# Лабораторная работа №3

## Функции и массивы

### Вариант 1

Цель работы: изучить особенности использования функций и массивов в языке C#.

### Задание к работе

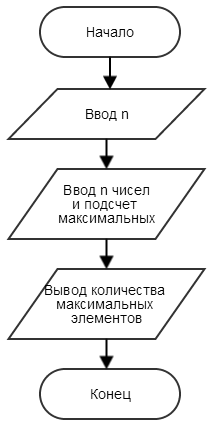
1. Изучить главы 5, 6 учебника «Visual C# 2010: полный курс».
2. Решить представленные задачи в соответствии с вариантом, оформив самостоятельные подзадачи в виде функций.

# Задача 1

Дана последовательность из вещественных чисел. Подсчитать количество максимальных элементов.

## Решение

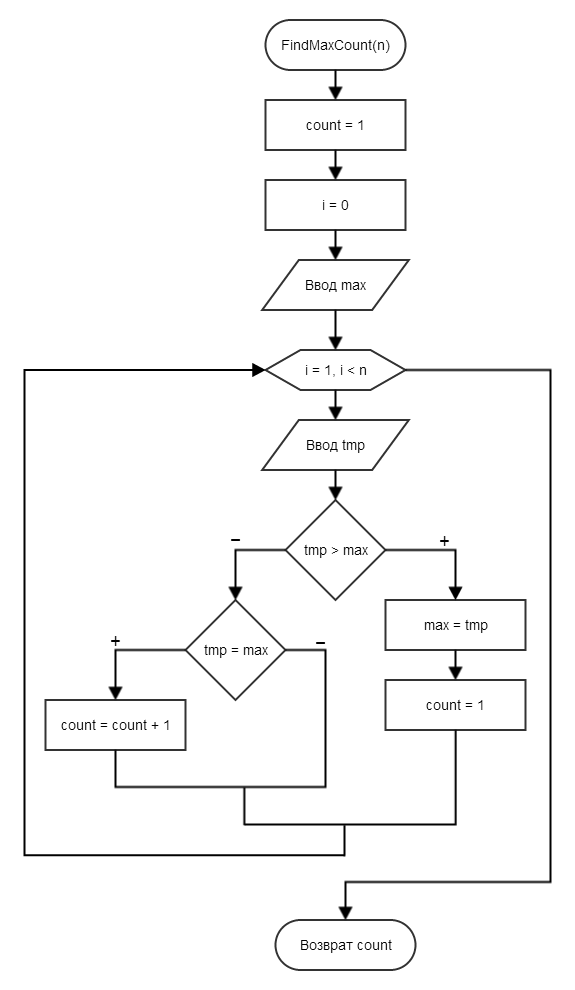
### Блок-схема алгоритма



### Описание подпрограмм

1. Спецификация функции FindMaxCount
   1. Заголовок: static int FindMaxCount(int n)
   2. Назначение: вводит n элементов, подсчитывает и возвращает количество максимальных.
   3. Входные параметры: n
   4. Выходные параметры: нет.

Блок-схема функции



### Блок-схема программы с предопределенными блоками

### C:\Users\Dante\Desktop\01_all_2.png

### Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Исходные данные | | Результат работы программы |
| n | a1…an | Сообщение на экране |
| 1 | 7 | 1 2 3 1 2 3 3 | Number of maximum elements: 3 |

### Текст программы

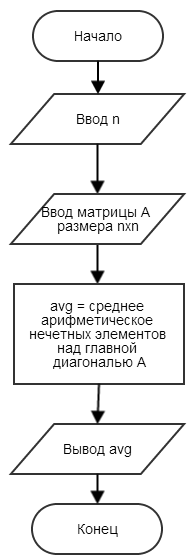
|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 |  |
| 03 | **namespace** \_01 |
| 04 | { |
| 05 | **class** Program |
| 06 | { |
| 07 | **static void** Main(**string**[] args) |
| 08 | { |
| 09 | Console.Write(*"Enter n, please: "*); |
| 10 | **int** n = Convert.ToInt32(Console.ReadLine()); |
| 11 |  |
| 12 | Console.WriteLine(*"Number of maximum elements: {0}"*, FindMaxCount(n)); |
| 13 | } |
| 14 |  |
| 15 | **static int** FindMaxCount(**int** n) |
| 16 | { |
| 17 | **int** tmp, count = 1; |
| 18 |  |
| 19 | Console.WriteLine(*"Enter {0} elements: "*, n); |
| 20 | Console.Write("\t1: "); |
| 21 | **int** max = Convert.ToInt32(Console.ReadLine()); |
| 22 |  |
| 23 | **for** (**int** i = 1; i < n; i++) |
| 24 | { |
| 25 | Console.Write(*"\t{0}: "*, i + 1); |
| 26 | tmp = Convert.ToInt32(Console.ReadLine()); |
| 27 |  |
| 28 | **if** (tmp > max) |
| 29 | { |
| 30 | max = tmp; |
| 31 | count = 1; |
| 32 | } |
| 33 | **else if** (tmp == max) |
| 34 | { |
| 35 | count++; |
| 36 | } |
| 37 | } |
| 38 |  |
| 39 | **return** count; |
| 40 | } |
| 41 | } |
| 42 | } |

# Задача 2

Дан массив размером , элементами которого являются целые числа. Подсчитать среднее арифметическое нечётных элементов, расположенных выше главной диагонали.

## Решение

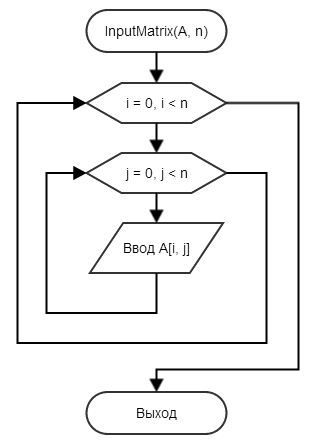
### Блок-схема алгоритма



### Описание подпрограмм

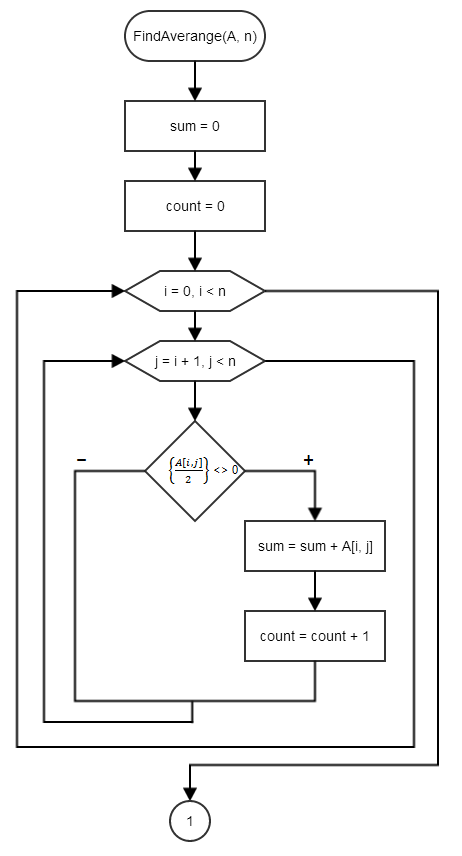
1. Спецификация функции InputMatrix
   1. Заголовок: static void InputMatrix(out int[,] A, int n)
   2. Назначение: вводит матрицу размера и сохраняет ее в двумерный массив A.
   3. Входные параметры: n
   4. Выходные параметры: A

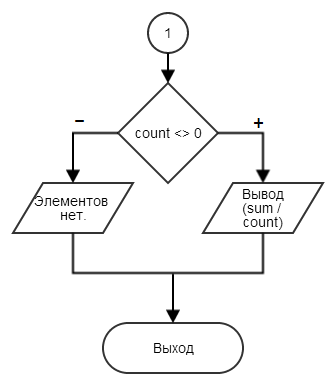
Блок-схема функции



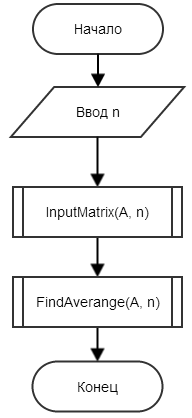
1. Спецификация функции FindAverange
   1. Заголовок: static void FindAverange(int[,] A, int n)
   2. Назначение: ищет в (размера ) нечетные элементы выше главной диагонали и высчитывает их среднее арифметическое, а затем выводит его, если оно есть, или ошибку в противоположном случае.
   3. Входные параметры: A, n.
   4. Выходные параметры: нет.

Блок-схема функции





### Блок-схема алгоритма с предопределенными блоками



### Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Исходные данные | | Результат работы программы |
| n | a1,1…an,n | Сообщение на экране |
| 1 | 3 | 1 3 5 2 4 6 3 5 7 | Averange: 4.00 |
| 2 | 3 | 1 2 4 3 6 8 0 1 2 | Don’t have odd elements. |

### Текст программы

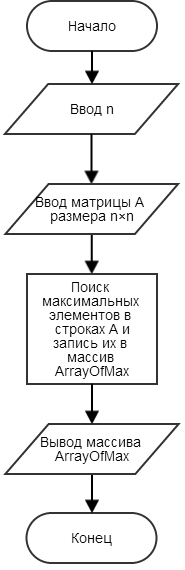
|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Threading; |
| 03 |  |
| 04 | **namespace** \_02 |
| 05 | { |
| 06 | **class** Program |
| 07 | { |
| 08 | **static void** Main(**string**[] args) |
| 09 | { |
| 10 | Thread.CurrentThread.CurrentCulture =  **new** System.Globalization.CultureInfo("ru-RU"); |
| 11 | Thread.CurrentThread.CurrentCulture.NumberFormat.NumberDecimalSeparator   = "."; |
| 12 |  |
| 13 | Console.Write(*"Enter n, please: "*); |
| 14 | **int** n = Convert.ToInt32(Console.ReadLine()); |
| 15 | **int**[,] A; |
| 16 | InputMatrix(**out** A, n); |
| 17 | FindAverange(A, n); |
| 18 | } |
| 19 |  |
| 20 | **static void** InputMatrix(**out** **int**[,] A, **int** n) |
| 21 | { |
| 22 | A = **new** **int**[n, n]; |
| 23 | Console.WriteLine(*"Enter nxn matrix's elements one by one, please:"*); |
| 24 | **for** (**int** i = 0; i < n; i++) |
| 25 | **for** (**int** j = 0; j < n; j++) |
| 26 | { |
| 27 | Console.Write(*"\t{0}, {1}: "*, i + 1, j + 1); |
| 28 | A[i, j] = Convert.ToInt32(Console.ReadLine()); |
| 29 | } |
| 30 | } |
| 31 |  |
| 32 | **static void** FindAverange(**int**[,] A, **int** n) |
| 33 | { |
| 34 | **int** sum = 0, count = 0; |
| 35 | **for** (**int** i = 0; i < n; i++) |
| 36 | **for** (**int** j = i + 1; j < n; j++) |
| 37 | { |
| 38 | **if** ((A[i, j] % 2) != 0) |
| 39 | { |
| 40 | sum += A[i, j]; |
| 41 | count++; |
| 42 | } |
| 43 | } |
| 44 |  |
| 45 | **if** (count != 0) |
| 46 | { |
| 47 | Console.WriteLine(*"Averange: {0:N2}"*, (**double**)sum / count); |
| 48 | } |
| 49 | **else** |
| 50 | { |
| 51 | Console.WriteLine(*"Don't have odd elements."*); |
| 52 | } |
| 53 | } |
| 54 | } |
| 55 | } |

# Задача 3

Дан массив размером , элементами которого являются целые числа. Найти максимальный элемент в каждой строке и записать данные в новый массив.

## Решение

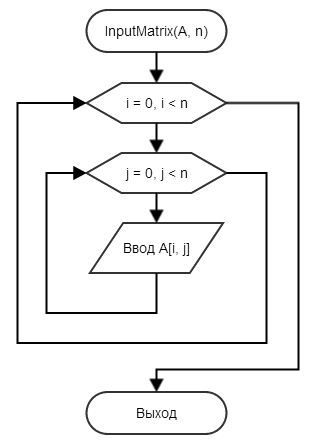
### Блок-схема алгоритма



### Описание подпрограмм

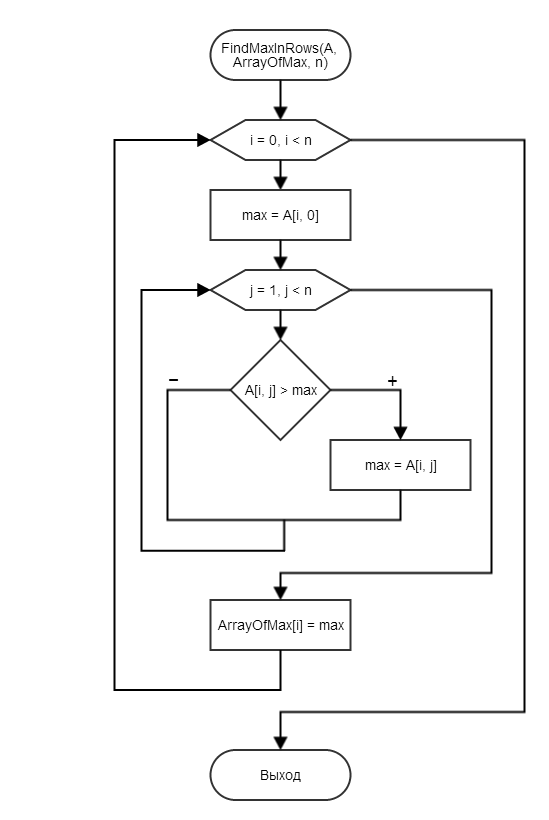
1. Спецификация функции InputMatrix
   1. Заголовок: static void InputMatrix(out int[,] A, int n)
   2. Назначение: вводит матрицу размера и сохраняет ее в двумерный массив A.
   3. Входные параметры: n
   4. Выходные параметры: A

Блок-схема функции



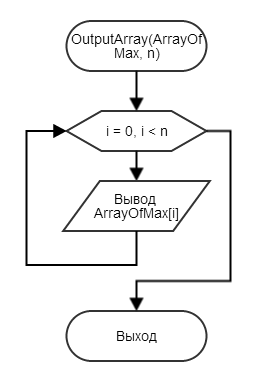
1. Спецификация функции FindMaxInRows
   1. Заголовок: static void FindMaxInRows(int[,] A, out int[] ArrayOfMax, int n)
   2. Назначение: поиск максимальных элементов в строках матрицы A размера и запись их в массив ArrayOfMax.
   3. Входные параметры: A, n
   4. Выходные параметры: ArrayOfMax

Блок-схема функции



1. Спецификация функции OutputArray
   1. Заголовок: static void OutputArray(int[] ArrayOfMax, int n)
   2. Назначение: выводит на экран элементы массива ArrayOfMax длины n.
   3. Входные параметры: ArrayOfMax
   4. Выходные параметры: нет

Блок-схема функции



### Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Исходные данные | | Результат работы программы |
| n | a1,1…an,n | Сообщение на экране |
| 1 | 3 | 1 2 3 4 5 6 7 8 9 | Array of maximum elements in rows:  1: 3  2: 6  3: 9 |

### Текст программы

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 |  |
| 03 | **namespace** \_03 |
| 04 | { |
| 05 | **class** Program |
| 06 | { |
| 07 | **static void** Main(**string**[] args) |
| 08 | { |
| 09 | Console.Write(*"Enter n, please: "*); |
| 10 | **int** n = Convert.ToInt32(Console.ReadLine()); |
| 11 | **int**[,] A; |
| 12 | **int**[] ArrayOfMax; |
| 13 | InputMatrix(out A, n); |
| 14 | FindMaxInRows(A, **out** ArrayOfMax, n); |
| 15 | OutputArray(ArrayOfMax, n); |
| 16 | } |
| 17 |  |
| 18 | **static void** InputMatrix(**out** **int**[,] A, **int** n) |
| 19 | { |
| 20 | A = **new** **int**[n, n]; |
| 21 | Console.WriteLine(*"Enter nxn matrix's elements one by one, please:"*); |
| 22 | **for** (**int** i = 0; i < n; i++) |
| 23 | **for** (**int** j = 0; j < n; j++) |
| 24 | { |
| 25 | Console.Write(*"\t{0}, {1}: "*, i + 1, j + 1); |
| 26 | A[i, j] = Convert.ToInt32(Console.ReadLine()); |
| 27 | } |
| 28 | } |
| 29 |  |
| 30 | **static void** FindMaxInRows(**int**[,] A, **out** **int**[] ArrayOfMax, **int** n) |
| 31 | { |
| 32 | ArrayOfMax = **new** **int**[n]; |
| 33 | **int** max; |
| 34 |  |
| 35 | **for** (**int** i = 0; i < n; i++) |
| 36 | { |
| 37 | max = A[i, 0]; |
| 38 | **for** (**int** j = 1; j < n; j++) |
| 39 | { |
| 40 | **if** (A[i, j] > max) |
| 41 | { |
| 42 | max = A[i, j]; |
| 43 | } |
| 44 | } |
| 45 | ArrayOfMax[i] = max; |
| 46 | } |
| 47 | } |
| 48 |  |
| 49 | **static void** OutputArray(**int**[] ArrayOfMax, **int** n) |
| 50 | { |
| 51 | Console.WriteLine(*"Array of maximum elements in rows:"*); |
| 52 | **for** (**int** i = 0; i < n; i++) |
| 53 | { |
| 54 | Console.WriteLine(*"\t{0}: {1}"*, i + 1, ArrayOfMax[i]); |
| 55 | } |
| 56 | } |
| 57 | } |
| 58 | } |