|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Гуцол Степан Дмитриевич  **ЛАБОРАТОРНАЯ РАБОТА №1**  по направлению подготовки *38.03.05 Бизнес-информатика*  образовательная программа «Разработка информационных систем для бизнеса»   |  |  | | --- | --- | |  | Руководитель  Преподаватель кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Найданов И.В. |   Пермь, 2022 |

**Задача №1**

**1. Постановка задачи**

Найти значения выражений:

1) --m-n++

2) m\*m<n++

3) n-->++m

4)

**2. Анализ**

Исходные данные:

* m, n, x – целые числа.

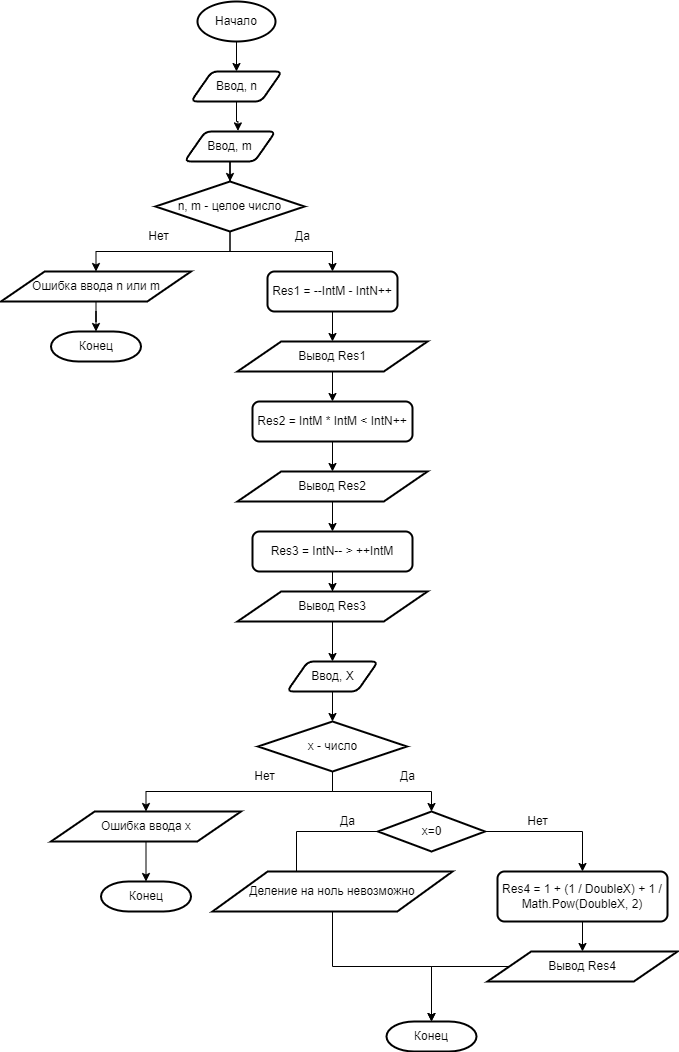
Выходные данные:

* res4 – целое число,
* res4 – вещественное число,
* res2, res3 – логические переменные.

***Табл. 1. Анализ классов исходных данных.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| 1) --m-n++  2) m\*m<n++  3) n-->++m | M, n | целое число | целое число |
| строки (не числа) | сообщение об ошибке ввода |
| 4) | x | вещественное число <>0 | вещественное число |
| x=0 | сообщение об ошибке деления на ноль |
| строка (не число) | сообщение об ошибке ввода |

**3. Алгоритм (блок-схема)**



***Рис. 1. Блок-схема 1 задания***

**4. Программа (листинг)**

// (1) --m-n++

Console.WriteLine("Введите переменную n:");

string N = Console.ReadLine();

Console.WriteLine("Введите переменную m:");

string M = Console.ReadLine();

int IntN; //Переменная N в целочисленном типе

int IntM; // Переменная M в целочисленном типе

int Res1;

bool SuccessN = int.TryParse(N, out IntN); // Проверка переменной N

bool SuccessM = int.TryParse(M, out IntM); // Проверка переменной M

if (SuccessN && SuccessM) //Проверка на то, являются ли переменные целыми числами

{

Res1 = --IntM - IntN++;

Console.WriteLine(Res1);

}

else

{

Console.WriteLine("Операция невозможна при введённых значениях");

}

// (2) m\*m<n++

Console.WriteLine("Введите переменную n:");

string N = Console.ReadLine();

Console.WriteLine("Введите переменную m:");

string M = Console.ReadLine();

int IntN; //Переменная N в целочисленном типе

int IntM; // Переменная M в целочисленном типе

bool Res2; // Переменная результата будет логического типа, т.к пример является неравенством

bool SuccessN = int.TryParse(N, out IntN); // Проверка переменной N

bool SuccessM = int.TryParse(M, out IntM); // Проверка переменной M

if (SuccessN && SuccessM) //Проверка на то, являются ли переменные целыми числами

{

Res2 = IntM \* IntM < IntN++; // Само неравенство

if (Res2)

{

Console.WriteLine("Неравенство истинно.");

}

else

{

Console.WriteLine("Неравенство ложно.");

}

// Console.WriteLine(Result2);

}

else

{

Console.WriteLine("Невозможно решить неравенство!");

}

//3) n-- > ++m

Console.WriteLine("Введите переменную n:");

string N = Console.ReadLine();

Console.WriteLine("Введите переменную m:");

string M = Console.ReadLine();

int IntN; //Переменная N в целочисленном типе

int IntM; // Переменная M в целочисленном типе

bool Res3; // Переменная результата будет логического типа, т.к пример является неравенством

bool SuccessN = int.TryParse(N, out IntN); // Проверка переменной N

bool SuccessM = int.TryParse(M, out IntM); // Проверка переменной M

if (SuccessN && SuccessM) //Проверка на то, являются ли переменные целыми числами

{

Res3 = IntN-- > ++IntM; // Само неравенство

if (Res3)

{

Console.WriteLine("Неравенство истинно.");

}

else

{

Console.WriteLine("Неравенство ложно.");

}

}

else

{

Console.WriteLine("Невозможно решить неравенство!");

}

// 4) 1 + 1/x + 1/x^2

Console.WriteLine("Введите переменную X:");

string X = Console.ReadLine();

double DoubleX;// Переменная X в типе с плавающей точкой, чтобы просиходило нормально деление

double Res4;

bool SuccessX; // Логическая переменная для проверки Х

SuccessX = double.TryParse(X, out DoubleX);

if (SuccessX) // Проверка Х на целочисленность

{

if (DoubleX == 0)

{

Console.WriteLine("Деление на ноль невозможно!");

}

else

{

Res4 = 1 + (1 / DoubleX) + 1 / Math.Pow(DoubleX, 2);

Console.WriteLine(Res4);

}

}

else

{

Console.WriteLine("Операция невозможна!");

}

**5. Тестирование**

***Табл. 2. Тестирование методом черного ящика.***

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | n=’A’ m=’A’ | Операция невозможна при введённых значениях | Операция невозможна при введённых значениях |
| **2** | n=’123gba’  m=’123ga’ | Операция невозможна при введённых значениях | Операция невозможна при введённых значениях |
| **3** | n=0 m=0 | 1)n=1 m=-1 res1 = -1  2)n=1 m=0 res2=false  3)n=-1 m=1 res3=false | 1)n=1 m=-1 res1 = -1  2)n=1 m=0 res2=false  3)n=-1 m=1 res3=false |
| **4** | n=2 m=4,3 | Операция невозможна при введённых значениях  (из-за того, что m – вещественное число) | Операция невозможна при введённых значениях  (из-за того, что m – вещественное число) |
| **5** | n=10,2 m=3 | Операция невозможна при введённых значениях  (из-за того, что n – вещественное число) | Операция невозможна при введённых значениях  (из-за того, что n – вещественное число) |
| **6** | n=7 m=5 | 1)n=8 m=4 res1=-3  2)n=8 m=5 res2=false  3)n=6 m=6 res3=true | 1)n=8 m=4 res1=-3  2)n=8 m=5 res2=false  3)n=6 m=6 res3=true |
| **7** | n=5 m=7 | 1)n=6 m=6 res1=1  2)n=6 m=7 res2=false  3)n=4 m=8 res3=false | 1)n=6 m=6 res1=1  2)n=6 m=7 res2=false  3)n=4 m=8 res3=false |
| **8** | x=’XY’ | Операция невозможна при введённых значениях | Операция невозможна при введённых значениях |
| **9** | x=0 | Деление на ноль невозможно | Деление на ноль невозможно |
| **10** | x=5 | res4=1.24 | res4=1.24 |

***Табл. 3. Классы входных данных.***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Строки** | **+** | **+** |  |  |  |  |  | **+** |  |  |
| **Вещ. числа** |  |  |  | **+** | **+** |  |  |  |  |  |
| **Ноль** |  |  | **+** |  |  |  |  |  | **+** |  |
| **Целые числа** |  |  |  |  |  | **+** | **+** |  |  | **+** |

***Табл. 4. Классы выходных данных.***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Вещ. число** |  |  |  |  |  |  |  |  |  | **+** |
| **Логический результат** |  |  | **+** |  |  | **+** | **+** |  |  |  |
| **Ошибка ввода** | **+** | **+** |  | **+** | **+** |  |  | **+** | **+** |  |
| **Ошибка X должно быть вещ. числом** |  |  |  |  |  |  |  | **+** |  |  |
| **Деление на ноль невозможно** |  |  |  |  |  |  |  |  | **+** |  |

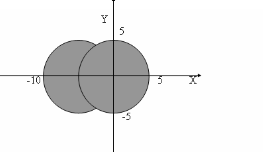
***Табл. 5. Тест по методу белого ящика (известен код программы).***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **if (SuccessN && SuccessM) / n вывод ошибки ввода n** | **+** |  |  | **+** |  |  | **+** | **+** |  |  |  |
| **-** | **+** | **+** |  | **+** | **+** |  |  |  |  |  |
| **if (SuccessN && SuccessM) /m вывод ошибки ввода n** | **+** |  |  | **+** |  |  | **+** | **+** |  |  |  |
| **-** | **+** | **+** |  | **+** | **+** |  |  |  |  |  |
| **If (x == 0) /вывод ошибки деления на 0** | **+** |  |  |  |  |  |  |  |  | **+** |  |
| **-** |  |  |  |  |  |  |  |  |  | **+** |

**Задача №2**

**1. Постановка задачи**

Определить принадлежность точки к заданной области.



***Рис. 1. Постановка задачи №2***

**2. Анализ**

Исходные данные:

* x, y – целые числа

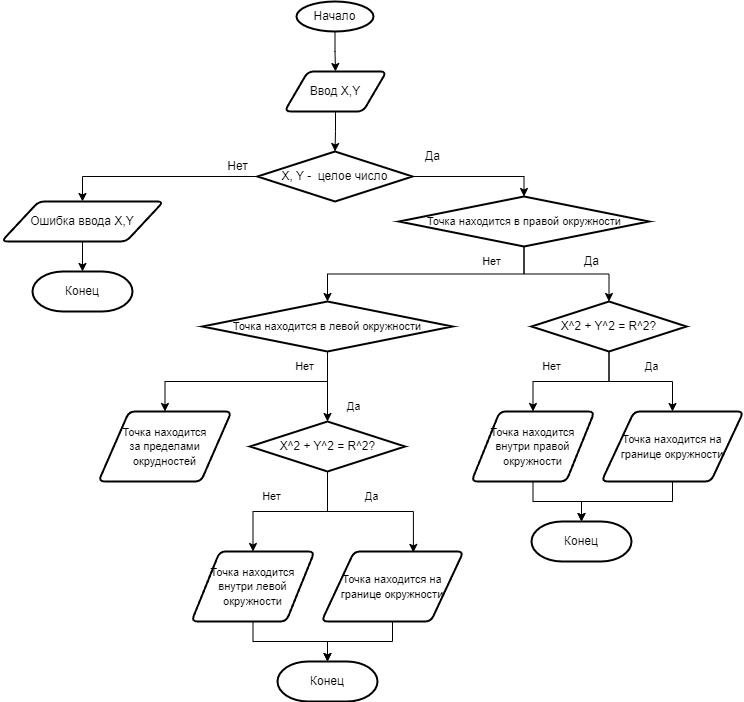
Выходные данные:

* SuccessX, SuccessY – логические переменные

***Табл. 1. Анализ классов исходных данных.***

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| x, y | Строка  Символ | Ошибка ввода |
| Целое число | Логическая переменная |

**3. Блок-схема**

 ***Рис. 2. Блок-схема 2 задания***

**4. Программирование**

// задача 2

Console.WriteLine("Введите координату X точки :");

string X = Console.ReadLine();

Console.WriteLine("Введите координату Y точки :");

string Y = Console.ReadLine();

bool SuccessX; // Логическая перемена для проверки X

bool SuccessY; // Логическая перемена для проверки Y

int IntX; // Переменная из строки в число

int IntY; // Переменная из строки в число

SuccessX = int.TryParse(X, out IntX); // Проверка X

SuccessY = int.TryParse(Y, out IntY); // Проверка Y

if (SuccessX && SuccessY) // Проверка по типу данных

{

if (IntX<=5 && IntX>=-5 && IntY<=5 && IntY >= -5 &&(Math.Pow(IntX, 2) + Math.Pow(IntY, 2) <= 25)) //Проверка на то, находится ли точка в правой окружности

{

if (Math.Pow(IntX,2)+ Math.Pow(IntY, 2) == 25) // Проверка по формуле X^2 + Y^2 = R^2

{

Console.WriteLine("Точка находится на границе одной из окружностей ");

}

else

{

Console.WriteLine("Точка находится внутри окружностей");

}

}

else

{

if (IntX <= 0 && IntX >= -10 && IntY <= 5 && IntY >= -5 && (Math.Pow(IntX+5, 2) + Math.Pow(IntY, 2) <= 25)) // Проверка на нахождение в первой окружности

{

if (Math.Pow(IntX+5, 2) + Math.Pow(IntY, 2) == 25) // Проверка по формуле X^2 + Y^2 = R^2

{

Console.WriteLine("Точка находится на границе одной из окружностей ");

}

else

{

Console.WriteLine("Точка находится внутри окружностей");

}

}

else

{

Console.WriteLine("Точка находится за пределами закрашенной области!");

}

}

}

else

{

Console.WriteLine("Невозможно определить!");

}

**5. Тестирование**

***Табл. 2. Тестирование методом черного ящика.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | X=’abc’ | Невозможно определить | Невозможно определить |
| **2** | Y=’abc’ | Невозможно определить | Невозможно определить |
| **3** | X=0 y=5 | Точка находится на границе одной из окружностей | Точка находится на границе одной из окружностей |
| **4** | X=-5 y=5 | Точка находится на границе одной из окружностей | Точка находится на границе одной из окружностей |
| **5** | X=-2,7 y=4,2 | Точка находится внутри окружностей | Точка находится внутри окружностей |
| **6** | X=7,1 y=-1,1 | Точка находится за пределами закрашенной области | Точка находится за пределами закрашенной области |

***Табл. 3. Классы исходных данных.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **Текст** | **+** | **+** |  |  |  |  |
| **Число целое** |  |  | **+** | **+** |  |  |
| **Число дробное** |  |  |  |  | **+** | **+** |

***Табл. 4. Классы выходных данных.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
| **Ответ: да** |  |  | **+** | **+** | **+** |  |
| **Ответ: нет** |  |  |  |  |  | **+** |
| **Ошибка ввода** | **+** | **+** |  |  |  |  |

***Табл. 5. Тестирование методом белого ящика.***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** | **3** | **4** | **5** | **6** |
| **if (SuccessX && SuccessY)/x** | **+** | **+** |  |  |  |  |  |
| **-** |  |  | **+** | **+** | **+** | **+** |
| **if (SuccessX && SuccessY)/y** | **+** |  | **+** |  |  |  |  |
| **-** |  |  | **+** | **+** | **+** | **+** |

**Задача №3**

**1. Постановка задачи**

Для задачи 3 вычислить значение выражения, используя различные вещественные типы данных (float и double):

***Исходные данные:***

* a = 1000,
* b = 0.0001.

**2. Анализ**

***Классы входных данных:***

* a, b – константы float,
* DoubleA, DoubleB – константы double.

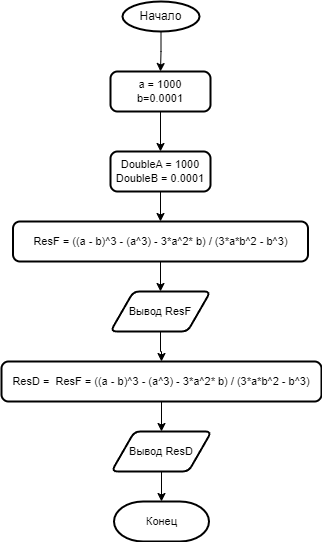
***Классы выходных данных:***

* ResD – тип double,
* ResF – тип float.

***Табл. 1. Анализ классов исходных данных.***

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Класс исходных данных** | **Класс выходных данных** |
| a=1000 b=0.0001 | a, b – вещественные числа (double) | Вещественное число (double) |
| a, b – вещественные числа (float) | Вещественное число (float) |

**3. Блок-схема**



***Рис. 1. Блок-схема 3 задания***

**4. Программа**

// ((a-b)^3 - (a^3 -3\*(a^2)\*b)) / 3\*a\*(b^2) - b^3

float a = 1000f;

float b =0.0001f;

float ResF;

double DoubleA = 1000;

double DoubleB = 0.0001;

double ResD;

ResF = (((float)Math.Pow(a - b, 3) - (float)(Math.Pow(a, 3) - 3.0f \* (float)Math.Pow(a, 2) \* b)) / (3.0f \* a \* (float)Math.Pow(b, 2) - (float)Math.Pow(b, 3)));

Console.WriteLine("a и b - float");

Console.WriteLine(ResF);

Console.WriteLine();

ResD = (Math.Pow(DoubleA - DoubleB, 3) - (Math.Pow(DoubleA, 3) - 3 \* Math.Pow(DoubleA, 2) \* DoubleB)) / (3 \* DoubleA \* Math.Pow(DoubleB, 2) - Math.Pow(DoubleB, 3));

Console.WriteLine("a, b - double");

Console.WriteLine(ResD);

Console.WriteLine();

**5. Объяснение выходных значений**

**Результат float:** -2133333,8. **Результат double:** 1,0013580656051646.

Полученные результаты отличаются, потому что при типе данных float, результаты операций получаются менее точными, в данном случае результаты получились очень различающимися.