

Sommersemester 2010

## Übungen zu Informatik II - Blatt 3

Abgabe in der Übung am 27. 04 / 28. 04. 2010

Bitte beachten Sie, dass die Programmieraufgaben von Ihnen in der Übung vorgeführt und erklärt werden müssen. Zusätzlich senden Sie die Lösung, unter Angabe ihres Namens, an **dm@tu-clausthal.de**.

### Aufgabe 1 (Infix – Postfix, 2+2 Punkte)

- a) Erweitern Sie den Algorithmus zur Konvertierung von Infix nach Postfix auf Folie 44 des Kapitels über „Einfache Datenstrukturen“ so, daß in Ausdrücken die gewöhnliche Punkt-vor-Strich-Präcedenzregel beachtet wird. In den Ausdrücken dürfen vorkommen:  
Buchstaben/Zahlen (= Operanden), Klammern und die 4 Grundrechenarten.  
Ihr Algorithmus soll also den Ausdruck

$$(a + b * c) / (d - e)$$

umwandeln in

$$abc * +de - /$$

- b) Implementieren Sie Ihren Algorithmus in das Framework und testen Sie ihn. Das Framework liest die Eingabe ein und zerlegt diese in ihre Einzelteile. Das Resultat wird dann in der Variablen `inputString` (Liste) abgespeichert und kann von Ihnen weiter verarbeitet werden.

### Aufgabe 2 (Einfach verkettete Liste, 3+4 Punkte)

Wir gehen von folgenden Listenelementen aus:

```
class ListElement(object):  
  
    def __init__( self ):  
        self.item = self.next = None
```

Außerdem sind die drei Variablen `head`, `tail` und `cursor`, wie in der Vorlesung besprochen, gegeben.

Ihre Aufgabe besteht nun darin, für die folgenden Listenoperationen anzugeben, wie die Zeiger umgebogen werden müssen. Für einige Listenoperationen wurde dies bereits in der Vorlesung vor gestellt. Orientieren Sie sich an diesen Beispielen.

- a) `removeCurrent` — löscht aktuelles Element aus der Liste  
b) `insertBeforeCurrent( item )` — fügt Element `item` vor die aktuelle Position ein

Sie brauchen kein komplettes Python-Programm abliefern, sondern nur die relevanten Anweisungen aufschreiben.

### Aufgabe 3 (Doppelt verkettete Liste, 2+2 Punkte)

Gegeben ist eine doppelt verkettete Liste, deren Listenelemente so definiert sind:

```
class ListElement2(object):  
  
    def __init__( self ):  
        self.item = self.next = self.prev = None
```

Gegeben ist außerdem wieder die Variable `cursor`, wie in der Vorlesung besprochen.

Gegeben ist nun ein Element `z`, d.h., dieses wurde erzeugt mit `z = ListElement2()`.

- Geben Sie die Anweisungen an, mit der `z` *nach* dem Listenelement, auf das `cursor` zeigt, in die Liste eingefügt wird. (Denken Sie auch an die Spezialfälle!)
- Wie müssen die Zeiger gesetzt werden, wenn anschließend ein Element *vor* `z` in die Liste eingefügt werden soll.

Auch hier sind wieder nur die relevanten Anweisungen gefragt, kein komplettes Python-Programm.

### Aufgabe 4 (Stack, 2 Punkte)

Ein Buchstabe in der nachfolgenden Sequenz steht für die Stack-Operation `push` und ein  $\diamond$  steht für die Operation `pop` mit Ausgabe des ge-pop-ten Zeichens:

E A S  $\diamond$  Y  $\diamond$  Q U E  $\diamond$   $\diamond$   $\diamond$  S T  $\diamond$   $\diamond$   $\diamond$  I O  $\diamond$  N  $\diamond$   $\diamond$   $\diamond$

Geben Sie die Folge der Buchstaben an, welche durch die `pop`-Methode ausgegeben wird. Gehen Sie davon aus, dass der Stack zu Beginn leer ist.