

23. ΔΙΑΔΙΚΑΣΙΕΣ – ΕΜΒΕΛΕΙΑ ΜΕΤΑΒΛΗΤΩΝ (Θ Κεφ. 6.4 & 10) (Ο Κεφ. 5.1)

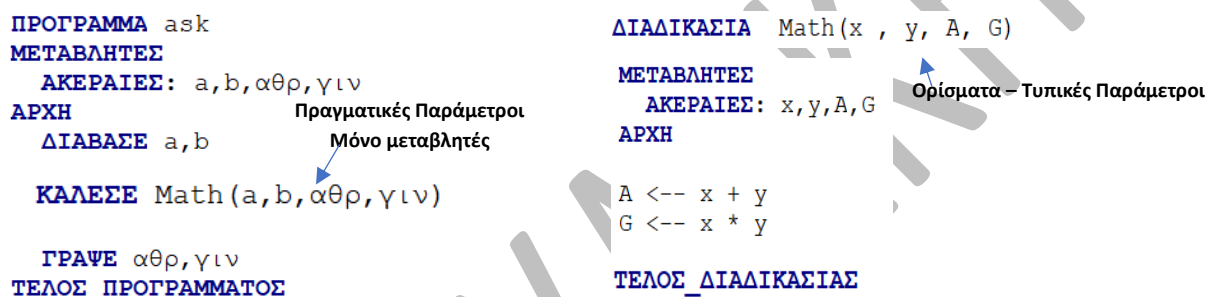
Η **διαδικασία** είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος.

Σύνταξη

ΔΙΑΔΙΚΑΣΙΑ Όνομα (λίστα παραμέτρων)
 Τμήμα δηλώσεων μεταβλητών διαδικασίας
ΑΡΧΗ
 Εντολές
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Παράδειγμα 1

Να γραφεί ένα πρόγραμμα το οποίο διαβάζει τις τιμές 2 μεταβλητών στην συνέχεια χρησιμοποιεί ένα υποπρόγραμμα το οποίο υπολογίζει και επιστρέφει το άθροισμα και το γινόμενο των αριθμών.



Κατά την κλήση της διαδικασίας μια προς μια οι πραγματικές παράμετροι του προγράμματος αντιγράφουν τις τιμές τους στις τυπικές παραμέτρους της συνάρτησης.

a	→	x
b	→	y
αθρ	→	A
γιν	→	G

a	←	x
b	←	y
αθρ	←	A
γιν	←	G

Στο τέλος της διαδικασίας μια προς μια οι τυπικές παράμετροι της διαδικασίας αντιγράφουν τις τιμές τους στις πραγματικές παραμέτρους του προγράμματος.

Ιδιαίτερη προσοχή χρειάζονται τα παρακάτω σημεία:

- **Τοπικότητα μεταβλητών.** Οι μεταβλητές του κυρίου προγράμματος, είναι γνωστές μόνο στο πρόγραμμα. Έξω από το πρόγραμμα, κανένα υποπρόγραμμα δεν τις γνωρίζει και δεν μπορεί να τις χρησιμοποιήσει. Το ίδιο ισχύει και για τις μεταβλητές των υποπρογραμμάτων. Για αυτό λέμε, ότι οι μεταβλητές έχουν ισχύ τοπική, δηλαδή **περιορισμένη εμβέλεια** εντός του προγράμματος/υποπρογράμματος που δηλώνονται.
 Αν και υπάρχουν μεταβλητές με απεριόριστη (καθολική) ή μερικώς περιορισμένη εμβέλεια δεν είναι κάτι που συνιστάται αφού με αυτό τον τρόπο τα υποπρόγραμμα χάνουν την αυτονομία τους.
- **Επικοινωνία υποπρογραμμάτων.** Ο μοναδικός τρόπος επικοινωνίας είναι μέσω των παραμέτρων (πραγματικών-τυπικών). Μία παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.
Αντιστοιχία παραμέτρων - κανόνες τυπικών και πραγματικών παραμέτρων.
 1. Ο αριθμός των πραγματικών και τυπικών παραμέτρων πρέπει να είναι ο ίδιος.
 2. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση.
 3. Η τυπική παράμετρος και η αντίστοιχη πραγματική παράμετρος πρέπει να είναι του ίδιου τύπου.
- **Χωροθέτηση.** Πρώτα πάντα είναι γραμμένο το κύριο πρόγραμμα. Έπειτα ακολουθούν τα υποπρόγραμματα.
- **Εκτέλεση.** Η εκτέλεση ξεκινά πάντα από το κύριο πρόγραμμα, και προφανώς τελειώνει στο κύριο πρόγραμμα.

Παράδειγμα 2

Να γραφεί ένα πρόγραμμα που με την βοήθεια μιας συνάρτησης ή μιας διαδικασίας, να υπολογίζει την παρακάτω μαθηματική παράσταση για $X > 0$

$$g(x) = x^2 + x^3 - \frac{\sqrt{2+x^3}}{x+3*\sqrt{x}}$$

```

ΔΙΑΔΙΚΑΣΙΑ  ΥΠΟΛ (x,g)
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: g,x
ΑΡΧΗ
  g <-- x^2 + x^3 - T_P(2+x^3) / (x+3*T_P(x))
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

```

ΣΥΝΑΡΤΗΣΗ  ΥΠΟΛ (x) : ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: x
ΑΡΧΗ
  ΥΠΟΛ <-- x^2 + x^3 - T_P(2+x^3) / (x+3*T_P(x))
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
    
```

```

ΠΡΟΓΡΑΜΜΑ ΤΕΣΤ1
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: x,ap
ΑΡΧΗ
  ΔΙΑΒΑΣΕ x
  ΚΑΛΕΣΕ ΥΠΟΛ (x,ap)
  ΓΡΑΨΕ ap
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
    
```

```

ΠΡΟΓΡΑΜΜΑ ΤΕΣΤ1
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: x,ap
ΑΡΧΗ
  ΔΙΑΒΑΣΕ x
  ap <-- ΥΠΟΛ (x)
  ΓΡΑΨΕ ap
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
    
```

Παράδειγμα 3

Να γράψετε ένα υποπρόγραμμα το οποίο εμφανίζει στην οθόνη το παρακάτω μενού επιλογών.

1. Είσοδος Πελάτη
 2. Έξοδος Πελάτη
 3. Υπολογισμός Κερδών
 4. Έξοδος προγράμματος
- Επιλογή >

```

ΔΙΑΔΙΚΑΣΙΑ  menu ()
ΑΡΧΗ
  ΓΡΑΨΕ '1. Είσοδος Πελάτη'
  ΓΡΑΨΕ '2. Έξοδος Πελάτη'
  ΓΡΑΨΕ '3. Υπολογισμός Κερδών'
  ΓΡΑΨΕ '4. Έξοδος Προγράμματος'
  ΓΡΑΨΕ 'Επιλογή >'
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

```

ΠΡΟΓΡΑΜΜΑ ΤΕΣΤ
ΑΡΧΗ
  ΚΑΛΕΣΕ menu ()
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
    
```

Διαφορές Διαδικασίας/Συναρτήσης

Διαδικασίες	Συναρτήσεις
Οι διαδικασίες μπορούν να εκτελέσουν μια οποιαδήποτε λειτουργία, π.χ. να εισάγουν δεδομένα, να εκτελούν υπολογισμούς, να τυπώνουν αποτελέσματα και να αλλάζουν τις τιμές των μεταβλητών.	Οι συναρτήσεις υπολογίζουν μόνο μία τιμή και μόνο αυτή επιστρέφουν στο κύριο πρόγραμμα ή στο υποπρόγραμμα που τις κάλεσε. Η τιμή αυτή μπορεί να είναι ΑΡΙΘΜΗΤΙΚΗ (ΑΚΕΡΑΙΑ ή ΠΡΑΓΜΑΤΙΚΗ), ΧΑΡΑΚΤΗΡΑΣ ή ΛΟΓΙΚΗ.
Οι διαδικασίες μεταφέρουν τα αποτελέσματα στα άλλα υποπρογράμματα με τη χρήση παραμέτρων.	Οι συναρτήσεις μεταφέρουν το αποτέλεσμα στο κύριο πρόγραμμα ή στο υποπρόγραμμα που τις κάλεσε με το όνομά τους και όχι με τη χρήση παραμέτρων. Μοιάζουν με τις μαθηματικές συναρτήσεις.
Οι διαδικασίες εκτελούνται αν γράψουμε την εντολή ΚΑΛΕΣΕ και μετά το όνομα της διαδικασίας.	Οι συναρτήσεις εκτελούνται με τη χρήση του ονόματός τους μέσα σε οποιαδήποτε εντολή.

Εμβέλεια μεταβλητών-σταθερών

Απεριόριστη εμβέλεια: Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν. Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

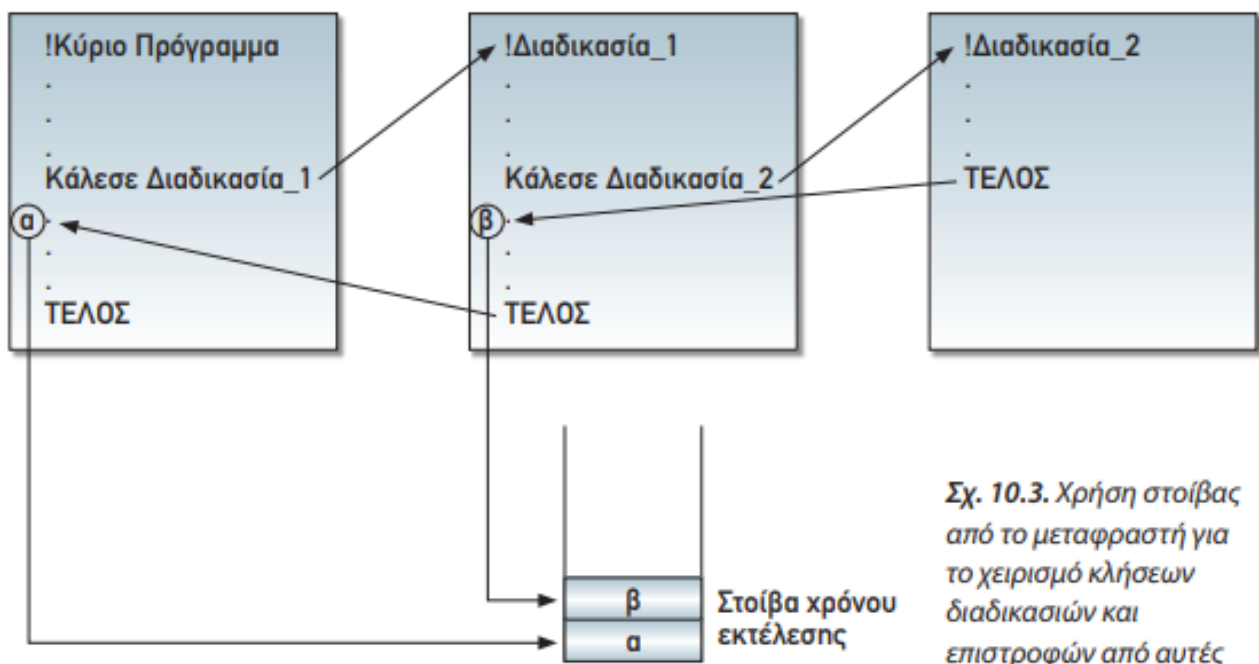
Περιορισμένη εμβέλεια: Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι τοπικές, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια. Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

Μερικώς περιορισμένη εμβέλεια: Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι τοπικές και άλλες καθολικές. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται οι μεταβλητές ως τοπικές ή καθολικές. Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

Στοίβα κλήσης υποπρογραμμάτων

Η ΧΡΗΣΗ ΣΤΟΙΒΑΣ ΣΤΗΝ ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΩΝ

Η έννοια της στοίβας είναι πολύ χρήσιμη στο ίδιο το λογισμικό των γλωσσών προγραμματισμού. Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται *διεύθυνση επιστροφής* (return address), αποθηκεύεται από το μεταφραστή σε μία στοίβα που ονομάζεται *στοίβα χρόνου εκτέλεσης* (execution time stack). Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Η τεχνική αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μία διαδικασία ή συνάρτηση καλεί μία διαδικασία ή συνάρτηση. Για παράδειγμα, έστω ότι μία διαδικασία *a* καλεί τη διαδικασία *b*, που με τη σειρά της καλεί τη διαδικασία *c* κ.ο.κ. Στην περίπτωση αυτή οι διευθύνσεις επιστροφής εμφανίζονται στη στοίβα με σειρά *c*, *b*, *a*. Μετά την εκτέλεση κάθε διαδικασίας, η διεύθυνση επιστροφής απωθείται από τη στοίβα και ο έλεγχος μεταβιβάζεται στη διεύθυνση αυτή. Το παράδειγμα αυτό δείχνει μία από τις πολλές χρησιμότητες της LIFO ιδιότητας της στοίβας.



Σχ. 10.3. Χρήση στοίβας από το μεταφραστή για το χειρισμό κλήσεων διαδικασιών και επιστροφών από αυτές