Пермский филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет «Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Шульжик Кирилл Владимирович

**Отчёт по лабораторной работе №1 (Вариант 16)**

студента образовательной программы бакалавриата «Программная инженерия» по направлению подготовки 09.03.04 Программная инженерия

Преподаватель

В.Д Марквирер

Оглавление

[**Задача 1** 3](#_Toc86342870)

[**Тесты** 4](#_Toc86342871)

[**Тестирование первого выражения** 4](#_Toc86342872)

[**Тестирование второго выражения** 5](#_Toc86342873)

[**Тестирование второго выражения** 6](#_Toc86342874)

[**Тестирование четвертого выражения** 8](#_Toc86342875)

[**Программа 1 выражения** 10](#_Toc86342876)

[**ИСПРАВИЛ ВЫРАЖЕНИЕ (ограничение n=1, вывод исправил)** 10](#_Toc86342877)

[**Программа 2 выражения** 10](#_Toc86342878)

[**Программа 3 выражения** 11](#_Toc86342879)

[**Программа 4 выражения** 12](#_Toc86342880)

[**ИСПРАВИЛ (while rez<0)** 12](#_Toc86342881)

[**Блок-схема к первому выражению** 13](#_Toc86342882)

[**Блок-схема ко второму выражению** 14](#_Toc86342883)

[**Блок-схема к третьему выражению** 15](#_Toc86342884)

[**Блок-схема к четвертому выражению** 16](#_Toc86342885)

[**Задача 2** 17](#_Toc86342886)

[**Анализ** 17](#_Toc86342887)

[**Код программы** 18](#_Toc86342888)

[**Система тестов** 19](#_Toc86342889)

[**Блок-схема ко второму заданию** 22](#_Toc86342890)

[**Задача 3** 23](#_Toc86342891)

[**Анализ** 23](#_Toc86342892)

[**Код программы (double)** 23](#_Toc86342893)

[**Код программы (float)** 25](#_Toc86342894)

[**Блок-схема к третьей задаче** 26](#_Toc86342895)

**Задача 1**

Определить тип заданных выражений и найти их значения. Составить систему тестов и вычислить полученное выражение для нескольких значений x. Определить, при каких х значение не может быть вычислено.

1) (m/--n)++

2) m/n<n—

3) m+n++>n+m

4) 

В первом и третьем выражении содержится инкремент, в первом и втором выражении – декремент (префиксные/постфиксные выражения).

3 и 2 выражение логического типа.

Все заданные выражения являются арифметическими

В первом и четвертом выражении переменные типа double. Во втором и третьем выражении переменная логического типа bool.

Стоит также отметить, что логическая переменная типа bool в третьем выражении будет всегда принимать значение False, поскольку до знака «>» стоит постфиксный инкремент, который увеличивает значение переменной n, использующейся в данном выражении после знака «>».

Ограничения в данных выражениях существуют для n в 1 и 2 выражении. n не может быть равно единице в 1 выражении, поскольку на единицу в таком случае накладывается префиксный декремент, и значение единицы становится равно 0. Деление на ноль не определено. Во втором выражении n не может быть равно 0 также из-за деления на ноль. Помимо этого, в 4 выражении sin(x) + |x²+x| не может быть меньше нуля. Соответственно при sin(x) + |x²+x| < 0 значение не может быть вычислено.

**Тесты**

**Тестирование первого выражения**

Составим систему тестов для выражения 1.

Таблица 1 – система тестов для выражения 1

|  |  |  |  |
| --- | --- | --- | --- |
| Тест, № | Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 | m = 4  n = 4 | m = 4  n = 4  (4 / --4)++ = 1,3333333333333333 | m = 4  n = 4  (4 / --4)++ = 1,3333333333333333 |
| 2 | m = -4  n = -123 | m = -4  n = -123  (-4 / -- -123)++ = 0,03225806451612903 | m = -4  n = -123  (-4 / -- -123)++ = 0,03225806451612903 |
| 3 | m = 4  n = 1 | Ошибка. Деление на ноль. Введите число n заново. | Ошибка. Деление на ноль. Введите число n заново. |
| n = fs | Введите число! | Введите число! |
| n = 4 | m = 4  n = 4  (4 / --4)++ = 1,3333333333333333 | m = 4  n = 4  (4 / --4)++ = 1,3333333333333333 |
| 4 | m = 5,4  n = 4 | m = 5,4  n = 4  (5,4 / --4)++ = 1,8 | m = 5,4  n = 4  (5,4 / --4)++ = 1,8 |
| 5 | m = «абщихба»  n = «ыыы» | Введите число! | Введите число! |

В таблице 1 представлена система тестов для первого выражения первой задачи.

Произведём тестирование по критериям черного ящика для первого выражения первой задачи.

Таблица 2 – Тестирование по критериям ЧЯ для первого выражения первой задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Проверяемая ситуация | Т1 | Т2 | Т3 | Т4 | Т5 |
| Характеристика классов входных данных | | | | | |
| m и n – положительные числа | + |  |  | + |  |
| m и n – отрицательные числа |  | + |  |  |  |
| n = 1 |  |  | + |  |  |
| m или n – не целое число |  |  |  | + |  |
| m и n – не числа |  |  |  |  | + |

В таблице 2 представлено тестирование методом черного ящика для первого выражения первой задачи

Произведём тестирование по критериям белого ящика (МГТ) для первого выражения первой задачи

Таблица 3 – Тестирование по критериям белого ящика (МГТ) для первого выражения первой задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | Т1 | Т2 | Т3 | Т4 | Т5 |
| while (n==1) |  |  | + |  |  |
| while (!Double.TryParse(bug, out m)) |  |  |  |  | + |
| while (!Double.TryParse(bug1, out n)) |  |  | + |  | + |

В таблице 3 представлено тестирование методом белого ящика (МГТ) для первого выражения первой задачи

**Тестирование второго выражения**

Составим систему тестов для выражения 2.

Таблица 4 – система тестов для второго выражения первой задачи

|  |  |  |  |
| --- | --- | --- | --- |
| Тест, № | Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 | m = 4  n = 4 | m = 4  n = 4  4 / 4 < 4 = True | m = 4  n = 4  4 / 4 < 4 = True |
| 2 | m = 4  n = 1 | m = 4  n = 1  4 / 1 < 1 = False | m = 4  n = 1  4 / 1 < 1 = False |
| 3 | m = 4  n = 0 | Введите ненулевое число (n) | Введите ненулевое число (n) |
| 4 | m = 5,4  n = 4 | m = 5,4  n = 4  5,4 / 4 < 4 = True | m = 5,4  n = 4  5,4 / 4 < 4 = True. |
| 5 | m = «абщихба»  n = «ыыы» | Введите число! (Повторный ввод) | Введите число! (Повторный ввод) |

В таблице 4 представлена система тестов для второго выражения первой задачи

Произведём тестирование по критериям черного ящика для второго выражения первой задачи.

Таблица 5 – Тестирование по критериям ЧЯ для второго выражения первой задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Проверяемая ситуация | Т1 | Т2 | Т3 | Т4 | Т5 |
| Характеристика классов входных данных | | | | | |
| m и n не числа |  |  |  |  | + |
| m или n не целые числа |  |  |  | + |  |
| n = 0 |  |  | + |  |  |
| Характеристика классов выходных данных | | | | | |
| Результат = True | + |  |  | + |  |
| Результат = False |  | + |  |  |  |
| Результат = Введите число! |  |  |  |  | + |
| Результат = Введите ненулевое число! |  |  | + |  |  |

В таблице 5 представлено тестирование по критериям черного ящика для второго выражения первой задачи

Произведём тестирование по критериям белого ящика (МГТ) для второго выражения первой задачи

Таблица 6 – Тестирование по критериям белого ящика (МГТ) для второго выражения первой задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | Т1 | Т2 | Т3 | Т4 | Т5 |
| while (!Double.TryParse(bug, out m)) |  |  |  |  | + |
| while (!Double.TryParse(bug1, out n) || bug1 == “0”) |  |  | + |  | + |

В таблице 6 представлено тестирование по критериям белого ящика (МГТ) для второго выражения.

**Тестирование второго выражения**

Составим систему тестов для 3 выражения первой задачи

Таблица 7 – система тестов для выражения 3

|  |  |  |  |
| --- | --- | --- | --- |
| Тест, № | Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 | m = 4  n = 4 | m = 4  n = 4  4 + 4++ > 5 + 4 == False | m = 4  n = 4  4 + 4++ > 5 + 4 == False |
| 2 | m = 4  n = 5,4 | m = 4  n = 5,4  4 + 5,4++ > 6,4 + 4 == False | m = 4  n = 5,4  4 + 5,4++ > 6,4 + 4 == False |
| 3 | m = «абщихба»  n = «ыыы» | Введите число! (повторный ввод) | Введите число! (повторный ввод) |

В таблице 7 представлена система тестов для 3 выражения первой задачи.

Произведём тестирование по критериям черного ящика для третьего выражения первой задачи

Таблица 8 – Тестирование по критериям черного ящика для третьего выражения первой задачи

|  |  |  |  |
| --- | --- | --- | --- |
| Проверяемая ситуация | Т1 | Т2 | Т3 |
| Характеристика классов входных данных | | | |
| m и n – целые числа | + |  |  |
| m и/или n – вещественные числа |  |  |  |
| m и/или n – не числа |  |  | + |
| Характеристика классов выходных данных | | | |
| Результат = False | + | + |  |

В таблице 8 представлено тестирование по критериям черного ящика для третьего выражения первой задачи.

Произведём тестирование по критериям белого ящика (МГТ) для третьего выражения первой задачи.

Таблица 9 – Тестирование по критериям белого ящика (МГТ) для третьего выражения первой задачи

|  |  |  |  |
| --- | --- | --- | --- |
| Операторы | Т1 | Т2 | Т3 |
| while (!Double.TryParse(bug, out m)){} |  |  | + |
| while (!Double.TryParse(bug1, out n)){} |  |  | + |

В таблице 9 представлено тестирование по критериям белого ящика (МГТ) для третьего выражения первой задачи.

**Тестирование четвертого выражения**

Составим систему тестов для выражения 4.

Таблица 10 – система тестов для выражения 4

|  |  |  |  |
| --- | --- | --- | --- |
| Тест, № | Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 | x = 5 | x = 5  Значение выражения 5,388977242978195 | x = 5  Значение выражения 5,388977242978195 |
| 2 | x = -1 | Результат не вещественное число. Введите другой x. (повторный ввод) | Результат не вещественное число. Введите другой x. (повторный ввод) |
| x = «баба» | Введите число! (повторный ввод) | Введите число! (повторный ввод) |
| x = 4 | x = 4  Значение выражения 4,386706908911521 | x = 4  Значение выражения 4,386706908911521 |
| 3 | x = «Эаа» | Введите число! (повторный ввод) | Введите число! (повторный ввод) |
| 4 | x = 5,5 | x = 5,5  Значение выражения 5,91983611888282 | x = 5,5  Значение выражения 5,91983611888282 |

В таблице 10 представлена система тестов для 4 выражения первой задачи.

Произведём тестирование по критериям черного ящика для четвертого выражения первой задачи.

Таблица 11 – Тестирование по критериям черного ящика для четвертого выражения первой задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Проверяемая ситуация | Т1 | Т2 | Т3 | Т4 |
| Характеристика классов входных данных | | | | |
| Подкоренное выражение – неотрицательное число (sin(x) + |x²+x|>=0) | + |  |  | + |
| Подкоренное выражение – отрицательное число |  | + |  |  |
| x – не число |  | + | + |  |
| x – вещественное число |  |  |  | + |
| Характеристика классов выходных данных | | | | |
| Значение выражения равно rez | + | + |  | + |
| Результат не вещественное число. Введите другой x. (повторный ввод) |  | + |  |  |
| Введите число! (повторный ввод) |  | + | + |  |

В таблице 11 представлено тестирование по критериям черного ящика для четвертого выражения первой задачи.

Произведём тестирование по критериям белого ящика (МГТ) для четвертого выражения первой задачи.

Таблица 12 – Тестирование по критериям белого ящика (МГТ) для четвертого выражения первой задачи.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Операторы | Т1 | Т2 | Т3 | Т4 |
| while (!Double.TryParse(bug, out x)){} |  | + | + |  |
| while (rez<0){} |  | + |  |  |

В таблице 12 представлено тестирование по критериям белого ящика (МГТ) для четвертого выражения первой задачи.

**Программа 1 выражения**

**ИСПРАВИЛ ВЫРАЖЕНИЕ (ограничение n=1, вывод исправил)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication9

{

class Program

{

static void Main(string[] args)

{

double m;

double n;

double rez;

Console.WriteLine("Введите переменную m");

string bug = Console.ReadLine();

while (!Double.TryParse(bug, out m))

{

Console.WriteLine("Введите число!");

bug = Console.ReadLine();

}

m = Convert.ToDouble(bug);

Console.WriteLine("Введите переменную n");

string bug1 = Console.ReadLine();

while (!Double.TryParse(bug1, out n))

{

Console.WriteLine("Введите число");

bug1 = Console.ReadLine();

}

n = Convert.ToDouble(bug1);

while (n == 1)

{

Console.WriteLine("Ошибка. Деление на ноль. Введите значение n заново");

bug1 = Console.ReadLine();

while (!Double.TryParse(bug1, out n))

{

Console.WriteLine("Введите число!");

bug1 = Console.ReadLine();

}

n = Convert.ToDouble(bug1);

}

rez = (m / --n);

rez = rez++;

Console.WriteLine("m = " + m + "\nn = " + (n + 1) + "\n" + "(" + m + " / " + "--" + (n + 1) + ")" + "++ = " + rez);

}

}

}

**Программа 2 выражения**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication9

{

class Program

{

static void Main(string[] args)

{

double m;

double n;

bool rez;

Console.WriteLine("Введите переменную m");

string bug = Console.ReadLine();

while (!Double.TryParse(bug, out m))

{

Console.WriteLine("Введите число!");

bug = Console.ReadLine();

}

m = Convert.ToDouble(bug);

Console.WriteLine("Введите переменную n");

string bug1 = Console.ReadLine();

while (!Double.TryParse(bug1, out n) || bug1 == "0")

{

Console.WriteLine("Введите ненулевое число!");

bug1 = Console.ReadLine();

}

n = Convert.ToDouble(bug1);

rez = (m / n < n--);

Console.WriteLine("m = " + m + "\nn = " + (n + 1) + "\n" + m +

" / " + (n + 1) + " < " + (n + 1) + "-- = " + rez);

}

}

}

**Программа 3 выражения**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication9

{

class Program

{

static void Main(string[] args)

{

double m;

double n;

bool rez;

Console.WriteLine("Введите переменную m");

string bug = Console.ReadLine();

while (!Double.TryParse(bug, out m))

{

Console.WriteLine("Введите число!");

bug = Console.ReadLine();

}

m = Convert.ToDouble(bug);

Console.WriteLine("Введите переменную n");

string bug1 = Console.ReadLine();

while (!Double.TryParse(bug1, out n))

{

Console.WriteLine("Введите число!");

bug1 = Console.ReadLine();

}

n = Convert.ToDouble(bug1);

rez = (m + n++ > n + m);

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

+ " > " + n + " + " + m + " == " + rez);

}

}

}

**Программа 4 выражения**

**ИСПРАВИЛ (while rez<0)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication9

{

class Program

{

static void Main(string[] args)

{

double x;

double rez;

Console.WriteLine("Введите числовую переменную x");

string bug = Console.ReadLine();

while (!Double.TryParse(bug, out x))

{

Console.WriteLine("Введите число!");

bug = Console.ReadLine();

}

x = Convert.ToDouble(bug);

rez = Math.Sin(x) + Math.Abs(Math.Pow(x, 2) + x);

Console.WriteLine(rez);

while (rez<0)

{

Console.WriteLine("Результат не вещественное число, введите другой x");

bug = Console.ReadLine();

while (!Double.TryParse(bug, out x))

{

Console.WriteLine("Введите число!");

bug = Console.ReadLine();

}

x = Convert.ToDouble(bug);

rez = Math.Sin(x) + Math.Abs(Math.Pow(x, 2) + x);

}

rez = Math.Sqrt(rez);

Console.WriteLine("x = " + x + "\nЗначение выражения " + rez);

}

}

}

**Блок-схема к первому выражению**

Составим блок схему к первому выражению первой задачи.

Ввод m, n

Начало

Нет

Да

Ошибка. Деление на ноль. Введите значение n заново.

rez = m/ - -n++

Вывод rez

Конец

n и/или m – числа?

n = 1?

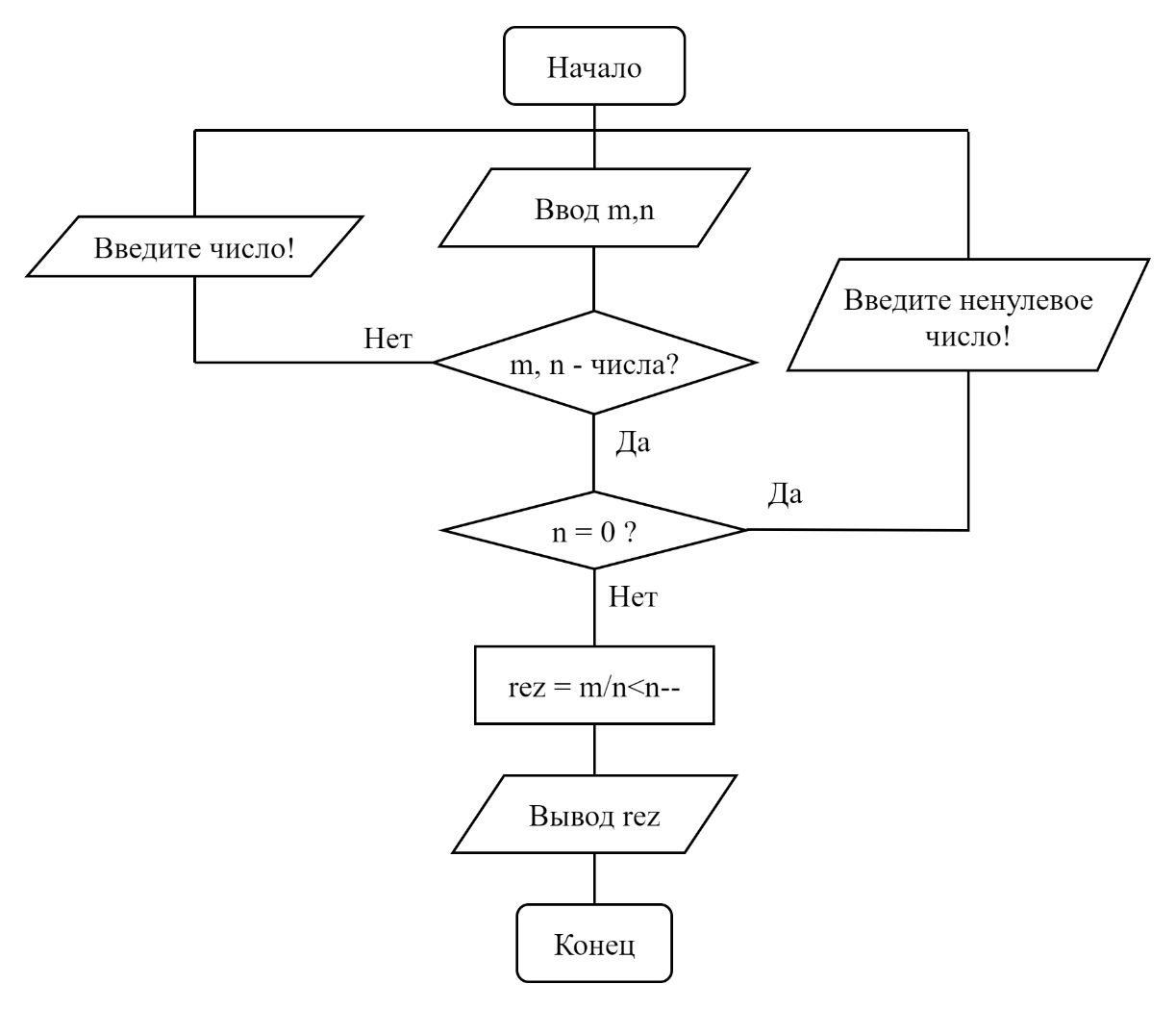
Введите число!

**Рисунок 1 – Блок схема первого выражения**

Мы составили блок-схему (рисунок 1). Суть программы заключается в воде пользователем переменных m и n, последующем вычислении переменной rez на основе полученных данных m и n, вывод значения на экран и завершения программы.

**Блок-схема ко второму выражению**

Составим блок-схему ко второму выражению первой задачи.



**Рисунок 2 – Блок-схема второго выражения**

На рисунке 2 представлена Блок-схема к второму выражению первой задачи. Суть программы аналогична первому выражению.

**Блок-схема к третьему выражению**

Составим блок-схему к третьему выражению первой задачи

Ввод m, n

rez = m+n++>n+m

Вывод rez

Конец

Начало

m и/или n – числа?

Нет

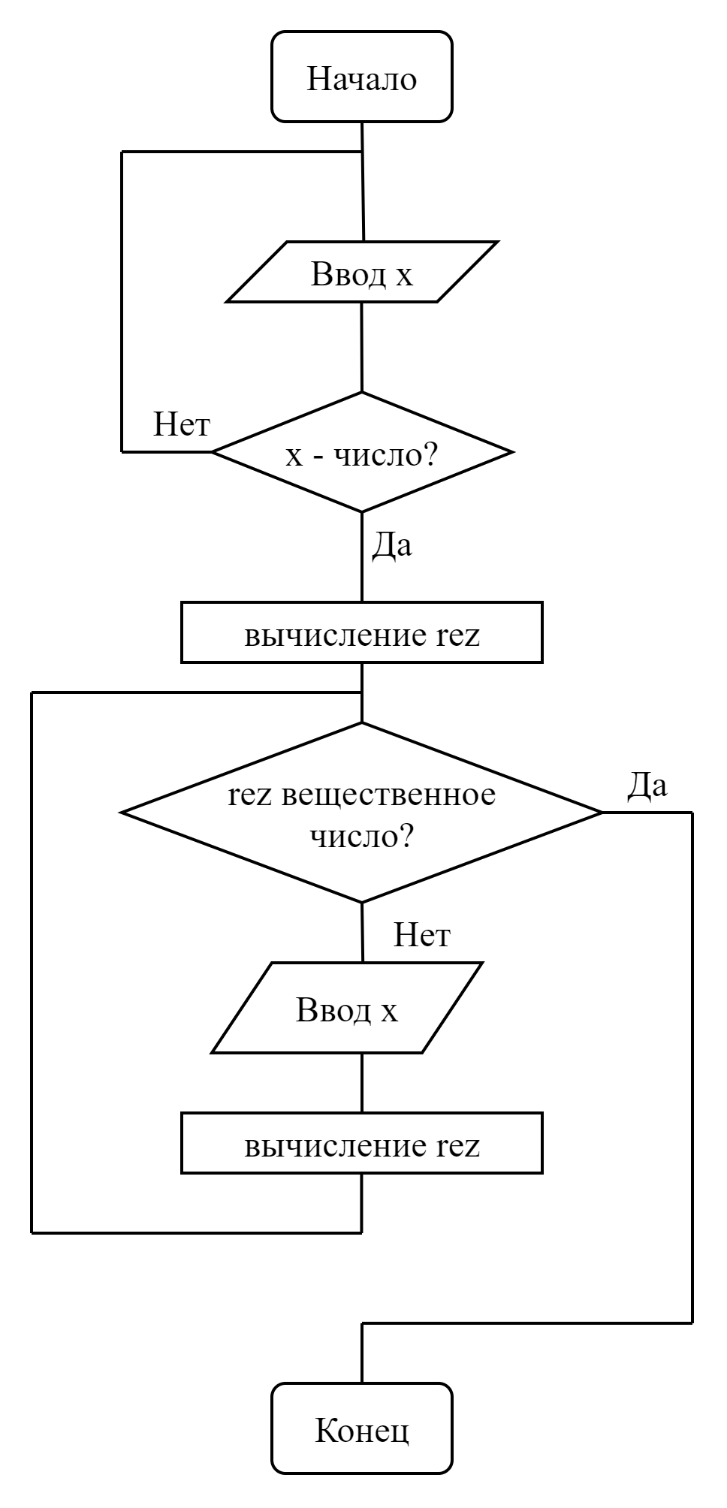
Введите число!

**Рисунок 3 – Блок-схема третьего выражения**

На рисунке 3 представлена Блок-схема к третьему выражению. Суть программы аналогична первому выражению.

**Блок-схема к четвертому выражению**

Составим блок-схему к четвертому выражению первой задачи (рисунок 4).

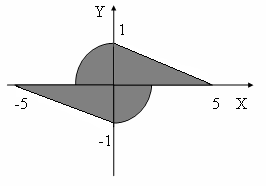


**Рисунок 4 – Блок-схема четвертого выражения**

На рисунке 4 изображена Блок-схема к четвертому выражению. Суть программы аналогична первому выражению, за исключением ввода переменных. Здесь вводится переменная x. m и n здесь нет.

**Задача 2**

Записать выражение, зависящее от координат точки x1 и y1 и принимающее значение TRUE, если точка принадлежит заштрихованной области (рисунок 5), и FALSE, если не принадлежит. Составить систему тестов и вычислить полученное выражение для нескольких точек, принадлежащих и не принадлежащих заштрихованной области.



**Рисунок 5 – Иллюстрация к задаче 2**

**Анализ**

Исходные данные: x, y – вещественные.

Выходные данные: Точка принадлежит заштрихованной области \ Точка не принадлежит заштрихованной области.

Анализ:

Уравнение прямой с точками (0;1) и (5;0) имеет следующий вид: y = -0,2x + 1. Прямая лежит в первой четверти и ограничена осью абсцисс и ординат, соответственно первое условие на попадание в заштрихованную область y <= -0.2 \* x + 1 && x >= 0 && x <= 5 && y >= 0 && y <= 1.

Аналогично вторая прямая, расположенная в 3 четверти, задаётся следующим уравнением: y = -0,2x – 1. Тогда второе условие на попадание в заштрихованную область имеет вид: y> = -0.2 \* x - 1 && y> = -1 && y <= 0 && x <= 0 && x> = -5.

Окружности, расположенные во 2 и 4 четверти, не пересекают прямые, лежащие в 1 и 3 четверти. Значит, последние условие можно представить как Math.Pow(x, 2) + y \* y <= 1 (использовалось уравнение окружности).

**Код программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace рар

{

class Program

{

static void Main(string[] args)

{

double x;

double y;

bool OK;

Console.WriteLine("Введите координату x");

string xix = Console.ReadLine();

while (!Double.TryParse(xix, out x))

{

Console.WriteLine("Введите число!");

xix = Console.ReadLine();

}

x = Convert.ToDouble(xix);

Console.WriteLine("Введите координату y");

xix = Console.ReadLine();

while (!Double.TryParse(xix, out y))

{

Console.WriteLine("Введите число!");

xix = Console.ReadLine();

}

y = Convert.ToDouble(xix);

OK = ((y <= -0.2 \* x + 1 && x >= 0 && x <= 5 && y >= 0 && y <= 1)

|| (y >= -0.2 \* x - 1 && y >= -1 && y <= 0 && x <= 0 && x >= -5)

|| (Math.Pow(x, 2) + y \* y <= 1));

if (OK) Console.WriteLine("Точка принадлежит заштрихованной области");

else Console.WriteLine("Точка не принадлежит заштрихованной области");

}

}

}

**Система тестов**

Составим систему тестов для 2 задачи

|  |  |  |  |
| --- | --- | --- | --- |
| Тест, № | Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 | x = 0  y = 0 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 2 | x = 0  y = 1 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 3 | x = 2  y = 0,3 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 4 | x = 10  y = 3 | Точка не принадлежит заштрихованной области | Точка не принадлежит заштрихованной области |
| 5 | x = -0,2  y = 0,2 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 6 | x = -4  y = 5 | Точка не принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 7 | x = -2  y = -0,3 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 8 | x = -7  y = -4 | Точка не принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 9 | x = 0,3  y = -0,3 | Точка принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 10 | x = 4  y = -4 | Точка не принадлежит заштрихованной области | Точка принадлежит заштрихованной области |
| 11 | x = «вфжыда» | Введите число! (повторный ввод) | Введите число! (повторный ввод) |
| 12 | x = 4  y = «аыфд» | Введите число! (повторный ввод) | Введите число! (повторный ввод) |

Таблица 13 – система тестов для 2 задачи

В таблице 13 представлена система тестов для 2 задачи.

Произведём тестирование по критериям черного ящика для 2 задачи.

Таблица 14 – тестирование по критериям черного ящика для 2 задачи

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Проверяемая ситуация | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 | Т7 | Т8 | Т9 | Т10 | Т11 | Т12 |
| Характеристика классов входных данных | | | | | | | | | | | | |
| х или y – не числа |  |  |  |  |  |  |  |  |  |  | + | + |
| Точка лежит в I четверти, попадает в заштрихованную область |  |  | + |  |  |  |  |  |  |  |  |  |
| Точка лежит в I четверти, не попадает в заштрихованную область |  |  |  | + |  |  |  |  |  |  |  |  |
| Точка лежит во II четверти, попадает в заштрихованную область |  |  |  |  | + |  |  |  |  |  |  |  |
| Точка лежит во II четверти, не попадает в заштрихованную область |  |  |  |  |  | + |  |  |  |  |  |  |
| Точка лежит в III четверти, попадает в заштрихованную область |  |  |  |  |  |  | + |  |  |  |  |  |
| Точка лежит в III четверти, не попадает в заштрихованную область |  |  |  |  |  |  |  | + |  |  |  |  |
| Точка лежит в IV четверти, попадает в заштрихованную область |  |  |  |  |  |  |  |  | + |  |  |  |
| Точка лежит в IV четверти, не попадает в заштрихованную область |  |  |  |  |  |  |  |  |  | + |  |  |
| Точка лежит на границе |  | + |  |  |  |  |  |  |  |  |  |  |
| Точка – центр оси координат | + |  |  |  |  |  |  |  |  |  |  |  |

В таблице 14 представлено тестирование по критериям черного ящика для второй задачи.

Произведём тестирование по критериям белого ящика (МГТ) для второй задачи.

Таблица 15 – тестирование по критериям белого ящика (МГТ) для второй задачи

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Операторы | | Т1 | Т2 | Т3 | Т4 | Т5 | Т6 | Т7 | Т8 | Т9 | Т10 | Т11 | Т12 |
| while (!Double.TryParse(xix, out x)){} | |  |  |  |  |  |  |  |  |  |  | + |  |
| while (!Double.TryParse(xix, out y)){} | |  |  |  |  |  |  |  |  |  |  |  | + |
| if (OK) | Да | + | + | + |  | + |  | + |  | + |  |  |  |
| Нет |  |  |  | + |  | + |  | + |  | + |  |  |

В таблице 15 представлено тестирование по критериям белого ящика (МГТ) для второй задачи.

**Блок-схема ко второй задаче**

Конец

Начало

Ввод x

Ввод y

Точка x,y удовлетворяет заданному условию?

да

нет

Точка принадлежит заштрихованной области

Точка принадлежит заштрихованной области

x и y –

числа?

Нет

Да

**Рисунок 6 – Блок-схема ко второму заданию**

**Задача 3**

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

а = 10, b = 0.01

**Анализ**

Исходные данные:

1) a = 10 (const);

2) b= 0,01 (const).

Результат:

Используя формулы сокращенного умножения, получим результат 1. Однако из-за ограниченного количества знаков после запятой у типа Double и float получим, что результат будет и у типа Double, и у типа float равен лишь приблизительно единице. Но так как тип Double имеет большую точность по сравнению с типом float, то ближе к настоящему значению будет выражение, вычисленное из переменных с типом double.

**Код программы (double)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Zadanie3

{

class Program

{

static void Main(string[] args)

{

double a = 10;

double b = 0.01;

double a1;

double a2;

double a3;

double a4;

double a5;

double a6;

double c;

a1 = (Math.Pow((a - b), 4));

a2 = (Math.Pow(a, 4));

a3 = (Math.Pow(a, 2) \* 6 \* Math.Pow(b, 2));

a4 = (Math.Pow(b, 3) \* 4 \* a);

a5 = (Math.Pow(b, 4));

a6 = (Math.Pow(a, 3) \* b \* 4);

c = ((a1 - a2) / (a3 - a4 + a5 - a6));

Console.WriteLine("a = {0}, b = {1}, значение выражения = {2}", a, b, c);

}

}

}

Вывод: 0,9999999999999672

**Код программы (float)**

using System;

namespace Programma3

{

class Program

{

static void Main(string[] args)

{

float a = 10;

float b = 0.01f;

float a1;

float a2;

float a3;

float a4;

float a5;

float a6;

float c;

a1 = (float)(Math.Pow((a - b), 4));

a2 = (float)Math.Pow(a, 4);

a3 = (float)(Math.Pow(a, 2) \* 6 \* Math.Pow(b, 2));

a4 = (float)(Math.Pow(b, 3) \* 4 \* a);

a5 = (float)(Math.Pow(b, 4));

a6 = (float)(Math.Pow(a, 3) \* b \* 4);

c = (float)((a1 - a2) / (a3 - a4 + a5 - a6));

Console.WriteLine("a = {0}, b = {1}, значение выражения = {2}", a, b, c);

}

}

}

Вывод: 1,0000342

**Блок-схема к третьей задаче**

Начало

Вычисление выражения

Вывод выражения на экран

Конец