

Java - Garbage Collection (GC)

WIE Objekte UNERREICHBAR WERDEN

1. Scope der Variable endet

Beispiel: lokale Variable einer Methode ist nach beendeter Ausführung der Methode nicht mehr erreichbar

2. Neuweisung der Referenzvariable des Objekt

Beispiel: `Cat myCat = new Cat("Tom");` // Original-Objekt
`myCat = new Cat("Garfield");` // Original-Objekt wird unerreichbar

3. Referenz wird auf null gesetzt

Beispiel: `myCat = null;`

↳ weist den Compiler darauf hin, dass das Objekt nicht länger benötigt wird.

Java - SPEICHERSTRUKTUR

	Heap - Speicher		
	<u>Young-Generation</u>	<u>Old Generation</u>	<u>Metaspace</u>
speichert	neu erstellte Objekte	langlebige Objekte, die mehrere GC- Zyklen überlebt haben	Klassen-Metadaten & Methoden-Informationen
Häufigkeit der GC	regelmäßig	seltener	—* _n

*_n) ab bestimmter Speicher-
grenze wird GC ausge-
löst

⇒ Unterteilung ermöglicht effizientes Speichermanagement

auf Basis der Nutzungsintensität eines Objekts

(da man vermeiden, fast sicher verwendete Objekte regelmäßig zu überprüfen → Zeitergebnis)

ABLAUF der GC: „Mark-and-Sweep“-Prozess

1. Mark Phase

- GC durchläuft alle Objektreferenzen im Code & **markiert** alle Objekte, die noch in Gebrauch sein
- Startpunkt: Wurzelobjekte (Root Objects)
z.B.: globale Variablen, aktive Threadstacks, statische Variablen
- Vorgehensmethode: Traversierung des Objektgraphen
→ **Markierung** aller erreichbaren Objekte als „**lebend**“

2. Sweep Phase

- alle nicht markierten Objekte werden aus dem Speicher **entfernt**
 - Speicherbereich wird bereinigt und für neue Nutzung freigegeben
- ⇒ Lösung nicht mehr referenzierter Objekte & Gewinnung Speicherplatz

↻ Prozess wiederholt sich iterativ

Traversierung

Durchquerung bzw. „systematisches Durchgehen“ von Datenstrukturen

Beispiel: Detektiv durchsucht einen Verwandtschaftsbaum, um alle Verbindungen zu finden

ZWECK der Garbage Collection ist die automatisierte Speicher-
verwaltung, um Speicherlecks zu vermeiden und Entwick-
ler von Aufgaben der manuellen Speicherverwaltung zu entlasten.