|  |
| --- |
| 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 №2**  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**

Решить указанные в варианте задачи, используя основные операторы языка С#. При решении задачи, использовать разные типы циклов (for, while, do while):

Дана последовательность из n целых чисел. Найти минимальный элемент в этой последовательности.

**2. Анализ**

Исходные данные:

* number – целое число,
* n – целое число большее или равное нулю.

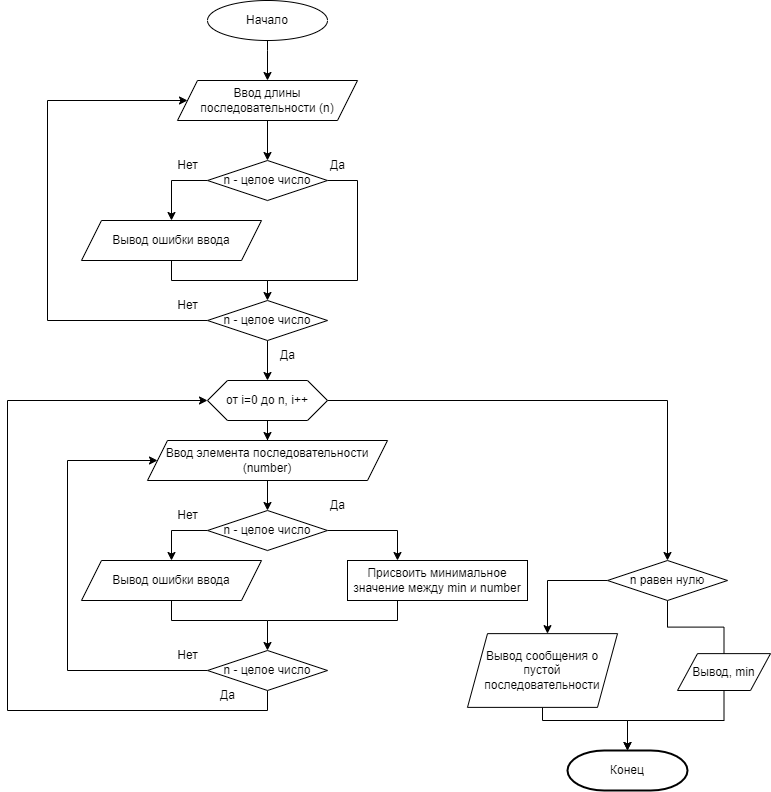
Выходные данные:

* min – целое число.

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

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| 1) Дана последовательность из n целых чисел. Найти минимальный элемент в этой последовательности. | number, n | n, number – целое число <> 0 | целое число |
| * n, number – строки (не числа) * n –отрицательные числа | сообщение об ошибке ввода |
| n – нуль | сообщение о пустой последовательности |

**3. Алгоритм (блок-схема)**



***Рис. 1. Блок-схема 1 задания***

**4. Программа (листинг)**

//6. Дана последовательность из n целых чисел. Найти минимальный элемент в этой последовательности.

int n,number=int.MaxValue,min=int.MaxValue;

bool ok;

do //Цикл проверки на то, является ли n - целым числом большим или равным нулю

{

Console.WriteLine("Пожалуйста, введите целое число n - длину последовательности");

string StringN = Console.ReadLine();

ok = ((int.TryParse(StringN, out n))&&(n>=0));

if (!ok)

{

Console.WriteLine("Вы ввели некорректные данные, n - целое число, n>=0");

}

} while (!ok);

for (int i = 0; i < n; i++) //цикл ввода элементов последовательности от 0 до n

{

do //цикл проверки ввода элементов на то являются ли они целыми числами или нет

{

Console.WriteLine("Пожалуйста, введите целое число - элемент последовательности");

string StringNum = Console.ReadLine();

ok = int.TryParse(StringNum, out number);

if (!ok)

{

Console.WriteLine("Вы ввели не целое число, попробуйте ещё раз!");

}

if (ok)

{

min = Math.Min(min, number);

}

} while (!ok);

}

if (n == 0){ // Проверка на то является последовательность пустой или нет(не уверен, что нужно, а не просто выдавать ошибку)

Console.WriteLine("Последовательность пуста");

}

else

{

Console.WriteLine($"Минимальное число последовательности = {min}");

}

**5. Тестирование**

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

| **№** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **1** | n=’A’ number=’A’ | Ошибка ввода на длине последовательности | Ошибка ввода на длине последовательности |
| **2** | n=’0’  number=’0’ | Пустая последовательность | Пустая последовательность |
| **3** | n=’1’ number= ‘3’ | min = 3 | min = 3 |
| **4** | n= ‘2’ number=’1 5’ | min = 1 | min = 1 |
| **5** | n=’2’  number= ‘0,3 три 12 -16’ | Две ошибки ввода на первом элементе  min = -16 | Две ошибки ввода на первом элементе  min = -16 |
| **6** | n=’3’  number = ‘-30 0 -21’ | min = -30 | min = 30 |
| **7** | n=’3’  number =’10 40 0’ | min = 0 | min =0 |
| **8** | n=’1,5’  number = ‘0’ | Ошибка ввода на длине последовательности | Ошибка ввода на длине последовательности |

***Табл. 3. Классы входных данных.***

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

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

|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Целое число** |  |  | **+** | **+** | **+** | **+** | **+** |  |
| **Ошибка ввода** | **+** |  |  |  | **+** |  |  | **+** |
| **Пустая последовательность** |  | **+** |  |  |  |  |  |  |

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

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Do While (!ok) / Ввод n** | **1** |  | **+** | **+** | **+** | **+** | **+** | **+** |  |
| **n** | **+** |  |  |  |  |  |  | **+** |
| **Do While (!ok)/**  **Ввод number** | **1** | **+** | **+** | **+** | **+** |  | **+** | **+** | **+** |
| **n** |  |  |  |  | **+** |  |  |  |
| **for (int i = 0; i < n; i++) /** | **Выполнялось** |  |  | **+** | **+** | **+** | **+** | **+** |  |
| **Не выполнялось** |  | **+** |  |  |  |  |  |  |
| **if (!ok) / Вывод ошибки ввода n** | **+** | **+** |  |  |  |  |  |  | **+** |
| **-** |  | **+** | **+** | **+** | **+** | **+** | **+** |  |
| **if (!ok) / Вывод ошибки ввода number** | **+** | **+** |  |  |  | **+** |  |  |  |
| **-** |  |  | **+** | **+** |  | **+** | **+** |  |
| **If (ok) / Присваивание min** | **+** |  |  | **+** | **+** |  | **+** | **+** |  |
| **-** | **+** |  |  |  | **+** |  |  |  |
| **If (n==0) / последовательность пустая** | **+** |  | **+** |  |  |  |  |  |  |
| **-** | **+** |  | **+** | **+** | **+** | **+** | **+** | **+** |

**Задача №2**

**1. Постановка задачи**

Решить указанные в варианте задачи, используя основные операторы языка С#. При решении задачи, использовать разные типы циклов (for, while, do while):

Дана последовательность целых чисел, за которой следует 0. Найти разность минимального и максимального элементов в этой последовательности.

**2. Анализ**

Исходные данные:

* number – целое число.

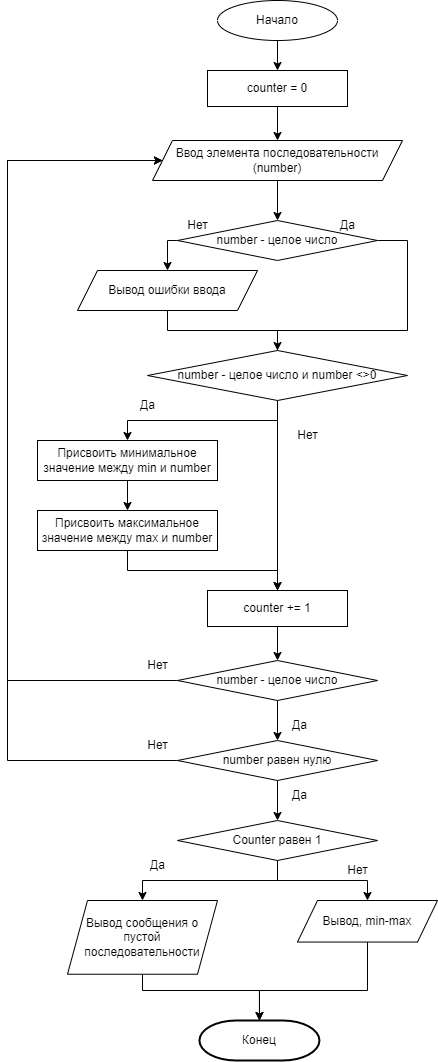
Выходные данные:

* (min-max), min, max – целые числа.

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

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
| Дана последовательность целых чисел, за которой следует 0. Найти разность минимального и максимального элементов в этой последовательности. | number | целое число | целое число |
| единственный нуль | сообщение о пустой последовательности |
| * n – строки (не числа) * вещественные числа | сообщение об ошибке ввода |

**3. Блок-схема**



***Рис. 2. Блок-схема 2 задания***

**4. Программирование (листинг)**

//28. Дана последовательность целых чисел, за которой следует 0.

//Найти разность минимального и максимального элементов в этой последовательности.

int number = int.MaxValue, min = int.MaxValue, max = int.MinValue, counter=0;

bool ok, end;

Console.WriteLine("Данная программа будет работать до тех пор, пока не будет введён 0"); // Пояснение для пользователя

do

{

do // цикл проверки на то, является ли введённый элемент числом

{

Console.WriteLine("Пожалуйста, введите целое число - элемент последовательности");

string StringNum = Console.ReadLine();

ok = int.TryParse(StringNum, out number);

if (!ok)

{

Console.WriteLine("Вы ввели не целое число, попробуйте ещё раз!");

Counter -= 1;

}

end = (ok) && (number == 0);

if ((ok)&&(number!=0))

{

min = Math.Min(min, number);

max = Math.Max(max, number);

}

counter += 1;

} while (!ok);

} while (!end);

if (counter == 1){

Console.WriteLine("Последовательность пустая!");

}

else

{

Console.WriteLine($"Разность минимального и максимального элементов = {min - max}");

}

**5. Тестирование**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | Number = ‘0’ | Сообщение о пустой последовательности | Сообщение о пустой последовательности |
| **2** | Number = ‘3 0,5 два 6 7 1 0’ | Два сообщения об ошибке  Min-max = -6 | Два сообщения об ошибке  Min-max = -6 |
| **3** | Number = ‘-5 -13 0’ | Min - max = -8 | Min -max = -8 |
| **4** | Number = ’10 2 5 2 10’ | Min - max = -8 | Min- max = -8 |
| **5** | Number = ’22 22 0’ | Min - max = 0 | Min - max = 0 |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Строка (не число)** |  | **+** |  |  |  |
| **Число целое** |  | **+** | **+** | **+** | **+** |
| **Число дробное** |  | **+** |  |  |  |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Целое число** |  | **+** | **+** | **+** | **+** |
| **Ошибка ввода** |  | **+** |  |  |  |
| **Пустая последовательность** | **+** |  |  |  |  |

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

|  |  | **1** | **2** | **3** | **4** | **5** |
| --- | --- | --- | --- | --- | --- | --- |
| **if (!ok) / number - число** | **+** |  | **+** |  |  |  |
| **-** | **+** |  | **+** | **+** | **+** |
| **if ((ok)&&(number!=0)) / number – число не равное нулю** | **+** |  | **+** | **+** | **+** | **+** |
| **-** | **+** |  |  |  |  |
| **if (counter == 1) / Пустая последовательность** | **+** | **+** |  |  |  |  |
| **-** |  | **+** | **+** | **+** | **+** |
| **Do While (!end) / Ввод number** | **1** |  | **+** | **+** | **+** | **+** |
| **n** | **+** |  |  |  |  |
| **Do While (!ok) /** | **1** | **+** |  | **+** | **+** | **+** |
| **n** |  | **+** |  |  |  |

**Задача №3**

**1. Постановка задачи**

Решить указанные в варианте задачи, используя основные операторы языка С#. При решении задачи, использовать разные типы циклов (for, while, do while):

 где n>2.

**2. Анализ**

Исходные данные:

* n – целое число большее двух. (Задача 3)

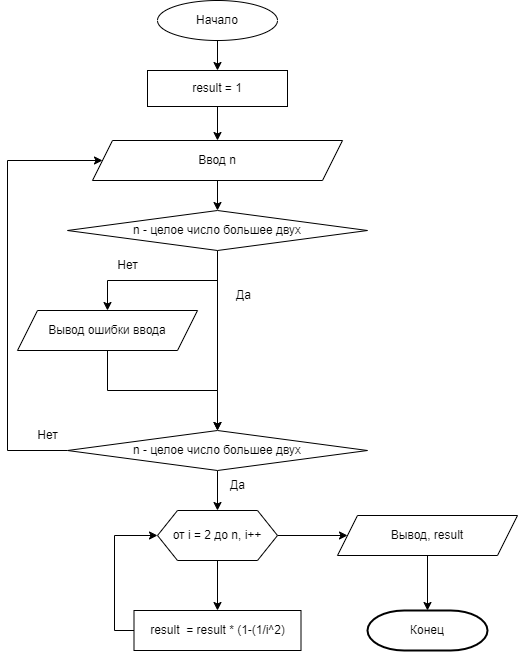
Выходные данные:

* result – вещественное число.

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

| **Задача** | **Исходные данные** | **Класс входных данных** | **Класс выходных данных** |
| --- | --- | --- | --- |
|  | n | целое число >2 | вещественное число |
| * n – строки (не числа) * вещественные числа * целые числа <=2 | сообщение об ошибке ввода |

**3. Блок-схема**



***Рис. 1. Блок-схема 3 задания***

**4. Программа (листинг)**

//57. P =(1-1/2^2)\*(1-1/3^2)\*..\*(1-1/n^2), n>2

int n;

double result=1;

bool ok;

do // Цикл проверки на то, является ли введённый элемент целым числом и больше двух

{

Console.WriteLine("Пожалуйста, введите целое число n, где n>2");

string StringN = Console.ReadLine();

ok = (int.TryParse(StringN, out n))&&(n>2);

if (!ok)

{

Console.WriteLine("Вы ввели некорректные данные!");

}

} while (!ok);

for (int i = 2; i <= n; i++) // Цикл подсчёта по формуле из условия

{

result \*= (1 - (1 / Math.Pow(i, 2)));

}

Console.WriteLine($"Результат вычислений P = {result}");

**5. Тестирование**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Исходные данные** | **Ожидаемый результат** | **Полученный результат** |
| **1** | n = ‘0’ | Сообщение об ошибке | Сообщение об ошибке |
| **2** | n= ‘2’ | Сообщение об ошибке | Сообщение об ошибке |
| **3** | n = ‘4.5’ | Сообщение об ошибке | Сообщение об ошибке |
| **4** | n= ’Два’ | Сообщение об ошибке | Сообщение об ошибке |
| **5** | n = ‘6’ | Result = 0,583 | Result = 0,5833333333333333 |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Строка (не число)** |  |  |  | **+** |  |
| **Число целое** |  | **+** |  |  | **+** |
| **Число дробное** |  |  | **+** |  |  |
| **Нуль** | **+** |  |  |  |  |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **Целое число** |  |  |  |  | **+** |
| **Ошибка ввода** | **+** | **+** | **+** | **+** |  |

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

|  |  | **1** | **2** | **3** | **4** | **5** |
| --- | --- | --- | --- | --- | --- | --- |
| **if (!ok) / number – число большее двух** | **+** |  |  |  |  | **+** |
| **-** | **+** | **+** | **+** | **+** |  |
| **for (int i = 2; i <= n; i++) / Нахождение result** | **Выполнялось** |  |  |  |  | **+** |
| **Не выполнялось** | **+** | **+** | **+** | **+** |  |
| **Do While (!ok) / Ввод number** | **1** |  |  |  |  | **+** |
| **n** | **+** | **+** | **+** | **+** |  |