

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό
Περιβάλλον
κεφ.6 Εισαγωγή στον Προγραμματισμό


Μάριος Αραποστάθης
Καθηγητής πληροφορικής Βαρβάκειου Λύκειου
<http://users.sch.gr/mariosarapostathis>




6.1 Η έννοια του προγράμματος

Τα στάδια αντιμετώπισης ενός προβλήματος είναι:
Κατανόηση – Ανάλυση – Επίλυση

Η Επίλυση περιλαμβάνει 3 στάδια:

- 
- Τον ακριβή προσδιορισμό του προβλήματος
 - Προσδιορισμός δεδομένων, ζητούμενων, συνθηκών κτλ
 - Την ανάπτυξη του αντίστοιχου αλγορίθμου
 - φυσική γλώσσα, διαγράμματα ροής, ελεύθερο κείμενο
 - Τη κωδικοποίηση του αλγορίθμου
 - Σε μια γλώσσα προγραμματισμού



Πρόγραμμα = σύνολο εντολών που πρέπει να δοθούν στον Η/Υ για την υλοποίηση του αλγορίθμου επίλυσης ενός προβλήματος. Περιλαμβάνει τον κώδικα, τα δεδομένα και τις δομές δεδομένων που ενεργεί.



Προγραμματισμός = η διαδικασία ανάπτυξης προγραμμάτων σε μια γλώσσα προγραμματισμού

6.2 Ιστορική αναδρομή

- οι πρώτοι ηλεκτρονικοί υπολογιστές δημιουργήθηκαν στις αρχές του 40' (ENIAC, Mark I)
- την έννοια του ηλεκτρονικού υπολογιστή με αποθηκευμένο πρόγραμμα ανέπτυξε το 1945 ο Τζον φον Νόιμαν
- ο ίδιος διατύπωσε την αρχιτεκτονική των σύγχρονων Η/Υ (ΚΜΕ, μνήμη, μονάδες Ε/Ε) και θεωρείται ο πατέρας της πληροφορικής
- πολλές γλώσσες προγράμμου συνδέονται με την αρχιτεκτονική του Η/Υ και αν και εξελίσσονται συνεχώς η βασικές αρχές τους παραμένουν ίδια



6.2 Είδη γλωσσών προγράμμου

- Γλώσσες Μηχανής
- Συμβολικές γλώσσες
- Γλώσσες υψηλού επιπέδου
- Γλώσσες 4^{ης} γενιάς



```
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

C/C++

6.2.1 Γλώσσες Μηχανής

- ο Η/Υ αντιλαμβάνεται τα δεδομένα ως ακολουθίες bit (0/1) και όχι ως λέξεις, προτάσεις και αριθμούς που αντιλαμβάνεται ο άνθρωπος
- οι εντολές του προγραμματιστή μετατρέπονται σε γλώσσα μηχανής (ακολουθίες bit)
- οι ακολουθίες αυτές εκτελούνται κατευθείαν από τον υπολογιστή (μηχανή)



6.2.2 Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

- επειδή δεν ήταν πρακτικό να προγραμματίζουμε κατευθείαν σε γλώσσα μηχανής επινοήθηκαν οι γλώσσες χαμηλού επιπέδου (assembly)
- το λεξιλόγιο τους αποτελείται από απλές λέξεις π.χ. ADD – πρόσθεση, LDA – φόρτωση στη μνήμη κτλ που είναι μνημονικές
- ο συμβολομεταφραστής (assembler) ειδικό πρόγραμμα μετατρέπει τις εντολές σε γλώσσα μηχανής για να εκτελεστεί από τον Η/Υ
- η assembly εξαρτάται από την αρχιτεκτονική του Η/Υ (-)
- συνεπώς τα προγράμματα δεν μπορούν να μεταφερθούν (-)
- ο προγρ/της πρέπει να γνωρίζει το μέγεθος της RAM, της μνήμης flash, των καταχωρητών (-)
- δεν διαθέτουν εντολές σύνθετων λειτουργιών και συνεπώς τα προγράμματα είναι μακροσκελή (-)



6.2.3 Γλώσσες υψηλού επιπέδου (1/3)

- τα μειονεκτήματα των συμβολικών γλωσσών οδήγησαν στην ανάπτυξη των γλωσσών προγραμματισμού υψηλού επιπέδου

Όνομασία	έτος	ερμηνεία	κατεύθυνση
Fortran	1957	Formula Translation - Μετάφραση τύπων	επίλυση μαθ/κων προβλημάτων (επιστημονική)
Cobol	1960	Common Business Oriented Language – επιχειρήσεις	εμπορική
Algol	1958	Algorithmic Language – αλγοριθμική γλώσσα	γενικής χρήσης
PL/I	1964	Programming Language I	γενικής χρήσης
Lisp	1958	List Processor – επεξεργαστής λίστας	τεχνητής νοημοσύνης
Prolog	1972	Programming Logic – λογικός προγράμμος	τεχνητής νοημοσύνης

6.2.3 Γλώσσες υψηλού επιπέδου (2/3)

Όνομασία	έτος	ερμηνεία	κατεύθυνση
Basic	1964	Beginner's all Purpose Symbolic Instruction Code – για αρχάριους	γενικής χρήσης
Pascal	1970	Μαθηματικές και εμπορικές εφαρμογές με δόμηση	γενικής χρήσης
C / C++	1969	Για επικοινωνία με μηχανή αλλά και υψηλού επιπέδου εφαρμογές	γενικής χρήσης
Java	1995	αντικειμενοστραφής	εφαρμογές στο διαδίκτυο
Visual Basic	1991	Εφαρμογές σε γραφικά περιβάλλοντα - αντικειμενοστραφής	γενικής χρήσης
Visual C++	1993	Εφαρμογές σε γραφικά περιβάλλοντα - αντικειμενοστραφής	γενικής χρήσης

6.2.3 Γλώσσες υψηλού επιπέδου (3/3)

3ης γενιάς

- Οι γλώσσες αυτές μεταφράζονται σε γλώσσα μηχανής με τη χρήση μεταγλωττιστών (compilers)
- Με την ανάπτυξη των γραφικών περιβαλλόντων διακρίνουμε τα εξής είδη προγραμματισμού:

- αντικειμενοστραφής (C++, Java)
- οπτικός (VB)
- οδηγούμενος από γεγονός

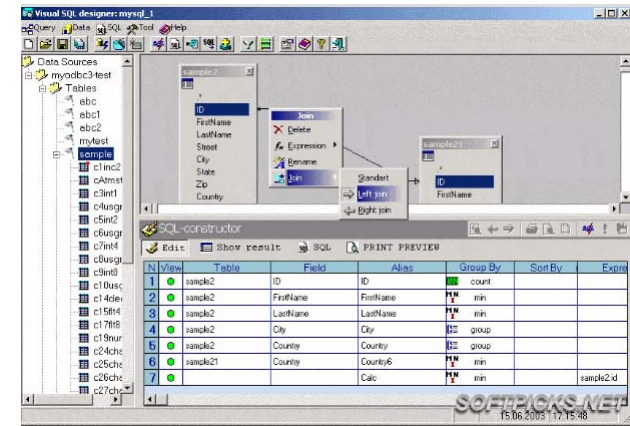
Πλεονεκτήματα γλωσσών υψηλού επιπέδου

- α) πιο φυσικός τρόπος έκφρασης για τον χρήστη
- β) ανεξαρτησία από το υλικό – φορητότητα
- γ) ευκολία εκμάθησης
- δ) ευκολότερη διόρθωση λαθών και συντήρηση



6.2.4 Γλώσσες 4^{ης} γενιάς

- οι γλώσσες 3^{ης} γενιάς απευθύνονται σε προγραμματιστές με εξειδικευμένες γνώσεις
- Οι γλώσσες 4^{ης} γενιάς είναι εφοδιασμένες με εργαλεία προγραμματισμού
- αποκρύπτουν λεπτομέρειες της αρχιτεκτονικής του υλικού από τον χρήστη
- Συνήθως ο χρήστης υποβάλει ερωτήσεις με τη μορφή κώδικα και παίρνει απαντήσεις από το σύστημα
- Τέτοιο είδος γλώσσας είναι η SQL για το χειρισμό μικρών Βάσεων Δεδομένων (ΒΔ)



MySQL




Ταξινόμηση γλωσσών προγραμματισμού

■ ανάλογα με τον τύπο

- αλγοριθμικές (διαδικασιακές) (Pascal, C, Cobol)
- αντικειμενοστραφείς (C++, VB)
- συναρτησιακές (LISP)
- μη διαδικασιακές (Prolog)
- ερωτοαπαντήσεων (SQL)

■ ανάλογα με την περιοχή χρήσης

- γλώσσες γενικής χρήσης
 - επιστημονικής κατεύθυνσης (Fortran)
 - εμπορικής κατεύθυνσης (Cobol)
 - προγραμματισμού συστημάτων (C)
 - τεχνητής νοημοσύνης (Lisp, Prolog)
 - ειδικής χρήσης (SQL)
- 

6.3 Φυσικές και τεχνητές γλώσσες

Στοιχεία μιας γλώσσας:

- αλφάβητο = το σύνολο των στοιχείων που χρησιμοποιεί: α..ω, Α..Ω, α..z, Α..Z, 0..9, “”, +, () κτλ
- λεξιλόγιο = όλες οι αποδεκτές λέξεις από τη γλώσσα, δεσμευμένες και μη
- γραμματική = αποτελείται από :
 - Τυπικό = καθορίζει ποιες λέξεις είναι αποδεκτές π.χ. διάβασε και όχι διαβασε, χ3 και όχι 3χ
 - Συντακτικό = καθορίζει τον τρόπο σύνδεσης λέξεων για τη δημιουργία προτάσεων πχ αρχή_επανάληψης
- σημασιολογία = καθορίζει το νόημα των λέξεων
- Διαφορές φυσικών και τεχνητών γλωσσών
 - Οι φυσικές γλώσσες εξελίσσονται με νέες λέξεις
 - Οι τεχνητές είναι στάσιμες



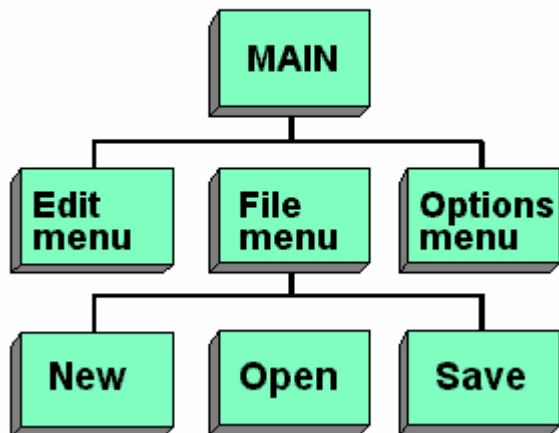
6.4 Τεχνικές σχεδίασης προγραμμάτων (1/3)

- Ιεραρχική σχεδίαση προγράμματος
 - αρχικά καθορίζονται οι λειτουργίες του προγράμματος σε ανώτερο επίπεδο
 - στη συνέχεια διασπώνται σε μικρότερες λειτουργίες σε κατώτερο επίπεδο
 - γίνεται χρήση διαγραμματικών τεχνικών
 - η δομή είναι δεντρική

Τμηματικός προγραμματισμός

- Υλοποιεί την ιεραρχική σχεδίαση προγράμματος
- Το κάθε υπό-πρόβλημα αποτελεί ανεξάρτητη ενότητα
- Πλεονεκτήματα του είναι η εύκολη δημιουργία, παρακολούθηση, κατανόηση και συντήρηση του προγράμματος και η ελάττωση των λαθών.

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.



6.4 Τεχνικές σχεδίασης προγραμμάτων (2/3)

■ Δομημένος προγραμματισμός

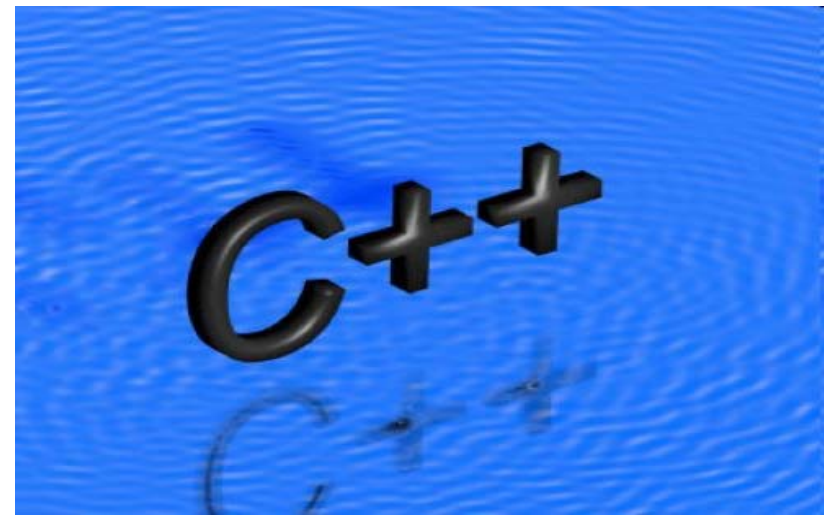
- αποτελεί την κυρίαρχη μεθοδολογία σύνταξης και γραφής προγραμμάτων σε υπολογιστή
- θεωρεί ότι όλοι οι αλγόριθμοι υλοποιούνται από συνδυασμούς 3 θεμελιωδών δομών:
 - ακολουθίας
 - επιλογής
 - επανάληψης
- κάθε πρόγραμμα ή υποπρόγραμμα έχει είσοδο και έξοδο
- αποφεύγει τη χρήση της εντολής GOTO
- επινοήθηκε από τον ιδρυτή της Pascal



6.4 Τεχνικές σχεδίασης προγραμμάτων (3/3)

Πλεονεκτήματα του δομημένου προγράμμου:

- δημιουργία πιο απλών προγραμμάτων
- άμεση μεταφορά αλγορίθμων σε προγράμματα
- διευκόλυνση ανάλυση προγράμματος σε τμήματα
- περιορισμός λαθών κατά την ανάπτυξή τους
- πιο εύκολη ανάγνωση και κατανόηση τους
- πιο εύκολη διάρθρωση και συντήρηση

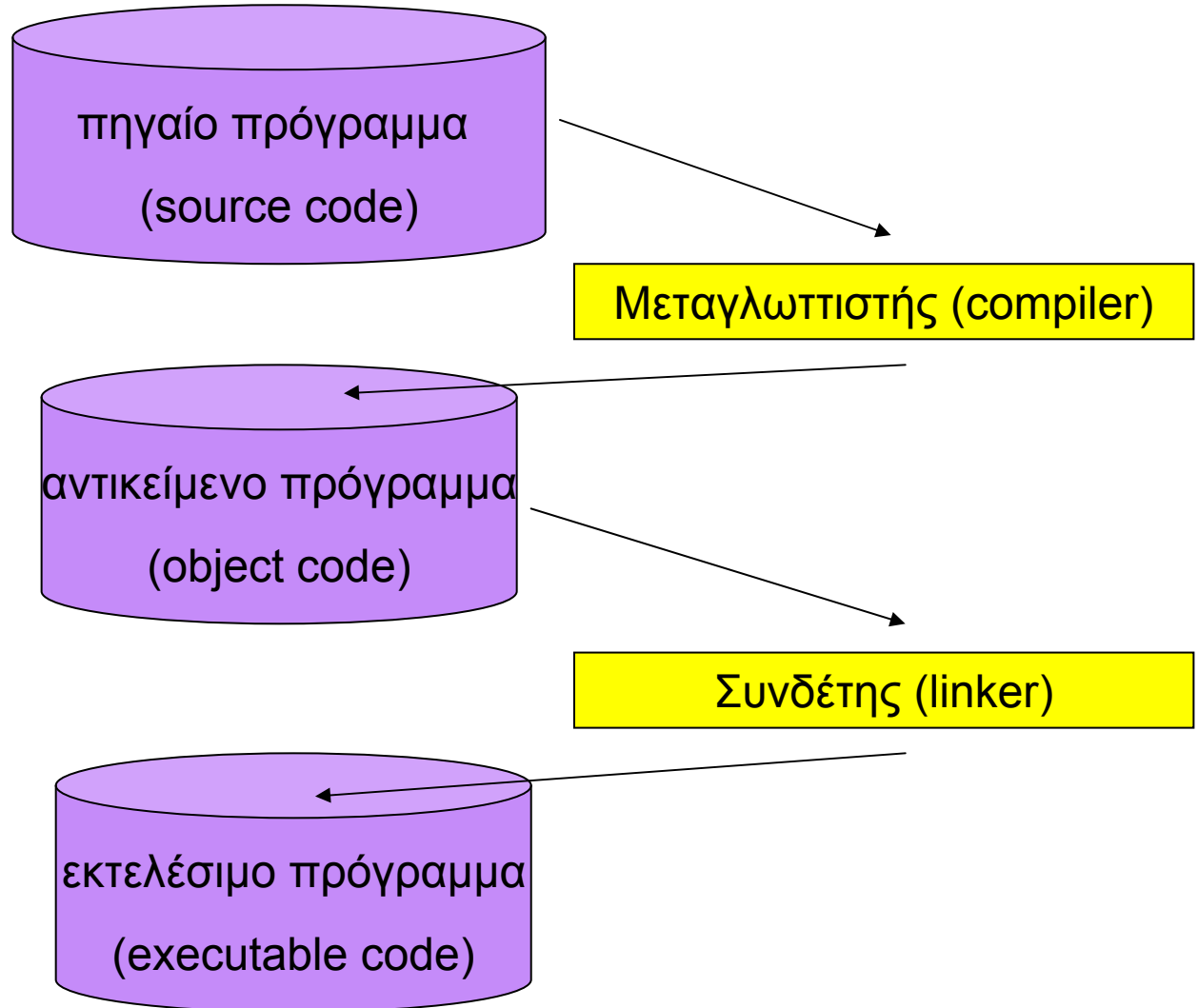


6.7 Προγραμματιστικά περιβάλλοντα

- τα προγράμματα σε οποιαδήποτε γλώσσα πρέπει να μετατραπούν σε εντολές γλώσσας μηχανής (exe)
- οι Η/Υ μόνο εκτελέσιμα αρχεία «καταλαβαίνουν»
- Υπάρχουν 2 κατηγορίες μεταφραστικών προγραμμάτων που κάνουν την μετατροπή:
 - οι μεταγλωττιστές (compilers) που δέχεται σαν είσοδο τον κώδικα του χρήστη και παράγει ισοδύναμο σε γλώσσα μηχανής
 - οι διερμηνευτές (interpreters)
- το αρχικό πρόγραμμα του χρήστη ονομάζεται πηγαίο
- το πρόγραμμα που παράγεται από τον compiler ονομάζεται αντικείμενο
- το αντικείμενο δεν είναι απαραίτητα εκτελέσιμο αλλά γίνεται με τη σύνδεσή του με τις βιβλιοθήκες της γλώσσας
- Τη λειτουργία αυτή κάνει ο συνδέτης (linker)



6.7 Προγραμματιστικά περιβάλλοντα



6.7 Προγραμματιστικά περιβάλλοντα

- για να δημιουργηθεί ο εκτελέσιμος κώδικας το πρόγραμμα δεν πρέπει να περιέχει λάθη. Υπάρχουν 2 κατηγορίες σφαλμάτων:
 - Συντακτικά – αναγραμματισμοί, παράλειψη ειδικών χαρακτήρων ,” ”, παράλειψη τερματισμού δομής, εντοπίζονται στο στάδιο της μεταγλώττισης
 - Λογικά – εσφαλμένη κωδικοποίηση του αλγορίθμου, εντοπίζονται κατά την εκτέλεση του αλγορίθμου, πιο δύσκολος εντοπισμός
- όταν εντοπίζονται συντακτικά λάθη ο μεταγλωττιστής εμφανίζει κατάλληλα διαγνωστικά μηνύματα
- τα λάθη διορθώνονται και η διαδικασία μεταγλώττισης επαναλαμβάνεται
- το πηγαίο πρόγραμμα γράφεται μέσω ενός συντάκτη
- τα σύγχρονα περιβάλλοντα διαθέτουν συντάκτη, διερμηνευτή και συνδέτη με ενιαίο τρόπο

