

### Εισαγωγή

Η γλώσσα προγραμματισμού Lisp ( από το *List processing language*) σχεδιάστηκε ως γλώσσα συμβολικού προγραμματισμού.

Γενικά, τα σύμβολα μπορούν να χρησιμοποιηθούν για να αντιπροσωπεύσουν διαφορετικά πράγματα. Π.χ. μπορούμε να χρησιμοποιήσουμε το σύμβολο γιάννης για τον Γιάννη, και τα σύμβολα cube1, cube2, και cube3 για τρεις διαφορετικούς κύβους. Μπορούμε να ορίσουμε ιδιότητες, σχέσεις και μεθόδους συνδυασμών για τα σύμβολά μας, με όποιο τρόπο είναι επιθυμητός. Για παράδειγμα, η έκφραση (ψηλός γιάννης) θα μπορούσε να χρησιμοποιηθεί για να προσδιορίσει το ύψος του Γιάννη, και η έκφραση (on cube1 cube2 cube3) τη σχετική θέση των κύβων, όταν ο ένας είναι τοποθετημένος πάνω στον άλλο.

Η Lisp μοιάζει σε πολλά με άλλες γλώσσες, όπως η Pascal και η C, που επιτρέπουν την δημιουργία διαδικασιών υπό μορφή συναρτήσεων και καλούν τις διαδικασίες αυτές μέσα από άλλες διαδικασίες. Η Lisp διαφέρει στο ότι έχει μια απλούστερη σύνταξη, είναι λιγότερο περιοριστική σε όρους όσον αφορά απαιτήσεις συγκεκριμένου τρόπου πληκτρολόγησης των δεδομένων, και προσφέρει μια πλούσια συλλογή από έτοιμες διαδικασίες-συναρτήσεις και επαρκή υποστήριξη ανάπτυξης διαλογικών προγραμμάτων, όπως αποσφαλματοποιητή (debugger), μεταγλωττιστή (compiler), κλπ.

### Σύνταξη της Lisp

Η σύνταξη της Lisp είναι τόσο απλή με αποτέλεσμα να φαίνεται χωρίς δομή στους προγραμματιστές που είναι συνηθισμένοι σε γλώσσες με πιο αυστηρή σύνταξη. Στη Lisp, προγράμματα και δεδομένα παρουσιάζονται ως λίστες, όπου μια λίστα αναπαρίσταται με μια παρένθεση που ανοίγει, ακολουθούμενη από καμιά ή περισσότερες *εκφράσεις* και από μια παρένθεση που κλείνει. Μια έκφραση αντιστοιχεί σε ένα οποιοδήποτε αντικείμενο της Lisp (π.χ. έναν αριθμό ή μια συμβολοσειρά), ή σε μια άλλη λίστα που περιέχει καμιά ή περισσότερες εκφράσεις. Για παράδειγμα οι (), (1), (1 (2)) είναι όλες εκφράσεις, η πρώτη δε καλείται η “κενή” (null) λίστα. Η λίστα (1 (2)) αποτελείται από δυο εκφράσεις, τον αριθμό 1 και τη λίστα (2). Θα μπορούσαμε να πούμε, κατά συνεκδοχή, ότι κάθε λίστα αποτελεί ένα συντακτικά σωστό πρόγραμμα της Lisp.

Στη Lisp χρησιμοποιούνται δείκτες, των οποίων ο χειρισμός είναι πολύ εύκολος. Επίσης χρησιμοποιούνται αριθμοί ακέραιοι, κινητής υποδιαστολής αλλά και μιγαδικοί, και συμβολοσειρές, που περικλείονται μεταξύ διπλών εισαγωγικών (“string”), και μια ποικιλία από άλλους στοιχειώδεις τύπους δεδομένων, που καλούνται “πρωτογενείς” (*primitives*).

**Βασικά στοιχεία**

Αριθμητικοί Τελεστές / Αριθμητικές συναρτήσεις	Ενέργεια
(+ $x_1$ $x_2$ ... $x_n$ )	$x_1 + x_2 + \dots + x_n$
(* $x_1$ $x_2$ ... $x_n$ )	Το γινόμενο των : $x_1, x_2, \dots, x_n$
(- $x$ $y$ )	$x - y$
(/ $x$ $y$ )	$x / y$
(max $x_1$ $x_2$ ... $x_n$ )	Το μεγαλύτερο των : $x_1, x_2, \dots, x_n$
(min $x_1$ $x_2$ ... $x_n$ )	Το μικρότερο των : $x_1, x_2, \dots, x_n$
(abs $x$ )	Απόλυτη τιμή του $x$
(exp $x$ )	$e^x$
(expt $x$ $y$ )	$x^y$
(sqrt $x$ )	$\sqrt{x}$

Συγκριτικοί Τελεστές	Ενέργεια
(= $x$ $y$ )	$x = y$
(/= $x$ $y$ )	$x \neq y$
(< $x$ $y$ )	$x < y$
(> $x$ $y$ )	$x > y$
(<= $x$ $y$ )	$x \leq y$
(>= $x$ $y$ )	$x \geq y$

Λογικοί Τελεστές	Ενέργεια
(or $x_1$ $x_2$ ... $x_n$ )	Λογικό ή (θέλουμε να ισχύει τουλάχιστον ένα)
(and $x_1$ $x_2$ ... $x_n$ )	Λογικό και (θέλουμε να ισχύουν ταυτόχρονα όλα)
(not $x$ )	Λογικό όχι (ακριβώς το αντίθετο του $x$ )

## Αντικείμενα της Lisp

t = σύνολο όλων των αντικειμένων της lisp

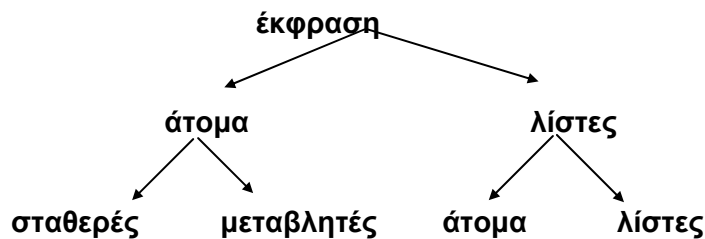
nil = κενός τύπος

Είδη Αντικειμένων: Άτομα - Λίστες

**Άτομα:** σταθερές ή μεταβλητές, π.χ. χαρακτήρες, αριθμοί, ονόματα συμβόλων

**Λίστες:** περιέχουν μεταξύ παρενθέσεων κανένα ή περισσότερα στοιχεία (**elements**), που μπορεί να είναι άτομα ή λίστες.

**Έκφραση:** Μια συλλογή από άτομα, λίστες και συνδεδεμένες λίστες, με συγκεκριμένη δομή, απαραίτητη για μια δεδομένη επεξεργασία, π.χ. συναρτήσεις, σύνθετες δομές (όπως π.χ. πίνακες) και διανύσματα.



## Βασικές Εντολές

Εντολή απόδοσης τιμής

(**setq** a b) a=b (Βάλε στο a την τιμή του b)

(**setq** a b c d e f) a=b , c=d , e=f

(**print** a) τύπωσε το περιεχόμενο του a

## Λίστες

(**list**  $x_1$   $x_2$  ...  $x_n$ ) : Δημιουργεί μία λίστα με τα στοιχεία  $x_1$   $x_2$  ...  $x_n$

(**setq** a (**list**  $x_1$   $x_2$  ...  $x_n$ )) : Απόδωσε στο a την λίστα με τα στοιχεία  $x_1$   $x_2$  ...  $x_n$

(**car** a) : Δίνει την κεφαλή της λίστας (στο προηγούμενο παράδειγμα θα δώσει  $x_1$ )

(**cdr** a) : Δίνει την ουρά της λίστας (στο προηγούμενο παράδειγμα θα δώσει  $(x_2 \dots x_n)$  )

**Συνδυασμοί της car και cdr**

<i>caaaar</i>	<i>cadaar</i>	<i>cdaaar</i>	<i>cddaar</i>
<i>caaadr</i>	<i>cadadr</i>	<i>cdaadr</i>	<i>cddadr</i>
<i>caaar</i>	<i>cadar</i>	<i>cdaar</i>	<i>cddar</i>
<i>caadar</i>	<i>caddar</i>	<i>cdadar</i>	<i>cdddar</i>
<i>caaddr</i>	<i>caddr</i>	<i>cdaddr</i>	<i>cddddr</i>
<i>caadr</i>	<i>caddr</i>	<i>cdadr</i>	<i>cdddr</i>
<i>caar</i>	<i>cadr</i>	<i>cdar</i>	<i>cddr</i>

(caar x) είναι ίδιο με το (car (car x))  
 (cdar x) είναι ίδιο με το (cdr (car x))  
 (cadar x) είναι ίδιο με το (car (cdr (car x)))  
 (cadr x) είναι ίδιο με το (car (cdr x))  
 (cddr x) είναι ίδιο με το (cdr (cdr x))  
 (caddr x) είναι ίδιο με το (car (cdr (cdr x)))

(**nth** n a) : Δίνει το n-οστό στοιχείο (ξεκινώντας από 0 – το πρώτο ...) της λίστας a

(**append** L1 L2) : ενώνει τις λίστες L1 και L2

**Πρόσβαση στο σχέδιο**

(**getpoint**) : Περιμένει ο χρήστης να εισάγει ένα σημείο (να κάνει κλικ) πάνω στο σχέδιο . Το σημείο επιστρέφεται με μορφή λίστας με (x y z) μορφή

(**getpoint** “μήνυμα”) : Περιμένει ο χρήστης να εισάγει ένα σημείο (να κάνει κλικ) πάνω στο σχέδιο βγάζοντας στην γραμμή εντολών το μήνυμα που έχουμε γράψει. Το σημείο επιστρέφεται με μορφή λίστας με (x y z) μορφή

Ασκήσεις :

1. Να γραφεί η αντίστοιχη έκφραση της Lisp της παράστασης :

$$\frac{x^2 - 4\sqrt{(x-1)^2 + (x-4)^2}}{|e^x - 2|}$$

2. Να γραφούν οι εντολές που διαβάζουν δύο σημεία από το σχέδιο (Grid) και επιστρέφουν την απόσταση (d αριθμός) των δύο σημείων καθώς και το μέσο τους (M σημείο – λίστα).

Δίνεται ότι η απόσταση των δύο σημείων A(x1,y1,z1) και B(x2,y2,z2) είναι :

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

Και το μέσο 
$$M\left(\frac{x1 + x2}{2} \quad \frac{y1 + y2}{2} \quad \frac{z1 + z2}{2}\right)$$

3. Μία λίστα L=((x1 y1) (x2 y2) (x3 y3)) περιέχει τις συντεταγμένες τριών σημείων στο επίπεδο. Να γραφούν οι εντολές που επιστρέφουν σημείο A το οποίο έχει

συντεταγμένες  $A(\max(x1, x2, x3) \quad \min(y1, y2, y3))$

4. Να γραφούν οι εντολές που αποδίδουν την συνένωση δύο λιστών (L1,L2) σε μία (L3) που περιέχει όλα τα στοιχεία της αρχικής και μίας (L4) που περιέχει δύο στοιχεία (κάθε στοιχείο είναι κάθε λίστα ξεχωριστά)