**Теория**

В примере ниже показан код для объявления двумерного массива из пяти строк и трех столбцов (5 х 3).

* Объявление массива:

|  |
| --- |
| **int**[,] arr = new **int**[5, 3]; |

* Инициализация массива значениями:

|  |
| --- |
| *// 1. Инициализация массива при объявлении:*  **int**[,] arr1 = new **int**[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  *// Тот же массив с заданными размерами:*  **int**[,] arr1 = new **int**[4, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };    *// 2. Инициализация массива без указания размерности:*  **int**[,] arr2 = { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };    *// 3. Инициализация массива после его объявления:*  **int**[,] arr;  arr = new **int**[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };    *// Присваивание значения конкретному элементу массива:*  arr[2, 1] = 25; |

* Доступ к элементам массива:

|  |
| --- |
| **int**[,] arr1 = new **int**[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  System.Console.WriteLine(arr1[0, 0]); *//1*  System.Console.WriteLine(arr1[0, 1]); *//2*  System.Console.WriteLine(arr1[1, 0]); *//3*  System.Console.WriteLine(arr1[1, 1]); *//4*  System.Console.WriteLine(arr1[3, 0]); *//7* |

* Получение значения определенного элемента массива и присвоение его переменной:

|  |
| --- |
| **int** elementValue = arr1[2, 1]; |

* Получение общего количества элементов (или размерности массива):

|  |
| --- |
| **int**[,] arr = new **int**[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  **var** arrayLength = arr.Length;  **var** total = 1;  **for** (**int** i = 0; i < arr.Rank; i++) {  total \*= arr.GetLength(i);  }  System.Console.WriteLine("{0} равно {1}", arrayLength, total); *// 8 равно 8* |

* Получение количества строк и столбцов:

|  |
| --- |
| **int**[,] matrix = new **int**[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };  **int** r=matrix.GetLength(0); *// строки*  **int** c=matrix.GetLength(1); *// столбцы* |

**Задания и лабораторные работы**

**Лабораторная работа 1.**  
    
**Выполнить:** Инициализируйте двумерный массив\* с именем arr2d заданными целочисленными значениями: ((1, 2); (3, 4); (5, 6)).  
**1)** Сначала выведите **все** элементы массива.  
**2)** После этого выведите следующие элементы в консоль (элементы для вывода отмечены синим цветом):

(1, 2)

(3, 4)

(5, 6)

\* Двумерный массив может называться *матрицей*.

**Пример выполнения:**

Массив:

1 2

3 4

5 6

1-й элемент = 3, 2-й элемент = 5

[Название проекта: Lesson\_10Lab1, название файла L10Lab1.cs]

 Алгоритм:

* Создайте новый проект с именем файла, как указано в задаче.
* Подключите следующую библиотеку, чтобы не печатать каждый раз название класса console:

using **static** System.Console;

* Чтобы объявить двумерный массив целочисленного типа с именем arr2d, мы можем использовать оператор int[,] arr2;. Сразу в этой же строке инициализируем массив значениями. Для этого добавьте следующий код в функцию Main:

**int**[,] arr2d= { {1, 2}, {3, 4}, {5, 6} };

Теперь у нас есть двумерный массив со значениями.

* Чтобы вывести все элементы массива, мы должны использовать вложенные циклы; внешний цикл должен перебирать строки, а внутренний цикл должен перебирать элементы в каждой строке. Внутри внутреннего цикла будем выводить значения элементов. Чтобы получить доступ к каждому элементу по его индексу, необходимо использовать счетчики цикла в качестве индексов элемента:

for (**int** i = 0; i < arr2d.GetLength(0); i++)

{

for (**int** j = 0; j < arr2d.GetLength(1); j++)

{

Console.Write($"{arr2d[i, j]} ");

}

Console.WriteLine();

}

Чтобы вывести значение конкретного элемента в окно консоли, нам нужно понять, в какой строке и столбце находится элемент. Обратите внимание, что индексы начинаются со значения 0. Итак, к элементу со значением **3** можно получить доступ как arr2[1,0] (где **1** — строка и **0** — столбец), а к элементу со значением **5** можно получить доступ так: arr2[2,0].

* Вывод элементов:

Console.WriteLine($"1-й элемент = {arr2d[1,0]}, 2-й элемент = {arr2d[2, 0]}");

* Запустите приложение еще раз и проверьте выходные данные.
* Поэкспериментируйте с различными значениями.
* Добавьте комментарий с текстом задачи и сохраните проект.

**Задание 1:**  
Приведены значения температуры воздуха за 4 дня; значения взяты с трех метеостанций, расположенных в разных регионах страны:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Station number** | **1-й день** | **2-й день** | **3-й день** | **4-й день** |
| 1 | -8 | -14 | -19 | -18 |
| 2 | 25 | 28 | 26 | 20 |
| 3 | 11 | 18 | 20 | 25 |

То есть в двумерном массиве это выглядело бы так:

|  |  |  |  |
| --- | --- | --- | --- |
| t[0,0]=-8; | t[0,1]=-14; | t[0,2]=-19; | t[0,3]=-18; |
| t[1,0]=25; | t[1,1]=28; | t[1,2]=26; | t[1,3]=20; |
| t[2,0]=11; | t[2,1]=18; | t[2,2]=20; | t[2,3]=25; |

**Указание:** Инициализация массива:

|  |
| --- |
| **int** [,] t = {  {-8,-14,-19,-18},  { 25,28, 26, 20},  { 11,18, 20, 25}  }; |

**Выполнить:**

1. Выведите значение температуры на **2-й** метеостанции в течение **4-го** дня и на **3-й** метеостанции в течение **1-го** дня (результат должен быть 20 и 11).
2. Выведите значения температуры **всех метеостанций** в течение **2-го** дня (необходим for цикла).
3. Выведите значения температуры **всех метеостанций** в течение всех дней.
4. Рассчитайте среднюю температуру на **3-й** метеостанции (необходим for цикла).
5. Выведите дни и номера метеостанций, где температура была в диапазоне **24**—**26** градусов.

**Пример вывода:**

1-е задание: 20 and 11

2-е задание: 25 28 26 20

3-е задание:

-8 -14 -19 -18

25 28 26 20

11 18 20 25

4-е задание: 18

5-е задание:

станция 1 день 0

станция 1 день 2

станция 2 день 3

[Название проекта: Lesson\_10Task1, название файла L10Task1.cs]

**Задание 2:**

**To do:** Вводятся элементы двумерного массива *2 х 3* ([2,3]). Посчитайте сумму элементов.  
     
**Указание 1:**Для объявления массива размерностью *2 х 3* используйте код:

**int**[,] arr=new **int**[2, 3];

**Указание 2:** Для ввода значений элементов массива и перебора элементов, воспользуйтесь вложенными циклами:

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 3; j++)

{

...

}

}

**Пример вывода:**

Введите 6 значений для матрицы 2 х 3

2

5

1

6

7

8

Матрица:

2 5 1

6 7 8

Сумма = 29

[Название проекта: Lesson\_10Task2, название файла L10Task2.cs]

**Задание 3:**

**Выполнить:** Вводятся элементы двумерного массива 4 х 3. Посчитайте количество положительных элементов.

**Пример вывода:**

Пожалуйста, введите значения матрицы 4 х 3

2 5 -1 6 7 8 1 8 6 -3 1 -6

Матрица:

2 5 -1

6 7 8

1 8 6

-3 1 -6

кол-во положительных = 9

[Название проекта: Lesson\_10Task3, название файла L10Task3.cs]

**Лабораторная работа 2. Использование функции Random**

**Выполнить:** Создайте метод FillMatrix для заполнения двумерного массива случайными числами в диапазоне от **-10** до **10**.

**Указание**: Заголовок функции FillMatrix должен быть таким:

|  |
| --- |
| FillMatrix(matrix, minValue, maxValue); |

**Пример выполнения:**

Введите кол-во строк:

3

Введите кол-во столбцов:

4

Массив:

-10 -7 -3 -4

-6 1 4 0

-5 -8 -1 -5

[Название проекта: Lesson\_10Lab2, название файла L10Lab2.cs]

 Алгоритм:

* Создайте новый проект с названием файла, которое указано в задании.
* Подключите следующую библиотеку чтобы не указывать название класса console:

using static System.Console;

* В функции Main добавьте код для того, чтобы попросить пользователя ввести количество строк и количество столбцов матрицы. Присвойте введенные значения переменным:

...

Console.WriteLine("введите кол-во строк");

**int** rows = int.Parse(ReadLine());

Console.WriteLine("введите кол-во столбцов");

**int** columns = int.Parse(ReadLine());

* Объявите матрицу:

**int**[,] matrix = **new int**[rows,columns];

* Инициализируйте значениями переменные для границ генерируемого диапазона:

**int** minValue=-10;

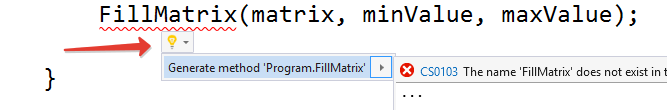
**int** maxValue=10;

Чтобы сделать программу более эффективной лучше использовать **ReSharper** (для инициализации метода):

* Вызовите метод FillMatrix в коде функции Main (несмотря на то, что мы еще ее не создали). В функцию Main() введите следующий код:

FillMatrix(matrix, minValue, maxValue);

* Название функции подчеркнуто. Если поднести курсор мыши, вы увидите изображение с желтой лампочкой. Значит, мы можем использовать *ReSharper comments*, щелкнув по лампочке (или Alt+Enter). Следует выбрать пункт *«Генерировать метод Program.FillMatrix»*.



* Заголовок функции есть в задании.
* Change the code:

FillMatrix(**int**[,] matrix, **int** minValue = -10, **int** maxValue = 10);

* Метод с тремя параметрами:

matrix — a declared matrix without initialization;  
minValue -устанавливать нижнюю границу для функции *Random*, значение по умолчанию — **-10**  
maxValue — устанавливать верхнюю границу для функции *Random*, значение по умолчанию **10**

* Внутри созданной функции установите генератор случайных чисел:

Random rand = new Random();

* Сгенерируйте значения для матрицы и присвойте их элементам (границы генерируемых чисел: от minValue до maxValue):

...

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

matrix[i, j] = rand.Next(minValue, maxValue);

Write(matrix[i, j].ToString().PadLeft(4));

}

WriteLine();

}

Обратите внимание, что matrix.GetLength(0) означает число столбцов, а matrix.GetLength(1)означает число столбцов.

Для прохода по элементам матрицы необходимо использовать вложенный цикл. Счетчики циклов должны быть разными переменными, можно использовать для их названий i и j. Метод GetLength(0) возвращает кол-во строк матрицы, метод GetLength(1) возвращает кол-во столбцов.

* Щелкните CTRL+a, CTRL+k, CTRL+F для форматирования вашего кода.
* Запустите программу.
* Добавьте комментарии. Загрузите файл L10Lab2.cs.

**Задание 4:**

**Выполнить:** Создайте метод FillMatrix для заполнения двумерного массива случайными числами от *-15* до *15*. Создайте еще один метод для нахождения минимального и максимального элементов массива (FindMinMaxArr).  
     
**Указание 1:**Сигнатура (заголовок) метода FindMinMaxArr должен быть таким:

private static **void** FindMinMaxArr(**int**[,] matrix, ref **int** min, ref **int** max)

**Указание 2:**Алгоритм поиска минимального и максимального элемента можно посмотреть в одном из прошлых уроков [Урок 5](https://labs-org.ru/c-sharp5/) -> *Лаб. раб. 4*.

**Пример вывода:**

введите кол-во строк

4

введите кол-во столбцов

5

Массив:

10 2 9 3 4

-3 -10 -14 -4 2

2 -9 11 3 -10

-1 -13 -5 -2 3

min = -14, max = 11

[Название проекта: Lesson\_10Task4, название файла L10Task4.cs]

**Лабораторная работа 3. Обмен определенных столбцов матрицы**

**Выполнить:** Создайте метод ChangeColumns для обмена определенных столбцов матрицы размера 3 х 4 (порядковые номера столбцов вводятся).

**Указание**: Помимо метода ChangeColumns необходимо создать метод FillMatrix для заполнения матрицы случайными числами, и метод PrintMatrix для вывода элементов матрицы.

**Пример вывода:**

Матрица:

-10 -7 -3 -4

-6 1 4 0

-5 -8 -1 -5

введите номера столбцов для обмена (первый столбец - 0-й):

1

2

Результирующая матрица:

-10 -3 -7 -4

-6 4 1 0

-5 -1 -8 -5

[Название проекта: Lesson\_10Lab3, название файла L10Lab3.cs]

 Fkujhbnv:

* Создайте новый проект и переименуйте главный файл согласно заданию.
* Необходимо создать матрицу с тремя строками и четырьмя столбцами. Объявите матрицу в функции Main:

int[,] matrix = **new** int[3, 4];

* Скопируйте код метода FillMatrix из *Лабораторной работы 2* и вставьте его в текущий проект. Или создайте метод заново. После этого у Вас будет код:

**static** void FillMatrix(int[,] matrix, int minValue = -10, int maxValue = 10) {

Random rand = new Random();

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

matrix[i, j] = rand.Next(minValue, maxValue);

}

}

}

* Создайте метод PrintMatrix для вывода элементов матрицы. Метод принимает один параметр — саму матрицу:

**static** void PrintMatrix(int[,] m)

{

for (int i = 0; i < m.GetLength(0); i++)

{

for (int j = 0; j < m.GetLength(1); j++)

{

Console.Write(m[i, j].ToString().PadLeft(4));

}

Console.WriteLine();

}

}

Чтобы выводить элементы с одинаковыми отступами построчно мы конвертировали их в строкой тип данных и использовали метод PadLeft для формирования одинаковых отступов слева.

* Затем в функции Main необходимо вызвать созданные методы: сначала — метод FillMatrix и затем — метод PrintMatrix.

...

FillMatrix(matrix);

Console.WriteLine("Матрица 3 x 4: ");

PrintMatrix(matrix);

* Затем необходимо ввести номера столбцов для обмена:

...

Console.WriteLine("введите номера столбцов для обмена (первый столбец - 0-й):");

int col1= int.**Parse**(Console.ReadLine());

int col2 = int.**Parse**(Console.ReadLine());

* Создайте метод ChangeColumns для обмена столбцов с указанными номерами:

static **void** ChangeColumns(int[,] matrix, int col1, int col2)

{

System.Diagnostics.Debug.Assert((col1 < matrix.GetLength(1)) && (col2 < matrix.GetLength(1)), "указанный номер столбца выходит за границы размерности массива!");

for (int i = 0; i < matrix.GetLength(0); i++)

{

int temp = matrix[i, col1];

matrix[i, col1] = matrix[i, col2];

matrix[i, col2] = temp;

// если используется Visual studio 2019:

// (matrix[i, col1], matrix[i, col2]) = (matrix[i, col2], matrix[i, col1]);

}

}

Счетчик i цикла проходится по строкам матрицы и в то же время каждый элемент col1 обменивается значениями с элементом col2 в каждой строке.

System.Diagnostics.Debug.Assert используется для того, чтобы вызвать исключение при вводе значения большего, чем кол-во столбцов матрицы.

* Вызовите метод в функции main. Затем вызовите метод PrintMatrix , чтобы увидеть результат:

...

ChangeColumns(matrix, col1, col2);

Console.WriteLine("The resulting matrix:");

PrintMatrix(matrix);

* Запустите программу, нажав горячие клавиши CTRL+F5.
* Загрузите файл L10Lab3.cs.

**Задание 5:**

**To do:** Создайте метод PlaceZero, который замещает некоторые элементы матрицы (5 х 5) нулями (0). Посмотрите на пример вывода, чтобы понять, какие элементы должны быть замещены нулями.  
     
**Указание 1:**Создайте методы для заполнения и вывода матрицы.  
**Указание 2:**Заголовок метода PlaceZero должен быть следующим:

|  |
| --- |
| **static** **void** PlaceZero(**int**[,] matrix) |

**Указание 3:**Для замены элементов нулями используйте циклы.  
    
**Пример вывода:**

Матрица:

10 2 9 3 6

3 10 14 4 8

2 9 11 3 2

1 13 5 2 7

12 3 11 4 5

Результирующая матрица:

0 0 0 0 0

0 10 14 4 0

0 9 11 3 0

0 13 5 2 0

0 0 0 0 0

[Название проекта: Lesson\_10Task5, название файла L10Task5.cs]

**Задание 6:**

**To do:** Создайте метод FindProductOfColumn для нахождения произведения элементов матрицы M-го столбца (матрица 3 х 4) (M вводится). M — это параметр out (вывода).

**Note**: Сигнатура метода FindProductOfColumn должна быть следующей:

|  |
| --- |
| **static** **void** FindProductOfColumn(**int**[,] matrix, **int** M,**out** **int** product) |

**Пример вывода:**

Матрица:

-10 -3 -7 -4

-6 4 1 0

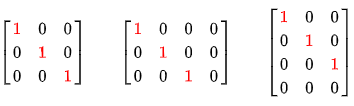
-5 -1 -8 -5

Введите номер столбца (первый столбец - 0-й):

2

Произведение элементов 2-го столбца = -64

[Название проекта: Lesson\_10Task6, название файла L10Task6.cs]

**Главная и побочная диагонали**  
    
В алгебре **главная диагональ** матрицы A — это все элементы матрицы Ai,j**, для которых i=j**. Элементы вне главной диагонали обозначены нулями, а элементы главной диагонали обозначены красными единицами:  
  
**Побочная диагональ** матрицы B размерности N — это элементы Bi,j, для которых i+j=N+1 для всех 1<=i,j<=N. Но если i и j начинаются с **нуля**, то формула будет i+j=N-1:  


**Лабораторная работа 4. Работа с диагоналями матрицы**

**Выполнить:** Создайте квадратную матрицу *M*-х-*M* (значение M вводится). Посчитайте сумму элементов побочной диагонали матрицы.

**Примерный вывод:**

Введите размерность матрицы:

3

Матрица 3-х-3:

10 -7 -3

6 1 4

5 -8 -1

сумма элементов побочной диагонали = 3

[Название проекта: Lesson\_10Lab4, название файла L10Lab4.cs]

 Алгоритм:

* Создайте новый проект и измените название главного файла согласно заданию.
* Попросите пользователя ввести размерность матрицы и присвойте введенное значение переменной M:

Console.WriteLine("Введите размерность матрицы:");

int M = int.**Parse**(Console.ReadLine());

* Мы собираемся создать матрицу размерностью M-х-M. Объявите матрицу в коде функции Main:

int[,] matrix = **new** int[M, M];

* Скопируйте код метода FillMatrix с предыдущей лабораторной работы и вставьте его до или после функции Main.
* Скопируйте код метода PrintMatrix и также вставьте его после метода FillMatrix.
* Создайте метод AntidiagonalSum, который будет суммировать элементы побочной диагонали матрицы.

**static** int AntidiagonalSum(int[,] matrix)

{

...

}

* Вспомним, что для индексов элементов побочной диагонали следует использовать формулу: i+j=M-1. Для прохода по элементам матрицы будем использовать вложенные циклы:

...

int sum = 0;

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

if ((i+j == M-1)

{

sum+ = matrix[i,j];

}

}

}

* Метод должен возвратить значение суммы, поэтому будем использовать ключевое слово return:

...

return sum;

* Вызовите метод внутри функции Main:

int result = AntidiagonalSum(matrix);

Console.WriteLine($"сумма элементов побочной диагонали = {result}");

* Запустите программу без отладки (CTRL+F5) и проверьте результат.
* Добавьте комментарий с заданием и загрузите главный файл.

**Задание 7:**

**Выполнить:** Create square matrix *M*-by-*M* (M is inputted). Calculate a product of its elements within a main diagonal.

**Указание**: a main diagonal elements are such as i=j.

**Примерный вывод:**

Введите размерность матрицы:

3

Матрица 3-х-3:

10 -7 -3

6 1 4

5 -8 -1

Произведение элементов главной диагонали = -10

[Название проекта: Lesson\_10Task7, название файла L10Task7.cs]

**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**Доп. задание 1:**

**Выполнить:** Запрашиваются натуральные M и N. Создайте метод FillMatrix для заполнения матрицы размерностью M-х-N; все элементы J-го столбца вычисляются по формуле X\*J (J - номер столбца, J = 1, ..., N), а X - число - аргумент метода.  
    
**Указание:** заголовок метода должен быть следующим:

|  |
| --- |
| **static** **int**[,] FillMatrix(**int** M, **int** N, **int** X) |

**Примерный вывод:**

введите кол-во строк:

3

введите кол-во столбцов:

4

Введите значение множителя (аргумента):

3

Матрица 3-х-3:

3 6 9 12

3 6 9 12

3 6 9 12

[Название проекта: Lesson\_10ExTask1, название файла L10ExTask1.cs]

**Доп. задание 2:**

**Выполнить:** Даны две матрицы одинаковой размерности (необходимо создать метод для заполнения матриц случайными числами). Посчитайте сумму элементов матрицы, используйте для этого еще одну матрицу.  
    
**Указание:** Заголовок метода:

|  |
| --- |
| **static** **int**[,] SumMat(**int**[,] a, **int**[,] b) |

**Примерный вывод:**

Matrix 1:

-3 8 3

-9 -6 4

6 4 -2

Matrix 2:

4 -4 1

-5 1 -1

4 8 -10

Result matrix:

1 4 4

-14 -5 3

10 12 -12

[Название проекта: Lesson\_10ExTask2, название файла L10ExTask2.cs]