

## — Übungsblatt 4 (T & P) —

### Aufgabe 8 (T & P)

(Array-Liste)

Überlegen Sie sich eine Vorgehensweise, mit der Sie eine Array-Liste so nutzen können, dass ein neues Element stets am Ende eingefügt wird und beim Löschen jeweils das erste Element im Feld gelöscht wird, ohne dass die nachfolgenden Elemente aufrücken. Dadurch kann der Aufwand für das Löschen auf  $O(1)$  gesenkt und die vorderen frei werdenden Feldelemente können später wiederverwendet werden.

Hinweis: Stellen Sie sich die Array-Liste ringförmig angeordnet vor!

### Aufgabe 9 (T & P)

(Einfach verkettete Liste)

Überlegen Sie sich eine oder mehrere alternative Instanzmethoden

```
ElementL<T> insert (T o, ElementL<T> pos)
```

für die in der Vorlesung behandelte Klasse `ListeL` zur Darstellung einer einfach verketteten Liste, die das Objekt `o` an der von `pos` referenzierten Stelle einfügen können. Es liegt also keine Referenz auf den Vorgänger in der Liste vor. Mit Aufwand  $O(n)$  kann man sich diese jedoch besorgen.

Mit einem Trick lässt sich die Methode aber auch mit Aufwand  $O(1)$  realisieren!

### Aufgabe 10 (T & P)

(Doppelt verkettete Liste)

Vervollständigen Sie die in der Vorlesung nur unvollständig angegebene Klasse `ListeDL`.

Zu schreiben sind die Methoden

- `public ElementDL<T> insert (T o)`  
(Einfügen von `o` am Ende)
- `public ElementDL<T> insert (T o, ElementDL<T> pos)`  
(Einfügen von `o` an der Position `pos`)
- `public void remove (ElementDL<T> e)`  
(Löschen des Elements `e`)
- `public String toString()`  
(Liste als Zeichenkette)

## Aufgabe 11 (T & P)

(Bäume)

Erweitern Sie die in der Vorlesung kennengelernte Klasse Baum:

1. Schreiben Sie eine Methode, die das größte Element im Baum findet (rekursiv und iterativ).
2. Lesen Sie den Baum in Postorder aus.
3. Was macht diese Methode?

```
public int wasMachIch() {  
    int x=0;  
    if (links != null) {  
        x = Math.max(links.wasMachIch(); x);  
    }  
    if (rechts != null) {  
        x = Math.max(rechts.wasMachIch(), x);  
    }  
    x++;  
    return x;  
}
```