

Dynamische Speicherverwaltung

python: $a = \text{list}(i^2 \text{ for } i \text{ in range}(10^6))$
 $a = 5$ / $a = \text{None}$

Java: $a = \text{new ...}$
 $a = \text{null};$

Haskell: ?

Speicher wird nicht mehr gebraucht

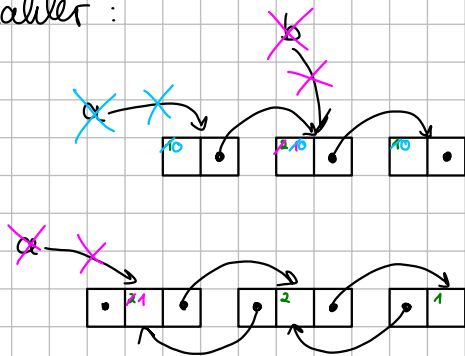
Die Verwaltung des Speichers wird durch die Laufzeitumgebung der Programmiersprache (in Zusammenarbeit mit Betriebssystem) durchgeführt.

- Aufgaben:
- Bereitstellen von zusammenhängenden Speicherbereichen für Programm
 - Wiederverwendung von nicht benötigtem Speicher (Freigabe)

In python, Java, Haskell passiert das automatisch

In z.B. C, C++ muss der Programmierende den Speicher explizit freigeben

1. Referenzzähler:

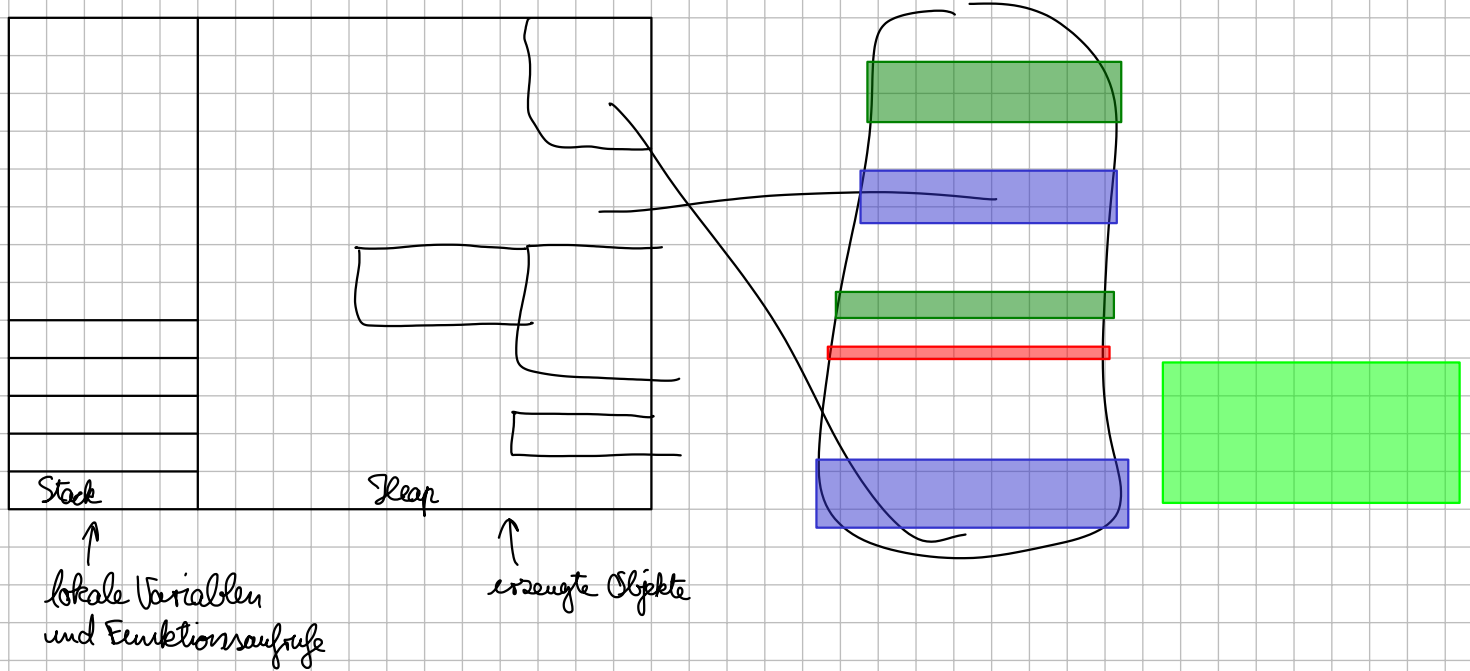


funktioniert bei zyklischen Datenstrukturen nicht

2. Garbage Collection:

Ausgehend von allen statischen und aktuell lokalen Variablen werden alle erreichbaren Objekte markiert (z.B. durch DFS/BFS)

Systeme mit automatischer Speicherverwaltung sind i.d.R. für Echtzeitanwendungen nicht geeignet (mit einigen Ausnahmen, z.B. Java EE)



Ein zu behandelndes Problem ist Speicherfragmentierung

Fazit: Es ist ein schwerer Programmierfehler, unbenötigten Speicher nicht freizugeben

Sonst: Speicherleck
(auch in Java / Python, z.B. durch $a = \text{null} / \text{none}$)