



ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

AGRICULTURAL UNIVERSITY OF ATHENS

# Εισαγωγή στην Επιστήμη των Υπολογιστών

## Ενότητα 4:

Λογισμικό Υπολογιστή (2/3), 2ΔΩ

Τμήμα: Αγροτικής Οικονομίας & Ανάπτυξης

Διδάσκων: Θεόδωρος Τσιλιγκιρίδης



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





# Μαθησιακοί στόχοι

Η Ενότητα 4 διαπραγματεύεται τα παρακάτω θέματα που αφορούν το Λογισμικό του Υπολογιστή υπολογιστών:

- Ιστορικό των Λειτουργικών Συστημάτων
- Αρχιτεκτονική Λειτουργικών Συστημάτων
- Ανάπτυξη Λογισμικού
- Γλώσσες Προγραμματισμού



# Λέξεις κλειδιά

- Αλγόριθμοι
- Φάσεις ανάπτυξης λογισμικού
- Αναπαράσταση αλγορίθμων
- Δομές ελέγχου
- Δομές επανάληψης
- Αναδρομικές δομές
- Πρωτογενείς δομές δεδομένων
- Αποδοτικότητα και ορθότητα αλγορίθμων
- Ορθότητα αλγορίθμων



# Ενότητα 4: Λογισμικό Υπολογιστή 1/2

- ΕΝΟΤΗΤΑ 4.1: Εισαγωγή στα Λειτουργικά Συστήματα (2ΔΩ)
- ΕΝΟΤΗΤΑ 4.2: Ανάπτυξη Λογισμικού (2ΔΩ)
- ΕΝΟΤΗΤΑ 4.3: Γλώσσες Προγραμματισμού (1ΔΩ)
- ΕΝΟΤΗΤΑ 4.4: Αλγόριθμοι - Παραδείγματα (2ΔΩ) –  
Φροντιστήριο (\*)



# Λογισμικό Υπολογιστή:

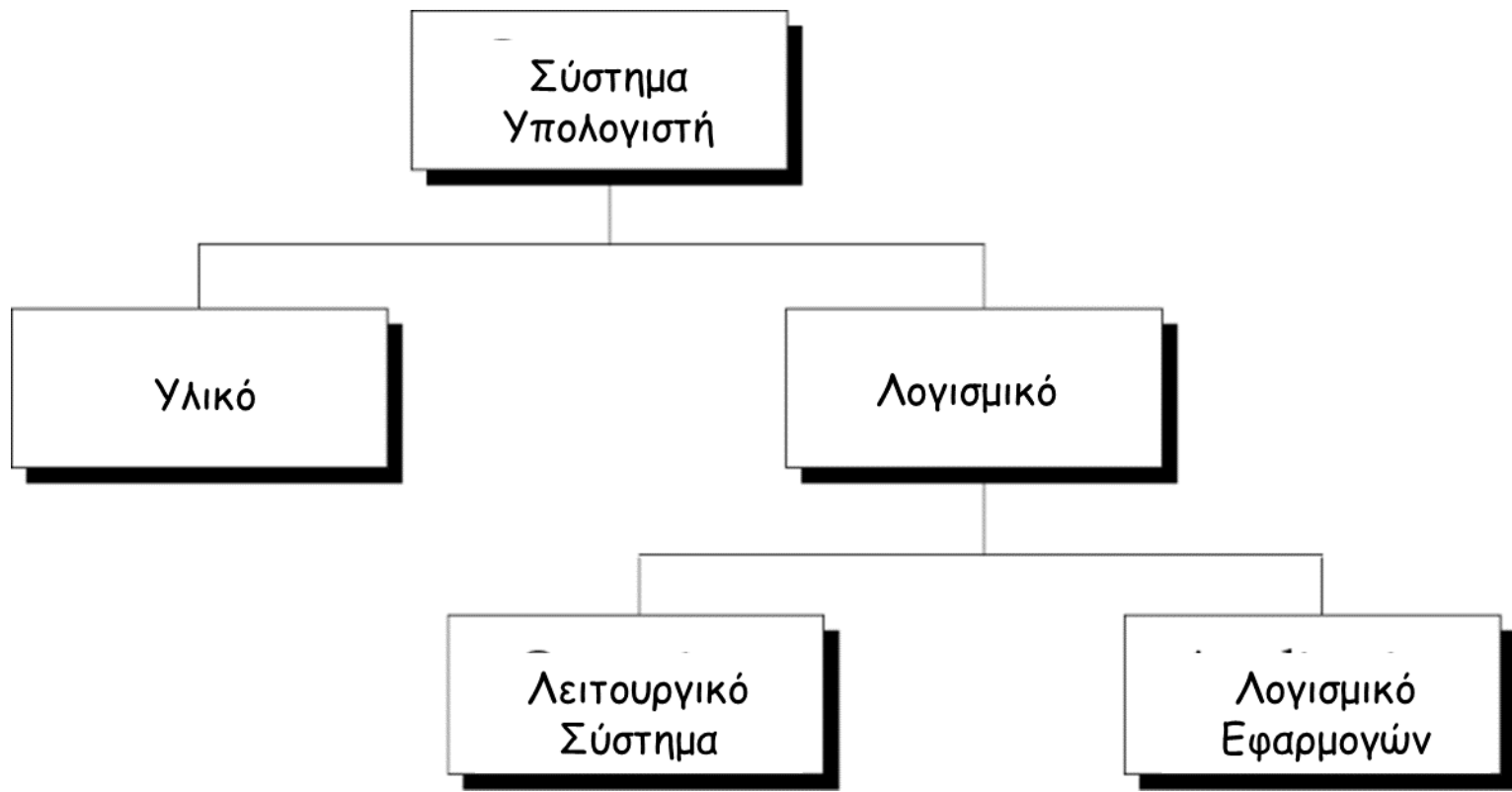
## Ορισμοί 1/2

- Πρόγραμμα υπολογιστή είναι ένα σύνολο ανενεργών οδηγιών που κατευθύνουν τον υπολογιστή και απαιτούνται προκειμένου να μετατραπούν τα δεδομένα σε πληροφορίες.
  - Οι οδηγίες απαρτίζονται από εντολές οι οποίες και συνιστούν τον κώδικα ή τη γλώσσα προγραμματισμού
    - Λογισμικό === Προγράμματα
    - === Αλγόριθμοι + Δομές δεδομένων
  - Εργασία: Ένα πρόγραμμα που πρόκειται να εκτελεστεί.
  - Διεργασία: Ένα πρόγραμμα που βρίσκεται στη μνήμη και περιμένει πόρους.
- Το Λειτουργικό Σύστημα (ΛΣ) –Operating System (OS) - είναι μια διεπαφή μεταξύ του υλικού ενός υπολογιστή και του χρήστη που διευκολύνει τόσο την εκτέλεση άλλων προγραμμάτων όσο και την πρόσβαση στο υλικό και λογισμικό.



# Λογισμικό Υπολογιστή:

## Ορισμοί 2/2





# Φάσεις Ανάπτυξης Λογισμικού

- Βήμα 1ο: Προδιαγραφές Προγράμματος
- Βήμα 2ο: Σχεδίαση
- Βήμα 3ο: Κωδικοποίηση
- Βήμα 4ο: Δοκιμές
- Βήμα 5ο: Τεκμηρίωση
- Βήμα 6ο: Συντήρηση



# Ανάπτυξη Αλγορίθμου (Προγράμματος)

## Βήμα 1<sup>ο</sup>: Προδιαγραφές

- Καθορισμός των στόχων
- Καθορισμός της επιθυμητής εξόδου
- Καθορισμός της απαιτούμενης εισόδου
- Καθορισμός των απαιτούμενων διαδικασιών
- Τεκμηρίωση των προδιαγραφών του προγράμματος





# Ανάπτυξη Αλγορίθμου (Προγράμματος)

## Βήμα 2<sup>ο</sup>: Σχεδίαση

**Σχεδίαση:** Παρουσίαση της λογικής λύσης των απαιτούμενων διαδικασιών

Χρησιμοποίηση τεχνικών δομημένου προγραμματισμού

- Από πάνω προς τα κάτω σχεδίαση προγράμματος
- Δομές Ελέγχου

Αλγόριθμος - Εργαλεία

- Διαγράμματα ροής
- Ψευδοκώδικες - Κώδικας



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Δομημένος προγραμματισμός

Από πάνω προς τα κάτω σχεδίαση  
προγράμματος

- Τμηματικός (modular) προγραμματισμός  
Συναρτήσεις - Υποπρογράμματα

Δομές ελέγχου

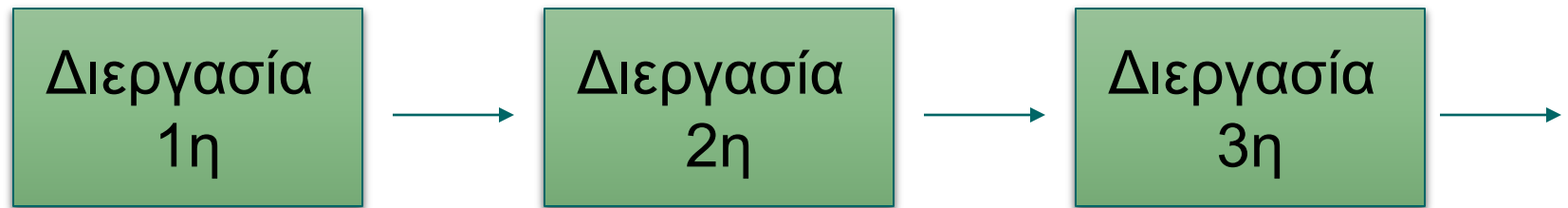
- Σειραϊκή δομή
- Δομή επιλογής (υπό συνθήκη εκτέλεση)
- Δομή επανάληψης

Παρατήρηση: Μία είσοδος - Μία έξοδος



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Δομές ελέγχου - Σειραϊκή δομή





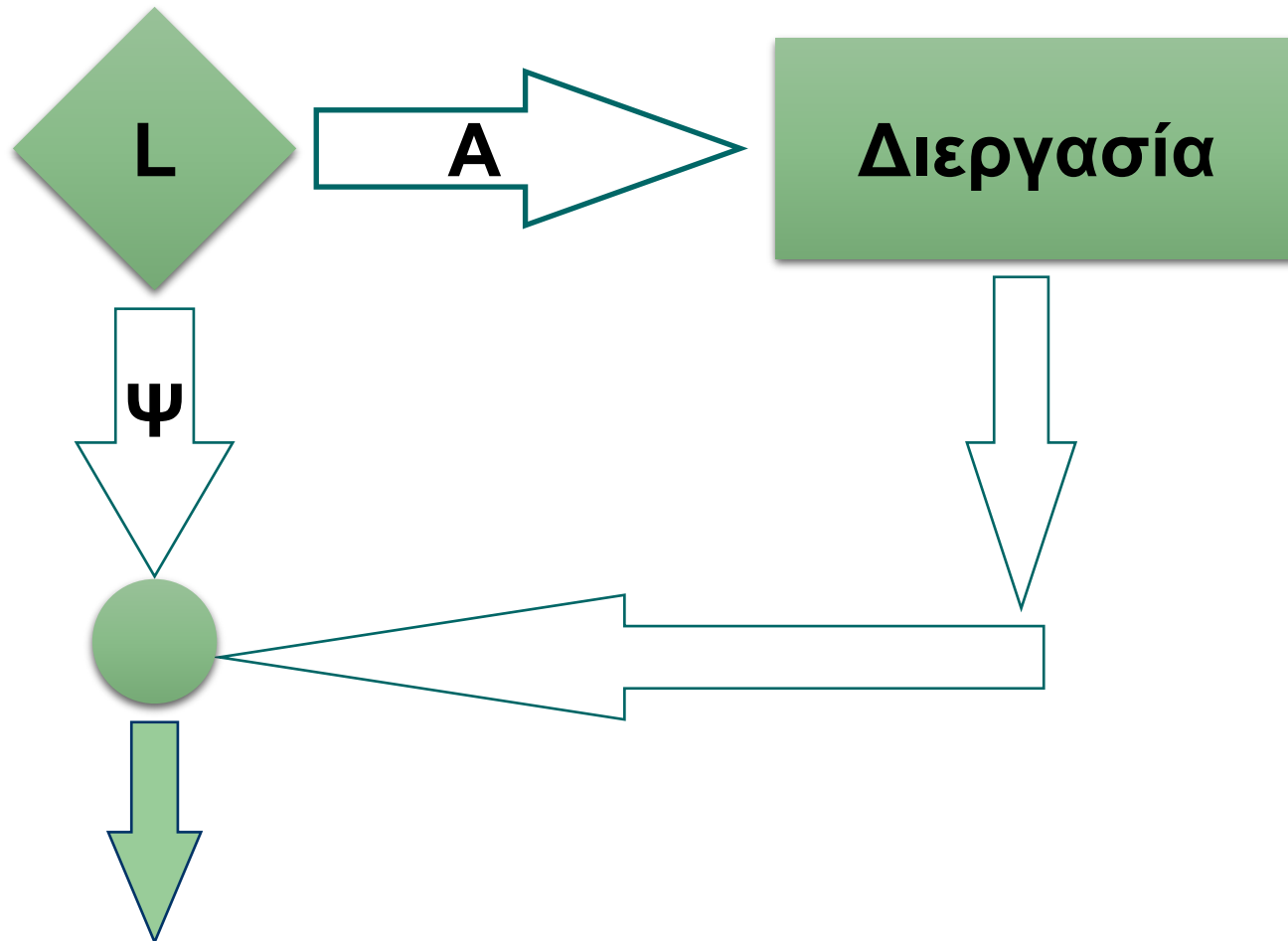
# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Δομές ελέγχου - Δομή επιλογής

- Δομή *if - end if*
- Δομή *if - then - else - end if*
- Δομή φωλιάς *if - then - else - end if*
- Δομή *case*

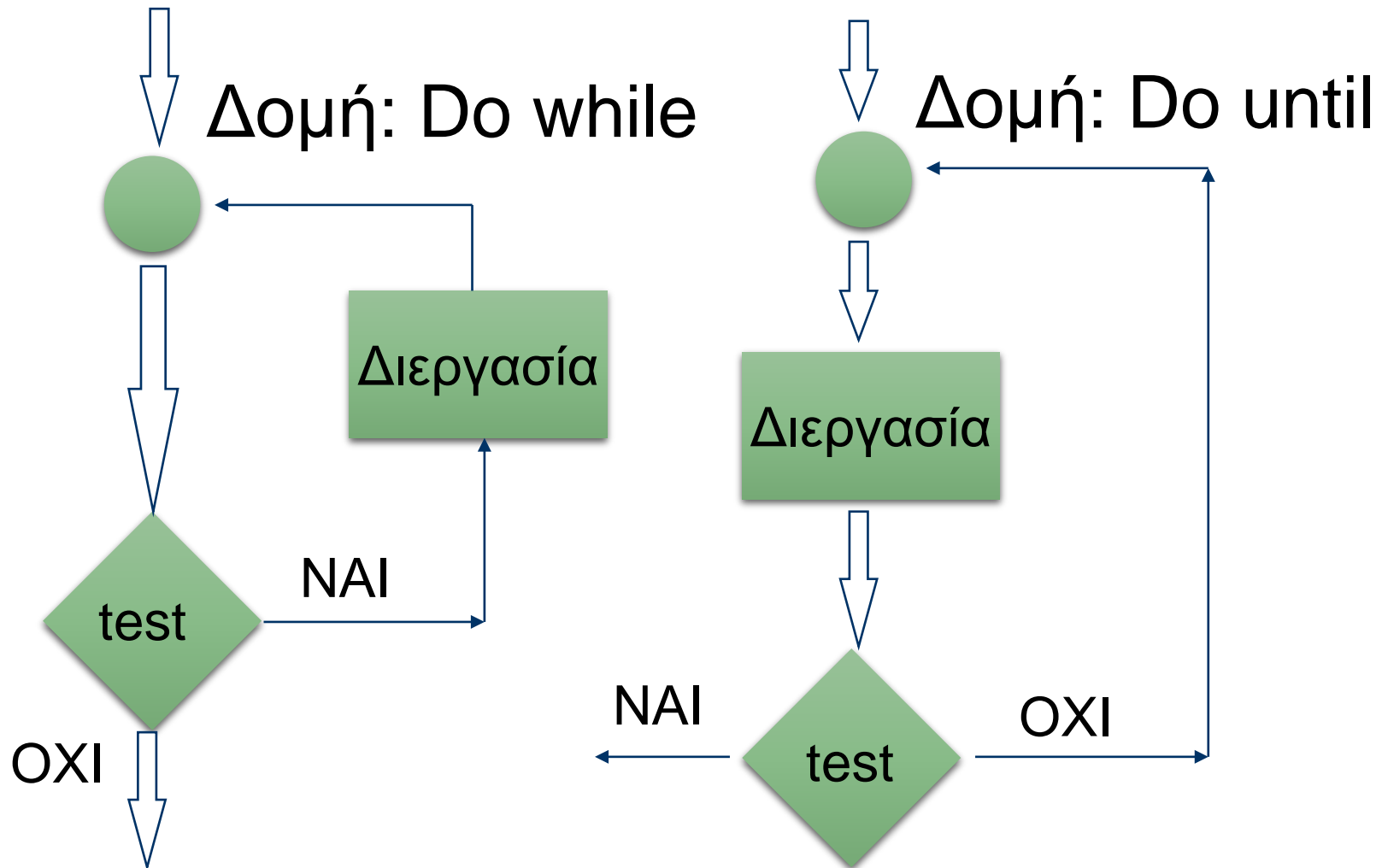


# Ανάπτυξη Προγράμματος Βήμα 2<sup>ο</sup>: Δομές ελέγχου: Δομή επιλογής - Λογικό if





# Ανάπτυξη Προγράμματος Βήμα 2<sup>ο</sup>: Δομές ελέγχου: Δομή επανάληψης





# Ανάπτυξη Προγράμματος

## Βήμα 2ο: Αλγόριθμος - Εργαλεία

- Αλγόριθμος:** Ακολουθία των βημάτων που απαιτούνται για να λυθεί το φυσικό πρόβλημα
- Παρουσίαση μέσω σύντομης περιγραφής
  - Παρουσίαση μέσω διαγράμματος ροής
  - Παρουσίαση μέσω τυπικής αλγοριθμικής γλώσσας



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος - Παράδειγμα 1/2

**Παράδειγμα:** Να διατάξετε τα  $n$  στοιχεία ενός πίνακα κατά αύξουσα σειρά

*1ο run*

a(1)	60	42	42	42	42
a(2)	42	60	60	60	60
a(3)	75	75	75	75	75
a(4)	83	83	83	83	27
a(5)	27	27	27	27	83





# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παράδειγμα 2/2

**Παράδειγμα:** Να διατάξετε τα  $n$  στοιχεία ενός πίνακα κατά αύξουσα σειρά (ταξινόμηση)

<i>Run</i>	<i>αρχή</i>	<i>1ο</i>	<i>2ο</i>	<i>3ο</i>	<i>4ο</i>
<b>a(1)</b>	<b>60</b>	<b>42</b>	<b>42</b>	<b>42</b>	<b>27</b>
<b>a(2)</b>	<b>42</b>	<b>60</b>	<b>60</b>	<b>27</b>	<b>42</b>
<b>a(3)</b>	<b>75</b>	<b>75</b>	<b>27</b>	<b>60</b>	<b>60</b>
<b>a(4)</b>	<b>83</b>	<b>27</b>	<b>75</b>	<b>75</b>	<b>75</b>
<b>a(5)</b>	<b>27</b>	<b>83</b>	<b>83</b>	<b>83</b>	<b>83</b>



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος - Σύντομη περιγραφή

- Αν το πρώτο στοιχείο του πίνακα  $n$  στοιχείων είναι μεγαλύτερο του δευτέρου εναλλάσσουμε τη θέση τους.
- Επαναλαμβάνουμε τη διαδικασία με το δεύτερο και τρίτο στοιχείο του πίνακα. Συνεχίζουμε με τον ίδιο τρόπο μέχρι να εξαντληθούν όλα τα στοιχεία του πίνακα και να εμφανισθεί το μεγαλύτερο στοιχείο του στη τελευταία θέση.
- Επαναλαμβάνουμε την ίδια διαδικασία στο νέο πίνακα των  $n-1$  στοιχείων, εξαιρώντας το  $n$ -οστό και συνεχίζουμε μέχρι να διαταχθεί όλος ο πίνακας



# Ανάπτυξη Προγράμματος

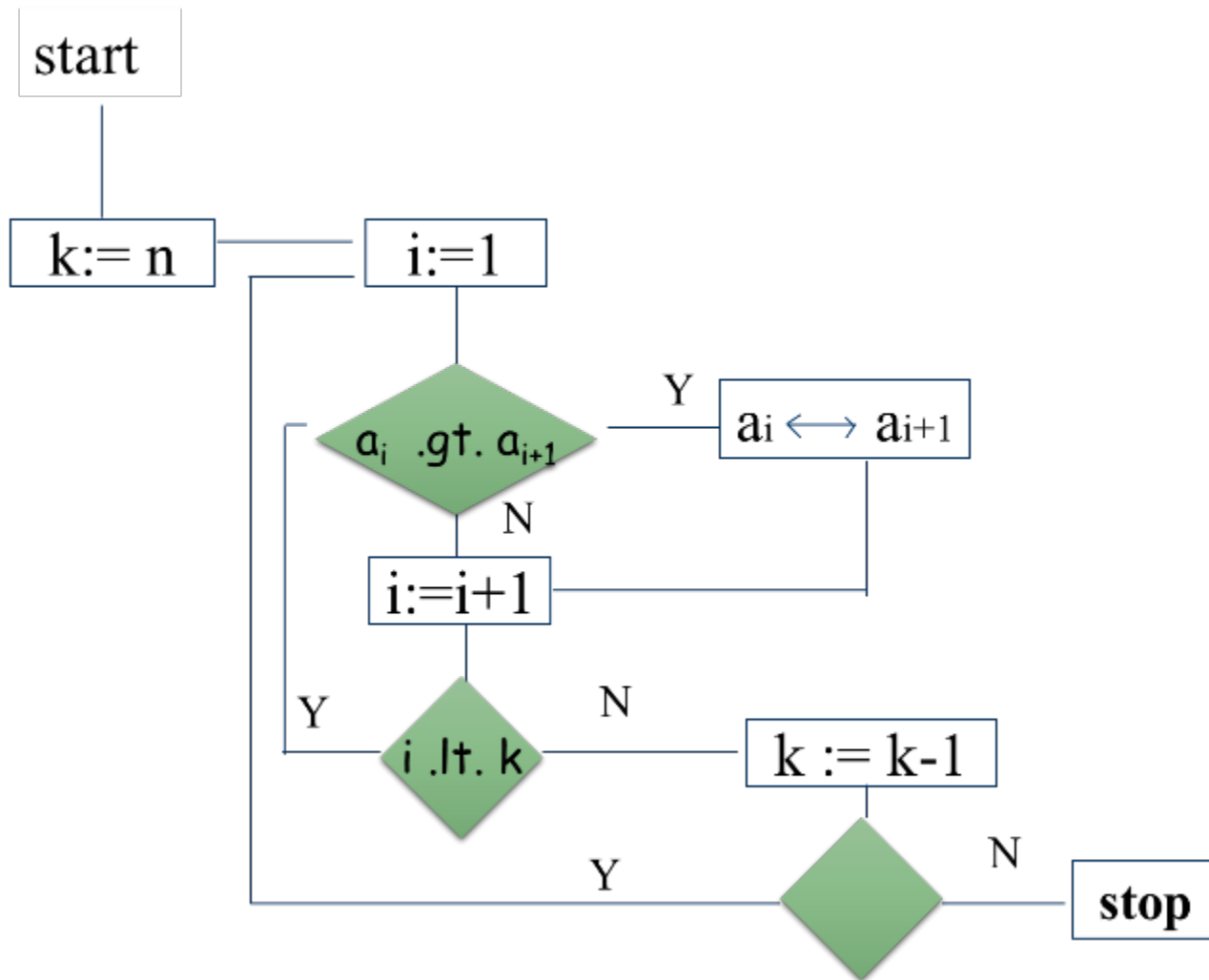
## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Αλγοριθμική γλώσσα

$k \leftarrow n$   
next scan :  $i \leftarrow 1$   
next pair: *if*  $a_i > a_{i+1}$  *then*  $a_i \leftrightarrow a_{i+1}$   
 $i \leftarrow i+1$   
*if*  $i \nless k$  *then go to next pair*  
 $k \leftarrow k-1$   
*if*  $k > 1$  *then go to next scan*



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος - Διάγραμμα ροής





# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 1/8

### Άσκηση

Να γράψετε σε μορφή ψευδοκώδικα τον αλγόριθμο που υπολογίζει την παρακάτω συνάρτηση

$$F(n) = \begin{cases} 0 & n \leq 0 \\ \sum_{i=0}^n i(i+1) & n > 0 \end{cases}$$



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 2/8

### Άσκηση (τι κάνει ο αλγόριθμος;)

Array integer A(10)

Integer I, J, X

Repeat ( I = 2 έως 10)

J=10

While (J >= 1)

do { if A(J-1) > A(J)

then { X = A(J-1)

A(J-1) = A(J)

A(J) = X }

J = J-1 }

end while do

print A

end repeat



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 3/8

### Παραδείγματα δομών δεδομένων

Δομή δεδομένων είναι ένα σύνολο οργανωμένων δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών (πράξεων) επί αυτών.

- Πρωτογενείς δομές δεδομένων: ακέραιοι, πραγματικοί, μιγαδικοί, λογικοί, συμβολοσειρές.
- Απλές – μη πρωτογενείς δομές δεδομένων: Πίνακες

### Παραδείγματα βασικών λειτουργιών δομών δεδομένων

- Προσπέλαση, Εισαγωγή, Διαγραφή, Αναζήτηση, Ταξινόμηση, Αντιγραφή, Συγχώνευση, Διαχωρισμός.

### Στατικές - δυναμικές δομές. Διαφορές

- Στις στατικές δομές το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά την μετάφραση και όχι κατά την εκτέλεση του προγράμματος.
- Τα στοιχεία των στατικών δομών αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 4/8

- Παραδείγματα σύνθετων στατικών δομών δεδομένων και λειτουργιών (πράξεων) τους
  - Εγγραφή (αποτελείται από πεδία που αποθηκεύουν χαρακτηριστικά διαφορετικού τύπου) –  
επιτρεπτή λειτουργία: διαγραφή, αντιγραφή, κλπ.
- **Εγγραφή**(record, structure)  
Σύνολο ανόμοιων δεδομένων:  
structure **spoudastis**{  
    char onoma[20]  
    char eponimo[20]  
    integer arithmos-mitroou  
    integer vathmos-theoria  
    integer vathmos-ergastirio  
}





# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 5/8

```
structure spoudastis{  
    char onoma[20]  
    char eponimo[20]  
    integer arithmos-mitroou  
    integer vathmos-theoria  
    integer vathmos-ergastirio  
}
```

Δήλωση: spoudastis

stud1

Τιμές:

stud1.onoma= “ΘΕΟΔΩΡΟΣ”

stud1.eponimo= “ΤΣΙΛΙΓΚΙΡΙΔΗΣ”

stud1.arithmos-mitroou=19854

stud1.vathmos-ergastirio= 8

stud1.vathmos-theoria= 7



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 6/8

```
structure spoudastis{  
  char onoma[20]  
  char eponimo[20]  
  integer arithmos-mitroou  
  integer vathmos-theoria  
  integer vathmos-ergastirio  
}
```

Δήλωση: spoudastis

Τιμές:

**protoeteis[400]**

```
protoeteis[0].onoma= "ΚΩΣΤΑΝΤΙΝΟΣ"  
protoeteis[0].eponimo= "ΚΟΥΣΟΥΛΑΣ"  
protoeteis[0].vathmos-ergastirio= 9  
protoeteis[1].onoma= "ΓΕΩΡΓΙΟΣ"  
protoeteis[1].eponimo="ΑΝΑΣΤΑΣΙΟΥ"  
protoeteis[1].vathmos-theoria= 6
```



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 7/8

**Παραδείγματα στατικών δομών δεδομένων και λειτουργιών (πράξεων) τους**

– Αρχείο (σύνολο εγγραφών) –

επιτρεπτή λειτουργία: σειραϊκή προσπέλαση.

– Πίνακες –

επιτρεπτές λειτουργίες: πράξεις πινάκων



# Ανάπτυξη Προγράμματος

## Βήμα 2<sup>ο</sup>: Αλγόριθμος – Παραδείγματα 8/8

### Παραδείγματα δυναμικών δομών δεδομένων και λειτουργιών (πράξεων) τους

- Στοίβα – επιτρεπτές λειτουργίες:
  - ώθηση (στοιχείου στην κορυφή της στοίβας)
  - απώθηση (στοιχείου από τη στοίβα)
- Ουρά – επιτρεπτές λειτουργίες:
  - εισαγωγή (στοιχείου στο πίσω άκρο της ουράς)
  - εξαγωγή (στοιχείου από την ουρά)
- Λίστες
- Δένδρα
- Γράφοι



# Ανάπτυξη Προγράμματος

## Βήμα 3<sup>ο</sup> - Κωδικοποίηση

### Κωδικοποίηση

Διαδικασία γραφής των οδηγιών - εντολών του προγράμματος.

- Ποιότητα προγράμματος
- Επιλογή γλώσσας

Διεθνή πρότυπα ANSI

Κατάλληλη για το πρόβλημα

Διεπαφές με άλλες γλώσσες

Φορητή σε άλλα συστήματα

Ισχυρές βιβλιοθήκες



# Ανάπτυξη Προγράμματος

## Βήμα 4<sup>ο</sup> – Διορθώσεις

### Διορθώσεις (Debugging)

Εκτέλεση του προγράμματος και διόρθωση των συντακτικών και λογικών λαθών

- Επανελέγχος (χωρίς τον υπολογιστή)
- Έλεγχος με δεδομένα (χωρίς τον υπολογιστή)
- Μεταφραστικά λάθη (μέσω του υπολογιστή)
- Έλεγχος με δεδομένα (μέσω του υπολογιστή)
- Beta έλεγχος

Πρόγραμμα διορθωτής (Debugger)



# Ανάπτυξη Προγράμματος

## Βήμα 5<sup>ο</sup> -Τεκμηρίωση

### Τεκμηρίωση

Περιγραφή του στόχου, των διαδικασιών και του τρόπου χρήσεως του προγράμματος.

- Χρήστες
- Χειριστές
- Προγραμματιστές



# Ανάπτυξη Προγράμματος

## Βήμα 5<sup>ο</sup> - Συντήρηση

### Συντήρηση

Διατήρηση της λειτουργικότητας,  
παραγωγικότητας και βέλτιστης  
χρησιμοποίησης του προγράμματος

- Λειτουργία του προγράμματος χωρίς λάθη
- Παραγωγικότητα αποδοτική - αποτελεσματική
- Ενημερώσεις - Επεκτάσεις - Τυποποιήσεις





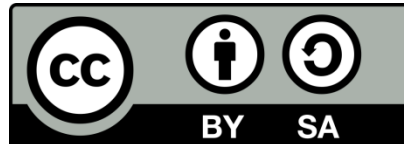
# Βιβλιογραφία

- Α. Σιδερίδης, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Αθήνα.
- J. Glenn Brookshearr, *Η επιστήμη των Υπολογιστών - Μια Ολοκληρωμένη Παρουσίαση*, Κλειδάριθμος.
- L. Goldschlager, A. M. Lister, *Εισαγωγή στη Σύγχρονη Επιστήμη των Υπολογιστών*, Δίαυλος
- Behrouz Forouzan, Firouz Mosharrafa, *Εισαγωγή στην Επιστήμη των Υπολογιστών - 2η Αγγλική Έκδοση - Επιμέλεια: Γιώργος Στεφανίδης, Αλέξανδρος Χατζηγεωργίου*. Κλειδάριθμος.



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδεια χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα Γεωπονικού Πανεπιστημίου 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.





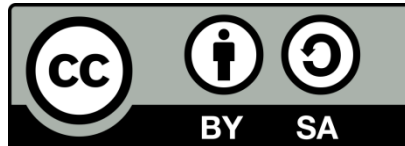
# Σημείωμα Αναφοράς

- Copyright Γεωπονικό Πανεπιστήμιο Αθηνών. Τμήμα Αγροτικής Οικονομίας και Ανάπτυξης, Θεόδωρος Τσιλιγκιρίδης, «Εισαγωγή στην Επιστήμη των Υπολογιστών». Έκδοση: 1.0. Αθήνα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<https://oceclass.aua.gr/courses/OCDAERD111/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, 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».



Η άδεια αυτή ανήκει στις άδειες που ακολουθούν τις προδιαγραφές του Ορισμού Ανοικτής Γνώσης [2], είναι ανοικτό πολιτιστικό έργο [3] και για το λόγο αυτό αποτελεί ανοικτό περιεχόμενο [4].

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

[2] <http://opendefinition.org/okd/ellinika/>

[3] <http://freedomdefined.org/Definition/EI>

[4] <http://opendefinition.org/buttons/>



# 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.