

## ΚΕΦΑΛΑΙΟ 1 Ανάλυση Προβλήματος - Δομή Προβλήματος

**Πρόβλημα:** Μία κατάσταση που χρειάζεται αντιμετώπιση, απαιτεί λύση. Η λύση της δεν είναι γνωστή ή προφανής.

**Κατανόηση Προβλήματος :**→ Σαφήνεια διατύπωσης από τον δημιουργό.  
→ Σωστή ερμηνεία από αυτόν που θα το λύσει.

**Επίλυση:** Η διαδικασία μέσω της οποίας βρίσκουμε το ζητούμενο

**Δεδομένο:** Οποιοδήποτε στοιχείο γίνεται αντιληπτό από ένα παρατηρητή με μία από τις πέντε αισθήσεις του.

**Πληροφορία:** Οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.

**Επεξεργασία Δεδομένων:** Η διαδικασία που ένας "μηχανισμός" δέχεται δεδομένα, τα επεξεργάζεται με ένα προκαθορισμένο τρόπο και αποδίδει πληροφορίες.

**Δομή Προβλήματος:** Τα συστατικά μέρη του προβλήματος, τα επιμέρους τμήματα που το αποτελούν και ο τρόπος που συνδέονται μεταξύ τους.

**Ανάλυση Προβλήματος:**

- **Φραστικά :** περιγράφουμε με λόγια πως και σε ποια επιμέρους προβλήματα αναλύεται το κυρίως πρόβλημα και τα υποπροβλήματα του.
- **Διαγραμματική αναπαράσταση :** περιγράφουμε με ένα διάγραμμα σε σχήμα γενεαλογικού δένδρου. Κάθε πρόβλημα έχει παιδιά τα υποπροβλήματα στα οποία αναλύεται.

**Στάδια Αντιμέτωπισης Προβλήματος :**

- **Κατανόηση :** Απαιτείται η σωστή και πλήρης αποσαφήνιση των δεδομένων και των ζητούμενων.
- **Ανάλυση :** Το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα.
- **Επίλυση :** Υλοποιείται η λύση μέσω της επίλυσης και σύνθεσης των επιμέρους προβλημάτων.

**Κατηγοριοποίηση Προβλημάτων :**

- **Δυνατότητα επίλυσης :**
  - **Επιλύσιμα :** Η λύση τους είναι ήδη γνωστή ή η συνάφειά τους με άλλα επιλυμένα μας επιτρέπει να θεωρούμε σαν βέβαιη την επίλυσή τους
  - **Ανοικτά :** Δεν έχει βρεθεί λύση αλλά και δεν έχει αποδειχθεί η μη ύπαρξη λύσης τους.
  - **Άλυτα :** Έχει αποδειχθεί ότι δεν υπάρχει λύση.
- **Βαθμός Δόμησης :**
  - **Δομημένα :** Η λύση προέρχεται από μία αυτοματοποιημένη λύση.
  - **Ημιδομημένα :** Η λύση μπορεί να επιλεγεί ανάμεσα από ένα εύρος δυνατών λύσεων.

- **Αδόμητα** : Οι λύσεις δεν μπορούν να δομηθούν ή δεν έχει διερευνηθεί η δυνατότητα δόμησης τους.
- **Είδος επίλυσης :**
  - **Απόφασης** : Η λύση του απαντά σε κάποιο ερώτημα.
  - **Υπολογιστικά** : Η απάντηση απαιτεί την διενέργεια υπολογισμών.
  - **Βελτιστοποίησης** : Η λύση πρέπει να είναι το βέλτιστο αποτέλεσμα για τα δεδομένα που διαθέτουμε.

#### Λόγοι ανάθεσης σε υπολογιστή της επίλυση ενός προβλήματος :

- Πολυπλοκότητα υπολογισμών
- Επαναληπτικότητα Διαδικασιών
- Ταχύτητα Εκτέλεσης Πράξεων
- Μεγάλο Πλήθος Δεδομένων

#### Βασικές Λειτουργίες που Εκτελεί ο Υπολογιστής :

- **Πρόσθεση** (βασική αριθμητική πράξη - οι άλλες πράξεις είναι διαδικασίες πρόσθεσης)
- **Σύγκριση** (εκτέλεση όλων των λογικών πράξεων)
- **Μεταφορά Δεδομένων** (προηγείται και έπεται της επεξεργασίας δεδομένων)

## ΚΕΦΑΛΑΙΟ 2 Βασικές Έννοιες Αλγορίθμων

**Αλγόριθμος** : Μία πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

#### Κριτήρια αλγόριθμου :

- **Είσοδος** τιμές δεδομένων που δίνονται στον αλγόριθμο (από καμία έως πολλές).
- **Έξοδος** τουλάχιστον μία τιμή δεδομένου που δημιουργεί ο αλγόριθμος ως αποτέλεσμα προς τον χρήστη ή άλλο αλγόριθμο
- **Καθοριστικότητα** κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.
- **Περατότητα** ο αλγόριθμος πρέπει να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών της
- **Αποτελεσματικότητα** κάθε εντολή να είναι απλή, να είναι εκτελέσιμη.

#### Σκοπιές που μελετώνται οι αλγόριθμοι :

**Υλικού** (hardware) - μελέτη τεχνολογίας του υπολογιστή - ταχύτητα εκτέλεσης ενός αλγορίθμου σε σχέση με το πως είναι δομημένα τα συστατικά του υπολογιστή

**Γλώσσας προγραμματισμού** εξετάζει τη δομή και των αριθμό των εντολών ενός αλγορίθμου ανάλογα με τη γλώσσα

**Θεωρητική** - αν υπάρχει αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος

**Αναλυτική** - (computer resources) εξετάζει την ανάγκη σε υπολογιστικούς πόρους

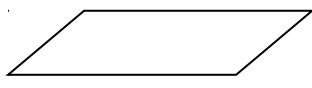
### Περιγραφή και αναπαράσταση αλγορίθμων :

**Ελεύθερο κείμενο** ο πιο ανεπεξέργαστος και αδόμετος τρόπος παρουσίασης. Μπορεί να οδηγήσει σε μη εκτελέσιμη μορφή κώδικα - μπορεί να παραβιασθεί το χαρακτηριστικό της αποτελεσματικότητας

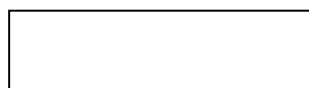
**Διαγραμματική Τεχνική** γραφικός τρόπος παρουσίασης του αλγόριθμου. Η πιο γνωστή είναι το διάγραμμα ροής (flow chart) όπου κάθε ενέργεια αναπαρίσταται με την χρήση ειδικού γεωμετρικού σχεδίου



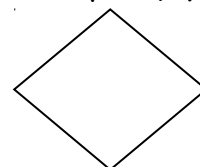
Αρχή / Τέλος



Είσοδος / Έξοδος



Εκτέλεση Πράξεων



Επιλογή

• **Φυσική γλώσσα** κατά βήματα - κάθε εντολή είναι ξεχωριστή παράγραφος - μπορεί να παραβιασθεί το χαρακτηριστικό του καθορισμού

• **Κωδικοποίηση** με ένα πρόγραμμα

### **Δεσμευμένη λέξη**

-σε γλώσσα προγραμματισμού : έχει αυστηρά προκαθορισμένη σημασία, χρησιμοποιείται κυρίως σαν εντολή

-σε ψευδογλώσσα : κάνει πιο ομοιόμορφη τη γλώσσα του κώδικα και περιγράφει τις εντολές της ψευδογλώσσας

### Στοιχεία ψευδογλώσσας (και όχι μόνο):

- **Σταθερές** - προκαθορισμένες τιμές που παραμένουν αμετάβλητες σε όλη την διάρκεια εκτέλεσης ενός αλγόριθμου
  - Αριθμητικές : αριθμοί, + , - , (ως δεκαδικό σημείο)
  - Αλφαριθμητικές : χαρακτήρες μέσα σε διπλά εισαγωγικά
  - Λογικές : Αληθής και Ψευδής
- **Μεταβλητές** - ένα γλωσσικό αντικείμενο που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου. Η τιμή του μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του αλγόριθμου
  - Αριθμητικές - παίρνουν μόνο αριθμούς
  - Αλφαριθμητικές - παίρνουν γράμματα ή / και αριθμούς
  - Λογικές : Αληθής και Ψευδής
- **Τελεστές**
  - Αριθμητικοί : + - \* / ^
  - Συγκριτικοί : < , ≤ , <= , > , ≥ , >= , = , ≠ , <>
  - Λογικοί : και , ή , όχι
- **Εκφράσεις** - διαμορφώνονται από σταθερές και μεταβλητές, συναρτήσεις, τελεστές και παρενθέσεις
- **Εντολή εκχώρησης** : μεταβλητή ← έκφραση
- **Ρήματα σε προστατική** π.χ. "Διάβασε", "Γράψε", "Εκτέλεσε" κ.λπ.

- **Ουσιαστικά** όταν οι ζητούμενες ενέργειες είναι πολλές ή προφανείς, καθορίζονται με τη χρήση ουσιαστικών αντί ρημάτων π.χ. "Εισαγωγή δεδομένων", "εμφάνιση πεδίων στην οθόνη" κ.λπ.
- **Σχόλια** με το σύμβολο ! στην αρχή της γραμμής διαχωρίζονται τα σχόλια από τις λέξεις-κλειδιά του αλγορίθμου.
- **Πρώτη και τελευταία γραμμή** ενός αλγορίθμου είναι  
**Αλγόριθμος** <όνομα\_αλγόριθμου>  
.....  
**Τέλος** <όνομα\_αλγόριθμου>
- **Δεδομένα και αποτελέσματα** Εντός των συμβόλων //...// στην δεύτερη γραμμή του αλγορίθμου περιγράφονται τα δεδομένα εισόδου, και στη προτελευταία γραμμή τα αποτελέσματα εξόδου.

### ΚΕΦΑΛΑΙΟ 3 Δομές Δεδομένων και Αλγόριθμοι

**Πληροφορία** η συλλογή ακατέργαστων δεδομένων και ο συσχετισμός τους δίνει ως αποτέλεσμα πληροφορίες.

Με βάση αυτές τις πληροφορίες λαμβάνονται διάφορες αποφάσεις και γίνονται ενέργειες που παράγουν νέα δεδομένα, νέες πληροφορίες, νέες αποφάσεις κ.ο.κ.

Η Πληροφορική μελετά τα δεδομένα από τις ακόλουθες σκοπιές:

- **Υλικού** η μηχανή επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στη κύρια μνήμη και στις περιφερειακές του συσκευές (με δυαδική μορφή, ASCII, EBCDIC κλπ)
- **Γλωσσών προγραμματισμού** χρησιμοποιούν μεταβλητές για να περιγράψουν τα δεδομένα. Ο μεταφραστής κάθε γλώσσας φροντίζει για την αποδοτικότερη μορφή αποθήκευσης κάθε μεταβλητής στον υπολογιστή.
- **Δομές Δεδομένων** (data structure) ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Αρχείο αποτελείται από εγγραφές - records, αποτελείται από πεδία - fields που αποθηκεύουν χαρακτηριστικά.
- **Ανάλυση δεδομένων** τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων για να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Βάσεις Δεδομένων - Μοντελοποίηση Δεδομένων κλπ.

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

**Κόμβος** κάθε μορφή δομής δεδομένων.

### **Βασικές λειτουργίες:**

- **Προσπέλαση** πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
- **Εισαγωγή** προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.
- **Διαγραφή** ένας κόμβος αφαιρείται από μία δομή.
- **Αναζήτηση** προσπελαύνονται οι κόμβοι μίας δομής προκειμένου να εντοπισθούν ένας ή περισσότεροι που έχουν μία συγκεκριμένη ιδιότητα.
- **Ταξινόμηση** οι κόμβοι μίας δομής διατάσσονται σε αύξουσα ή φθίνουσα σειρά.
- **Αντιγραφή** όλοι ή μερικοί κόμβοι μίας δομής αντιγράφονται σε μία άλλη δομή.
- **Συγχώνευση** δύο ή περισσότερες δομές συγχωνεύονται σε μία ενιαία δομή.
- **Διαχωρισμός** η αντίστροφη πράξη της συγχώνευσης.

**Δυναμική δομή δεδομένων** οι δομές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται δεδομένα.

**Στατική δομή δεδομένων** το ακριβές μέγεθος της απαιτούμενης κυρίας μνήμης καθορίζεται κατά τη στιγμή του προγραμματισμού.

**Πίνακας** μία δομή που περιέχει στοιχεία του ίδιου τύπου.

**LIFO - Στοιίβα** μία δομή δεδομένων με ένα άκρο, που υλοποιείται με την βοήθεια πινάκων. Το τελευταίο στοιχείο που εισάγεται είναι το πρώτο που μπορεί να εξαχθεί.

Λειτουργίες - **ώθηση** στοιχείου στη κορυφή της στοιίβας. Πριν από την ώθηση θα πρέπει να γίνεται έλεγχος για **υπερχείλιση** ότι δηλ. η στοιίβα δεν είναι γεμάτη.

- **απώθηση** στοιχείου από την στοιίβα. Πριν από την απώθηση θα πρέπει να γίνεται έλεγχος για **υποχείλιση** ότι δηλ. υπάρχει στην στοιίβα τουλάχιστον ένα στοιχείο.

**FIFO - Ουρά** μία δομή δεδομένων με δύο άκρα στην οποία το πρώτο στοιχείο που εισάγεται είναι το πρώτο που μπορεί να εξαχθεί.

Λειτουργίες - **εισαγωγή** στοιχείου στο πίσω άκρο της ουράς

- **εξαγωγή** στοιχείου από το εμπρός άκρο της ουράς.

Χρησιμοποιούνται δύο δείκτες ο **εμπρός**, που μας δείχνει την θέση του στοιχείου που θα εξαχθεί πρώτο και ο **πίσω**, που μας δείχνει την θέση του στοιχείου εισήλθε τελευταίο.

**Αναζήτηση** σε πίνακα

- **Σειριακή ή γραμμική** - γίνεται σε στοιχείο προς στοιχείο του πίνακα μέχρι να βρεθεί αυτό που αναζητάτε. Είναι η πιο απλή και η λιγότερο αποτελεσματική μέθοδος. Επιλέγεται όταν ο πίνακας είναι :
  - Μη ταξινομημένος.
  - Μικρού μεγέθους ( $\leq 20$ ).
  - Η αναζήτηση στον πίνακα γίνεται σπάνια.

**Ταξινόμηση** γίνεται η τοποθέτηση των στοιχείων ενός πίνακα σε αύξουσα ή φθίνουσα σειρά ως προς την τιμή των μεγεθών των στοιχείων του πίνακα.

**Ταξινόμηση ευθείας ανταλλαγής** βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων μέχρις ότου διαταχθούν όλα τα στοιχεία.

## ΚΕΦΑΛΑΙΟ 4 Τεχνικές Σχεδίασης Αλγορίθμων

Η ανάλυση ενός προβλήματος περιλαμβάνει :

- Καταγραφή της υπάρχουσας πληροφορίας για το πρόβλημα
- Αναγνώριση των ιδιαιτεροτήτων του προβλήματος
- Αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησής του
- Πρόταση επίλυσης με χρήση κάποιας μεθόδου
- Τελική επίλυση με χρήση υπολογιστικών συστημάτων

Κατά την ανάλυση πρέπει να δοθεί απάντηση σε κάθε μία από τις ερωτήσεις :

- Ποια είναι τα δεδομένα και το μέγεθος του προβλήματος
- Ποιες είναι οι συνθήκες που πρέπει να πληρούνται για την επίλυση του προβλήματος
- Ποια είναι η πλέον αποδοτική μέθοδος επίλυσής τους
- Πως θα καταγραφεί η λύση σε ένα πρόβλημα
- Ποιος είναι ο τρόπος υλοποίησης στο συγκεκριμένο υπολογιστικό σύστημα (επιλογή γλώσσας)

Κάθε τεχνική έχει τα δικά της χαρακτηριστικά και τις δικές της ιδιαιτερότητες και χρειάζεται να υποστηρίζει τα εξής:

- να αντιμετωπίζει με τα δικά της τρόπο τα δεδομένα,
- να έχει τη δική της ακολουθία εντολών και
- να διαθέτει τη δική της αποδοτικότητα.

Τυποποιημένες κατηγορίες τεχνικών :

- Μέθοδος διαίρει και βασίλευε
- Μέθοδος δυναμικού προγραμματισμού
- Άπληστη μέθοδος

Ευριστικές τεχνικές : όταν για την επίλυση ενός προβλήματος απαιτείται η εφαρμογή μίας νέας αντίληψης

## ΚΕΦΑΛΑΙΟ 6 Εισαγωγή στον Προγραμματισμό

Η επίλυση ενός προβλήματος με χρήση υπολογιστή περιλαμβάνει :

- Ακριβή προσδιορισμό του προβλήματος
- Ανάπτυξη του αντίστοιχου αλγόριθμου
- Διατύπωση του αλγόριθμου σε κατανοητή μορφή από τον υπολογιστή

Πρόγραμμα ή αλγόριθμος είναι το σύνολο των εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί η επίλυση του προβλήματος.

Γλώσσα μηχανής είναι ακολουθίες από 0 και 1, τις οποίες εκτελεί ο υπολογιστής.

Συμβολική γλώσσα ή γλώσσα χαμηλού επιπέδου γλώσσα προγραμματισμού που ενώ έχει έννοια για τον άνθρωπο, μετατρέπεται εσωτερικά από τον υπολογιστή στις αντίστοιχες ακολουθίες από 0 και 1.

- Παραμένουν στενά συνδεδεμένες με την αρχιτεκτονική του κάθε υπολογιστή.
- Δεν διαθέτουν εντολές σύνθετων λειτουργιών - τα προγράμματα είναι δύσκολα στη γραφή και τη συντήρηση
- Δεν μπορούν να μεταφερθούν σε διαφορετικό υπολογιστή

Συμβολομεταφραστής (assembler) ειδικό πρόγραμμα που μεταφράζει τις εντολές που δίνουμε μέσω της γλώσσας προγραμματισμού σε εντολές γλώσσας μηχανής.

Γλώσσες υψηλού επιπέδου (3ης γενιάς) καλύτερη επικοινωνία ανθρώπου με μηχανή.

Το πρόγραμμα μεταφράζεται σε γλώσσα μηχανής με τη βοήθεια του μεταγλωττιστή.

- FORTRAN (FORmula TRANslation) το 1957 από τη IBM - η πρώτη γλώσσα υψηλού επιπέδου. Κατάλληλη για τη επίλυση μαθηματικών και επιστημονικών προβλημάτων.
- COBOL (COmmon Business Oriented Language) αναπτύχθηκε το 1960 από το Αμερικάνικο πολεμικό ναυτικό. Η καλύτερη γλώσσα στη διαχείριση πολλών και μεγάλων αρχείων, για αυτό και χρησιμοποιείται από πολλές επιχειρήσεις
- ALGOL (ALGOrithmic Language) - από Ευρωπαίους. Με ελάχιστη πρακτική εφαρμογή, που επηρέασε την ανάπτυξη των επόμενων γλωσσών.
- PL/1 (Programming Language/1) - δεκαετία του 60. Αποτυχημένη γλώσσα γιατί προσπάθησε να καλύψει όλους τους τομείς προγραμματισμού.
- LISP (LIST Processor) - μέσα δεκαετίας 60. Για χειρισμό λιστών από σύμβολα.
- PROLOG (PROgramming LOGic) αρχές δεκαετίας 70. Για τεχνική νοημοσύνη.
- BASIC (Beginner's All Purpose Symbolic Instruction Code) - δεκαετία του 60. Για τη εκπαίδευση αρχαρίων στον προγραμματισμό. Εξελίχθηκε στη δημοφιλέστερη γλώσσα για τους προσωπικούς υπολογιστές.
- PASCAL το 1970. Γενικής χρήσης, για εκπαίδευση αλλά και για ισχυρά προγράμματα. Πάρα πολύ δημοφιλής. Κατάλληλη για τη δημιουργία δομημένων προγραμμάτων.
- C το 1972 στα εργαστήρια της Bell. Ισχυρή γλώσσα για δομημένο προγραμματισμό, αλλά και δυνατότητες γλώσσας χαμηλού επιπέδου.

- JAVA από την SUN. Ειδική για το internet. Μπορεί να εκτελείται από διαφορετικούς υπολογιστές με διαφορετικά λειτουργικά συστήματα.

Δομημένος Προγραμματισμός απλά προγράμματα στη συγγραφή και κατανόηση, και εύκολα στη διόρθωση.

Αντικειμενοστραφής προγραμματισμός

Οδηγούμενος από το γεγονός προγραμματισμός (object driven programming) η δυνατότητα να ενεργοποιούνται λειτουργίες του προγράμματος με την εκτέλεση ενός γεγονότος π.χ. επιλογή μίας εντολής ή το κλικ του ποντικιού.

Οπτικός προγραμματισμός (visual programming) δυνατότητα να δημιουργούμε γραφικά ολόκληρο το περιβάλλον της εφαρμογής.

Πλεονεκτήματα των γλωσσών υψηλού επιπέδου

- Φυσικότερος τρόπος έκφρασης των προβλημάτων
- Ανεξαρτησία από τον τύπο του υπολογιστή. Μπορούν να εκτελεστούν σε οποιονδήποτε υπολογιστή με ελάχιστες ή καθόλου μετατροπές. (Μεταφερσιμότητα).
- Ευκολία στην εκμάθηση και εκπαίδευση της γλώσσας
- Ευκολότερη διόρθωση των λαθών και η συντήρηση των προγραμμάτων.

Ταξινόμηση γλωσσών προγραμματισμού

- Ως προς τις ιδέες προγραμματισμού και τις κατηγορίες προβλημάτων
- Διαδικαστικές (procedural) ή Αλγοριθμικές για την υλοποίηση αλγορίθμων.
- Αντικειμενοστραφείς (object oriented language)
- Συναρτησιακές (functional language) LISP
- Μη διαδικαστικές (non procedural) PROLOG
- Ερωταπαντήσεις (query languages) SQL
- Γενικής χρήσης
  - Επιστημονικής κατεύθυνσης FORTRAN
  - Εμπορικής κατεύθυνσης BASIC
- Προγραμματισμού συστημάτων (system programming) C
- Τεχνητής νοημοσύνης (artificial intelligence) LISP, PROLOG
- Ειδικής χρήσης για γραφικά, ρομποτική, σχεδίαση ολοκληρωμένων κυκλωμάτων κλπ.

Γλώσσες 4<sup>ης</sup> γενιάς γλώσσες που αποκρύπτουν πολλές λεπτομέρειες από τις τεχνικές αρχιτεκτονικής και υλοποίησης του αλγόριθμου και έτσι ο απλός χρήστης μπορεί να επιλύει μόνος του μικρά προβλήματα εφαρμογών

Φυσικές και Τεχνητές Γλώσσες φυσικές - οι γλώσσες που μιλάμε. Τεχνητές - οι γλώσσες προγραμματισμού

Αλφάβητο το σύνολο των στοιχείων που χρησιμοποιεί μία γλώσσα

Λεξιλόγιο οι λέξεις που είναι δεκτές από την γλώσσα

Γραμματική αποτελείται από :

- Τυπικό το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή



- Συντακτικό το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη δημιουργία προτάσεων

Σημασιολογία το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μία γλώσσα.

#### Διαφορές φυσικών και τεχνητών γλωσσών

Οι φυσικές γλώσσες εξελίσσονται επειδή είναι ζωντανές και συνεχώς εμπλουτίζονται από νέες λέξεις και κανόνες.

Οι τεχνητές χαρακτηρίζονται από στασιμότητα αν και κάποιες φορές μπορεί να βελτιωθούν για να διορθωθούν αδυναμίες ή να καλύψουν μεγαλύτερο εύρος εφαρμογών.

#### Τεχνικές σχεδίασης προγραμμάτων

- Ιεραρχική σχεδίαση είναι η διαδικασία σχεδίασης «από πάνω προς τα κάτω». Καθορίζονται οι βασικές λειτουργίες σε ανώτερο επίπεδο και μετά διασπώνται σε όλο και μικρότερες και πιο απλές λειτουργίες ώστε να επιλυθούν ευκολότερα.
- Τμηματικός προγραμματισμός κάθε υποπρόβλημα αποτελεί ανεξάρτητη ενότητα που γράφεται ξεχωριστά από τα υπόλοιπα τμήματα του προγράμματος.
  - Διευκολύνει το γράψιμο του προγράμματος
  - Μειώνει τα λάθη
  - Επιτρέπει ευκολότερη παρακολούθηση, κατανόηση και συντήρηση του προγράμματος
- Δομημένος προγραμματισμός ξεκίνησε για να αποφευχθεί η χρήση της εντολής GOTO. Είναι μία μεθοδολογία σύνταξης προγραμμάτων με τα εξής πλεονεκτήματα :
  - Δημιουργεί απλούστερα προγράμματα
  - Άμεση μεταφορά των αλγόριθμων σε προγράμματα
  - Διευκολύνει την ανάλυση του προγράμματος σε τμήματα
  - Βοηθάει στην ανάπτυξη σύνθετων προγραμμάτων
  - Μειώνει τα λάθη κατά τον προγραμματισμό
  - Εξασφαλίζει την εύκολη κατανόηση των προγραμμάτων
  - Διευκολύνει την ανάγνωση και κατανόηση του προγράμματος
  - Διευκολύνει τις διορθώσεις, αλλαγές, και συντήρηση.

Στηρίζεται στη χρήση των δομών ακολουθίας, επιλογής και επανάληψης.

Χρησιμοποιούνται μόνο αυτές οι δομές και οι συνδυασμοί τους. Κάθε πρόγραμμα και κάθε ενότητα έχει μόνο μία είσοδο και μία έξοδο.

Αντικειμενοστραφής προγραμματισμός η επικρατούσα κατάσταση της τελευταίας δεκαετίας. Περιγράφει ενέργειες που εφαρμόζονται πάνω σε δεδομένα. Πρωτεύοντα δομικά στοιχεία ενός προγράμματος είναι τα δεδομένα από τα οποία με κατάλληλη μορφοποίηση δημιουργούν τα αντικείμενα.

Παράλληλος προγραμματισμός Υπάρχουν υπολογιστές με περισσότερους από έναν επεξεργαστή. Οι επεξεργαστές μοιράζονται την ίδια μνήμη και λειτουργούν παράλληλα εκτελώντας διαφορετικές εντολές του ίδιου προγράμματος. Πτευχαιίνουν μεγάλες ταχύτητες επεξεργασίας αλλά το πρόβλημα πρέπει να διαιρεθεί σε τμήματα που

εκτελούνται παράλληλα και να προγραμματισθεί σε ένα περιβάλλον που να επιτρέπει τον παράλληλο προγραμματισμό.

### Προγραμματιστικά περιβάλλοντα

Οι εντολές κάθε προγράμματος ανεξάρτητα της γλώσσας που έχουν γραφεί για να εκτελεστούν πρέπει πρώτα να μετατραπούν σε μορφή αναγνωρίσιμη από τον υπολογιστή δηλαδή σε εντολές γλώσσας μηχανής.

Πηγαίο είναι το αρχικό πρόγραμμα που γράφεται σε μία γλώσσα υψηλού επιπέδου Η μετατροπή γίνεται με :

- Μεταγλωττιστή δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα υψηλού επιπέδου και επιστρέφει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. Αντικείμενο ονομάζεται το πρόγραμμα που παράγεται και περιέχει εκτός από τις εντολές του προγράμματος και τμήματα που βρίσκονται σε βιβλιοθήκες της γλώσσας.

Συνδετής-φορτωτής είναι το πρόγραμμα που συνδέει τις εντολές του προγράμματος με τα τμήματα από τις βιβλιοθήκες και έχει ως αποτέλεσμα την δημιουργία του εκτελέσιμου προγράμματος.

- Διερμηνευτή διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μία από αυτές εκτελεί μία ισοδύναμη ακολουθία εντολών γλώσσας μηχανής.

Συντάκτης (editor) ένας μικρός επεξεργαστής κειμένου με δυνατότητες για την γρήγορη γραφή των εντολών του προγράμματος.

## ΚΕΦΑΛΑΙΟ 7 Βασικές έννοιες Προγραμματισμού

- Όλες οι εντολές της ΓΛΩΣΣΑΣ γράφονται με κεφαλαία.
- Οι μεταβλητές έχουν το πρώτο γράμμα πάντα κεφαλαίο ενώ τα υπόλοιπα μπορεί να είναι μικρά ή κεφαλαία.
- Γράμματα επιτρέπονται τα :
  - Κεφαλαία ελληνικά
  - Πεζά ελληνικά
  - Κεφαλαία αγγλικά
  - Πεζά αγγλικά
- Ψηφία από 0 έως 9
- Ειδικοί χαρακτήρες + - \* / = · ( ) . , ' ! &
- Τύποι δεδομένων
  - Ακέραιοι - θετικοί, αρνητικοί, μηδέν
  - Πραγματικοί - θετικοί, αρνητικοί, μηδέν
  - Χαρακτήρες - μπορούν να περιέχουν οποιοδήποτε χαρακτήρα παράγεται από το πληκτρολόγιο. Πρέπει να βρίσκονται σε απλά εισαγωγικά ''. Ονομάζονται και αλφαριθμητικοί.
  - Λογικοί - δέχονται δύο τιμές την ΑΛΗΘΗ και την ΨΕΥΔΗ.

- Σταθερές - είναι προκαθορισμένες τιμές που δεν μεταβάλλονται κατά την διάρκεια εκτέλεσης του προγράμματος. Μπορεί να είναι ακέραιες, πραγματικές αλφαριθμητικές ή λογικές.
- Μεταβλητές - η τιμή τους μπορεί να μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Μπορεί να είναι ακέραιες, πραγματικές αλφαριθμητικές ή λογικές.
- Ονόματα σταθερών και μεταβλητών - πρέπει να αρχίζουν με κεφαλαίο γράμμα του ελληνικού ή αγγλικού αλφαβήτου, τα υπόλοιπα μπορεί να είναι μικρά γράμματα του ελληνικού ή αγγλικού αλφαβήτου, αριθμοί και οι λέξεις μπορούν να ενώνονται με \_. Δεν μπορεί να χρησιμοποιηθεί σαν όνομα σταθερής ή μεταβλητής λέξη που είναι εντολή της γλώσσας (δεσμευμένη λέξη).
- Αριθμητικοί τελεστές - είναι το + , - , \* , / , ^ , DIV (το ακέραιο μέρος του πηλίκου μίας διαίρεσης π.χ.  $10 \text{ div } 3 = 3$  ) , MOD (το υπόλοιπο μίας διαίρεσης π.χ.  $10 \text{ mod } 3 = 1$ ).
- Συναρτήσεις
  - HM(X) Υπολογισμός ημίτονου
  - ΣΥΝ(X) Υπολογισμός συνημίτονου
  - ΕΦ(X) Υπολογισμός εφαπτόμενης
  - Τ\_P(X) Υπολογισμός τετραγωνικής ρίζας
  - ΛΟΓ(X) Υπολογισμός φυσικού λογάριθμου
  - Ε(X) Υπολογισμός του  $e^x$
  - A\_M(X) Ακέραιο μέρος του X
  - A\_T(X) Απόλυτη τιμή του X
- Αριθμητική έκφραση - όταν μία τιμή προκύπτει από υπολογισμό. Χρησιμοποιούνται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί τελεστές και παρενθέσεις. Σε μία έκφραση πρώτα εκτελείται η ύψωση σε δύναμη, μετά ο πολλαπλασιασμός και η διαίρεση, και τελευταία η πρόσθεση και η αφαίρεση. Ότι υπάρχει μέσα σε παρενθέσεις εκτελείται πρώτο.
- ΔΙΑΒΑΣΕ για την εισαγωγή δεδομένων κατά την εκτέλεση ενός προγράμματος
- ΓΡΑΨΕ για την εμφάνιση των αποτελεσμάτων
- ΠΡΟΓΡΑΜΜΑ η πρώτη λέξη του προγράμματος ακολουθούμενη από την ονομασία του προγράμματος
- ΑΡΧΗ μετά την δήλωση των μεταβλητών και πριν την πρώτη εντολή
- ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ η τελευταία εντολή του προγράμματος (εκτός αν χρησιμοποιούνται διαδικασίες οπότε μπαίνουν μετά) με το όνομα του προγράμματος.

## ΚΕΦΑΛΑΙΟ 8 Επιλογή και Επανάληψη

**ΑΝ** συνθήκη **ΤΟΤΕ**

εντολή 1

εντολή 2

..

εντολή ν

**ΤΕΛΟΣ\_ΑΝ**

Αν ισχύει η συνθήκη τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ του **ΤΟΤΕ** και του **ΤΕΛΟΣ\_ΑΝ**. Αν δεν ισχύει η συνθήκη εκτελείται η εντολή που βρίσκεται μετά το **ΤΕΛΟΣ\_ΑΝ**.

**ΑΝ** συνθήκη **ΤΟΤΕ**

εντολή 1

εντολή 2

..

εντολή ν

**ΑΛΛΙΩΣ**

εντολή-1

εντολή-2

..

εντολή-ν

**ΤΕΛΟΣ\_ΑΝ**

Αν ισχύει η συνθήκη τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ του **ΤΟΤΕ** και του **ΑΛΛΙΩΣ** διαφορετικά εκτελούνται οι εντολές που βρίσκονται μεταξύ του **ΑΛΛΙΩΣ** και του **ΤΕΛΟΣ\_ΑΝ**.

**ΑΝ** συνθήκη1 **ΤΟΤΕ**

εντολή 1

εντολή 2

..

εντολή ν

**ΑΛΛΙΩΣ\_ΑΝ** συνθήκη2 **ΤΟΤΕ**

εντολή-1

εντολή-2

..

εντολή-ν

..

..

**ΑΛΛΙΩΣ**

εντολή\_1

εντολή\_2

..  
εντολή\_v  
**ΤΕΛΟΣ\_ΑΝ**

**ΕΠΙΛΕΞΕ** έκφραση (μεταβλητή)  
**ΠΕΡΙΠΤΩΣΗ** τιμή1 (ή τιμές1)  
εντολή1α  
εντολή1β

...  
**ΠΕΡΙΠΤΩΣΗ** τιμή2 (ή τιμές2)  
εντολή2α  
εντολή2β

...  
...  
**ΠΕΡΙΠΤΩΣΗ** τιμή v (ή τιμές v)  
εντολή να  
εντολή νβ

...  
**ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ**  
εντολή αλλιώς α  
εντολή αλλιώς β

...  
**ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ**

Βάση της τιμής που έχει η έκφραση εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση. Αν η τιμή της έκφρασης δεν αντιστοιχεί σε καμία από τις τιμές των περιπτώσεων τότε εκτελούνται οι εντολές της **ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ**.

**ΟΣΟ** συνθήκη **ΕΠΑΝΑΛΑΒΕ**

εντολή 1  
εντολή 2

..  
εντολή v  
**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

Αν ισχύει η συνθήκη εκτελούνται οι εντολές μεταξύ του **ΕΠΑΝΑΛΑΒΕ** και του **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**. Μετά ελέγχεται ξανά η συνθήκη και εφόσον συνεχίζει να ισχύει εκτελούνται ξανά οι εντολές. Αυτό συνεχίζει να επαναλαμβάνεται μέχρις ότου πάψει να ισχύει η συνθήκη οπότε και εκτελείται η εντολή που βρίσκεται μετά το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

εντολή 1

εντολή 2

..

εντολή ν

**ΜΕΧΡΙΣ\_ΟΤΟΥ** συνθήκη

Εκτελούνται οι εντολές μεταξύ του **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ**.

Μετά ελέγχεται η συνθήκη και εάν δεν ισχύει τότε εκτελούνται πάλι οι εντολές. Αυτό επαναλαμβάνεται μέχρις ότου η συνθήκη να γίνει αληθής οπότε και εκτελείται η εντολή που υπάρχει μετά το **ΜΕΧΡΙΣ\_ΟΤΟΥ**.

Προσοχή οι εντολές που βρίσκονται μέσα στο βρόγχο θα εκτελεστούν οπωσδήποτε μία φορά (μέχρι να γίνει ο έλεγχος της συνθήκης του **ΜΕΧΡΙΣ\_ΟΤΟΥ**).

**ΓΙΑ** μεταβλητή **ΑΠΟ** τιμή1 **ΜΕΧΡΙ** τιμή2 **ΜΕ ΒΗΜΑ** τιμή3

εντολή 1

εντολή 2

..

εντολή ν

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

Τη χρησιμοποιούμε όταν είναι γνωστός ο αριθμός των επαναλήψεων. Οι εντολές του βρόγχου εκτελούνται για όλες τις τιμές της μεταβλητής ξεκινώντας από την τιμή1 και φτάνοντας στη τιμή2 αυξανόμενης κάθε φορά με την τιμή της τιμή3. Αν η τιμή3 είναι ίση με 1 τότε παραλείπεται το **ΜΕ ΒΗΜΑ** τιμή3.

## ΚΕΦΑΛΑΙΟ 9 Πίνακες

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.

Κάθε πίνακας περιέχει δεδομένα του ίδιου τύπου - ακέραια, πραγματικά, αλφαριθμητικά ή λογικά. Ο τύπος του δηλώνεται μαζί με τις άλλες μεταβλητές του προγράμματος και επίσης δηλώνεται και ο αριθμός των στοιχείων που περιέχει.

Πλεονεκτήματα :

- Οι τιμές του πίνακα παραμένουν προς χρήση μέχρι τον τερματισμό του αλγόριθμου.
- Εύκολη η διαχείριση πολλών δεδομένων ίδιου τύπου.
- Χρησιμοποιούμε μόνο μία μεταβλητή.

Μειονεκτήματα :

- Δεσμεύουν από την αρχή του προγράμματος πολλές θέσεις μνήμης.
- Περιορίζουν τις δυνατότητες του προγράμματος γιατί έχουν στατική δομή - το μέγεθός τους ορίζεται στην αρχή του προγράμματος.

**Μονοδιάστατοι πίνακες** χρησιμοποιούν ένα δείκτη για την αναφορά των στοιχείων τους.

**Πολυδιάστατοι πίνακες** χρησιμοποιούν δύο ή περισσότερους δείκτες για την αναφορά των στοιχείων τους.

Τυπικές επεξεργασίες πινάκων

- Υπολογισμός αθροισμάτων στοιχείων - για στοιχεία με κοινά χαρακτηριστικά π.χ. να βρίσκονται στην ίδια στήλη ή στην ίδια γραμμή
- Εύρεση του μέγιστου ή του ελάχιστου - αν ο πίνακας είναι ταξινομημένος τότε το μέγιστο και το ελάχιστο βρίσκονται στις δύο άκρες του πίνακα, αν δεν είναι ταξινομημένος τότε πρέπει να συγκριθούν όλα τα στοιχεία ένα προς ένα.
- Ταξινόμηση των στοιχείων
- Αναζήτηση ενός στοιχείου
  - **Σειριακή** - χρησιμοποιείται σε μη ταξινομημένους πίνακες, είναι η πιο απλή αλλά καθυστερεί την εκτέλεση του αλγόριθμου
  - **Διαδική** - χρησιμοποιείται σε ταξινομημένους πίνακες, αλλά είναι πιο πολύπλοκη
- Συγχώνευση - όταν από δύο ή περισσότερους ταξινομημένους πίνακες δημιουργείται ένας καινούργιος πίνακας επίσης ταξινομημένος.

**Ταξινόμηση Φυσαλίδας σε Αύξουσα σειρά** (για φθίνουσα αλλάζουμε το πρόσημο της σύγκρισης)

(Δεδομένα: Μονοδιάστατος πίνακας ΠΙΝ με N θέσεις)

ΓΙΑ Χ ΑΠΟ 2 ΜΕΧΡΙ Ν

    ΓΙΑ Ψ ΑΠΟ Ν ΜΕΧΡΙ Χ ΜΕ\_ΒΗΜΑ -1

        ΑΝ ΠΙΝ[Ψ-1] > ΠΙΝ[Ψ]

            ΤΟΤΕ TEMP ← ΠΙΝ[Ψ-1]

            ΠΙΝ[Ψ-1] ← ΠΙΝ[Ψ]

            ΠΙΝ[Ψ] ← TEMP

        } ΑΝΤΙΜΕΤΑΘΕΣΕ ΠΙΝ[Ψ-1], ΠΙΝ[Ψ]

    ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

## ΚΕΦΑΛΑΙΟ 10 Υποπρογράμματα

Διαιρούμε πολύπλοκα προγράμματα σε μικρότερα υποπροβλήματα. Η επίλυση των υποπροβλημάτων αυτών οδηγεί στην επίλυση του αρχικού προβλήματος.

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα :

- Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο. Κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό, εκτελεί ορισμένες ενέργειες, και απενεργοποιείται με την έξοδο από αυτό.
- Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα. Κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα.
- Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο. Κάθε υποπρόγραμμα να είναι τόσο, ώστε να είναι εύκολα κατανοητό για να μπορεί να ελέγχεται. Κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

Πλεονεκτήματα του τμηματικού προγραμματισμού :

- Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος.
- Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.
- Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.
- Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.

Παράμετροι είναι οι τιμές που περνούν από ένα υποπρόγραμμα σε άλλο. Είναι σαν τις κοινές μεταβλητές ενός προγράμματος με τη διαφορά ότι χρησιμοποιούνται για να περνούν τιμές στα υποπρογράμματα.

Διαδικασίες μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα.

- να εισάγουν δεδομένα
- να εκτελέσουν υπολογισμούς
- να μεταβάλλουν τις τιμές των μεταβλητών
- να τυπώσουν αποτελέσματα

Για να εκτελεστεί η διαδικασία χρησιμοποιείται η ειδική εντολή **ΚΑΛΕΣΕ** και το όνομα της διαδικασίας.

Συναρτήσεις υπολογίζουν μόνο μία τιμή, αριθμητική, χαρακτήρα ή λογική και μόνο αυτήν επιστρέφουν στο υποπρόγραμμα που την κάλεσε.

Πραγματικές παράμετροι ονομάζονται οι μεταβλητές του προγράμματος που αποστέλλουν τιμές στο υποπρόγραμμα.

Τυπικές παράμετροι ονομάζονται οι μεταβλητές του υποπρογράμματος που δέχονται τιμές από το πρόγραμμα.



Τοπικές είναι οι μεταβλητές που ισχύουν μόνο στο τμήμα του προγράμματος που έχουν δηλωθεί. (Ισχύει για όλες τις μεταβλητές της ΓΛΩΣΣΑΣ).

#### Κανόνες για τις λίστες παραμέτρων

- Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.
- Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση.
- Η τυπική παράμετρος και η αντίστοιχη της πραγματική παράμετρος πρέπει να είναι του ίδιου τύπου.

#### Κατηγορίες υποπρογραμμάτων και οι μεταξύ τους διαφορές.

Το είδος υποπρογράμματος καθορίζεται από το είδος της λειτουργίας που επιτελεί. Οι συναρτήσεις και οι διαδικασίες τοποθετούνται μετά το τέλος του κυρίου προγράμματος και καλούνται στα σημεία που χρειάζονται.

Χαρακτηριστικά / διαφορές των διαδικασιών και των συναρτήσεων.

Συναρτήσεις	Διαδικασίες
Οι συναρτήσεις υπολογίζουν μόνο μία τιμή, αριθμητική, χαρακτήρα ή λογική και μόνο αυτήν επιστρέφουν στο υποπρόγραμμα που την κάλεσε.	Οι διαδικασίες μπορούν να εκτελέσουν οποιαδήποτε λειτουργία, π.χ. να εισάγουν δεδομένα, να εκτελέσουν υπολογισμούς, να μεταβάλλουν τις τιμές των μεταβλητών και να τυπώσουν αποτελέσματα.
Οι συναρτήσεις μοιάζουν με τις συναρτήσεις των μαθηματικών και η χρήση τους είναι όμοια με τη χρήση των ενσωματωμένων συναρτήσεων που υποστηρίζει η γλώσσα προγραμματισμού.	Με τη χρήση παραμέτρων οι διαδικασίες μεταφέρουν τα αποτελέσματα τους στα άλλα υποπρογράμματα.
Οι συναρτήσεις εκτελούνται απλά με την εμφάνιση του ονόματος τους σε οποιαδήποτε έκφραση.	Για να ενεργοποιηθούν οι διαδικασίες χρησιμοποιείται η ειδική εντολή ΚΑΛΕΣΕ και το όνομα της διαδικασίας.

#### Παρατηρήσεις στη δομή της διαδικασίας :

1. Το όνομα μιας διαδικασίας μπορεί να είναι οποιοδήποτε έγκυρο όνομα της ψευδογλώσσας.
2. Η λίστα παραμέτρων είναι μια λίστα μεταβλητών, των οποίων οι τιμές μεταβιβάζονται στη διαδικασία, κατά την κλήση της από το πρόγραμμα, αλλά και επιστρέφονται στο κύριο πρόγραμμα μετά την εκτέλεση των εντολών της διαδικασίας.
3. Στο σώμα της διαδικασίας μπορούν να υπάρχουν οποιεσδήποτε εντολές της ψευδογλώσσας.
4. Πριν τη λέξη ΑΡΧΗ δηλώνονται οι τύποι των μεταβλητών εισόδου, καθώς και οι τυχόν βοηθητικές μεταβλητές ή σταθερές που χρησιμοποιεί η διαδικασία.