

## ΚΕΦΑΛΑΙΟ 2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΑΛΓΟΡΙΘΜΩΝ

**Αλγόριθμος** είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

Κάθε αλγόριθμος απαραίτητα **ικανοποιεί τα επόμενα κριτήρια:**

- **Είσοδος (input).** Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο.

*Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.*

- **Έξοδος (output).** Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
- **Καθοριστικότητα (definiteness).** Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης λαμβάνει μηδενική τιμή.
- **Περατότητα (finiteness).** Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του.

*Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).*

- **Αποτελεσματικότητα (effectiveness).** Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

**Η Πληροφορική, λοιπόν, μπορεί να ορισθεί ως η επιστήμη που μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:**

- **Υλικού.** Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή.
- **Γλωσσών Προγραμματισμού.** Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (δηλαδή, χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου.
- **Θεωρητική.** Το ερώτημα που συχνά τίθεται είναι, αν πράγματι υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος.
- **Αναλυτική.** Μελετώνται οι υπολογιστικοί πόροι (computer resources) που απαιτούνται από έναν αλγόριθμο, όπως για παράδειγμα το μέγεθος της κύριας και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες CPU και για λειτουργίες εισόδου/εξόδου κ.λπ.

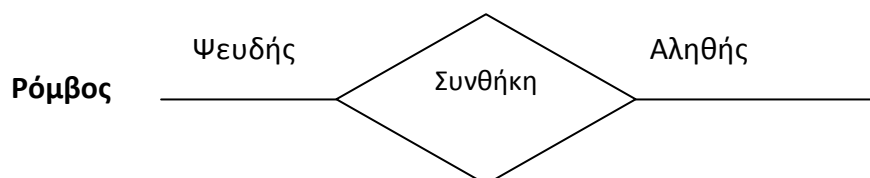
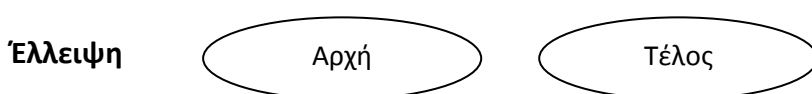
**Τρόποι αναπαράστασης ενός αλγορίθμου:**

- με ελεύθερο κείμενο
- με διαγραμματικές τεχνικές
- με φυσική γλώσσα
- με κωδικοποίηση

# Διάγραμμα ροής

Ένα **διάγραμμα ροής** αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών.

Τα **κυριότερα** χρησιμοποιούμενα **γεωμετρικά σχήματα** είναι τα εξής:



## Δομή ακολουθίας

Η ακολουθιακή δομή εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών.

## Δομή Επιλογής

Συνήθως τα προβλήματα έχουν κάποιες ιδιαιτερότητες και δεν ισχύουν τα ίδια βήματα για κάθε περίπτωση. Η πλέον συνηθισμένη περίπτωση είναι να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά για κάθε διαφορετικό στιγμιότυπο ενός προβλήματος.

Γενικά η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης.

**Αν <συνθήκη> τότε <εντολή>**

**Αν <συνθήκη> τότε  
<διαδικασία\_1>**

**αλλιώς  
<διαδικασία\_2>**

**Τέλος\_αν**

**Αν <συνθήκη\_1> τότε  
<διαδικασία\_1>**

**αλλιώς\_αν <συνθήκη\_2> τότε  
<διαδικασία\_2>**

.....

**αλλιώς\_αν <συνθήκη\_n> τότε  
<διαδικασία\_n>**

**αλλιώς  
<διαδικασία\_αλλιώς>**

**Τέλος\_αν**

## Δομή Επανάληψης

Η διαδικασία της επανάληψης είναι ιδιαίτερα συχνή, αφού πλήθος προβλημάτων μπορούν να επιλυθούν με κατάλληλες επαναληπτικές διαδικασίες. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό.

Όσο <συνθήκη> επανάλαβε  
    Διαδικασία  
Τέλος\_επανάληψης

Αρχή\_επανάληψης  
    Διαδικασία  
Μέχρις\_ότου <συνθήκη>

Για μεταβλητή από τ1 μέχρι τ2 με\_βήμα β  
    Διαδικασία  
Τέλος\_επανάληψης

# Στοιχεία Ψευδογλώσσας

## 1. Σταθερές

- **Αριθμητικές:** χρησιμοποιούνται οι αριθμητικοί χαρακτήρες, το +, το – και το κόμμα ως δεκαδικό σημείο
- **Αλφαριθμητικές:** σχηματίζονται από οποιουδήποτε χαρακτήρες εντός διπλών εισαγωγικών,
- **Λογικές:** υπάρχουν δύο, οι Αληθής και Ψευδής.

## 2. Μεταβλητές

Για τη σύνθεση του ονόματος μιας μεταβλητής χρησιμοποιούνται οι αριθμητικοί χαρακτήρες, οι αλφαβητικοί χαρακτήρες πεζοί και κεφαλαίοι, καθώς και ο χαρακτήρας `_` (underscore). Οι μεταβλητές μπορούν επίσης να είναι αριθμητικές, αλφαριθμητικές και λογικές.

## 3. Τελεστές

- **Αριθμητικοί:** +, -, \*, /, Λ, div, mod
- **Συγκριτικοί:** <, <=, #, >, >=
- **Λογικοί:** και (σύζευξη), ή (διάζευξη), όχι (άρνηση)

## 4. Εκφράσεις

Σχηματίζονται από σταθερές, μεταβλητές, συναρτήσεις, τελεστές και παρενθέσεις.

## 5. Εντολή εκχώρησης

Μεταβλητή <- έκφραση

## 6. Ρήματα σε προστακτική

Για παράδειγμα, "Διάβασε", "Γράψε", "Εκτέλεσε" κ,λπ.

## **7. Ουσιαστικά**

Σε ορισμένες περιπτώσεις όταν οι ζητούμενες ενέργειες είναι πολλές ή προφανείς, καθορίζονται με τη χρήση ουσιαστικών αντί ρημάτων, όπως "εισαγωγή δεδομένων", "εμφάνιση πεδίων στην οθόνη" κ.λπ.

## **8. Σχόλια**

Προκειμένου να διαχωρίζονται οι επεξηγηματικές φράσεις από τις λέξεις-κλειδιά του αλγορίθμου, στις πρώτες προτάσσεται το σύμβολο !, για παράδειγμα !Σχόλια.

**9. Πρώτη και τελευταία γραμμή** ενός αλγορίθμου είναι αντίστοιχα

**Αλγόριθμος** <όνομα\_αλγορίθμου> και **Τέλος** <όνομα\_αλγορίθμου>

## **10. Δεδομένα και αποτελέσματα**

Τα δεδομένα εισόδου (αν υπάρχουν) περιγράφονται στη δεύτερη γραμμή του αλγορίθμου εντός των συμβόλων //... //. Αντίστοιχα τα αποτελέσματα εξόδου δίνονται στην προτελευταία γραμμή του αλγορίθμου εντός των συμβόλων //... //.

## Πολλαπλασιασμός αλά ρωσικά

- Η μέθοδος αυτή χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ό,τι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασιασμού.
- Πιο συγκεκριμένα, απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση.
- Σε επίπεδο, λοιπόν, κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση διά δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης (shift), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία.

Αλγόριθμος: Πολλαπλασιασμός δύο θετικών ακεραίων (αλά ρωσικά)	
Είσοδος:	Δύο ακέραιοι $M1$ και $M2$ , όπου $M1, M2 \geq 1$
Έξοδος:	Το γινόμενο $P=M1*M2$
Βήμα 1	Θέσε $P=0$
Βήμα 2	Αν $M2>0$ , τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7
Βήμα 3	Αν ο $M2$ είναι περιττός, τότε θέσε $P=P+M1$
Βήμα 4	Θέσε $M1=M1*2$
Βήμα 5	Θέσε $M2=M2/2$ (θεώρησε μόνο το ακέραιο μέρος)
Βήμα 6	Πήγαινε στο Βήμα 2
Βήμα 7	Τύπωσε τον $P$ .