|  |
| --- |
| 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 №3**  студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.03.04 Программная инженерия*   |  |  | | --- | --- | |  | Руководитель к.т.н., доцент кафедры ИТБ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  О. Л. Викентьева |   Пермь, 2023 |

# Задача 1

## Постановка задачи

Для х изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x) (рис. 1.1.1), используя ее разложение в степенной ряд в двух случаях:

а) для заданного n;

б) для заданной точности e (e=0.0001).

, (1.1.1)

Для сравнения найти точное значение функции.

## Анализ

Исходные данные:

* константа a = 0.1;
* константа b = 0.8;
* константа k = 10;
* константа n = 30;
* константа e = 0.0001;

С учетом эти значений нужно вычислить Sn (рис. 1.2.1) и Se (рис. 1.2.1).

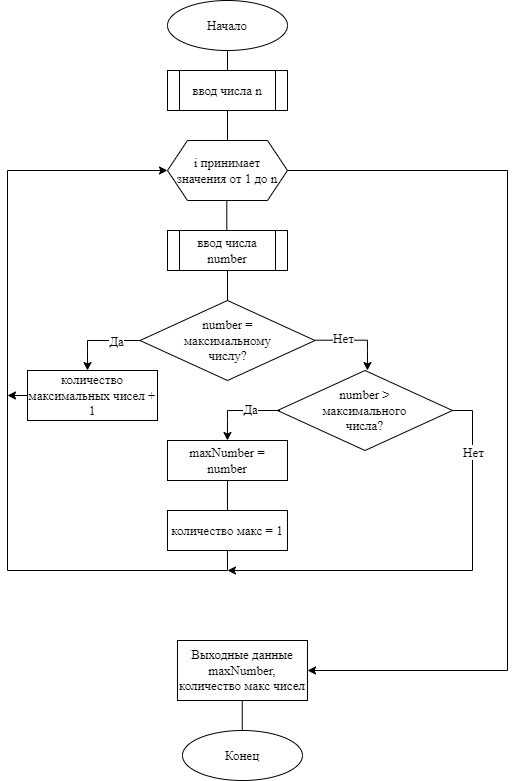
Исходные и выходные данные каждой функции представлены в таблице 1.

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

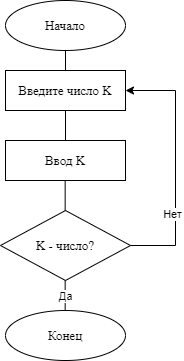
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Описание функции** | **Исходные данные** | **Выходные данные** |
| 1 | Вычисление функции f(x) | x | Вещественное число |
| 2 | Вычисление Sn | n, x | Вещественное число |
| 3 | Вычисление Se | e, x | Вещественное число |

## Алгоритм (блок-схема)

Для дальнейшей реализации проектирования необходимо составить блок схему. Она состоит из двух частей: основная часть программы (рис. 1.3.1) и функции ввода числа (рис. 1.3.2)

**

***Рис. 1.3.1. Блок-схема 1 задания***

******

***Рис. 1.3.2. Блок-схема операции “Ввод числа”***

## 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.");

// считывание числа натурального числа n

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);

// переменные для вычисления максимального числа из последовательности и их кол-ва

int number, maxNumber, countMax;

maxNumber = -1000000000;

countMax = 1;

// последовательное считывание числа number

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);

// перезапись максимального числа из предыдущих введеных и текущего и количества введеных макс. чисел

// по заданию нужно вывести только макс. из последовательности, но я доработал программу и подсчитываю еще их количество

if (maxNumber == number)

{

countMax += 1;

}

else if (number > maxNumber)

{

maxNumber = number;

countMax = 1;

}

}

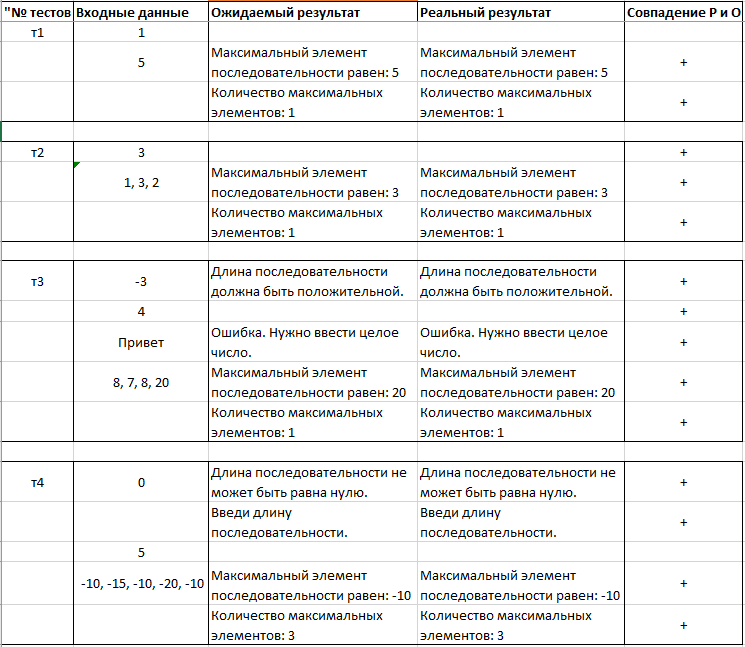
Console.WriteLine($"Максимальный элемент последовательности равен: {maxNumber}");

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

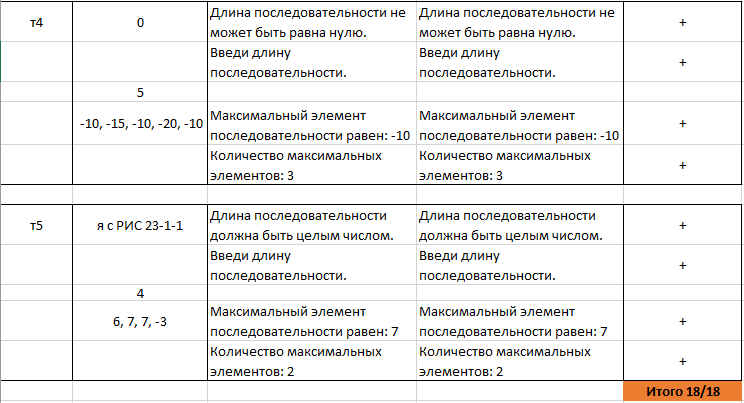
}

## Тестирование

Результаты тестирования представлены на рисунках 1.5.1 и 1.5.2 На рисунках видно, что все ожидаемые результаты совпадают с реальными. В ходе этого тестирования программа была протестирована с использованием различных входных данных. Тесты составлялись таким образом, чтобы покрыть все возможные варианты ввода данных пользователем.



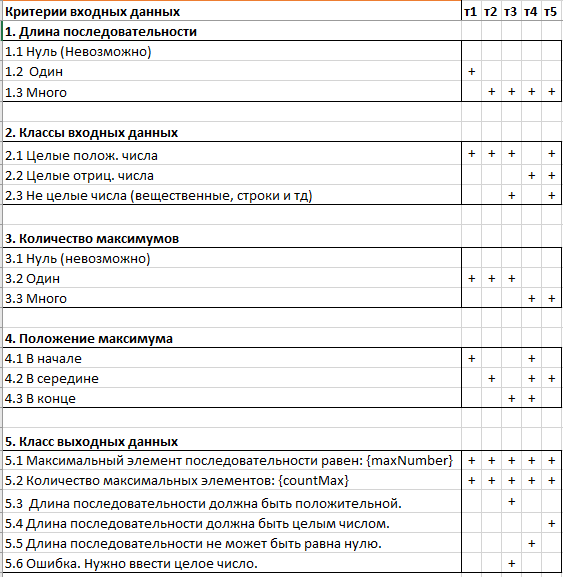
***Рисунок 1.5.1. – Таблица тестов для задания 1***

******

***Рисунок 1.5.2. – Таблица тестов для задания 1***

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

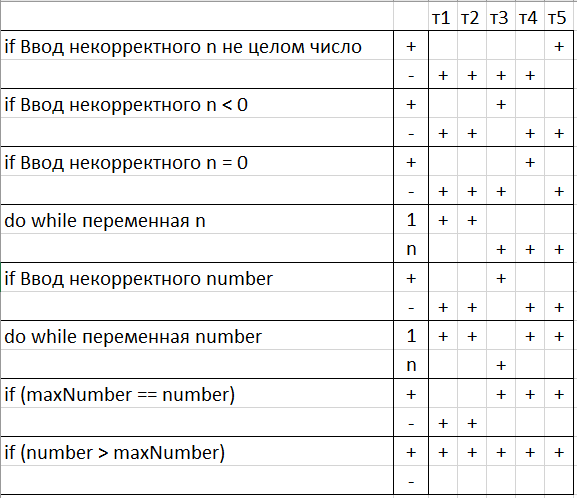
Рисунок 1.5.3. содержит таблицу с критериями черного ящика. Она состоит из классов входных и выходных данных, рассмотренных в анализе, а также других критериев.



***Рисунок 1.5.3. – 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 1.5.4. (см. ниже).

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



***Рисунок 1.5.4. – Таблица критериев белого ящика задания 1***