

ΚΕΦΑΛΑΙΟ 4

ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ

Γενικά για τις Διαδικασίες και τις Συναρτήσεις

Όπως όλες οι γλώσσες προγραμματισμού, έτσι και ο Clipper χρησιμοποιεί κατά κόρο τις διαδικασίες (procedures) και τις συναρτήσεις (functions), που περιγράφονται και με τον γενικό όρο *υπορουτίνες* (subroutines). Οι υπορουτίνες διευκολύνουν τον δομημένο προγραμματισμό και κάνουν τα προγράμματά μας ευκολότερα στη γραφή, στον έλεγχο και στην κατανόηση.

Η υπορουτίνα είναι απλά ένα μικρό, αλλά μερικές φορές και μεγάλο, ευέλικτο πρόγραμμα γραμμένο στη γλώσσα του Clipper. Το βασικό πρόγραμμα που γράφουμε στον Clipper έχει τη δυνατότητα να καλεί αυτές τις υπορουτίνες για να εκτελέσει εργασίες που επαναλαμβάνονται πολλές φορές μέσα στο πρόγραμμα.

Η κάθε υπορουτίνα εκτελεί μια συγκεκριμένη εργασία, όπως π.χ. εκτύπωση κάποιων δεδομένων, υπολογισμό πάνω σε κάποια στοιχεία κ.ά. και μπορεί να επικοινωνεί με το κύριο πρόγραμμα που γράφουμε καθώς και μ' άλλες υπορουτίνες, με τις οποίες μπορεί να ανταλλάσσει δεδομένα.

Η κύρια διαφορά μεταξύ των διαδικασιών (procedures) και των συναρτήσεων (functions) είναι ότι οι συναρτήσεις επιστρέφουν *μία και μόνο μία τιμή* με το όνομά τους στο πρόγραμμα ή την υπορουτίνα που τις κάλεσε, ενώ οι διαδικασίες μπορεί να μην επιστρέψουν καμία τιμή, αλλά μπορεί να επιστρέψουν και περισσότερες από μία τιμές στο πρόγραμμα ή την υπορουτίνα που τις κάλεσε.

Τόσο οι διαδικασίες, όσο και οι συναρτήσεις μπορούν να δεχθούν ορίσματα, τιμές δηλ. μεταβλητών που τους μεταβιβάζονται και μπορούν να επιστρέψουν αυτές τις τιμές αλλαγμένες.

Οι Διαδικασίες (Procedures)

Όπως ήδη είπαμε, οι διαδικασίες είναι μικρά προγράμματα που είτε δεν επιστρέφουν καμία τιμή ή επιστρέφουν περισσότερες από δύο τιμές στο πρόγραμμα ή την υπορουτίνα που τις κάλεσε.

Η γενική σύνταξη μιας διαδικασίας είναι ως εξής :

```
PROCEDURE <όνομα> [<λίστα παραμέτρων>]  
    <δηλώσεις μεταβλητών>  
    <εντολές>  
[RETURN]
```

Κάθε διαδικασία έχει ένα μοναδικά ορισμένο όνομα με βάση το οποίο και καλείται. Η λίστα παραμέτρων μιας διαδικασίας είναι προαιρετική. Κάθε διαδικασία έχει και δικές της (τοπικές) μεταβλητές, εκτός από τα ορίσματα (παραμέτρους) με τα οποία επικοινωνεί (ανταλλάσσει δεδομένα) με το κύριο πρόγραμμα ή με τις άλλες υπορουτίνες.

Η δήλωση RETURN δεν είναι υποχρεωτική σε μια διαδικασία, αλλά είναι καλό να υπάρχει για να δείχνει πού τελειώνει η διαδικασία. Παραδείγματα με διαδικασίες είδαμε στο προηγούμενο κεφάλαιο.

Οι Συναρτήσεις (Functions)

Όπως ήδη είπαμε, οι συναρτήσεις είναι μικρά προγράμματα που επιστρέφουν μία και μόνο μία τιμή στο πρόγραμμα ή την υπορουτίνα που τις κάλεσε.

Η γενική σύνταξη μιας συνάρτησης είναι ως εξής :

```
FUNCTION <όνομα> [<λίστα παραμέτρων>]
    <δηλώσεις μεταβλητών>
    <εντολές>
RETURN <επιστρεφόμενη τιμή ή έκφραση>
```

Η δήλωση RETURN είναι υποχρεωτική στις συναρτήσεις.

Δείτε την παρακάτω συνάρτηση που επιστρέφει τη μεγαλύτερη από δύο μεταβλητές.

```
FUNCTION Greater(var1, var2)
    LOCAL Greatest

    IF var1 > var2
        Greatest := var1
    ELSE
        Greatest := var2
    ENDIF
RETURN (Greatest)
```

Σημειώστε ότι οι παράμετροι (ορίσματα) που δέχεται η συνάρτηση Greater μπορεί να είναι οποιουδήποτε τύπου δεδομένων. Αυτό είναι ένα από πιο τα αξιοσημείωτα χαρακτηριστικά του Clipper, το ότι δηλ. οι μεταβλητές μπορούν να αλλάζουν τύπο δεδομένων μέσα σ' ένα πρόγραμμα και το ότι δεν είναι υποχρεωτικό να δηλώνουμε πάντα τον τύπο δεδομένων των μεταβλητών.

Σε γενικές γραμμές, οι δηλώσεις RETURN σε μια συνάρτηση μπορούν να περιλαμβάνουν οποιαδήποτε παραδεκτή από τον Clipper τιμή ή έκφραση. Επειδή η δήλωση RETURN, εκτός από την επιστροφή της τιμής, τερματίζει και τη συνάρτηση, μπορούμε να έχουμε περισσότερες από μία δηλώ-

σεις RETURN στο εσωτερικό μιας συνάρτησης. Η συνάρτηση θα τερματίσει και θα επιστρέψει τον έλεγχο από εκεί που κλήθηκε, όταν συναντήσει την πρώτη δήλωση RETURN.

Δείτε το παρακάτω παράδειγμα :

```
FUNCTION Check_Type(nVar)
  IF VALTYPE(nVar) = "C"
    RETURN ("Αλφαριθμητικό"), + RTRIM(nVar))
  ENDIF
RETURN ("Δεν αποτελεί αλφαριθμητικό")
```

Η παραπάνω συνάρτηση δέχεται σαν όρισμα την τιμή της nVar και ελέγχει αν αυτή είναι του τύπου δεδομένων χαρακτήρα (Character). Αν ναι, τότε επιστρέφει το μήνυμα που είναι στην τρίτη γραμμή, αλλιώς επιστρέφει την πρόταση που είναι στο τέλος της συνάρτησης.

Προσέξτε ότι η παραπάνω συνάρτηση καλεί κι αυτή δύο ενσωματωμένες συναρτήσεις του Clipper, την VALTYPE() και την RTRIM(). Η VALTYPE() δέχεται ένα όρισμα και επιστρέφει τον τύπο δεδομένων του, εδώ μας ενδιαφέρει βέβαια το "C", που είναι για τους χαρακτήρες, ενώ η RTRIM() δέχεται ένα αλφαριθμητικό σαν όρισμα και το επιστρέφει χωρίς τα κενά διαστήματα που τυχόν υπάρχουν στο τέλος του.

Κλήση των Διαδικασιών και των Συναρτήσεων

Οι διαδικασίες και οι συναρτήσεις καλούνται με τον ίδιο τρόπο. Γράφουμε το όνομά τους και μέσα στις παρενθέσεις τα ορίσματα που δέχονται χωρισμένα με κόμμα. Αν δεν δέχονται κανένα όρισμα, τότε πρέπει να γράψουμε τις παρενθέσεις κενές.

Επειδή οι συναρτήσεις επιστρέφουν μία τιμή, χρησιμοποιούνται συνήθως για την εκχώρηση τιμών σε μια παράσταση, ενώ οι διαδικασίες καλούνται σαν εντολές (δηλώσεις) με την απλή αναγραφή του ονόματός τους.

Δείτε τα παρακάτω παραδείγματα :

```
Cost := Greater(Cost1, Cost2)
/* κλήση μιας συνάρτησης και καταχώρηση της τιμής που επιστρέφει
στη μεταβλητή Cost */
```

```
SETCOLOR("B/G", "W+/N")
/* κλήση της συνάρτησης SETCOLOR() του Clipper για την αλλαγή των
χρωμάτων. Εδώ η ίδια η συνάρτηση λειτουργεί σαν δήλωση */
```

```
Zip_Code := LEFT(q_zip, 5) + "-" + RIGHT(q_zip, 4)
/* καλεί τις ενσωματωμένες συναρτήσεις του Clipper LEFT() και
RIGHT() */
```

```
My_Proc()
/* κλήση μιας διαδικασίας που δεν έχει ορίσματα */
```

```
Samp_Proc(Samp1, Samp2, Samp3)
/* διαδικασία με τρία ορίσματα */
```

Παράμετροι, Ορίσματα και Μεταβίβαση Τιμών

Όπως είδαμε, κατά την κλήση μιας διαδικασίας ή συνάρτησης, μπορούμε να μεταβιβάσουμε κάποιες τιμές σ' αυτήν. Οι τιμές που μεταβιβάζονται σε μια διαδικασία ή συνάρτηση λέγονται *ορίσματα* (*arguments*), ενώ οι μεταβλητές που θα δεχθούν τις τιμές των ορισμάτων καλούνται *παράμετροι* (*parameters*). Σε μια υπορουτίνα μπορούμε να μεταβιβάσουμε μέχρι και 128 ορίσματα.

Οι παράμετροι θεωρούνται τοπικές (local) μεταβλητές της υπορουτίνας όπου ανήκουν. Μπορούμε να παραλείψουμε τη μεταβίβαση μιας τιμής σε μια παράμετρο, αφήνοντας απλά κενό ανάμεσα στα αντίστοιχα κόμματα.

Δείτε την παρακάτω κλήση μιας διαδικασίας, όπου δεν μεταβιβάζουμε τιμή στη δεύτερη παράμετρο :

```
Samp_Proc(Samp1, , Samp3)
```

Στις παραμέτρους στις οποίες δεν αντιστοιχεί μεταβιβαζόμενη τιμή, εκχωρείται η τιμή NIL.

Ο εξ ορισμού τρόπος μεταβίβασης παραμέτρων είναι μέσω της τιμής τους (*κλήση κατ' αξία* – *call by value*). Αυτό σημαίνει ότι στην καλούμενη υπορουτίνα δεν μεταβιβάζεται το πραγματικό όρισμα, αλλά ένα αντίγραφο της μεταβιβαζόμενης μεταβλητής, το οποίο και χρησιμοποιείται από την καλούμενη υπορουτίνα. Έτσι, η υπορουτίνα δεν μπορεί να αλλάξει την πρωτότυπη μεταβλητή.

Η γενική μορφή κλήσης κατ' αξία είναι ως εξής :

- Για διαδικασία : DO Subroutine WITH (x) ή Subroutine(x)
- Για συνάρτηση : Subroutine(x)

Ο δεύτερος τρόπος μεταβίβασης παραμέτρων είναι με αναφορά (*κλήση κατ' αναφορά* – *call by reference*). Όταν μεταβιβάζουμε μια παράμετρο με αναφορά σε κάποια υπορουτίνα, περνάμε μια αναφορά (ή δείκτη) στη διεύθυνση της μνήμης όπου βρίσκεται αποθηκευμένη η ίδια η μεταβιβαζόμενη τιμή και όχι ένα αντίγραφό της.

Αυτό σημαίνει πρακτικά ότι, αν η καλούμενη υπορουτίνα τροποποιήσει την τιμή της παραμέτρου, θα αλλάξει και η τιμή της πρωτότυπης μεταβλητής.

Η γενική μορφή κλήσης κατ' αναφορά είναι ως εξής :

- Για διαδικασία : DO Subroutine WITH x ή Subroutine(@x)
- Για συνάρτηση : Subroutine(@x)

Για να μεταβιβάσουμε μια μεταβλητή με αναφορά, πρέπει να τοποθετήσουμε μπροστά από το όνομά της τον τελεστή @, όταν καλούμε την υπορουτίνα που την περιέχει.

Δείτε το παρακάτω παράδειγμα :

```
LOCAL Name := "Fred Flintstone"  
LOCAL nSuffix := "Jr."  
Join_Name(@Name, nSuffix)  
? Name
```

```
PROCEDURE Join_Name(First, Add_On)  
  IF Add_On <> NIL  
    First := First + "," + Add_On  
  ENDIF  
RETURN
```

Η διαδικασία Join_Name έχει δύο παραμέτρους, από τις οποίες η First δέχεται την τιμή του ορίσματος Name με αναφορά (χρήση του τελεστή @) και η Add_On δέχεται την τιμή του ορίσματος nSuffix με τιμή. Έτσι, αν αλλάξει η First μέσα στην υπορουτίνα, την ίδια αλλαγή θα υποστεί και η Name.

Η μεταβίβαση τιμών με αναφορά θα πρέπει να χρησιμοποιείται σπάνια και με μεγάλη προσοχή και μόνο σε εξαιρετικά αναγκαίες περιπτώσεις.