# Arrays in C#

آرایه مجموعه یا collections از یک نوع داده است که مکانی از حافظه تحت یک نام آن ها را نگه داری می کند.

آرایه نوعی ساختار داده ترتیبی است sequential data structure که مجموعه ای داده های یک نوع را در خود نگه داری می کند.

**How does using [] this work in real memory?**

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

زمانی که از علامت [] استفاده می کنید به کامپایلر می گویید که نوع داده ی که در حال استفاده از آن هستید یک آرایه است و بلوکی از حافظه به آن تخصیص داده شود.

**Types of Arrays in C#:**

1. **Single dimensional array تک بعدی**
2. **Multi-dimensional array چند بعدی**

در آرایه تک بعدی دیتا به صورت row یا ردیف در کنار هم قرار می گیرند در حالی که در آرایه های چند بعدی داده های به صورت ردیف و ستون rows and columns در کنار هم قرار می گیرند.

آرایه های چند بعدی 2 نوع هستند

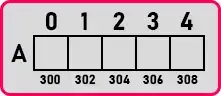
1. **Jagged array**: Whose rows and columns are not equal دندانه دار ردیف و ستون نا برابر
2. **Rectangular array**: Whose rows and columns are equal مستطیلی ردیف و ستون ها برابر

**نکته:** برای دسترسی به آرایه می توان از index استفاده کرده و همیشه اولین خانه دارای مقدار برابر است با صفر 0 و تعداد داده های آرایه نیز برابر است با تعداد خانه ها -1

ساختن آرایه با 5 خانه

**Example: int[] A = new int[5];**

فرض کنید خانه حافظه به شکل زیر است از 300 شروع شده و آرایه ما از نوع int است که هر کدام 2byte فضا می گیرد پس حافظه به شکل زیر خواهد بود.



در این مثال خانه a[0] برابر است با 300 و 3001 , a[1] برابر است با 302 و 303

Indexer

در آرایه در واقع یک pointer است که برای دسترسی به داده آرایه در یک خانه مشخص استفاده می شود و از 0 تا n-1 ادامه دارد.

Array is fast?

به دلیل این که مقادیر در حافظه به صورت پیوسته یا پشت سر هم قراردارند دسترسی به آن ها سریع است.

**2 نوع آرایه در سی شارپ داریم طول ثابت و متغییر در طول متغییر سایز آرایه به صورت خودکار با افزایش ایتم های جدید اضافه می شود.**

**نکته: array is reference type and heap memory is allocate**

**چرا می توان روی آرایه foreach زد؟**

به دلیل پیاده سازی رابط IEnumerable می توانیم بر روی عناصر آرایه پیمایش کنیم

Int[] arr = {1,23,4,5,6} روش initialize کردن آرایه.

Foreach

به صورت مخصوص برای دسترسی به عناصر آرایه ها ایجاد شده است و از عنصر اول تا آخر بدون نیاز به شرط پیمایش می کند.

Foreach vs for

For نیاز به دسترسی به index دارد ولی foreach مستقیما به داده دسترسی دارد

نکته : for برای دسترسی و assign کردن کاربرد دارد ولی foreach فقط برای دسترسی به داده.

نکته: دسترسی به عنصری از آرایه که خارج از محدوده است سبب exception می شود

**Array base class is implementing IList interface**

# 2D Arrays in C#

آرایه های که در سی شارپ که عناصر آن در ردیف و ستون rows & column ذخیره می شوند دو بعدی نام دارند.

انوع آرایه 2 بعدی

Rectangular در این نوع آرایه ها تعداد ردیف ها و ستون ها با هم برابر است

Jagged در این نوع آرایه ها تعداد ردیف ها و ستون ها با هم برابر نیست.

در این نوع آرایه ها به هر عنصر از آرایه با 2 index دسترسی می توان داشت المان ها در این نوع آرایه ها به صورت matrix ذخیره می شوند.

در این نوع آرایه ها index اول به ردیف row و index دوم به column یا ستون اشاره می کند.

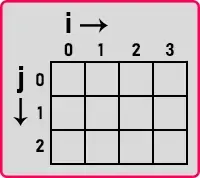
**int[,] matrix = new int[3,3];**



برای دسترسی به خانه ی اول matrix[0][0] به این روش عمل می کنیم.

**How 2D Array is Created and Accessed in C#?**

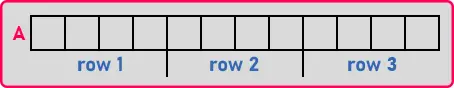
**int[,] A = new int[3,4];**



**برای دسترسی و چاپ عنصر آرایه از روش زیر استفاده می کنیم.**

**Console.WriteLine(A[1,2]);**

نحوه ذخیره سازی و دسترسی به عناصر آرایه 2 بعدی در حافظه Ram به این صورت است که در ردیف های پشت سر هم در حافظه داده ها ذخیره می شوند و تقسیم بندی منطقی ردیف ها انجام می شود



نحوه تعریف یک آرایه با 2 ردیف داده و 3 ستون

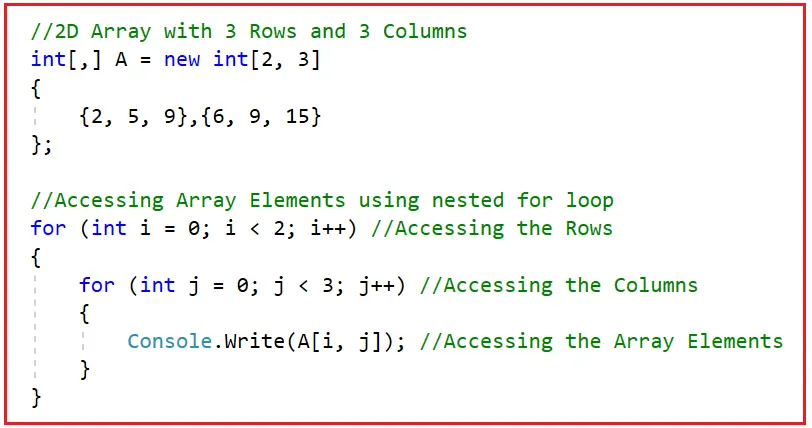
**int[,] A = new int[2,3]**  
**{**  
**{2, 5, 9},{6, 9, 15}**  
**};**

**OR**

**int[,] A = {{2, 5, 9},{6, 9, 15}};**

**Accessing the Elements of the 2D array in C#:**

برای دسترسی به عناصر آرایه های 2 بعدی به 2 حلقه تو در تو نیاز داریم یکی برای دسترسی به عناصر row و دیگری برای دسترسی به عناصر columns به مثال زیر توجه کنید.



**نکته** برای دسترسی به تعداد ردیف ها GetLength(0) و برای دسترسی به تعداد ستون ها GetLenght(1) استفاده می شود.

مثال پرکردن عناصر آرایه:

//Assigning values to the 2D array by using nested for loop

//arr.GetLength(0): Returns the size of the Row

//arr.GetLength(0): Returns the size of the Column

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

**{**

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

**{**

a += 5;

RectangleArray**[**i, j**]** = a;

**}**

**}**

**Jagged Array in C#:**

نوع دیگری از آرایه های 2 بعدی هستند که شباهتی بسیاری به آرایه های 2 بعدی مستطیلی دارد با این تفاوت که در این نوع آرایه ها می توان هر ردیف داده دارای تعداد ستون داده متفاوتی باشد مثلا ردیف اول دارای 4 ستون داده و ردیف دوم دارای 2 ستون داده باشد

به این نوع آرایه، آرایه ی از آرایه ها نیز می گویند به این دلیل که هر ردیف خود یک آرایه تک بعدی است.

در تعریف این نوع آرایه ها فقط کافی است تعداد row را مشخص کنید. و سپس برای هر row می توانید تعداد ستون ها را مشخص کنید.

**int [][] arr = new int[4][];**

**arr[0] = new int[5];** // we want five columns in the first row  
**arr[1] = new int[6];** // we want six columns in the first row  
**arr[2] = new int[4];** // we want four columns in the first row  
**arr[3] = new int[5];** // we want five columns in the first row

در ادامه نحوه دسترسی به عناصر آرایه را بررسی می کنیم.

//Printing the values of Jagged array using nested for loop

//It will print the default values as we are not assigning any

//values to the array

//GetLength(0): Returns the Size of the Rows (4)

Console.WriteLine**(**"Printing the Default Values of Jagged Array:"**)**;

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

**{**

//arr[i].Length: Returns the Length of Each Row

**for** **(int** j = 0; j **<** arr**[**i**]**.Length; j++**)**

**{**

Console.Write**(**arr**[**i**][**j**]** + " "**)**;

**}**

**}**

# Advantages and Disadvantages of Arrays in C#

مزایا Advantages

ذخیره کردن چندین مقدار از یک نوع تحت عنوان یک نام

برای پیاده سازی سایر ساختار های داده مانند linked list graphs stacks queues کاربرد دارد

آرایه های 2 بعدی برای پیاده سازی ماتریکس کاربرد دارند.

از نوع strongly type هستند.

چون از نوع strongly هستند مزایای دارند :

Performance اپلیکیشن ما بهتر می شود چون boxing , unboxing رخ نمی دهد و خطای RunTime رخ نمی دهد چون در زمان compile متوجه خطا ها می شویم.

**نکته : loosely type collection مثل ArrayList نوع های کالکشنی هستند که می توان در آن ها هر نوع داده ی را نگه داری کرد و در زمان کامپایل خطا نمی دهد.**

**معایت Disadvantages**

Fix size به معنی این است که زمانی که ارایه ایجاد می شود دیگر قابلیت increase or decrease برای آن آرایه وجود ندارد

اگر حافظه ی بیشتر از نیاز خود اختصاص دهیم waste می شود حافظه و اگر کمتر از نیاز تخصیص دهیم دچار مشکل می شویم

هیچ وقت نمی توانیم که مقداری را از middle of array اضافه کنیم یا همچنین نمی توان از وسط آرایه حذف کرد.