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 № 6

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Циклы с пред- и постусловием»

Выполнил: ст. гр. ТУУ-111

Сойка С.А.

Вариант №6

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

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

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

(дата приёмки)

Москва – 2022 г.

**1. Цель работы.**

Решить поставленную задачу программирования по разделу «Циклы с пред- и постусловием» на языке Visual C#.

**2. Формулировка задачи.**

Учесть все возможные ограничения, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости (если они указаны в задании по варианту).

Ввести контроль исходных данных. Реализовать схему «ввод до победного» без возможности прерывания. При некорректном вводе исходных данных заставлять пользователя вводить сведения до тех пор, пока он не введёт их корректно. Продумать побуждающие сообщения-подсказки, направляющие пользователя ко вводу корректных значений.

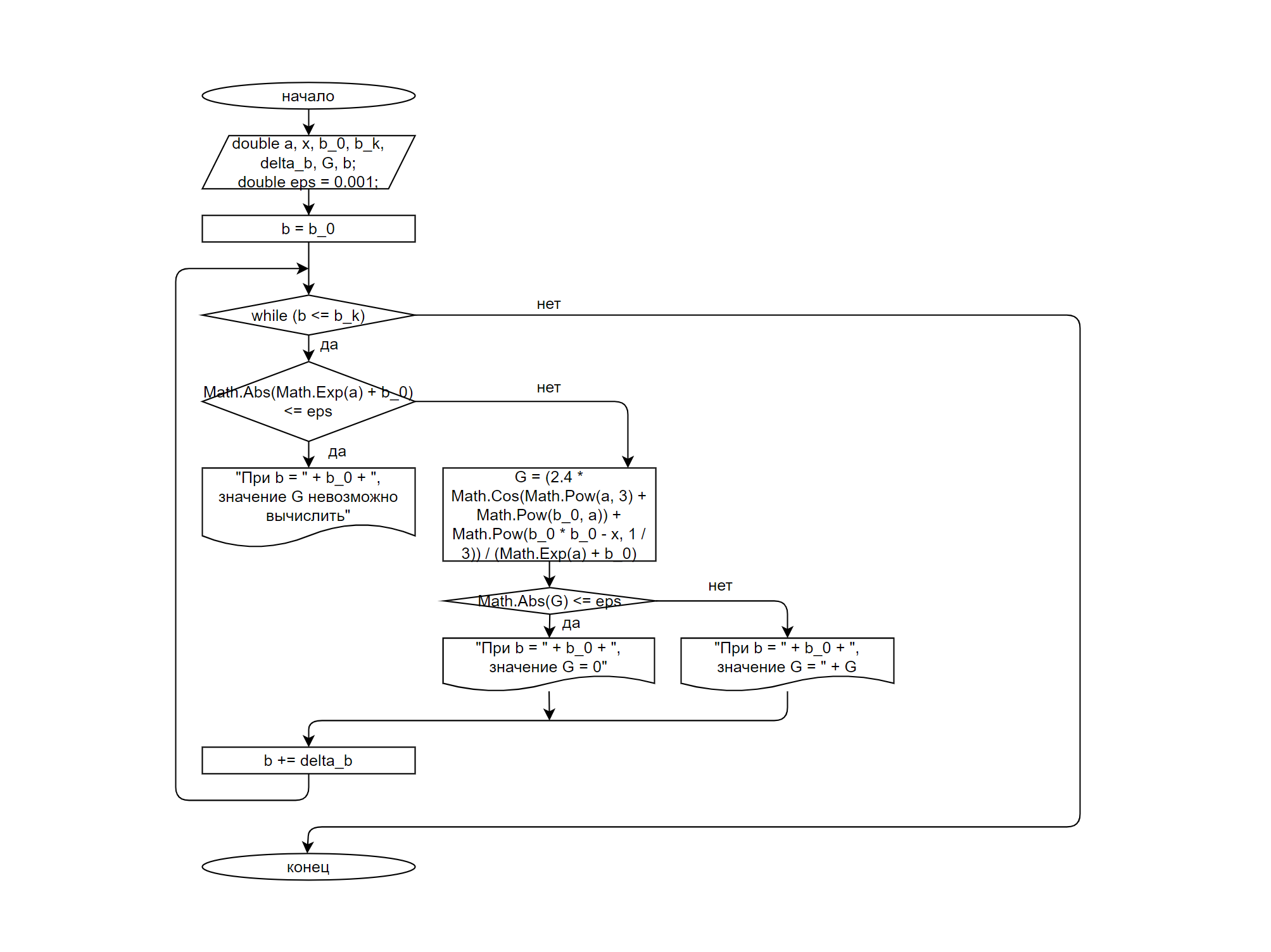
Использовать для контроля исходных данных различные состояния метода «TryParse», входящего в перечень доступных компонентов интересующего значащего (valuable) типа данных, например: int, byte, float, double.

Поставленная задача:

Изображение выглядит как текст

Автоматически созданное описание

**3. Блок-схема алгоритма.**



**4. Подбор тестовых примеров.**

При , ,

* при G = 0,01298753
* при G = 0,05419079
* при G = 0,05407836

При , ,

* при G невозможно вычислить
* при G = 1,14836276
* при G = 0,57418138

Вычисления сделаны с помощью инженерного калькулятора.

**5. Листинг (код) программы.**

using System;

namespace task\_6

{

internal class Program

{

static void Main(string[] args)

{

double a, x, b\_0, b\_k, delta\_b, G, b;

double eps = 0.001;

Console.WriteLine("Введите значение переменной а");

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

Console.WriteLine("Введите значение переменной x");

double.TryParse(Console.ReadLine(), out x);

Console.WriteLine("Введите начальное значение b");

double.TryParse(Console.ReadLine(), out b\_0);

Console.WriteLine("Введите конечное значение b");

double.TryParse(Console.ReadLine(), out b\_k);

Console.WriteLine("Введите значение шага");

double.TryParse(Console.ReadLine(), out delta\_b);

b = b\_0;

while (b <= b\_k)

{

if (Math.Abs(Math.Exp(a) + b) <= eps)

Console.WriteLine("При b = " + b + ", значение G невозможно вычислить");

else

{

G = (2.4 \* Math.Cos(Math.Pow(a, 3.0) + Math.Pow(b, a)) + Math.Pow(b \* b - x, 1 / 3)) / (Math.Exp(a) + b);

if (Math.Abs(G) <= eps)

Console.WriteLine("При b = " + b + ", значение G = 0");

else

Console.WriteLine("При b = " + b + ", значение G = " + G);

}

b += delta\_b;

}

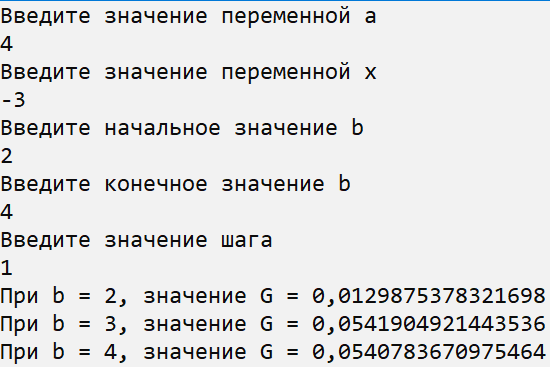
Console.ReadKey();

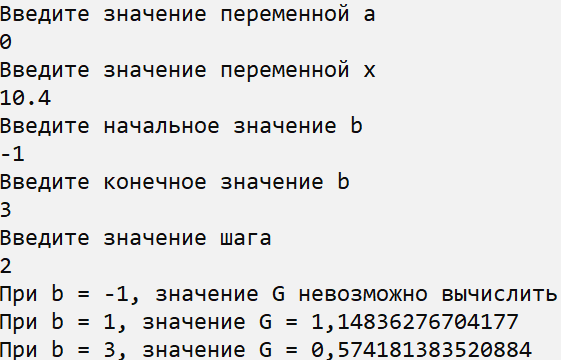
}

}

}

**6. Расчет тестовых примеров на ПК.**





**7. Вывод.**

В ходе лабораторной работы я решила поставленную задачу программирования по разделу «Циклы с пред- и постусловием» на языке Visual C#, не используя условный оператор. Результаты тестовых примеров совпали с результатами тестовых примеров на ПК, что говорит о правильности выполнения задачи.