

Übungsaufgaben Nr. 1

- 1.) Definieren Sie eine Funktion “bilde-paare-1”.
Eingabe: Eine Liste mit 4 Elementen (arg1 arg2 arg3 arg4)
Wert: eine Liste mit 2-elementigen Unterlisten der Form ((arg1 arg2) (arg3 arg4))
- 2.) Definieren Sie eine Funktion “vorne-einfuegen”.
Eingabe: arg1, arg2: zwei beliebige Ausdrücke
liste: eine Liste
Wert: Eine Liste, so dass die Liste (arg1 arg2) als erstes Element in die Liste liste eingefügt wird

Aufrufbeispiel: (vorne-einfuegen 'a 'b '((c d) (e f))) --> ((a b) (c d) (e f))
- 3.) Definieren Sie eine Funktion “hinten-einfuegen”.
Eingabe: arg1, arg2: zwei beliebige Ausdrücke
liste: eine Liste
Wert: Eine Liste, so dass die Liste (arg1 arg2) als letztes Element in die Liste liste eingefügt wird

Aufrufbeispiel: (hinten-einfuegen 'a 'b '((c d) (e f))) --> ((c d) (e f) (a b))
- 4.) Definieren Sie eine Funktion “funkdef-1arg”.
Eingabe: fktname: ein Symbol
op: der Name einer Lisp-Funktion
const, var: Symbole
Wert: Eine Funktionsdefinition für fktname mit dem Parameter var, sodass op auf const und var angewendet werden.

Aufrufbeispiele: (funkdef-1arg 'add1 '+ 1 'zahl) --> (defun add1 (zahl) (+ zahl 1))
(funkdef-1arg 'add2 '+ 2 'zahl) --> (defun add2 (zahl) (+ zahl 2))
(funkdef-1arg 'mult3 '* 3 'zahl) --> (defun mult3 (zahl) (* zahl 3))
- 5.) Definieren Sie eine Funktion “funkdef-1arg+eval”.
Eingabe: fktname: ein Symbol
op: der Name einer Lisp-Funktion
const, var: Symbole
arg: eine Zahl
Effekt: Durch Aufruf der Funktion funkdef-1arg soll eine entsprechende Funktionsdefinition erzeugt und ausgewertet werden. Anschließend ist die neu definierte Funktion mit arg anzuwenden.
Wert: Der Wert, der bei Anwendung von fktname auf arg entsteht

Aufrufbeispiel: (funkdef-1arg+eval 'add2 '+ 2 'zahl 55)
Es soll eine Funktionsdefinition für add2 gebildet und ausgewertet werden.
add2 soll dann mit dem Argument 55 aufgerufen werden.
Als Wert des Aufrufs (funkdef-1arg+eval 'add2 '+ 2 'zahl 55) soll 57 geliefert werden.

Aufrufbeispiel: (funkdef-1arg+eval 'sub5 '- 5 'zahl 55)
Funktionswert ist 50
Als Seiteneffekt ist die Funktion sub5 definiert.

6.) Definieren Sie eine Funktion “Listenop”.

Eingabe: op: der Name einer LISP-Funktion

liste1, liste2: 2 Listen

Wert: Der Wert der durch Anwendung von op auf die beiden Listen entsteht

Aufrufbeispiele: (listenop 'list '(3 4) '((c d) (e f))) --> ((3 4) ((c d) (e f)))
(listenop 'append '(3 4) '((c d) (e f))) --> (3 4 (c d) (e f))

7.) Definieren Sie eine Funktion “key-unterliste”.

Eingabe: a, ein ausdruck

l, eine Liste mit Unterlisten

Wert: Die Unterliste von l, die a als erstes Element enthält

Aufrufbeispiel: (key-unterliste 'c '((a 4) (b 9) (c d) (e f))) --> (c d)

Hinweis: Die Aufgabe soll mit member und einer geeigneten Testfunktion gelöst werden.