

Übungsblatt 6 zu Programmiersprachenkonzepte

WS 2012/2013

Ausgabe: 14.11.12

Abgabe bis zum Testattermin,
spätestens am 21.11.2012, 8 Uhr

Aufgabe 1: (4 Punkte)

Schreiben Sie eine Lisp-Funktion *matmulmat*, die zwei beliebige Arrays als Parameter erhält und diese multipliziert. Dazu muss innerhalb der Funktion vorab getestet werden, ob die Dimensionen der Matrizen eine Multiplikation zulassen. Geht dies nicht, soll die Funktion mit einer Fehlermeldung beendet werden, ansonsten gibt sie die Ergebnismatrix aus.

Aufgabe 2: (8 Punkte)

Schreiben Sie ein Funktionspaket in Lisp, welches folgende Funktionen enthält:

1. *quadmatp*, welche eine Matrix als Parameter erhält und T liefert, wenn die Matrix quadratisch ist, sonst NIL.
2. *ematp*, welche eine Matrix als Parameter erhält und T liefert, wenn die Matrix die Einheitsmatrix ist.
3. *transpm*, welche eine Matrix als Parameter erhält und die dazu transponierte Matrix ausgibt.
4. *orthom*, welche eine Matrix als Parameter erhält und T liefert, falls die Matrix orthogonal ist, sonst NIL.

Sie dürfen innerhalb der vier Funktionen alle auf diesem Aufgabenblatt zuvor definierten Funktionen benutzen. Die Funktionen sollen immer dann mit einer Fehlermeldung abbrechen, wenn die Matrix wegen ihrer Dimensionen nicht für die Funktion als Parameter geeignet ist.

Aufgabe 3: (10 Punkte)

- Definieren Sie in impure Lisp eine Struktur, die einen Studenten/eine Studenten modelliert. Diese haben eine 6-stellige Matrikelnummer, einen Vornamen (max. 15 Stellen) einen Nachnamen (max. 20 Stellen), ein Studienfach (max. 20 Stellen) und ein Geschlecht (m/w).

- Schreiben Sie eine Funktion *save*, die einen Dateinamen und einen Studenten/In als Parameter hat, und den Studenten/In in einer neuen Zeile an den alten Dateiinhalt anhängt.
- Schreiben Sie eine Funktion *loadstudis*, die einen Dateinamen als Parameter erhält und eine Liste von Studenten/Innen (mit allen Informationen) liefert. Jeder Student/In sei dabei selbst in einer (Unter-)Liste dargestellt.
- Schreiben Sie eine Funktion *GeschlechtFach*, die als Parameter einen Dateinamen (in der Datei stehen die Studenten/Innen) und eine Fachbezeichnung erhält. Als Ergebnis liefert diese Funktion, wie viele männliche und wie viele weibliche Hörer dieses Fach studieren, zum Beispiel:

(*geschlechtfach "studenten.txt" 'Informatik'*) →
 Informatik studieren 1 Frauen und 3 Maenner

- Schreiben Sie ein Testprogramm mit Hilfe der progn-Funktion, in welchem Sie mind. 10 Studenten/Innen erzeugen, diese mittels Ihrer *save*-Funktion in einer Datei namens *studenten.txt* speichern, anschließend alle Studenten aus der Datei mittels *loadstudis* einlesen und die Liste auf standard-out ausgeben. Schließlich soll durch Aufruf Ihrer Funktion *geschlechtfach* für das Fach Informatik ausgegeben werden, wie viele weibliche und wie viele männliche Studenten dieses Fach studieren.