

6.1 Η έννοια του προγραμματισμού

Η επίλυση ενός προβλήματος με τον υπολογιστή περιλαμβάνει, όπως έχει ήδη αναφερθεί, τρία εξίσου σημαντικά στάδια.

- Τον ακριβή προσδιορισμό του προβλήματος.
- Την ανάπτυξη του αντίστοιχου αλγορίθμου.
- Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή.

Ο προγραμματισμός ασχολείται με το τρίτο αυτό στάδιο, τη δημιουργία του προγράμματος δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος.

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.

6.3 Φυσικές και τεχνητές γλώσσες

Οι γλώσσες προγραμματισμού είναι τεχνητές γλώσσες που απευθύνονται σε ανθρώπους που επιθυμούν να επικοινωνήσουν με τον Η/Υ.

Κάθε γλώσσα προγραμματισμού προσδιορίζεται από:

- ο Το αλφάβητό της
- ο Το λεξιλόγιό της
- ο Τη γραμματική της
- ο Τη σημασιολογία της (Semantics)

Αλφάβητο γλώσσας

Ως αλφάβητο ορίζουμε το σύνολο των αποδεκτών χαρακτήρων της γλώσσας. Από τους χαρακτήρες αυτούς σχηματίζονται οι λέξεις της γλώσσας.

Σε μία φυσική γλώσσα, όπως τα Ελληνικά, οι αποδεκτοί χαρακτήρες είναι τα γράμματα Α-Ω (κεφαλαία και πεζά), τα ψηφία 0-9 και τα σημεία στίξης.

Λεξιλόγιο γλώσσας

Το λεξιλόγιο μίας γλώσσας περιλαμβάνει όλες τις έγκυρες και αποδεκτές λέξεις. Στην ουσία, είναι ένα υποσύνολο από όλες τις δυνατές ακολουθίες που μπορούμε να σχηματίσουμε από τα στοιχεία του αλφαβήτου.

Σε μία φυσική γλώσσα, όπως τα Ελληνικά, η λέξη ΔΙΑΒΑΖΩ είναι αποδεκτή ενώ η λέξη ΖΩΒΑΓΩ όχι.

Η Γραμματική της γλώσσας

Η γραμματική περιλαμβάνει το **τυπολογικό** και το **συντακτικό**.

Το **τυπολογικό** ορίζει τους κανόνες σύμφωνα με του οποίους μία λέξη θα είναι αποδεκτή.

Για παράδειγμα, στην ελληνική γλώσσα για τη λέξη "δίνω", αποδεκτές μορφές είναι και το "δίνεις", "δίνουν" αλλά όχι το "δίνουτ".

Το **συντακτικό** είναι ένα σύνολο κανόνων που ορίζει το πώς πρέπει να σχηματίζονται οι προτάσεις από τις λέξεις της γλώσσας ώστε οι προτάσεις αυτές να είναι έγκυρες και αποδεκτές. Σε μία γλώσσα προγραμματισμού αυτό που ενδιαφέρει είναι η σωστή σύνταξη των εντολών.

Σημασιολογία της γλώσσας

Είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και προτάσεων της γλώσσας. Σε μία γλώσσα προγραμματισμού αυτό καθορίζεται από το δημιουργό της ενώ σε μία φυσική γλώσσα από αυτόν που εκφέρει την πρόταση.

Διαφορές μεταξύ φυσικών και τεχνητών γλωσσών

Φυσικές	Τεχνητές
Χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπων.	Χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπου και Η/Υ.
Έχουν μεγάλες δυνατότητες εξέλιξης. Νέες λέξεις μπορεί να εισαχθούν, κανόνες γραμματικής και σύνταξης να αλλάξουν κλπ.	Οι δυνατότητες εξέλιξης είναι περιορισμένες. Τις περισσότερες φορές η εξέλιξη αυτή αφορά την επέκταση του ρεπερτορίου των εντολών της γλώσσας (π.χ. Basic και Visual Basic).

6.4 Τεχνικές σχεδίασης προγραμμάτων

Δεν αρκεί κάποιος να γνωρίζει απλά μία γλώσσα προγραμματισμού. Θα πρέπει να ακολουθήσει και μία τεχνική σχεδίασης του προγράμματός του.

Για την σύνταξη σωστών, κατανοητών και εύκολα συντηρήσιμων προγραμμάτων ακολουθήθηκαν διάφορες μεθοδολογίες ανάπτυξης που παρουσιάζονται παρακάτω:

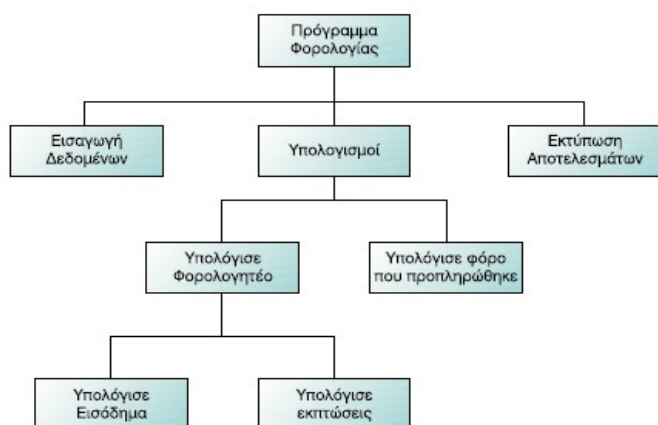
6.4.1 Ιεραρχική σχεδίαση ή Top down σχεδίαση

Η τεχνική αυτή συνιστά στον καθορισμό των βασικών λειτουργιών του προγράμματος σε ανώτερο επίπεδο και στη συνέχεια τη διάσπαση καθεμιάς σε μικρότερες και απλούστερες μέχρι του σημείου να είναι τόσο απλές που μπορούν να επιλυθούν άμεσα.

Συνεπώς, σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε μικρότερα κι απλούστερα υπο-προβλήματα τα οποία είναι ευκολότερο να επιλυθούν.

Για την Top-down σχεδίαση χρησιμοποιούμε το **ιεραρχικό διάγραμμα**.

Παράδειγμα: Πρόγραμμα Υπολογισμός Καθαρού Μισθού



6.4.2 Τμηματικός προγραμματισμός

Η ιεραρχική σχεδίαση υλοποιείται με τον τμηματικό προγραμματισμό.

Μετά την ανάλυση του προβλήματος σε μικρότερα και απλούστερα υπο-προβλήματα, κάθε υπο-πρόβλημα αποτελεί μία ξεχωριστή και ανεξάρτητη **ενότητα** (module). Τώρα, για κάθε ενότητα θα γραφτεί το κατάλληλο πρόγραμμα ή τμήμα προγράμματος.

6.4.3 Δομημένος προγραμματισμός

Είναι η μεθοδολογία που έχει επικρατήσει σήμερα. Εμπεριέχει τις αρχές της ιεραρχικής σχεδίασης και του τμηματικού προγραμματισμού. Επιπλέον, αναφέρει ότι για τη δημιουργία σωστών προγραμμάτων να χρησιμοποιούμε μόνο τις 3 στοιχειώδεις δομές : Ακολουθίας, Επιλογής και Επανάληψης.

Όλα τα προγράμματα, οσοδήποτε μεγέθους, μπορούν να γραφτούν στηριζόμενα στη χρήση μόνο αυτών των δομών, αποφεύγοντας πλήρως τη χρήση τη δομής GOTO. Επίσης, κάθε ενότητα προγράμματος έχει μόνο μία είσοδο και μόνο μία έξοδο.

Πλεονεκτήματα του δομημένου προγραμματισμού

- Άμεση υλοποίηση των αλγορίθμων σε πρόγραμμα.
- Διευκόλυνση της ανάλυσης του προγράμματος σε τμήματα (ενότητες) . Το κάθε τμήμα μπορεί να γραφτεί από διαφορετικές ομάδες προγραμματιστών.
- Ευκολότερη και συντομότερη ανάπτυξη προγραμμάτων.
- Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
- Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
- Ευκολότερη διόρθωση και συντήρηση του προγράμματος.

6.5 Αντικειμενοστρεφής προγραμματισμός

Η φιλοσοφία του αντικειμενοστραφούς προγραμματισμού είναι η εξής: Να θεωρήσουμε τα δεδομένα και τις αποδεκτές ενέργειες που γίνονται πάνω σε αυτά ως ένα ενιαίο αντικείμενο (object). Αυτό το αντικείμενο θα μπορεί να χρησιμοποιηθεί πολύ εύκολα οπουδήποτε αλλού, δηλαδή θα είναι επαναχρησιμοποιήσιμο.

Για παράδειγμα, θα μπορούσαμε να θεωρήσουμε μία στοίβα ως ένα αντικείμενο : Ένας πίνακας Π[1:20] που θα δέχεται ακεραίους. Οι μόνες ενέργειες που θα ήταν επιτρεπτές θα ήταν η ώθηση (push) ενός στοιχείου στη στοίβα και η αφαίρεση (pop) ενός στοιχείου από τη στοίβα. Αυτό το αντικείμενο θα μπορούσαμε να το δώσουμε και σε όποιον άλλο το είχε ανάγκη.

Ο αντικειμενοστρεφής προγραμματισμός ακολουθεί 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, Mac Os 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..

Τα σύγχρονα προγραμματιστικά περιβάλλοντα είναι κτισμένα πάνω στις αρχές του οπτικού και καθοδηγούμενου από γεγονότα προγραμματισμού.

.7 Προγραμματιστικά περιβάλλοντα

Ένα πρόγραμμα που φτιάχνεται σε μία γλώσσα υψηλού επιπέδου δεν είναι άμεσα κατανοητό από τον Η/Υ. Θα πρέπει να μεταφραστεί σε ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δυαδική μορφή). Την διαδικασία μετάφρασης την πραγματοποιούν τα μεταφραστικά προγράμματα. Είναι δύο ειδών:

- Μεταγλωττιστές (Compilers)
- Διερμηνευτές (Interpreters)

Πηγαίο πρόγραμμα (Source program): Το πρόγραμμα που είναι φτιαγμένο σε γλώσσα υψηλού επιπέδου.

Αντικείμενο πρόγραμμα (Object program): Το πρόγραμμα που είναι μεταφρασμένο σε γλώσσα μηχανής αλλά όχι άμεσα εκτελέσιμο.

Εκτελέσιμο πρόγραμμα (Executable program): Το πρόγραμμα σε γλώσσα μηχανής που είναι έτοιμο πλέον να εκτελεστεί.

Μεταγλωττιστής: Παίρνει, ως είσοδο, το πηγαίο πρόγραμμα και αναλαμβάνει να το μεταφράσει εξ' ολοκλήρου παράγοντας το αντικείμενο πρόγραμμα.

Διερμηνευτής: Παίρνει, ως είσοδο, μία-μία εντολή του πηγαίου, την μεταφράζει και την εκτελεί αμέσως. Η λειτουργία του μοιάζει με τον άνθρωπο-διερμηνέα που μεταφράζει επί τόπου κάθε πρόταση.

Η διαδικασία μεταγλώττισης και σύνδεσης προγράμματος φαίνεται παρακάτω:



Βιβλιοθήκες (Libraries): Έτοιμες ενότητες (modules) αντικείμενου προγράμματος της γλώσσας, απαραίτητες για την παραγωγή του εκτελέσιμου προγράμματος.

Συνδέτης – Φορτωτής (Linker-Loader): Ειδικό πρόγραμμα που αναλαμβάνει να συνδέσει το αντικείμενο πρόγραμμα με τις βιβλιοθήκες και να παράγει το εκτελέσιμο.

Κατά τη δημιουργία ενός προγράμματος σχεδόν πάντα ενυπάρχουν λάθη. Τα λάθη τα χωρίζουμε σε δύο κατηγορίες :

- Συντακτικά λάθη
- Λογικά λάθη

Τα **συντακτικά λάθη** ανιχνεύονται κατά την διαδικασία της μεταγλώττισης ή διερμίνευσης. Αφορούν παραβιάσεις του τυπολογικού και συντακτικού της γλώσσας. (π.χ. μία εντολή έχει γραφτεί συντακτικά λάθος). Στην περίπτωση αυτή ο προγραμματιστής πρέπει να επιστρέψει στο πηγαίο πρόγραμμα, να διορθώσει τα λάθη και να το ξανά υποβάλλει για μεταγλώττιση.

Τα **λογικά λάθη** είναι και τα πλέον δύσκολα στην ανίχνευσή τους. Έχουν να κάνουν με σφάλματα στη λογική επίλυσης του προβλήματος ή λανθασμένης διατύπωσης του αλγορίθμου (π.χ. το πρόγραμμα παράγει άλλα αποτελέσματα κι όχι τα ζητούμενα!).

Για την σύνταξη των προγραμμάτων χρησιμοποιούμε ένα ειδικό πρόγραμμα που ονομάζεται **συντάκτης (editor)**. Μοιάζει με επεξεργαστή κειμένου με επιπλέον δυνατότητες που διευκολύνουν την γρήγορη σύνταξη των πηγαίων προγραμμάτων.

Τα σύγχρονα προγραμματιστικά περιβάλλοντα περιέχουν οπωσδήποτε 3 ειδών προγράμματα : Τον συντάκτη, το μεταγλωττιστή και το συνδέτη - φορτωτή. Πέραν αυτών, περιέχουν όλα εκείνα τα εργαλεία που διευκολύνουν την εύκολη, ταχύτατη σύνταξη, διόρθωση και συντήρηση των προγραμμάτων γι' αυτό και πολλές φορές ονομάζονται ως **RAD περιβάλλοντα (Rapid Application Development)**.

Όλα αυτά παρέχονται με έναν ενιαίο και λειτουργικό τρόπο στον προγραμματιστή.