

# Μάθημα 112

Τμηματικός προγραμματισμός  
Συνάρτηση

# Τμηματικός προγραμματισμός (1)

## Τι ονομάζουμε τμηματικό προγραμματισμό;

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

## Τι είναι το υποπρόγραμμα;

Υποπρόγραμμα (subprogram) είναι ένα τμήμα προγράμματος που επιτελεί αυτόνομο έργο και γράφεται ξεχωριστά από το υπόλοιπο πρόγραμμα.

π.χ. η καταχώρηση, ο έλεγχος, η εκτύπωση δεδομένων, ο υπολογισμός του μέσου όρου κάποιων αριθμών κ.τ.λ.

## Ιδιότητες των υποπρογραμμάτων

- **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο**

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

- **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 βέβαια η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.

- **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 για να μπορεί να ελέγχεται.

Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία.

Αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.

## Τμηματικός προγραμματισμός (2)

Ποια είναι τα πλεονεκτήματα του τμηματικού προγραμματισμού;

- **Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος**

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

- **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 πρόγραμμα. Επίσης διευκολύνει οποιονδήποτε χρειαστεί να διαβάσει και να κατανοήσει τον τρόπο που λειτουργεί το πρόγραμμα.

- **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 το ίδιο να καλείται από πολλά σημεία του προγράμματος. Έτσι μειώνονται το μέγεθος του προγράμματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται πιο εύληπτο και κατανοητό.

- **Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού**

Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πολύ εύκολα και σε άλλα προγράμματα. Αν λοιπόν χρειάζεται συχνά κάποια λειτουργία που δεν υποστηρίζεται απευθείας από τη γλώσσα, η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά, ουσιαστικά επεκτείνουν την ίδια τη γλώσσα προγραμματισμού.

# Συναρτήσεις (1)

## Τι είναι παράμετρος ενός υποπρογράμματος;

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

Τα υποπρογράμματα ενεργοποιούνται (καλούνται) είτε από άλλα υποπρογράμματα είτε από το κυρίως πρόγραμμα.

Παρόλο που το υποπρόγραμμα είναι αυτόνομο και ανεξάρτητο τμήμα, κάποιες φορές ίσως να πρέπει να επικοινωνεί με το υπόλοιπο πρόγραμμα, δηλαδή να δέχεται τιμές από το τμήμα προγράμματος που το καλεί, αλλά και να επιστρέφει, μετά την εκτέλεσή του, νέες τιμές.

Οι παράμετροι είναι λοιπόν οι μεταβλητές, μέσω των οποίων πραγματοποιείται το πέρασμα αυτών των τιμών.

## Τι είναι συνάρτηση, πως καλείται και ποια η δομή της;

Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).

Κάθε συνάρτηση εκτελείται, όπως ακριβώς εκτελούνται οι ενσωματωμένες συναρτήσεις της γλώσσας. Απλώς αναφέρεται το όνομά της σε μια έκφραση ή σε μία εντολή και επιστρέφεται η τιμή της.

Κάθε συνάρτηση έχει την ακόλουθη δομή:

**ΣΥΝΑΡΤΗΣΗ** όνομα (λίστα παραμέτρων) : τύπος συνάρτησης

Τμήμα δηλώσεων

**ΑΡΧΗ**

εντολές

όνομα ← έκφραση

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

Ξεκινά με τη λέξη ΣΥΝΑΡΤΗΣΗ και ένα έγκυρο όνομα στη ΓΛΩΣΣΑ.

Ακολουθούν οι παράμετροι εισόδου και ο τύπος της συνάρτησης, δηλαδή ο τύπος της τιμής που επιστρέφει.

Ακολουθεί η δήλωση των σταθερών (εάν υπάρχουν) και των μεταβλητών.

Ανάμεσα στις δεσμευμένες λέξεις ΑΡΧΗ και στο ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ, υπάρχει το τμήμα εντολών και **υποχρεωτικά** μία εντολή εκχώρησης της τιμής που θα επιστραφεί στο όνομα της συνάρτησης.

## Συναρτήσεις (2)

### Κλήση συνάρτησης

πραγματοποιείται με αναγραφή του ονόματός της (μαζί με τις παραμέτρους εισόδου) σε μία αλγοριθμική έκφραση.

π.χ.

η έκφραση:

$$X \leftarrow \text{Κάνε\_κάτι}(\alpha, \beta)$$

καλεί τη συνάρτηση Κάνε\_κάτι, πληροφωρώντας την για τις τιμές των μεταβλητών  $\alpha$ ,  $\beta$ .

Η συνάρτηση εκτελεί τον κώδικά της, υπολογίζει μία τιμή η οποία, μετά το τέλος εκτέλεσης της συνάρτησης, περνά στο πρόγραμμα μέσω του ονόματος της συνάρτησης και εκχωρείται στην  $X$ .

### άλλα χαρακτηριστικά συνάρτησης

- Στην είσοδό της πρέπει να έχει μία τουλάχιστον μεταβλητή (πaráμετρο).
- Υπολογίζει και επιστρέφει οπωσδήποτε μία και μόνο μία τιμή (ακέραια, πραγματική, χαρακτήρα ή λογική).
- Δεν διαβάζει ούτε γράφει (εκτυπώνει) δεδομένα δηλαδή δεν θα περιέχει τις εντολές ΔΙΑΒΑΣΕ και ΓΡΑΨΕ.
- Η συνάρτηση δηλώνεται μετά το τέλος του κυρίως προγράμματος.

## Συναρτήσεις / παράδειγμα

Πρόγραμμα που για κάθε ζευγάρι θετικών αριθμών που διαβάζει ως πλευρές ενός ορθογωνίου, υπολογίζει μέσω συνάρτησης και εκτυπώνει το εμβαδόν του.

**ΠΡΟΓΡΑΜΜΑ** παράδειγμα\_1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** α, β, εμβαδόν

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

α ← 2, β ← 3

**ΟΣΟ** (α > 0) **ΚΑΙ** (β > 0) **ΕΠΑΝΑΛΑΒΕ**

εμβαδόν ← ΟΡΘ(α, β)

εμβαδόν ← ΟΡΘ = 60

**ΓΡΑΨΕ** εμβαδόν

εκτυπώνεται το 60

**ΔΙΑΒΑΣΕ** α, β

α ← 2, β ← 4

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ** ΟΡΘ(x, y): **ΠΡΑΓΜΑΤΙΚΗ**

x ← α = 2,5, y ← β = 4

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** x, y

**ΑΡΧΗ**

ΟΡΘ ← x \* y

ΟΡΘ ← 2.5 \* 4 = 10

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## Συνάρτηση / Παράδειγμα 1

Να γραφούν δύο προγράμματα. Το πρώτο θα ζητά από το χρήστη 100 τιμές και θα υπολογίζει και εμφανίζει την απόλυτη τιμή κάθε μίας από αυτές, ενώ το δεύτερο θα ζητά από το χρήστη δύο τιμές και θα υπολογίζει την απόστασή τους στον άξονα των πραγματικών (απόλυτη τιμή της διαφοράς τους).

Ο υπολογισμός της απόλυτης τιμής θα γίνεται με τη χρήση της συνάρτησης  $A\_T(X)$ , την οποία θα υλοποιήσουμε. Δηλαδή θα θεωρήσουμε πως δεν υπάρχει και πως είμαστε οι πρώτοι που την υλοποιούμε.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_112\_ΑΠΟΛΥΤΗ\_ΤΙΜΗ\_1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** A

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

**ΔΙΑΒΑΣΕ** A

**ΓΡΑΨΕ**  $A\_T(A)$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_112\_ΑΠΟΛΥΤΗ\_ΤΙΜΗ\_2

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** A, B, Γ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** A, B

**Γ**  $\leftarrow A\_T(A - B)$

**ΓΡΑΨΕ** Γ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ**  $A\_T(X)$  : **ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** X

**ΑΡΧΗ**

**ΑΝ** X < 0 **ΤΟΤΕ**

X  $\leftarrow X * (-1)$

**ΤΕΛΟΣ\_ΑΝ**

**A\_T**  $\leftarrow X$

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## Συνάρτηση / Παράδειγμα 2

Να γραφεί πρόγραμμα που θα ζητά από το χρήστη μη αρνητικές και ακέραιες τιμές και θα εμφανίζει αν είναι άρτιες οι περιττές. Η επανάληψη θα σταματά μόλις δοθεί αρνητική τιμή, οπότε και θα εμφανίζει το πλήθος των περιττών και το πλήθος των άρτιων αριθμών που δόθηκαν.

Ο έλεγχος του αν είναι άρτιες οι περιττές, θα γίνεται με τη χρήση συνάρτησης, που θα επιστρέφει την τιμή ΑΛΗΘΗΣ εάν είναι άρτιος και ΨΕΥΔΗΣ εάν είναι περιττή.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_112\_ΣΥΝΑΡΤΗΣΗ\_ΑΡΤΙΟΙ\_ΠΕΡΙΤΤΟΙ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** X, C1, C2

**ΑΡΧΗ**

C1 ← 0

C2 ← 0

**ΔΙΑΒΑΣΕ** X

**ΟΣΟ** X >= 0 **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** ΕΛΕΓΧΟΣ(X) = ΨΕΥΔΗΣ **ΤΟΤΕ**

**ΓΡΑΨΕ** 'ΠΕΡΙΤΤΟΣ'

C1 ← C1 + 1

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'ΑΡΤΙΟΣ'

C2 ← C2 + 1

**ΤΕΛΟΣ\_ΑΝ**

**ΔΙΑΒΑΣΕ** X

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** C1, C2

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ** ΕΛΕΓΧΟΣ(X) : **ΛΟΓΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** X

**ΛΟΓΙΚΕΣ:** done

**ΑΡΧΗ**

**ΑΝ** X mod 2 = 0 **ΤΟΤΕ**

done ← **ΑΛΗΘΗΣ**

**ΑΛΛΙΩΣ**

done ← **ΨΕΥΔΗΣ**

**ΤΕΛΟΣ\_ΑΝ**

ΕΛΕΓΧΟΣ ← done

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**





## Άσκηση 2 / σελίδα 284

Να γραφεί συνάρτηση η οποία να επιστρέφει τον μέγιστο δύο πραγματικών αριθμών που θα δέχεται ως παραμέτρους. Στη συνέχεια να γραφεί πρόγραμμα που να διαβάζει τρεις πραγματικούς αριθμούς και κάνοντας χρήση της προαναφερθείσας συνάρτησης, να εμφανίζει τον μεγαλύτερο από αυτούς.

# Ενότητα 5

## Ασκήσεις

13, 14 / σελίδα 259

1, 3, 4 / σελίδα 284

(στις ασκήσεις 1, 3 και 4 δεν είναι απαραίτητο να γράψετε το πρόγραμμα, ξεκινήστε κατευθείαν από τη συνάρτηση)

Και την άσκηση στην προηγούμενη σελίδα