

# Übungsblatt 5 zu Programmiersprachenkonzepte

WS 2012/2013

Ausgabe: 7.11.12

Abgabe bis zum Testattermin,  
spätestens am 14.11.2012, 8 Uhr

## Aufgabe 1: (5 Punkte)

Schreiben Sie eine Lisp-Funktion *myreverse*, die die Elemente einer Liste in ihrer Reihenfolge „bis in die Tiefe“ umgekehrt. Sie dürfen dabei die vordefinierte Lisp-Funktion *reverse* NICHT benutzen.

Also:

<code>(myreverse '(a b c d))</code>	$\rightarrow$ <code>(d c b a)</code>
<code>(myreverse '(a (b c) d))</code>	$\rightarrow$ <code>(d (c b) a)</code>
<code>(myreverse '(a (b (c d e) f) g))</code>	$\rightarrow$ <code>(g (f (e d c) b) a)</code>
<code>(myreverse '(a (b c) (d (e f))))</code>	$\rightarrow$ <code>((f e) d) (c b) a)</code>

## Aufgabe 2: (3 Punkte)

1. Schreiben Sie eine format-Anweisung, die als Parameter eine 6-stellige Matrikelnummer, einen Vor- und einen Nachnamen erhält und die folgende Ausgabe erzeugt:

```
982355 Max                Mustermann
```

Ruft man diese format-Funktion nacheinander mit unterschiedlichen Nummern und Namen auf, so sollen die Vor- und Nachnamen bündig untereinander stehen. Vornamen sollen bis zu 15 Zeichen lang sein dürfen, Nachnamen bis zu 20:

```
982355 Max                Mustermann
912345 Sabine-Susanne    Meier-Schulze
```

2. Schreiben Sie mind. zwei verschiedene format-Anweisungen, die die folgende Ausgabe erzeugen:

```
Das Zeichen fuer ~ ist ~~
```

3. Schreiben Sie eine format-Anweisung, die die folgende Ausgabe erzeugt:

```
Laenge:    22.34
Breite:    134.20
Hoehe:     1.0
```

### Aufgabe 3: (12 Punkte)

Gegeben sei die Datei `preise.txt`, in der zeilenweise Preise (float, mit max. 2 Nachkommastellen) stehen. Schreiben Sie eine LISP-Funktion *durchschnitt*, welche als Parameter den Dateinamen der Datei bekommt, die Preise liest, ihren Durchschnitt berechnet und am Ende der Datei dann eine neue Zeile einfügt, die folgendes Aussehen hat:

```
durchschnittlicher Preis: xxx.yy Euro
```

### Aufgabe 4: (6 Punkte)

Was liefert die folgende Funktion? Wie arbeitet sie?

```
1  (defun xxx (list)
2      (do ( (lst (cdr list) (cdr lst))
3          (head (car list) (car lst))
4          (lasthead list head)
5          )
6          ((null lst) NIL)
7          (if (eq head lasthead)
8              (return lst)
9              )
10         )
11 )
```