

Grafik, Schaubild:

z.B. `new String[]`

Beispiel

Verwendung  
"echter" Objekte

z.B. `String[]`

length: 2

`int[][] a2`

STACK

HEAP

Speicherbild

Zugriff:

`a2[Zeile][Spalte]`

z.B. `new int[2][4]`

es werden nur  
Null Referenzen erzeugt.  
Die ~~Arrays~~ Arrays in  
jeder Zeile müssen extra  
erzeugt werden.

Objekt-  
Arrays

Mehr-  
dimensionale  
Arrays

"nichts" ( $\neq 0$ )

(Vorsicht vor NullPointerExceptions)

`0122`

`[0] null`

`[1] null`

length: 2

STACK

HEAP

Arrays mit variabler Länge

z.B. `new int[2][ ]`

Zugriff:

`a2a[Zeile][Spalte]`

leerlassen

Merksätze / Das Wichtigste in Kürze:

Arrays in Java können auch **mehrdimensional** sein. Das bedeutet, dass in einem Array **mehrere** Arrays mit fester/variabler Länge gespeichert sein können. Um auf ein Element zugreifen zu können, so muss der Speicherort in Form von **Zeile und Spalte** bekannt sein. (verschachtelte for-Schleifen)

**Objekt-Arrays** funktionieren gleich wie normale (eindim.) Arrays. Es ist wichtig zu wissen, dass bei der Erzeugung **Null** Referenzen verwendet werden, es werden also **KEINE** Objekte erzeugt! Diese müssen separat erstellt & initialisiert werden. Im Array selbst werden logischerweise die **Referenzen** gespeichert. Wird eine solche gelöscht, so hat das zugehörige Objekt keine Referenz ( $\rightarrow$  Zugriff) mehr und wird, soweit keine Referenz außerhalb des Arrays existiert, vom **Garbage-Collector** gelöscht.

Quellen, weiterführende Links: