

Εισαγωγή στον προγραμματισμό

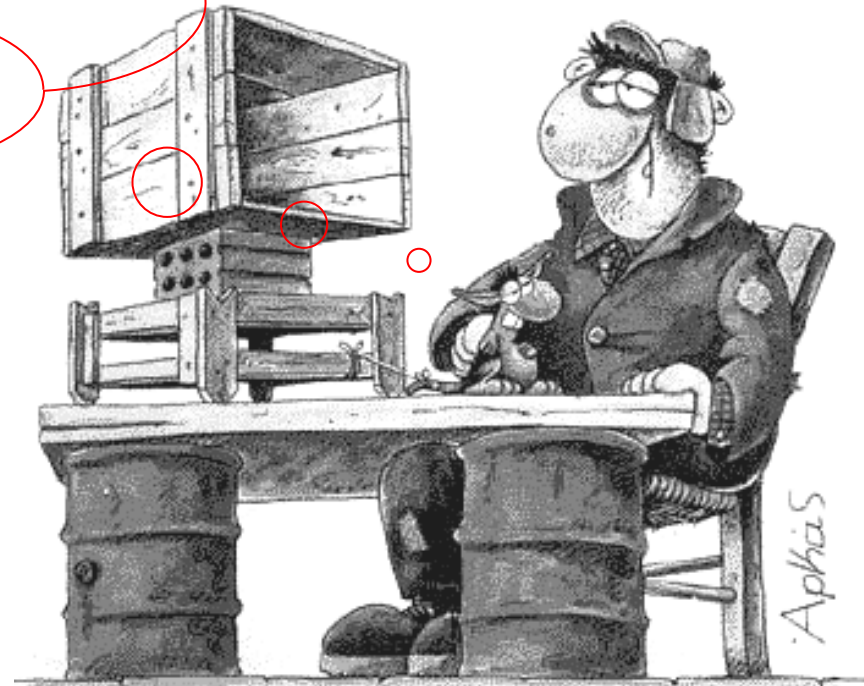
Ενότητες:

- Η έννοια του προγράμματος
- Φυσικές και τεχνητές γλώσσες
- Τεχνικές σχεδίασης προγραμμάτων
 - Ιεραρχική Σχεδίαση
 - Τμηματικός Προγραμματισμός
 - Δομημένος προγραμματισμός
- Αντικειμενοστραφής 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 πρόγραμμα είναι απλά μια «συνταγή» που εφαρμόζει ο υπολογιστής πάνω στα δεδομένα 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.

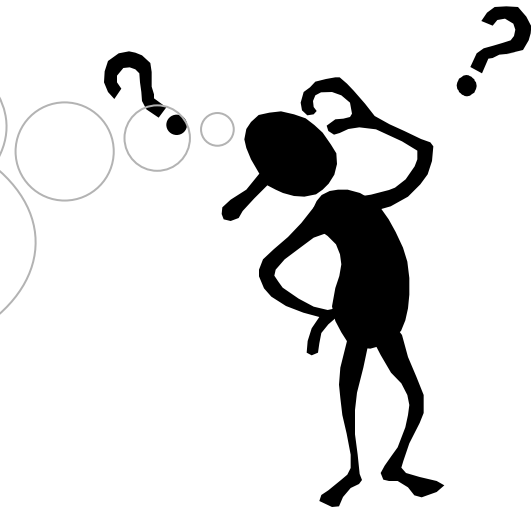
Τεχνητές γλώσσες

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.

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	
10001100	00000010	ADD INDEX	
11111111		DEC INDEX	
		BNE LOOP	
		STA SUM	
		BRK	

Υπολογισμός του αθροίσματος των αριθμών 1 έως 10

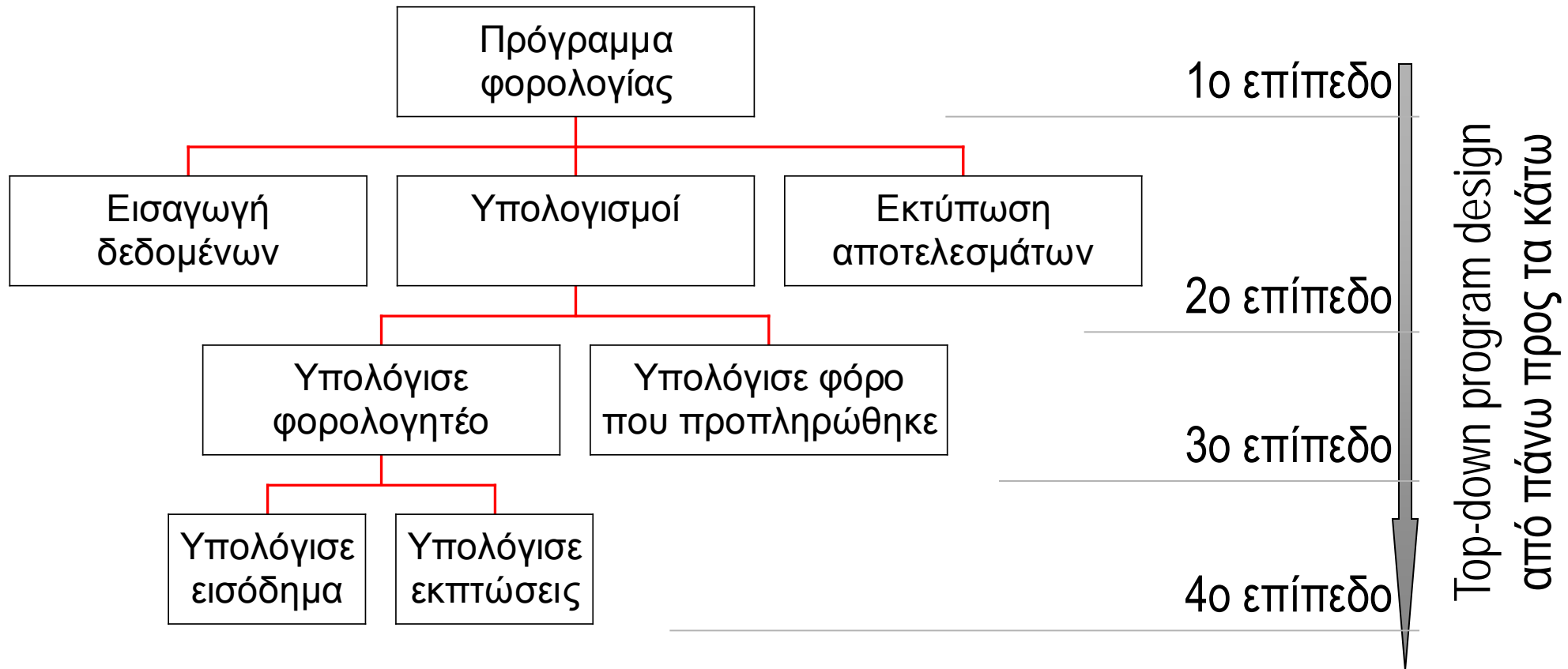
➤ Γλώσσες υψηλού επιπέδου

- ⌘ FORTAN (FORmula TRANslation) 1957
- ⌘ COBOL (Common Business Oriented Language) 1960
- ⌘ ALGOL (ALGOrithmic Language) 1960
- ⌘ PL/1 Programming Language/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

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:

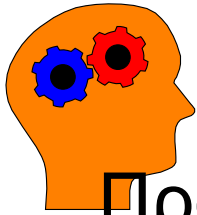
- τη δομή της ακολουθίας
- τη δομή της επιλογής
- και τη δομή της επανάληψης

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

Διερμηνευτής

Μεταγλωττιστής

Αρχικό Πρόγραμμα

Ανάλυση-Έλεγχος
Ανίχνευση-Εκτέλεση
Εντολής 1

Ανάλυση-Έλεγχος
Ανίχνευση-Εκτέλεση
Εντολής 2

Ανάλυση-Έλεγχος
Ανίχνευση-Εκτέλεση
Εντολής n

Εντολή 1

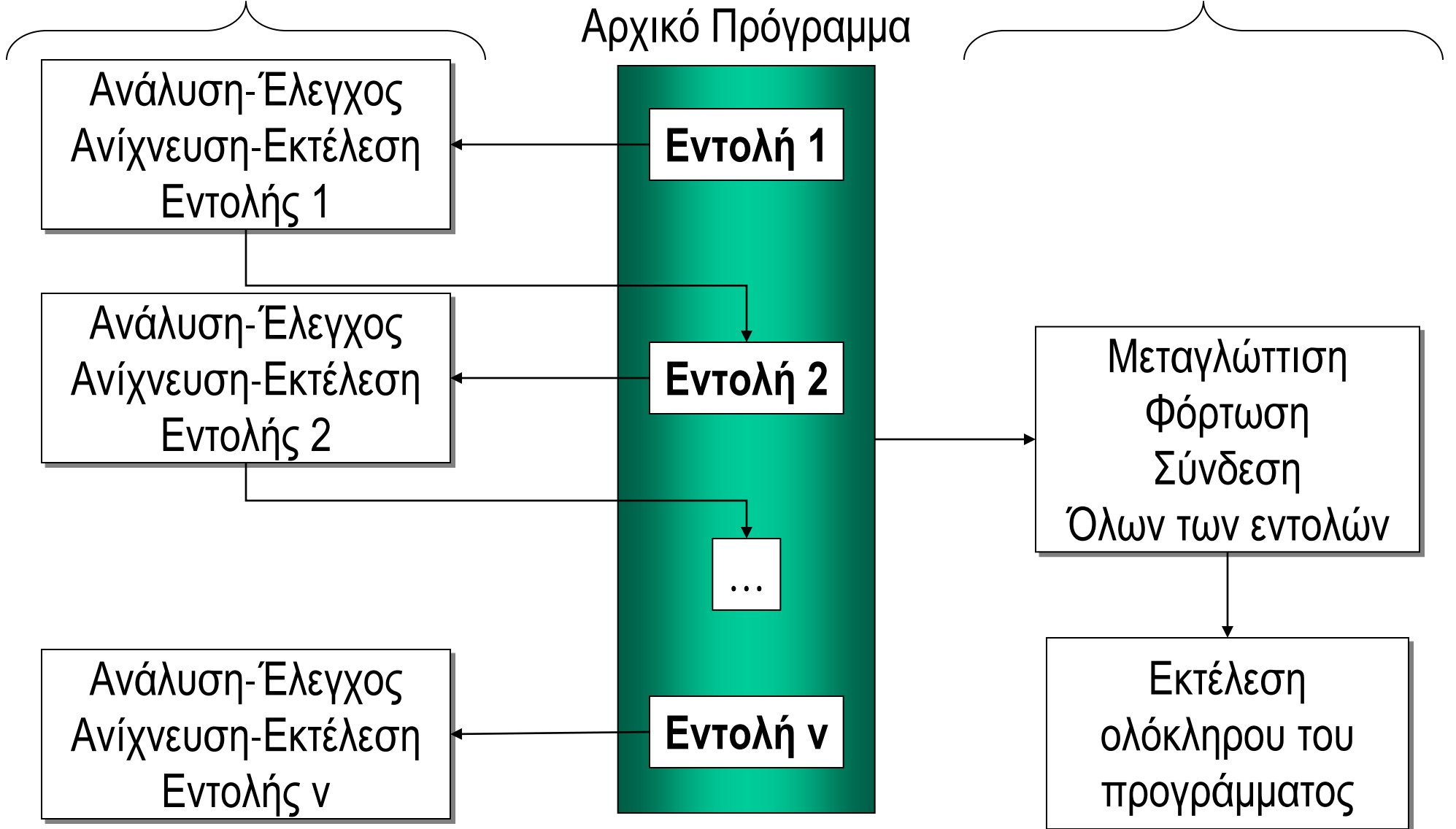
Εντολή 2

...

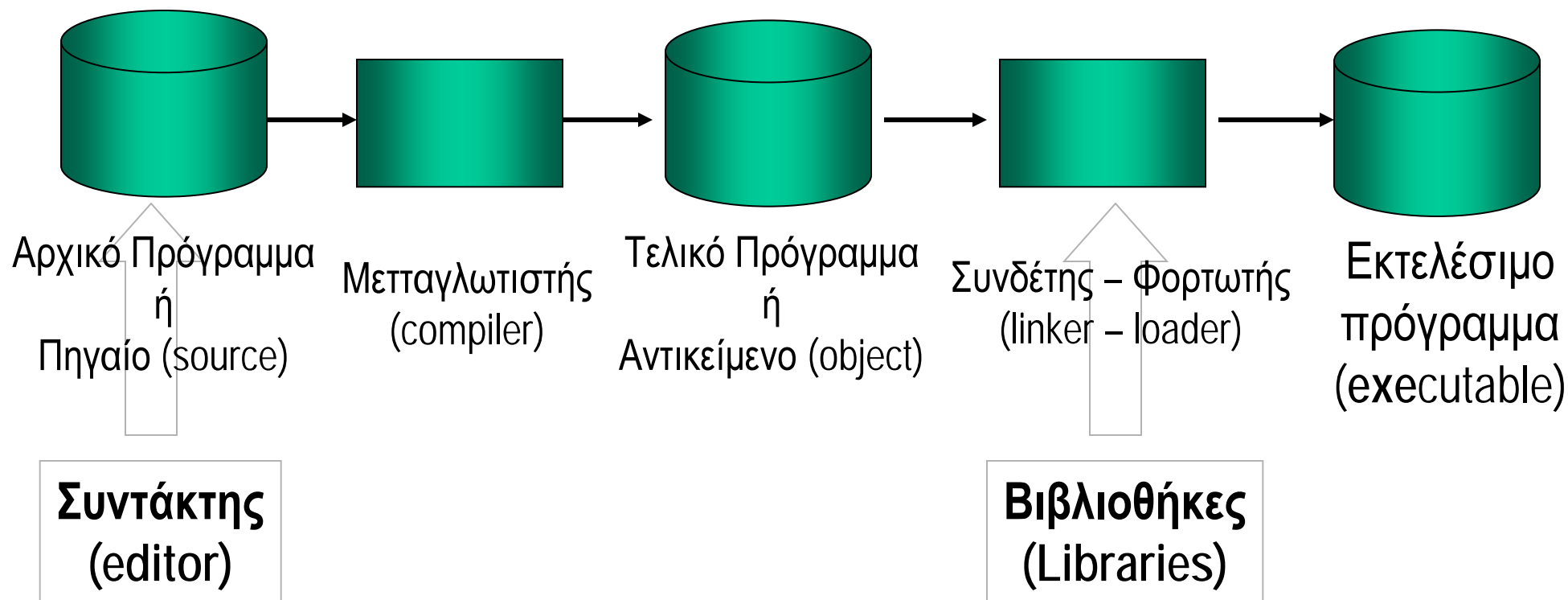
Εντολή n

Μεταγλώττιση
Φόρτωση
Σύνδεση
Όλων των εντολών

Εκτέλεση
ολόκληρου του
προγράμματος



Μεταγλώττιση και σύνδεση προγράμματος



ΚΑΤΗΓΟΡΙΕΣ ΛΑΘΩΝ

ΣΥΝΤΑΚΤΙΚΑ

Κατά την μεταγλώττιση

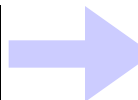
- Λανθασμένη γραφή δεσμευμένων λέξεων
- Λανθασμένη γραφή αναγνωριστικών
- Λανθασμένη χρήση σημείων στίξης
- Λανθασμένη σύνταξη εντολών
- Λανθασμένη σύνταξη παραστάσεων
- Παράλειψη δήλωσης

ΛΟΓΙΚΑ

Κατά την εκτέλεση

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

ΕΛΕΓΧΟΣ



ΕΚΣΦΑΛΜΑΤΩΣΗ

ΚΕΦΑΛΑΙΟ 6.5 - Αντικειμενοστραφής προγραμματισμός

1. Ποια είναι πρωτεύοντα δομικά στοιχεία ενός προγράμματος στην αντικειμενοστραφή σχεδίαση;

Η αντικειμενοστραφής σχεδίαση εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα αντικείμενα (objects).

2. Ποια τα πλεονεκτήματα της αντικειμενοστραφούς σχεδίασης;

Η αντικειμενοστραφής σχεδίαση αποδείχθηκε ότι επιφέρει καλύτερα αποτελέσματα, αφού τα προγράμματα που δημιουργούνται είναι περισσότερο ευέλικτα και επαναχρησιμοποιήσιμα.

3. Ο αντικειμενοστραφής προγραμματισμός χρησιμοποιεί τεχνικές του κλασσικού διαδικασιακού προγραμματισμού;

Ο αντικειμενοστραφής προγραμματισμός εκτός από τον τρόπο που χειρίζεται τα δεδομένα, συνεχίζει να χρησιμοποιεί την ιεραρχική σχεδίαση, τον τμηματικό προγραμματισμό και να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

Ερωτήσεις κεφαλαίου

- 6.1 Ποια είναι τα στάδια επίλυσης ενός προβλήματος με χρήση υπολογιστή;
- 6.2 Από ποια στοιχεία προσδιορίζεται μια γλώσσα, φυσική ή τεχνητή;
- 6.3 Ποιες είναι οι διαφορές φυσικών και τεχνητών γλωσσών;
- 6.4 Τι γνωρίζετε για την ιεραρχική σχεδίαση προγράμματος;
- 6.5 Τι γνωρίζετε για τον τμηματικό προγραμματισμό;
- 6.6 Τι γνωρίζετε για τον αντικειμενοστραφή προγραμματισμό;
- 6.7 Τι είναι ο δομημένος προγραμματισμός και στην χρήση ποιων δομών στηρίζεται;
- 6.8 Ποια είναι τα πλεονεκτήματα του δομημένου προγραμματισμού;
- 6.9 Τι είναι οι μεταγλωττιστές και τι οι διερμηνευτές;
- 6.10 Περιγράψτε τη διαδικασία μεταγλώττισης και σύνδεσης ενός προγράμματος
- 6.11 Ποιες είναι οι κατηγορίες λαθών και τι γνωρίζετε σε γενικές γραμμές για αυτές; Ποια λάθη ανιχνεύονται από το μεταγλωττιστή ή τον διερμηνευτή;
- 6.12 Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα του μεταγλωττιστή και του διερμηνευτή;
- 6.13 Ποια είναι τα προγράμματα που χρησιμοποιούνται για τη δημιουργία, τη μετάφραση και την εκτέλεση ενός προγράμματος και αναφέρατε λίγα λόγια για το κάθε ένα.