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

Богомолов В.Н.

Вариант №5

23.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

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

(дата приёмки)

Москва – 2024 г.

1. **Цель**

Научиться применять циклы с предусловием и постусловием для решения вычислительных задач на языке программирования C# в среде разработки Microsoft Visual Studio.

1. **Формулировка задачи**

Разработать программу, которая вычисляет значение функции e в заданном диапазоне с шагом da, используя циклы. Проверить корректность введённых данных и отобразить результаты вычислений или сообщения об ошибках.

1. **Блок-схема алгоритма**

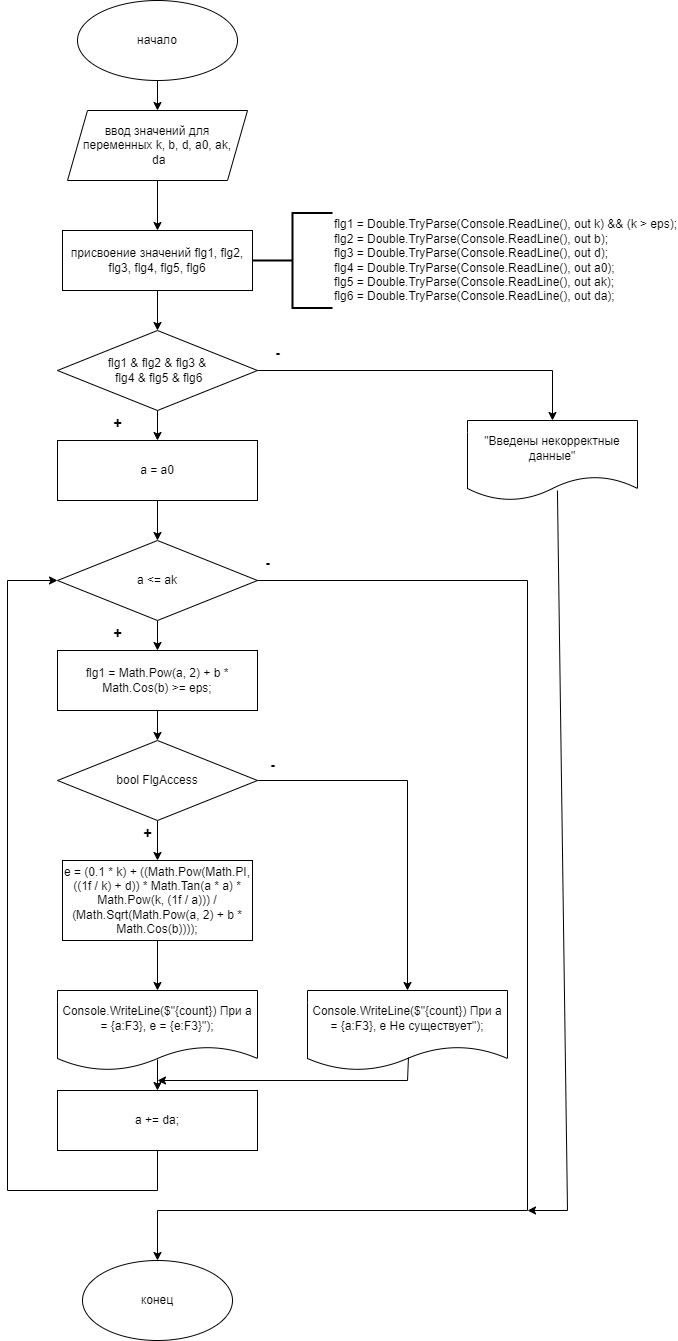
****

Рисунок 3.1.1 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. **Первый ввод**

k**=** 5

b= 0

d=5

a0 = -0, 5

ak = 0,5

da = 0,001

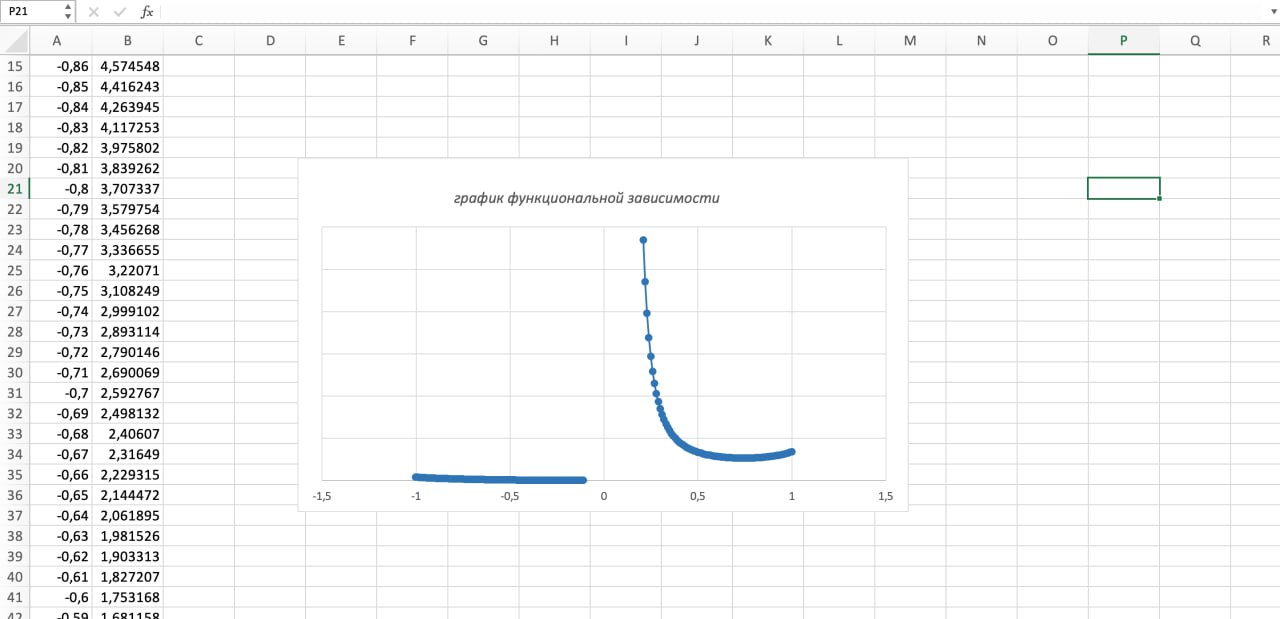


Рисунок 4.1.1 – График зависимости выполненный в Exel

* 1. **Второй ввод**

k**= -**5

b= 0

d=5

a0 = -0, 5

ak = 0,5

da = 0,001

Значение k является отрицательным, следовательно, "Введены некорректные данные"

* 1. **Третий ввод**

k**=** 5

b= 0

d=5

a0 = t

ak = 0,5

da = 0,001

Значение a0 не является числом, следовательно, "Введены некорректные данные"

# Листинг (код программы)

using System; // Подключение базовой библиотеки для работы с основными функциями языка C#

class Program // Объявление главного класса программы

{

static void Main(string[] args) // Основной метод программы, с которого начинается выполнение

{

// Объявление переменных, используемых в программе

double k, b, d, a, a0, ak, da, e; // Числовые переменные для расчётов

int count = 0; // Переменная для подсчёта итераций

const double eps = 0.001; // Константа для минимального допустимого значения

// Логические флаги для проверки условий

bool flg1, flg2, flg3, flg4;

// Вывод приглашения к вводу данных пользователем

Console.WriteLine("--- Ввод данных ---");

// Ввод значения k

Console.Write("Введите значение k (должно быть больше 0): ");

flg1 = double.TryParse(Console.ReadLine(), out k) && k > eps; // Проверка корректности ввода и значения

if (!flg1)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Ввод значения b

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

flg2 = double.TryParse(Console.ReadLine(), out b); // Проверка корректности ввода

if (!flg2)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Ввод значения d

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

flg3 = double.TryParse(Console.ReadLine(), out d); // Проверка корректности ввода

if (!flg3)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Ввод начального значения диапазона a0

Console.Write("Введите начало диапазона (a0): ");

flg4 = double.TryParse(Console.ReadLine(), out a0); // Проверка корректности ввода

if (!flg4)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Ввод конечного значения диапазона ak

Console.Write("Введите конец диапазона (ak): ");

flg4 = double.TryParse(Console.ReadLine(), out ak); // Повторное использование flg4 для проверки ввода

if (!flg4)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Ввод шага da

Console.Write("Введите шаг (da): ");

flg4 = double.TryParse(Console.ReadLine(), out da) && da > 0; // Проверка корректности ввода и значения

if (!flg4)

{

Console.WriteLine("Ошибка! Некорректные данные."); // Сообщение об ошибке

return; // Завершение программы

}

// Установка начального значения переменной a для итераций

a = a0; // Присваиваем начальное значение диапазона

// Основной цикл программы с предусловием

while (a <= ak) // Условие выполнения цикла: пока a меньше или равно конечному значению диапазона

{

count++; // Увеличиваем счётчик итераций

// Проверяем условия существования результата e

flg1 = Math.Pow(a, 2) + b \* Math.Cos(b) >= eps; // Условие 1: выражение должно быть больше или равно eps

flg2 = Math.Abs(Math.Sqrt(Math.Pow(a, 2) + b \* Math.Cos(b))) >= eps; // Условие 2: значение под корнем должно быть допустимым

flg3 = Math.Abs(Math.Cos(a \* a)) >= eps; // Условие 3: косинус значения не должен быть слишком мал

flg4 = Math.Abs(k) >= eps && Math.Abs(a) >= eps; // Условие 4: значения k и a должны быть больше минимального значения

if (flg1 && flg2 && flg3 && flg4) // Если все условия выполнены

{

// Вычисляем значение e по заданной формуле

e = (0.1 \* k) + ((Math.Pow(Math.PI, (1 / k) + d) \* Math.Tan(a \* a) \* Math.Pow(k, 1 / a))

/ Math.Sqrt(Math.Pow(a, 2) + b \* Math.Cos(b)));

Console.WriteLine($"{count}) При a = {a:F3}, e = {e:F3}"); // Вывод результата расчёта

}

else // Если хотя бы одно условие не выполнено

{

Console.WriteLine($"{count}) При a = {a:F3}, e не существует"); // Сообщение о невозможности расчёта

}

// Увеличиваем значение a на шаг da

a += da;

}

// Сообщение об окончании расчётов

Console.WriteLine("\nРасчёты завершены.");

Console.ReadKey(); // Ожидание нажатия клавиши для завершения программы

}

}

1. **Расчёт тестовых примеров на ПК (Тестирование)**

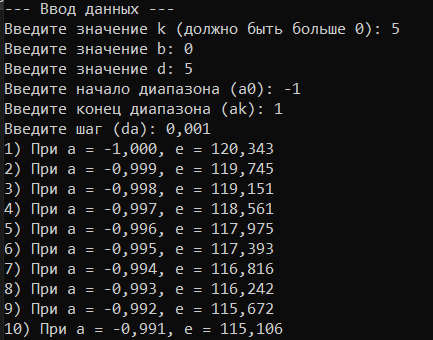


Рисунок 6.1.1 – Тестирование №1

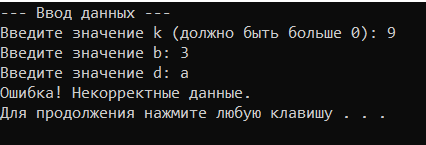


Рисунок 6.1.2 – Тестирование №2

1. **Вывод**

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