

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΜΟΝΑΔΕΣ & ΟΡΙΣΜΑΤΑ

Το ανώτερο επίπεδο προγραμματιστικής μονάδας στη Fortran είναι το **Κυρίως Πρόγραμμα (Main Program)**. Οι μεγάλες εφαρμογές δεν μπορούν να γραφούν σαν ένα μονοκόμματο κυρίως πρόγραμμα και έτσι χωρίζονται σε **υποπρογράμματα**. Όπως είδαμε και στο πρώτο μέρος, τα υποπρογράμματα αυτά είναι τριών ειδών, οι υπορουτίνες (**Subroutine**), οι συναρτήσεις (**Function**) και οι αυτοτελείς λογισμικές μονάδες (**Module**). Τα δύο πρώτα έχουν ήδη συζητηθεί. Στη συνέχεια θα εξηγήσουμε τις αυτοτελείς λογισμικές μονάδες (modules) και γενικότερα θέματα που αφορούν τα υποπρογράμματα (procedures), τις εντολές κλήσης τους, και, το πέρασμα τιμών και δεδομένων μεταξύ τους.

MODULES

Ένα ή περισσότερα συναφή υποπρογράμματα μαζί με τυχόν σφαιρικές δηλώσεις τύπων, μπορούν να ενωθούν σε μια ενότητα η οποία καλείται Module. Σε μεγάλες εφαρμογές συναντάμε και ένα απλούστερο είδος Module που περιέχει μόνο δηλώσεις. Οι ενιαίες αυτές προγραμματιστικές μονάδες δηλώνονται με την εντολή MODULE ... END.

Module Δηλώσεων

Τα απλούστερα Module είναι αυτά που περιέχουν μόνο δηλωτικές εντολές. Τα module αυτά ενεργοποιούνται στην αρχή άλλων προγραμμάτων ή υποπρογραμμάτων με τη εντολή USE, πριν την εντολή IMPLICIT NONE. Π.χ.

```
MODULE StatheresParametroi
  IMPLICIT NONE
  REAL, PARAMETER:: pi=3.14159, c=3.0E8
END MODULE

PROGRAM
  USE StatheresParametroi
  IMPLICIT NONE
  ...
END
```

Module Υποπρογραμμάτων

Τα σύνθετα Module περιέχουν εκτός των δηλωτικών εντολών, υπορουτίνες και συναρτήσεις που μπορούν να κληθούν από άλλα προγράμματα ή υποπρογράμματα.

```
MODULE Routines
  REAL:: A, B, C, GAMMA
  CONTAINS
  SUBROUTINE ALFA(A,B)
  ...
  END SUBROUTINE
  SUBROUTINE BETA(B,C)
  ...
  END SUBROUTINE
  FUNCTION GAMMA(A,B,C)
  ...
  END FUNCTION
END MODULE

PROGRAM
  USE Routines
  ...
  CALL ALFA(A,B)
  ...
  CALL BETA(B,C)
  ...
  G = GAMMA(A,B,C)
  ...
END
```

PROCEDURES (Υποπρογράμματα)

Subroutine

Η υπορουτίνα είναι ένα υποπρόγραμμα που καλείται με τη **CALL**, και η σύνδεση της με το καλούν πρόγραμμα (δεδομένα & αποτελέσματα) γίνεται με τα ορίσματα στη παρένθεση που ακολουθεί το όνομά της. Οι υπορουτίνες δηλώνονται με την εντολή **SUBROUTINE ... END**. Μια υπορουτίνα μπορεί να χαρακτηρίζεται σαν **RECURSIVE** (όταν επιτρέπεται η αναδρομική κλήση της), **PURE** (ότι δεν δημιουργεί παρενέργειες στη ροή ή τη κατάσταση του προγράμματος), ή, **ELEMENTAL** (μια pure που μπορεί να εφαρμόζεται σε μητρώα στοιχείο-στοιχείο).

Function

Η συνάρτηση (του χρήστη) είναι μια απλουστευμένη subroutine που καλείται με το όνομά της, και επιστρέφει μέσω αυτού το αποτέλεσμα. Τα δεδομένα της δίνονται από το καλούν πρόγραμμα με τα ορίσματα στη παρένθεση που ακολουθεί το όνομά της. Οι συναρτήσεις δηλώνονται με την εντολή **FUNCTION ... END**. Η χρήση μιας συνάρτησης στις αριθμητικές πράξεις μοιάζει με αυτή των μεταβλητών, γι' αυτό ο τύπος της συνάρτησης πρέπει να έχει δηλωθεί μαζί με τις μεταβλητές. Μια συνάρτηση μπορεί να χαρακτηρίζεται και σαν **RECURSIVE** (όταν επιτρέπεται η αναδρομική κλήση της), **PURE** (ότι δεν δημιουργεί παρενέργειες στη ροή ή τη κατάσταση του προγράμματος), ή, **ELEMENTAL** (μια pure που μπορεί να εφαρμόζεται σε μητρώα στοιχείο-στοιχείο).

Εσωτερικά Υποπρογράμματα (Internal Procedures)

Τα υποπρογράμματα που βρίσκονται στο ίδιο αρχείο με το κυρίως πρόγραμμα τα λέμε Εσωτερικά υποπρογράμματα. Τα Εσωτερικά υποπρογράμματα τοποθετούνται στο τέλος του κυρίως προγράμματος ανάμεσα στις εντολές **STOP & END**, με τη χρήση της εντολής **CONTAINS**.

Εξωτερικά Υποπρογράμματα (External Procedures)

Τα υποπρογράμματα που δεν περιέχονται σε ένα άλλο υποπρόγραμμα, σε Module, ή, στο κυρίως πρόγραμμα, τα λέμε Εξωτερικά υποπρογράμματα. Τα Εξωτερικά υποπρογράμματα μπορούν να καλούνται κατ' ευθείαν με την εντολή **CALL** χωρίς να προηγηθεί η εντολή **USE**.

ΣΥΝΑΡΤΗΣΗ ΕΝΤΟΛΗΣ

Απλές **συναρτήσεις** μπορούν να δημιουργηθούν μέσα σε ένα πρόγραμμα και **χωρίς τη χρήση υποπρογραμμάτων** αρκεί να περιγράφονται σε μια εντολή. Οι **συναρτήσεις εντολής** ισχύουν μόνο για το συγκεκριμένο πρόγραμμα.

```
PROGRAM
REAL:: A, B, C, MesosOros, P,... ! Τύπος της συνάρτησης
...
MesosOros (A,B,C) = (A+B+C)/3.    ! Ορισμός της συνάρτησης
...
mo = MesosOros (TEST1,TEST2,XLAB) ! Χρήση της συνάρτησης
...
IF (MesosOros (P,D,Q) .LT. MesosOros (X,Y,Z)) STOP
...
END
```

ΕΝΤΟΛΕΣ

CALL

Η **CALL** μεταφέρει τον έλεγχο σε μια υπορουτίνα αντικαθιστώντας τα εικονικά ορίσματα με υπαρκτά ορίσματα που έχουν διευθύνσεις και τιμές.

```
CALL onoma_routinas ( a, b, c, ... λίστα υπαρκτών ορισμάτων)
```

USE

Η USE ενεργοποιεί τη χρήση ενός Module από ένα πρόγραμμα ή υποπρόγραμμα, και επιτρέπει τη πρόσβαση στα δεδομένα και τα υποπρογράμματα που περιέχει. Η εντολή τοποθετείται αμέσως μετά την πρώτη εντολή (Program, Subroutine, ή, Function) και πριν την IMPLICIT NONE. Η USE απαιτείται για να μπορεί να γίνει κλήση συναρτήσεων (με όνομα) ή υπορουτινών (με CALL) από το module.

CONTAINS

Η εισάγει το τμήμα που περιέχει τα εσωτερικά υποπρογράμματα μέσα σε ένα κύριο πρόγραμμα, module, ή, υποπρόγραμμα. Το τμήμα αυτό τοποθετείται πριν το τελικό END της προγραμματιστικής μονάδας.

RETURN

Η εντολή RETURN μεταφέρει την εκτέλεση από το υποπρόγραμμα στο καλούν πρόγραμμα. Μπορεί να βρῖσκειται οπουδήποτε στο υποπρόγραμμα, όπως περίπου η εντολή STOP στο κυρίως πρόγραμμα.

ΔΗΛΩΣΕΙΣ & ΟΡΙΣΜΑΤΑ

PUBLIC, PRIVATE

Όλες οι μεταβλητές και τα προγράμματα που περιέχονται σε ένα Module θεωρούνται ότι είναι **PUBLIC**, δηλαδή διαθέσιμα στα προγράμματα που το χρησιμοποιούν με τη USE. Ένα module όμως μπορεί να περιέχει και αντικείμενα που είναι μόνο για εσωτερική του χρήση. Μπορούμε να 'κρύψουμε' αυτές τις ποσότητες από το καλούν πρόγραμμα με την εντολή **PRIVATE**. Στο παρακάτω module, οι υπορουτίνες Routin1 & Routin2 διατίθενται κανονικά, ενώ η βοηθητική συνάρτηση AuxFunction1 που χρησιμοποιείται μόνο από τις δυο ρουτίνες δεν δίνεται για εξωτερική χρήση.

```
MODULE
...
PUBLIC:: Routin1, Routin2
PRIVATE:: AuxFunction1
...
```

INTENT (IN/OUT/INOUT)

Στις υπορουτίνες, η παρένθεση που ακολουθεί το όνομα περιέχει όλα τα ορίσματα, δεδομένα και αποτελέσματα. Υπάρχει η δυνατότητα να προσδιοριστεί αν τα ορίσματα είναι εισερχόμενα (IN), εξερχόμενα (OUT), ή, και τα δυο (INOUT). Το τελευταίο είναι και η προεπιλογή (default) όταν δεν οριστεί τίποτα. Τα εισερχόμενα ορίσματα δεν μπορούν να τροποποιηθούν από την υπορουτίνα, ενώ οι τιμές που έχουν τα εξερχόμενα ορίσματα τη στιγμή της εισόδου, αγνοούνται.

```
SUBROUTINE Intent (i, j, k, a)
REAL, INTENT (IN):: i, j
REAL, INTENT (OUT):: k
REAL, INTENT (INOUT):: a
k = (i * j) + a
a = a / k
...
```

Οι μεταβλητές i, j είναι εισερχόμενες και δεν αλλάζουν, η k υπολογίζεται αγνοώντας τελείως την όποια τιμή είχε κατά την είσοδο της, και η a χρησιμοποιεί τη τιμή της κατά την είσοδο και παίρνει τη νέα της τιμή κατά την έξοδο.

Συνάφεια Ορισμάτων

Το πιο σημαντικό και δύσκολο σημείο στην ανάπτυξη υποπρογραμμάτων είναι η συνάρφεια, το ταίριασμα, των πραγματικών ορισμάτων του καλούντος προγράμματος, με τα εικονικά (dummy) ορίσματα του υποπρογράμματος.

Παράδειγμα Συνάρτησης Χρήστη

Το επόμενο πρόγραμμα υπολογίζει το μέσο όρο των στοιχείων ενός διανύσματος με τη βοήθεια μιας εσωτερικής συνάρτησης που περιέχει (contains). Η συνάρτηση χρήστη καλεί και δυο συναρτήσεις βιβλιοθήκης.

```
PROGRAM MOfunction
  REAL:: mo, A(6)=[1, 2, 3, 4, 5, 6]
  mo = MesosOros(A) ; WRITE(*,*) mo
STOP
CONTAINS
  FUNCTION MesosOros(A)
    REAL:: MesosOros
    REAL, INTENT(IN):: A(:)
    MesosOros = SUM(A)/SIZE(A)
  END FUNCTION
END
```

Παράδειγμα Υπορουτίνας Χρήστη

Το επόμενο πρόγραμμα υπολογίζει το μήκος ενός διανύσματος από τις συντεταγμένες X-Y.

```
PROGRAM veclen
  REAL:: X, Y, L
  READ(*,*) X,Y
  CALL MHKOSdianysmatos(X,Y,L)
  WRITE(*,*) 'MHKOS = ',L
STOP
CONTAINS
  SUBROUTINE MHKOSdianysmatos(x,y,z)
    REAL,INTENT(IN):: x,y
    REAL,INTENT(OUT):: z
    z = SQRT(x**2 + y**2)
  END SUBROUTINE
END
```
