



Программирование на C#

Семинар №11

Модуль №1

Тема:

Представление матриц разными способами. Инициализация элементов многомерных массивов.



Задания преподавателя к семинару

Выполняем задания категорий ToDo и Self.

Уровень сложности заданий для реализации и последовательность проектов выбирайте сами.



Полезные материалы к семинару

1. Массивы. Многомерные массивы. Массивы массивов.

<https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/builtin-types/arrays#multidimensional-arrays>

2. Класс Array. Методы <https://learn.microsoft.com/ru-ru/dotnet/api/system.array?view=net-7.0>

3. Класс Array [<https://docs.microsoft.com/ru-ru/dotnet/api/system.array?view=net-5.0>

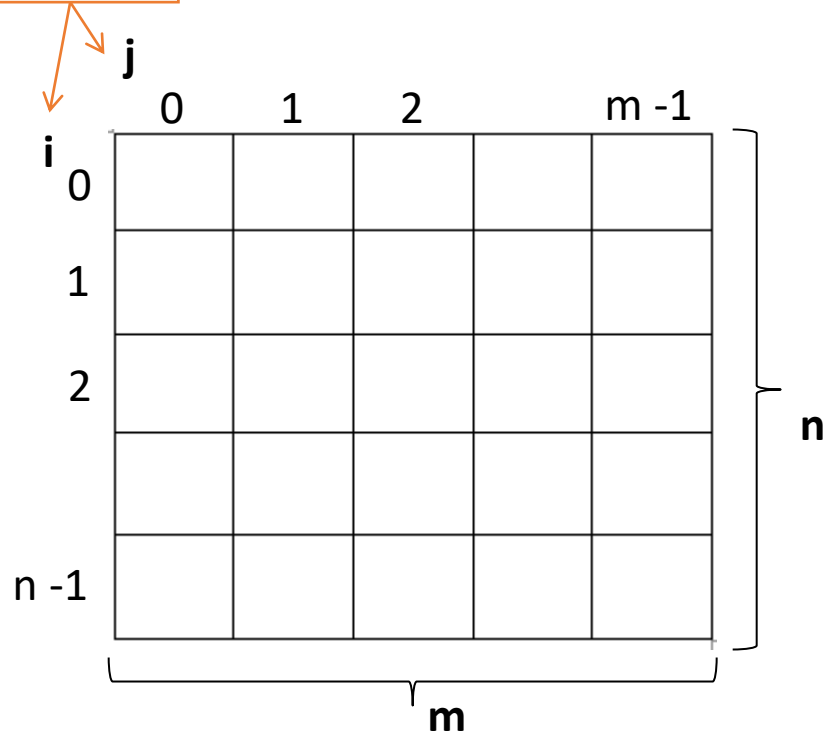
ToDo 01. Пропишите Условие Попадания Элементов На Побочную Диагональ

Матрица как двумерный массив:

```
int m = 5, n = 5
```

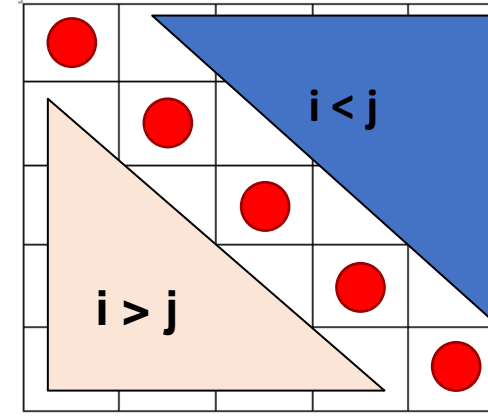
```
int[,] matr = new int[n, m]
```

индексы

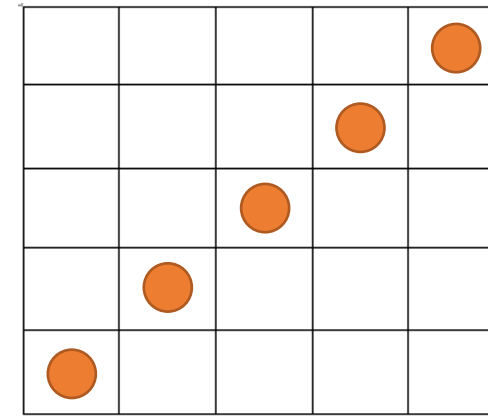


Главная диагональ

$i = j$



Побочная диагональ ???





Demo 01. Создать двумерный массив и инициализировать элементы.

Используя свойства и методы класса Array вывести:

- ранг массива,
- общее число его элементов,
- число элементов по разным измерениям, предельные значения всех индексов,
- признак фиксированных размеров...

```
int[,] matr = new int[3, 4] { { 0, 1, 3, 4 }, { 5, 6, 7, 8 }, { 9, -1, -2, -3 } };
```

```
Console.WriteLine("matr.GetType() = " + matr.GetType());
```

```
Console.WriteLine("matr.IsFixedSize = " + matr.IsFixedSize);
```

```
Console.WriteLine("matr.Rank = " + matr.Rank);
```

```
Console.WriteLine("matr.Length = " + matr.Length);
```

```
Console.WriteLine("matr.GetLength(1) = " + matr.GetLength(1));
```

```
Console.WriteLine("matr.GetUpperBound(1) = " +  
matr.GetUpperBound(1));
```

Класс Array. Методы <https://learn.microsoft.com/ru-ru/dotnet/api/system.array?view=net-7.0>



Демо 01. Продолжение.

Вывести элементы массива, используя цикл foreach

Вывести элементы массива по строкам (в виде таблицы)...

```
foreach (int memb in matr)//Все элементы матрицы в одну строку.
```

```
{  
    Console.Write("{0,3}", memb);  
}
```

Свойство, осуществляет
переход новую строку

```
Console.WriteLine(Environment.NewLine);
```

Длина по первому
измерению (строки)

```
// Вывод матрицы по строкам.
```

```
for (int i = 0; i < matr.GetLength(0); i++, Console.WriteLine())  
    for (int j = 0; j < matr.GetLength(1); j++)  
        {Console.Write("{0,3}", matr[i, j]);}
```

Длина по второму
измерению (столбцы)

Self 01. Задание.



Получить от пользователя целое число N . Создать двумерный массив размера в $N \times N$ и заполнить его по правилу:

$$\begin{pmatrix} 1 & 2 & \dots & N-2 & N-1 & N \\ 2 & 3 & \dots & N-1 & N & 1 \\ 3 & 4 & \dots & N & 1 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ N-1 & N & 1 & 2 & \dots & N-2 \\ N & 1 & 2 & \dots & \dots & N-1 \end{pmatrix}$$

Демо 02 . Формирование Матрицы по Правилу.



Ввести положительные значения N и M. Построить двумерный целочисленный массив (матрицу) с размерами N на M, элементы которого $a[i, j] = (i+1) * (2*j+1)$, для i от 0 до (N-1), j от 0 до (M-1).

Вывести матрицу в виде таблицы, а также значения свойств Rank и Length.

```
uint N, M;    // Размеры массива.
int[,] ar;    // Ссылка на массив.
do
{
    do
    { Console.Write("Введите число строк (N>0): "); }
    while (!uint.TryParse(Console.ReadLine(), out N) || N == 0);

    do
    { Console.Write("Введите число столбцов (M>0): "); }
    while (!uint.TryParse(Console.ReadLine(), out M) || M == 0);
```


Demo 02. Продолжение



```
ar = new int[N, M];  
    for (int i = 0; i < N; i++)  
        for (int j = 0; j < M; j++)  
            { ar[i, j] = (i + 1) * (2 * j + 1); } // Заполняем матрицу по правилу.
```

```
Console.WriteLine("Элементы массива:");  
for (int i = 0; i < N; i++, Console.WriteLine())  
    for (int j = 0; j < M; j++)  
        { Console.Write("{0,3}", ar[i, j]); }
```

```
Console.WriteLine("ar.Length:  " + ar.Length);  
Console.WriteLine("ar.Rank:    " + ar.Rank);  
Console.WriteLine("Для выхода нажмите клавишу ESC");
```

```
} while (Console.ReadKey(true).Key != ConsoleKey.Escape) ;
```

```
C:\Users\Irina\Desktop\Оперативно семинары 1 модуль\Переделя  
Введите число строк (N>0): 4  
Введите число столбцов (M>0): 6  
Элементы массива:  
  1  3  5  7  9 11  
  2  6 10 14 18 22  
  3  9 15 21 27 33  
  4 12 20 28 36 44  
ar.Length:  24  
ar.Rank:    2  
Для выхода нажмите клавишу ESC
```

Весь код заключен в блок повтора – цикл с постусловием.

ToDo 02. Задание к Demo 02



1. Формирование матрицы вынести в отдельный метод.
2. Написать метод, заменяющий в квадратной матрице, переданной в качестве параметра, все элементы, стоящие под побочной диагональю нулями. Если матрица не квадратная, никаких действий не производить.
3. Применить метод к матрице, сформированной в задаче. Результат вывести на экран.

Демо 03. Переброска строк матрицы поэлементно



Дана вещественная таблица размером $N \times M$ элементов. Размеры таблицы вводятся с клавиатуры. Поменять местами строки таблицы по правилу: строка с номером 0 меняется с последней, строка с номером 1 с предпоследней и т.д.

Пример: Таблица исходная ->

1.0	2.0	3.0	4.0
5.0	6.0	7.0	8.0
9.0	10.0	11.0	12.0

Таблица перевернутая ->

9.0	10.0	11.0	12.0
5.0	6.0	7.0	8.0
1.0	2.0	3.0	4.0

Решение "в лоб": таблица рассматривается как двумерный массив. Элементы строк физически меняются местами в памяти.

Demo 03. Переброска Строк Матрицы Поэлементно



```
static void ReadValue(double[,] a, int n, int m)
{
    int i, j;
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
        {
            Console.WriteLine("Элемент[{0},{1}]: ", i, j);
            a[i, j] = double.Parse(Console.ReadLine());
        }
}

// a - исходная таблица
// ном1 - номер первой строки в паре
// ном2 - номер второй строки в паре
// m - количество столбцов

static void Change (double[,] a, int ном1, int ном2, int m)
{
    int j; // номер столбца
    double b; // буфер для временного хранения элемента таблицы

    for (j = 0; j < m; j++)
    {
        b = a[ном1, j];
        a[ном1, j] = a[ном2, j];
        a[ном2, j] = b;
    }
}
```

Demo 03. Main()



```
int n, //Количество строк.
m, //Количество столбцов.
mp, //Количество пар строк.
ном1, //Номер первой строки в паре.
ном2; //Номер второй строки в паре.
    double[,] a; //Исходная матрица.
    Console.Write("Сколько строк: ");
    n = int.Parse(Console.ReadLine());
    Console.Write("Сколько столбцов: ");
    m = int.Parse(Console.ReadLine());

    a = new double[n, m];
    ReadValue(a, n, m);
    Console.WriteLine("\nИсходная таблица");
    Print_Matrix(a, n, m, "{0,8:f2}");

    mp = n/2;
    for(ном1=0,ном2=n-1; ном1<mp; ном1++,ном2--)
        { Change(a, ном1, ном2 ,m); }

    Console.WriteLine("\nТаблица после перестановки строк");
    Print_Matrix(a, n, m, "{0,8:f2}");
```

ToDo 03. Написать метод
вывода матрицы

Self. Задачи.

Self 02. Сформировать и заполнить случайными значениями целочисленную матрицу размером $M \times N$ (M и N задаются с клавиатуры). На экран вывести сумму и произведение элементов k -ой строки (k – задается с клавиатуры).



Self 03. Сформировать матрицу целых чисел $N \times M$. Задать инициализацию матрицы при помощи датчика случайных чисел в диапазоне $[-5 ; 5]$. Отсортировать строки матрицы без использования методов класса Array. Сортировку массива вынести в метод. Простейший **алгоритм сортировки**:

1. Каждый элемент массива сравнивается с последующим и если элемент $[i] >$ элемент $[i+1]$ происходит замена. Таким образом самые "легкие" элементы "всплывают" - перемещаются к началу списка, а самые тяжелые "тонут" - перемещаются к концу.
2. Повторяем шаг 1 $n-1$ раз, где n – количество элементов в строке матрицы.

Self 04. Сформировать для матрицы целых чисел $N \times M$ методы:

- заполнения при помощи датчика случайных чисел,
- заполнения при помощи ввода с клавиатуры,
- вывода на экран и подсчета сумм столбцов с занесением в вектор
- вывода вектора

Организовать выбор способа заполнения матрицы в виде меню.



Self 05. Зеркально отобразить элементы нижнетреугольной части квадратной матрицы, размер которой задает пользователь, относительно побочной диагонали. Матрица заполняется случайными числами от -9,25 до 9,33

Self 06. Создать матрицу целых чисел A размером $N \times M$ и заполнить ее случайными числами из диапазона от 10 до 100. Значения N и M вводятся с клавиатуры. Запомнить в массиве B минимальный элемент каждой строки матрицы. Вывести: исходную матрицу A и массив B .

Пример:

Матрица A	23	10	45
	12	15	9
Массив B	10	9	

Self 07 Квадратная матрица символьного типа размером $N \times N$ заполняется от датчика случайных чисел по правилу: в строки с четными номерами заносятся прописные латинские буквы, а в строки с нечетными номерами – цифры. Нумерация строк выполняется с нуля. Номер 0 – четный. Значение N определяет пользователь. Сформировать строку символов путем копирования в нее символов на главной диагонали. Просмотр диагонали выполняется сверху вниз. Для приведенного примера строка должна иметь вид: A1A9V.



A	M	W	Q	E
6	1	7	8	3
Q	Z	A	D	T
4	2	3	9	5
A	X	R	C	V