

Aufgabenblatt 3

Algorithmen und Datenstrukturen, SS 2015

Prof. Dr. Ulrich Hedtstück, HTWG Konstanz

Ein *Stack* (Stapel, Kellerspeicher) ist eine Datenstruktur, die zum Speichern von Objekten gleichen Typs verwendet wird, wobei immer nur von einer Seite (anschaulich von oben) Objekte hinzugefügt oder gelöscht werden können (push und pop). Das Zugriffsprinzip wird LIFO (Last In First Out) genannt.

Realisieren Sie eine Klasse **Stack** für Character-Zeichen, die einen Stack mit Hilfe eines Arrays realisiert.

Die Klasse **Stack** enthalte die privaten Komponenten

- `char[] stackArray` Array für den Stack
- `int top` Position des obersten Stack-Elements

Als public-Methoden sind vorzusehen:

- Konstruktor reserviert benötigten Speicher, setzt top auf -1
- `void push(char)` fügt ein Element hinzu
- `char pop()` löscht das oberste Element und liefert es zurück
- `boolean isEmpty()` prüft, ob der Stack leer ist
- `boolean isFull()` prüft, ob der Stack voll ist

Erledigen Sie mit Hilfe Ihres Stacks die folgenden Aufgaben:

(a) Überprüfen Sie, ob in Ihrem Programm zur ersten Übungsaufgabe alle geschweiften, eckigen und runden Klammern korrekt gesetzt sind. (Streichen Sie zur Vermeidung von Komplikationen alle Klammern in Kommentaren.) Lesen Sie dazu Ihr Programm Zeichen für Zeichen aus der entsprechenden Datei und entscheiden Sie jeweils, was mit dem Zeichen gemacht werden soll (überlesen, speichern, löschen).

Geben Sie geeignete Meldungen auf dem Bildschirm aus, falls alles korrekt geklammert ist bzw. falls ein Fehler gefunden wurde.

(b) Verwenden Sie eine kleine Textdatei, in der nur Klammern vorkommen, und testen Sie Ihr Programm anhand von typischen korrekten und inkorrekten Klammerkombinationen.

(c) Ändern Sie Ihr Programm so ab, dass einfache und verschachtelte Kommentare auf Korrektheit überprüft werden können. Ein verschachtelter Kommentar wäre z. B. `/*ababab/*cdcdcdc*/efef/*ghghgh*/`

Das Programm soll in den Übungen vorgeführt und erklärt werden, spätestens am Freitag, den 15.05.2015.