

Δομημένες τεχνικές

Ο στόχος των δομημένων τεχνικών είναι:

- Υψηλής ποιότητας προγράμματα
- Εύκολη τροποποίηση προγραμμάτων
- Απλοποιημένα προγράμματα
- Μείωση κόστους και χρόνου ανάπτυξης.

Οι βασικές αρχές τους είναι:

- Αρχή της αφαίρεσης
- Αρχή της τυπικότητας.
- Έννοια της διαίρεσης και υποταγής
- Έννοια της ιεραρχικής ταξινόμησης.

Αρχή της αφαίρεσης

Απλοποιεί τα γεγονότα έτσι ώστε να περιγράψει τι έχει γίνει και όχι πως έχει γίνει.

Αρχή της τυπικότητας.

Η ανάπτυξη του αλγορίθμου είναι μία σταδιακή και συστηματική διαδικασία. Τυπικότητα δεν σημαίνει διαδικασία ρουτίνας.

Έννοια της διαίρεσης και υποταγής

Αντιμετωπίζει την πολυπλοκότητα, διαιρώντας τον αλγόριθμο σε ανεξάρτητα μέρη, με αποτέλεσμα να είναι εύκολη η διαχείριση τους.

Έννοια της ιεραρχικής ταξινόμησης

Τακτοποιεί τις ενότητες από το πιο αφηρημένο ψηλότερο επίπεδο στο λεπτομερέστερο κατώτερο επίπεδο.

Ο Αρθρωτός Προγραμματισμός (Modular programming)

Στόχος: Διάσπαση του προβλήματος σε μικρότερες και απλούστερες ενότητες.

Αποτέλεσμα: Μείωση της πολυπλοκότητας.

Στηρίζεται στην αρχή της διαίρεσης και υποταγής. Συστατικό του είναι η ενότητα.

Πλεονεκτήματα

- Δυνατότητα δημιουργίας βιβλιοθηκών.
- Ευκολότερη κωδικογράφηση και έλεγχος λαθών.
- Ευκολότερη συντήρηση.
- Καταμερισμός εργασίας.

Μειονεκτήματα

- Λανθασμένη χρήση του Α.Π.
- Αυξημένος χρόνος για τον καθορισμό των ενοτήτων.

Ιεραρχικός Προγραμματισμός (hierarchical programming)

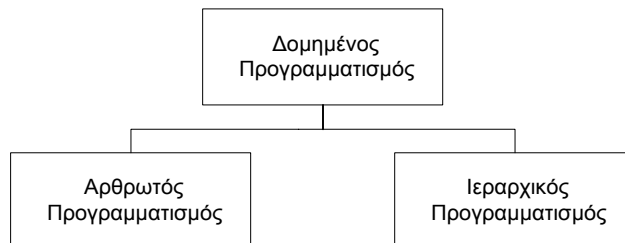
Στόχος: Εύρεση και καθορισμός της διασύνδεσης των κυριοτέρων λειτουργιών.

Αποτέλεσμα: Σωστή επικοινωνία ενοτήτων.

Καθορίζει τον τρόπο προσέγγισης της λύσης. Τα βήματα σχεδιασμός, κωδικοποίηση, και έλεγχος επαναλαμβάνονται διαδοχικά. Ο τρόπος αυτός είναι η μέθοδος top-down.

Δομημένος Προγραμματισμός

Συγκροτείται από τον Αρθρωτό Προγραμματισμό και τον Ιεραρχικό Προγραμματισμό.



Ενότητα.

Ενότητα είναι ένα λογικό σύνολο εντολών, που εκτελεί μία καθορισμένη λειτουργία και είναι ανεξάρτητο από τα άλλα μέρη του προγράμματος.

Ο βαθμός ανεξαρτησίας καθορίζει το αν και πόσο οι αλλαγές σε μία ενότητα, επιβάλλουν αλλαγές σε άλλες ενότητες.

Χαρακτηριστικά Ενότητας

- Κάθε ενότητα αντιπροσωπεύει μία αυτοτελή λογική.
- Η ενότητα αντιστοιχεί σε μία ανεξάρτητη λειτουργία του προγράμματος ή σε ένα άλλο πρόγραμμα.
- Κάθε ενότητα έχει μία είσοδο, μία έξοδο και εκτελεί μία καθορισμένη επεξεργασία (3ε).
- Κάθε ενότητα καλείται από μία άλλη ενότητα και επιστρέφει σ' αυτήν όταν τελειώσει.
- Κάθε ενότητα αποτελείται από τις βασικές δομές.
- Σε κάθε ενότητα δεν πρέπει να υπάρχει:
 - Απότομη διακοπή.
 - Άπειρη επανάληψη.
- Δεν πρέπει να υπάρχει ενότητα που δεν εκτελείται.
- Πολλές ενότητες κάνουν το πρόγραμμα πολύπλοκο και αυξάνουν τον χρόνο εκτέλεσης.
- Λίγες ενότητες απομακρύνουν τη λύση από τις αρχές του Α.Π.

Κλήση ενότητας / αλγόριθμου από αλγόριθμο

Η χρήση των δομημένων τεχνικών δίνει την δυνατότητα δημιουργίας βιβλιοθηκών υποπρογραμμάτων, τα οποία δέχονται ή όχι συγκεκριμένες παραμέτρους και αποδίδουν συγκεκριμένα αποτελέσματα.

Ένας αλγόριθμος μπορεί να κληθεί από έναν άλλο αλγόριθμο με χρήση της εντολής **Κάλεσε**.

Η εντολή αυτή συνοδεύεται με το όνομα του καλούμενου αλγορίθμου ακολουθούμενο από λίστα μεταβλητών ή σταθερών μέσα σε παρενθέσεις. Οι τιμές των μεταβλητών ή σταθερών μεταβιβάζονται κατά την κλήση στις αντίστοιχες μεταβλητές της γραμμής Αλγόριθμος του καλούμενου αλγορίθμου. Όταν ο καλούμενος αλγόριθμος τερματίσει τη λειτουργία του, γίνεται επιστροφή στην αμέσως επόμενη εντολή της Κάλεσε. Κατά την επιστροφή μπορεί επίσης να μεταβιβάζονται τιμές μέσω των μεταβλητών που αναφέρονται στην παρένθεση.

Αλγόριθμος Καλών
Δεδομένα //α, β, γ//
Διάβασε α, β
Κάλεσε Καλούμενος (α, β, γ)
Γράψε "Αποτέλεσμα = ", γ
Τέλος Καλών

Αλγόριθμος Καλούμενος (x, y, z)
 z ← x + y
Τέλος Καλούμενος

Η αντιστοιχία των μεταβλητών των δύο αλγορίθμων γίνεται με τη σειρά που αναφέρονται στις αντίστοιχες γραμμές, δηλ. $\alpha \leftrightarrow x$, $\beta \leftrightarrow y$ και $\gamma \leftrightarrow z$.

Τα ονόματα των μεταβλητών σε διαφορετικούς αλγόριθμους δεν έχουν καμία σχέση.

Όταν ένας αλγόριθμος έχει σχεδιαστεί με χρήση ενοτήτων τότε ο κύριος αλγόριθμος προκαλεί την εκτέλεση των επιμέρους ενοτήτων με την εντολή **Εκτέλεσε**. Η σύνταξη γίνεται:

```

Αλγόριθμος Κύριος
Δεδομένα //α, β, γ//
Διάβασε α, β
Εκτέλεσε 1000_Ελεγχος (α, β, γ)
Γράψε "Αποτέλεσμα = ", γ
Τέλος
1000_Ελεγχος (x, y, z)
 $z \leftarrow x + y$ 
Έξοδος
Τέλος Κύριος

```

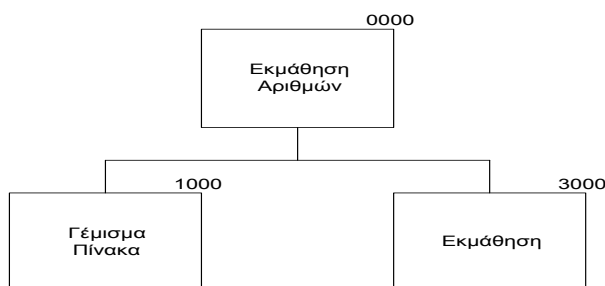
Η εντολή **έξοδος** σημειώνει το τέλος της κάθε ενότητας. Κάθε νεότητα γράφεται μετά το λογικό τέλος του κύριου αλγόριθμου, που ορίζεται με την εντολή **Τέλος**.

Εντολή Έξοδος

Η εντολή αυτή προκαλεί την έξοδο από μια επαναληπτική διαδικασία ή από έναν αλγόριθμο, όταν πληρείται κάποια συνθήκη. Παρέχει έναν εναλλακτικό τρόπο τερματισμού μιας επανάληψης ή μιας διαδικασίας (αλγόριθμου).

Οπτικός πίνακας περιεχομένων (Visual table of contents, V.T.O.C.)

Σε κάθε αλγόριθμο που εφαρμόζονται οι αρχές του Δ.Π. είναι χρήσιμο να εμφανίζονται οι λειτουργίες / ενότητες με ένα ξεχωριστό τρόπο. Αυτό γίνεται απαραίτητο όταν ο αλγόριθμος αφορά σε πολύπλοκο πρόβλημα. Μία τέτοια παρουσίαση εντάσσεται στις εργασίες τεκμηρίωσης και επηρεάζει την κωδικοποίηση του αλγορίθμου.

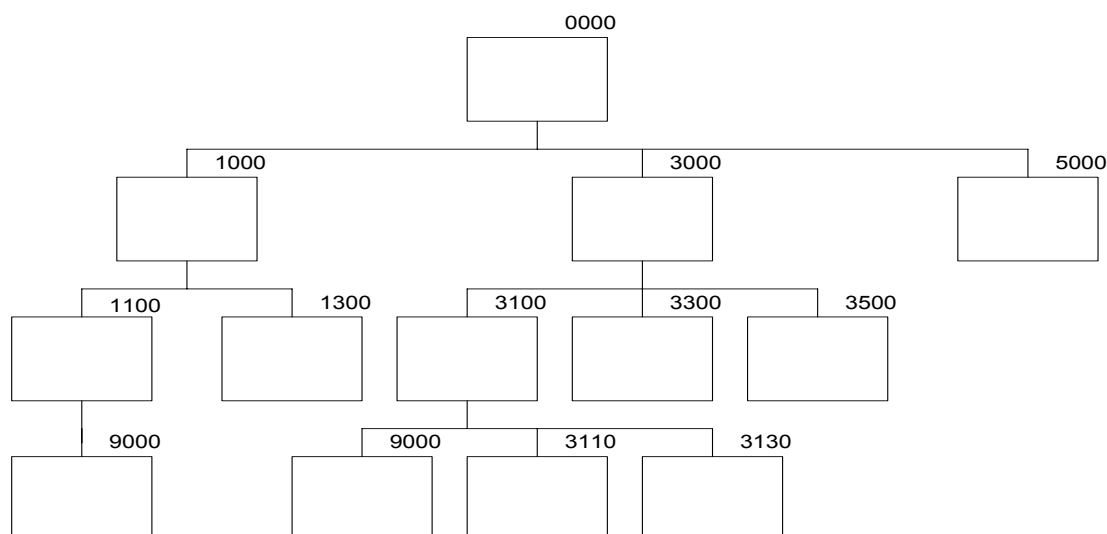


Το παραπάνω σχήμα είναι ένας οπτικός πίνακας περιεχομένων για την άσκηση εκμάθησης αριθμών. Η παρουσίαση υποδηλώνει ότι ο αλγόριθμος έχει δύο ενότητες την 1000 Γέμισμα Πίνακα και την 3000 Εκμάθηση. Στον V.T.O.C. δηλώνεται η σειρά εκτέλεσης των ενότητων.

Ο οπτικός πίνακας περιεχομένων "διαβάζεται" από επάνω προς τα κάτω και από αριστερά προς τα δεξιά.

Δηλαδή οι ενότητες που παρουσιάζονται εκτελούνται από επάνω προς τα κάτω και από αριστερά προς τα δεξιά. Οι αριθμοί που εμφανίζονται στην άνω δεξιά γωνία αφορούν στην καλλίτερη τεκμηρίωση. Σχετίζονται με τον βαθμό ανεξαρτησίας της ενότητας και αποτελούν τον οδηγό ελέγχου στην φάση συντήρησης του προγράμματος.

Στο επόμενο σχήμα παρουσιάζεται ένας οπτικός πίνακας περιεχομένων ενός αλγορίθμου με περισσότερες λειτουργίες. Σημειώστε ότι ο αριθμός 9000 που παρουσιάζεται αφορά σε ενότητες που καλούνται από διαφορετικούς κλάδους του αλγορίθμου και σε διαφορετικά επίπεδα. Αυτές οι ενότητες χαρακτηρίζονται "κοινής χρήσης" και είναι συνήθως οι ενότητες εισαγωγής στοιχείων, εκτύπωσης ή εμφάνισης.



Αναδρομικοί αλγόριθμοι.

Ένας αλγόριθμος μπορεί να καλεί τον εαυτό του (αναδρομική κλήση). Στο επόμενο παράδειγμα παρουσιάζεται ο αναδρομικός αλγόριθμος υπολογισμού του μέγιστου κοινού διαιρέτη (αλγόριθμος του Ευκλείδη).

Σημειώνεται ότι η γραμμή Δεδομένα αποτελεί την πύλη επικοινωνίας μεταξύ των μεταβλητών στην αναδρομική κλήση. Δηλαδή η μεταβλητή y δίνει τιμή στην x και η z στην y ενώ η μεταβλητή m είναι η.

Αλγόριθμος ΜΚΔ

Δεδομένα //x, y//

Αν $y \neq 0$

τότε **Κάλεσε** ΜΚΔ ($y, x \bmod y$)

$z \leftarrow x \bmod y$

αλλιώς $z \leftarrow x$

Τέλος_αν

Εμφάνισε "Μ.Κ.Δ.=" , z

Τέλος ΜΚΔ

Παραθέτουμε για σύγκριση τη λύση του προβλήματος με αλγόριθμο χωρίς αναδρομή:

Αλγόριθμος Ευκλείδης

Δεδομένα //x, y $\in \mathbb{N}^*$, z:ακέραια μεταβλητή//

$z \leftarrow x \bmod y$

Όσο $z \neq 0$ **επανάλαβε**

$x \leftarrow y$

$y \leftarrow z$

$z \leftarrow x \bmod y$

Τέλος Επανάληψης

Εμφάνισε "Μ.Κ.Δ.=" , y

Τέλος Ευκλείδης

Την δυνατότητα αναδρομικής κλήσης δεν έχουν όλες οι γλώσσες προγραμματισμού. Αρά ένας αλγόριθμός με αναδρομή δεν είναι κατάλληλος για κωδικοποίηση σε κάθε γλώσσα.