

# Εισαγωγή στον προγραμματισμό

---

Μέρος 1<sup>ο</sup>

Ιστορική αναδρομή γλωσσών προγραμματισμού

§ 6.1

# Η έννοια του προγράμματος

---

Η επίλυση ενός προβλήματος με τη χρήση υπολογιστή περιλαμβάνει τρία **στάδια**

1. Τον ακριβή προσδιορισμό του προβλήματος
2. Την ανάπτυξη του αντίστοιχου αλγορίθμου
3. Τη διατύπωση του αλγορίθμου σε μορφή κατανοητή απ' τον υπολογιστή

# Η έννοια του προγράμματος

---

- Ο προγραμματισμός ασχολείται με τη διατύπωση του αλγορίθμου σε μορφή κατανοητή απ' τον υπολογιστή
- Βασικά στοιχεία του προγράμματος είναι
  - τα δεδομένα και οι δομές δεδομένων στις οποίες ενεργεί
- **Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα**

# Η έννοια του προγράμματος

---

- Ο προγραμματισμός δίνει την ψευδαίσθηση ότι οι υπολογιστές είναι **έξυπνες μηχανές** με την ικανότητα να επιλύουν πολύπλοκα προβλήματα
- Οι υπολογιστές στην πραγματικότητα «καταλαβαίνουν» μόνο δύο καταστάσεις: **αληθές και ψευδές**, μηδέν και ένα, ανοιχτό και κλειστό
- Ο υπολογιστής κάνει στοιχειώδεις ενέργειες σε ακολουθίες **δυαδικών ψηφίων** με πολύ μεγάλη ταχύτητα

# Εισαγωγή στον προγραμματισμό

---

Μέρος 2<sup>ο</sup>

Φυσικές και τεχνητές γλώσσες  
Τεχνικές σχεδίασης προγραμμάτων  
§§ 6.3 – 6.5

# Εισαγωγή στον προγραμματισμό

---

- Φυσικές και τεχνητές γλώσσες
- Τεχνικές σχεδίασης προγραμμάτων
- Αντικειμενοστρεφής προγραμματισμός

# Φυσικές και τεχνητές γλώσσες

---

- Αλφάβητο
  - ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
  - αβγδεζηθικλμνξοπρστυφχψω
  - 0123456789
  - .,-;!
- Λεξιλόγιο
  - ΓΡΑΨΕ ≠ ΓΡΑΨΑ

# Φυσικές και τεχνητές γλώσσες

---

- **Γραμματική**
  - **Τυπικό ή τυπολογικό (accidence):** σύνολο κανόνων που ορίζει τις μορφές που μια λέξη είναι αποδεικτή
  - **Συντακτικό (syntax):** σύνολο κανόνων που ορίζει τη νομιμότητα της διάταξης και σύνδεση των λέξεων

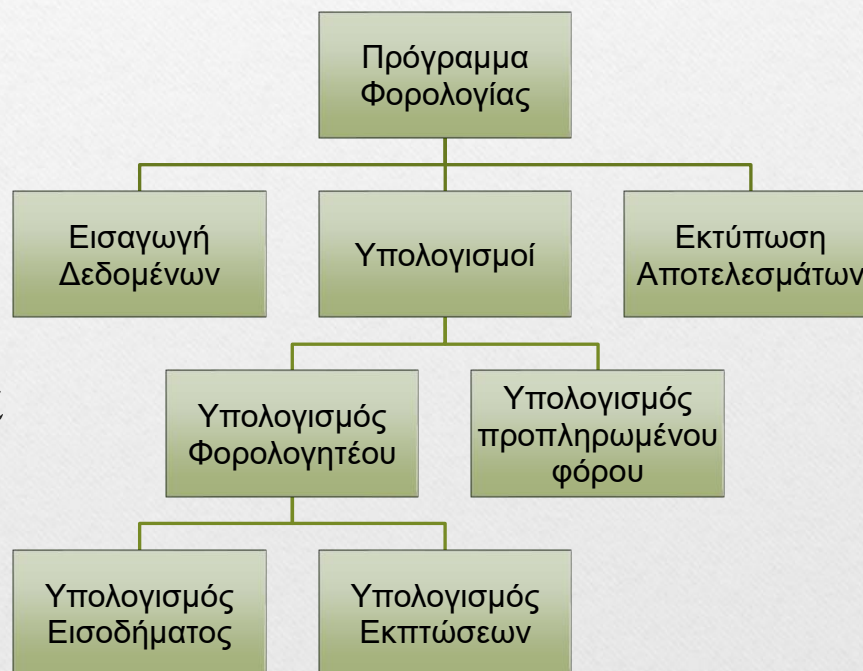
# Φυσικές και τεχνητές γλώσσες

---

- **Σημασιολογία** (semantics): σύνολο κανόνων που ορίζει το νόημα των λέξεων και των εκφράσεων
- **Διαφορές** φυσικών και τεχνητών γλωσσών
  - Δυνατότητα εξέλιξης

# Ιεραρχική σχεδίαση προγράμματος

- Top-down program design
- Διαιρούμε το πρόβλημα σε μια σειρά από απλούστερα υποπροβλήματα



# Τμηματικός προγραμματισμός

---

- Υλοποιεί την ιεραρχική σχεδίαση
- Κάθε υποπρόβλημα αποτελεί ανεξάρτητη **ενότητα** (module)
- Κάθε module γράφεται ξεχωριστά απ' τα υπόλοιπα
- Ο τμηματικός προγραμματισμός:
  - **Διευκολύνει τη δημιουργία του προγράμματος**
  - **Μειώνει τα λάθη**
  - **Επιτρέπει την ευκολότερη παρακολούθηση, κατανόηση και συντήρηση του προγράμματος από τρίτους**

# Δομημένος προγραμματισμός

---

- Έχει επικρατήσει απόλυτα
- Τον υποστηρίζουν σχεδόν όλες οι γλώσσες προγραμματισμού
- Παρουσιάστηκε στα μέσα του 1960

# Δομημένος προγραμματισμός

---

- Στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών ελέγχου της ροής του προγράμματος:
  - Δομή ακολουθίας
  - Δομή επιλογής
  - Δομή επανάληψης
- Κάθε πρόγραμμα, όπως και κάθε ενότητα προγράμματος, έχει μόνο μία **είσοδο** και μόνο μία **έξοδο**

# GOTO Το μαύρο πρόβατο του προγραμματισμού

---

- Η GOTO επέτρεπε την αλλαγή της ροής του προγράμματος πηγαίνοντας σε όποια εντολή του προγράμματος θέλαμε
- Οι αλληπάλληλες διακλαδώσεις κάνανε τα προγράμματα δύσκολα στην συντήρησή τους
- Ο Δομημένος Προγραμματισμός προσέφερε εναλλακτικές δομές ελέγχου της ροής του προγράμματος καθιστώντας τη GOTO περιττή

## Τα 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

# Αντικειμενοστρεφής προγραμματισμός

---

- Αντικείμενα που δρουν το ένα πάνω στο άλλο
- Κάθε **αντικείμενο** είναι ικανό
  - να λαμβάνει μηνύματα,
  - να επεξεργάζεται δεδομένα και
  - να στέλνει μηνύματα σε άλλα αντικείμενα.

# Αντικειμενοστρεφής προγραμματισμός

---

- Τάξη (class)
- Αντικείμενο (object)
- Εγκιβωτισμός (encapsulation)
- Κληρονομικότητα (inheritance)
- Αφαίρεση (abstraction)
- Πολυμορφισμός (polymorphism)

# Εισαγωγή στον προγραμματισμό

---

Μέρος 3<sup>ο</sup>

Προγραμματιστικά περιβάλλοντα

§ 6.7

# Προγραμματιστικά περιβάλλοντα

---

- Μεταγλωττιστές και διερμηνευτές
- Πηγαίο και αντικείμενο πρόγραμμα
- Βιβλιοθήκες
- Συνδέτης-φορτωτής
- Εκτελέσιμο πρόγραμμα

# Μεταγλωττιστές (Compilers)

---

- Ο μεταγλωττιστής δέχεται ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (0100100011)

# Διερμηνευτές (Interpreters)

---

- Ο διερμηνευτής διαβάσει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής

# Πηγαίο και αντικείμενο πρόγραμμα

---

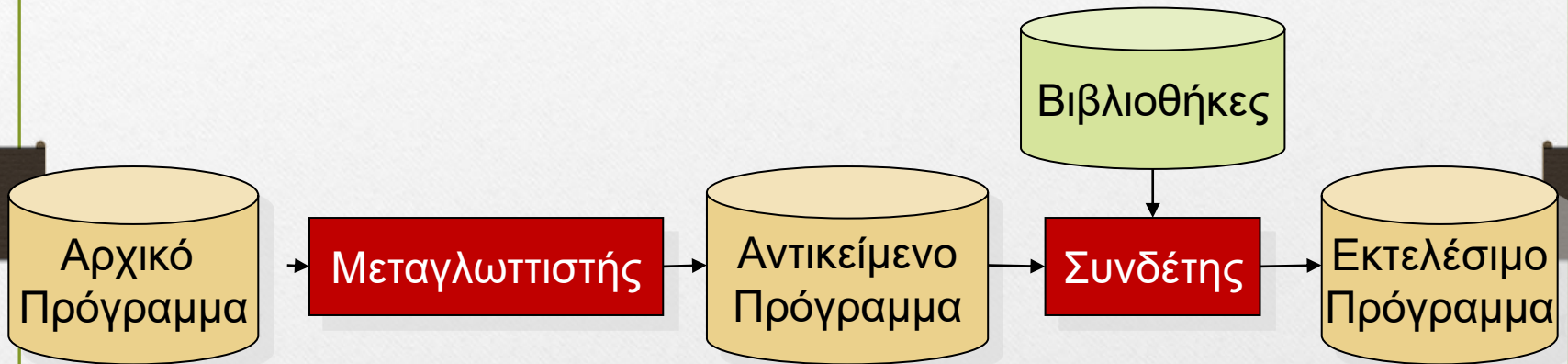
- **Πηγαίο πρόγραμμα** (source code)  
λέγεται το αρχικό πρόγραμμα που είναι γραμμένο σε γλώσσα υψηλού επιπέδου
- **Αντικείμενο πρόγραμμα** (object code)  
λέγεται το πρόγραμμα που παράγεται απ' τον μεταγλωττιστή

## Βιβλιοθήκες (libraries) και συνδέτης-φορτωτής (linker-loader)

---

- Το **αντικείμενο** πρόγραμμα είναι γλώσσα μηχανής αλλά συνήθως χρειάζεται επιπλέον κώδικα για να εκτελεστεί
- Ο επιπλέον κώδικας είναι ήδη μεταγλωττισμένος και αποθηκευμένος σε **βιβλιοθήκες**
- Ο **συνδέτης-φορτωτής** συνδέει το αντικείμενο πρόγραμμα με τις βιβλιοθήκες και παράγει το **εκτελέσιμο πρόγραμμα** (executable)

# Μεταγλώττιση και σύνδεση προγράμματος



# Λάθη προγράμματος

---

- Ένα πρόγραμμα μπορεί να έχει
  - **Γραμματικά και συντακτικά λάθη** που εντοπίζονται κατά τη φάση της μεταγλώττισης
  - **Λογικά λάθη**, λάθη δηλαδή στο σχεδιασμό και στους αλγορίθμους του προγράμματος, τα οποία εμφανίζονται μόνο κατά την εκτέλεση του προγράμματος