

ΚΕΦ.10 - ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Όπως γνωρίζουμε ήδη, τα σύνθετα προβλήματα προκειμένου να αντιμετωπισθούν και να γραφούν τα αντίστοιχα προγράμματα, πρέπει πρώτα να ακολουθηθεί η τεχνική της Ιεραρχικής Σχεδίασης (top-down design) από πάνω προς τα κάτω.
- Έτσι κάθε δοθέν πρόβλημα αναλύεται σε μικρότερα και αυτά σε ακόμα μικρότερα μέχρι το σημείο στο οποίο φτάνουμε σε προβλήματα με προφανή και εύκολη λύση.
- Τότε οι αλγόριθμοι και τα τμήματα προγράμματος μπορούν να σχεδιαστούν και να γραφούν εύκολα για κάθε υποπρόβλημα.

Τμηματικός Προγραμματισμός είναι η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Υποπρόγραμμα (Subprogram) είναι ένα τμήμα προγράμματος που επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

- Ο χωρισμός του προγράμματος σε ανεξάρτητα υποπρογράμματα προϋποθέτει το αρχικό πρόβλημα να έχει αναλυθεί σε υποπροβλήματα βάσει της ιεραρχικής σχεδίασης.
- Δεν υπάρχουν κανόνες σωστής ανάλυσης.
- Ο τμηματικός προγραμματισμός απαιτεί μελέτη και ανάλυση του προβλήματος, εμπειρία στο προγραμματισμό, γνώσεις και ταλέντο.

ΙΔΙΟΤΗΤΕΣ ΥΠΟΠΡΟΓΡΑΜΜΑΤΟΣ

▶ Να έχει μία μόνο είσοδο και μία έξοδο

Η είσοδος γίνεται πάντα από την αρχή του, ακολουθεί σειρά ενεργειών επεξεργασίας και η έξοδος γίνεται πάντα από το τέλος του.

▶ Να είναι ανεξάρτητο από τα άλλα

Δηλαδή κάθε υποπρόγραμμα σχεδιάζεται, αναπτύσσεται και συντηρείται αυτόνομα, χωρίς να επηρεάζει τα άλλα.

▶ Να μην είναι πολύ μεγάλο

Πρέπει να εκτελεί μόνο μία λειτουργία, ώστε να είναι κατανοητό και να ελέγχεται

ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΜΗΜΑΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**► Εύκολη ανάπτυξη αλγορίθμου και προγράμματος**

Με σταδιακή επίλυση των υποπροβλημάτων και δημιουργία αντίστοιχων υποπρογραμμάτων και όχι κατευθείαν αντιμετώπιση του αρχικού προβλήματος.

► Εύκολη κατανόηση και διόρθωση προγράμματος

Ένα υποπρόγραμμα μπορεί να διορθωθεί ευκολότερα από ότι το συνολικό πρόγραμμα, χωρίς βέβαια η διόρθωση να επηρεάζει κάποιο άλλο. Επίσης με την τμηματοποίηση, το πρόγραμμα είναι ευκολότερο να κατανοηθεί και να διαβαστεί από κάποιον συντηρητή προγραμματιστή.

► Λιγότερος χρόνος και προσπάθεια για συγγραφή προγράμματος

Επειδή η ίδια λειτουργία μέσα σε ένα πρόγραμμα μπορεί να επαναλαμβάνεται, για το λόγο αυτό γράφουμε ένα υποπρόγραμμα που εκτελεί την λειτουργία αυτή και το καλούμε από πολλά σημεία μέσα στο πρόγραμμα. Έτσι μειώνεται ο χρόνος συγγραφής καθώς και η πιθανότητα λάθους.

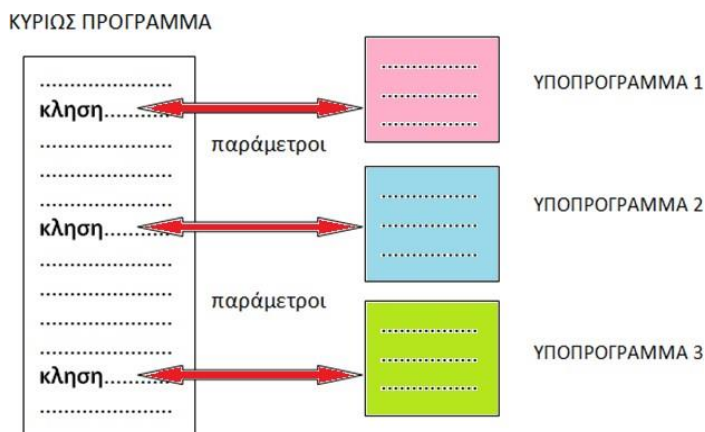
► Επέκταση δυνατοτήτων στις γλώσσες προγραμματισμού

Ένα υποπρόγραμμα μπορεί να χρησιμοποιηθεί σε περισσότερα του ενός προγράμματα. Η χρήση του είναι ίδια με αυτήν των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού (π.χ. ημίτονο, συνημίτονο). Αν μια γλώσσα δεν υποστηρίζει μια λειτουργία, τότε αυτή παρέχεται από τα υποπρογράμματα μιας άλλης γλώσσας μέσω της χρήσης βιβλιοθηκών υποπρογραμμάτων.

ΠΑΡΑΜΕΤΡΟΙ

- Η ενεργοποίηση ενός υποπρογράμματος γίνεται μέσα από την κλήση του (το κάλεσμά του).
- Η κλήση ενός υποπρογράμματος γίνεται από το αρχικό πρόγραμμα ή από κάποιο άλλο υποπρόγραμμα που λέγεται **κύριο πρόγραμμα**.
- Το υποπρόγραμμα δέχεται τιμές από το τμήμα προγράμματος που το καλεί και αφού κάνει την εκτέλεσή του, επιστρέφει σε αυτό νέες τιμές ως αποτελέσματα.

Παράμετροι (parameters) είναι οι τιμές που περνούν μεταξύ των υποπρογραμμάτων και είναι σαν τις κοινές μεταβλητές.



ΚΑΤΗΓΟΡΙΕΣ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

Ανάλογα με την λειτουργία που εκτελούν, τα υποπρογράμματα διακρίνονται σε δύο βασικές κατηγορίες :

- διαδικασίες (εκτελούν κάθε λειτουργία που μπορεί να εκτελέσει και το πρόγραμμα)
- συναρτήσεις (η χρήση τους είναι πιο περιορισμένη).

Και οι δύο τύποι υποπρογραμμάτων τοποθετούνται μετά το τέλος του κυρίου προγράμματος, ενώ ο τρόπος κλήσης και σύνταξής τους είναι διαφορετικός.

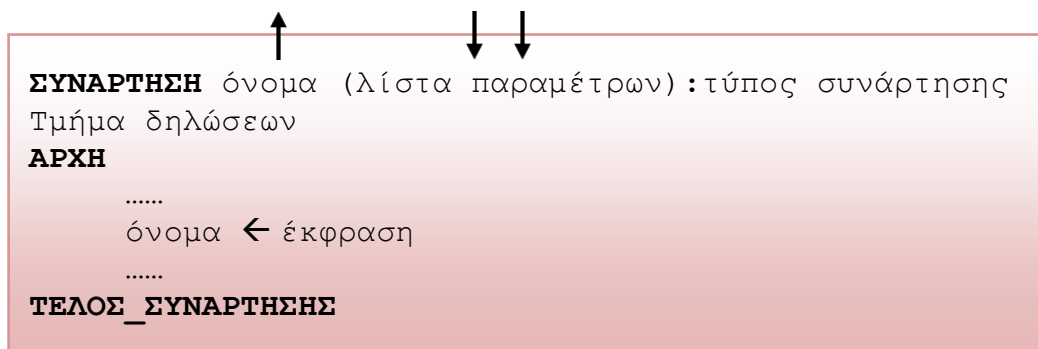
ΣΥΝΑΡΤΗΣΗ (Function)

Η **συνάρτηση** είναι τύπος υποπρογράμματος που υπολογίζει μία τιμή (αριθμητική, χαρακτήρα ή λογική) και την επιστρέφει μέσα από το όνομά της, όπως οι μαθηματικές συναρτήσεις.

Η συνάρτηση επιστρέφει την τιμή στο υποπρόγραμμα που την κάλεσε και η χρήση της είναι παρόμοια με αυτή των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού.

Η τιμή που επιστρέφει μπορεί να είναι κάθε δυνατού τύπου δεδομένων, δηλαδή ακεραίου, πραγματικού, χαρακτήρα, λογικού.

Η βασική δομή μιας συνάρτησης είναι η ακόλουθη :



Το όνομα της συνάρτησης πρέπει να είναι έγκυρο σύμφωνα με τους κανόνες ονομάτων της ΓΛΩΣΣΑΣ.

Η λίστα παραμέτρων είναι μια λίστα από μεταβλητές, στις οποίες μεταβιβάζονται τιμές από το κυρίως πρόγραμμα κατά την κλήση της συνάρτησης.

Επειδή η συνάρτηση επιστρέφει την τιμή της στο κυρίως πρόγραμμα μέσα από το όνομά της, θα πρέπει στο σώμα της να περιέχει μια εντολή εκχώρησης τιμής στο όνομα της συνάρτησης (δηλαδή **όνομα ← έκφραση**).

ΣΗΜΕΙΩΣΗ

Μια συνάρτηση μπορεί να κληθεί ακριβώς στα ίδια σημεία του προγράμματος που καλούνται οι ενσωματωμένες μαθηματικές συναρτήσεις της ΓΛΩΣΣΑΣ, δηλαδή:

- ♦ εντολές εκχώρησης π.χ. $X \leftarrow \text{συναρ}(A, B)$
- ♦ εντολές εξόδου π.χ. ΓΡΑΨΕ 'Αποτέλεσμα = ', $\text{συναρ}(A, B)$
- ♦ λογικές εκφράσεις π.χ. $\text{ΑΝ } \text{συναρ}(A, B) \geq 12 \text{ ΤΟΤΕ } \dots$

ΠΑΡΑΔΕΙΓΜΑ

κλήση συνάρτησης



```

ΠΡΟΓΡΑΜΜΑ Παράδειγμα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A, B, M
ΑΡΧΗ
    ΔΙΑΒΑΣΕ A, B
    M ← MAX(A, B)
    ΓΡΑΨΕ M
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
    
```

```

ΣΥΝΑΡΤΗΣΗ MAX(X, Y) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: X, Y
ΑΡΧΗ
    ΑΝ X > Y ΤΟΤΕ
        MAX ← X
    ΑΛΛΙΩΣ
        MAX ← Y
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
    
```

ΔΙΑΔΙΚΑΣΙΑ (Procedure)

Η **διαδικασία** είναι τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος (είσοδο δεδομένων, εκτύπωση αποτελεσμάτων, τροποποίηση τιμής μεταβλητών κλπ).

Η διαδικασία μεταβιβάζει τιμές στα άλλα υποπρογράμματα με χρήση παραμέτρων και όχι του ονόματός της

Το όνομα της διαδικασίας πρέπει να είναι έγκυρο σύμφωνα με τους κανόνες της ΓΛΩΣΣΑΣ. Η βασική δομή μιας διαδικασίας είναι η ακόλουθη :

↓ ↑ ↑

```

ΔΙΑΔΙΚΑΣΙΑ όνομα (λίστα παραμέτρων)
Τμήμα δηλώσεων
ΑΡΧΗ
    .....
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

Από την λίστα παραμέτρων (μεταβλητών), κάποιες χρησιμοποιούνται για να μεταβιβάσουν τιμές μέσα στη διαδικασία από την κλήση της και κάποιες για να επιστρέψουν τιμές μέσα στο κυρίως πρόγραμμα.

Να σημειωθεί ότι η λίστα παραμέτρων στην διαδικασία δεν είναι υποχρεωτική.
Η κλήση της διαδικασίας γίνεται ξεχωριστή εντολή με την δεσμευμένη λέξη **ΚΑΛΕΣΕ** που ακολουθείται από το όνομα και την λίστα παραμέτρων της διαδικασίας:

ΚΑΛΕΣΕ όνομα διαδικασίας (λίστα παραμέτρων)

Με την κλήση της διαδικασίας, η εκτέλεση του κυρίως προγράμματος **διακόπτεται** και εκτελούνται οι εντολές του σώματος της διαδικασίας.

Όταν ολοκληρωθεί η εκτέλεση των εντολών της διαδικασίας, η εκτέλεση του προγράμματος **συνεχίζεται** από την επόμενη εντολή (επιστροφή στο σημείο κλήσης).

ΠΑΡΑΔΕΙΓΜΑ

κλήση διαδικασίας



ΠΡΟΓΡΑΜΜΑ Παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, μεγ, ελαχ

ΑΡΧΗ

ΔΙΑΒΑΣΕ A, B

ΚΑΛΕΣΕ TEST (A, B, μεγ, ελαχ)

ΓΡΑΨΕ μεγ

ΓΡΑΨΕ ελαχ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ TEST (X, Y, max, min)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X, Y, max, min

ΑΡΧΗ

ΑΝ X > Y **ΤΟΤΕ**

max ← X

min ← Y

ΑΛΛΙΩΣ

max ← Y

min ← X

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΠΡΑΓΜΑΤΙΚΕΣ & ΤΥΠΙΚΕΣ ΠΑΡΑΜΕΤΡΟΙ

Κατά την κλήση ενός υποπρογράμματος γίνεται πέρασμα τιμών από τις παραμέτρους του κυρίως προγράμματος στο υποπρόγραμμα που καλείται.

Οι παράμετροι που βρίσκονται στην εντολή κλήσης αποτελούν την λίστα **πραγματικών παραμέτρων** (formal parameters), ενώ αυτές που βρίσκονται μέσα στο υποπρόγραμμα που καλείται αποτελούν την λίστα **τυπικών παραμέτρων** (actual parameters).

Οι πραγματικές παράμετροι δηλώνονται πάντα μέσα στο κυρίως πρόγραμμα ενώ οι τυπικές μέσα στο υποπρόγραμμα στο οποίο χρησιμοποιούνται.

Τα ονόματα των πραγματικών και των αντίστοιχων τυπικών παραμέτρων μπορεί να είναι και τα ίδια (όμως οι μεταβλητές είναι διαφορετικές).

Όλες οι μεταβλητές ισχύουν τοπικά, δηλαδή μόνο μέσα στο τμήμα προγράμματος στο οποίο έχουν δηλωθεί.

Σε κάποιες γλώσσες προγραμματισμού οι τυπικές παράμετροι ονομάζονται **ορίσματα**, ενώ οι πραγματικές παράμετροι απλά **παραμέτροι**.

Η **λίστα τυπικών παραμέτρων** καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος, ενώ η **λίστα των πραγματικών παραμέτρων** καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.

Οι **λίστες των παραμέτρων** πρέπει να ακολουθούν τους εξής **κανόνες**:

- **Ίδιος αριθμός** των πραγματικών και των τυπικών παραμέτρων.
- Κάθε πραγματική παράμετρος αντιστοιχίζεται με την τυπική παράμετρο που βρίσκεται στην **αντίστοιχη θέση** μέσα στη λίστα παραμέτρων του υποπρογράμματος.
- Οι αντίστοιχες τυπικές και πραγματικές παράμετροι (που συνδέονται μεταξύ τους κατά την κλήση) πρέπει να είναι του **ιδίου τύπου** δεδομένων.

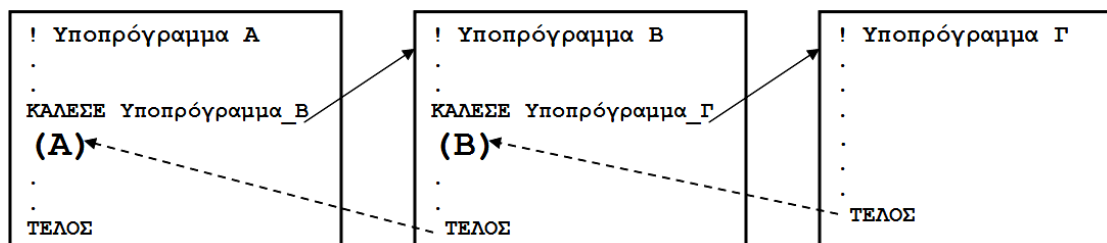
ΚΛΗΣΗ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ ΜΕ ΧΡΗΣΗ ΣΤΟΙΒΑΣ

Όταν ένα υποπρόγραμμα καλείται από το κυρίως πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του προγράμματος, γνωστή και ως **διεύθυνση επιστροφής** (return address), αποθηκεύεται (ωθείται - push) προσωρινά από τον μεταφραστή σε μία στοίβα που ονομάζεται **στοίβα χρόνου εκτέλεσης** (execution time stack).

Όταν το υποπρόγραμμα ολοκληρώνεται, η **διεύθυνση επιστροφής απωθείται** (pop) από την στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κυρίως πρόγραμμα.

Με τον τρόπο αυτό το κυρίως πρόγραμμα μπορεί να θυμάται το σημείο στο οποίο θα πρέπει να επιστρέψει μετά την ολοκλήρωση του υποπρογράμματος, έτσι ώστε να συνεχίσει την εκτέλεση των εντολών του π.χ.

Έστω ότι το υποπρόγραμμα Α καλεί το υποπρόγραμμα Β που καλεί το υποπρόγραμμα Γ. Οι διευθύνσεις επιστροφής των υποπρογραμμάτων Α και Β ωθούνται με την σειρά σε μία στοίβα και απωθούνται με την αντίστροφη σειρά κατά την φάση της εκάστοτε επιστροφής στο σημείο της κλήσης.



Μετά το τέλος της εκτέλεσης της διαδικασίας ή της συνάρτησης γίνεται επιστροφή ακριβώς μετά το σημείο απ' όπου κλήθηκε.

ΕΜΒΕΛΕΙΑ ΜΕΤΑΒΛΗΤΩΝ - ΣΤΑΘΕΡΩΝ

Κάθε κύριο πρόγραμμα όπως και κάθε υποπρόγραμμα περιλαμβάνει τις δικές του μεταβλητές και σταθερές.

Όλες οι μεταβλητές στη ΓΛΩΣΣΑ είναι γνωστές μόνο στο αντίστοιχο υποπρόγραμμα που δηλώνονται. Θεωρούνται **τοπικές** στο συγκεκριμένο τμήμα προγράμματος και άρα ισχύουν μόνο μέσα σε αυτό.

Ο μόνος τρόπος για να περάσει μία τιμή από ένα υποπρόγραμμα σε ένα άλλο ή από το κυρίως πρόγραμμα σε ένα υποπρόγραμμα είναι διαμέσου των παραμέτρων κατά το στάδιο της κλήσης του υποπρογράμματος και μετά το τέλος της εκτέλεσης του υποπρογράμματος.

Εμβέλεια μεταβλητών (variables scope) ονομάζεται το συγκεκριμένο τμήμα του προγράμματος που ισχύουν οι μεταβλητές.

Πολλές γλώσσες προγραμματισμού επιτρέπουν τη χρήση των μεταβλητών και των σταθερών, όχι μόνο στο τμήμα προγράμματος που δηλώνονται, αλλά και σε άλλα ή ακόμη και σε όλα τα υπόλοιπα υποπρογράμματα.

ΑΠΕΡΙΟΡΙΣΤΗ ΕΜΒΕΛΕΙΑ	ΠΕΡΙΟΡΙΣΜΕΝΗ ΕΜΒΕΛΕΙΑ	ΜΕΡΙΚΩΣ ΠΕΡΙΟΡΙΣΜΕΝΗ ΕΜΒΕΛΕΙΑ
- Όλες οι μεταβλητές και σταθερές είναι γνωστές σε οποιοδήποτε τμήμα του προγράμματος. - Συνεπώς, μπορούν να χρησιμοποιούνται σε κάθε τμήμα άσχετα από το που έχουν δηλωθεί. - Οι μεταβλητές και σταθερές αυτές ονομάζονται καθολικές (global). - Βασικό μειονέκτημα είναι ότι η χρήση τους καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων. - Δημιουργεί πολλά προβλήματα όντας αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα.	- Όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, πρέπει να δηλώνονται σε αυτό το τμήμα. - Όλες οι μεταβλητές είναι τοπικές (local), δηλαδή ισχύουν μόνο για το υποπρόγραμμα στο οποίο δηλώθηκαν. - Βασικό πλεονέκτημα είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων. - Δυνατότητα χρήσης οποιουδήποτε ονόματος, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.	- Κάποιες μεταβλητές είναι τοπικές και κάποιες άλλες είναι καθολικές. - Κάθε γλώσσα έχει τους δικούς της κανόνες για τον τρόπο καθορισμού των μεταβλητών ως τοπικές ή καθολικές. - Αυτή η μορφή εμβέλειας δίνει πλεονεκτήματα στον έμπειρο προγραμματιστή. - Στον αρχάριο προγραμματιστή καθιστά δυσκολότερη την ανάπτυξη του προγράμματος.