

9 Η έννοια του προγράμματος



Στοιχεία θεωρίας

Είδαμε στα προηγούμενα ότι η επίλυση ενός προβλήματος έχει τρία σημαντικά στάδια:

- Τον ακριβή προσδιορισμό του προβλήματος ,
- Την ανάλυση του προβλήματος και τη κατασκευή του αλγορίθμου και
- Την τη διατύπωση του αλγορίθμου σε μορφή κατανοητή από τον υπολογιστή.

Ο **Προγραμματισμός** λοιπόν έρχεται να μας βοηθήσει στο τρίτο βήμα. Στη δημιουργία δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή έτσι ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος.



Ο υπολογιστής δεν παίρνει αποφάσεις. Καταλαβαίνει μόνο δύο καταστάσεις που αντιπροσωπεύονται με τα δύο ψηφία του διαδικού συστήματος, το «0» και το «1». Όλες οι εντολές είναι ακολουθίες των ψηφίων αυτών και υλοποιούνται από τον υπολογιστή με στοιχειώδεις αριθμητικές και λογικές πράξεις.

9.1 Γλώσσες προγραμματισμού



Στοιχεία θεωρίας

Βασικές κατηγορίες που κατατάσσονται οι γλώσσες προγραμματισμού.

Τις γλώσσες προγραμματισμού μπορούμε να τις κατατάξουμε σε τέσσερις κατηγορίες:

- Γλώσσες μηχανής
- Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου.
- Γλώσσες υψηλού επιπέδου
- Γλώσσες 4ης γενιάς.

Παρατηρήσεις:



- Υπάρχουν εκατοντάδες γλώσσες προγραμματισμού.
- Ο προγραμματιστής επιλέγει την κατάλληλη για το είδος εφαρμογής που θέλει να υλοποιήσει, ανάλογα με τον διαθέσιμο εξοπλισμό, τις γνώσεις και τις προτιμήσεις.
- Κάθε γλώσσα προγραμματισμού σχεδιάζεται για συγκεκριμένο σκοπό, όπως για μαθηματικές, εμπορικές, επιστημονικές, λογιστικές και άλλες ανάγκες.
- Οι γλώσσες ανανεώνονται, εξελίσσονται και εμπλουτίζονται.
- Σχεδόν όλες οι γλώσσες προγραμματισμού έχουν κοινά βασικά χαρακτηριστικά, τύπους δεδομένων, βασικές δομές και εντολές.

9.1.1 Γλώσσες μηχανής.

Οι **γλώσσες μηχανής** είναι οι πρώτες γλώσσες προγραμματισμού στις οποίες οι εντολές για την υλοποίηση ενός προγράμματος έπρεπε να δοθούν από τον προγραμματιστή σε ακολουθίες από 0 και 1, δηλαδή σε μορφή κατανοητή από τον υπολογιστή αλλά δυσνόητη από τον άνθρωπο.

Ο προγραμματισμός στη γλώσσα αυτή ήταν πολύ κοπιαστικός, επίπονος και απαιτούσε καλή γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή.

9.1.2 Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου.

Οι **γλώσσες χαμηλού επιπέδου** οι οποίες αποτελούν εξέλιξη της γλώσσας μηχανής, είναι στενά εξαρτημένες με την αρχιτεκτονική του υπολογιστή. Χρησιμοποιούσαν μια συμβολική γλώσσα η οποία είχε έννοια για τον άνθρωπο και εσωτερικά μετατρεπόταν από τους υπολογιστές, σε ακολουθίες από 0 και 1.

Είναι δηλαδή μια εξέλιξη των γλωσσών μηχανής.

Το έργο της μετάφρασης το αναλάμβανε ένα ειδικό πρόγραμμα που ονομαζόταν **συμβολομεταφραστής** (assembler). Αναλάμβανε δηλαδή να μεταφράζει την συμβολική γλώσσα στη γλώσσα μηχανής του δεδομένου υπολογιστή.

Όμως και πάλι παρέμεναν στενά συνδεδεμένοι με την αρχιτεκτονική του υπολογιστή και δεν διέθεταν εντολές σύνθετων λειτουργιών.

Έτσι ο προγραμματισμός και στη γλώσσα αυτή ήταν πολύ κοπιαστικός, και επίπονος.



Μειονεκτήματα των συμβολικών γλωσσών προγραμματισμού.

- Είναι στενά συνδεδεμένες με την αρχιτεκτονική του υπολογιστή.
- Δεν διαθέτουν εντολές πιο σύνθετων λειτουργιών, οδηγώντας έτσι σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και κυρίως να συντηρηθούν.
- Τα προγράμματα δεν μπορούν να μεταφερθούν σε άλλο διαφορετικό υπολογιστή, ακόμη και του ίδιου κατασκευαστή.

9.1.3 Γλώσσες υψηλού επιπέδου. (Απλή ανάγνωση)

Στα τέλη της δεκαετίας του 50 εμφανίστηκαν οι πρώτες **γλώσσες υψηλού επιπέδου** που πρόσφεραν καλύτερη επικοινωνία ανθρώπου – μηχανής.

Το 1957 η IBM πρώτη παρουσίασε μια γλώσσα υψηλού επιπέδου, την **FORTAN**. Το όνομα της προέρχεται από τις λέξεις FORmula TRANslation, που σημαίνουν μετάφραση τύπων. Ήταν μια γλώσσα για επίλυση μαθηματικών και επιστημονικών προβλημάτων.

Τότε εμφανίζεται και ο **μεταγλωτιστής**. Ένα ειδικό πρόγραμμα που μεταφράζει το πρόγραμμα που γράφεται σε FORTRAN ή σε οποιαδήποτε γλώσσα υψηλού επιπέδου, σε ακολουθίες των εντολών της γλώσσας μηχανής. Έτσι τα προγράμματα που είναι γραμμένα σε Fortran μπορούν να μεταφερθούν και να εκτελεστούν σε οποιοδήποτε υπολογιστή αρκεί αυτός να διαθέτει τον αντίστοιχο μεταγλωτιστή.

Το 1960 αναπτύχθηκε μια άλλη γλώσσα, η **COBOL**. Το όνομά της προέρχεται από τα αρχικά των λέξεων Common Business Oriented Language - κοινή γλώσσα προσανατολισμένη στις επιχειρήσεις. Είναι γλώσσα κατάλληλη για ανάπτυξη εμπορικών και γενικότερα διαχειριστικών εφαρμογών και αποτέλεσε σταθμό για τον προγραμματισμό. Καθιερώθηκε πολύ σύντομα και χρησιμοποιήθηκε από την πλειονότητα των επιχειρήσεων αλλά και από τη δημόσια διοίκηση. Ακόμη και σήμερα πολλές εφαρμογές στηρίζουν τη λειτουργία τους στη γλώσσα αυτή.

Η **ALGOL** (ALGOrithmic Language – Αλγοριθμική γλώσσα) αναπτύχθηκε και αυτή το 1960 από ευρωπαίους επιστήμονες σαν γλώσσα γενικών εφαρμογών χωρίς να γνωρίσει σημαντική πρακτική εφαρμογή.

Η γλώσσα προγραμματισμού **BASIC** (Beginner's All purpose Symbolic Instruction Code – Συμβολικός Κώδικας Εντολών Γενικής Χρήσης για Αρχάριους) αναπτύχθηκε στα μέσα της δεκαετίας του '60 για εκπαιδευτικούς σκοπούς. Σχεδιάστηκε για την εκπαίδευση αρχαρίων στο προγραμματισμό και για τη κατασκευή σύντομων και απλών προγραμμάτων. Γρήγορα έγινε μια πολύ δημοφιλής γλώσσα προγραμματισμού και δημιουργήθηκαν γι' αυτήν πολλές εκδόσεις.

Η γλώσσα προγραμματισμού **Pascal** δημιουργήθηκε το 1970 από τον Niklaus Wirth. Είναι μια εκπαιδευτική γλώσσα προγραμματισμού αλλά μπορεί να υποστηρίξει και πολλές σύνθετες και ισχυρές εφαρμογές. Γνώρισε και γνωρίζει μεγάλη εξάπλωση γιατί είναι μια γλώσσα κατάλληλη για δημιουργία δομημένων προγραμμάτων.

Η εμφάνιση του **δομημένου προγραμματισμού** έφερε ριζικές αλλαγές στον προγραμματισμό. Άλλαξε τη σχεδίαση των προγραμμάτων και δημιούργησε νέες τεχνικές. Καθιέρωσε απλά προγράμματα εύκολα στη συγγραφή αλλά και στη διόρθωσή τους.

Μια γλώσσα με κοινά χαρακτηριστικά με τη Pascal είναι και η **C**. Κατάλληλη για ανάπτυξη δομημένων εφαρμογών αλλά και με πολλά ισχυρά χαρακτηριστικά. Αναπτύχθηκε στα εργαστήρια της BELL και χρησιμοποιήθηκε για τη δημιουργία του λειτουργικού συστήματος Unix.

Εξέλιξη της C αποτελεί η **C++**. Η C++ αποτελεί ένα χαρακτηριστικό εκπρόσωπο του αντικειμενοστραφούς προγραμματισμού. Μια ιδέα που παρουσιάστηκε για πρώτη φορά στη δεκαετία του 1970 και άλλαξε τον παραδοσιακό προγραμματισμό.

Μια γλώσσα που παρουσιάστηκε τα τελευταία χρόνια είναι η **java**. Είναι μια αντικειμενοστραφής γλώσσα που αναπτύχθηκε με σκοπό ειδικά για προγραμματισμό εφαρμογών που θα εκτελούνται σε κατακευματισμένα περιβάλλοντα όπως το διαδίκτυο. Δηλαδή σε διαφορετικούς υπολογιστές προσωπικούς ή μεγάλα συστήματα που θα είναι συνδεδεμένοι μεταξύ τους ανεξάρτητα και με διαφορετικά λειτουργικά συστήματα.

Η εμφάνιση των **γραφικών περιβαλλόντων εργασίας** δημιούργησε την ανάγκη να αναπτυχθούν προγράμματα που θα εκμεταλλεύονται τον γραφικό τρόπο επικοινωνίας χρήστη και υπολογιστή. Έτσι κατασκευάστηκαν νέες εκδόσεις γλωσσών προγραμματισμού που είχαν για χαρακτηριστικό τους τον οδηγούμενο από το γεγονός προγραμματισμό και τον οπτικό προγραμματισμό.

Με τον όρο **οπτικό προγραμματισμό** εννοούμε την δυνατότητα να δημιουργούμε γραφικά ολόκληρο το περιβάλλον της εφαρμογής για παράδειγμα τα πλαίσια διαλόγου ή τα “μενού”.

Με τον όρο **οδηγούμενο από το γεγονός** εννοούμε τη δυνατότητα να ενεργοποιούνται λειτουργίες του προγράμματος με την εκτέλεση ενός γεγονότος, για παράδειγμα την επιλογή μιας εντολής από ένα μενού ή το “κλικ” του ποντικιού.

Χαρακτηριστικά παραδείγματα είναι οι γλώσσες προγραμματισμού **Visual Basic** και **Visual Pascal**.

Συνοπτικά οι γλώσσες προγραμματισμού υψηλού επιπέδου και είδος των δραστηριοτήτων για την οποία χρησιμοποιείται:

Όνομα Γλώσσας	Τομέας Εφαρμογών - χαρακτηριστικά
Fortran	Επιστημονικός.
Cobol	Εμπορικός.
Algol	Γενικής Χρήσης.
Prolog	Τεχνητής νοημοσύνης.
Lisp	Τεχνητής νοημοσύνης.
Pascal	Επίλυση μαθηματικών και επιστημονικών προβλημάτων – Εκπαίδευση – Δομημένος προγραμματισμός.
Basic	Γενικής χρήσης – Εκπαίδευση.
C, C++	Προγραμματισμός συστημάτων - Αντικειμενοστραφής προγραμματισμός.
Java	Προγραμματισμός στο διαδίκτυο - Αντικειμενοστραφής προγραμματισμός.
PL/1	Επιστημονικός και Εμπορικός.
Visual Basic	Προγραμματισμός σε περιβάλλοντα Windows (Οπτικός προγραμματισμός).
Visual Pascal	Προγραμματισμός σε περιβάλλοντα Windows (Οπτικός προγραμματισμός).

Πλεονεκτήματα των γλωσσών υψηλού επιπέδου σε σχέση με τις συμβολικές γλώσσες προγραμματισμού.

Τα πλεονεκτήματα των γλωσσών υψηλού επιπέδου σε σχέση με τις συμβολικές γλώσσες προγραμματισμού είναι τα ακόλουθα:



- Φυσικότερος και πιο ανθρώπινος τρόπος έκφρασης των προβλημάτων. Τα προγράμματα σε γλώσσα υψηλού επιπέδου είναι πιο κοντά στο πρόβλημα που επιλύουν.
- Η ανεξαρτησία από τον τύπο του υπολογιστή. Τα προγράμματα σε γλώσσα υψηλού επιπέδου μπορούν να μεταφερθούν και να εκτελεστούν σε οποιονδήποτε υπολογιστή με ελάχιστες ή καθόλου μετατροπές, δηλαδή έχουν την ικανότητα της **μεταφεριμότητας**.
- Η ευκολία της εκμάθησης και εκπαίδευσης ως απόρροια των προηγούμενων.

- © Η διόρθωση λαθών και η συντήρηση προγραμμάτων γίνεται ευκολότερα μέσα από σύγχρονα και φιλικά προς τον χρήστη περιβάλλοντα εργασίας.

Οι γλώσσες υψηλού επιπέδου δηλαδή ελάττωσαν το χρόνο και το κόστος παραγωγής προγραμμάτων αφού πολύ λιγότεροι προγραμματιστές χρειάζονται για να αναπτυχθούν προγράμματα που έχουν τη δυνατότητα να χρησιμοποιούνται σε περισσότερους υπολογιστές.

9.1.4 Γλώσσες 4ης γενιάς.

Οι **γλώσσες 4ης γενιάς** είναι οι γλώσσες εφοδιασμένες με εργαλεία προγραμματισμού που αποκρύπτουν πολλές λεπτομέρειες από τις τεχνικές υλοποίησης και με αυτά ο χρήστης μπορεί να επιλύει μόνος του μικρά προβλήματα εφαρμογών.

Στις γλώσσες αυτές ο χρήστης ενός υπολογιστή έχει τη δυνατότητα σχετικά εύκολα να υποβάλλει ερωτήσεις στο σύστημα ή να αναπτύσσει εφαρμογές που αποκτούν πληροφορίες από βάσεις δεδομένων και να καθορίζει τον ακριβή τρόπο εμφάνισης αυτών των πληροφοριών.