

ΔΟΜΕΣ ΤΥΠΩΝ ΔΕΔΟΜΕΝΩΝ - ΑΝΤΙΚΕΙΜΕΝΑ

Εκτός από τους ενσωματωμένους τύπους μεταβλητών της γλώσσας, ο προγραμματιστής μπορεί να δημιουργήσει και τους δικούς του τύπους. Οι νέοι αυτοί τύποι αποτελούνται από ένα συνδυασμό των τύπων που διαθέτει η Fortran. Οι οριζόμενοι από τον χρήστη τύποι είναι ιδιαίτερα χρήσιμοι όταν θέλουμε να δημιουργήσουμε λίστες ή μητρώα με ανόμοια δεδομένα (διαφορετικού τύπου).

Για παράδειγμα έστω ότι έχουμε πολλούς μετεωρολογικούς σταθμούς σε ένα νομό και συγκεντρώνουμε από τον κάθε ένα στοιχεία για το πρόγραμμα πρόβλεψης καιρού. Από κάθε σταθμό παίρνουμε ένα σετ από δεδομένα διαφορετικού τύπου, π.χ.: τον κωδικό (ακέραιος), το όνομα (χαρακτήρας), το γεωγραφικό μήκος, πλάτος & υψόμετρο (πραγματικοί), και τη θερμοκρασία, υγρασία & βροχόπτωση του 24ώρου (πραγματικά διανύσματα 24 θέσεων). Κανονικά θα έπρεπε να δηλωθεί κάθε τύπος χωριστά και μάλιστα σαν μητρώο ή διάνυσμα με μέγεθος ίσο με το πλήθος των σταθμών. Π.χ.:

```
INTEGER, PARAMETER:: Nst=400
INTEGER:: codikos(Nst)
CHARACTER(LEN=15):: onoma(Nst)
REAL:: Gplatos(Nst), Gmikos(Nst), Ypsometro(Nst)
REAL:: therm(24,Nst), ygras(24,Nst), broxopt(24,Nst)
```

Αν μάλιστα έχουμε και έναν άλλο νομό, θα πρέπει να δημιουργήσουμε άλλα τόσα μητρώα με διαφορετικά όμως ονόματα και διαστάσεις από τα προηγούμενα. Αυτός ο τρόπος οδηγεί σε χαοτικά προγράμματα με πολλά και δυσεύρετα λάθη και με μηδενική επεκτασιμότητα. Από τη στιγμή που το σετ των ανόμοιων δεδομένων επαναλαμβάνεται για κάθε σταθμό, δημιουργούμε σε ένα Module ένα νέο τύπο μεταβλητής, τον τύπο STATHMOS ως εξής:

```
MODULE NeoiTypoi
  TYPE STATHMOS
    INTEGER:: codikos
    CHARACTER(LEN=15):: onoma
    REAL:: Gplatos, Gmikos, Ypsometro
    REAL, DIMENSION(24):: therm, ygras, broxopt
  END TYPE
END MODULE Typedef
```

Ο νέος τύπος μπορεί να περιέχει τα δεδομένα ενός σταθμού. Αν τον δηλώσουμε σαν μητρώο κάθε στοιχείο του θα αντιστοιχεί σε έναν σταθμό και θα περιέχει 8 διαφορετικά είδη δεδομένων. Στο κυρίως πρόγραμμα χρησιμοποιούμε (USE) το Module 'NeoiTypoi' ώστε να μπορούμε να δηλώσουμε με το νέο τύπο STATHMOS όσες μεταβλητές χρειάζεται. Π.χ.:

```
PROGRAM Kairos
  USE NeoiTypoi
  IMPLICIT NONE
  ...
  INTEGER, PARAMETER:: Nattiki=200, Nviotia=130, Nevoia=150
  ...
  TYPE(STATHMOS):: attiki(Nattiki)
  TYPE(STATHMOS):: viotia(Nviotia), evoia(Nevoia)
  ...
  READ(10,20) (attiki(i), i=1,Nattiki)
  20 FORMAT(I5,A15,3F6.1,3(24F8.2))
  ...
```

Η δήλωση γίνεται με την εντολή TYPE(STATHMOS) και η ανάγνωση των δεδομένων γίνεται με ένα απλό πλάγιο DO. Προσοχή θα χρειαστεί στο FORMAT όπου θα εμφανίζονται με τη σειρά κωδικοί περιγραφής για όλες τις μεταβλητές που περιέχονται στον τύπο STATHMOS. Πράγματι, η

εντολή 20 διαβάζει: 1 ακέραιο, 1 σειρά χαρακτήρων, 3 πραγματικούς & 3 διανύσματα των 24 πραγματικών.

Κατά την εκτέλεση του προγράμματος όταν αναφερόμαστε σε ένα από τα περιεχόμενα του τύπου χρησιμοποιούμε το σύμβολο % για να χωρίσουμε το όνομα της μεταβλητής από τα ονόματα των περιεχομένων (π.χ.: `attiki(78)%broxopt(10:14)` θα δώσει ένα διάνυσμα με 5 τιμές βροχόπτωσης από το σταθμό 78 της Αττικής. Το ημερήσιο άθροισμα βροχόπτωσης σε κάθε σταθμό θα είναι:

```
...
DO i = 1, Nattiki
  attiki_broxHmeras(i)=SUM(attiki(i)%broxopt(1:24))
END DO
DO i = 1, Nviotia
  viotia_broxHmeras(i)=SUM(viotia(i)%broxopt(1:24))
END DO
DO i = 1, Nevoia
  evoia_broxHmeras(i)=SUM(evoia(i)%broxopt(1:24))
END DO
...
```

Τέλος, για κάθε τύπο δεδομένων μπορούν να οριστούν και πράξεις ή συναρτήσεις που ισχύουν μόνο για τον συγκεκριμένο τύπο. Γενικά, η χρήση οριζόμενων τύπων δεδομένων έχει υπέρ και κατά, και, η επιλογή της χρήσης τους ή όχι, έχει πολλά κοινά με την επιλογή μιας βάσης δεδομένων έναντι ενός λογιστικού φύλου.