
3. Übung Algorithmen und Datenstrukturen WS 2016/17

Klaus Kriegel

Abgabe: 11.11.2016, 12:00 Uhr

Allgemeine Forderungen zur Bearbeitung von Programmieraufgaben:

- Die abgegebenen Programme sollten auf den Pool-Rechnern der Informatik lauffähig und an hinreichend vielen Beispielen getestet sein.
 - Der Quellcode ist ausreichend zu kommentieren, damit alle Ideen leicht nachvollziehbar sind, und sollte gut lesbar formatiert sein. Er muss zweifach abgeliefert werden: elektronisch im KVV unter Assignments und gedruckt (für die Korrektur) in den Tutorenfächern.
-

Aufgabe 1: String, Char und Character (4 Punkte)

Machen Sie sich in der Java-API mit den String-Klasse und der Wrapper-Klasse Character bekannt. Verwenden Sie die String-Methode `toCharArray()` und die Character-Klassenmethoden `isLetter(char c)`, `isDigit(char c)`, um die Anzahl der Buchstaben und Ziffern in einem in der Kommandozeile übergebenen String (ohne Leerzeichen) zu bestimmen. Bei der Eingabe `Was345kommt-raus?` sollte man die folgende Ausgabe bekommen:

Der String `Was345kommt-raus?` enthaelt 12 Buchstaben und 3 Ziffern.

Implementieren Sie dazu eine Klasse `StringTest`, welche nur eine `main`-Methode mit der beschriebenen Funktionalität enthält.

Aufgabe 2: Stacks I (8 Punkte)

Stacks sind gut dazu geeignet, die Korrektheit von Klammerausdrücken zu überprüfen. Dabei wird der Ausdruck von links nach rechts gelesen. Nur die öffnenden Klammern werden beim Lesen auf den Stapel gelegt. Ist das aktuelle Symbol eine schließende Klammer, muss geprüft werden, ob das oberste Symbol eine öffnende Klammer gleichen Typs ist. Bei positiver Antwort wird diese öffnende Klammer gelöscht und man geht zum nächsten Symbol über, anderenfalls ist der Ausdruck nicht korrekt. Ist kein solch vorzeitiger Abbruch erfolgt, so ist der Klammerausdruck genau dann korrekt, wenn der Stack am Ende leer ist. Wir beschränken uns auf runde und eckige Klammern wie z.B. in dem gültigen Ausdruck `() [() [()]]`.

Verwenden Sie die Klasse `Stack<E>` aus `java.util` zur Implementierung dieses Tests. Beachten Sie, dass man zur Verwendung dieser Datenstruktur Elemente der Wrapper-Klasse als Einträge verwenden muss. Der zu testende Ausdruck soll über die Tastatur eingegeben werden.

Hinweis: Es bietet sich an, `stack<Charakter> st` als Attribut zu halten und eine darauf zugreifende Methode `boolean processChar(char c)` zu implementieren, die `false` zurückgibt, wenn `c` kein Klammersymbol ist oder eine schließende Klammer ohne passendes Gegenstück auf `st` ist, anderenfalls nach Verarbeitung des Symbols `c` auf `st` den Wert `true` zurückgibt.

Aufgabe 3 auf der nächsten Seite!

Aufgabe 3:**Stacks II**

(8 Punkte)

Implementieren Sie die Methode zur Berechnung der Kursspannen aus der Vorlesung (siehe Skript ADTStapelSchlange.pdf). Die Eingabe der Kursfolge sollte dabei über die Tastatur erfolgen, wobei zuerst n als `int`-Wert und nachfolgend die $n + 1$ Kurswerte a_0, a_1, \dots, a_n als `float`-Werte eingegeben werden. Verwenden Sie wieder die Klasse `Stack<E>` aus `java.util` zur Implementierung. Da `E` ein Klassentyp sein muss, müssen Sie eine passende Klasse `RateEntry` implementieren.