

Lineare Datenstrukturen

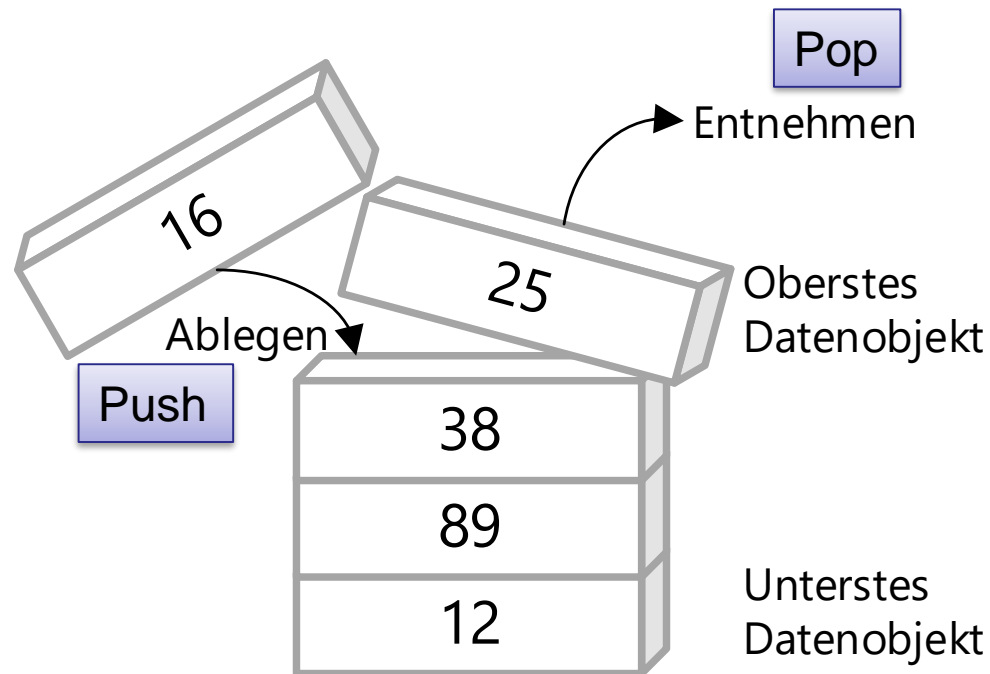
Stack

Lernziele

- Sie wissen, was ein Stack ist
- Sie kennen den Unterschied eines Stack zu einer Liste
- Sie wissen, wie Sie einen Stack implementieren können

Definition Stack

- Deutsch: Stapel, Keller, LIFO-Speicher
- Ablegen und Entnahme der Elemente von oben, d.h. die Elemente, die zuletzt eingefügt wurden, werden als nächstes wieder entnommen (Last-in-First-out).



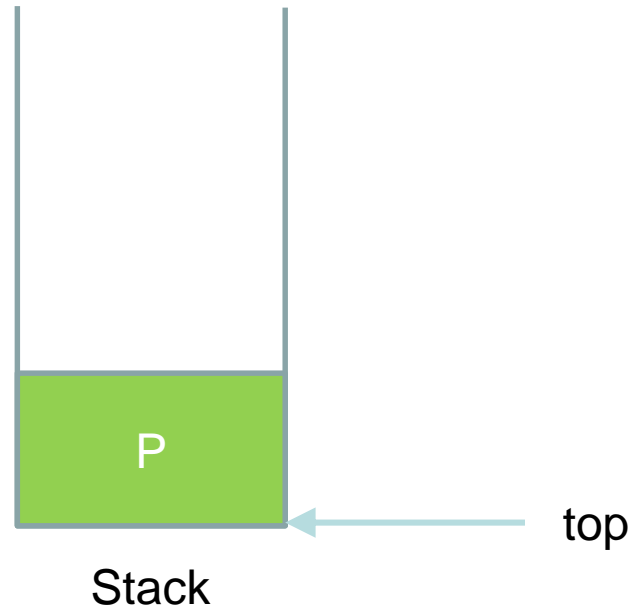
Beispiele



Funktionsweise

- Einfügen: Push
- Löschen: Pop

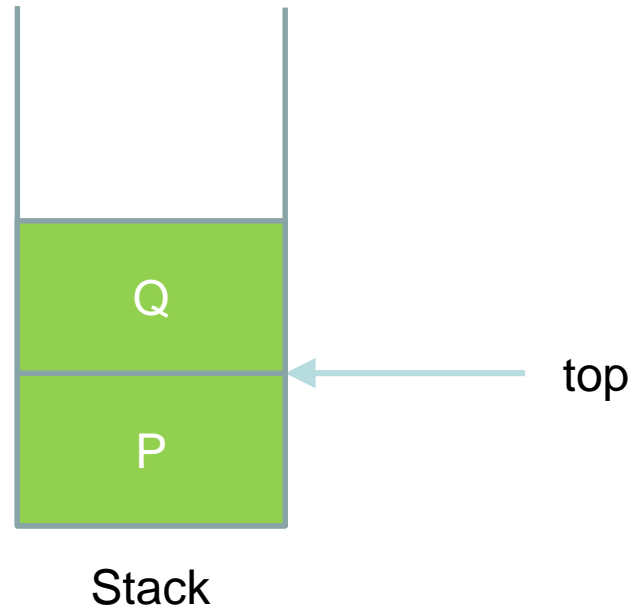
Push P



Funktionsweise

- Einfügen: Push
- Löschen: Pop

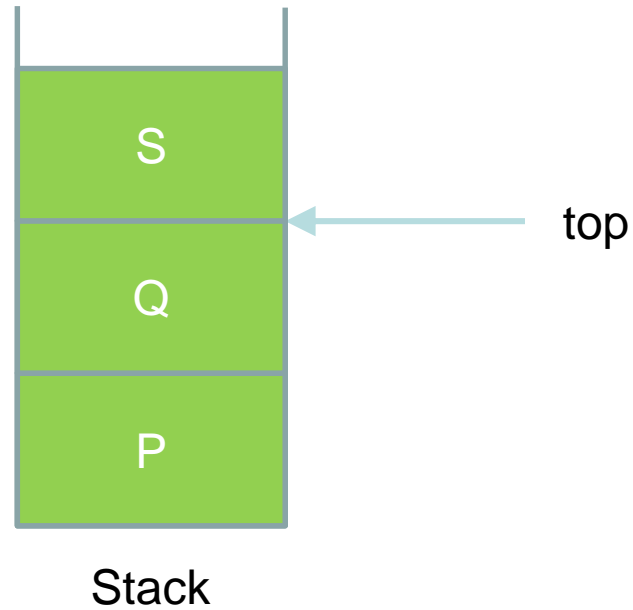
Push Q



Funktionsweise

- Einfügen: Push
- Löschen: Pop

Push S

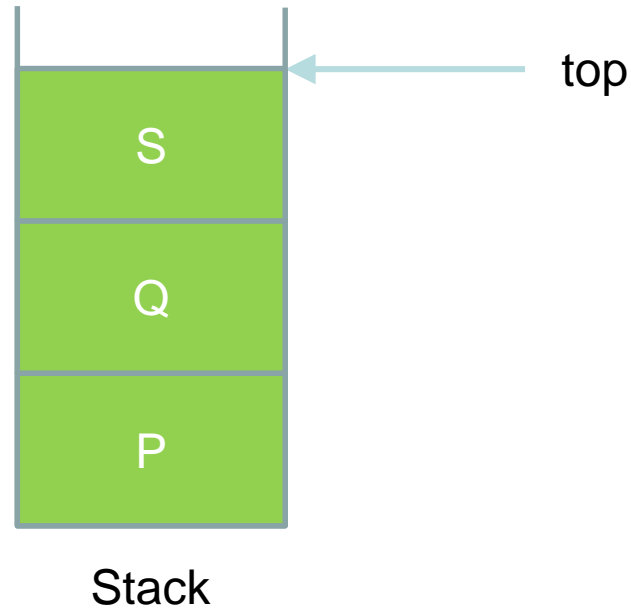


Funktionsweise

- Einfügen: Push
- Löschen: Pop

Pop

Popped Item = S



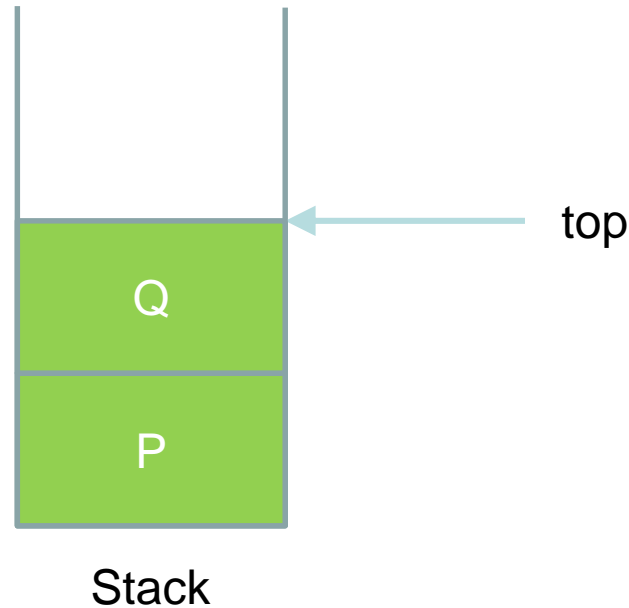
Funktionsweise

- Einfügen: Push
- Löschen: Pop

Pop

Popped Item = S

Popped Item = Q



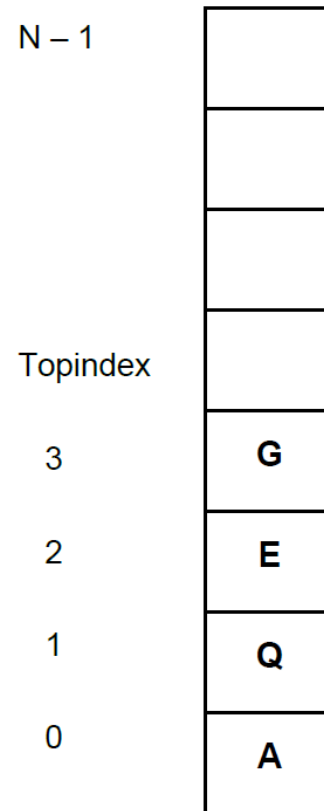
Implementierung

- Welche Methoden und Eigenschaften sollen für einen Stack implementiert werden?

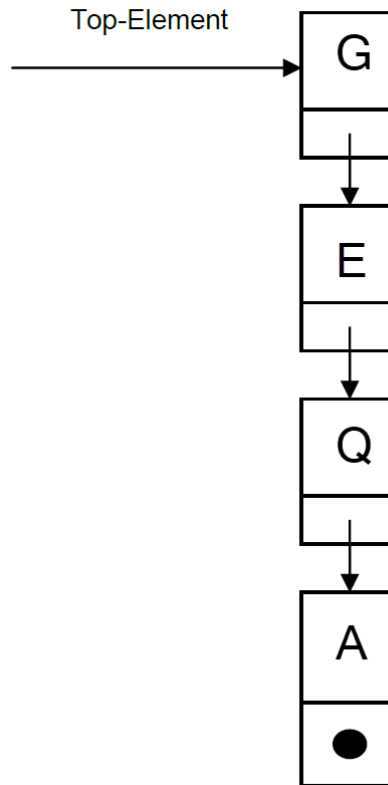
```
public void Push(T item)
public T Pop()
public T Peek()
public void Clear()
public int Count
```

- Implementierung als
 - Array
 - SinglyLinkedList

Implementierung als Array



Implementierung als SinglyLinkedList



Spezialfälle

- Beim Stack müssen üblicherweise zwei Spezialfälle näher betrachtet werden:
 - **Overflow:** Der Stack überläuft – d.h. das Array hat keinen Platz mehr. Entweder wird das Array vergrößert oder der Aufrufer erhält eine Fehlermeldung und die Operation `Push()` wird abgebrochen
 - **Underflow:** Es wird auf einen leeren Stack die Operation `Pop()` ausgeführt. Der Aufrufer erhält eine Fehlermeldung und die Operation `Pop()` wird abgebrochen.

Selbststudium

- Lesen Sie Kapitel 2.3 in [Cordts2018], Lösen Sie die Aufgaben zum Kapitel
- Bearbeiten Sie die Beispiele in [Cordts2018] (Beachten Sie auch die Quellcodes zum Buch – siehe Slides «Einführung»):
 - PostScript (S. 58ff)
 - FloodFill-Algorithmus (S. 63ff)