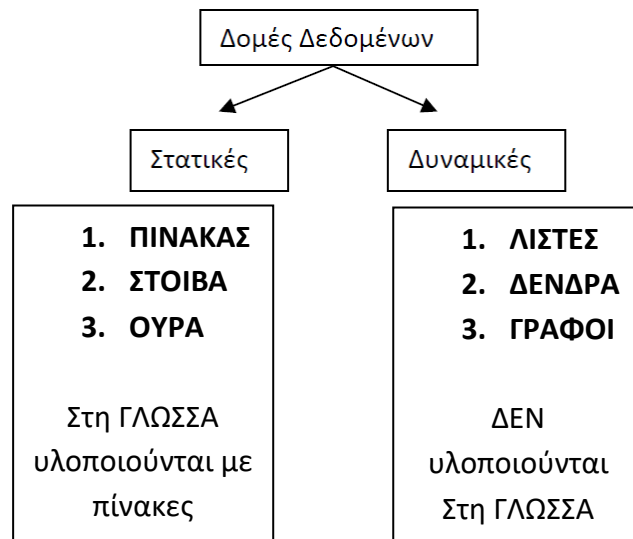


ΟΡΙΣΜΟΣ

Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

- Οι **βασικές λειτουργίες** (ή αλλιώς πράξεις) **επί των δομών δεδομένων** είναι οι:
 1. Προσπέλαση
 2. Εισαγωγή
 3. Διαγραφή
 4. Αναζήτηση
 5. Ταξινόμηση
 6. Αντιγραφή
 7. Συγχώνευση
 8. Διαχωρισμός
- Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα
- Οι δομές δεδομένων διακρίνονται σε δύο μεγάλες κατηγορίες: τις **στατικές** και τις **δυναμικές**.

ΣΤΑΤΙΚΕΣ ΚΑΙ ΔΥΝΑΜΙΚΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ



Διαφορές – Χαρακτηριστικά :

- Οι **στατικές** δομές έχουν **σταθερό μέγεθος** που δηλώνεται στην αρχή του προγράμματος ενώ οι **δυναμικές** δομές **δεν έχουν σταθερό μέγεθος**, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει κατά την εκτέλεση τους προγράμματος.
 - Τα **στοιχεία** των **στατικών** δομών αποθηκεύονται σε **συνεχόμενες θέσεις μνήμης** ενώ τα στοιχεία των **δυναμικών** δομών αποθηκεύονται σε **τυχαίες θέσεις μνήμης** σύμφωνα με την **τεχνική της δυναμικής παραχώρησης μνήμης**.
- Πίνακας είναι μια δομή δεδομένων που περιέχει στοιχεία του ίδιου τύπου.**
Η αναφορά στα στοιχεία ενός πίνακα γίνεται με το **όνομα του πίνακα** και την **τιμή** ενός ή περισσότερων **δεικτών** σε αγκύλη. Π.χ. **ΠΙΝ[2,5]**

ΟΡΙΣΜΟΣ

Πίνακας είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Κάθε ένα από τα αντικείμενα που απαρτίζουν τον πίνακα λέγεται στοιχείο του πίνακα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από ένα δείκτη.

- **Τυπικές επεξεργασίες πινάκων:**

1. Υπολογισμός αθροισμάτων στοιχείων του πίνακα.
2. Εύρεση του μέγιστου ή του ελάχιστου στοιχείου.
3. Ταξινόμηση των στοιχείων του πίνακα.
4. Αναζήτηση ενός στοιχείου του πίνακα.
5. Συγχώνευση δύο πινάκων.

- **Πλεονεκτήματα χρήσης Πινάκων:**

Οι πίνακες χρησιμοποιούνται στην περίπτωση που χρειάζεται να διαχειριστούμε **πολλά δεδομένα του ίδιου τύπου** και τα οποία απαιτείται να είναι **αποθηκευμένα για μεταγενέστερη χρήση** τους.

- **Μειονεκτήματα χρήσης Πινάκων:**

Οι πίνακες, από την αρχή του προγράμματος, **δεσμεύουν συγκεκριμένο πλήθος θέσεων μνήμης** - ανάλογα με το μέγεθός τους- και **περιορίζουν τις δυνατότητες του προγράμματος**, καθώς το μέγεθός τους είναι προκαθορισμένο (δε μεταβάλλεται).

- **Παράλληλοι πίνακες**

Οι πίνακες αυτοί συνδέονται μεταξύ τους με τέτοιο τρόπο, ώστε η **κάθε εγγραφή να έχει όλα τα στοιχεία της με την ίδια τιμή δείκτη**, δηλ. να βρίσκονται **στην ίδια θέση στους πίνακες**.

- **Δύο είναι οι πλέον διαδεδομένοι αλγόριθμοι αναζήτησης:**

1. Η σειριακή ή γραμμική αναζήτηση
2. Η δυαδική αναζήτηση

- **Η σειριακή αναζήτηση είναι η πιο απλή, αλλά και η λιγότερο αποτελεσματική μέθοδος.**

Η χρήση της δικαιολογείται μόνο στις παρακάτω περιπτώσεις:

1. Ο πίνακας δεν είναι ταξινομημένος.
2. Ο πίνακας έχει μικρό μέγεθος (συνήθως $n \leq 20$).
3. Η αναζήτηση στον πίνακα γίνεται σπάνια.

- **Η δυαδική αναζήτηση είναι αποτελεσματικότερη από τη σειριακή**, διότι χρειάζεται **πολύ λιγότερο χρόνο εκτέλεσης**- κάτω από τον μισό χρόνο- από ότι η σειριακή αναζήτηση.

Απαραίτητη προϋπόθεση είναι ότι **ο πίνακας πρέπει να είναι ταξινομημένος**.

Η μέθοδος αυτή στηρίζεται στη συνεχή διαίρεση του πίνακα. «**ΜΕΘΟΔΟΣ ΔΙΑΙΡΕΙ ΚΑΙ ΒΑΣΙΛΕΥΕ**»

- **Ταξινόμηση στοιχείων πινάκα**

Ταξινόμηση ονομάζουμε την τακτοποίηση των κόμβων μιας δομής με μια ιδιαίτερη σειρά.

1. κατ' αύξουσα σειρά (δηλ. από τον μικρότερο προς το μεγαλύτερο)
Αλφαβητική σειρά = Αύξουσα σειρά
2. κατά φθίνουσα σειρά (δηλ. από τον μεγαλύτερο προς το μικρότερο)

- **Αλγόριθμοι ταξινόμησης πίνακα:**

- ✓ Οι σχετικά απλοί αλγόριθμοι είναι η **ταξινόμηση με επιλογή** και η ταξινόμηση με παρεμβολή.
- ✓ Ο πιο γρήγορος αλγόριθμος ταξινόμησης είναι η "γρήγορη ταξινόμηση" (quicksort).
- ✓ Η **ταξινόμηση φυσαλίδας** είναι ο πιο απλός και ταυτόχρονα ο πιο αργός αλγόριθμος ταξινόμησης.

- **Ταξινόμηση ευθείας ανταλλαγής ή φυσαλίδας (Bubblesort)**, Με τη μέθοδο αυτή της ταξινόμησης συγκρίνουμε και ανταλλάσσουμε ζεύγη γειτονικών στοιχείων μέχρι να ταξινομηθούν όλα τα στοιχεία.

-

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (SELECTION SORT)

Η ταξινόμηση με επιλογή (selection sort), αποτελεί βασικό τρόπο ταξινόμησης, που υλοποιείται σε έναν μονοδιάστατο πίνακα σε τρία βήματα:

1. Επιλογή του ελάχιστου στοιχείου
2. Ανταλλαγή του ελάχιστου με το πρώτο στοιχείο
3. Επανάληψη των βημάτων 1 και 2 για τα υπόλοιπα στοιχεία του πίνακα

```
1  ΠΡΟΓΡΑΜΜΑ Selection_Sort
2  ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: A[20], k, x, i, j
4  ΑΡΧΗ
5      ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
6          ΓΡΑΨΕ 'Δώσε το ', i, ' στοιχείο του πίνακα'
7          ΔΙΑΒΑΣΕ A[i]
8      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
9      ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 19
10         k <- i
11         x <- A[i]
12         ΓΙΑ j ΑΠΟ i + 1 ΜΕΧΡΙ 20
13             ΑΝ x > A[j] ΤΟΤΕ
14                 k <- j
15                 x <- A[j]
16             ΤΕΛΟΣ_ΑΝ
17         ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
18         A[k] <- A[i]
19         A[i] <- x
20     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
21     ΓΡΑΨΕ 'Εκτύπωση με ταξινομημένα τα στοιχεία'
22     ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
23         ΓΡΑΨΕ A[i]
24     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
25 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΑΣΚΗΣΗ 1

Σε έναν αγώνα δισκοβολίας συμμετέχουν 20 αθλητές. Κάθε αθλητής έκανε μόνο μια έγκυρη προσπάθεια που καταχωρείται σαν επίδοση του αθλητή και εκφράζεται σε μέτρα. Να αναπτύξετε πρόγραμμα το οποίο:

1. Να διαβάσει για κάθε αθλητή το όνομα και την επίδοση του και να τα καταχωρεί σε κατάλληλους πίνακες, ελέγχοντας ότι η επίδοση είναι θετικός αριθμός.
2. Να ταξινομεί τους αθλητές ως προς την επίδοσή τους.
3. να εμφανίζει τα ονόματα και τις επιδόσεις των 3 πρώτων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση.
4. Να εμφανίζει τα ονόματα και τις επιδόσεις των 5 τελευταίων αθλητών, αρχίζοντας από εκείνον με την καλύτερη επίδοση.

Σημείωση: Να θεωρήσετε ότι δεν υπάρχουν αθλητές με την ίδια επίδοση.

ΠΡΟΓΡΑΜΜΑ A1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j

ΠΡΑΓΜΑΤΙΚΕΣ: ΕΠ[20], T1

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[20], T2

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20

ΔΙΑΒΑΣΕ ΟΝ[i]

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΕΠ[i]

ΜΕΧΡΙΣ_ΟΤΟΥ ΕΠ[i] > 0

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 20

ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ ΕΠ[j - 1] < ΕΠ[j] ΤΟΤΕ

T1 ← ΕΠ[j - 1]

ΕΠ[j - 1] ← ΕΠ[j]

ΕΠ[j] ← T1

T2 ← ΟΝ[j - 1]

ΟΝ[j - 1] ← ΟΝ[j]

ΟΝ[j] ← T2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3

ΓΡΑΨΕ ΟΝ[i], ΕΠ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 16 ΜΕΧΡΙ 20

ΓΡΑΨΕ ΟΝ[i], ΕΠ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Αλγόριθμος ταξινόμησης με επιλογή

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 19

k ← i

x ← ΕΠ[i]

Ψ ← ΟΝ[i]

ΓΙΑ j ΑΠΟ i + 1 ΜΕΧΡΙ 20

ΑΝ x < ΕΠ[j] ΤΟΤΕ

k ← j

x ← ΕΠ[j]

Ψ ← ΟΝ[j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΕΠ[k] ← ΕΠ[i]

ΕΠ[i] ← x

ΟΝ[k] ← ΟΝ[i]

ΟΝ[i] ← Ψ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΠΑΡΑΤΗΡΗΣΕΙΣ:

1. Η δήλωση του πίνακα στο πρόγραμμα γίνεται στον τύπο δεδομένων που έχει στα στοιχεία του και γίνεται με το όνομά του και με τον αριθμό των κελιών που έχει.
2. Κάνουμε ταξινόμηση στον πίνακα των επιδόσεων αλλά **κρατάμε συσχέτιση** με τον πίνακα των ονομάτων. Αυτό γίνεται γιατί αν δεν αλλάξουμε και τις θέσεις των ονομάτων δεν θα αντιστοιχεί η i επίδοση στο i όνομα.
3. Εμφανίζουμε τους πρώτους 3 με μια επανάληψη από 1 μέχρι 3 και τους τελευταίους 5 με μια αντίστοιχη επανάληψη από 16 μέχρι 20.

ΑΡΧΕΙΟ

ΕΓΓΡΑΦΗ				
ΠΕΔΙΟ/ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΠΕΔΙΟ/ΑΦΜ	ΠΕΔΙΟ/ΔΙΕΥΘΥΝΣΗ	ΠΕΔΙΟ/ΤΗΛΕΦΩΝΟ	ΠΕΔΙΟ/Α.Δ.Τ.
ΑΓΡΙΟΠΟΥΛΟΥ ΕΛΕΝΗ	036662287	ΚΑΛΛΙΚΡΑΤΟΥΣ 1	6974796909	A0345678

- Τα **αρχεία** (files) είναι ειδικές δομές για την αποθήκευση των δεδομένων στη δευτερεύουσα μνήμη, δηλαδή κυρίως στο μαγνητικό δίσκο, όπου διατηρούνται ακόμη και μετά τον τερματισμό ενός προγράμματος.
- Τα στοιχεία ενός αρχείου ονομάζονται **εγγραφές** (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα **πεδία** (fields), που **ταυτοποιούν** την εγγραφή, και από άλλα πεδία που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής.
- Το πεδίο που **ταυτοποιεί** την εγγραφή και ονομάζεται **πρωτεύον κλειδί** (ΑΦΜ) ή απλά κλειδί.
- Το πρόβλημα της **αναζήτησης μίας εγγραφής** σε αρχεία γίνεται **με βάση την τιμή του πρωτεύοντος** ή ενός δευτερεύοντος **κλειδιού**.