

Κεφάλαιο 2.3: Προγραμματισμός

2.3.1 Αναφορά σε γλώσσες προγραμματισμού και «Προγραμματιστικά Υποδείγματα»

2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Πρόγραμμα: σύνολο εντολών που χρειάζεται να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος. Βασικό στοιχείο του προγράμματος, εκτός από τον αλγόριθμο που υλοποιεί, είναι τα δεδομένα και οι δομές δεδομένων που επεξεργάζεται.

Προγραμματισμός: η εργασία σύνταξης των προγραμμάτων σε κάποια γλώσσα προγραμματισμού

Προγραμματιστές: τα άτομα που γράφουν και συντάσσουν προγράμματα

Γλώσσες μηχανής: ακολουθία δυαδικών ψηφίων (bits 1/0), που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Ήταν επίπονες και ελάχιστοι μπορούσαν να τις υλοποιήσουν, αφού απαιτούσαν βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή. Οι εντολές ενός προγράμματος και σήμερα μετατρέπονται σε γλώσσα μηχανής.

Συμβολικές γλώσσες: εντολές που αποτελούνται από συμβολικά ονόματα τα οποία αντιστοιχούν σε εντολές της γλώσσας μηχανής. Το έργο της μετάφρασης των εντολών σε γλώσσα μηχανής το αναλαμβάνει ένα ειδικό πρόγραμμα, ο συμβολομεταφραστής (assembler). Παραμένουν στενά συνδεδεμένες με την αρχιτεκτονική του κάθε υπολογιστή και η έλλειψη εντολών σύνθετων λειτουργιών οδηγεί σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και να συντηρηθούν.

2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Γλώσσες υψηλού επιπέδου: φτιάχτηκαν για την καλύτερη επικοινωνία ανθρώπου – μηχανής. Πλεονεκτήματα:

- φυσικότερος και πιο ανθρώπινος ο τρόπος έκφρασης των προβλημάτων.
- «μεταφερισιμότητα»: δυνατότητα εκτέλεσης των προγραμμάτων σε οποιοδήποτε υπολογιστή.
- εύκολη εκμάθηση, διόρθωση των λαθών και συντήρηση των προγραμμάτων.

Γλώσσες:

- FORTRAN (FORmula TRANslation – μετάφραση τύπων): κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων. (1957 – IBM)
- COBOL (COMmon Oriented Language – Κοινή γλώσσα προσανατολισμένη στις επιχειρήσεις): κατάλληλη για την ανάπτυξη εμπορικών εφαρμογών. (1960)
- LISP (LISt Processor – Επεξεργαστής Λίστας) και PROLOG (PROgramming LOGic – Λογικός Προγραμματισμός): γλώσσες της Τεχνητής Νοημοσύνης που χρησιμοποιούνται στα έμπειρα συστήματα, τα παιχνίδια, την επεξεργασία φυσικών γλωσσών κλπ. (μέσα δεκ. 60)
- BASIC (Beginner's All Purpose Symbolic Instruction Code – Συμβολικός Κώδικας Εντολών Γενικής Χρήσης για Αρχάριους): για την εκπαίδευση αρχάριων στον προγραμματισμό (δεκ. 60). Αργότερα έγινε η δημοφιλέστερη γλώσσα στους προσωπικούς ΗΥ (Visual Basic).

2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

- PASCAL (Niklaus Wirth): για την εκπαίδευση και τη δημιουργία ισχυρών προγραμμάτων. Ήταν η βάση για ισχυρότερες γλώσσες (ADA, Modula-2) (1970).
- C : ανάπτυξη του λειτουργικού συστήματος UNIX και δομημένων εφαρμογών. Έχει δυνατότητες γλώσσας χαμηλού επιπέδου. (1972-Bell)
- C++ : αντικειμενοστραφής γλώσσα, εξέλιξη της C.
- JAVA : ανάπτυξη αντικειμενοστραφών εφαρμογών εκτελέσιμων σε κατανομημένα περιβάλλοντα, σε διαφορετικούς (με διαφορετικά λειτουργικά συστήματα) ΗΥ δηλ. συνδεδεμένους στο Διαδίκτυο.

Σε ένα **οπτικό περιβάλλον** προγραμματισμού υπάρχει η δυνατότητα να δημιουργείται γραφικά ολόκληρο το περιβάλλον της εφαρμογής, όπως για παράδειγμα τα πλαίσια διαλόγου ή τα μενού.

Γλώσσες 4ης γενιάς

Παρέχουν τη δυνατότητα υποβολής ερωτήσεων στο σύστημα, ανάπτυξης εφαρμογών που ανακτούν πληροφορίες από βάσεις δεδομένων και καθορισμού του τρόπου εμφάνισης αυτών.

π.χ. γλώσσες ερωταπαντήσεων (query languages) όπως η SQL.

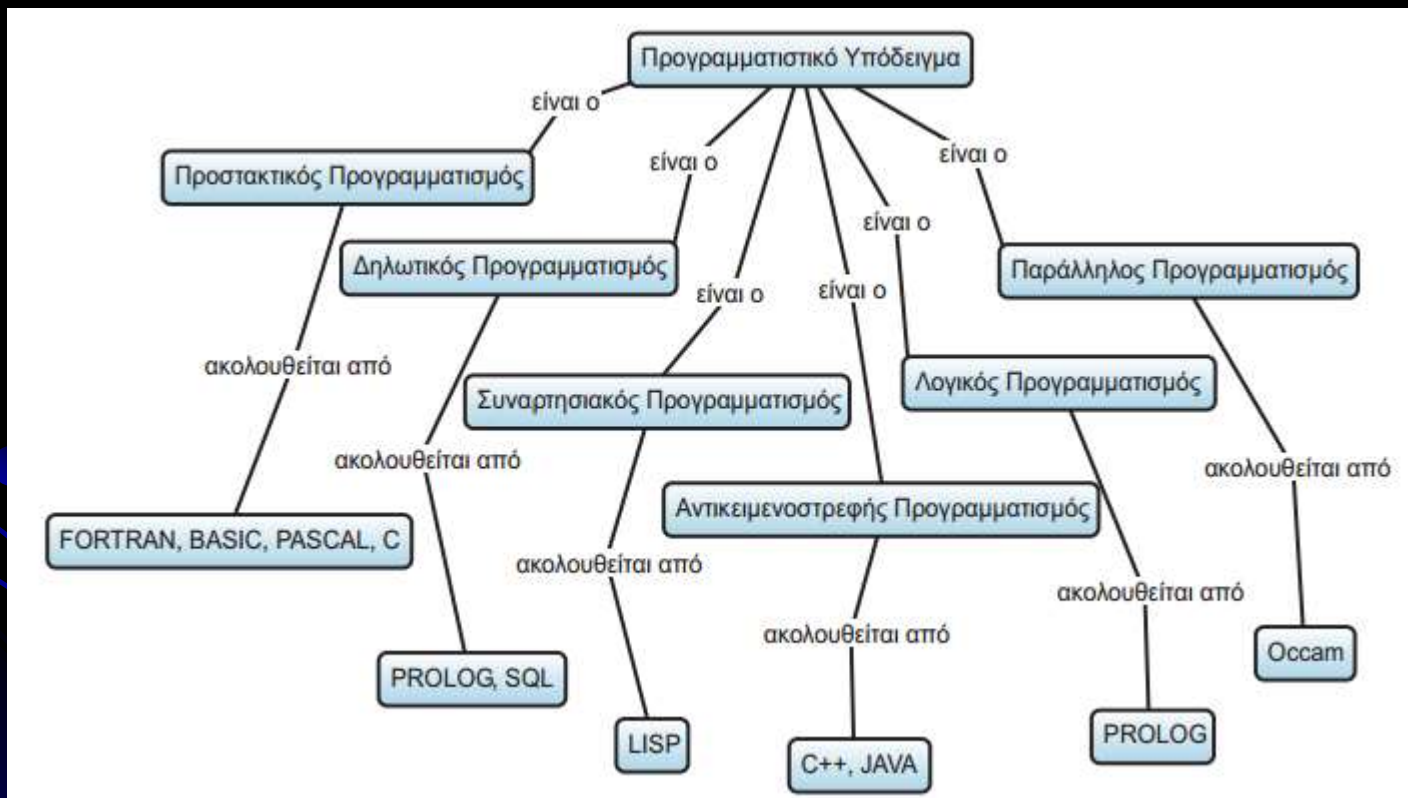
2.3.1.2 Προγραμματιστικά Υποδείγματα

Ως «**Προγραμματιστικό Υπόδειγμα**» εννοείται ένα πρότυπο ανάπτυξης προγραμμάτων, δηλαδή μία καθορισμένη μεθοδολογία με βάση την οποία αναπτύσσονται η δομή και τα στοιχεία του προγράμματος. Τα κυριότερα προγραμματιστικά υποδείγματα είναι:

- Ο **προστακτικός προγραμματισμός** όπου τα προγράμματα αναπτύσσονται με απλές εντολές σε προστακτική (Διάβασε, Εμφάνισε, Επανάλαβε). Π.χ. FORTRAN, BASIC, PASCAL, C
- Ο **δηλωτικός προγραμματισμός** όπου το πρόβλημα επιλύεται δηλώνοντας απλώς τις επιθυμητές ιδιότητες του αποτελέσματος. Το πρόγραμμα περιέχει λογικές εκφράσεις, ενώ κατά την εκτέλεσή του γίνεται έλεγχος για το ποιες ακριβώς ικανοποιούνται. Π.χ. PROLOG, SQL.
- Ο **συναρτησιακός προγραμματισμός** επιλύει το πρόβλημα με τη χρήση μαθηματικών συναρτήσεων. Π.χ. LISP.
- Ο **αντικειμενοστραφής προγραμματισμός** βασίζεται στην έννοια του αντικειμένου. Τα αντικείμενα δημιουργούνται από τις κλάσεις. Μία κλάση ορίζει τα χαρακτηριστικά και τη συμπεριφορά ενός τύπου αντικειμένου, λειτουργεί δηλαδή ως πρότυπο. Ένα αντικείμενο είναι μία δομή δεδομένων η οποία περιέχει τόσο τα δεδομένα (χαρακτηριστικά που την περιγράφουν) όσο και τις διαδικασίες (μεθόδους) που επενεργούν σε αυτά. Π.χ. C++ και η JAVA.
- Ο **λογικός προγραμματισμός** όπου τα προγράμματα είναι γραμμένα ως ένα σύνολο από προτάσεις σε μορφή λογικών εκφράσεων. Π.χ. PROLOG.

2.3.1.2 Προγραμματιστικά Υποδείγματα

- Ο **παράλληλος προγραμματισμός** στον οποίο τα προγράμματα εκμεταλλεύονται την ύπαρξη υπολογιστών που διαθέτουν περισσότερους από έναν επεξεργαστές. Έτσι επιτυγχάνεται η μείωση του χρόνου εκτέλεσης της εφαρμογής. Θα πρέπει όμως το πρόβλημα προς επίλυση να διαιρεθεί σε τμήματα που μπορούν να επιλυθούν παράλληλα. Π.χ. Occam.



2.3.1.3 Δομημένος Προγραμματισμός

Δομημένος Προγραμματισμός: μεθοδολογία ανάλυσης, σχεδίασης και συγγραφής προγραμμάτων. Χρησιμοποιεί:

- την **ιεραρχική σχεδίαση**: ανάλυση «από πάνω προς τα κάτω» χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα
- τον **τμηματικό προγραμματισμό**: υλοποιεί την ιεραρχική σχεδίαση όπου κάθε υποπρόβλημα αποτελεί ανεξάρτητη ενότητα που ονομάζεται υποπρόγραμμα
- τη δομή ακολουθίας, τη δομή επιλογής και τη δομή επανάληψης ή συνδυασμό τους που καθιστούν άσκοπη τη χρήση της εντολής GOTO (Πήγαινε).

Υποπρόγραμμα: τμήμα προγράμματος που επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα.

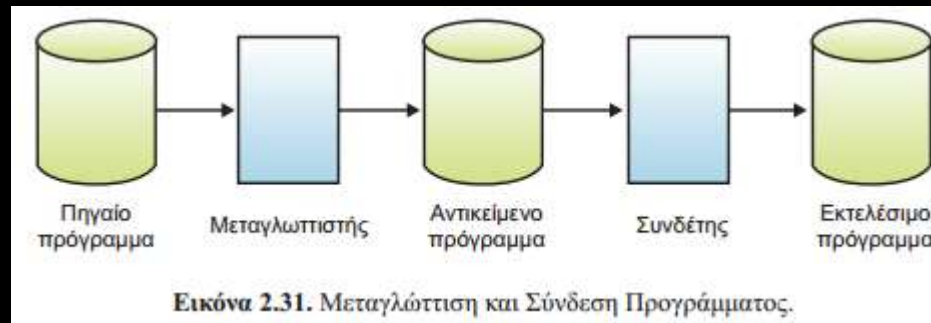
Εντολή **GOTO**: έχει ως αποτέλεσμα την αλλαγή της ροής του προγράμματος, δηλαδή της διακλάδωσης σε μία άλλη εντολή του προγράμματος εκτός από την επόμενη

2.3.1.3 Δομημένος Προγραμματισμός

Παράδειγμα 2.32. Να γραφεί πρόγραμμα το οποίο θα διαβάζει αριθμούς και θα υπολογίζει το μέσο όρο τους. Η διαδικασία εισαγωγής αριθμών θα σταματά αν διαβαστεί ο αριθμός 0, χωρίς να συνυπολογίζεται. Αν ο πρώτος αριθμός που θα διαβαστεί είναι το μηδέν, θα εκτυπώνει το μήνυμα «Δεν δόθηκε άλλο στοιχείο εκτός του μηδενός»

```
1 Αλγόριθμος Αριθμοί
2 πλ ← 0
3 Σ ← 0
4 Εμφάνισε "Πληκτρολογήστε έναν αριθμό"
5 Διάβασε α
6 Όσο α ≠ 0 επανάλαβε
7     Σ ← Σ + α
8     πλ ← πλ + 1
9     Εμφάνισε "Πληκτρολογήστε έναν αριθμό"
10    Διάβασε α
11 Τέλος_επανάληψης
12 Αν πλ ≠ 0 τότε
13     ΜΟ ← Σ / πλ
14     Εμφάνισε "Ο μέσος όρος είναι:", ΜΟ
15 αλλιώς
16     Εμφάνισε "Δε δόθηκε άλλο στοιχείο εκτός του μηδενός"
17 Τέλος_αν
18 Τέλος Αριθμοί
```


2.3.2 Σχεδίαση και συγγραφή κώδικα



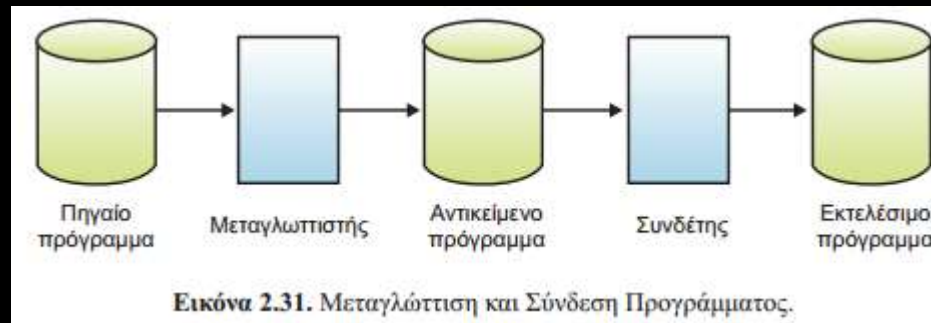
Πηγαίο πρόγραμμα (source program): Το πρόγραμμα που γράφεται σε κάποια γλώσσα προγραμματισμού

Συντάκτης (editor): ειδικό πρόγραμμα για τη σύνταξη του πηγαίου προγράμματος

Μεταφραστικά προγράμματα:

- **Μεταγλωττιστές** (compilers): δέχονται στην είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. (-): υποχρεωτική μεταγλώττιση και σύνδεση πριν την εκτέλεση (+): ταχύτερη εκτέλεση
- **Διερμηνευτές** (interpreters): διαβάζουν μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μία εκτελεί αμέσως μία ισοδύναμη ακολουθία εντολών μηχανής. (+): άμεση εκτέλεση και διόρθωση (-) : πιο αργή εκτέλεση

2.3.2 Σχεδίαση και συγγραφή κώδικα



Είδη λαθών:

- **Συντακτικά:** εμφανίζονται κατά τη μετάφραση. Οφείλονται σε αναγραμματισμούς, σε λανθασμένη σύνταξη, παράλειψη δήλωσης μεταβλητών κ.ά. Ο μεταφραστής ανιχνεύει τα λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα. Ακολουθεί η διόρθωσή τους από τον προγραμματιστή.
- **Λογικά:** δεν ανιχνεύονται από τα μεταφραστικά προγράμματα (π.χ. διαίρεση με το μηδέν). Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν εργαλεία εκσφαλμάτωσης για την εκτέλεση του προγράμματος εντολή προς εντολή με παρακολούθηση των τιμών των μεταβλητών

Αντικείμενο (object) πρόγραμμα: το πρόγραμμα σε γλώσσα μηχανής που παράγεται από τους μεταγλωττιστές

Συνδέτης– φορτωτής (linker - loader): συνδέει το αντικείμενο πρόγραμμα με τις βιβλιοθήκες (libraries: συλλογή από έτοιμα υποπρογράμματα που μπορούν να χρησιμοποιούνται κατά τη συγγραφή νέων προγραμμάτων)

Εκτελέσιμο (executable) πρόγραμμα: το αποτέλεσμα του συνδέτη

2.3.2 Σχεδίαση και συγγραφή κώδικα

Χρονική Διάρκεια διαφήμισης (δευτερόλεπτα)	Κόστος (€ / δευτερόλεπτο)
Μέχρι 15	90
Μέχρι 30	75
Πάνω από 30	50

Παράδειγμα 2.40. Σε ένα ραδιοφωνικό σταθμό το κόστος ενός διαφημιστικού μηνύματος σε σχέση με τα δευτερόλεπτα μετάδοσης, υπολογίζεται κλιμακωτά σύμφωνα με τον παραπάνω πίνακα.

A. Να αναπτύξετε πρόγραμμα το οποίο:

- i. να διαβάζει το πλήθος των διαφορετικών μηνυμάτων που πρόκειται να μεταδώσει ο σταθμός την επόμενη εβδομάδα
- ii. να διαβάζει τη χρονική διάρκεια κάθε μηνύματος και να υπολογίζει καλώντας κατάλληλο υποπρόγραμμα το κόστος του
- iii. να εμφανίζει με κατάλληλο μήνυμα τα συνολικά έσοδα του σταθμού καθώς και το ποσοστό (%) των μηνυμάτων με χρονική διάρκεια άνω των 30 δευτερολέπτων.

B. Να γράψετε τον αλγόριθμο για τον υπολογισμό του κόστους του κάθε μηνύματος.

2.3.2 Σχεδίαση και συγγραφή κώδικα

```
1 Αλγόριθμος Ραδιόφωνο
2  $\Sigma \leftarrow 0$ 
3  $\pi_{30} \leftarrow 0$ 
4 Εμφάνισε "Πληκτρολογήστε το πλήθος των μηνυμάτων"
5 Διάβασε n
6 Για i από 1 μέχρι n
7     Εμφάνισε "Πληκτρολογήστε τη διάρκεια του", i, "ου μηνύματος"
8     Διάβασε χδ
9     Κάλεσε Κόστος(χδ, κμ)
10     $\Sigma \leftarrow \Sigma + κμ$ 
11    Αν  $χδ > 30$  τότε
12         $\pi_{30} \leftarrow \pi_{30} + 1$ 
13    Τέλος_αν
14 Τέλος_επανάληψης
15 Εμφάνισε "Τα συνολικά έσοδα του σταθμού είναι",  $\Sigma$ 
16 ποσοστό  $\leftarrow \pi_{30} / n * 100$ 
17 Εμφάνισε "Το ποσοστό είναι", ποσοστό, "%"
18 Τέλος Ραδιόφωνο
```

```
20 Αλγόριθμος Κόστος
21 Δεδομένα // δ //
22 Αν  $\delta \leq 15$  τότε
23      $\kappa \leftarrow \delta * 90$ 
24 αλλιώς_αν  $\delta \leq 30$  τότε
25      $\kappa \leftarrow 15 * 90 + (\delta - 15) * 75$ 
26 αλλιώς
27      $\kappa \leftarrow 15 * 90 + 15 * 75 + (\delta - 30) * 50$ 
28 Τέλος_αν
29 Αποτελέσματα // κ//
30 Τέλος Κόστος
```