Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**о лабораторной работе**

Выполнение простой программы. Типы данных. Организация ввода и вывода данных

Выполнила студентка группы РИС-23-2Б

Жилина А. А.

Проверил доцент кафедры ИТАС

Викентьева О. Л.

Пермь 2024

**Постановка задачи**

**Общая постановка задачи №1:**

1. Найти значения выражений. Если необходимо, то определить при каких исходных данных выражение не может быть вычислено и выдать сообщение об ошибке.
2. Составить систему тестов и вычислить полученное выражение для нескольких значений Х.

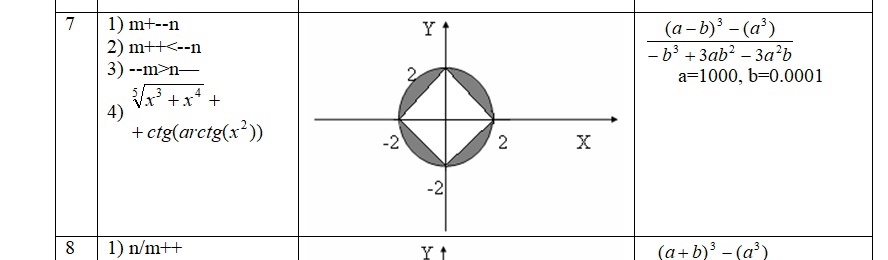
**Общая постановка задачи №2:**

1. Записать выражение, зависящее от координат точки X1 и Y1 и принимающее значение TRUE, если точка принадлежит заштрихованной области, и FALSE, если не принадлежит.
2. Составить систему тестов и вычислить полученное выражение.

**Общая постановка задачи №3:**

1. Вычислить значение выражения, используя различные вещественные типы данных (float и double).
2. Результаты всех вычислений вывести на печать.
3. Объяснить полученные результаты.

**Конкретная постановка для варианта 7:**



**Анализ классов входных и выходных данных**

**Задача №1**

Выражения 1-3:

*Входные данные:* Два целых числа m и n.

*Ограничения:* Отсутствуют. Вводятся любые целые числа.

*Выходные данные:* Три целых числа r1, r2 и r3, представляющие результаты вычисления выражений.

Выражение 4:

*Входные данные:* Одно число x, представляющее угол в радианах.

*Ограничения:* Угол x не может быть кратным Pi, так как в этом случае котангенс не определен.

*Выходные данные:* Если arctg(x^2), то выводится сообщение о том, что значение котангенса не может быть вычислено. В противном случае, выводится значение выражения ⁵√(x³ + x⁴) + ctg(arctg(x²)).

**Задача №2**

Уравнение x^2 + y^2 <= 4 && (y + x) >= 2 && (y - x) >= 2 && (y + x) <= -2 && (y - x) <= - 2 описывает область, ограниченную окружностью радиуса 2 с центром в начале координат и осью Oy.

*Входные данные:* Два числа с плавающей запятой: координата по оси X; координата по оси Y.

*Ограничения:* Отсутствуют. Пользователь может вводить любые действительные числа.

*Выходные данные:* Булевое значение true, если точка с координатами (x, y) принадлежит области, определяемой уравнением; false, если точка не принадлежит этой области.

**Задача №3**

*Входные данные:* Отсутствуют. В программе не используется ввод данных от пользователя.

*Ограничения:* Вычисления в программе выполняются с использованием фиксированных значений, что исключает ограничения.

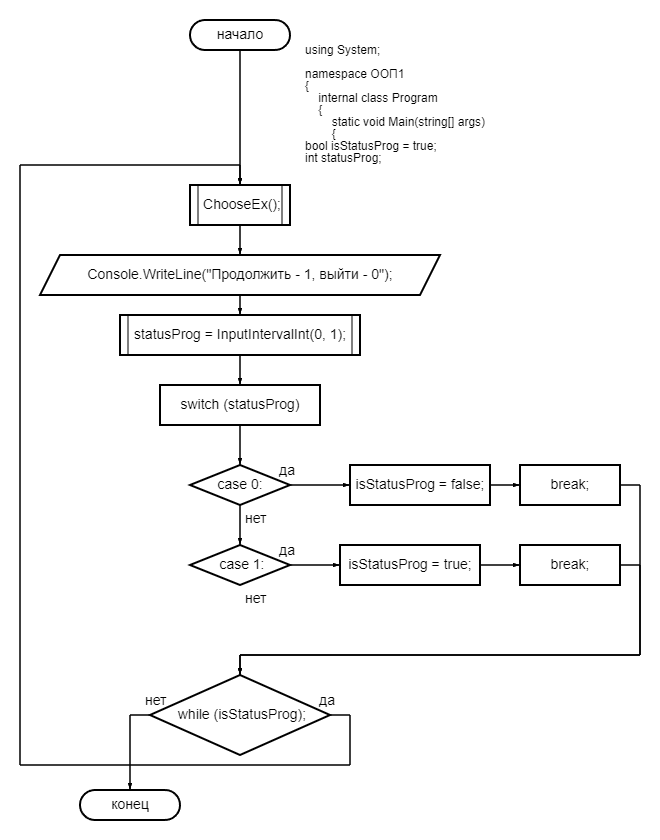
*Выходные данные:* Два числа с плавающей запятой: результат вычисления выражения с использованием типа float и результат вычисления того же выражения, но с использованием типа double.

В первом случае вычисления выполняются с использованием типа float. Данный тип данных имеет ограниченный диапазон и точность. В результате вычислений с использованием float получается приближенное значение.

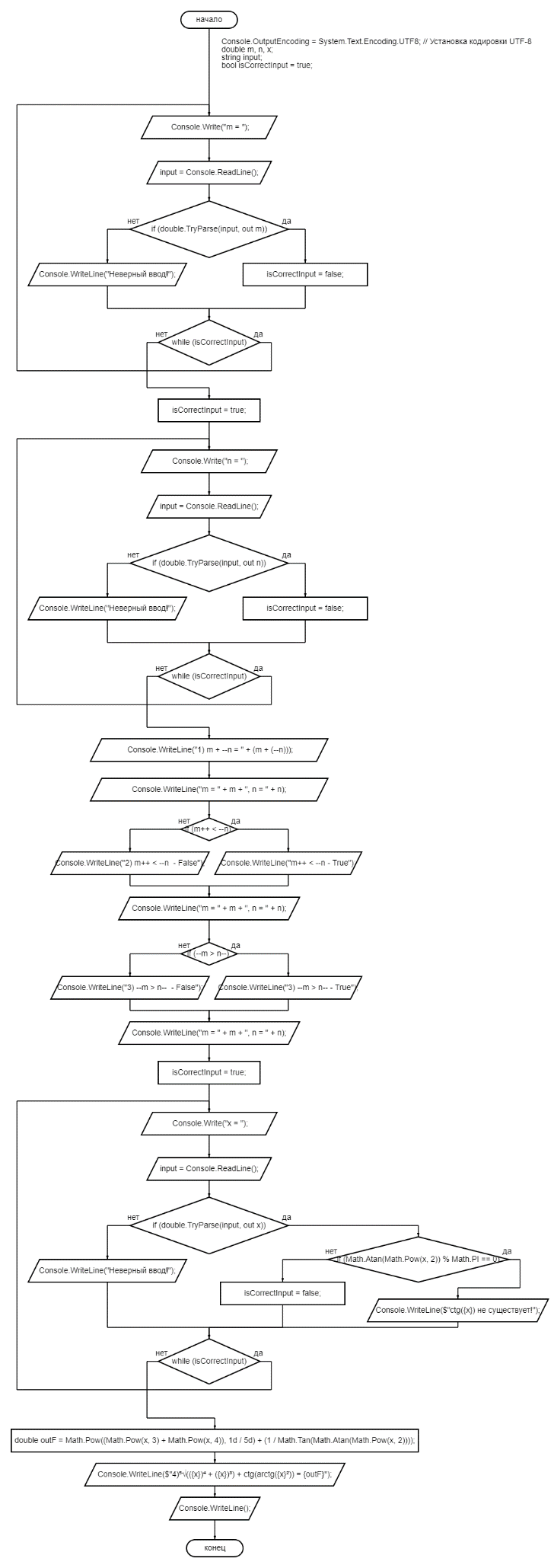
Во втором случае вычисления выполняются с использованием типа double. Данный тип данных имеет больший диапазон и точность, чем float. Результат вычислений с использованием double ближе к точному значению выражения.

Разница в результатах с использованием float и double показывает, как ограниченная точность float может привести к потерям точности при выполнении вычислений. В данном случае, использование double дает более точный результат.

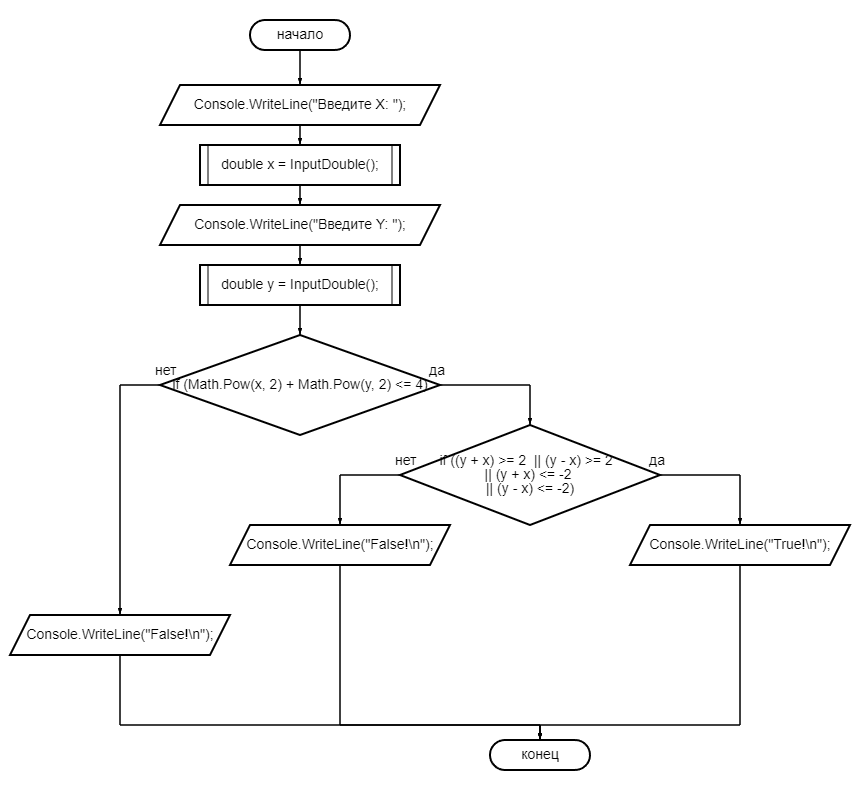
**Алгоритм работы программы**

****

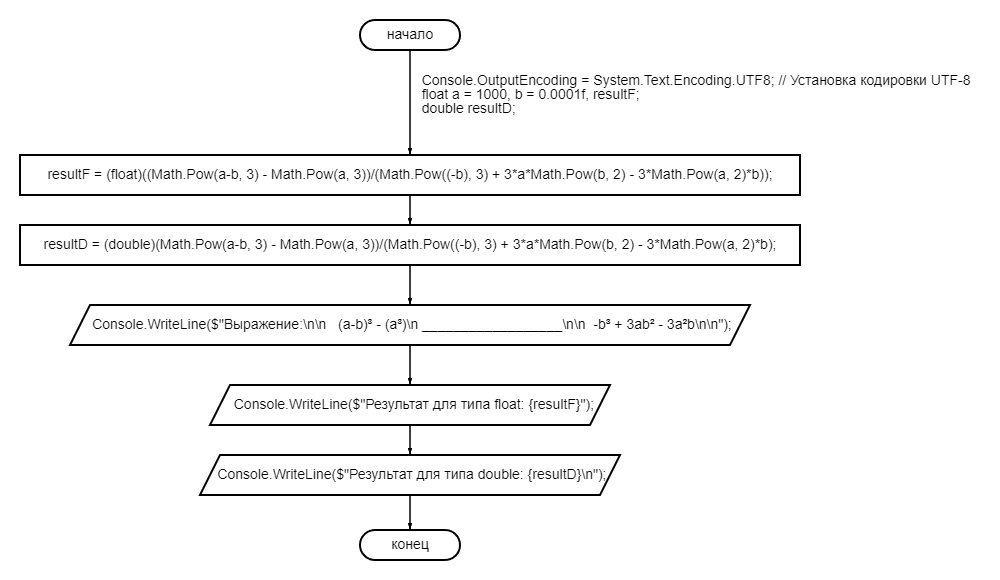
**Задача №1**



**Задача №2**



**Задача №3**



**Листинг**

using System;

namespace ООП1

{

internal class Program

{

static void Main(string[] args)

{

bool isStatusProg = true;

do

{

ChooseEx();

Console.WriteLine("Продолжить - 1, выйти - 0");

int statusProg = InputIntervalInt(0, 1);

switch (statusProg)

{

case 0:

{

isStatusProg = false;

break;

}

case 1:

{

isStatusProg = true;

break;

}

}

}

while (isStatusProg);

}

//проверка на ввод числа типа int

static int InputInt()

{

int number;

bool isCorrectInput = true;

do

{

string input;

input = Console.ReadLine();

if (int.TryParse(input, out number))

{

isCorrectInput = false;

}

else Console.WriteLine("Неверный ввод!");

}

while (isCorrectInput);

return number;

}

//проверка на ввод числа типа float

static float InputDouble()

{

float number;

bool isCorrectInput = true;

do

{

string input;

input = Console.ReadLine();

if (float.TryParse(input, out number))

{

isCorrectInput = false;

}

else Console.WriteLine("Неверный ввод!");

}

while (isCorrectInput);

return number;

}

//проверка на ввод числа типа int из интервала от left до right

static int InputIntervalInt(int left, int right)

{

bool isCorrectInput = true;

int input;

do

{

input = InputInt();

if (left <= input && input <= right)

isCorrectInput = false;

else Console.WriteLine($"Ввод должен находиться в интервале от {left} до {right}");

}

while (isCorrectInput);

return input;

}

static void ChooseEx()

{

Console.WriteLine("Введите номер задания (1-3): ");

int numberEx = InputIntervalInt(1, 3);

switch (numberEx)

{

case 1:

{

FirstEx();

break;

}

case 2:

{

SecondEx();

break;

}

case 3:

{

ThirdEx();

break;

}

}

}

static void FirstEx()

{

Console.OutputEncoding = System.Text.Encoding.UTF8; // Установка кодировки UTF-8

double m, n, x;

string input;

bool isCorrectInput = true;

do

{

Console.Write("m = ");

input = Console.ReadLine();

if (double.TryParse(input, out m))

{

isCorrectInput = false;

}

else Console.WriteLine("Неверный ввод!");

}

while (isCorrectInput);

isCorrectInput = true;

do

{

Console.Write("n = ");

input = Console.ReadLine();

if (double.TryParse(input, out n))

{

isCorrectInput = false;

}

else Console.WriteLine("Неверный ввод!");

}

while (isCorrectInput);

Console.WriteLine("1) m + --n = " + (m + (--n)));

Console.WriteLine("m = " + m + ", n = " + n);

if (m++ < --n)

Console.WriteLine("m++ < --n - True");

else Console.WriteLine("2) m++ < --n - False");

Console.WriteLine("m = " + m + ", n = " + n);

if (--m > n--)

Console.WriteLine("3) --m > n-- - True");

else Console.WriteLine("3) --m > n-- - False");

Console.WriteLine("m = " + m + ", n = " + n);

isCorrectInput = true;

do

{

Console.Write("x = ");

input = Console.ReadLine();

if (double.TryParse(input, out x))

{

if (Math.Atan(Math.Pow(x, 2)) % Math.PI == 0)

Console.WriteLine($"ctg({x}) не существует!");

else isCorrectInput = false;

}

else Console.WriteLine("Неверный ввод!");

}

while (isCorrectInput);

double outF = Math.Pow((Math.Pow(x, 3) + Math.Pow(x, 4)), 1d / 5d) + (1 / Math.Tan(Math.Atan(Math.Pow(x, 2))));

Console.WriteLine($"4)⁵√(({x})⁴ + ({x})³) + ctg(arctg({x}²)) = {outF}");

Console.WriteLine();

}

static void SecondEx()

{

Console.WriteLine("Введите X: ");

double x = InputDouble();

Console.WriteLine("Введите Y: ");

double y = InputDouble();

if (Math.Pow(x, 2) + Math.Pow(y, 2) <= 4)

{

if ((y + x) >= 2

|| (y - x) >= 2

|| (y + x) <= -2

|| (y - x) <= -2)

Console.WriteLine("True!\n");

else

Console.WriteLine("False!\n");

}

else

Console.WriteLine("False!\n");

}

static void ThirdEx()

{

Console.OutputEncoding = System.Text.Encoding.UTF8; // Установка кодировки UTF-8

float a = 1000, b = 0.0001f, resultF;

double resultD;

resultF = (float)((Math.Pow(a-b, 3) - Math.Pow(a, 3))/(Math.Pow((-b), 3) + 3\*a\*Math.Pow(b, 2) - 3\*Math.Pow(a, 2)\*b));

resultD = (double)(Math.Pow(a-b, 3) - Math.Pow(a, 3))/(Math.Pow((-b), 3) + 3\*a\*Math.Pow(b, 2) - 3\*Math.Pow(a, 2)\*b);

Console.WriteLine($"Выражение:\n\n (a-b)³ - (a³)\n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n -b³ + 3ab² - 3a²b\n\n");

Console.WriteLine($"Результат для типа float: {resultF}");

Console.WriteLine($"Результат для типа double: {resultD}\n");

}

}

}