

Πρόγραμμα είναι το σύνολο των εντολών που χρειάζεται να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος.

Αρχικά τα προγράμματα γράφονταν σε **γλώσσα μηχανής**, δηλαδή ακολουθίες δυαδικών ψηφίων, που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Ο συγκεκριμένος τρόπος γραφής προγραμμάτων είναι επίπονος και ελάχιστοι μπορούν να τον κατανοήσουν και να τον υλοποιήσουν, αφού απαιτεί βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή.

Στη συνέχεια αναπτύχθηκαν οι **συμβολικές γλώσσες** οι οποίες κάνουν χρήση εντολών 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).

Οι συμβολικές γλώσσες ήταν σαφώς μια εξέλιξη αλλά παραμένουν στενά συνδεδεμένες με την αρχιτεκτονική του κάθε υπολογιστή. Επιπλέον η έλλειψη εντολών σύνθετων λειτουργιών στις παραπάνω γλώσσες οδηγεί σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και να συντηρηθούν. Ακόμη, δεν είναι δυνατό να μεταφερθούν και να εκτελεστούν σε υπολογιστή διαφορετικής αρχιτεκτονικής.

Οι παραπάνω ανεπάρκειες και η προσπάθεια για καλύτερη επικοινωνία ανθρώπου – μηχανής οδήγησαν στην εμφάνιση των **γλωσσών υψηλού επιπέδου**. Σε σχέση με τις συμβολικές γλώσσες στις γλώσσες υψηλού επιπέδου:

- είναι φυσικότερος και πιο ανθρώπινος ο τρόπος έκφρασης των προβλημάτων.
- υπάρχει δυνατότητα μεταφοράς, «μεταφερσιμότητα» δηλαδή, εκτέλεσης των προγραμμάτων σε οποιοδήποτε υπολογιστή.
- είναι εύκολη η εκμάθηση, η διόρθωση των λαθών και η συντήρηση των προγραμμάτων.



Εντολές σε γλώσσα μηχανής που καταχωρούν το άθροισμα των τιμών δύο θέσεων μνήμης σε μία άλλη.

```
0000001001011010
0000101001011110
0000011011011110
```

Η περιγραφή των παραπάνω εντολών είναι η εξής:

- Μετάφερε στον καταχωρητή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011010.
- Πρόσθεσε στο περιεχόμενο του καταχωρητή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011110.
- Μετάφερε και αποθήκευσε το περιεχόμενο του καταχωρητή στη θέση μνήμης με διεύθυνση 11011110.



Εντολές σε **συμβολική γλώσσα** που καταχωρούν το άθροισμα των τιμών δύο θέσεων μνήμης σε μία άλλη.

```
LDA B
ADD C
STA A
```

Η περιγραφή των παραπάνω εντολών είναι η εξής:

- Μετάφερε στον καταχωρητή το περιεχόμενο της θέσης μνήμης με όνομα B.
- Πρόσθεσε στο περιεχόμενο του καταχωρητή το περιεχόμενο της θέσης μνήμης με όνομα C.
- Μετάφερε και αποθήκευσε το περιεχόμενο του καταχωρητή στη θέση μνήμης με όνομα A.

Εντολή στη γλώσσα **BASIC** που καταχωρεί το άθροισμα των τιμών δύο μεταβλητών B και C στη μεταβλητή A.

```
A = B + C
```

Η εμφάνιση των γραφικών περιβαλλόντων εργασίας δημιούργησε την ανάγκη για προγράμματα που να εκμεταλλεύονται τον γραφικό τρόπο επικοινωνίας χρήστη – υπολογιστή. Έτσι γλώσσες όπως η BASIC, η C++, η PASCAL που είναι μεν κειμενικές, εξελίχθηκαν (Visual Basic, Visual C++, Delphi) ώστε να διαθέτουν και οπτικό περιβάλλον προγραμματισμού.

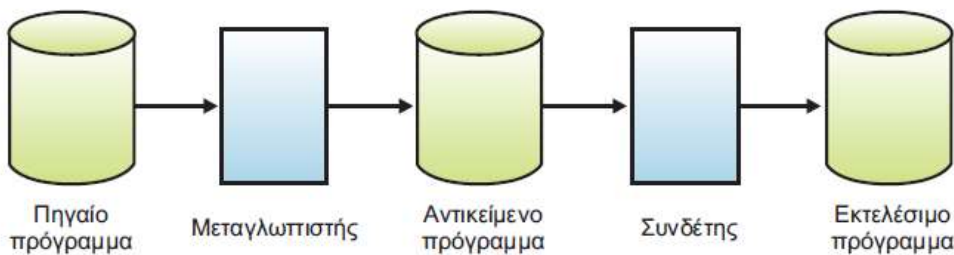
Αναπτύχθηκαν όμως και γλώσσες όπως η **SCRATCH – BYOB** και η **Google AppInvertor** που είναι αποκλειστικά οπτικές γλώσσες προγραμματισμού (Visual Programming Languages, VPL). Οι συγκεκριμένες δίνουν τη δυνατότητα στον προγραμματιστή να δημιουργήσει προγράμματα μέσα από το γραφικό χειρισμό προγραμματιστικών στοιχείων (αντί κειμένου).

Η χρήση των υπολογιστών σχεδόν σε όλες τις εκφάνσεις της ανθρώπινης δραστηριότητας δημιούργησε την ανάγκη για γλώσσες κατάλληλες στην επίλυση συγκεκριμένων προβλημάτων. Έτσι αναπτύχθηκαν γλώσσες όπως η **LOGO** ή η **GameMaker** για εκπαιδευτικούς σκοπούς, η **LabView** που χρησιμοποιείται από τους επιστήμονες και τους μηχανικούς στο σχεδιασμό, τον έλεγχο και τη δοκιμή καταναλωτικών προϊόντων κ.ά.

Σχεδίαση και συγγραφή κώδικα

Οι περισσότερες γλώσσες προγραμματισμού υποστηρίζονται από προγραμματιστικά περιβάλλοντα τα οποία παρέχουν εργαλεία που διευκολύνουν την εργασία του προγραμματιστή. Για τη σύνταξη του **πηγαίου προγράμματος** χρησιμοποιείται ένα ειδικό πρόγραμμα το οποίο ονομάζεται **συντάκτης** (editor). Στη συνέχεια το πηγαίο πρόγραμμα πρέπει να μεταφραστεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή δηλαδή σε εντολές γλώσσας μηχανής. Το έργο της μετάφρασης το αναλαμβάνουν δύο προγράμματα ο **μεταγλωττιστής** ή ο **διερμηνευτής**.

Για να μεταφραστεί το πηγαίο πρόγραμμα σε εντολές γλώσσας μηχανής δεν θα πρέπει να ανιχνευθούν λάθη. Τα λάθη που εμφανίζονται κατά τη μετάφραση ονομάζονται **συντακτικά**. Τα συντακτικά λάθη μπορεί να οφείλονται σε αναγραμματισμούς, σε λανθασμένη σύνταξη εντολών, παράλειψη δήλωσης μεταβλητών κ.ά.. Ο μεταφραστής ανιχνεύει τα λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα. Στη συνέχεια ακολουθεί η διόρθωσή τους από τον προγραμματιστή.



Πέρα όμως από τα συντακτικά λάθη υπάρχουν και τα **λογικά** που δεν είναι δυνατό να ανιχνευθούν από τα μεταφραστικά προγράμματα. Τα περισσότερα όμως προγραμματιστικά περιβάλλοντα παρέχουν εργαλεία εκσφαλμάτωσης που βοηθούν τον προγραμματιστή να εκτελέσει το πρόγραμμα εντολή προς εντολή μέχρι συγκεκριμένο σημείο ή να παρακολουθεί τις τιμές των μεταβλητών έτσι ώστε να εντοπίσει τα λάθη στην υλοποίηση του αλγορίθμου.

Οι **μεταγλωττιστές** (compilers) δέχονται στην είσοδο ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγουν ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. Το πρόγραμμα που παράγεται ονομάζεται **αντικείμενο** (object) πρόγραμμα. Το αντικείμενο πρόγραμμα δεν είναι σε θέση να εκτελεστεί. Χρειάζεται να συνδεθεί με άλλα τμήματα προγράμματος τα οποία είτε τα γράφει ο προγραμματιστής, είτε βρίσκονται στις βιβλιοθήκες (libraries) της γλώσσας. Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης - φορτωτής** (linker - loader). Το αποτέλεσμα είναι η παραγωγή του **εκτελέσιμου** (executable) προγράμματος (Εικόνα 2.31). Το τελευταίο μπορεί να εκτελείται οποτεδήποτε από τον υπολογιστή και είναι τελειώς ανεξάρτητο από το πηγαίο πρόγραμμα.

Οι **διερμηνευτές** (interpreters) διαβάζουν μία προς μία τις εντολές του πηγαίου προγράμματος και για καθεμία εκτελούν αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

Οι κατηγορίες των γλωσσών προγραμματισμού με βάση την εξέλιξή τους:

Γλώσσες	Περιγραφή	Παράδειγμα	Συγκριτικά	
ΜΗΧΑΝΗΣ	Το πρόγραμμα περιέχει εντολές που είναι ακολουθίες δυαδικών ψηφίων 0 ή 1 και είναι κατανοητές από τον Η/Υ, όχι όμως από τον άνθρωπο.	00000100 01000011 00100000 11001100 10001001	Πλεονεκτήματα	Μειονεκτήματα
ΣΥΜΒΟΛΙΚΕΣ ή ΧΑΜΗΛΟΥ ΕΠΙΠΕΔΟΥ	Οι εντολές γράφονται από τον άνθρωπο με λέξεις που εύκολα κατανοούνται. Έπειτα αντιστοιχίζονται από τους μεταφραστές σε 0 και 1.	INDEX=\$01 LDA N ADD +2 STA N BRK	Πλεονεκτήματα	Μειονεκτήματα
ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ	Οι εντολές δίνονται πλέον με μια μορφή που πλησιάζει τη φυσική γλώσσα.	10 FOR i=1 TO 8 20 PRINT i 30 NEXT i 40 END	Πλεονεκτήματα	Μειονεκτήματα
4ης ΓΕΝΙΑΣ	Πλέον ο χρήστης μπορεί να υποβάλει ερωτήσεις στο σύστημα ή να αναπτύσσει εφαρμογές που ανακτούν πληροφορίες από βάσεις δεδομένων.	SELECT * FROM Book WHERE price>50 ORDER BY title;	Πλεονεκτήματα	Μειονεκτήματα

Σημαντικότερες γλώσσες προγραμματισμού:

Γλώσσα	Χρονολ.	Πληροφορίες
FORTRAN	1957	Κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων
COBOL	1960	Κατάλληλη για την ανάπτυξη εμπορικών εφαρμογών
ALGOL	1960	Κατάλληλη για τη δημιουργία γενικής φύσεως προγραμμάτων
PL/1	μέσα '60	Χωρίς επιτυχία προσπάθησε να καλύψει όλους τους τομείς του προγραμματισμού
LISP	μέσα '60	Δημιουργήθηκε για την ανάπτυξη προγραμμάτων στο χώρο της τεχνητής νοημοσύνης
PROLOG	αρχές '70	Δημιουργήθηκε για την ανάπτυξη προγραμμάτων στο χώρο της τεχνητής νοημοσύνης
BASIC	μέσα '60	Δημιουργήθηκε με σκοπό την εκπαίδευση των αρχάριων στον προγραμματισμό
PASCAL	1970	Κατάλληλη για τη δημιουργία δομημένων προγραμμάτων
C	αρχές '70	Δημιουργήθηκε για την ανάπτυξη του λειτουργικού συστήματος UNIX
JAVA	1995	Δημιουργήθηκε για την ανάπτυξη διαδικτυακών εφαρμογών