

ΚΕΦΑΛΑΙΟ 6 - ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Προγραμματισμός Η/Υ

- Ο προγραμματισμός είναι η διατύπωση του αλγορίθμου σε μορφή κατανοητή από τον Η/Υ ώστε να τον εκτελέσει («τρέξει» όπως λέμε στην ορολογία της Πληροφορικής).
- Η διατύπωση γίνεται χρησιμοποιώντας μία γλώσσα προγραμματισμού

Κατηγορίες Γλωσσών Προγραμματισμού

- Γλώσσα μηχανής
- Γλώσσες χαμηλού επιπέδου ή Συμβολικές γλώσσες
- Γλώσσες υψηλού επιπέδου (ή Τρίτης γενιάς)

1. **Γλώσσες Μηχανής** : Το πρόγραμμα περιέχει εντολές που είναι σε δυαδική μορφή, άμεσα κατανοητή από τον Η/Υ (όχι όμως από τον άνθρωπο). Δηλαδή, το πρόγραμμα αποτελείται από ακολουθίες 0 και 1 π.χ.

10101000 00001010

11000000 00000001

Πλεονεκτήματα :

- Ταχύτατη εκτέλεση των εντολών.
- Δεν απαιτείται μεταφραστικό πρόγραμμα.

Μειονεκτήματα :

- Το γράψιμο του προγράμματος είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία.
- Απαιτείται βαθιά γνώση της αρχιτεκτονικής του Η/Υ.
- Το πρόγραμμα «τρέχει» μόνο στο συγκεκριμένο τύπο του Η/Υ.

Γλώσσες χαμηλού Επιπέδου ή συμβολικές (Assembly). Οι εντολές που είναι σε μορφή 0 και 1 αντικαθίστανται από μνημονικά (συμβολικά) ονόματα. Για παράδειγμα, η εντολή 100001100 αντικαθίστανται από το ADD.

Πλεονεκτήματα :

- Ταχύτατη εκτέλεση των εντολών.
- Η μορφή του προγράμματος είναι καλύτερα κατανοητή από τον άνθρωπο σε σχέση με τη γλώσσα μηχανής.

Μειονεκτήματα :

- Η αντιστοιχία 1 προς 1 με τις εντολές της γλώσσας μηχανής παρέμενε.

- Απαιτείται η χρήση ενός μεταφραστικού προγράμματος ώστε οι συμβολικές εντολές να μετατραπούν στις αντίστοιχες δυαδικές. Το ειδικό αυτό πρόγραμμα ονομάζεται συμβολομεταφραστής (assembler).
- Το γράψιμο του προγράμματος εξακολουθεί να είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία.
- Απαιτείται βαθιά γνώση της αρχιτεκτονικής του Η/Υ.
- Το πρόγραμμα «τρέχει» μόνο στο συγκεκριμένο τύπο του Η/Υ.

2. **Γλώσσες Υψηλού Επιπέδου:** Γλώσσες που οι εντολές προσεγγίζουν κατά πολύ την κατανοητή στον άνθρωπο γλώσσα. Έχουν τα εξής πλεονεκτήματα:

- Έχουν πιο φυσικό και «ανθρώπινο» τρόπο έκφρασης

Πλεονεκτήματα :

- Η ανεξαρτησία από τον τύπο του Επεξεργαστή (Υπολογιστή) που εκτελούνται. Μπορούν να εκτελεστούν με ελάχιστες ή καθόλου μετατροπές σε οποιονδήποτε Υπολογιστή αρκεί να έχει εγκατασταθεί στον Υπολογιστή το κατάλληλο γι' αυτόν τον Υπολογιστή μεταφραστικό πρόγραμμα που μετατρέπει το πρόγραμμα σε Γλώσσα μηχανής. Έχουν δηλαδή το πλεονέκτημα της μεταφερσιμότητας.
- Η ευκολία εκμάθησης και εκπαίδευσης στις γλώσσες αυτές
- Η διόρθωση λαθών και η συντήρηση του προγράμματος είναι πολύ πιο εύκολη.

Μειονεκτήματα :

- Απαιτείται η χρήση ενός μεταφραστικού προγράμματος ώστε οι εντολές να μετατραπούν σε πολλές δυαδικές εντολές (δεν έχουμε εδώ αντιστοιχία 1 προς 1). Έχουμε δύο ειδών μεταφραστικά προγράμματα: τους μεταγλωττιστές (compilers) και τους διερμηνείς (interpreters).
- Το πρόγραμμα «τρέχει» πιο αργά σε σχέση με τα προγράμματα των συμβολικών γλωσσών ή της γλώσσας μηχανής.

Κατηγορίες γλωσσών υψηλού επιπέδου βάσει του τρόπου δόμησης των προγραμμάτων

Διαδικασιακές ή αλγοριθμικές γλώσσες (Procedural) : λέγονται έτσι διότι επιτρέπουν την εύκολη υλοποίηση αλγορίθμων π.χ. Pascal, Basic.

Αντικειμενοστραφείς γλώσσες (object – oriented) π.χ. C++

Συναρτησιακές γλώσσες (functional) π.χ. LISP

Μη-διαδικασιακές γλώσσες : χαρακτηρίζονται και γλώσσες πολύ υψηλού επιπέδου.π.χ. PROLOG

Γλώσσες ερωταπαντήσεων (Query languages) ή 4ης γενιάς π.χ. SQL

Κατηγορίες γλωσσών υψηλού επιπέδου βάσει περιοχής χρήσης

Γλώσσες γενικής χρήσης : Σκοπός τους είναι να επιλύουν πάσης φύσεως προβλήματα (αριθμητικά, εμπορικά, επιστημονικά). Τέτοιες είναι η Basic, Pascal. Μερικές γλώσσες, όμως, έχουν δημιουργηθεί αποκλειστικά για να επιλύουν ευκολότερα συγκεκριμένους τύπους προβλημάτων όπως :

Γλώσσες επιστημονικής κατεύθυνσης : π.χ. FORTRAN

Γλώσσες εμπορικής κατεύθυνσης : π.χ. COBOL.

Γλώσσες προγραμματισμού συστημάτων π.χ. C

Γλώσσες τεχνητής νοημοσύνης π.χ. LISP, PROLOG.

Γλώσσες ειδικής χρήσης. Σκοπός τους είναι να επιλύουν ειδικού τύπου προβλήματα όπως διαχείριση Βάσεων Δεδομένων κ.α. π.χ. SQL

Ονομαστική περιγραφή γλωσσών υψηλού επιπέδου

1. FORTRAN (Επιστημονικές και μαθηματικές εφαρμογές). Πρόκειται για μία παλιά γλώσσα
2. COBOL (Εμπορικές Εφαρμογές π.χ. διαχείριση αποθήκης)
3. ALGOL (Γενικής χρήσης). Μια από τις πρώτες γλώσσες που δεν πολυχρησιμοποιήθηκε αλλά επηρέασε την εξέλιξη των άλλων γλωσσών.
4. PL /1 (Γενικής χρήσης)
5. LISP (Χειρισμός Λιστών Προβλήματα Τεχνητής Νοημοσύνης π.χ. Ρομποτική)) ανήκει στις λεγόμενες Συναρτησιακές Γλώσσες
6. PROLOG (Προβλήματα Τεχνητής Νοημοσύνης π.χ. Ρομποτική) ανήκει στις λεγόμενες μη διαδικασιακές γλώσσες
7. PASCAL (Γενικής χρήσης)
8. BASIC (Γενικής χρήσης)
9. C++ (Σι Πλας Πλας) (Γλώσσα αντικειμενοστρεφής)
10. JAVA (Γλώσσα εφαρμογών Internet)
11. Visual Basic (Γλώσσα Οπτικού Προγραμματισμού)
12. Visual C++ (Γλώσσα Οπτικού Προγραμματισμού)
13. SQL (Γλώσσα 4ης Γενιάς) χρησιμοποιείται για άντληση δεδομένων μέσα από Βάσεις δεδομένων. Ανήκει στις λεγόμενες Γλώσσες Ερωταπαντήσεων Δεν απευθύνεται μόνο σε προγραμματιστές αλλά και χρήστες. Ο χρήστης μπορεί, σχετικά εύκολα, να υποβάλει ερωτήσεις στο σύστημα ή να αναζητά πληροφορίες από μία Βάση Δεδομένων.

Παράδειγμα,

```
SELECT Επώνυμο, Όνομα  
FROM Μαθητές  
WHERE Τάξη = 'Γ2'
```

Οι περισσότερες Γλώσσες Προγραμματισμού ανήκουν στην κατηγορία των Διαδικασιακών Γλωσσών ή Αλγοριθμικών Γλωσσών (Εξαιρέση όπως φαίνεται παραπάνω είναι η PROLOG).

Η επιλογή της γλώσσας προγραμματισμού για την δημιουργία ενός προγράμματος εξαρτάται από τα εξής:

1. Το είδος της εφαρμογής (πχ. Για εμπορική εφαρμογή μπορεί να χρησιμοποιηθεί λόγω εξειδίκευσης της γλώσσας η COBOL)
2. Το Υπολογιστικό περιβάλλον που θα εκτελεστεί η εφαρμογή
3. Ποιες γλώσσες διαθέτουμε σαν προγραμματιστικό περιβάλλον
4. Τι γνώσεις έχει ο προγραμματιστής.

Αντικειμενικά ιδανική γλώσσα για την ανάπτυξη μιας εφαρμογής δεν υπάρχει.

Φυσικές- Τεχνητές Γλώσσες

- Φυσικές είναι οι γλώσσες που μιλάμε.
- Τεχνητές είναι οι γλώσσες προγραμματισμού που αναπτύχθηκαν για να μπορεί ο προγραμματιστής να δίνει εντολές που μπορεί να εκτελέσει ο προγραμματιστής.

Χαρακτηριστικά που προσδιορίζουν μια Γλώσσα (Φυσική ή Τεχνητή)

1. Αλφάβητο (Γράμματα αλφαβήτου πεζά ή κεφαλαία, αριθμητικά ψηφία, σημεία στίξης)
2. Λεξιλόγιο (το σύνολο των λέξεων)
3. Γραμματική
 - a. Τυπικό (το σύνολο των κανόνων που καθορίζει τις μορφές με τις οποίες μια λέξη μπορεί να εμφανίζεται π.χ. γλώσσα, γλώσσας κλπ.)
 - b. Συντακτικό (το σύνολο των κανόνων που καθορίζει τον τρόπο με τον οποίο οι λέξεις συνδέονται για να σχηματίσουν προτάσεις)
4. Σημασιολογία (Το σύνολο των κανόνων που καθορίζουν το νόημα των λέξεων και κατ' επέκταση των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μία γλώσσα).

Διαφορές Φυσικών και Τεχνητών Γλωσσών

Η βασική διαφορά είναι ότι οι Φυσικές Γλώσσες εξελίσσονται συνεχώς ανάλογα με τις εποχές τον κοινωνικό περίγυρο κλπ.

Και οι τεχνητές βέβαια γλώσσες κατά καιρούς βελτιώνονται από τους κατασκευαστές τους.

Τεχνικές Σχεδίασης Προγραμμάτων

- **Ιεραρχική Σχεδίαση Προγραμμάτων**

Σήμερα χρησιμοποιείται η Τεχνική της Ιεραρχικής Σχεδίασης Προγραμμάτων (ή από πάνω προς τα κάτω Τεχνική ή Top Down Technique).

Ουσιαστικά περιλαμβάνει τον καθορισμό των βασικών λειτουργιών ενός προγράμματος και στη συνέχεια τη διάσπαση των λειτουργιών αυτών σε απλούστερες, μέχρι το τελευταίο επίπεδο που οι λειτουργίες είναι απλές και επιλύονται εύκολα.

Απαιτεί τη διάσπαση του προβλήματος σε επιμέρους προβλήματα όπως έχει αναφερθεί και στο 1ο κεφάλαιο.

Τμηματικός Προγραμματισμός

Η Ιεραρχική Σχεδίαση Υλοποιείται με τον Τμηματικό Προγραμματισμό. Περιλαμβάνει την διάσπαση του προβλήματος σε επιμέρους, την επίλυσή τους και τελικά τη σύνθεση της λύσης με σκοπό την επίλυση του συνολικού προβλήματος.

Δομημένος Προγραμματισμός

Η Ιεραρχική Σχεδίαση και Τμηματικός Προγραμματισμός συνιστούν τον λεγόμενο Δομημένο Προγραμματισμό.

Ο Δομημένος Προγραμματισμός έχει σαν κύριο χαρακτηριστικό την εξάλειψη ή την μείωση της χρήσης της εντολής GOTO η οποία μεταφέρει την εκτέλεση του προγράμματος σε άλλο σημείο.

(Η εντολή GOTO θεωρείται σήμερα το μαύρο πρόβατο του προγραμματισμού)

Γενικά ο Δομημένος προγραμματισμός στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών, τη δομή ακολουθίας, τη δομή της επιλογής και της επανάληψης. Όλα τα προγράμματα μπορούν να γραφτούν χρησιμοποιώντας τις τρεις αυτές δομές. Κάθε πρόγραμμα όπως και κάθε ενότητα προγράμματος έχει μόνο μία είσοδο και μόνο μία έξοδο.

Πλεονεκτήματα:

- 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 την κατανόηση του προγράμματος από τρίτους
- Ευκολότερη διόρθωση και συντήρηση

Οπτικός και καθοδηγούμενος από το γεγονός Προγραμματισμός

Με την ευρεία διάδοση των γραφικών περιβαλλόντων επικοινωνίας (π.χ. Windows, MacOS κλπ) δημιουργήθηκαν παραλλαγές κάποιων γλωσσών που απευθύνονται σε αυτά. Τέτοιες είναι η Visual Basic, Visual C++, Delphi (Visual Pascal), C# κ.α.

Αυτές οι γλώσσες ακολουθούν τη φιλοσοφία του οπτικού και του καθοδηγούμενου-από-γεγονότα προγραμματισμού χωρίς να απορρίπτουν τις αρχές του δομημένου προγραμματισμού.

Οπτικός προγραμματισμός.

Είναι η δυνατότητα να δημιουργούμε, με γραφικό τρόπο, ολόκληρο το περιβάλλον της εφαρμογής όπως για παράδειγμα τα μενού και τα πλαίσια διαλόγου και άλλα παράθυρα της εφαρμογής.

Ο οπτικός προγραμματισμός εκμεταλλεύεται τις δυνατότητες των γραφικών περιβαλλόντων επικοινωνίας (π.χ Windows, MacOS κλπ).

Καθοδηγούμενος από γεγονότα προγραμματισμός

Είναι η δυνατότητα να εκτελούνται οι διάφορες λειτουργίες του προγράμματος με την ενεργοποίηση ενός γεγονότος. Για παράδειγμα, αν κάνουμε κλικ σε κάποια εντολή ενός μενού ή σε κάποιο κουμπί σε ένα παράθυρο της εφαρμογής τότε θα εκτελεστεί μία λειτουργία..

Προγραμματιστικά Περιβάλλοντα

Κάθε πρόγραμμα αφού σχεδιαστεί γράφεται στο περιβάλλον μιας γλώσσας προγραμματισμού. Αυτό το πρόγραμμα ονομάζεται Πηγαίο Πρόγραμμα (source) και φυσικά δεν μπορεί να εκτελεστεί αν δεν μετατραπεί σε γλώσσα μηχανής.

Το περιβάλλον της γλώσσας που γράφεται το πρόγραμμα ονομάζεται Συντάκτης (editor.)

Το Πρόγραμμα σε Γλώσσα Μηχανής που θα προκύψει μετά τη μετατροπή του Πηγαίου, ονομάζεται Αντικείμενο Πρόγραμμα (object) και είναι το πρόγραμμα το οποίο μπορεί να εκτελεστεί.

Ένα πρόγραμμα εκτός από τις εντολές έχει και στοιχεία τα οποία μία γλώσσα προγραμματισμού διαθέτει έτοιμα να τα χρησιμοποιήσει ο Προγραμματιστή (όπως για παράδειγμα οι αριθμητικές πράξεις). Αυτά υπάρχουν σ' ένα χώρο της μνήμης που ονομάζεται Βιβλιοθήκη της Γλώσσας και κάθε πρόγραμμα πριν γίνει η πλήρης μετατροπή του σε Γλώσσα Μηχανής πρέπει να συνδεθεί με τη Βιβλιοθήκη της Γλώσσας με τη Βοήθεια άλλων προγραμμάτων που ονομάζονται Συνδέτες (Linkers) και η διαδικασία Σύνδεση (Link).

Άρα έχουμε τη διαδικασία:

Αρχικό Πρόγραμμα ⇒ Μετατροπέας Πηγαίου Προγράμματος ⇒ Αντικείμενο Πρόγραμμα σε Γλώσσα Μηχανής ⇒ Συνδέτης (Linker) ⇒ Εκτελέσιμο πρόγραμμα σε Γλώσσα μηχανής.

Οι μετατροπείς των Πηγαίων Προγραμμάτων χωρίζονται στις εξής κατηγορίες:

1. Συμβολομεταφραστές

Πρόκειται για τα προγράμματα που μετατρέπουν τις Γλώσσες χαμηλού

Επιπέδου (Assembly) σε γλώσσα μηχανής)

2. Μετατροπείς Γλωσσών Υψηλού Επιπέδου
 - i. Μεταγλωττιστές (Compilers) οι οποίοι παίρνουν όλο το αρχικό πρόγραμμα και το μετατρέπουν σε αντικείμενο το οποίο στη συνέχεια εκτελείται ενιαίο. (Μοιάζει με τη διαδικασία μετάφρασης βιβλίου)
 - ii. Διερμηνευτές (Interpreters) οι οποίοι παίρνουν κάθε εντολή μία τη μετατρέπουν σε γλώσσα εντολής την εκτελούν, και μετά κάνουν ξανά τη διαδικασία για την επόμενη εντολή (Μοιάζει με τη διαδικασία που κάνει ένας διερμηνέας σ' ένα συνέδριο)

Χαρακτηριστικά Μεταγλωττιστών

1. Πιο γρήγορη εκτέλεση του προγράμματος γιατί εκτελείται ενιαίο
2. Πιο δύσκολη, επίπονη και χρονοβόρα η διαδικασία Εντοπισμού και διόρθωσης λαθών

Χαρακτηριστικά Διερμηνευτών

1. Πιο αργή εκτέλεση του προγράμματος γιατί εκτελείται μία μία εντολή
2. Πιο εύκολος ο εντοπισμός και η διόρθωση σφαλμάτων

Σήμερα πολλές γλώσσες έχουν και διερμηνευτή που χρησιμοποιείται στη διαδικασία ελέγχου του προγράμματος και διόρθωσης λαθών και αφού διορθωθούν τα λάθη χρησιμοποιείται μεταγλωττιστής για την πιο γρήγορη εκτέλεση του Προγράμματος

Παρατηρήσεις

1. Με βάση τα παραπάνω λοιπόν αν χρησιμοποιείται διερμηνέας δεν υπάρχει η έννοια του αντικείμενου προγράμματος σαν σύνολο.
2. Σε κάθε Πρόγραμμα υπάρχουν δύο ειδών λάθη που μπορούμε να βρούμε
 - a. Συντακτικά που οφείλονται σε λάθος πληκτρολόγηση εντολών, λάθος χρήση σημείων στίξης κλπ.
 - b. Λογικά ή Αλγοριθμικά που οφείλονται σε λάθος στη λογική του προγράμματος (π.χ. να δημιουργηθεί ένα ατέρμονο loop)