



INITIALISIERUNGSBLÖCKE

Initialisierungsblöcke

- Initialisierungsblöcke stellen eine weitere Methode dar, um Code während der Erzeugung eines Objekts oder beim ersten Benutzen einer Klasse auszuführen.
- Neben den Konstruktoren, welche Explizit aufgerufen werden müssen, werden die Initialisierungsblöcke implizit ausgeführt.

„By convention, init blocks usually appear near the top of the class file, somewhere around the constructors. However, this is the OCA exam we're talking about. Don't be surprised if you find an init block tucked in between a couple of methods, looking for all the world like a compiler error waiting to happen! “

Initialisierungsblöcke

Statische Initialisierungsblöcke

- Klassengebunden – static
- werden nur einmal aufgerufen und ausgeführt.
- werden beim ersten nutzen einer Klasse ausgeführt.
- haben zugriff auf ausschließlich Statische Attribute und Methoden

Nicht - Statische Initialisierungsblöcke

- Instanz gebunden – ohne static
- werden mehrmals verwendet
- werden jedes Mal, wenn eine Instanz erzeugt wird aufgerufen
- haben zugriff auf Statische wie nicht Statische Attribute und Methode

Initialisierungsblöcke

- Regeln zu den Init-blöcken.
 - *init-Blöcke werden in der Reihenfolge abgearbeitet, wie sie erstellt wurden.*
 - *Static init-blöcke, werden nur einmal ausgeführt, wenn die Klasse das erste mal geladen wird.*
 - *Instanz init-Blöcke werden jedes Mal ausgeführt, wenn eine Instanz erzeugt wird*
 - *Instanz init-Blöcke werden nach dem der Konstruktor den „super()*“ Aufruf durchgeführt hat, abgearbeitet.*
- Daraus abgeleitete Reihenfolge der Abarbeitung:
Static Init-Blöcke -> „super()“ Vom Konstruktor -> Instanz init-Blöcke -> restlicher Quellcode vom Konstruktor.*

** super() ist der Aufruf einen Konstruktor der Elternklasse. Wenn wir nicht explizit einen anderen Aufruf angeben, steht super() implizit immer da. Auch wenn wir Garnichts angeben.*