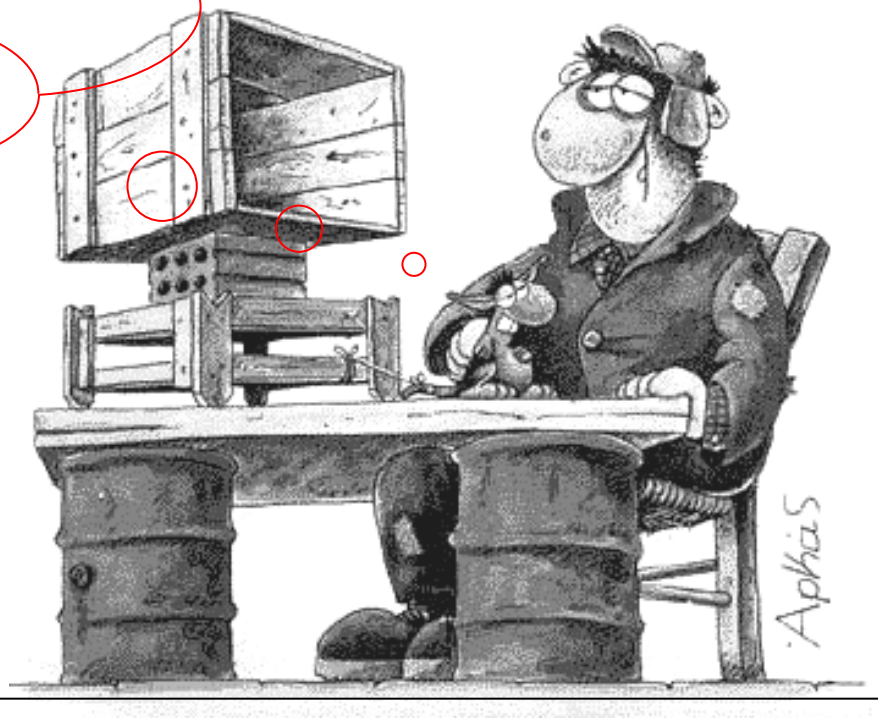
Εισαγωγή στον προγραμματισμό

Ενότητες:

- Η έννοια του προγράμματος
- Ιστορική αναδρομή
- Φυσικές και τεχνητές γλώσσες
- Τεχνικές σχεδίασης προγραμμάτων
  - Ιεραρχική Σχεδίαση
  - Τμηματικός Προγραμματισμός
  - Δομημένος προγραμματισμός
- Αντικειμενοστραφής προγραμματισμός
- Παράλληλος προγραμματισμός
- Προγραμματιστικά περιβάλλοντα

Και εγώ πως μπορώ να επικοινωνήσω με αυτό το «παλιοκούτι»;;; Σε τι γλώσσα να του δώσω οδηγίες για να μου λύσει τα προβλήματά μου ;;;

Μάλλον δεν έχει ακούσει ούτε για **προγράμματα** ούτε για τις **γλώσσες προγραμματισμού**. Τι άσχετος Θεέ μου!!!



Η έννοια του προγράμματος

- **Μια σειρά εντολών προς τον υπολογιστή (αλγόριθμο)**
- **Τα δεδομένα τα οποία πρέπει να επεξεργαστεί**

} **Πρόγραμμα**

Προγράμματα = Αλγόριθμοι + Δομές Δεδομένων

Η έννοια του προγράμματος

*Ένα **<u>πρόγραμμα</u>** είναι απλά **<u>μια «συνταγή» που εφαρμόζει ο υπολογιστής πάνω στα δεδομένα</u>** ώστε να αποκομίσει τα επιθυμητά αποτελέσματα.*

Είναι όπως περίπου η συνταγή ψησίματος ενός κέικ. Τα δεδομένα αντιστοιχούν στα συστατικά, συμπεριλαμβάνοντας και την θερμότητα που παρέχεται από την ηλεκτρική κουζίνα. Ο αλγόριθμος αντιστοιχεί στις οδηγίες της συνταγής για ανακάτεμα, αναμονή, θερμότητα, ψύξη και ότι άλλο μπορεί να επιτελεστεί επί των συστατικών. Η έξοδος μπορεί να αντιστοιχισθεί με το τελικό κέικ, έτοιμο για σερβίρισμα.

***Έτσι ένα πρόγραμμα μπορούμε να πούμε ότι συνίσταται από δυο μέρη, τα δεδομένα σύμφωνα με τα οποία αυτό ενεργεί και τον αλγόριθμο (κώδικα) που διαχειρίζεται τα δεδομένα**.*

Τεχνητές γλώσσες

Οι **γλώσσες προγραμματισμού** αναπτύχθηκαν, για να μπορεί ο προγραμματιστής να δίνει τις εντολές που πρέπει να εκτελέσει ο υπολογιστής.

Μια γλώσσα προσδιορίζεται από το:

- ✓ Αλφάβητο
- ✓ Λεξιλόγιο
- ✓ Γραμματική
  - ▪ Τυπικό ή τυπολογικό (accidence)
  - ▪ Συντακτικό (syntax)
- ✓ Σημασιολογία

Μα καλά γιατί έχουμε τόσες πολλές γλώσσες προγραμματισμού και δεν φτιάχνουμε μία γλώσσα που να είναι η καλύτερη από όλες

Η κάθε γλώσσα προγραμματισμού είναι φτιαγμένη για να εξυπηρετήσει συγκεκριμένους σκοπούς. Αν θέλουμε να επιλύσουμε μαθηματικά προβλήματα θα χρησιμοποιήσουμε την FORTAN, αν θέλουμε να μάθουμε προγραμματισμό θα χρησιμοποιήσουμε την BASIC ή την PASCAL, αν θέλουμε να φτιάξουμε ένα λειτουργικό σύστημα θα δουλέψουμε με την C. Ο προγραμματιστής καλείται να επιλέξει την “καλύτερη” γλώσσα για να υλοποιήσει το πρόγραμμα.

Η επιλογή της γλώσσας για την ανάπτυξη μιας εφαρμογής εξαρτάται:

- από το είδος της εφαρμογής,
- το υπολογιστικό περιβάλλον στο οποίο θα εκτελεστεί,
- τα προγραμματιστικά περιβάλλοντα που διαθέτουμε
- και κυρίως τις γνώσεις του προγραμματιστή.

Ιστορική αναδρομή- Γλώσσες προγραμματισμού

- **Γλώσσες μηχανής** : Ο Η/Υ μπορεί να καταλάβει μόνο μια ακολουθία από ψηφία 0 και 1
- **Συμβολικές γλώσσες (χαμηλού επιπέδου)** : Αντί για τις ακολουθίες του 0 και 1 χρησιμοποιούμε σύμβολα που είναι πιο κατανοητά από τον άνθρωπο μετά όμως ένα ειδικό πρόγραμμα (assembler) θα μεταφράσει τα σύμβολα σε δυαδικά ψηφία. Δεν υπάρχει μια συμβολική γλώσσα αλλά κάθε επεξεργαστής έχει το δικό του ρεπερτόριο εντολών που απαρτίζουν την συμβολική του γλώσσα. Επομένως κάθε συμβολική γλώσσα εξαρτάται από την αρχιτεκτονική του επεξεργαστή.
- **Γλώσσες υψηλού επιπέδου (3ης γενιάς)** Ο τρόπος έκφρασης είναι ακόμα πιο κατανοητός, υπάρχει δυνατότητα μεταφερσιμότητας, ευκολία στην εκμάθηση, και τέλος η διόρθωση και η συντήρηση των προγραμμάτων είναι ευκολότερη.
- **Γλώσσες 4ης γενιάς** Δεν απευθύνονται μόνο σε προγραμματιστές αλλά και προς τον απλό χρήστη του υπολογιστή αφού αποκρύπτουν όλο και περισσότερο την αρχιτεκτονική του υλικού και της τεχνικής του προγραμματισμού.

Ένα πρόγραμμα σε:

γλώσσα μηχανής		*συμβολική γλώσσα*	*γλώσσα υψηλού επιπέδου*
10101000 00001010		INDEX=$01	sum = 0
10001100 00000001		SUM=$02	**FOR** index=1 **TO** 10
00111100		LDA #10	sum=sum+index
01010001 00000001		STA INDEX	**NEXT** index
01000011 00000001		CLA	**END**
11000000 11111010	LOOP	ADD INDEX	
10001100 00000010		DEC INDEX	
11111111		BNE LOOP	
		STA SUM	
		BRK	

Υπολογισμός του αθροίσματος των αριθμών 1 έως 10

Γλώσσες υψηλού επιπέδου

- FORTAN (FORmula TRANslation) *1957*
- COBOL (Common Business Oriented Language) *1960*
- ALGOL (ALGOrithmic Language) *1960*
- PL/1 Programming Languagne/1 *μέσα ’60*
- LISP (LISt Processor) *1959*
- PROLOG (PROgramming LOGic) αρχές ’70
- LOGO (λόγος) 1967
- BASIC (Beginner’s All Purpose Symbolic Instruction Code) 1964
- PASCAL (1970)
- C (1972)
- JAVA

Ταξινόμηση γλωσσών προγραμματισμού

- Οι περισσότερες γλώσσες είναι **Διαδικασιακές** (procedural) ή αλγοριθμικές

Άλλες κατηγορίες είναι:

- **Αντικειμενοστραφείς** γλώσσες (object – oriented) π.χ. C
- **Συναρτησιακές** γλώσσες (functional) π.χ. LISP
- **Μη διαδικασιακές** γλώσσες (non procedural) π.χ. PROLOG
- Γλώσσες **ερωταπαντήσεων** (query) π.χ. SQL

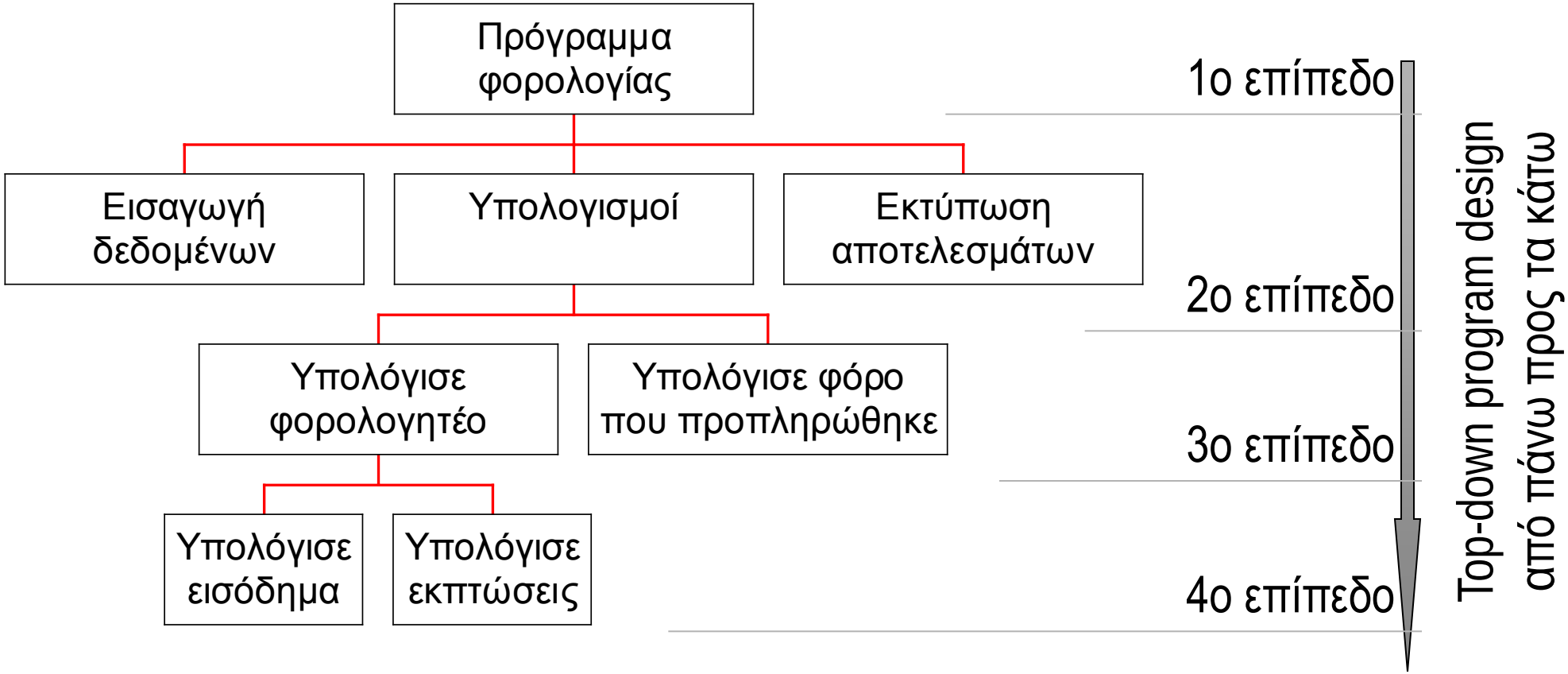
Ταξινόμηση γλωσσών προγραμματισμού με βάση την περιοχή χρήσης τους

- **Γενικής χρήσης** π.χ. BASIC, PASCAL
  - **Επιστημονικής** κατεύθυνσης (science – oriented) π.χ. FORTAN
  - **Εμπορικής** κατεύθυνσης (business oriented) π.χ. COBOL
- **Προγραμματισμού συστημάτων** (system programming) π.χ. C
- **Τεχνητής νοημοσύνης** (artificial intelligence) π.χ. LISP, PROLOG
- **Ειδικής χρήσης**. (χρησιμοποιούνται σε ειδικές περιοχές εφαρμογών όπως στα γραφικά, στη ρομποτική, στη σχεδίαση IC, στην εκπαίδευση κ.α.)

Τεχνικές σχεδίασης προγραμμάτων

- Ιεραρχική σχεδίαση προγράμματος
- Τμηματικός προγραμματισμός
- Δομημένος προγραμματισμός

Ιεραρχική σχεδίαση προγράμματος



Δομημένος Προγραμματισμός

Ο δομημένος προγραμματισμός **είναι μία μεθοδολογία σύνταξης προγραμμάτων** που έχει σκοπό να βοηθήσει τον προγραμματιστή στην ανάπτυξη σύνθετων προγραμμάτων, να μειώσει τα λάθη, να εξασφαλίσει την εύκολη κατανόηση των προγραμμάτων και να διευκολύνει τις διορθώσεις και τις αλλαγές σε αυτά. Ο δομημένος προγραμματισμός στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών:

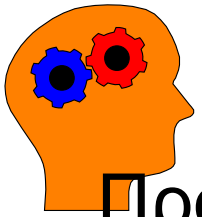
- τη δομή της ακολουθίας
- τη δομή της επιλογής
- και τη δομή της επανάληψης

Όλα τα προγράμματα μπορούν να γραφούν χρησιμοποιώντας μόνο αυτές τις τρεις δομές καθώς και συνδυασμό τους. Κάθε πρόγραμμα όπως και κάθε ενότητα προγράμματος έχει μόνο μία είσοδο και μόνο μία έξοδο.

Πλεονεκτήματα του δομημένου προγραμματισμού

- ✓ Δημιουργία απλούστερων προγραμμάτων.
- ✓ Άμεση μεταφορά των αλγορίθμων σε προγράμματα.
- ✓ Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα.
- ✓ Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
- ✓ Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
- ✓ Ευκολότερη διόρθωση και συντήρηση.

Προγραμματιστικά περιβάλλοντα



Πρόγραμμα
σε γλώσσα
προγραμματισμού

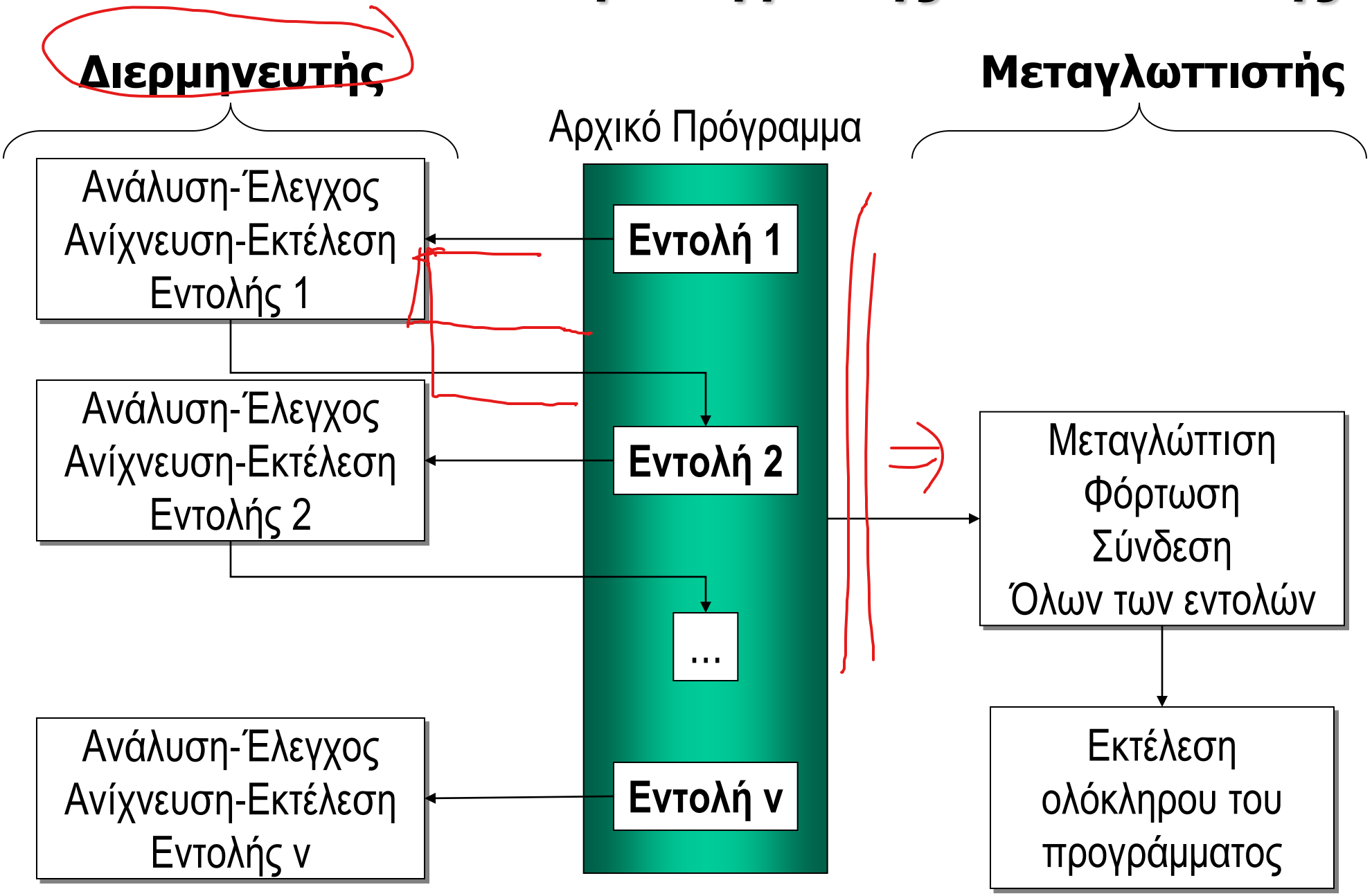
ΜΕΤΑΦΡΑΣΗ

Πρόγραμμα
σε γλώσσα
μηχανής

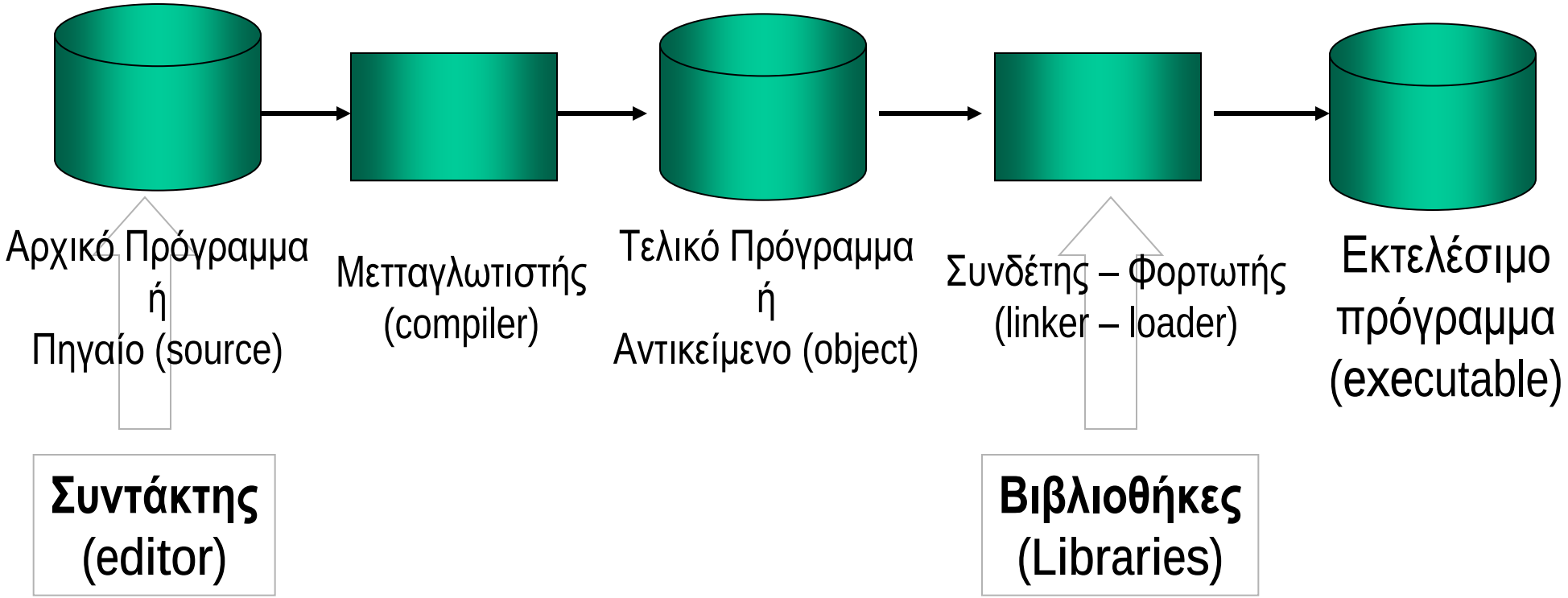
Μεταφραστικά προγράμματα:

- μεταγλωττιστές (compilers)
- διερμηνευτές (interpreters)

Διαδικασία μετάφρασης και εκτέλεσης



Μεταγλώττιση και σύνδεση προγράμματος



ΚΑΤΗΓΟΡΙΕΣ ΛΑΘΩΝ

ΣΥΝΤΑΚΤΙΚΑ

Κατά την μεταγλώττιση

- Λανθασμένη γραφή δεσμευμένων λέξεων
- Λανθασμένη γραφή αναγνωριστικών
- Λανθασμένη χρήση σημείων στίξης
- Λανθασμένη σύνταξη εντολών
- Λανθασμένη σύνταξη παραστάσεων
- Παράλειψη δήλωσης

ΛΟΓΙΚΑ

Κατά την εκτέλεση

- Λανθασμένη απόδοση αλγορίθμου
- Κακή σχεδίαση προγράμματος
- Μη σωστός καθορισμός του προβλήματος
- Περιορισμοί της γλώσσας ή του συστήματος

ΕΛΕΓΧΟΣ → **ΕΚΣΦΑΛΜΑΤΩΣΗ**