



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Ευφυής Προγραμματισμός

Ενότητα 5: Ειδικές Μεταβλητές-Χειρισμός Αρχείων

Ιωάννης Χατζηλυγερούδης

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# Περιεχόμενα ενότητας

## Ειδικές Μεταβλητές-Χειρισμός Αρχείων

1. Ειδικές Μεταβλητές
2. Αλφαριθμητικά-Χαρακτήρες
3. Διαχείριση Αρχείων



# Ειδικές Μεταβλητές

# Ειδικές Μεταβλητές (1)

Πχ. (defun first-of-first1 (x)  
 (first (auxiliary1 x)))

Πχ. (defun auxiliary1 (x)  
 (first x))

- $\text{> (first-of-first1 '((a b) (c d) (e f)))} \rightarrow A$

# Ειδικές Μεταβλητές (2)

Πχ.((defun first-of-first2 (x)  
 (first (auxiliary2))))

Πχ.((defun auxiliary2 ()  
 (first x)))

>(first-of-first2 '((a b) (c d) (e f))) →ERROR

## Ειδικές Μεταβλητές (3)

- Η LISP χρησιμοποιεί ιδεατούς φράχτες για να βρει τις τιμές των μεταβλητών.
- Στον ιδεατό φράχτη της `auxiliary2` δεν μπορεί να βρει την τιμή της `x`.
- **defvar**: Οι προσδέσεις της μεταβλητής τοποθετούνται στο «μητρώο κλήσεων» (record of calls) της LISP και είναι ορατές από παντού.

# Ειδικές Μεταβλητές (4)

```
(defvar *x*)
```

```
(defun first-of-first2 (*x*)  
  (first (auxiliary2)))
```

```
(defun auxiliary2 ()  
  (first *x*))
```

```
>(first-of-first2 '((a b) (c d) (e f))) →A
```

# Ειδικές Μεταβλητές (5)

```
(defvar *x* 15)
```

- δηλώνει ότι η `*x*` είναι special (ειδική) και της εκχωρεί το 15

```
> *x* →15
```

```
> (defvar *x* 20)
```

```
> *x* →15 (δεν αλλάζει η αρχική τιμή που δόθηκε)
```



# Ειδικές Μεταβλητές (6)

- (defun compute-taxythta (**litra-ana-km** xronos katanalosi-litra)

(/ (compute-diasthma katanalosi-litra) xronos))

- (defun compute-diasthma (katanalosi-litra)

(/ katanalosi-litra litra-ana-km))

>(compute-taxythta 0.10 5 30)→ERROR

litra-ana-km δεν έχει τιμή μέσα στην compute-diasthma , άρα είναι άγνωστη.

# Ειδικές Μεταβλητές (7)

- (defun compute-taxythta (litra-ana-km xronos katanalosi-litra)  
(declare (special litra-ana-km))  
(/ (compute-diasthma katanalosi-litra) xronos))
  - (defun compute-diasthma (katanalosi-litra)  
(/ katanalosi-litra litra-ana-km))
- >(compute-taxythta 0.10 5 30) →60

# Αλφαριθμητικά-Χαρακτήρες

# Αλφαριθμητικά-Χαρακτήρες (1)

- "abc", "ioannis" : αλφαριθμητικά (strings)
  - (elt "abcd" 0) → #\a
  - (length " abcd ") → 6
  - (reverse "abcd") → " dcba"
  - (subseq "ioannis" 2 4) → " an "

# Αλφαριθμητικά-Χαρακτήρες (2)

- (some #'<function-name> <sequence>)
  - (some #' (lambda(x) (char= #\b x)) " abcd ") → T
  - (some #'numberp '(a b 3 c d)) → T
- (every #'<function-name> <sequence>)
  - (every #' (lambda(x) (char= #\b x)) " abcd " ) →  
NIL

# Αλφαριθμητικά-Χαρακτήρες (3)

- `(string= "abcd" "xyzw")` → NIL
- `(string= "abcd" "abcd")` → T
- `(string= "abcd" "ABCD")` → NIL
- `(string-equal "abcd" "xyzw")` → NIL
- `(string-equal "abcd" "abcd")` → T
- `(string-equal "abcd" "ABCD")` → T

# Αλφαριθμητικά-Χαρακτήρες (4)

- `#\a` : χαρακτήρας (character)
  - `(char= #\c #\b) → NIL`
  - `(char= #\d #\d) → T`
  - `(char= #\d #\D) → NIL`
  - `(char-equal #\c #\b) → NIL`
  - `(char-equal #\D #\D) → T`
  - `(char-equal #\d #\D) → T`

# Αλφαριθμητικά-Χαρακτήρες (5)

- (search "Athens" "Athens Greece") → 0
- (search "Greece" "Athens Greece") → 7
- (search "Rhodos" "Athens Greece") → NIL
- (search "GREECE" "Athens Greece") → NIL
- (search "GREECE" "Athens Greece" :test #'char-equal) → 7

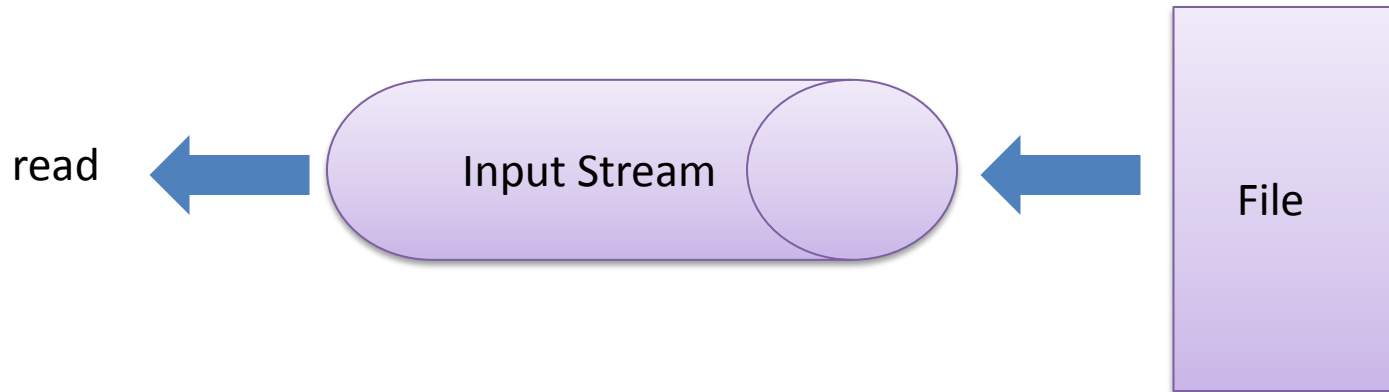
Χρησιμοποιείται και για λίστες:

- (search '(Athens) '(Athens Greece)) → 0
- (search '(Greece) '(Athens Greece)) → 1
- (search '(Rhodos) '(Athens Greece)) → NIL



# Διαχείριση Αρχείων

# Διαχείριση Αρχείων (1)



- Ανάγνωση από αρχείο
  - Δημιουργία stream
  - Πρόσδεση μεταβλητών στο stream
  - Χρήση του with-open-file

## Διαχείριση Αρχείων (2)

- (with-open-file (<όνομα-stream>  
                  <διαδρομή αρχείου>  
                  :direction :input)  
                  ...  
                  (read <όνομα-stream>  
                  ...))
- stream εισόδου: συνδετήρας read-αρχείου
- Το read διαβάζει ένα άτομο τη φορά

# Διαχείριση Αρχείων (3)

```
(defun read-from-file1 (file)
  (with-open-file (stud-stream file :direction :input)
    (dotimes (n 2)
      (print (read stud-stream))))))
```

- stud1.txt

```
((georgia papa) (ai systems))
((giannis xatzis) (prog alg db))
((kostas ntokis) (database ai))
((manos andreou) (alg))
((nikos kakos) (ds alg))
```

```
(read-from-file1 "c:/.../stud1.txt")
```



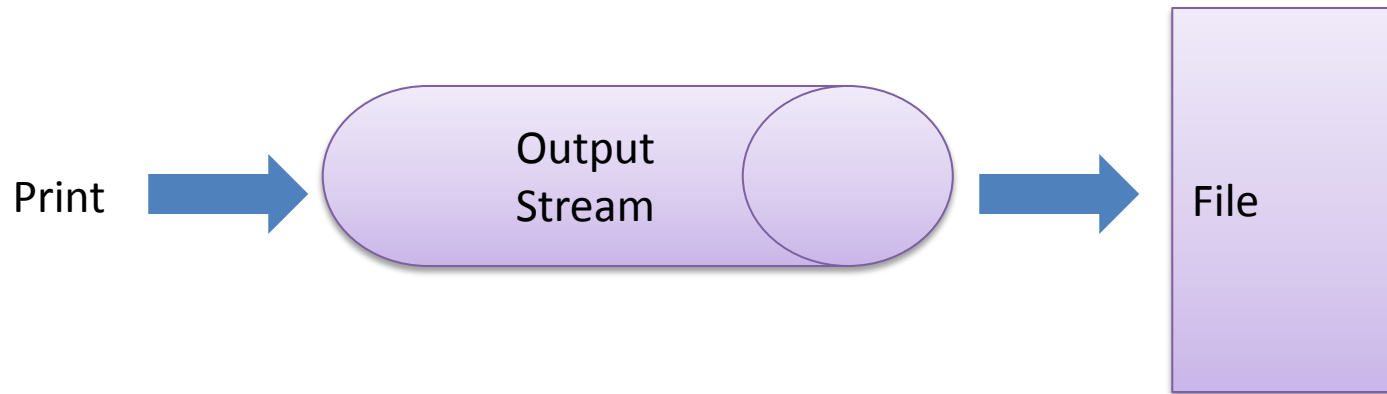
```
((georgia papa) (ai systems))
((giannis xatzis) (prog alg db))
```

# Διαχείριση Αρχείων (5)

## Άλλες συναρτήσεις ανάγνωσης

- `read-line`
  - Διαβάζει μέχρι το επόμενο CR ή EOF
  - Επιστρέφει δύο τιμές
    - Αλφαριθμητικό
    - NIL/T (αν συναντήσει eof ενώ διαβάζει γραμμή)
- `read-char`
  - Διαβάζει ένα μόνο χαρακτήρα (π.χ. `#\a`)

# Διαχείριση Αρχείων



- Εγγραφή σε αρχείο
  - Δημιουργία stream
  - Πρόσδεση μεταβλητών στο stream
  - Χρήση του with-open-file

# Διαχείριση Αρχείων

(with-open-file (<όνομα-stream>  
                  <διαδρομή αρχείου>  
                  :direction :output>)

...

(print <έκφραση-εκτύπωσης> <όνομα-stream>  
...)



θα αποτιμηθεί και η τιμή της θα εκτυπωθεί

- stream εισόδου: συνδετήρας print-αρχείου

# Διαχείριση Αρχείων (3)

```
(defun write-to-file1 (file)
  (with-open-file (stud-stream file :direction :output)
    (dotimes (n 5)
      (print (read) stud-stream))))
```

```
(write-to-file1 "c:/.../stud1.txt")
```



```
((georgia papa) (ai systems))
((giannis xatzis) (prog alg db))
((kostas ntokis) (database ai))
((manos andreou) (alg))
((nikos kakos) (ds alg))
```

stud1.txt

```
((georgia papa) (ai systems))
((giannis xatzis) (prog alg db))
((kostas ntokis) (database ai))
((manos andreou) (alg))
((nikos kakos) (ds alg))
```



# Διαχείριση Αρχείων (4)

Η with-open-file...

- σε κάθε προσπάθεια ανάγνωσης αρχείου με συγκεκριμένο όνομα, αρχίζει να το διαβάζει από την αρχή
- σε κάθε προσπάθεια εγγραφής αρχείου με συγκεκριμένο όνομα, δημιουργεί ένα νέο, κενό αρχείο
- φροντίζει, αν συμβεί κάποιο λάθος στην εκτέλεσή της, τα αρχεία που είναι ανοικτά να κλείσουν ομαλά

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Ιωάννης Χατζηλυγερούδης 2015.  
«Ευφυής Προγραμματισμός». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη  
δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1095/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

