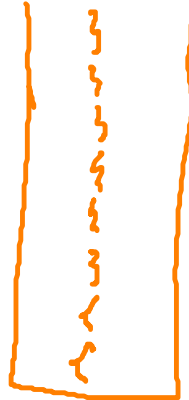


VL 08.01.

heute & morgen: Run-Time Stack, Rekursion

Stack = dyn. DS; Liste, bei der die Datensätze nur am Kopf eingefügt,
gelesen und gelöscht werden

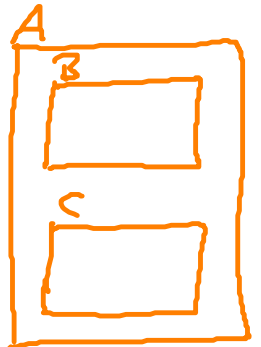
Anwendung: Erkennen von
Klammerausdrücken



Run-Time-Stack

dient der Speicherplatzverwaltung zur
Laufzeit eines Java Programms

Umfasst sogenannte Scopeblöcke, welche zur Laufzeit in
Environment mit Activation Records

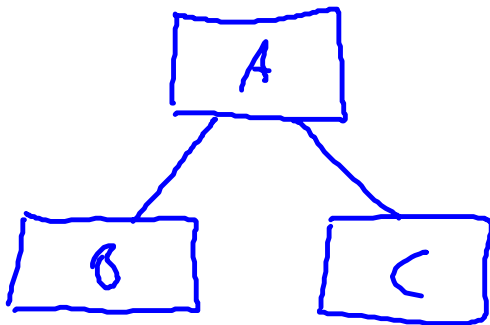


Activation
Record

1. Einträge für lokale Identifizierer (inkl. formaler Parameter und Rückgabewerte bei Methoden)
2. Punkte auf class-Identifizierer bzw. Identifizierer aus übergeordneten Blöcken, die nicht im momentanen Block definiert sind.
3. Adresse der Anweisung im übergeordneten Block, mit dem nach Verlassen des aktuellen Block weitergemacht wird
→ Rücksprungadresse

- Bei Eintritt in Scopeblock wird Record mit entsprechenden Einträgen auf den Run-Time-Stack abgelegt
- Bei Austritt aus Scopeblock wird Activation Record von Stack gelöscht
- Punkte auf Identifizier aus übergeordneten Blöcke werden z.B. dadurch realisiert, dass lediglich ein Pointer auf den übergeordneten Block gesetzt wird und man ggf. über mehrere Punkte auf den Block eines Identifizierers kommt.

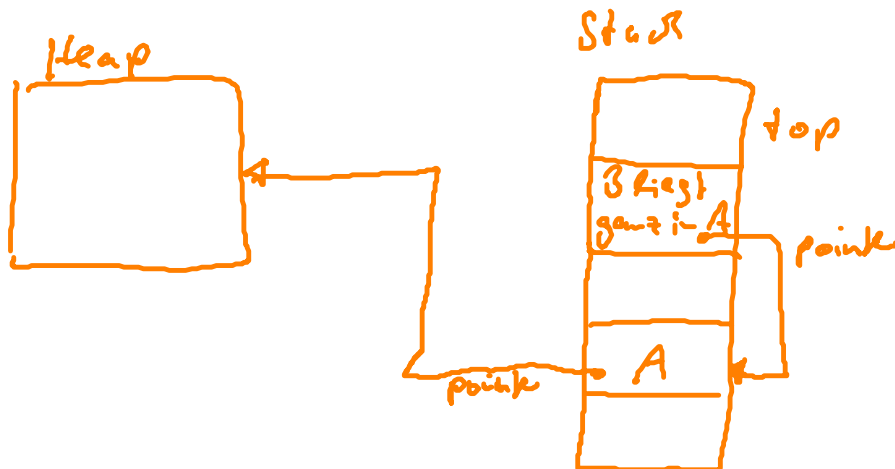
Anrufbaum



Heap

Neben dem Run-Time-Stack gibt es den sog. Heap, in dem die folgende Informationen abgelegt werden:

- alle Identifizier mit Klasse scope
- alle mit new erzeugten Objekte



map für Spielplatz einer Programms / Algorithmus
 ↳ Stack Overflow Error

- max. Anzahl von Activation Records auf dem Run-Time-Stack
- Zshg zum Aufrufbaum: Höhe des Aufrufbaums + 1

Kap 7: Rekursion

Rekursion: Eine Fkt wird durch sich selbst definiert

Bsp: $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ Fakultät

$$n! = \begin{cases} 1 & n=0 \\ n \cdot (n-1)! & n > 0 \end{cases}$$

Bellman Gleichungen

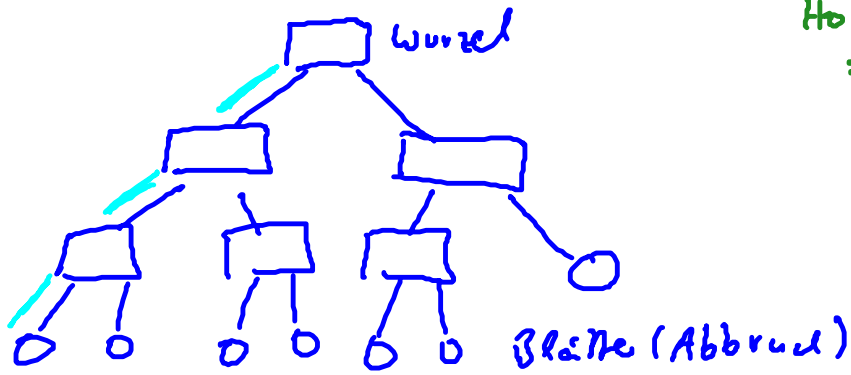
$$u_{ij}^{(n+1)} = \min_k [u_{ik}^{(n)} + a_{kj}]$$

$$ggT(x, y) \geq y$$

$$ggT(x, y) = \begin{cases} y & \text{falls } x \text{ mod } y = 0 \\ ggT(y, x \text{ mod } y) & \text{falls } y > 0 \end{cases}$$

Aufgrund von technischen Problemen mit den E-Kreide wurden die rekursive Implementierung der Methode zum ggT sowie der Aufrufbaum als auch der Run-Time-Stack nur als Bilder gezeigt und erklärt.

i.A. verzweigter Rekursionsbaum



Höhe des Rekursionsbaums
= max # von Kanten auf einem
Weg von der Wurzel zu einem
Blatt

Kenngrößen

Anzahl rekursiver Aufrufe $\hat{=}$ # Knoten im Rekursionsbaum $\rightarrow T(n)$
(Maß für Laufzeit)

max. benötigte Speicher-
platz auf Run-Time-Stack $\hat{=}$ Höhe des
Rekursionsbaum-1 + 1
 $\hat{=}$ Rekursionstiefe $\rightarrow R(n)$