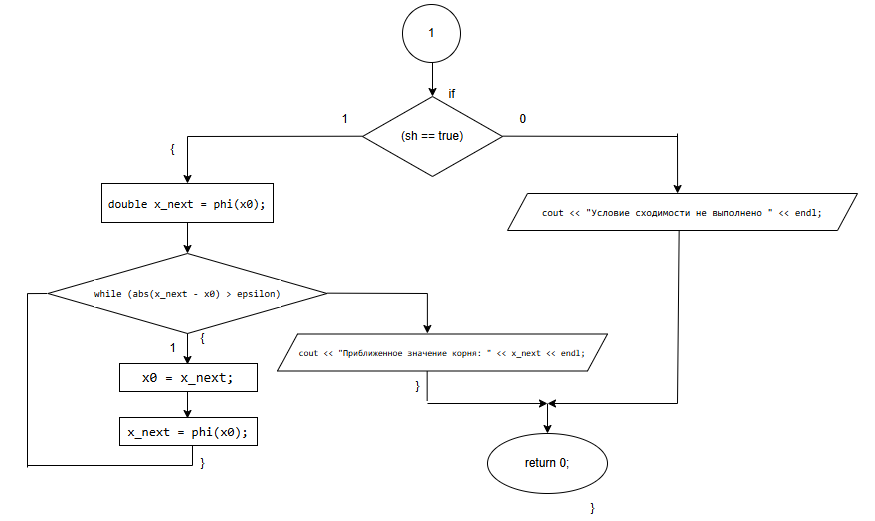
1. Преобразуем уравнение вида f(x) = 0 в вид , путем выражения x из уравнения.
2. Выбираем начальное значение, принадлежащее интервалу.
3. Вычисляем следующее значение по формуле
4. Алгоритм завершает тогда, когда модуль разницы двух последних значений меньше и равна заданной точности ().



1. Анализ задачи:
2. Выбираем начальным значением произвольное значение x0, которое удовлетворяет условию сходимости.
3. С помощью цикла находим следующее значение по формуле , будем искать корень до тех пор, пока модуль разницы двух последних найденных корней не будет удовлетворять точности.
4. Блок-схема:







1. Код программы:

