

جامعة البعث
الكلية التطبيقية
قسم تقنيات الحاسوب

برمجة متقدمة
السنة الثانية
عملي
محاضرة رابعة

م. سارة معروف

م. بتول اللبوس

م. زكي نقولا

المصفوفات متعددة الأبعاد Multi-dimensional Arrays

تعد المصفوفات متعددة الأبعاد Multi-dimensional Arrays من العناصر الأساسية في لغة البرمجة C# ، وتُستخدم لتنظيم البيانات في ترتيب متعدد الأبعاد. تتكون المصفوفات متعددة الأبعاد من مجموعة من العناصر التي يتم تخزينها في مواقع محددة ضمن الذاكرة. يمكن تعريف المصفوفات متعددة الأبعاد بأبعاد مختلفة، مثل المصفوفات ذات البُعدين (Two-Dimensional Arrays) و المصفوفات ذات الثلاثة أبعاد (Three-Dimensional Arrays) وهكذا.

تُعتبر المصفوفات متعددة الأبعاد Multi-dimensional Arrays أحد المكونات الأساسية والقوية في لغة البرمجة سي شارب (C#)، وهي توفر وسيلة فعالة لتخزين وتنظيم البيانات. المصفوفة متعددة الأبعاد هي مجموعة من العناصر ذات نفس النوع، ويتم الوصول إلى هذه العناصر باستخدام فهرس يشير إلى موقع كل عنصر داخل المصفوفة.

تتميز المصفوفات متعددة الأبعاد Multi-Dimensional Arrays في لغة سي شارب بقدرتها على تخزين البيانات بتنظيم مرن، حيث يمكن تعريف المصفوفات بأبعاد متعددة، مما يسمح بتمثيل بنية البيانات المعقدة بشكل فعال. على سبيل المثال، يمكن استخدام المصفوفات ثنائية الأبعاد لتمثيل الصور أو البيانات الجدولية.

تعريف المصفوفات متعددة الأبعاد Multi-dimensional Arrays

يمكن تعريف المصفوفات متعددة الأبعاد في C# باستخدام الكود التالي:

```
type[,] arrayName = new type[rowSize , columSize];
```

حيث:

- **type** يمثل نوع البيانات لعناصر المصفوفة.
- **arrayName** هو اسم المصفوفة التي ترغب في تعريفها.
- **rowSize** هو حجم المصفوفة في البُعد الأول (عدد الصفوف).
- **columnSize** هو حجم المصفوفة في البُعد الثاني (عدد الأعمدة).

على سبيل المثال، إذا كنا نرغب في تعريف مصفوفة ثنائية الأبعاد D2 لتخزين مجموعة من الأرقام الصحيحة، نكتب الكود التالي:

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

هذا الكود ستنشئ مصفوفة بإسم matrix ثنائية الأبعاد بحجم 3 صفوف و 4 أعمدة.

الوصول إلى عناصر المصفوفات متعددة الأبعاد:

يمكن الوصول إلى عناصر المصفوفات متعددة الأبعاد Multi-Dimensional Arrays في C# باستخدام مؤشرات (index) لكل بُعد. يتم تحديد المؤشرات بين قوسين مربعين [] عند الوصول إلى العنصر المطلوب في المصفوفة.

على سبيل المثال، إذا كان لدينا المصفوفة matrix التي تم تعريفها في المثال السابق، يمكننا تعيين قيمة للعنصر في الصف الأول والعمود الثاني كالتالي:

```
matrix[0,1] = 10;
```

وبالمثل، يمكننا الوصول إلى قيمة العنصر في الصف الثاني والعمود الثالث كالتالي:

```
int value = matrix[1, 2];
```

وبما أن المصفوفات متعددة الأبعاد هي هياكل بيانات مهمة في لغة البرمجة C# . ولكي تسمح للمطورين بتخزين وتنظيم البيانات في تنسيق ثنائي متعدد الأبعاد، وبشكل يسهل عليهم الوصول إلى البيانات والتعامل معها بشكل فعال، فهناك العديد من العمليات والدوال المتوفرة للعمل معها فيما شرح لطريقة هذا العمل.

العمل مع المصفوفات متعددة الأبعاد في لغة C#

يتضمن العمل مع المصفوفات متعددة الأبعاد الآتي:

الإشياء: يمكن إنشاء المصفوفات متعددة الأبعاد في لغة C# باستخدام الكلمة المحجوزة new متبوعة بنوع المصفوفة والأبعاد المحاطة بين قوسين مربعين. على سبيل المثال، لتهيئة مصفوفة ثنائية الأبعاد مكونة من 3 صفوف و 4 أعمدة، يمكنك استخدام الكود:

```
int[,] array = new int[3, 4];
```

الوصول إلى العناصر: يمكنك الوصول إلى العناصر في مصفوفة متعددة الأبعاد باستخدام المؤشرات الخاصة (Indexes) بكل بُعد. على سبيل المثال، للوصول إلى عنصر في الصف الثاني والعمود الثالث من مصفوفة ثنائية الأبعاد، يمكنك استخدام

```
array[1, 2];
```

- تبدأ الفهرسة للمؤشرات من الصفر، لذا يمكن الوصول إلى الصف أو العمود الأول باستخدام الفهرس 0.
- **الطول والرتبة (Length and Rank):** خاصية الطول Length لمصفوفة متعددة الأبعاد تمنحك إجمالي عدد العناصر في المصفوفة. بالإضافة إلى ذلك، توفر لك الخاصية Rank عدد الأبعاد في المصفوفة. على سبيل المثال، تحتوي المصفوفة ثنائية الأبعاد array2 D على رتبة 2.
- **التكرار على العناصر:** يمكنك استخدام الحلقات المتداخلة (Nested Loop) للتكرار على عناصر مصفوفة متعددة الأبعاد. (المزيد في مقال عن [الحلقات المتداخلة Nested Loop](#))، تتحكم الحلقة الخارجية في الصفوف، وتتحكم الحلقة الداخلية في الأعمدة (وحلقات إضافية لمزيد من الأبعاد). فيما يلي مثال يطبع جميع العناصر في مصفوفة ثنائية الأبعاد:

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

```
{
```

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

- **Array.Copy:** يمكنك استخدام طريقة `Array.Copy` لنسخ العناصر من مصفوفة متعددة الأبعاد إلى أخرى. تسمح لك هذه الطريقة بتحديد المصفوفة المصدر، والمؤشرات التي ستبدأ منها، والمصفوفة الوجهة، والطول المراد نسخه.
- **دوال الـ Class Array المصفوفة:** يوفر `Array Class` مجموعة `Methods` للعمل مع المصفوفات في لغة `C#` العديد من هذه الدوال تعمل مع المصفوفات متعددة الأبعاد (المزيد في مقال عن الـ [Class Array](#)). على سبيل المثال، يمكننا استخدام `Array.Clear` لتعيين كافة العناصر الموجودة في المصفوفة إلى قيمة معينة، أو `Array.Sort` لفرز العناصر الموجودة في المصفوفة.

تعد المصفوفات متعددة الأبعاد (Multi-Dimensional Arrays) في لغة `C#` أداة قوية للتعامل مع البيانات المنظمة ذات الأبعاد المتعددة. إنها تسمح لنا بتنظيم البيانات ومعالجتها بطريقة تشبه الشبكة، مما يجعلها مناسبة لمجموعة واسعة من التطبيقات.

خصائص المصفوفات متعددة الأبعاد في لغة البرمجة C#

لفهم اعمق للمصفوفات متعددة الأبعاد `Multi-Dimensional Arrays` في لغة سي شارب، علينا معرفة خصائص المصفوفات متعددة الأبعاد. فيما يلي بعض الخصائص المصفوفات متعددة الأبعاد:

- **المصفوفات ذات البُعدين العالية:** يمكن تعريف المصفوفات ذات البُعدين العالية في `C#`، وهي تحتوي على أكثر من بُعدين. على سبيل المثال، يُمكن تعريف مصفوفة ثنائية البُعدين واحدة داخل مصفوفة ثنائية البُعدين أخرى، مما يُنشئ مصفوفة ذات بُعدين عالي. هذا يسمح بتنظيم البيانات بشكل أكثر تعقيداً وترتيبها بطرق متعددة.
- **تعديل المصفوفات:** يُمكن تعديل قيم عناصر المصفوفات متعددة الأبعاد في أي وقت. يمكنك تعيين قيم جديدة لعناصر المصفوفة باستخدام المؤشرات المناسبة. هذا يعني أنه يُمكنك تحديث البيانات وتغييرها في المصفوفة بسهولة.
- **العمليات على المصفوفات:** يمكنك تنفيذ العديد من العمليات على المصفوفات متعددة الأبعاد في `C#`. على سبيل المثال، يُمكننا إجراء عمليات رياضية مثل الجمع والطرح والضرب على المصفوفات. يمكنك أيضاً تنفيذ عمليات مثل البحث والفرز وتحويل البيانات على المصفوفات.
- **مصفوفات متعددة الأبعاد غير متجانسة:** في `C#`، يُمكن أن تكون المصفوفات متعددة الأبعاد غير متجانسة، أي أنها يمكن أن تحتوي على أبعاد غير متساوية. على سبيل المثال، يُمكن تعريف مصفوفة ثنائية الأبعاد تحتوي على صفوف بأحجام مختلفة. ومع ذلك، يجب أن يكون عدد الأعمدة متساوياً في جميع الصفوف.
- **المصفوفات كمعاملات وقيم الإرجاع:** يُمكننا تمرير المصفوفات كمعاملات للدوال والأساليب في `C#`. يمكن للدوال أيضاً أن تعيد مصفوفات كقيم الإرجاع، مما يتيح إمكانية استخدام وتعديل المصفوفات في سياقات مختلفة.

يوفر استخدام المصفوفات متعددة الأبعاد في لغة `C#` مرونة وقدرات واسعة النطاق لتنظيم البيانات ومعالجتها. سواء كنت تعمل على تطبيقات رسومية، أو حسابات علمية، أو أي مهام أخرى كثيفة البيانات، يمكن أن تكون المصفوفات متعددة الأبعاد أداة قوية في ترسانة البرمجة الخاصة بك.

المصفوفات ثنائية الأبعاد D2 في سي شارب

إن المصفوفات متعددة الأبعاد في سي شارب هي المصفوفات التي تظهر قيام عناصرها بترتيب غير خطي أي لا يكون ببعد واحد، وأشهر هذه المصفوفات وأكثرها استخداماً هي المصفوفات ذات البعدين Two-Dimensional Arrays - 2D وهذا النوع يمثل قيم المصفوفة بشكل مربع أو مستطيل فهي تُمثل بجدول من صفوف وأعمدة و للإشارة إلى عنصر معين فيها يتم تحديد موقعه من خلال رقم الصف والعمود الذي يوجد به ويوضح الشكل التالي طريقة تمثيل عناصر مصفوفة ثنائية الأبعاد.

وتكون الصيغة العامة لتعريف مصفوفة متعددة الأبعاد من بعدين D Array2 في سي شارب تكون كالتالي :

```
ArrayType[,] ArrayName = new ArrayType[ Number of rows, Number of colums];
```

وعند الإعلان عن المصفوفة بهذه الطريقة يتم تعيين القيم الافتراضية لعناصر المصفوفة بحسب نوع هذه العناصر كما شرحنا سابقاً في مقال عن المصفوفات في C# ، يمكن أيضاً تعيين قيم العناصر عند الإعلان عن D Array 2 في سي شارب مثلاً:

```
int[ , ] arr = {{1 , -2 , 3} , {4 , 5 , 6} , {17 , 8 , 9} , {4 , 0 , 6}};
```

ويتم المرور أو الوصول إلى عناصر المصفوفة ثنائية الأبعاد باستخدام الحلقات المتداخلة Nested loop كما يتضح لنا في المثال التالي :

```
int[ , ] arr = {{1 , -2 , 3} , {4 , 5 , 6} , {17 , 8 , 9} , {4 , 0 , 6}};
```

```
for (int i = 0; i < 4; i++)
```

```
{
```

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

```
    {
```

```
        Console.WriteLine($"{arr[i,j]}");
```

```
    }
```

```
Console.WriteLine();
```

```
}
```

ملاحظة : حاصل ضرب أبعاد المصفوفة ثنائية الأبعاد يساوي عدد عناصر المصفوفة.

المصفوفة ثلاثية الأبعاد D3 في سي شارب

يمكنك إنشاء مصفوفة ثلاثية الأبعاد D3 في لغة البرمجة C#. يتم تعريف المصفوفة ثلاثية الأبعاد بنفس الطريقة التي يتم بها تعريف المصفوفات ذات البُعدين، ولكنها تحتوي على طبقة إضافية بإضافة بُعد ثالث.

فيما يلي مثال يوضح كيفية إنشاء واستخدام مصفوفة ثلاثية الأبعاد في C# :

```
int[ , , ] threeDimensionalArray = new int[3, 4, 2];
```

وطريقة تعيين قيم لعناصر مصفوفة ثلاثية الأبعاد تكون كالتالي:

```
threeDimensionalArray[0, 0, 0] = 1;
```

```
threeDimensionalArray[1, 2, 1] = 5;  
threeDimensionalArray[2, 3, 0] = 10;
```

أما الوصول إلى قيم عناصر المصفوفة تكون

```
int value = threeDimensionalArray[0, 0, 0];
```

```
int anotherValue = threeDimensionalArray[1, 2, 1];
```

```
int yetAnotherValue = threeDimensionalArray[2, 3, 0];
```

ولطباعة أبعاد المصفوفة ثلاثية الأبعاد :

```
int length1 = threeDimensionalArray.GetLength(0);
```

```
int length2 = threeDimensionalArray.GetLength(1);
```

`int length3 = threeDimensionalArray.GetLength(2);`

في هذا المثال، تم إنشاء مصفوفة ثلاثية الأبعاد باسم `threeDimensionalArray` وتحديد أبعادها بواقع $2 \times 4 \times 3$. ثم تم تعيين قيم لبعض عناصر المصفوفة والوصول إلى قيم العناصر المحددة. يمكن استخدام `(GetLength)` للحصول على أبعاد المصفوفة.

يمكنك استخدام المصفوفات ثلاثية الأبعاد في `C#` لتمثيل البيانات ثلاثية الأبعاد مثل الصور ثلاثية الأبعاد أو البيانات المكعبية في التحليل العلمي والعديد من السياقات الأخرى التي تتطلب تنظيم البيانات في ثلاثة أبعاد.

استخدام المصفوفات متعددة الأبعاد Multi-dimensional Arrays

تستخدم المصفوفات متعددة الأبعاد في لغة البرمجة `C#` في العديد من التطبيقات والمجالات. فيما يلي بعض الاستخدامات الشائعة للمصفوفات متعددة الأبعاد في `C#`:

- **الرسومات والصور:** يمكن استخدام المصفوفات متعددة الأبعاد لتمثيل الصور والرسومات. يمكن تخزين قيم البكسل لكل نقطة في الصورة في مصفوفة ثنائية الأبعاد، حيث يتم تعيين قيم لون البكسل لكل موقع في المصفوفة.
- **التحليل العلمي والرياضي:** تُستخدم المصفوفات متعددة الأبعاد في العديد من التطبيقات العلمية والرياضية. على سبيل المثال، يمكن استخدامها في تمثيل البيانات الجغرافية مثل الخرائط، أو في تنفيذ عمليات الحساب العلمي مثل المصفوفات الخطية والجبر الخطي.
- **الألعاب والرسوم المتحركة:** في تطبيقات الألعاب والرسوم المتحركة، تُستخدم المصفوفات متعددة الأبعاد لتخزين وإدارة البيانات المرتبطة بالأشكال والمستويات والشخصيات في اللعبة. يمكن استخدام المصفوفات لتمثيل الخرائط والمستويات ثلاثية الأبعاد وتخزين معلومات الارتفاع والألوان والتفاصيل الأخرى.
- **معالجة الصوت والفيديو:** يمكن استخدام المصفوفات متعددة الأبعاد لتمثيل ومعالجة البيانات الصوتية والفيديو. يمكن تخزين أشرطة الصوت والفيديو في مصفوفة ثلاثية الأبعاد، حيث يتم تعيين قيم العينات الصوتية أو الإطارات الفيديو لكل موقع في المصفوفة.
- **العمليات المصفوفية وتحليل البيانات:** تُستخدم المصفوفات متعددة الأبعاد في العديد من العمليات المصفوفية مثل الجمع والطرح والضرب والقسمة. يمكن استخدام المصفوفات لتنفيذ تحليل البيانات والإحصاءات، مثل حساب المتوسط والانحراف المعياري وتصنيف البيانات.

• **المحاكاة والنمذجة العلمية:** تُستخدم المصفوفات متعددة الأبعاد في النمذجة العلمية والحسابات العلمية لتخزين البيانات المتعلقة بالمحاكاة والنمذجة. يمكن تمثيل البعد الزمني والبعد الثلاثي وغيرها من المعلومات ذات الأبعاد المتعددة باستخدام المصفوفات.

و قد تكون هناك استخدامات أخرى للمصفوفات متعددة الأبعاد في **C#** بحسب احتياجات البرنامج أو التطبيق المحدد. إن استخدام المصفوفات متعددة الأبعاد في **C#** يساعد في تنظيم البيانات وتسهيل عمليات الوصول إليها ومعالجتها. فهي تعتبر أداة قوية ومرنة للعديد من التطبيقات والمجالات في برمجة **C#**.