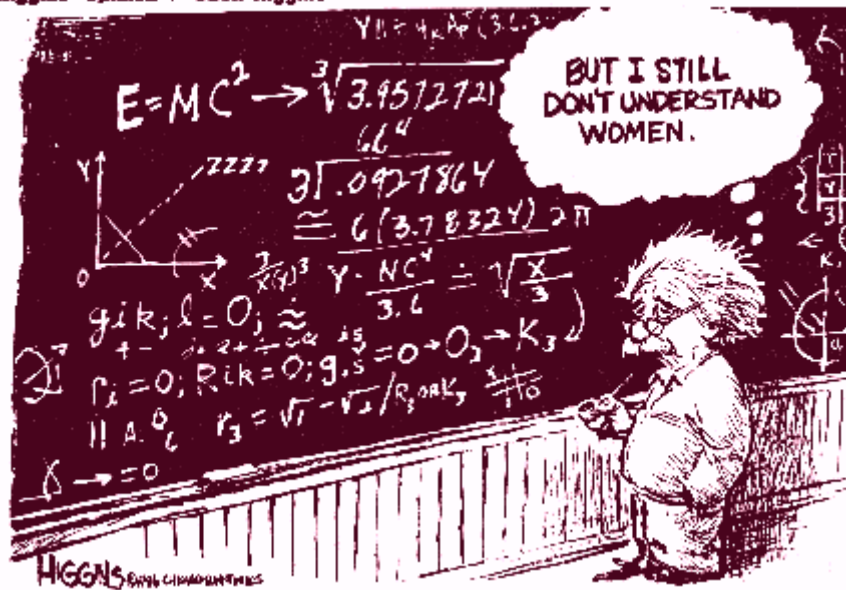


ΚΕΦΑΛΑΙΟ 10

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Higgins' Opinion / Jack Higgins





ΚΕΦΑΛΑΙΟ 10 ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Όπως είδαμε και σε προηγούμενο κεφάλαιο μια από τις βασικότερες τεχνικές στον Δομημένο Προγραμματισμό είναι ο Τμηματικός Προγραμματισμός.

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Χαρακτηριστικά των υποπρογραμμάτων:

1. Κάθε υποπρόγραμμα έχει μόνο μια είσοδο και μια έξοδο.

Κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό που γίνεται πάντοτε στην αρχή του, εκτελεί ορισμένες ενέργειες και απενεργοποιείται με την έξοδο από αυτό που γίνεται πάντοτε από το τέλος του.

2. Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα.

Αυτό σημαίνει ότι κάθε υποπρόγραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα. Στην πράξη η απόλυτη ανεξαρτησία είναι πολύ δύσκολο να επιτευχθεί.

3. Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο.

Η έννοια του μεγάλου προγράμματος είναι υποκειμενική, αλλά πρέπει κάθε υποπρόγραμμα να είναι τόσο, ώστε να είναι εύκολα κατανοητό και να μπορεί να ελέγχεται. Κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

Η χρήση υποπρογραμμάτων σε ένα πρόγραμμα παρουσιάζει πολλά πλεονεκτήματα.

1. Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντίστοιχου προγράμματος.

Επιτρέπει την εξέταση και την επίλυση απλών προβλημάτων και όχι την αντιμετώπιση του συνολικού προβλήματος. Με τη σταδιακή επίλυση των υποπροβλημάτων και τη δημιουργία των αντίστοιχων υποπρογραμμάτων τελικά επιλύεται το συνολικό πρόβλημα.

2. Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.

Ο χωρισμός του προγράμματος σε μικρότερα αυτοτελή τμήματα επιτρέπει τη γρήγορη διόρθωση ενός συγκεκριμένου τμήματός του χωρίς οι αλλαγές αυτές να επηρεάσουν όλο το υπόλοιπο πρόβλημα. Επίσης διευκολύνει οποιονδήποτε χρειαστεί να διαβάσει για να κατανοήσει τον τρόπο που λειτουργεί το πρόγραμμα. Αυτό είναι πολύ σημαντικό, αν σκεφτεί κανείς ότι ένα μεγάλο πρόγραμμα στον κύκλο της ζωής του χρειάζεται να συντηρηθεί από διαφορετικούς προγραμματιστές.

3. Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.

Πολύ συχνά χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία ενός προγράμματος. Από τη στιγμή που ένα υποπρόγραμμα έχει γραφεί, μπορεί να καλείται από πολλά σημεία του προγράμματος. Έτσι μειώνεται το μέγεθος του προγράμματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται εύληπτο και κατανοητό.

4. Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.

Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πολύ εύκολα και σε άλλα προγράμματα. Από τη στιγμή που έχει δημιουργηθεί, η χρήση του δε διαφέρει ουσιαστικά από την χρήση των ενσωματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού που χρησιμοποιούμε (όπως για τον υπολογισμό του ημίτονου), ή την εντολή με την οποία εκτελεί μια συγκεκριμένη διαδικασία (π.χ. η εντολή ΓΡΑΨΕ). Αν λοιπόν χρειάζεται συχνά μια διαδικασία που δεν υποστηρίζεται απευθείας από τη γλώσσα, όπως για παράδειγμα η εύρεση του μικρότερου δύο



αριθμών, τότε μπορεί να γραφεί το αντίστοιχο υποπρόγραμμα. Η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά, ουσιαστικά επεκτείνουν την ίδια τη γλώσσα προγραμματισμού.

Τα υποπρογράμματα καλούνται από άλλα υποπρογράμματα ή από κάποιο άλλο πρόγραμμα για να εκτελέσουν συγκεκριμένες λειτουργίες. Κάθε υποπρόγραμμα για να ενεργοποιηθεί καλείται από ένα άλλο υποπρόγραμμα ή το αρχικό πρόγραμμα, το οποίο ονομάζεται κύριο πρόγραμμα. Το υποπρόγραμμα είναι αυτόνομο και ανεξάρτητο τμήμα προγράμματος, αλλά συχνά πρέπει να επικοινωνεί με το υπόλοιπο πρόγραμμα. Συνήθως δέχεται τιμές από το τμήμα προγράμματος που το καλεί και μετά την εκτέλεση επιστρέφει νέες τιμές, αποτελέσματα. Οι τιμές που περνούν από το ένα υποπρόγραμμα στο άλλο λέγονται παράμετροι. Οι παράμετροι λοιπόν είναι σαν τις κοινές μεταβλητές ενός προγράμματος με μια ουσιώδη διαφορά, χρησιμοποιούνται για να περνούν τιμές στα υποπρογράμματα.

Μια παράμετρος είναι μια μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε άλλο.

Υπάρχουν δύο ειδών υποπρογράμματα, οι διαδικασίες και οι συναρτήσεις. Και τα δύο τοποθετούνται μετά το τέλος του κύριου προγράμματος.

Διαδικασίες

Οι διαδικασίες μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα. Να εισάγουν δεδομένα, να εκτελέσουν υπολογισμούς, να μεταβάλουν τις τιμές των μεταβλητών και να τυπώσουν αποτελέσματα. Με τη χρήση των παραμέτρων αυτές τις τιμές μπορούν να τις μεταφέρουν και στα υπόλοιπα υποπρογράμματα.

Η διαδικασία είναι ένας τύπος προγράμματος που μπορεί να εκτελέσει όλες τις λειτουργίες ενός προγράμματος.

Κάθε διαδικασία έχει την ακόλουθη δομή.

| <i>Σύνταξη</i> | <i>Παράδειγμα</i> |
|--|---|
| ΔΙΑΔΙΚΑΣΙΑ Όνομα (λίστα παραμέτρων) Τμήμα Δηλώσεων ΑΡΧΗ Εντολές ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ | ΔΙΑΔΙΚΑΣΙΑ MAX2 (α,β,γ) ΑΡΧΗ ΑΝ α>=β ΤΟΤΕ γ←α ΑΛΛΙΩΣ γ←β ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ |
| <i>Κλίση</i> ΚΑΛΕΣΕ όνομα-διαδικασίας(λίστα παραμέτρων) | <i>Κλίση</i> ΚΑΛΕΣΕ MAX2(x,y,z) |

Το όνομα της διαδικασίας μπορεί να είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ. Η λίστα παραμέτρων είναι μια λίστα μεταβλητών των οποίων οι τιμές μεταβιβάζονται από το κύριο πρόγραμμα προς τη διαδικασία κατά την κλίση ή/και επιστρέφονται στο κύριο πρόγραμμα μετά το τέλος της διαδικασίας. Στο σώμα της διαδικασίας μπορούν να υπάρχουν οποιοσδήποτε εντολές της γλώσσας. Με την κλίση της διαδικασίας από το κύριο πρόγραμμα οι εντολές του προγράμματος διακόπτονται και εκτελούνται οι εντολές της διαδικασίας. Μετά το τέλος της διαδικασίας η εκτέλεση του προγράμματος συνεχίζεται από την εντολή που ακολουθεί.

**Συναρτήσεις**

Η λειτουργία των συναρτήσεων είναι λίγο πιο περιορισμένη. Οι συναρτήσεις υπολογίζουν μια μόνο 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 συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μια τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).

Κάθε συνάρτηση έχει την ακόλουθη δομή.

| <i>Σύνταξη</i> | <i>Παράδειγμα</i> |
|---|---|
| ΣΥΝΑΡΤΗΣΗ Όνομα (λίστα παραμέτρων): Τύπος Τμήμα Δηλώσεων ΑΡΧΗ ... όνομα ← έκφραση ... ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ | ΣΥΝΑΡΤΗΣΗ MAX2 (α,β): ΠΡΑΓΜΑΤΙΚΟΣ ΑΡΧΗ AN $\alpha \geq \beta$ ΤΟΤΕ MAX2 ← α ΑΛΛΙΩΣ MAX2 ← β ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ |
| <i>Κλίση</i> μεταβλητή ← όνομα-συνάρτησης(λίστα παραμέτρων) | <i>Κλίση</i> μέγιστο ← MAX2(x,y) |

Το όνομα της διαδικασίας μπορεί να είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ. Η λίστα παραμέτρων είναι μια λίστα μεταβλητών των οποίων οι τιμές μεταβιβάζονται από το κύριο πρόγραμμα προς τη διαδικασία κατά την κλίση. Στο σώμα της διαδικασίας μπορούν να υπάρχουν οποιεσδήποτε εντολές της γλώσσας. Οποσδήποτε όμως πρέπει να υπάρχει μια εντολή εκχώρησης τιμής στο όνομα της συνάρτησης. Οι συναρτήσεις μπορούν να επιστρέφουν τιμές όλων των τύπων που υποστηρίζει η γλώσσα (πραγματικές, χαρακτήρες κλπ). Κάθε συνάρτηση εκτελείται όπως ακριβώς εκτελούνται και οι ενσωματωμένες συναρτήσεις της ΓΛΩΣΣΑΣ. Απλώς αναφέρεται το όνομά της σε μια έκφραση ή σε μια εντολή και επιστρέφεται η τιμή της. Με την κλίση της συνάρτησης από το κύριο πρόγραμμα οι εντολές του προγράμματος διακόπτονται και εκτελούνται οι εντολές της συνάρτησης. Μετά το τέλος της συνάρτησης, επιστρέφεται το αποτέλεσμα στην αντίστοιχη μεταβλητή και η εκτέλεση του προγράμματος συνεχίζεται από την εντολή που ακολουθεί.

Τυπικές Παράμετροι ονομάζονται οι μεταβλητές που χρησιμοποιεί το υποπρόγραμμα.

Πραγματικές Παράμετροι ονομάζονται οι μεταβλητές που χρησιμοποιεί ένα πρόγραμμα για να καλέσει το συγκεκριμένο υποπρόγραμμα.

Στο παράδειγμα της συνάρτησης MAX2 τυπικές παράμετροι είναι οι α, β ενώ πραγματικές παράμετροι είναι οι x, y. Τα ονόματα των τυπικών και των πραγματικών παραμέτρων μπορεί να είναι οποιαδήποτε. Αφού, όμως, είναι ονόματα μεταβλητών σε διαφορετικά τμήματα προγράμματος είναι υποχρεωτικά διαφορετικές μεταβλητές ακόμα και αν έχουν το ίδιο όνομα.

Όλες οι μεταβλητές έχουν ισχύ μόνο για το τμήμα προγράμματος στο οποίο έχουν δηλωθεί. Ισχύουν δηλαδή **τοπικά** για το συγκεκριμένο υποπρόγραμμα ή το κυρίως πρόγραμμα.

Οι λίστες των παραμέτρων ακολουθούν τους εξής κανόνες.

- 1) Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.



- 2) 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 κ.ο.κ.
- 3) Η τυπική παράμετρος και η αντίστοιχη πραγματική πρέπει να είναι ίδιου τύπου.

**Ασκήσεις**

- 1) Να γράψεις πρόγραμμα το οποίο να εκτελεί τις τέσσερις πράξεις των μιγαδικών αριθμών. Το πρόγραμμα θα οδηγείται από μενού επιλογής όπου ο χρήστης θα επιλέγει το είδος της πράξης. Το πρόγραμμα θα πρέπει να χρησιμοποιεί διαδικασίες και συναρτήσεις.
- 2) Να γράψετε μια συνάρτηση που να δέχεται ως παραμέτρους το ύψος και τη βάση ενός τριγώνου και να υπολογίζει και επιστρέφει το εμβαδόν του. Στη συνέχεια να γράψετε μια διαδικασία που να δέχεται ως παράμετρο ένα αριθμό A . Η διαδικασία θα πρέπει να διαβάσει μια τιμή από το πληκτρολόγιο και να την αποθηκεύει στην A , αν όμως η τιμή αυτή είναι αρνητική (ή μηδέν) τότε διαβάσει μια νέα τιμή κ.ο.κ μέχρι να διαβάσει θετική τιμή. Τέλος να χρησιμοποιηθούν αυτά τα υποπρογράμματα για να γραφεί ένα πρόγραμμα για τον υπολογισμό του εμβαδού ενός τριγώνου.
- 3) Όμοια για τον κύλινδρο.
- 4) Να γράψετε τους αλγορίθμους για τους πίνακες ως συναρτήσεις ή διαδικασίες.