Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

ΚΕΦΑΛΑΙΟ 10 - Υποπρογράμματα

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.
Όταν ένα τμήμα προγράμματος επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα, τότε αναφερόμαστε σε **υποπρόγραμμα (subprogram).**
Η σωστή εφαρμογή του τμηματικού προγραμματισμού απαιτεί :

- μελέτη στην ανάλυση του προβλήματος,
- εμπειρία στον προγραμματισμό,
- ταλέντο και
- γνώσεις.

Υπάρχουν πάντως τρεις **ιδιότητες** που πρέπει να διακρίνουν τα υποπρογράμματα:

- **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο.**
- **Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.**
- **Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο.**

Ο σωστός χωρισμός ενός σύνθετου προγράμματος σε υποπρογράμματα εξασφαλίζει **τέσσερα βασικά χαρακτηριστικά** του σωστού προγραμματισμού:

- **Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος.**
- **Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.**
- **Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.**
- **Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.**

Μία **παράμετρος** είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.
Οι διαδικασίες μπορούν να εκτελέσουν **οποιαδήποτε λειτουργία** από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα. Να εισάγουν δεδομένα, να εκτελέσουν υπολογισμούς, να μεταβάλλουν τις τιμές των μεταβλητών και να τυπώσουν αποτελέσματα. Με τη **χρήση των παραμέτρων** αυτές τις τιμές μπορούν να τις μεταφέρουν και στα άλλα υποπρογράμματα.
Οι συναρτήσεις **υπολογίζουν μόνο μία τιμή**, αριθμητική, χαρακτήρα ή λογική και **μόνο αυτήν επιστρέφουν στο υποπρόγραμμα που την κάλεσε**. Οι συναρτήσεις μοιάζουν με τις συναρτήσεις των μαθηματικών και η χρήση τους είναι όμοια με τη χρήση των ενσωματωμένων συναρτήσεων που υποστηρίζει η γλώσσα προγραμματισμού. Η λειτουργία των συναρτήσεων είναι πιο περιορισμένη.

ΣΥΝΑΡΤΗΣΗ όνομα (λίστα παραμέτρων):τύπος συνάρτησης
Τμήμα δηλώσεων
ΑΡΧΗ
....
όνομα <- έκφραση
...
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ Όνομα (λίστα παραμέτρων)
Τμήμα δηλώσεων
ΑΡΧΗ
εντολές
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Χ <- Εμβαδό_κύκλου(R).
ΚΑΛΕΣΕ όνομα_διαδικασίας (λίστα παραμέτρων)

*Η λίστα των **τυπικών παραμέτρων** (formal parameter list) καθορίζει τις παραμέτρους στη **δήλωση** του υποπρογράμματος. (ορίσματα)*
*Η λίστα των **πραγματικών παραμέτρων** (actual parameter list) καθορίζει τις παραμέτρους στην **κλήση** του υποπρογράμματος. (παράμετροι)*

Όλες οι **μεταβλητές** είναι γνωστές, **έχουν ισχύ** όπως λέγεται, μόνο για το **τμήμα** προγράμματος στο οποίο **έχουν δηλωθεί**, ισχύουν δηλαδή **τοπικά** για το συγκεκριμένο υποπρόγραμμα ή κυρίως πρόγραμμα.

Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η **αμέσως επόμενη διεύθυνση του κύριου προγράμματος**, που ονομάζεται ***διεύθυνση επιστροφής***, *αποθηκεύεται* από το μεταφραστή σε μία στοίβα που ονομάζεται ***στοίβα χρόνου*** *εκτέλεσης. 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.*

Κανόνες για τη χρήση παραμέτρων

- Ο **αριθμός** των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.
- Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην **αντίστοιχη θέση**. Για παράδειγμα η πρώτη της λίστας των τυπικών παραμέτρων στην πρώτη της λίστας των πραγματικών παραμέτρων κοκ.
- Η τυπική παράμετρος και η αντίστοιχη της πραγματική πρέπει να είναι του **ιδίου τύπου**.

Μετατροπή συνάρτησης σε διαδικασία

```
ΠΡΟΓΡΑΜΜΑ μετατροπή
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: χ, ψ, αποτέλεσμα
ΑΡΧΗ
  ΔΙΑΒΑΣΕ χ, ψ
  ! με κλήση συνάρτησης
  αποτέλεσμα <- υπολογισμός (χ, ψ)
  ΓΡΑΨΕ αποτέλεσμα
  ! με κλήση διαδικασίας
  ΚΑΛΕΣΕ κάνε_υπολογισμό (χ,ψ,αποτέλεσμα)
  ΓΡΑΨΕ αποτέλεσμα
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ υπολογισμός (α, β): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: α, β
ΑΡΧΗ
  υπολογισμός <- 3*11 + 3*12 + 3*15 + α + β
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ  κάνε_υπολογισμό  (α, β, υπολογισμός)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: α, β, υπολογισμός
ΑΡΧΗ
  υπολογισμός <- 3*11 + 3*12 + 3*15 + α + β
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Για τη μετατροπή χρησιμοποιούμε μια επιπλέον μεταβλητή η οποία θα επιστρέψει την τιμή που υπολόγιζε η συνάρτηση. Για τις λιγότερες μετατροπές μπορούμε να χρησιμοποιήσουμε ως όνομα μεταβλητής το όνομα της συνάρτησης, δίνουμε ένα όνομα στη διαδικασία και δηλώνουμε την μεταβλητή.

Παρατήρηση: Στη διαδικασία όλες οι παράμετροι επιστρέφουν τις τιμές τους, ενώ η συνάρτηση γυρίζει μόνο την τιμή της. Γι΄ αυτό, αν η συνάρτηση μετέβαλε τις παραμέτρους της, θα έπρεπε να τις αποθηκεύσουμε σε κάποιες προσωρινές μεταβλητές μπαίνοντας στη διαδικασία και πριν το ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ να τις επαναφέρουμε.

- **Τι ονομάζεται εμβέλεια;**

Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται **εμβέλεια** (scope) μεταβλητών.

- **Τι ονομάζεται Απεριόριστη εμβέλεια;**

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. Όλες οι μεταβλητές είναι καθολικές.

- **Ποιες μεταβλητές ονομάζονται καθολικές;**

Καθολικές ονομάζονται οι μεταβλητές που ισχύουν **σε οποιοδήποτε τμήμα** του προγράμματος, άσχετα που δηλώθηκαν και είναι γνωστές στο κύριο πρόγραμμα και όλα τα υποπρογράμματα.

- **Ποια τα μειονεκτήματα της απεριόριστης εμβέλειας;**

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.

- **Τι είναι η 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.

- **Ποιες μεταβλητές ονομάζονται τοπικές;**

Τοπικές ονομάζονται οι μεταβλητές που ισχύουν για το υποπρόγραμμα στο οποίο δηλώθηκαν.

- **Ποια τα πλεονεκτήματα της περιορισμένης εμβέλειας;**
  - ✓ απόλυτη αυτονομία όλων των υποπρογραμμάτων και
  - ✓ δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

- **Τι είναι η Μερικώς περιορισμένη εμβέλεια;**

Σύμφωνα με τη **μερικώς περιορισμένη εμβέλεια 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.

- **Πλεονεκτήματα και μειονεκτήματα της Μερικώς περιορισμένης εμβέλειας;**

Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά **πλεονεκτήματα στον πεπειραμένο** προγραμματιστή, αλλά για τον **αρχάριο** περιπλέκει το πρόγραμμα **δυσκολεύοντας** την ανάπτυξή του.