

Κεφάλαιο 6 (Βιβλίο Μαθητή)

- Τα στάδια επίλυσης ενός προβλήματος είναι τρία :
- - Ακριβής προσδιορισμός προβλήματος.
- - Ανάπτυξη αντίστοιχου αλγορίθμου.
- - Διατύπωση του αλγορίθμου σε μορφή κατανοητή στον υπολογιστή,
→ πρόγραμμα σε γλώσσα υψηλού επιπέδου.

Φυσικές & Τεχνητές γλώσσες

- Οι **φυσικές γλώσσες** αναπτύχθηκαν για την επικοινωνία μεταξύ των ανθρώπων.
- Οι **Γλώσσες Προγραμματισμού Υψηλού Επιπέδου (Γ.Π.Υ.Ε.)** αναπτύχθηκαν ώστε να μπορεί ο προγραμματιστής (άνθρωπος) να δίνει εντολές, εκφράζοντας τον σχετικό αλγόριθμο, με τρόπο που θα κατανοήσει ο υπολογιστής (μηχανή) για να τον εκτελέσει.
- Δηλαδή οι (Γ.Π.Υ.Ε.) χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπου και μηχανής.
- Άρα οι (Γ.Π.Υ.Ε.) είναι λοιπόν **τεχνητές γλώσσες**, που ακολουθούν όμως τις βασικές έννοιες και αρχές των φυσικών γλωσσών.

Διαφορές φυσικών & τεχνητών γλωσσών

Οι φυσικές :

- χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπων.
- εξελίσσονται συνεχώς ...αλλάζοντας

Οι τεχνητές :

- Χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπου και υπολογιστή (γενικά μηχανών)
- Χαρακτηρίζονται από στασιμότητα, γιατί φτιάχνονται για ένα συγκεκριμένο σκοπό. Όμως βελτιώνονται για να καλύψουν αδυναμίες και να ακολουθήσουν τις νέες εξελίξεις.

Μια γλώσσα προσδιορίζεται από :

- **Το Αλφάβητο** (α-ω, Α – Ω, a-z, A – Z, 0 – 9)
- **Το Λεξιλόγιο** (αποτελείται από υποσύνολα ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου και τις αποδεκτές λέξεις π.χ. μάθημα [δεκτή], θημακο [μη δεκτή]).
- **Τη Γραμματική** που αποτελείται από
 - **-το τυπικό** = σύνολο των κανόνων που ορίζει τις μορφές που μια λέξη είναι αποδεκτή
 - **-το συντακτικό** = σύνολο κανόνων που καθορίζει τη ορθότητα-νομιμότητα της διάταξης και νοητικής σύνδεσης των λέξεων της γλώσσας, για τη δημιουργία φράσεων και προτάσεων στις φυσικές γλώσσες και εντολών τεχνητές γλώσσες.
- **Τη Σημασιολογία** που είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατ' επέκταση των εκφράσεων & προτάσεων.

Τεχνικές Σχεδίασης Προγραμμάτων

- Ιεραρχική Σχεδίαση (από επάνω προς τα κάτω ...σε επίπεδα)
- Τμηματικός προγραμματισμός (Διαδικασίες - Συναρτήσεις)
- Δομημένος προγραμματισμός στηρίζεται στη χρήση 3^{ων} λογικών δομών [ακολουθίας, επιλογής, επανάληψης, καθώς και συνδυασμό τους],. **Όμως περιέχει τόσο την ιεραρχική σχεδίαση όσο και τον τμηματικό προγραμματισμό.**

Η εντολή GOTO ... μαύρο πρόβατο

- Έχει ως αποτέλεσμα την αλλαγή ροής του προγράμματος, της διακλάδωσης, σε μια άλλη εντολή του προγράμματος εκτός της επόμενης.
- Εύκολη στη χρήση, αλλά **προβληματική γιατί ήταν υπεύθυνη για τη :**
 - Δυσκολία στην αρχική σχεδίαση της λύσης
 - Παρακολούθηση ροής προγράμματος
 - Κατανόηση προγράμματος
 - Συντήρηση προγράμματος
- Ο δομημένος προγραμματισμός προήλθε από την ανάγκη περιορισμού (σχεδόν κατάργησης) της χρήσης της GOTO (πήγαινε).

Παράδειγμα....με GOTO

- Αν $A > 0$ τότε GOTO 1
- Αν $A = 0$ τότε GOTO 2
- Γράψε 'αρνητικός'
- GOTO 4
- 1 : Γράψε 'θετικός'
- GOTO 4
- 2 : Γράψε 'μηδέν'
- GOTO 4
- 4 : ... συνέχεια ...
- Αν $A > 0$ τότε
- Γράψε 'θετικός'
- Αλλιώς
- Αν $A = 0$ τότε
- Γράψε 'μηδέν'
- Αλλιώς
- Γράψε 'αρνητικός'
- Τέλος_Αν
- Τέλος_Αν
- συνέχεια ...

Πλεονεκτήματα Δομημένου προγραμματισμού

- Δημιουργία απλούστερων προγραμμάτων.
- Άμεση μεταφορά των αλγορίθμων σε προγράμματα.
- Διευκόλυνση ανάλυσης προγράμματος σε τμήματα.
- Περιορισμός λαθών κατά την ανάπτυξη του προγράμματος.
- Διευκόλυνση ανάγνωσης & κατανόησης του προγράμματος από τρίτους.
- Ευκολότερη διόρθωση και συντήρηση.

Οι γλώσσες Υψηλού επιπέδου – μοιάζουν στη φυσική μας γλώσσα και έχουν δικό τους αλφάβητο, λεξιλόγιο και συντακτικό.

(C, C++, Java, Pascal, Prolog, Python, Visual Basic κ.α.)

- Τα προγράμματα γραμμένα σε αυτές μεταφέρονται και εκτελούνται σε κάθε Η/Υ.
- Εύκολες στον εντοπισμό και τη διόρθωση λαθών.
- Ανάλογα με το είδος της εφαρμογής για ανάπτυξη (γενικής χρήσης, εμπορική, επιστημονική, παιχνίδι, τεχνητής νοημοσύνης κ.α.) επιλέγουμε την σχετικά πιο κατάλληλη γλώσσα.

Εργαλεία σε προγραμματιστικά περιβάλλοντα

- Η εργασία σύνταξης ενός προγράμματος ονομάζεται προγραμματισμός ή κωδικοποίηση. Χρειαζόμαστε :
- 1. τον **Συντάκτη** (editor) για να γράψουμε-δημιουργήσουμε το αρχικό πρόγραμμα που το ονομάζουμε **πηγαίο** (source).
- 2. τον **Μεταγλωττιστή** (Compiler) που ελέγχει όλο το πηγαίο πρόγραμμα για συντακτικά λάθη και μετά το μετατρέπει σε γλώσσα μηχανής, δημιουργώντας ένα ισοδύναμο που το λέμε **αντικείμενο** (object).
- 3. τον **Συνδέτη** (linker) που συνδέει το αντικείμενο με έτοιμα τμήματα **προγράμματος της βιβλιοθήκης της γλώσσας προγραμματισμού** ή του προγραμματιστή, ώστε να παραχθεί το τελικό πρόγραμμα που το ονομάζουμε **εκτελέσιμο** (executable).
- 4. τα **εργαλεία εντοπισμού λαθών** (debuggers) με τα οποία παρακολουθούμε (με μηνύματα) τι συμβαίνει στο παρασκήνιο κατά την εκτέλεση του προγράμματος.

Η γενική Λειτουργία στα Προγραμματιστικά περιβάλλοντα



Μεταγλωττιστής $\neq$ Διερμηνευτής

- Ο μεταγλωττιστής δέχεται ως είσοδο ένα πρόγραμμα σε γλώσσα υψηλού επιπέδου (πηγαίο – source) και παράγει ένα ισοδύναμο μεταφρασμένο σε γλώσσα μηχανής (0,1 – αντικείμενο-object), το οποίο είναι ανεξάρτητο από το αρχικό και δεν ξαναμεταφράζεται σε κάθε νέα εκτέλεση.
- Ενώ ο Διερμηνευτής διαβάζει μια προς μια τις εντολές τις εντολές του αρχικού προγράμματος και την ίδια στιγμή μεταφράζει και εκτελεί μια ισοδύναμη ακολουθία εντολών μηχανής (σε 0,1). Οπότε σε κάθε νέα εκτέλεση ξαναμεταφράζει σε γλώσσα μηχανής μια μια τις εντολές.