Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 7

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Циклы с пред- и постусловием»

Выполнил: ст. гр. ТУУ-111

Ванчаев Д.С.

Вариант №10

16.03.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

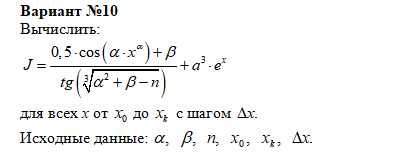
Москва – 2025 г.

1. **Цель работы**

Решить поставленную задачу программирования по разделу «Циклы с пред- и постусловием». Не использовать операторы циклов по известному диапазону (*for*) и известному множеству (*foreach*) значений.

1. **Формулировка задачи**

Вычислить:



1. **Блок** – **схема алгоритма**

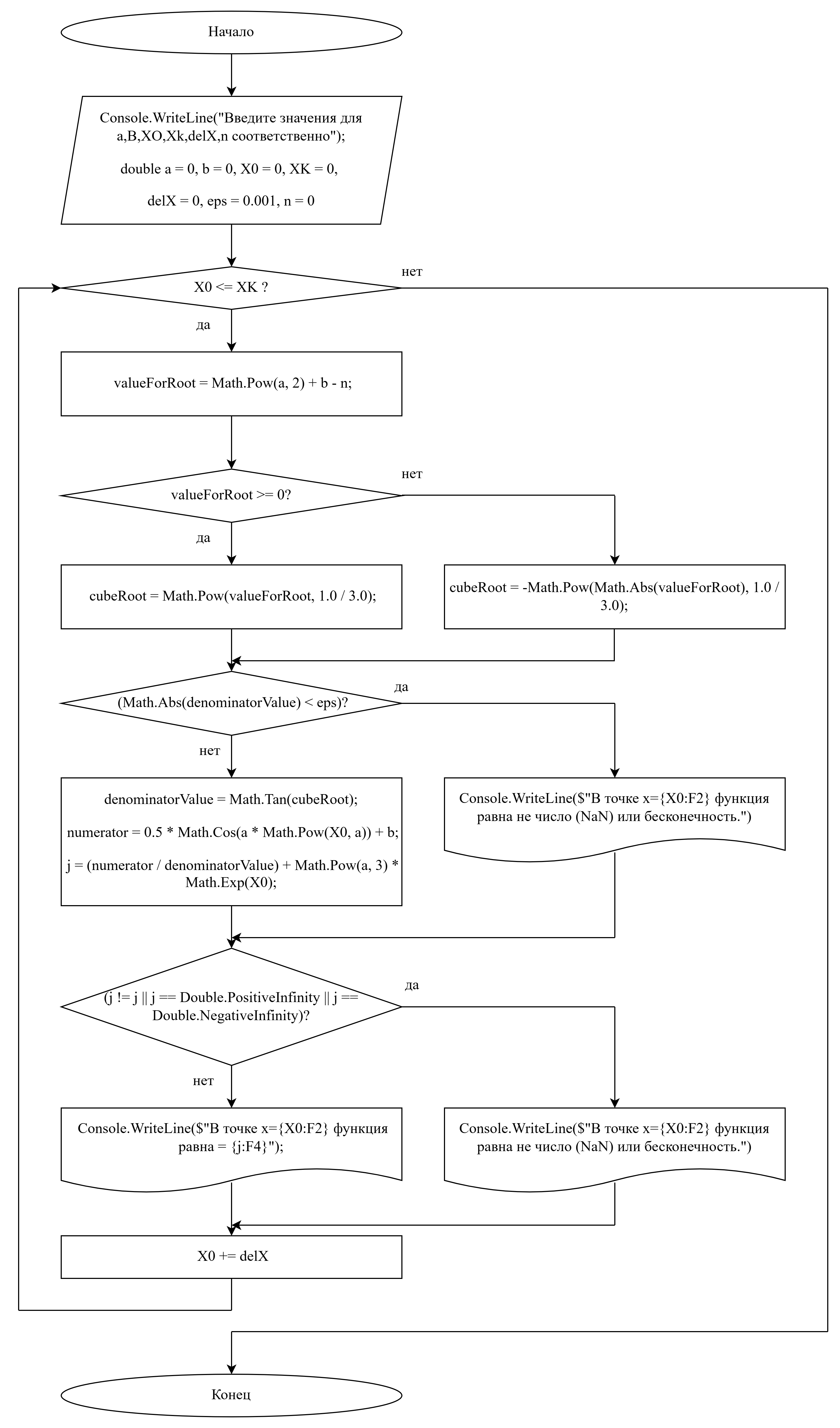


Рисунок 3.1 – Блок схема алгоритма

1. **Подбор тестовых примеров**

1)Тестовый пример решения задачи «Циклы с пред- и постусловием»:

a = 0,3

b = 0,4

n = 0,5

X0 = - 10

XK = 10

delX = 0,01

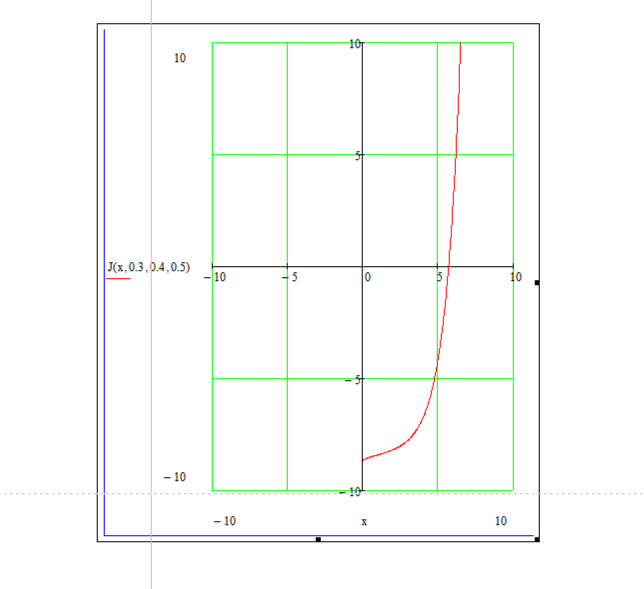


Рисунок 4.1 – График сделанный в приложении Mathcad

Значение подкоренного выражения в знаменателе будет меньше 0.

2)Тестовый пример решения задачи «Циклы с пред- и постусловием»:

a = 1

b = 0,1

n = 0,9

X0 = - 10

XK = 10

delX = 0,01

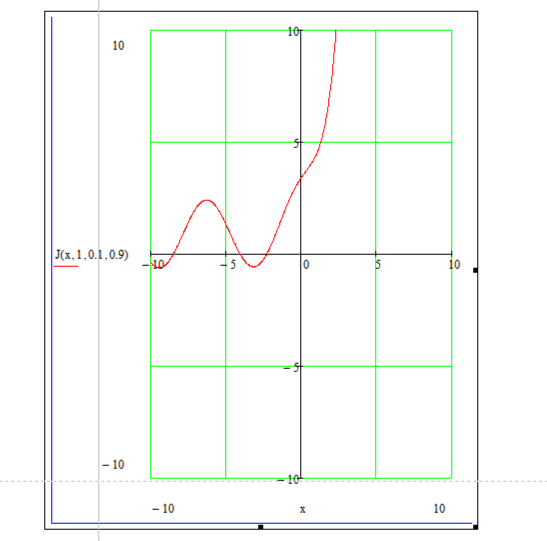


Рисунок 4.2 – График сделанный в приложении Mathcad

Значение подкоренного выражения в знаменателе будет больше 0.

1. **Листинг**

using System;

namespace Zadanie7

{

internal class Program

{

static void Main()

{

Console.WriteLine("Введите значения для a, B, X0, XK, delX, n соответственно");

double a = 0, b = 0, X0 = 0, XK = 0, delX = 0, eps = 0.001, n = 0;

double valueForRoot;

double cubeRoot;

double numerator;

double denominatorValue;

double j;

while (true)

{

Console.Write("Введите значение a: ");

if (double.TryParse(Console.ReadLine(), out a))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

while (true)

{

Console.Write("Введите значение B: ");

if (double.TryParse(Console.ReadLine(), out b))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

while (true)

{

Console.Write("Введите значение X0: ");

if (double.TryParse(Console.ReadLine(), out X0))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

while (true)

{

Console.Write("Введите значение XK: ");

if (double.TryParse(Console.ReadLine(), out XK))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

while (true)

{

Console.Write("Введите значение delX: ");

if (double.TryParse(Console.ReadLine(), out delX))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

while (true)

{

Console.Write("Введите значение n: ");

if (double.TryParse(Console.ReadLine(), out n))

break;

else { Console.WriteLine("Ошибка! Введите число!"); }

}

Console.WriteLine("\nРезультаты вычислений:");

while (X0 <= XK + eps)

{

// Расчёт кубического корня вручную

valueForRoot = Math.Pow(a, 2) + b - n;

if (valueForRoot >= 0)

cubeRoot = Math.Pow(valueForRoot, 1.0 / 3.0); // Корень для положительных чисел

else

cubeRoot = -Math.Pow(Math.Abs(valueForRoot), 1.0 / 3.0); // Корень для отрицательных чисел

denominatorValue = Math.Tan(cubeRoot);

// Проверка на деление на ноль

if (Math.Abs(denominatorValue) < eps)

{

Console.WriteLine($"В точке x={X0:F2} функция не определена (деление на ноль).");

}

else

{

numerator = 0.5 \* Math.Cos(a \* Math.Pow(X0, a)) + b;

j = (numerator / denominatorValue) + Math.Pow(a, 3) \* Math.Exp(X0);

// Проверка на NaN или бесконечность

if (j != j || j == Double.PositiveInfinity || j == Double.NegativeInfinity)

{

Console.WriteLine($"В точке x={X0:F2} функция равна не число (NaN) или бесконечность.");

}

else

{

Console.WriteLine($"В точке x={X0:F2} функция равна = {j:F4}");

}

}

X0 += delX;

}

Console.WriteLine("Выполнение завершено.");

Console.ReadKey();

}

}

}

1. **Расчёт тестовых примеров на ПК:**

**1)**

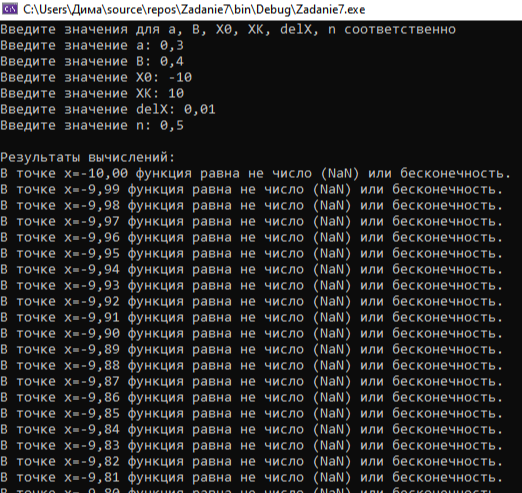


Рисунок 6.1 – Тестирование №1

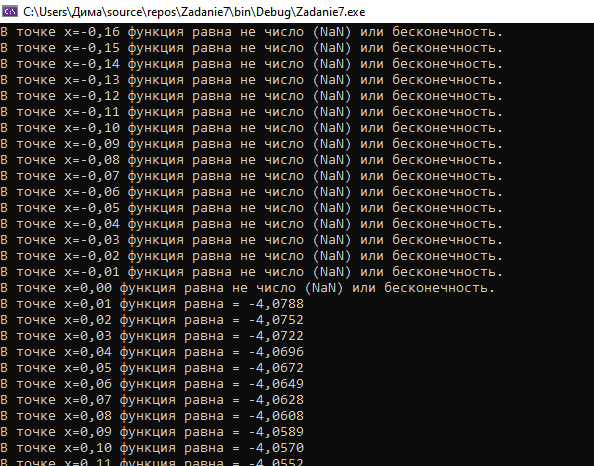


Рисунок 6.1.1 – Тестирование №1

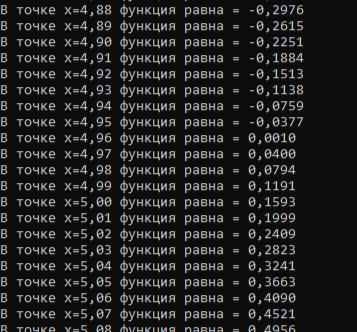


Рисунок 6.1.2 – Тестирование №1

**2)**

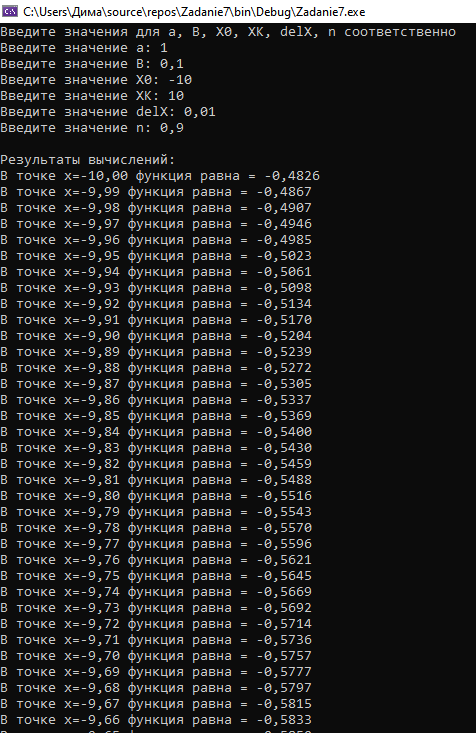


Рисунок 6.2 – Тестирование №2

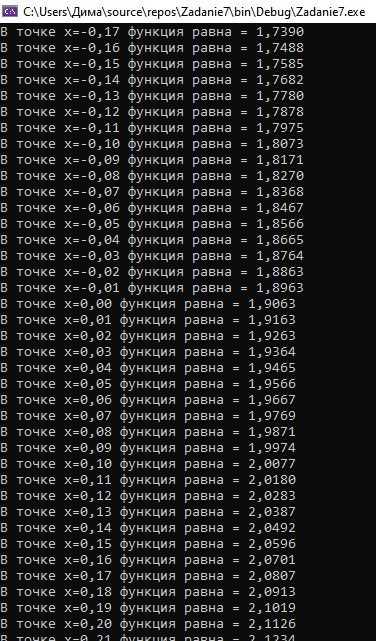


Рисунок 6.2.2 – Тестирование №2

1. **Вывод**

Работа позволила освоить использование циклов с предусловием для пошаговых вычислений. Программа выполняет корректные расчёты при заданных условиях и обрабатывает ошибки ввода данных.