Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

# Лабораторная работа №1

Выполнение простой программы. Типы данных.

Организация ввода и вывода данных

Выполнил:

студент группы РИС-23-2б

Борисов Н.А.

Проверила:

доцент кафедры ИТАС

Викентьева О.Л.

Пермь 2024

**Цель задания:**

1. Организация ввода и вывода данных.
2. Вычисление выражений простых типов с использованием стандартной библиотеки Math для вычисления математических функций.
3. Организация тестирования программы с проверкой достаточности тестов.

**Задания по варианту №2:**

|  |  |  |
| --- | --- | --- |
| Задача 1 | Задача 2 | Задача 3 |
| 1. ++n\*++m 2. m++<n 3. n++>m   4) |  | а=1000, b=0.0001 |

# Задача 1.

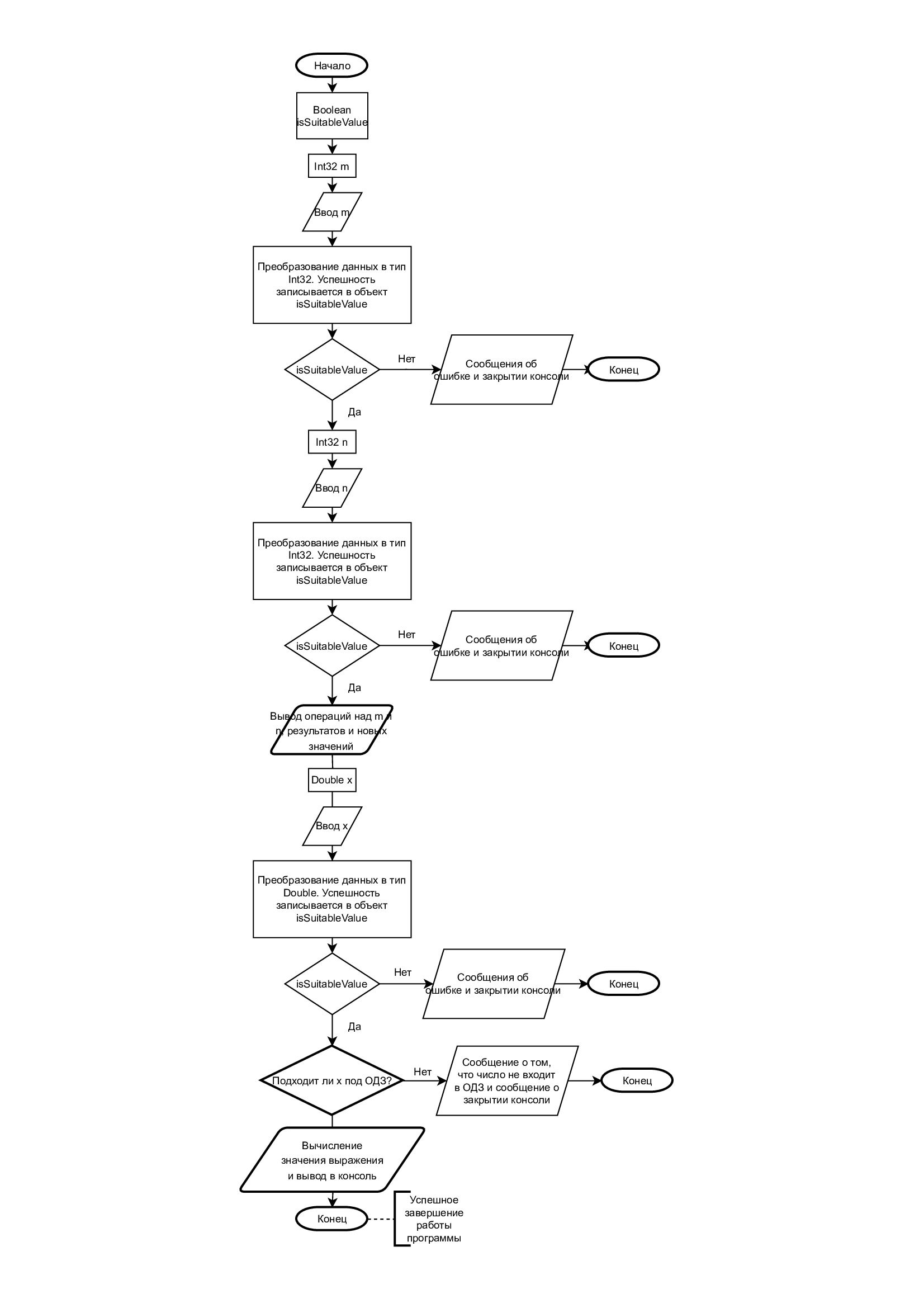
**Постановка задачи:**

* Для задачи 1 найти значения выражений. Определить при каких исходных данных выражение не может быть вычислено и выдать сообщение об ошибке.
* Составить систему тестов и вычислить полученное выражение для нескольких значений Х.

**Анализ классов входных и выходных данных:**

* Для первых трёх вычислений все вводимые данные должны стать числом Int32;
* Для четвёртого вычисления потребуется ввести число x типа Double, а также сделать проверку принадлежности x к ОДЗ;
* Выходные данные для примеров 1 и 4, будут иметь числовой вид;
* Выходные данные для примеров 2 и 3 будут логическими значениями.

**Блок-схема алгоритма действий:**

****

**Код программы на C#:**

String messageConversionError = "\nНевозможно преобразовать входные данные в указанный тип.";

String messageCloseConsole = "Нажмите любую клавишу, чтобы закрыть консоль.";

Console.Write("Введите число m типа Int32: ");

Int32 m;

Boolean isSuitableValue = Int32.TryParse(Console.ReadLine(), out m);

if (!isSuitableValue)

{

Console.WriteLine(messageConversionError);

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

Console.Write("Введите число n типа Int32: ");

Int32 n;

isSuitableValue = Int32.TryParse(Console.ReadLine(), out n);

if (!isSuitableValue)

{

Console.WriteLine(messageConversionError);

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

Console.WriteLine("\nВыполнение операций над m и n:");

Console.WriteLine($"1. ++n \* ++m = {++n \* ++m}");

Console.WriteLine($"Новые значения: m = {m}, n = {n}");

Console.WriteLine($"2. m++ < n = {m++ < n}");

Console.WriteLine($"Новые значения: m = {m}, n = {n}");

Console.WriteLine($"3. n++ > m = {n++ > m}");

Console.WriteLine($"Новые значения: m = {m}, n = {n}");

Console.Write("\nВведите число x типа Double: ");

Double x;

isSuitableValue = Double.TryParse(Console.ReadLine(), out x);

if (!isSuitableValue)

{

Console.WriteLine(messageConversionError);

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

if (x == -1 || x == 0 || x == 1)

{

Console.WriteLine("\nДанное число не входит в ОДЗ.");

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

Console.WriteLine("\nВычисление значения выражения:");

Console.WriteLine($"4. x + 1 / (x^3 - x) - 2 = {x + 1 / (Math.Pow(x, 3) - x) - 2}");

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

**Объяснение результатов работы программы**

при :

1. ++n \* ++m = 6. Производится префиксный инкремент для n и m, затем находится их произведение. Новые значения: m = 2, n = 3.
2. m++ < n = True. Сначала вычисляется значение логического выражения, только потом происходит постфиксный инкремент для m. Новые значения: m = 3, n = n++ > m = False.
3. Сначала вычисляется значение логического выражения, только потом происходит постфиксный инкремент для n. Новые значения: m = 3, n = 4.
4. Значение выражение, в которое подставляется x будет вычислено верно, т.к. x принадлежит ОДЗ.

# Задача 2.

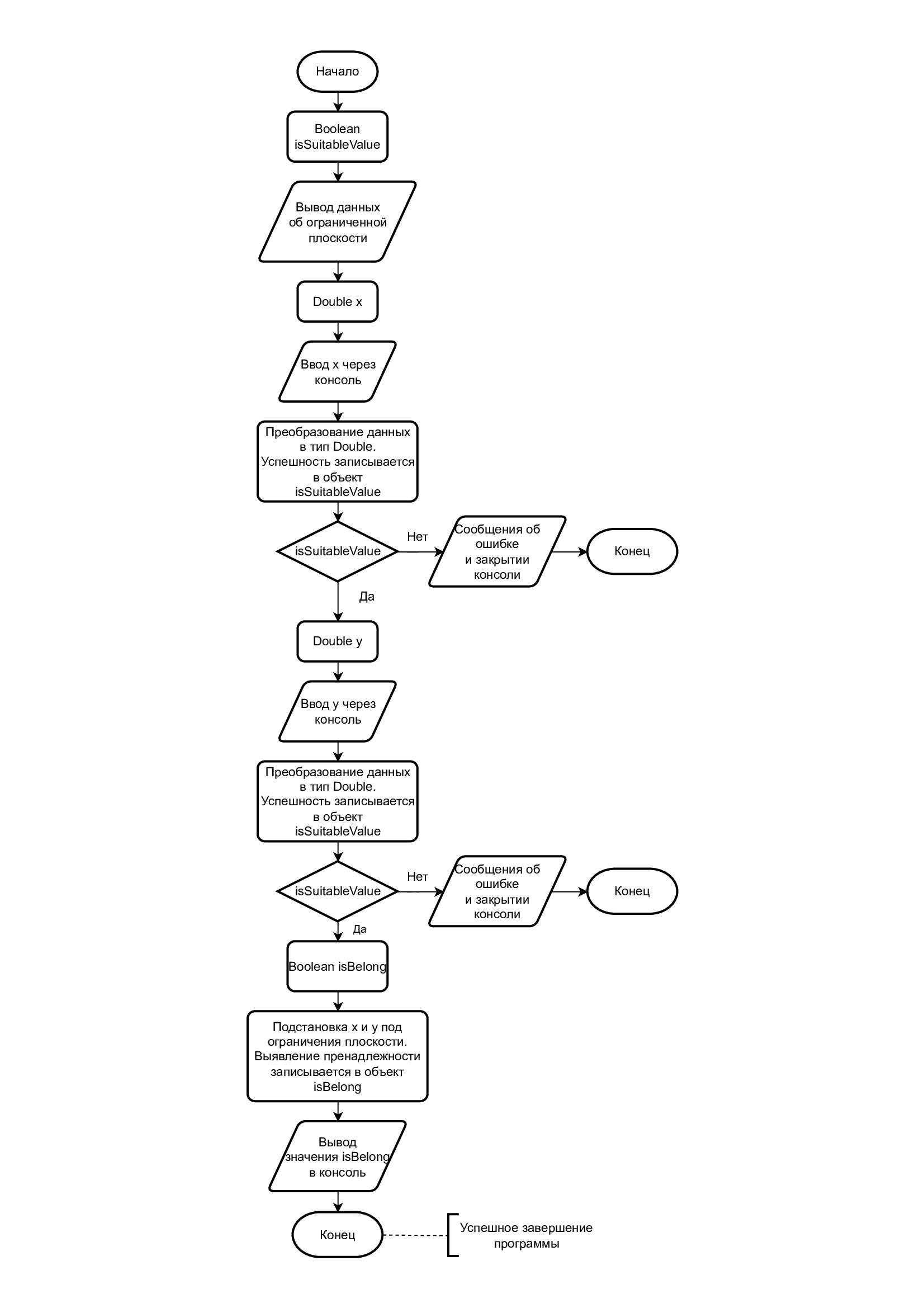
**Постановка задачи:**

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

**Анализ классов входных и выходных данных:**

* Входные данные – координаты x и y, – должны принадлежать типу Double;
* Полученное значение – логическая величина класса Boolean, которая зависит от координат x и y. Плоскость ограничена фигурой, заданной неравенствами: y <= 0; y >= -2; x >= -7; x <= 0.

**Блок-схема алгоритма действий:**



**Код программы на C#:**

String messageConversionError = "\nНевозможно преобразовать входные данные в указанный тип.";

String messageCloseConsole = "Нажмите любую клавишу, чтобы закрыть консоль.";

Console.WriteLine("Область на плоскости задана неравенствами:");

Console.WriteLine("1. y <= 0;");

Console.WriteLine("2. y >= -2;");

Console.WriteLine("3. x >= -7;");

Console.WriteLine("4. x <= 0.");

Console.WriteLine("\nВведите координаты точки, чтобы определить её принадлежность плоскости.");

Console.Write("Введите x типа Double: ");

Double x;

Boolean isSuitableValue = Double.TryParse(Console.ReadLine(), out x);

if (!isSuitableValue)

{

Console.WriteLine(messageConversionError);

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

Console.Write("Введите y типа Double: ");

Double y;

isSuitableValue = Double.TryParse(Console.ReadLine(), out y);

if (!isSuitableValue)

{

Console.WriteLine(messageConversionError);

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

return;

}

bool conditionOne = y <= 0;

bool conditionTwo = y >= -2;

bool conditionThree = x >= -7;

bool conditionFour = x <= 0;

bool isBelong = conditionOne && conditionTwo && conditionThree && conditionFour;

Console.WriteLine($"\nОтвет: {isBelong}");

Console.WriteLine(messageCloseConsole);

Console.ReadKey();

# Задача 3.

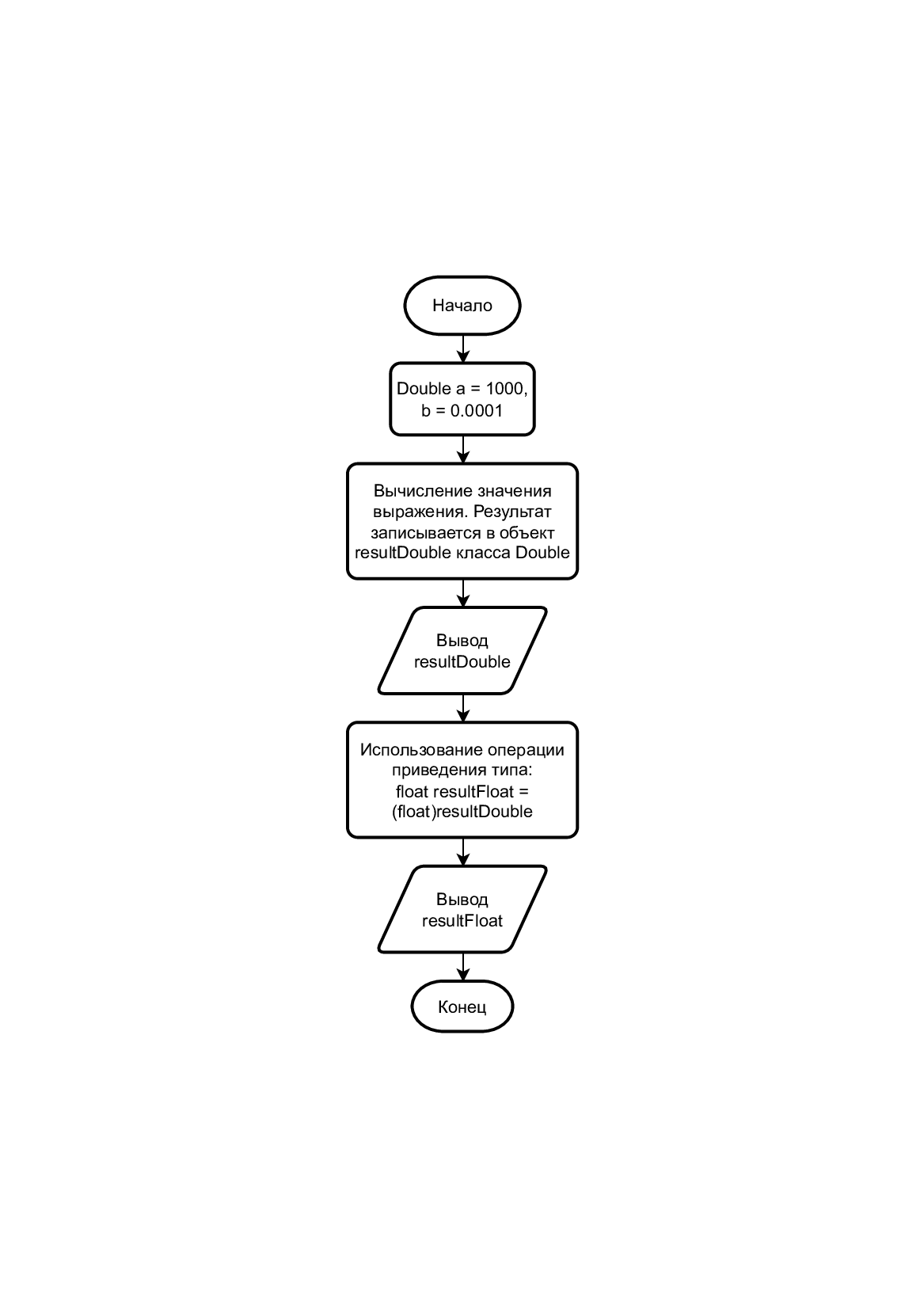
**Постановка задачи:**

* Для задачи 3 вычислить значение выражения, используя различные вещественные типы данных (float и double).
* Результаты всех вычислений вывести на печать.
* Объяснить полученные результаты.

**Анализ классов входных и выходных данных:**

* Входные данные – объекты класса Double a = 1000 и b = 0.0001.
* Выведенные значения – результат вычисления заданного выражения, записанный сначала в Double, а затем приведённый к типу float.

**Блок-схема алгоритма действий:**



**Код программы на C#:**

Double a = 1000, b = 0.0001;

Double resultDouble = (Math.Pow((a + b), 2) - (Math.Pow(a, 2) + 2 \* a \* b)) / Math.Pow(b, 2);

Console.WriteLine($"Результат вычислений точности double: {resultDouble}");

float resultFloat = (float)resultDouble;

Console.WriteLine($"Результат вычислений точности float: {resultFloat}");

Console.WriteLine("\nНажмите любую клавишу, чтобы закрыть консоль.");

Console.ReadKey();

**Объяснение результатов работы программы:**

Первое значение, которое выведется в консоль будет типа Double. Оно будет длиннее второго значения (типа float), т.к. тип Double хранит 8 байт информации, а тип float лишь 4 байта, поэтому при переходе к типу float теряется точность результата.