

Mehrdimensionale Arrays

Deklaration:

Datentyp[,] Variablenname

Initialisierung:

Variablenname = new Datentyp[Anzahl Reihen, Anzahl Spalten]

Deklaration + Initialisierung:

Datentyp[,] Variablenname = new Datentyp[Anzahl Reihen, Anzahl Spalten]

-> Bei mehr Dimensionen wird jeweils ein weiteres Komma hinzugefügt:

z.B. `int[, ,] dreidim = new int[3,2,1];`

Um die Gesamtlänge eines mehrdimensionalen Arrays zu erhalten, kann die Eigenschaft `Length` genutzt werden:

`return dreidim.Length;` gibt 6 zurück (3 * 2 * 1)

Um die Länge einer Dimension zu erhalten, kann die Methode `GetLength(int dimension)` verwendet werden:

`return dreidim.GetLength(1);` gibt 2 zurück

Aufgepasst!

Die Dimensionen beginnen wie der Index eines Arrays bei 0

Jagged Arrays

Deklaration:

```
Datentyp[][] Variablenname
```

Initialisierung:

```
Variablenname = new Datentyp[Anzahl Reihen][]
```

oder direkt:

```
Variablenname = new Datentyp[][] {  
    new Datentyp[Länge],  
    new Datentyp[Länge],  
    ...  
}
```

(Der Datentyp muss dabei überall identisch sein)

```
int[][] jagged = new int[][]  
{  
    new int[5], new int[4], new int[3]  
};
```

Wenn das Attribut `Length` genutzt wird, dann bekommt man als Rückgabewert die Anzahl an Unter-elementen innerhalb des Jagged Arrays:

```
jagged.Length => 3
```

Sofern die Länge eines Unter-elementes benötigt wird, dann muss dieses per Index spezifiziert werden.

Darauf kann dann auch das `Length` Attribut genutzt werden:

```
jagged[0].Length => 5
```