

ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΒΑΣΙΚΑ





ΘΕΩΡΙΑ

Η τεχνική σχεδίασης και ανάπτυξης υποπρογραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων

Εξασφαλίζει επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων

Υλοποιεί την φιλοσοφία της ιεραρχικής σχεδίασης

Πρακτικά μέσω του τμηματικού προγραμματισμού μας επιτρέπεται η διαίρεση ενός σύνθετου προβλήματος σε άλλα μικρότερα και η επίλυση καθενός από αυτά με την υλοποίηση αντίστοιχου υποπρογράμματος. Το κύριο πρόγραμμα θα παίζει τον ρόλο του συντονιστή και θα μπορεί να καλεί τα αντίστοιχα υποπρογράμματα που χρειάζεται περνώντας τα δεδομένα που αυτά χρειάζονται για την λειτουργία τους. Τα υποπρογράμματα μπορούν να επικοινωνούν και μεταξύ τους με κανόνες που θα αναλύσουμε στην συνέχεια.

ΤΙ ΕΙΝΑΙ ΤΟ ΥΠΟΠΡΟΓΡΑΜΜΑ;

Υποπρόγραμμα είναι ένα τμήμα προγράμματος που επιτελεί αυτόνομο έργο και γράφεται ξεχωριστά από το υπόλοιπο πρόγραμμα και επιτελεί κάποια αυτόνομη λειτουργία του προγράμματος. Κάθε υποπρόγραμμα δουλεύει αυτόνομα και διαθέτει δική του περιοχή μνήμης, ενώ η επαφή που έχει με άλλα τμήματα προγράμματος γίνεται αποκλειστικά και μόνο μέσω παραμέτρων. ΤΑ ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΕΚΤΕΛΟΥΝΤΑΙ ΜΟΝΟ ΟΤΑΝ ΤΑ ΚΑΛΕΣΟΥΜΕ

Έχουμε 2 ειδών υποπρογράμματα που θα αναλύσουμε.

ΔΙΑΔΙΚΑΣΙΕΣ | ΣΥΝΑΡΤΗΣΕΙΣ

3 ΙΔΙΟΤΗΤΕΣ ΤΩΝ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

1. Πρέπει να έχουν μία είσοδο και μια έξοδο
2. Να είναι ανεξάρτητα
3. Να μην είναι πολύ μεγάλα
4. Η είσοδος δεδομένων γίνεται πάντα στην αρχή τους

ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΤΜΗΜΑΤΙΚΟΥ ΠΡΟΓΡ/ΜΟΥ

1. Διευκόλυνση της ανάπτυξης του αλγορίθμου και του αντίστοιχου προγράμματος (Διαίρεση προβλήματος σε μικρότερα)
2. Διευκόλυνση της κατανόησης και της διόρθωσης του προγράμματος
3. Λιγότερος χρόνος και κώδικα για την **συγγραφή του προγράμματος (επαναχρησιμοποίηση υποπρογραμμάτων)**
4. Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού

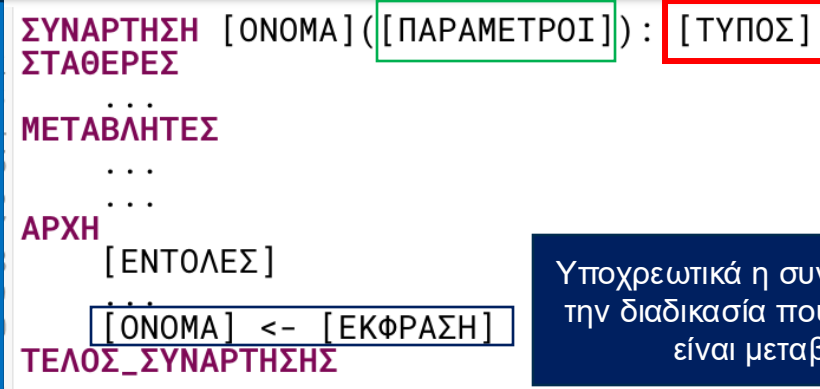
ΤΙ ΕΙΝΑΙ ΠΑΡΑΜΕΤΡΟΣ ΕΝΟΣ ΥΠΟΠΡΟΓΡΑΜΜΑΤΟΣ

Είναι μια μεταβλητή που επιτρέπει το πέρασμα τιμών από ένα υποπρόγραμμα σε ένα άλλο, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ τους



ΣΥΝΑΡΤΗΣΕΙΣ

Η συνάρτηση είναι ένας από τους 2 τύπους υποπρογραμμάτων. Υπολογίζει και επιστρέφει μόνο μια τιμή με το όνομα της. Δεν χρησιμοποιείται για κανέναν άλλο λόγο(διάδραση με το χρήστη, επιστροφή πολλαπλών τιμών κτλ.)



Είσοδος – παράμετροι εισόδου - Υποχρεωτικά
Τύπος της τιμής που επιστρέφει η συνάρτηση

Υποχρεωτικά η συνάρτηση υπολογίζει και επιστρέφει μια τιμή στο πρόγραμμα ή την διαδικασία που την κάλεσε, περνώντας στο όνομα της συνάρτησης(σαν να είναι μεταβλητή) το αποτέλεσμα κάποιας έκφρασης ή σταθεράς

ΠΟΥ ΚΑΙ ΠΩΣ ΚΑΛΕΙΤΑΙ ΜΙΑ ΣΥΝΑΡΤΗΣΗ;

Μια συνάρτηση μπορεί να κληθεί από οποιοδήποτε τμήμα προγράμματος:

- Από το κύριο πρόγραμμα
- Από διαδικασία
- Από κάποια άλλη συνάρτηση

Η κλήση μιας συνάρτησης πραγματοποιείται με το όνομα της ακολουθούμενο από τις παραμέτρους εισόδου που χρειάζεται.

ΙΔΙΟΤΗΤΕΣ ΣΥΝΑΡΤΗΣΕΩΝ

1. Η λίστα παραμέτρων για την συνάρτηση είναι υποχρεωτική και αποτελεί είσοδο δεδομένων για τη συνάρτηση
2. Ο τύπος μιας συνάρτησης που αποτελεί τον τύπο της επιστρεφόμενης τιμής μπορεί να είναι ΑΚΕΡΑΙΑ, ΧΑΡΑΚΤΗΡΑΣ, ΠΡΑΓΜΑΤΙΚΗ, ΛΟΓΙΚΗ
3. Η εκχώρηση τιμής μέσω έκφρασης στο όνομα της συνάρτησης είναι υποχρεωτική καθώς αποτελεί την τιμή επιστροφής της συνάρτησης
4. Οι παράμετροι, οι σταθερές και οι μεταβλητές που θα χρησιμοποιηθούν εντός της συνάρτησης θα πρέπει υποχρεωτικά να δηλώνονται
5. ΔΕΝ χρησιμοποιούμε τις εντολές ΓΡΑΨΕ / ΔΙΑΒΑΣΕ στη συνάρτηση(δεν είναι ο σκοπός τους η διάδραση με το χρήστη)
6. Ως παραμέτρους εισόδου σε μια συνάρτηση μπορούμε να περνάμε κατά την κλήση της μεταβλητές αλλά και σταθερές

ΠΑΡΑΔΕΙΓΜΑ 1 - ΣΥΝΑΡΤΗΣΕΙΣ

Δημιουργήστε ένα υποπρόγραμμα(διαδικασία ή συνάρτηση), το οποίο θα δέχεται 3 πραγματικές μή μηδενικές τιμές και θα επιστρέφει τον μέσο όρο τους. Στην συνέχεια δημιουργήστε ένα πρόγραμμα το οποίο θα διαβάζει από το χρήστη 3 πραγματικές θετικές τιμές και χρησιμοποιώντας το υποπρόγραμμα που υλοποιήσατε θα εμφανίζει στον χρήστη τον μέσο όρο τους

```
ΣΥΝΑΡΤΗΣΗ ΥΠΟΛΟΓΙΣΜΟΣ_ΜΟ(x, y, z): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: x, y, z
ΑΡΧΗ
    ΥΠΟΛΟΓΙΣΜΟΣ_ΜΟ <- (x + y + z) / 3
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

```
ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ1
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: a, b, c, MO
    ΛΟΓΙΚΕΣ: ΕΓΚΥΡΑ
ΑΡΧΗ
    ΓΡΑΨΕ 'ΔΩΣΕ 3 ΑΡΙΘΜΟΥΣ ΤΗΣ ΑΡΕΣΚΕΙΑΣ ΣΟΥ'
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΕΓΚΥΡΑ <- ΑΛΗΘΗΣ
        ΔΙΑΒΑΣΕ a, b, c
        ΑΝ a <= 0 Η b <= 0 Η c <= 0 ΤΟΤΕ
            ΕΓΚΥΡΑ <- ΨΕΥΔΗΣ
            ΓΡΑΨΕ 'Δώσε πραγματικές θετικές μη μηδενικές τιμές...'
        ΤΕΛΟΣ_ΑΝ
        ΜΕΧΡΙΣ_ΟΤΟΥ (ΕΓΚΥΡΑ)
        MO <- ΥΠΟΛΟΓΙΣΜΟΣ_ΜΟ(a, b, c)
    ΓΡΑΨΕ 'Ο μέσος όρος των 3 αριθμών είναι: ', MO
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Δημιουργούμε μια συνάρτηση(επιλέγουμε συνάρτηση γιατί θέλουμε τον υπολογισμό και την επιστροφή μιας τιμής). Η συνάρτηση δέχεται 3 **τυπικές παραμέτρους** για τις οποίες χρησιμοποιούμε τα ονόματα x,y,z(εμείς επιλέγουμε την ονομασία τους)

Όπως εξηγήσαμε, η επιστρεφόμενη τιμή της συνάρτησης εκχωρείται στο όνομα της, ενώ ορίζουμε και τον τύπο της επιστρεφόμενης αυτής τιμής που είναι πραγματική

Οι τυπικές παράμετροι x,y,z και (αν υπήρχαν) οι σταθερές και οι υπόλοιπες μεταβλητές θα έπρεπε να δηλωθούν τοπικά στη συνάρτηση

Στο κυρίως πρόγραμμα κατά τα γνωστά, διαβάζουμε από το χρήστη 3 πραγματικές τιμές πραγματοποιώντας και έλεγχο δεδομένων(να είναι θετικές πραγματικές τιμές)

Αφού ο χρήστης εισάγει έγκυρες πραγματικές τιμές τότε καλούμε την συνάρτηση με το όνομα της, περνώντας τις **πραγματικές παραμέτρους a,b,c** και ταυτόχρονα εκχωρούμε την **επιστρεφόμενη τιμή της συνάρτησης** σε μια νέα μεταβλητή για περαιτέρω χρήση

ΑΣΚΗΣΗ 2 - ΣΥΝΑΡΤΗΣΕΙΣ

Δημιουργήστε κατάλληλα υποπρογράμματα τα οποία:

1. Το πρώτο θα δέχεται 2 ακέραιους αριθμούς και θα επιστρέφει τον μεγαλύτερο από αυτούς
2. Υποπρόγραμμα το οποίο θα δέχεται 4 ακέραιους αριθμούς και χρησιμοποιώντας το προηγούμενο υποπρόγραμμα θα επιστρέφει τον μεγαλύτερο από αυτούς

```
ΣΥΝΑΡΤΗΣΗ ΜΕΓΙΣΤΟ2(k,z): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: k, z, max
ΑΡΧΗ
    max <- k
    ΑΝ z > k ΤΟΤΕ
        max <- z
    ΤΕΛΟΣ_ΑΝ
    ΜΕΓΙΣΤΟ2 <- max
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

```
ΣΥΝΑΡΤΗΣΗ ΜΕΓΙΣΤΟ4(x,y,z,w): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: x, y, z, w, max1, max2, maximum
ΑΡΧΗ
    max1 <- ΜΕΓΙΣΤΟ2(x,y)
    max2 <- ΜΕΓΙΣΤΟ2(z,w)

    maximum <- ΜΕΓΙΣΤΟ2(max1, max2)
    ΜΕΓΙΣΤΟ4 <- maximum
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Μια συνάρτηση μπορεί να κληθεί από οποιοδήποτε άλλο τμήμα προγράμματος(κυρίως πρόγραμμα, διαδικασία, άλλη συνάρτηση). Εδώ η μια συνάρτηση καλεί την άλλη

ΑΣΚΗΣΗ 3 - ΣΥΝΑΡΤΗΣΕΙΣ

Δημιουργήστε κατάλληλο υποπρόγραμμα το οποίο θα δέχεται ως παράμετρο έναν πραγματικό αριθμό και ακέραιο αριθμό που θα αντιπροσωπεύει το δεκαδικό του ψηφίο. Το υποπρόγραμμα θα υπολογίζει τον στρογγυλοποιημένο αριθμό στο ζητούμενο δεκαδικό ψηφίο

Στην συνέχεια δημιουργήστε ένα πρόγραμμα το οποίο θα διαβάζει έναν πραγματικό αριθμό και έναν ακέραιο αριθμό για τα δεκαδικά από το χρήστη και χρησιμοποιώντας το προηγούμενο υποπρόγραμμα θα εμφανίζει τον αριθμό στρογγυλοποιημένο

ΣΥΝΑΡΤΗΣΗ ROUNDING(k,z): **ΠΡΑΓΜΑΤΙΚΗ**
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: z, integer_part

ΠΡΑΓΜΑΤΙΚΕΣ: k, initial, decimal_part

ΑΡΧΗ

initial <- k * 10^z

integer_part <- A_M(initial)

decimal_part <- initial - integer_part

ΑΝ decimal_part > 0.5 **ΤΟΤΕ**

integer_part <- integer_part + 1

ΤΕΛΟΣ_ΑΝ

ROUNDING <- integer_part / 10^z

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ1

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: number, rounded_number

ΑΚΕΡΑΙΕΣ: floating_point

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε πραγματικό αριθμό...'

ΔΙΑΒΑΣΕ number

ΓΡΑΨΕ 'Δώσε ψηφίο δεκαδικών...'

ΔΙΑΒΑΣΕ floating_point

rounded_number <- ROUNDING(number, floating_point)

ΓΡΑΨΕ 'Το αποτέλεσμα της στρογγυλοποίησης είναι ', rounded_number

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ