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 № 4

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Составление последовательного вычислительного процесса на примере решения сложного арифметического выражения»

Выполнил: ст. гр. ТУУ-111

Богомолов В.Н.

Вариант №5

24.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

25.12.2024

(дата приёмки)

Москва – 2024 г.

1. **Цель работы**

Рассчитать результат сложного арифметического выражения на языке программирования *C#* в интегрированной среде разработки *Microsoft Visual Studio*. Реализовать два подхода: использование одной переменной для всего выражения и разбиение выражения на несколько шагов с использованием промежуточных переменных.

1. **Формулировка задачи**

Разработать программу на языке программирования *C#* для вычисления сложного арифметического выражения двумя способами:

* с использованием одной переменной для всего выражения;
* с пошаговым вычислением, где каждый этап записывается в отдельную переменную.

Программа должна быть реализована в интегрированной среде разработки Microsoft Visual Studio. Основной задачей является точное соответствие кода предоставленной блок-схеме, а также обеспечение ясности и читабельности алгоритма.

1. **Блок-схема**

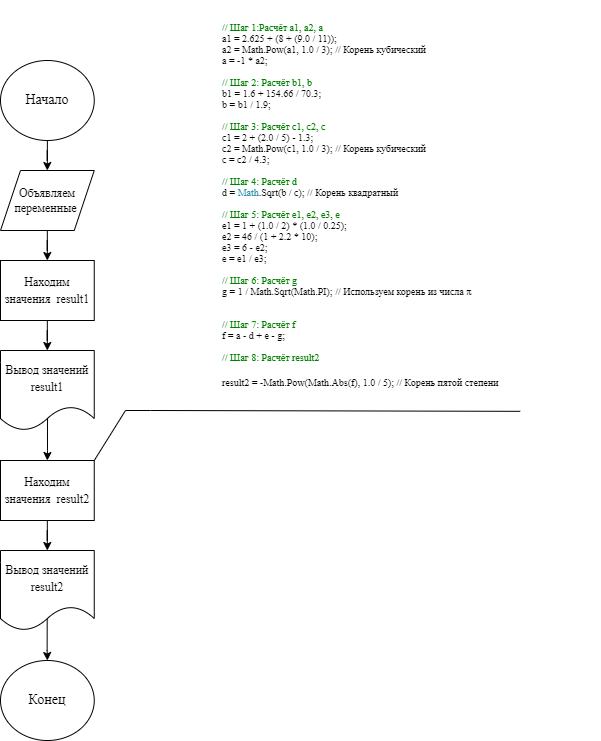
****

Рисунок 3.1 – Блок-схема алгоритма

1. **Подбор тестовых примеров**



Рисунок 4.1 – Тестовый пример

1. **Листинг (код прораммы)**

using System;

class Program

{

static void Main()

{

// Объявляем переменные согласно блок-схеме

double a1, a2, a, b1, b, c1, c2, c, d, e1, e2, e3, e, g, f, result1, result2;

// Расчёт первой переменной result1 (весь расчёт в одной строке)

result1 = -Math.Pow(

Math.Abs(

(-Math.Pow((2.625 + (8 + (9.0 / 11))), 1.0 / 3)) -

Math.Sqrt(((1.6 + 154.66 / 70.3) / 1.9) / (Math.Pow((2 + (2.0 / 5) - 1.3), 1.0 / 3) / 4.3)) +

((1 + (1.0 / 2) \* (1.0 / 0.25)) / (6 - 46 / (1 + 2.2 \* 10))) -

(1 / Math.Sqrt(Math.PI))

), 1.0 / 5

);

// Выводим результат первого метода

Console.WriteLine($"Результат (одной переменной): {result1}");

// Расчёт второй переменной result2 с использованием промежуточных вычислений

// Шаг 1: Расчёт a1, a2, a

a1 = 2.625 + (8 + (9.0 / 11));

a2 = Math.Pow(a1, 1.0 / 3); // Корень кубический

a = -1 \* a2;

// Шаг 3: Расчёт c1, c2, c

c1 = 2 + (2.0 / 5) - 1.3;

c2 = Math.Pow(c1, 1.0 / 3); // Корень кубический

c = c2 / 4.3;

// Шаг 5: Расчёт e1, e2, e3, e

e1 = 1 + (1.0 / 2) \* (1.0 / 0.25);

e2 = 46 / (1 + 2.2 \* 10);

e3 = 6 - e2;

e = e1 / e3;

// Шаг 6: Расчёт g

g = 1 / Math.Sqrt(Math.PI); // Используем корень из числа π

// Шаг 7: Расчёт f

f = a - d + e - g;

// Шаг 8: Расчёт result2

result2 = -Math.Pow(Math.Abs(f), 1.0 / 5); // Корень пятой степени

// Выводим результат второго метода

Console.WriteLine($"Результат (несколько переменных): {result2}");

// Ожидание ввода, чтобы программа не закрывалась сразу

Console.ReadKey();

}

}

1. **Тестирование (расчёт тестовых примеров ПК)**



Рисунок 6.1 – Тестирование

1. **Вывод**

В результате выполнения работы создана программа, которая выполняет расчёт сложного арифметического выражения двумя способами. Программа демонстрирует преимущества структурного подхода при использовании промежуточных переменных, что упрощает процесс понимания и проверки расчётов.