

Übungsblatt 1

Ausgabe: 14.11.2019

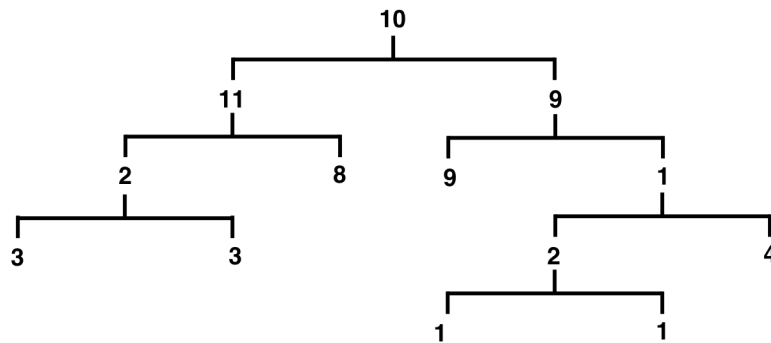
Abgabe: 28.11.2019

Aufgabe 1: Notenberechnung - optionale und Rest-Parameter

- Schreiben Sie eine Funktion (`defun endnote (prüfungsnote &optional vornote)`), die aus einer Prüfungsnote und einer evtl. übergebenen Vornote die Endnote berechnet. Das Resultat ist das arithmetische Mittel der beiden Noten. Die Vornote wird aber nur dann berücksichtigt, wenn sie die Prüfungsnote verbessert.
- Schreiben Sie eine weitere Funktion mit 2 optionalen Vornoten. Dabei muss jede jeweils besser sein als die Prüfungsnote.
- Schreiben Sie eine Funktion (`defun notendurchschnitt (listeVonNoten &optional bonus)`), die den Notenschnitt einer Liste von Noten berechnet. Zusätzlich kann ein Bonus (bezogen auf den Schnitt) angerechnet werden. Die beste Note ist eine 0.7 - auch nach Abzug eines evtl. angegebenen Bonus.
- Für die Berechnung der Klausurnote können (beliebig viele) Übungsnoten bis zu einem Gewicht von max. 50% berücksichtigt werden. Dabei ist der Faktor für eine Note 10%, für 2 Noten 20% etc. D.h. wenn Sie 5 und mehr Übungsnoten haben, ist das Gesamtgewicht 50%

Aufgabe 2: Mobiles

Ein Mobile ist eine Art abstrakter Skulptur, die aus Teilen besteht, die sich relativ zueinander bewegen können. Normalerweise besteht es aus Objekten, die mit feinen Schnüren an horizontalen Stöckchen befestigt und frei beweglich aufgehängt sind.



Wir können ein bestimmtes einfaches Mobile rekursiv definieren als aufgehängtes Objekt oder als Stöckchen mit daran hängenden (Teil-) Mobiles an jedem Ende. Wenn wir unterstellen, dass jedes Stöckchen in seiner Mitte aufgehängt ist, können wir das Mobile als binären Baum auffassen. Einzelne aufgehängte Objekte sollen durch Zahlen, entsprechend dem Gewicht der Objekte dargestellt werden. Dabei entspricht das erste Element dem Gewicht des Stöckchens, die beiden anderen Elemente stellen die Teil-Mobiles dar. Somit lässt sich z.B. der unterste Zweig dieses Mobiles durch die dreielementige Liste (2 1 1) darstellen.

Kompliziertere Teil-Mobiles werden durch geschachtelte 3-Element-Listen dargestellt. Im Beispiel das 2-stufige Teil-Mobile rechts durch die geschachtelte Listenstruktur (1 (2 1 1) 4)

Das abgebildete Mobile entspricht dann der Liste (10 (11 (2 3 3) 8) (9 9 (1 (2 1 1) 4)))

Ein Mobile sollte im Gleichgewicht sein. Das bedeutet, dass 2 Mobiles, die an den gegenüberliegenden Enden eines Stöckchens hängen, das gleiche Gewicht haben müssen. Das ist bei dem abgebildeten Mobile der Fall. Schreiben Sie eine Funktion `mobilep`, die für ein übergebenes Mobile feststellt, ob es das Mobile im Gleichgewicht ist. Der Rückgabewert ist `NIL`, bei Ungleichgewicht, im andern Fall das Gewicht.

(mobilep '(10 (11 (2 3 3) 8) (9 9 (1 (2 1 1) 4)))) → 64

(mobilep '(10 (11 (2 3 3) 8) (9 9 (1 (2 1 2) 4)))) → NIL

Achten Sie bei der Implementierung darauf, dass der Code möglichst kompakt und gut verständlich ist (dass er korrekt ist, versteht sich von selbst); Effizienz ist zweitrangig. Nutzen Sie Rekursion!