|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
|  |
| Ханжин Александр Евгеньевич  **ЛАБОРАТОРНАЯ РАБОТА №1**  студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.10.23 Программная инженерия*   |  |  | | --- | --- | |  | Руководитель к.т.н., доцент, доцент кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  О. Л. Викентьева |   Пермь, 2023 |

# Задача №1

## 1. Постановка задачи

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

Для полноты понимания результата работы программы так же реализуем ввод количества максимумов.

## 2. Анализ

Исходные данные:

* n – целое положительное число длины последовательности.
* number – целые числа
* maxNumber, countMax – целые числа числа,

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

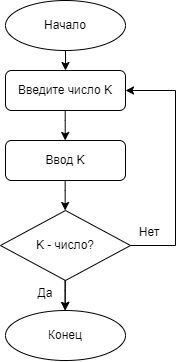
|  |  |  |  |
| --- | --- | --- | --- |
| **Описание функции** | **Исходные данные** | **Класс входных данных** | **Выходные данные** |
| 1) Поиск макс. элемента последовательности | n, numbers | Вещественные числа | Вещественное число |
| Строки (не числа) | Сообщение об ошибке ввода |
| 2) Подсчет количества максимальных элем-ов | n, numbers | Вещественные числа | Вещественное число |

## 3. Алгоритм (блок-схема)

Для дальнейшей реализации проектирования необходимо составить блок схему, она представлена ниже на рисунках 1 и 2.

## 

## *Рис. 1. Блок-схема 1 задания*

******

***Рис. 2. Блок-схема операции “Ввод числа”***

## 4. Программа (листинг)

static void FirstTask()

{

int n;

bool isConvert;

Console.WriteLine("Задание 1. Дана последовательность из n целых чисел. Найти максимальный элемент в этой последовательности.");

Console.WriteLine("Привет! Введи длину последовательности.");

do

{

isConvert = int.TryParse(Console.ReadLine(), out n);

if (!isConvert)

{

Console.WriteLine("Длина последовательности должна быть целым числом.");

Console.WriteLine("Введи длину последовательности.");

}

else if (n < 0)

{

Console.WriteLine("Длина последовательности должна быть положительной.");

Console.WriteLine("Введи длину последовательности.");

isConvert = false;

}

else if (n == 0)

{

Console.WriteLine("Длина последовательности не может быть равна нулю.");

Console.WriteLine("Введи длину последовательности.");

isConvert = false;

};

} while (!isConvert);

int number, maxNumber, countMax;

maxNumber = -1000000000;

countMax = 1;

for (int i = 0; i < n; i++)

{

Console.WriteLine($"Введи целое число. Уже введено {i} чисел из {n}.");

do

{

isConvert = int.TryParse(Console.ReadLine(), out number);

if (!isConvert)

Console.WriteLine("Ошибка. Нужно ввести целое число.");

} while (!isConvert);

if (maxNumber == number)

{

countMax += 1;

}

else if (number > maxNumber)

{

maxNumber = number;

countMax = 1;

}

}

Console.WriteLine($"Максимальный элемент последовательности равен: {maxNumber}");

Console.WriteLine($"Количество максимальных элементов: {countMax}");

}

## 5. Тестирование

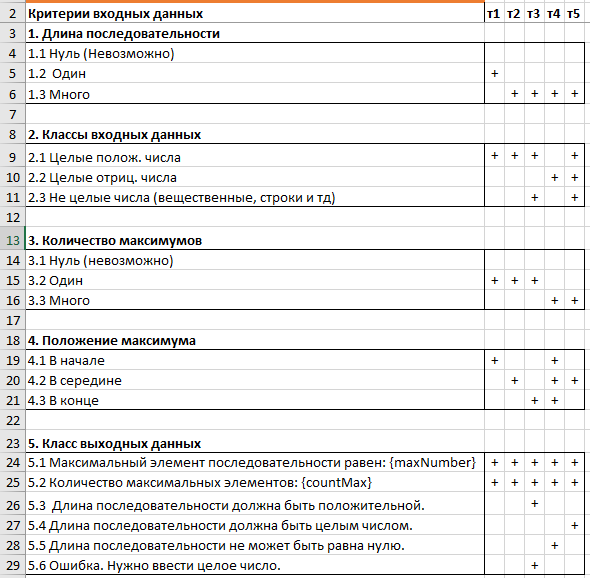
Результаты тестирования представлены на рисунке 3. На рисунке видно, что все ожидаемые результаты совпадают с реальными.



***Рисунок 3. – Таблица тестов для задания 1***

Таблица тестов составлялась с помощью дополнительных таблиц по критериям черного и белого ящиков. Они представлены ниже.

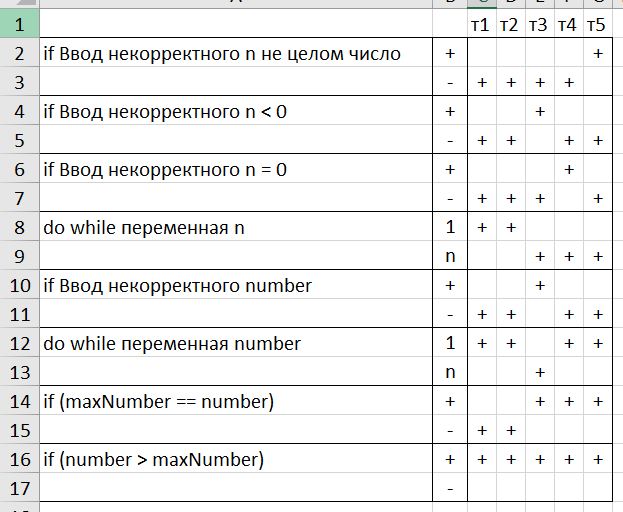
Рисунок 4. содержит таблицу с критериями черного ящика. Она состоит из классов входных и выходных данных, рассмотренных в анализе, а так же других критериев.



***Рисунок 4. – Таблица критериев черного ящика задания 1***

Также тестирование проводилось по критериям белого ящика. Так как циклы, отвечающие за проверку правильного ввода, не дадут это сделать. Таблица критериев белого ящика представлена на рисунке 5. (см. ниже).

Ознакомиться с полным списком можно в формате excel, который приложен к лабораторной работе.



***Рисунок 5. – Таблица критериев белого ящика задания 1***

# Задача №2

## Постановка задачи

Дана последовательность целых чисел, за которой следует 0. Найти максимальный элемент в этой последовательности.

Для полноты понимания результата работы программы так же реализуем ввод количества максимумов.

## Анализ

Исходные данные:

* number – целое положительное число длины последовательности

Выходные данные:

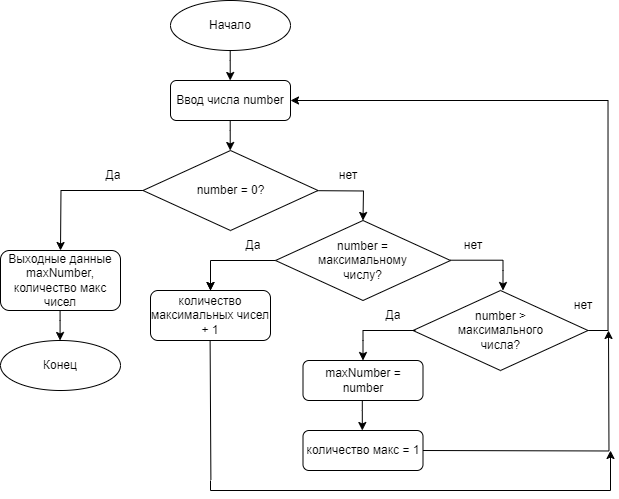
* maxNumber, countMax – целые числа

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

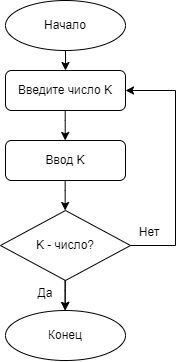
|  |  |  |  |
| --- | --- | --- | --- |
| **Описание функции** | **Исходные данные** | **Класс входных данных** | **Выходные данные** |
| 1) Поиск макс. элемента последовательности | n, numbers | Вещественные числа | Вещественное число |
| Строки (не числа) | Сообщение об ошибке ввода |
| 2) Подсчет количества максимальных элем-ов | n, numbers | Вещественные числа | Вещественное число |

## Блок-схема

Для дальнейшей реализации проектирования необходимо составить блок схему, она представлена ниже на рисунках 1 и 2.



***Рис. 1. Блок-схема 2 задания.***



***Рис. 2. Блок-схема операции “Ввод числа”***

## Программирование

static void SecondTask()

{

Console.WriteLine("Задача 2. Дана последовательность целых чисел, за которой следует 0. Найти максимальный элемент в этой последовательности.");

bool isConvert, numberNotZero;

int number, maxNumber, countMax, index;

maxNumber = -1000000000;

countMax = 1;

index = 0;

do

{

Console.WriteLine("Введите число. Если хотите завершить работу - введите 0.");

do

{

isConvert = int.TryParse(Console.ReadLine(), out number);

if (!isConvert)

Console.WriteLine("Ошибка. Нужно ввести целое число.");

} while (!isConvert);

if (number != 0)

{

numberNotZero = true;

// Количество введенных чисел

index += 1;

if (maxNumber == number)

{

countMax += 1;

}

else if (number > maxNumber)

{

maxNumber = number;

countMax = 1;

}

}

else

{

numberNotZero = false;

}

} while (numberNotZero);

// Проверка длины последовательности

if (index == 0)

{

Console.WriteLine("Последовательность пуста.");

}

else

{

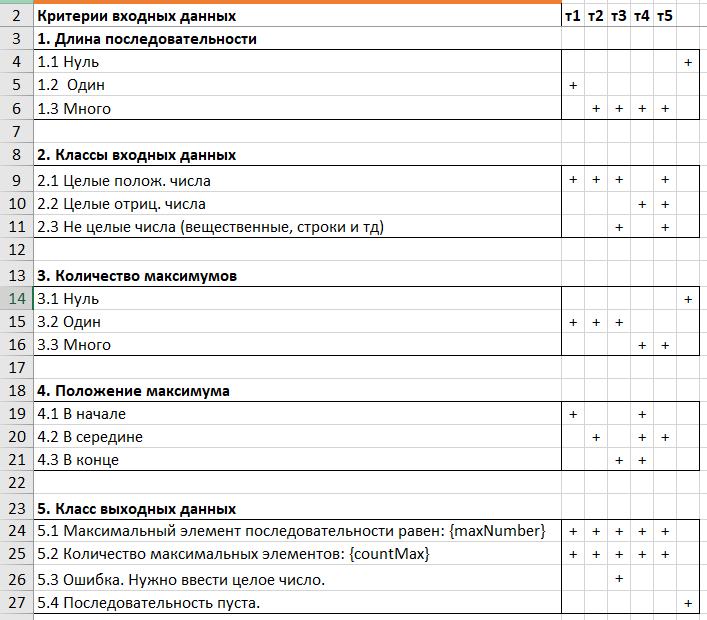
Console.WriteLine($"Максимальный элемент последовательности равен: {maxNumber}");

Console.WriteLine($"Количество максимальных элементов: {countMax}");

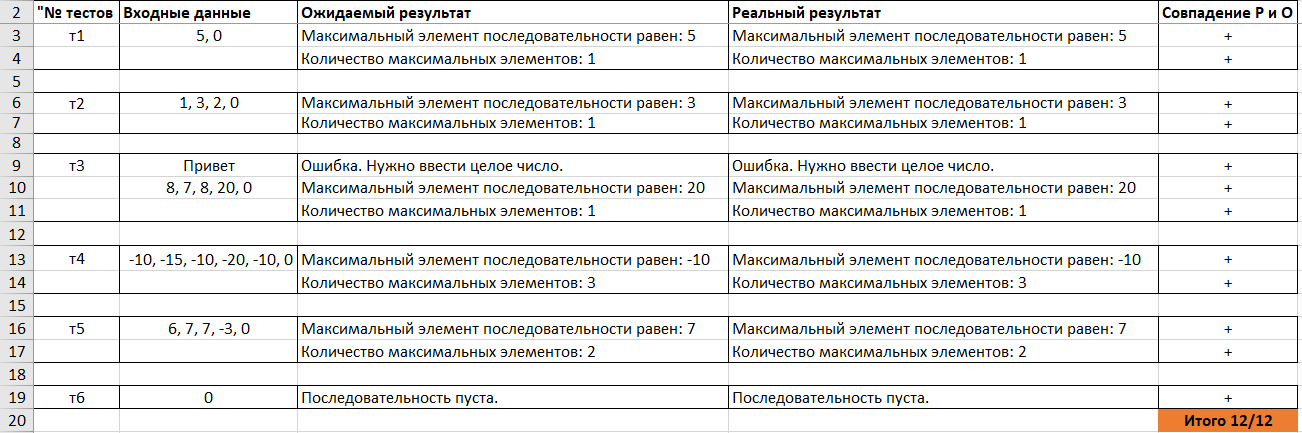
}

## Тестирование

Для тестирования программы была составлена таблица критериев Черного ящика. Она представлена на рисунке 3.

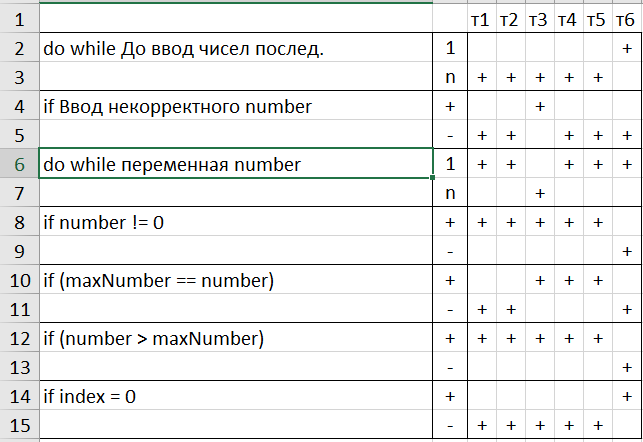
*** Рис. 3. Таблица критериев черного ящика задания 2.***

К ней были составлены тесты, представление на рисунке 4. Ожидаемые и реальные результаты совпали на 100%.



***Рисунок 4. – Таблица тестов для задания 2***

Таблица с тестированием по критериям белого ящика представлена на рисунке 5.

*** Рисунок 5. – Таблица критериев белого ящика задания 2***

Ознакомиться с полным списком можно в формате excel, который приложен к лабораторной работе.

# Задача №3

## Постановка задачи

Найти первое отрицательное число последовательности u=cos(ctg(n)), где n=1,2,3….

**Исходные данные:**

* n – целое число 1,2,3…

## Анализ

**Классы выходных данных:**

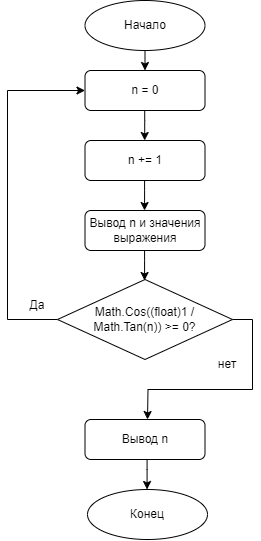
* n – целое число

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

|  |  |  |
| --- | --- | --- |
| **Описание функции** | **Исходные данные** | **Выходные данные** |
| 1) Поиск первого отриц. числа последовательности | n – целые натуральные числа | Вещественное число |

## Блок-схема

Для дальнейшей реализации проектирования необходимо составить блок схему, она представлена ниже на рисунке 1. По ходу выполнения программы пользователь получает информацию о каждом пройденном значении n и текущем результате выражения для отслеживания корректности итогового результата.



***Рис. 1. Блок-схема 1 задания***

## Программа

static void ThirdTask()

{

Console.WriteLine("Задание 3. Найти первое отрицательное число последовательности u=cos(ctg(n)), где n=1,2,3...");

int n = 0;

do

{

n += 1;

Console.Write($"При n = {n} принимает значение: ");

Console.WriteLine(Math.Cos((float)1 / Math.Tan(n)));

} while (Math.Cos((float)1 / Math.Tan(n)) >= 0);

Console.WriteLine($"Уравнение u=cos(ctg(n)) впервые принимает отрицательное значение при n = {n}.");

}

## Объяснение выходных значений

TODO