



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Ευφυής Προγραμματισμός

Ενότητα 4: Λίστες Ιδιοτήτων-Πίνακες-Δομές

Ιωάννης Χατζηλυγερούδης

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Περιεχόμενα ενότητας

Λίστες Ιδιοτήτων-Πίνακες-Δομές

1. Λίστες Ιδιοτήτων
2. Πίνακες
3. Δομές



Λίστες Ιδιοτήτων

Λίστες Ιδιοτήτων (1)

- Η LISP επιτρέπει σε ένα σύμβολο να έχει τιμές ιδιοτήτων (property values).
- Προσάρτηση ιδιοτήτων-τιμών σ' ένα σύμβολο
- Λίστα ιδιοτήτων (property list): το σύνολο των ιδιοτήτων-τιμών των προσαρτημένων σ' ένα σύμβολο, που παριστάνει μια συγκεκριμένη οντότητα
- Π.χ. 'giannis' είναι ένα σύμβολο που παριστάνει ένα συγκεκριμένο φοιτητή, στον οποίο μπορούμε να προσαρτήσουμε τις ιδιότητες, όπως 'department', 'parents' κλπ.

Λίστες Ιδιοτήτων (2)

Τύποι συναρτήσεων δημιουργίας-διαχείρισης λίστας ιδιοτήτων

- Δημιουργίας-`constructor` (δημιουργεί λίστα ιδιοτήτων και αποδίδει τιμές)
- Ανάγνωσης-`reader` (προσπελαύνει τιμή ιδιότητας)
- Διαγραφής-`remover` (διαγράφει ιδιότητα)

Λίστες Ιδιοτήτων (3)

Δημιουργία/απόδοση τιμών (`setf` σε συνδυασμό με την `get`)

```
(setf (get <symbol> <property name>) <property value>)
```

Προσπέλαση τιμής ιδιότητας (`get`)

```
(get <symbol> <property name>)
```

Λίστες Ιδιοτήτων (4)

Πχ.

- (setf (get 'giannis 'department) 'computer-engineering) → COMPUTER-ENGINEERING
- (setf (get 'giannis 'parents) '(kostas maria)) → (KOSTAS MARIA)
- (setf (get 'giannis 'siblings) '(giorgos eleni)) → (GIORGOS ELENI)
- (get 'giannis 'department) → COMPUTER-ENGINEERING
- (get 'giannis 'parents) → (KOSTAS MARIA)

Λίστες Ιδιοτήτων (5)

Διαγραφή ιδιότητας-τιμής (remprop, setf-nil)

(remprop <σύμβολο> <όνομα ιδιότητας>)

Πχ.

(remprop 'giannis 'parents) → T

(get 'giannis 'parents) → NIL

Πίνακες

Πίνακες (1)

Δημιουργία ενός μονοδιάστατου πίνακα:

- `(setf <array name> (make-array <dimension>))`
- `(setf grades (make-array 4)) → #(NIL NIL NIL NIL)`
- `(make-array 5) → #(NIL NIL NIL NIL NIL)`

Πίνακες (2)

Αρχικοποίηση ενός πίνακα:

- Χρήση της παραμέτρου `:initial-element`
- Χρήση της παραμέτρου `:initial-contents`
- Με την ανάθεση τιμών σε κάθε στοιχείο του μέσω της `aref`.

Πίνακες (3)

Αρχικοποίηση ενός μονοδιαστατου πίνακα:

- `(setf grades (make-array 4 :initial-element '0))` → `#(0 0 0 0)`
- `(setf grades (make-array 4 :initial-contents '(5 8 3 2)))` → `#(5 8 3 2)`
- `(setf (aref grades 0) '5)` → 5
- `(setf (aref grades 1) '8)` → 8
- `(setf (aref grades 2) '3)` → 3
- `(setf (aref grades 3) ' 2)` → 2

Πίνακες (4)

Η προσπέλαση των στοιχείων ενός πίνακα γίνεται ως εξής:

- (aref <array name> <element index>)
- (aref grades 1)→8
- (aref grades 3)→2
- (aref grades 0)→5

Πίνακες (5)

Δημιουργία ενός δισδιάστατου πίνακα:

- (setf <array name> (make-array <dimensions>))
- (setf grades (make-array '(2 3))) → #2A((NIL NIL NIL) (NIL NIL NIL))
- Δισδιάστατος πίνακα με όνομα grades που έχει $2 \times 3 = 6$ στοιχεία

Πίνακες (6)

Αρχικοποίηση

- `(setf grades (make-array '(2 3) :initial-element 0))` → `#2A((0 0 0) (0 0 0))`
- `(setf marks (make-array '(3 3) :initial-contents '((7 6 8) (9 5 6) (a b c))))` → `#2A((7 6 8) (9 5 6) (A B C))`
- `(setf (aref grades 0 0) 3)` → `3`
- `(setf (aref grades 0 1) 2)` → `2`

Πίνακες (7)

Προσπέλαση των στοιχείων ενός δισδιάστατου πίνακα

- (aref <array name> <element index>)
- (aref grades 0 0) \rightarrow 3
- (aref grades 0 1) \rightarrow 2

Πίνακες (8)

Συναρτήσεις Πίνακα:

- array-dimension:
(array-dimension <array name> <axis num>)
- array-dimensions
(array-dimensions <array name>)

Πίνακες (9)

Πχ.

- (array-dimension grades 0) $\rightarrow 2$
- (array-dimension grades 1) $\rightarrow 3$
- (array-dimensions grades) $\rightarrow (2\ 3)$

Δομές

Δομές (1)

- Δομή :
 - πεδία (fields)
 - τιμές (values) των πεδίων.
- (defstruct <structure name>
(<field1> <value1>
(<field2> <value2>
...
(<fieldn> <valuen>))

Δομές (2)

Πχ.

- (defstruct student
 (year-of-study nil)
 (sex nil)
 (grades nil))

Δομές (3)

- **defstruct**

- δημιουργεί μια δομή, αλλά όχι στιγμιότυπα της
- δημιουργεί όμως μια συνάρτηση-δημιουργό στιγμιοτύπων:

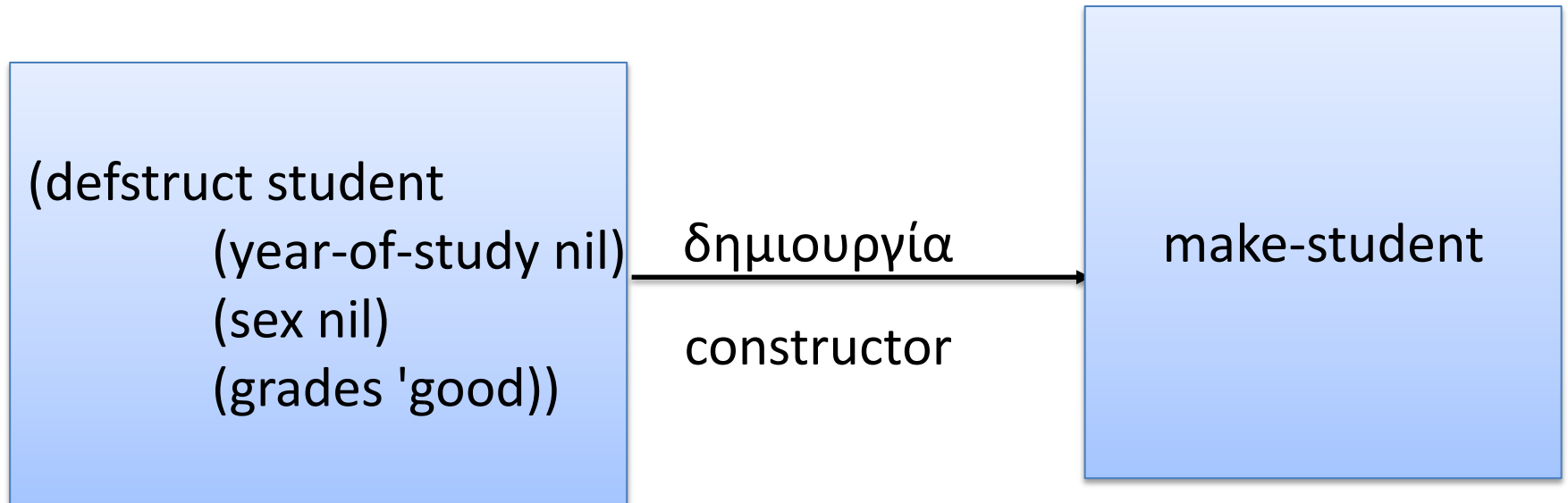
`make-<structure name>`

- Επίσης, δημιουργεί και συναρτήσεις ανάγνωσης για την προσπέλαση τιμών των πεδίων:

`<structure name>-<field name>`

- δημιουργεί ένα κατηγορημα (συνάρτηση) για έλεγχο του τύπου στιγμιοτύπων της δομής
- καλή πρακτική: όλες οι δομές στο δικό τους αρχείο

Δομές (4)



- `defstruct`: δημιουργεί τη δομή `person`
- `make-student`: δημιουργεί στιγμιότυπα της `student`

Δομές (5)

Πχ. `(setf student-inst1 (make-student))` → `#S(STUDENT :YEAR-OF-STUDY NIL :SEX NIL :GRADES GOOD)`

- Η `student-inst1` έχει μόνο μια “non nil” τιμή, την τιμή `good` → έχει τις προκαθορισμένες τιμές της δομής

Πχ. `(setf student-inst2 (make-student :sex 'male))`

→ `#S(STUDENT :YEAR-OF-STUDY NIL :SEX MALE :GRADES GOOD)`

→ Η `student-instance-2` έχει δύο “non nil” τιμές

- ‘male’ στο πεδίο `sex` (από την `make-person`)
- ‘good’ στο πεδίο `grades` (από την `defstruct`)

Δομές (6)

- `(student-sex student-inst2) → MALE`
- `(student-grades student-inst2) → GOOD`
- `(setf (student-sex student-inst1) 'female) → FEMALE`
- `(student-sex student-inst1) → FEMALE`
- Τώρα το στιγμιότυπο `student-inst1` έχει δύο “non nil” τιμές. Η επιπλέον προέρχεται από την `setf`.

Δομές (7)

- (student-p student-inst1) → T
- student-p ' (this is a list no a student instance) → NIL
- Συνάρτηση describe: Εκτυπώνει Περιγραφές
- (describe student-inst1)
 - #S(STUDENT :YEAR-OF-STUDY NIL :SEX FEMALE :GRADES GOOD)
is a STUDENT
 - YEAR-OF-STUDY NIL
 - SEX FEMALE
 - GRADES GOOD

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.

