

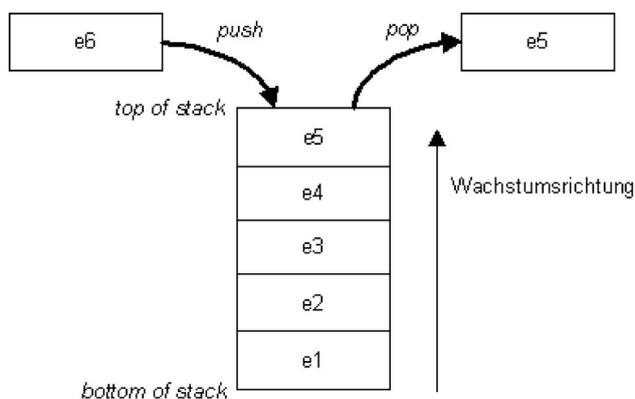
AUFGABENBLATT ZUR ÜBUNG 3

Rückfragen an: Jörn Fischer, j.fischer@hs-mannheim.de

AUFGABE 1

Implementieren Sie eine Klasse **Stack**, ein Stapelspeicher, auf den man mit einer **void push(char c)** Methode Buchstaben ablegen kann und mit der **char pop()** Methode Buchstaben wieder vom Stapel entfernen kann. (<https://de.wikipedia.org/wiki/Stapelspeicher>, die Quellcodes auf der Seite sind wenig hilfreich und können ignoriert werden.)

Ein Stack-Pointer zeigt immer auf die aktuell zu schreibende Position des Stack-Speichers, in diesem Fall ein Array. Der Stack-Pointer ist ein Index dieses Arrays und muss von allen Stack-Funktionen aus zugreifbar sein. Die push() Methode schreibt an diesen Index das übergebene Zeichen und erhöht (inkrementiert) den Stack-Pointer, die pop() Methode dekrementiert den Stack-Pointer wieder und liest dann das Zeichen aus dem Array.



Push

legt ein Objekt auf den Stack

Pop

entfernt das oberste Objekt und gibt es zurück

AUFGABE 2

Schreiben Sie ein Programm, welches einen beliebigen Ausdruck von der Konsole einliest und auf die semantische Korrektheit der Klammern überprüft. Nutzen sie dazu den Stack den sie in Aufgabe 1 implementiert haben: Jedes Zeichen des Ausdrucks wird nacheinander geprüft, ob es eine Klammer auf (, [, { ist. Falls das der Fall ist wird sie mit push auf den Stack gelegt. Wenn eine Klammer eine Klammer zu ist),], } dann wird die nächste Klammer mit pop() vom stack genommen und getestet, ob die Kammer vom Stack zur Klammer vom Ausdruck passt. Passt sie nicht, so gibt ihr Programm ein „Klammern ungültig“ aus.

Beispiele für richtig gesetzte Klammern:

- `()[]()`
- `{while(true){print(arg[5]);}}`
- `[(n+1)*5]+(7-1)`

Beispiele für falsch gesetzte Klammern:

- `(a[b(c]d)e)`
- `(5[i]`

LERNZIELE

- Klasse, Instanz
- Arrays
- Methoden
- Stack