

Leren Programmeren

MEERDIMENSIONALE ARRAYS

Multidimensionale Arrays

Lernen Programmieren in C#

Inhoud – Multidimensionale Arrays

1. Matrices (tabellen) en Multidimensionele Arrays

- Declaratie
- Gebruik



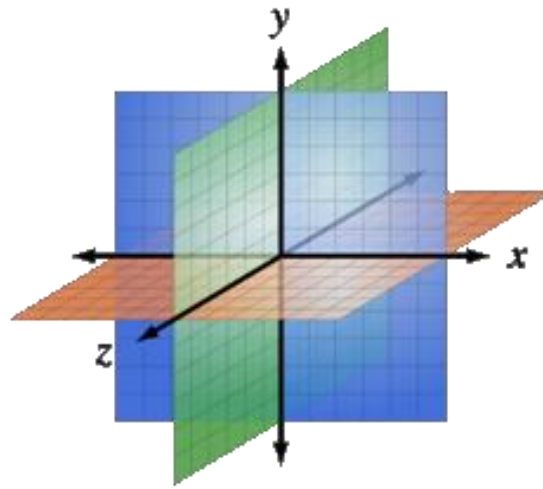
2. Jagged Arrays

- Declaratie
- Gebruik



Multidimensionele Arrays

Gebruik van Array van Arrays, Matrix en Kubus



Wat is een Multidimensionele Array?

- **Multidimensionele arrays** hebben meer dan 1 dimensie (2, 3, ...)
 - De belangrijkste multidimensionele arrays zijn de 2-dimensionele
 - Bekend als **matrices** of **tabellen**
- Voorbeeld van matrix (tabel) van integers met 2 rijen en 4 kolommen:

	0	1	2	3
0	5	0	-2	4
1	5	6	7	8

Multidimensionale Arrays

- C# ondersteunt 2 variaties van multidimensionale arrays
 - *Tabel of Matrix*:
array van meerdere dimensies met gelijke lengten
 - *Jagged array* (array van arrays met verschillende lengten)

Declaratie en creatie van Multidimensionele Arrays

- Declaratie van multidimensionele arrays:

```
int[,] intMatrix;  
float[,] floatMatrix;  
string[, ,] strCube;
```

- Creatie van een multidimensionele array
 - via **new** keyword
 - Specificeer de grootte van elke dimensie

```
int[,] intMatrix = new int[3, 4];  
float[,] floatMatrix = new float[8, 2];  
string[, ,] stringCube = new string[5, 5, 5];
```

Initialisatie van Multidimensional Arrays met waarden

- Creatie en initialisatie van multidimensionale array met waarden:

```
int[,] matrix =  
{  
    {1, 2, 3, 4}, // waarden van rij 0  
    {5, 6, 7, 8}, // waarden van rij 1  
}; // The tabel grootte is 2 x 4 (2 rijen, 4 kolommen)
```

- Tabellen worden in het geheugen bijgehouden als worden beschouwd als een lijst van rijen

De rijen bestaan uit een lijst van waarden

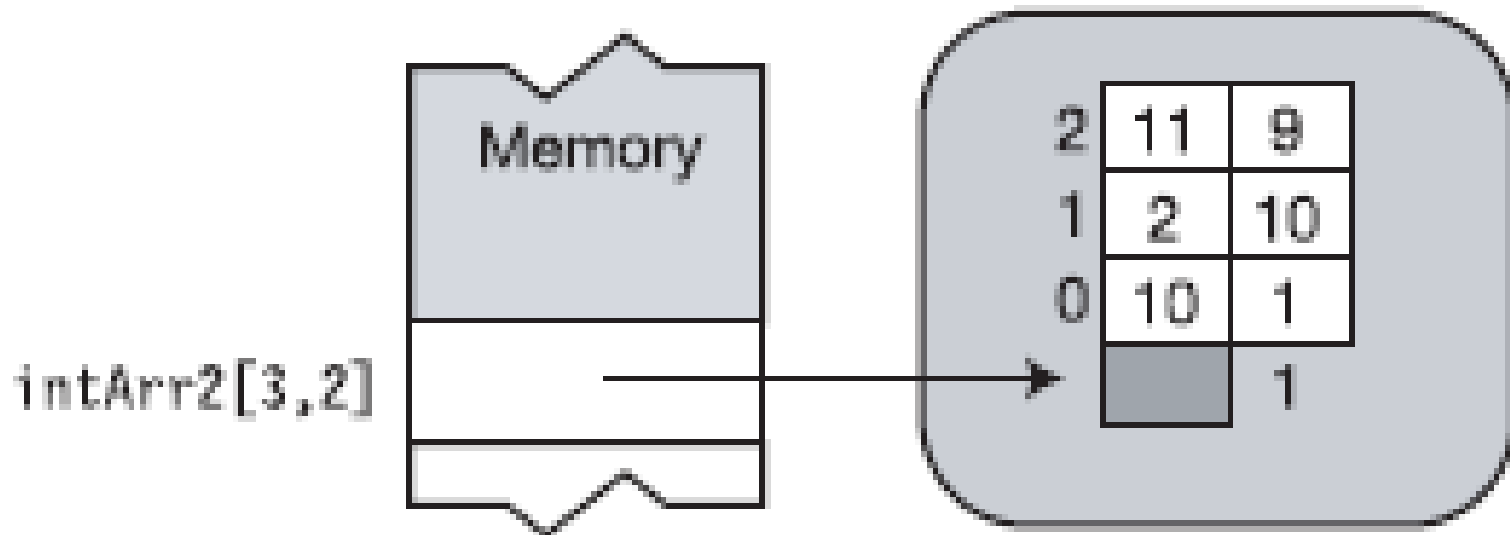
- Eerst de 1ste dimensie, de 2de daarna (binnen in de 1ste)

Geheugen allocatie bij tabel (matrix)

- Initialisatie van *Rectangular* Arrays en toekenning waarden

Voorbeeld:

```
int[, ] intArray2 = new int[, ] { {10,1}, {2,10}, {11,9} };
```



Multidimensionele Arrays

- Toegang tot N-dimensionele array element:

```
nDimensionalArray[index1, ... , indexn]
```

- Getting element value example:

```
int[,] array = { { 1, 2 }, { 3, 4 } };  
int element11 = array[1, 1]; // element11 = 4  
Console.WriteLine(element11);
```

- Voorbeeld invullen waarden van elementen

```
int[,] array = new int[3, 4];  
for (int row = 0; row < array.GetLength(0); row++)  
{  
    for (int col = 0; col < array.GetLength(1); col++)  
    {  
        array[row, col] = row + col;  
    }  
}
```

Aantal
rijen

Aantal
kolommen

Voorbeeld inlezen van Matrix

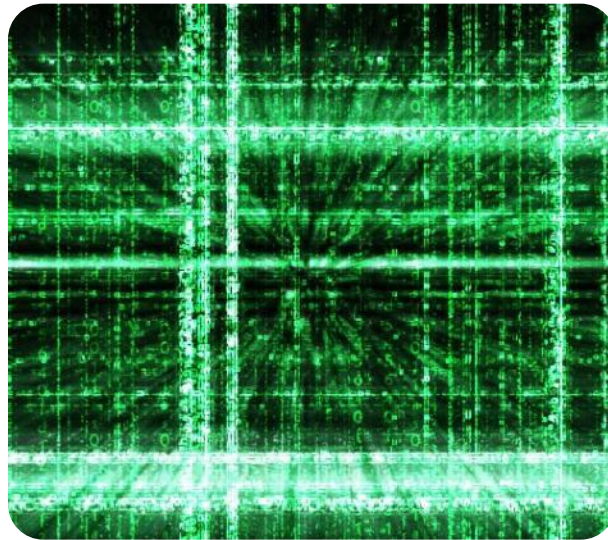
- Inlezen van matrix van de console

```
int rows = int.Parse(Console.ReadLine());
int columns = int.Parse(Console.ReadLine());
int[,] matrix = new int[rows, columns];
String inputNumber;
for (int row = 0; row < rows; row++)
{
    for (int column = 0; column < columns; column++)
    {
        Console.Write("matrix[{0},{1}] = ", row, column);
        inputNumber = Console.ReadLine();
        matrix[row, column] = int.Parse(inputNumber);
    }
}
```

Voorbeeld - Printen van Matrix

- Printen van matrix naar de console:

```
for (int row = 0; row < matrix.GetLength(0); row++)  
{  
    for (int col = 0; col < matrix.GetLength(1); col++)  
    {  
        Console.Write("{0} ", matrix[row, col]);  
    }  
    Console.WriteLine();  
}
```



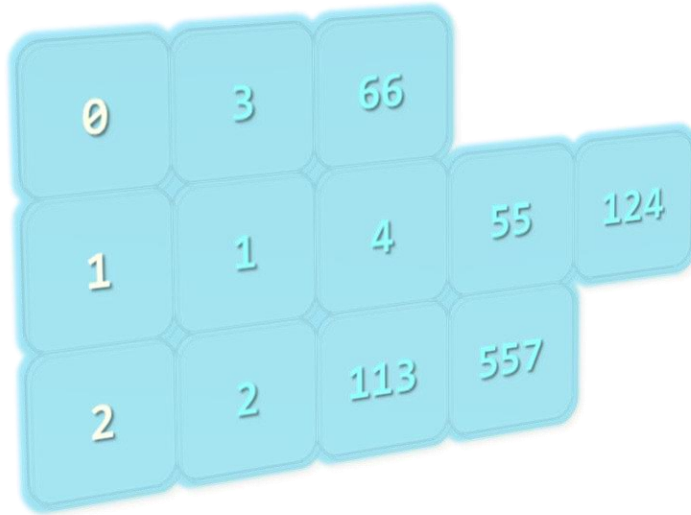
Inlezen en printen van Matrices (tabellen)

Live Demo



2-Dim Array

Live Demo



0	3	66		
1	1	4	55	124
2	2	113	557	



Jagged Arrays

Wat zijn Jagged Arrays en hoe te gebruiken?

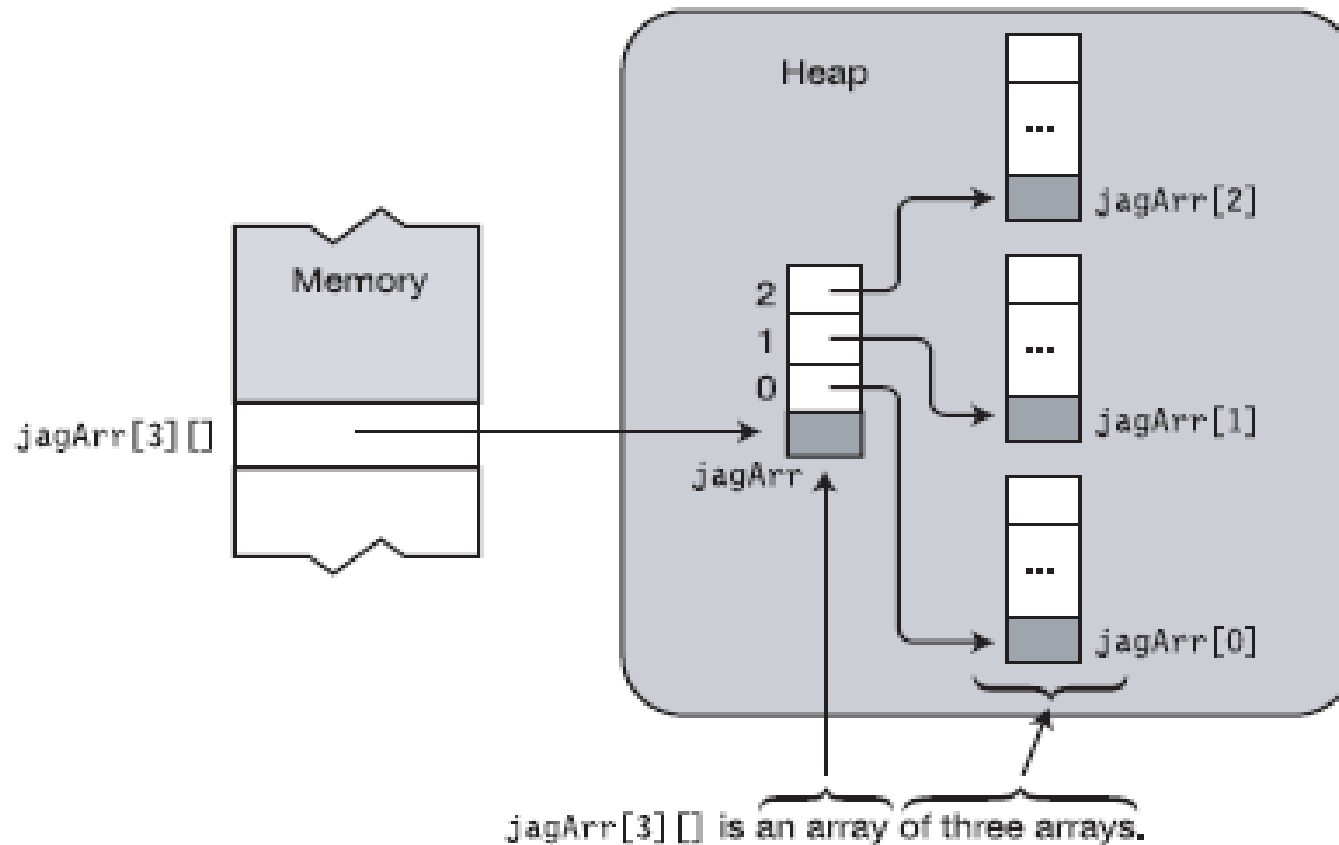
Jagged Arrays

- Jagged arrays zijn zoals multidimensionele arrays
 - Maar: elke dimensie heeft een verschillende grootte
 - Een jagged array is een array of arrays
 - Elk van de arrays kan een verschillende lengte hebben
- Hoe een jagged array creëren?

	0	1	2	3
0				
1				
2				
3				

```
int[][] jagged = new int[3][];  
jagged[0] = new int[3];  
jagged[1] = new int[2];  
jagged[2] = new int[5];
```

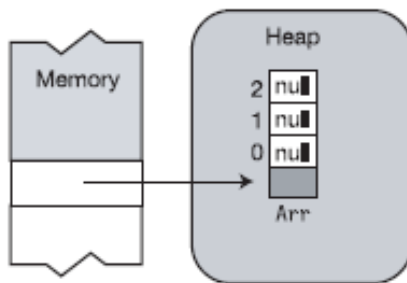

Geheugen allocatie bij 2dimensionale array



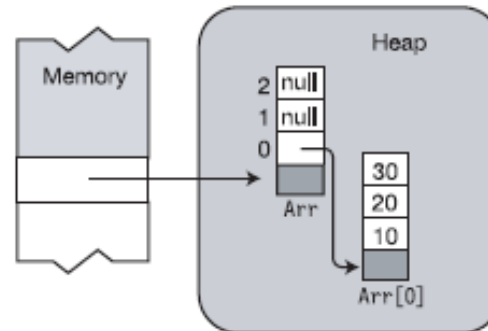
Jagged Array Geheugen Allocatie

```
int[][] arr = new int[3][];  
arr[0] = new int[] { 10, 20, 30 };  
arr[1] = new int[] { 40, 50, 60, 70 };  
arr[3] = new int[] { 80, 90, 100, 110, 120 };
```

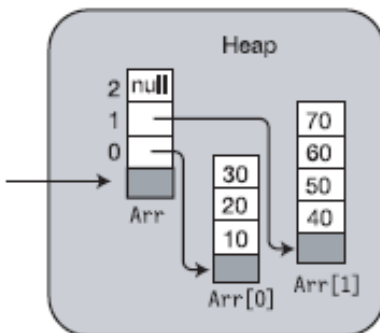
1.



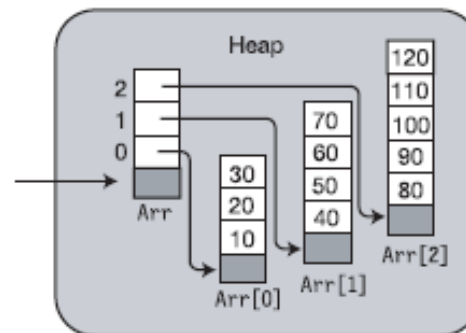
2.



3.



4.



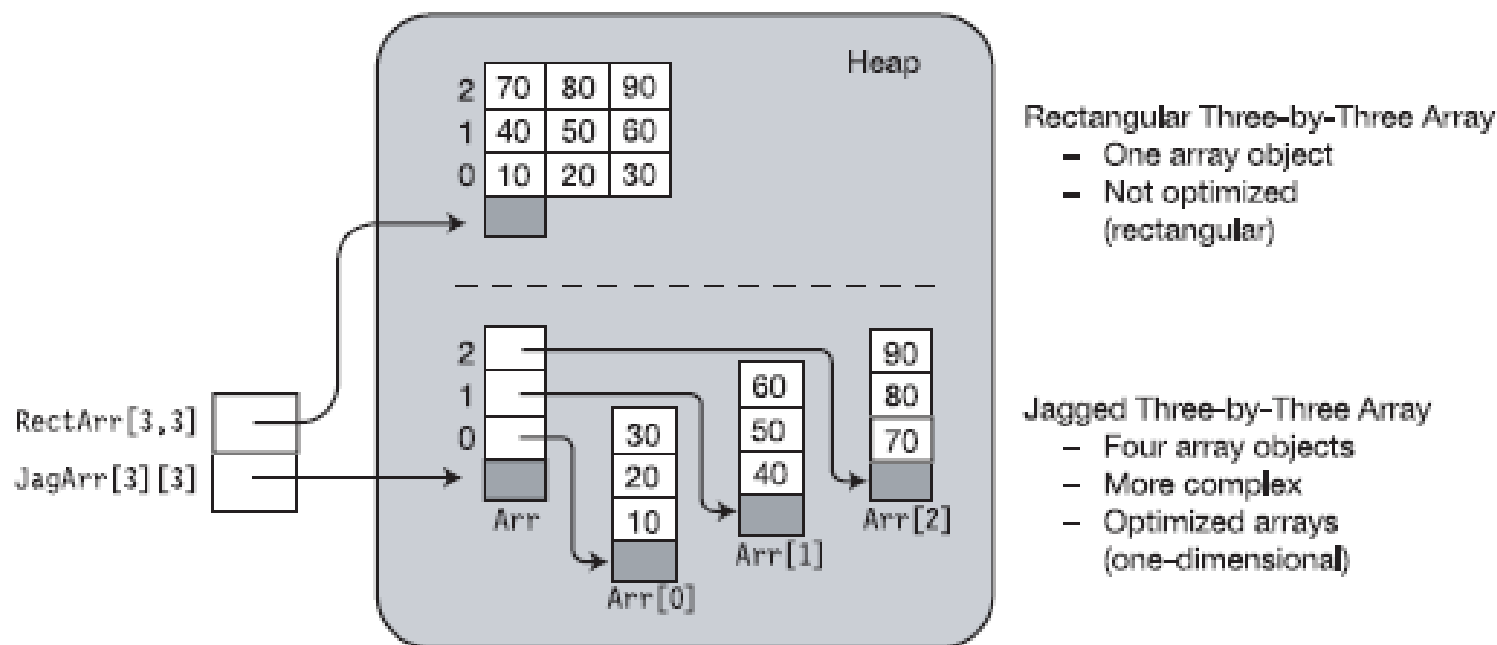
Initialisatie van Jagged Arrays

- Bij creatie van jagged arrays
 - Eerst wordt een array gecreëerd van **null** arrays
 - Elk van de null-Arrays worden daarna geïnitieerd

```
int[][] jagged = new int[n][];  
for (int i = 0; i < jagged.Length; i++)  
{  
    jagged[i] = new int[i+1];  
}
```

Jagged versus tabel(matrix)

- Matrix is steeds vierkantig, bezit geen arrays
- Jagged array bezit verschillende arrays die van verschillende lengte kunnen zijn





Jagged Arrays

Live Demo

Vragen?



Referenties

<http://learncs.org/>

<https://www.educba.com/2d-arrays-in-c-sharp/>

Fundamentals of computer programming with c#

© svetlin nakov & co., 2013