

# Objektorientiertes Programmieren in Objective-C unter Mac OS X

## Aufgabe 3 (Zeiger und Felder)

Definieren Sie ein Feld

```
int c[8];
```

und initialisieren Sie dieses mit Integer-Werten. Danach definieren Sie drei Zeiger

- `cp` (Zeiger auf ein aktuelles Feldelement)
- `maxp` (Zeiger auf das größte Feldelement) und
- `minp` (Zeiger auf das kleinste Feldelement)

Ermitteln Sie mit Hilfe der Zeiger den größten und den kleinsten Integer-Wert des Feldes und geben Sie diese auf den Bildschirm aus. Verwenden Sie dafür – außer ggf. Laufvariablen – keine weiteren Variablen, insbesondere keine Indizes.

**Testattermin: 5.5.2004**

## Aufgabe 4 (Doppelt verkettete Listen)

In dieser Aufgabe soll eine Datenstruktur erarbeitet werden, in der doppelt verkettete Listen für allgemeine Datenobjekte verwaltet werden können. Dafür werden wir generische Zeiger verwenden.

- Entwerfen Sie eine Datenstruktur `struct dlist` für doppelt verkettete Listen, die als Datenelement jeweils einen generischen Zeiger enthält.
- Schreiben Sie die folgenden Funktionen, die doppelt verkettete Listen manipulieren:
  - `struct dlist *appfront(struct dlist *list, void* d)`  
Fügt ein neues Element vor `list` in die Liste ein. Falls `*list` den Wert `NULL` hat, so wird eine neue Liste erstellt.
  - `struct dlist *appback(struct dlist *list, void* d)`  
Fügt analog ein neues Element hinter `list` in die Liste ein.
  - `void *delete(struct dlist *list)`  
Entfernt das Element `list` aus der Liste und gibt die darin enthaltenen Daten zurück. Die entsprechende `dlist`-Struktur wird freigegeben.
  - `void swap(struct dlist *elem1, struct dlist *elem2)`  
Vertauscht zwei Elemente der verketteten Liste.

- (c) Schreiben Sie zwei Makros `NEXT` und `PREV`, die jeweils das nächste bzw. das vorhergehende Element der Liste zurückgeben, sowie ein Makro `FIRST`, welches das erste Element einer Liste liefert und ein Makro `LAST`, welches das letzte Element einer Liste liefert
- (d) Fassen Sie die Funktionen in einer Datei `dlist.c` zusammen und schreiben Sie eine Header-Datei `dlist.h`, in der die Datendefinitionen, die Funktionsköpfe für die Listenverwaltung und die Makros enthalten sind.

**Testattermin: 5.5.2004**