

Γλώσσες Προγραμματισμού



Οι γλώσσες που καταλαβαίνουν οι υπολογιστές είναι τεχνητές γλώσσες που ονομάζονται γλώσσες προγραμματισμού. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για την επικοινωνία του ανθρώπου με τη μηχανή, όπως οι φυσικές γλώσσες (ελληνική, αγγλική, γαλλική, κ.λ.π) χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων.

Οι γλώσσες προγραμματισμού έχουν και αυτές το δικό τους λεξιλόγιο και το δικό τους συντακτικό. Αν θλέουμε να προγραμματίσουμε τον υπολογιστή, για να εκτελεί πιστά τις λειτουργίες που του ζητάμε, πρέπει να μάθουμε κάποια γλώσσα προγραμματισμού. Δυστηχώς οι υπολογιστές δεν έχουν σχεδιαστεί να καταλαβαίνουν τη γλώσσα που μιλάμε δηλαδή τη φυσική γλώσσα. Η πρόοδος, όμως, στον τομέα αυτό είναι σημαντική και πιθανόν στο μέλλον να δίνουμε οδηγίες στον υπολογιστή με την ομιλία.

Όπως ξέρουμε η λειτουργία των υπολογιστών βασίζεται στην αναπαράσταση 2 μόνο ψηφίων, των 0 και 1. Στα πρώτα βήματα της ιστορίας των υπολογιστών οι άνθρωποι, για να επικοινωνήσουν με τον υπολογιστή, έπρεπε να χρησιμοποιούν μια γλώσσα που ονομάζεται Assembly αρκετά δύσκολη για να την μάθει κάποιος, γιατί είναι πολύ διαφορετική από τη φυσική μας γλώσσα. Επίσης δεν είναι ενιαία σε όλους τους υπολογιστές, μια και κάθε τύπος υπολογιστή (με διαφορετικό επεξεργαστή) έχει τη δική του γλώσσα μηχανής.

Χαρακτηριστικά Γλωσσών Προγραμματισμού

Με τη πάροδο των χρόνων οι γλώσσες προγραμματισμού εξελίχθηκαν, ώστε να μοιάζουν όλο και περισσότερο με τη φυσική μας γλώσσα. Στις μέρες μας υπάρχουν διάφορες γλώσσες προγραμματισμού, που χρησιμοποιούνται για την ανάπτυξη γενικών εφαρμογών, ενώ άλλες είναι πιο εξειδικευμένες και χρησιμοποιούνται για πιο ειδικά επιστημονικά προβλήματα (ανωτέρων μαθηματικών, μηχανικής, προσομοίωσης πειραμάτων κ.λ.π) και εξειδικευμένες εφαρμογές (προγραμματισμός ιστοσελίδων, διαχείριση εμπορικών δεδομένων κ.λ.π.). Μερικές γνωστές γλώσσες προγραμματισμού είναι η Visual Basic, η Fortran, η Python, η C, η Java, PHP και άλλες.

Όπως και οι φυσικές γλώσσες, έτσι και κάθε γλώσσα προγραμματισμού έχει ως βασικά χαρακτηριστικά:

- το αλφάβητο,
- το λεξιλόγιο και
- συντακτικό

Το αλφάβητο μιας γλώσσας προγραμματισμού είναι το σύνολο των χαρακτηριστικών που χρησιμοποιούνται από τη γλώσσα.

Το λεξιλόγιο μιας γλώσσας είναι το σύνολο των λέξεων που αναγνωρίζει η γλώσσα και έχουν συγκεκριμένη και μοναδική σημασία. Στις γλώσσες προγραμματισμού το λεξιλόγιο είναι πολύ περιορισμένο (μερικές δεκάδες λέξεις), για να μπορούμε να το μάθουμε εύκολα.

Το συντακτικό μιας γλώσσας προγραμματισμού είναι το σύνολο των κανόνων που πρέπει να ακολουθούμε, για να συνδέουμε λέξεις σε προτάσεις. Σε μια γλώσσα προγραμματισμού η σύνδεση λέξεων δημιουργεί ολοκληρωμένες εντολές προς τον υπολογιστή. Αν δεν ακολουθήσουμε αυστηρά το συντακτικό μιας γλώσσας, είναι αδύνατο για τον υπολογιστή να καταλάβει ποια εντολή του δίνουμε.

Το ολοκληρωμένο προγραμματιστικό περιβάλλον

Οι σύγχρονες γλώσσες προγραμματισμού μας προσφέρουν ένα φιλικό περιβάλλον, API έτσι ώστε γρήγορα να αναπτύσσουμε τα προγράμματά μας. Ένα περιβάλλον προγραμματισμού αποτελείται από διάφορα εργαλεία που βοηθάνε τον προγραμματιστή να γράψει και να διορθώσει τα προγράμματά του. Τα κύρια εργαλεία είναι:

- ένας εξειδικευμένος κειμενογράφος, που χρησιμεύει για τη σύνταξη και τη διόρθωση του προγράμματος και
- ένα πρόγραμμα-μεταφραστής που μετατρέπει τις οδηγίες μας στη μορφή που τις καταλαβαίνει ο επεξεργαστής, δηλαδή σε μια σειρά από 0 και 1.

Αν σε κάποια οδηγία έχουμε κάνει λάθος στο αλφάβητο, στο λεξιλόγιο ή στο συντακτικό τότε το πρόγραμμα που μετατρέπει τις οδηγίες μας σε μια σειρά από 0 και 1 θα μας δώσει ένα κατάλληλο μήνυμα λάθους, ώστε να μας βοηθήσει να διορθώσουμε το λάθος μας. Τα λάθη αυτά ονομάζονται συντακτικά. Τα προγράμματα που μετατρέπουν τις οδηγίες μας σε 0 και 1 μπορούν να χωριστούν σε δύο κατηγορίες:

- στους μεταγλωττιστές και
- στους διερμηνείς.

Η διαφορά τους είναι ότι οι μεταγλωττιστές (compilers) θα ελέγξουν όλο το πρόγραμμα για συντακτικά λάθη και μετά θα το μετατρέψουν όλο σε μια κατάλληλη σειρά από 0 και 1, ώστε να μπορεί να εκτελεστεί στον επεξεργαστή του υπολογιστή.

Αντίθετα οι διερμηνείς (interpreters) ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία.

Δεν πρέπει να ξεχνάμε ότι ο υπολογιστής εκτελεί πιστά, όποιες συντακτικά ορθές εντολές και αν του δώσουμε. Αν το αποτέλεσμα που προκύπτει τελικά από την εκτέλεση του προγράμματος, δεν είναι το αναμενόμενο, τότε το πρόβλημα δεν βρίσκεται στον τρόπο εκτέλεσης, αλλά στον αλγόριθμο που κατασκευάσαμε για τη λύση του προβλήματος μας. Στην περίπτωση αυτή λέμε ότι έχουμε κάνει ένα λογικό λάθος και πρέπει να ελέγξουμε ένα προς ένα τα βήματα-εντολές του αλγορίθμου μας, ώστε να διαπιστώσουμε, αν δίνουμε τις κατάλληλες εντολές με τη σωστή σειρά.

Ένα δεύτερο σημείο που πρέπει να γνωρίζουμε όταν προγραμματίζουμε, είναι ότι για τον υπολογιστή τίποτα δεν είναι αυτονόητο. Ενώ εμείς οι άνθρωποι έχουμε την ικανότητα συμπληρώνουμε τις οδηγίες κάποιου με τη λογική και την εμπειρία μας, ο υπολογιστής χρειάζεται να περιγράψουμε με μεγάλη ακρίβεια τις εντολές μας στον υπολογιστή, για να τις εκτελέσει. Αν, για παράδειγμα, του δώσουμε μια εντολή να υπολογίσει ένα άθροισμα, δεν είναι αυτονόητο ότι θα μας εμφανίσει και το αποτέλεσμα. Αν φαίνεται ότι οι υπολογιστές επιλύουν πολύ έξυπνα διάφορα προβλήματα, είναι γιατί κάποιοι άνθρωποι τους προγραμματίσαν γι'αυτό και όχι γιατί οι μηχανές είναι <<έξυπνες>>. Για να φτιάξουμε λοιπόν ένα καλό πρόγραμμα, πρέπει πρώτα να έχουμε σχεδιάσει ένα καλό αλγόριθμο. Ο ρόλος του αλγορίθμου είναι θεμελιώδης.

Ανακεφαλαιώνουμε λέγοντας ότι τα στάδια της ζωής ενός προγράμματος είναι τα ακόλουθα:

- 1.Αλγόριθμος.
- 2.Πρόγραμμα.
- 3.Μετατροπή του προγράμματος σε γλώσσα μηχανής.
- 4.Εκτέλεση στην Κ.Μ.Ε.

