

Εισαγωγή στην Έννοια του Αλγορίθμου και στον Προγραμματισμό

Ως **πρόβλημα** θεωρούμε κάθε ζήτημα που τίθεται προς επίλυση, κάθε κατάσταση που μας απασχολεί και πρέπει να αντιμετωπιστεί. Η λύση ενός προβλήματος δεν μας είναι γνωστή, ούτε προφανής.

Δεδομένα προβλήματος είναι τα στοιχεία που μας είναι γνωστά και μπορούν να μας βοηθήσουν στη λύση του προβλήματος. Σε κάθε πρόβλημα ψάχνουμε να βρούμε την απάντηση σε μια ερώτηση. Αυτό που ψάχνουμε είναι το **ζητούμενο**.

Για να επιλύσουμε ένα πρόβλημα πρέπει αρχικά να το **κατανοήσουμε**. Πρέπει δηλαδή να καταλάβουμε καλά το **περιεχόμενο** του, να διακρίνουμε τα **δεδομένα** που έχουμε στη διάθεσή μας και τα **ζητούμενά** του. Είναι σημαντικό, όμως, να προσδιορίσουμε και το «**περιβάλλον**» ή το **πλαίσιο** μέσα στο οποίο εντάσσεται το πρόβλημα (χώρος του προβλήματος).

Για να μπορέσουμε να επιλύσουμε ένα σύνθετο πρόβλημα, είναι αναγκαίο να το **αναλύσουμε** σε απλούστερα προβλήματα.

Αλγόριθμος

Οι οδηγίες που δίνουμε με λογική σειρά, ώστε να εκτελέσουμε μια εργασία ή να επιλύσουμε ένα πρόβλημα, συνθέτουν έναν Αλγόριθμο. **Αλγόριθμο ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών - βημάτων, με σκοπό την επίλυση ενός προβλήματος.**

Ιδιότητες

Όταν σχεδιάζουμε έναν αλγόριθμο, πρέπει να είμαστε ιδιαίτερα προσεκτικοί, ώστε να βάζουμε με **λογική σειρά** τις οδηγίες (instructions) που θα μας οδηγήσουν στη λύση του προβλήματός μας.

Τα βήματα που αποτελούν έναν αλγόριθμο ονομάζονται **οδηγίες ή εντολές**. Αν ακολουθηθούν οι οδηγίες ενός αλγορίθμου στο τέλος πρέπει να προκύπτει ένα αποτέλεσμα, ένα έργο. Πρέπει να είμαστε σίγουροι ότι, αν υλοποιήσουμε τον αλγόριθμο, **κάποτε θα τελειώσει επιτυχάνοντας τον αρχικό σκοπό**. Οι εντολές ενός αλγορίθμου πρέπει να έχουν **ακρίβεια και σαφήνεια**, ώστε να μην μπερδευτεί αυτός που θα υλοποιήσει τον αλγόριθμο και τις εκτελέσει με λανθασμένο τρόπο. Οι εντολές ενός αλγορίθμου πρέπει να είναι εκφρασμένες **με απλά λόγια**, ώστε να είναι απόλυτα κατανοητές.

Ένα πρόγραμμα είναι η αναπαράσταση ενός αλγορίθμου γραμμένη σε γλώσσα κατανοητή για έναν υπολογιστή. Ένα πρόγραμμα, δηλαδή, αποτελείται από μία σειρά εντολών που δίνονται στον υπολογιστή με σκοπό να εκτελέσει κάποια συγκεκριμένη λειτουργία ή να υπολογίσει κάποιο επιθυμητό αποτέλεσμα. Η εργασία σύνταξης των προγραμμάτων ονομάζεται προγραμματισμός, ενώ τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται προγραμματιστές.

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 κ.λπ.) χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων.

Γλώσσα Μηχανής (ιστορικά στοιχεία)

Όπως έχει αναφερθεί στη Β' Γυμνασίου, η λειτουργία των υπολογιστών βασίζεται στην αναπαράσταση μόνο δύο ψηφίων, των «0» και «1». Στα πρώτα βήματα της ιστορίας των

υπολογιστών οι άνθρωποι, για να επικοινωνήσουν με τον υπολογιστή, έπρεπε να χρησιμοποιούν μία γλώσσα που είχε ως αλφάβητο το «0» και το «1». Η γλώσσα αυτή ονομάστηκε **γλώσσα μηχανής**. Η γλώσσα μηχανής είναι **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

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

- Το **αλφάβητο** μιας γλώσσας προγραμματισμού είναι το σύνολο των χαρακτήρων 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).
- 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.

Ολοκληρωμένο προγραμματιστικό περιβάλλον

Τα κύρια εργαλεία είναι: α) ένας **εξειδικευμένος κειμενογράφος**, που χρησιμεύει για τη σύνταξη και τη διόρθωση του προγράμματος και β) ένα **πρόγραμμα-μεταφραστής** που μετατρέπει τις οδηγίες μας στη μορφή που τις καταλαβαίνει ο επεξεργαστής, δηλαδή σε μια σειρά από 0 και 1.

Μεταγλωττιστές και Διερμηνείς

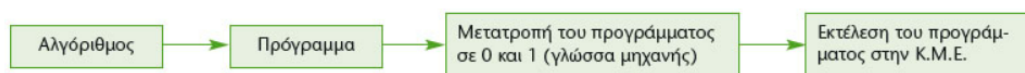
Η διαφορά τους είναι ότι οι **μεταγλωττιστές** (compilers) θα ελέγξουν όλο το πρόγραμμα για συντακτικά λάθη και μετά θα το μετατρέψουν όλο σε μια κατάλληλη σειρά από 0 και 1, ώστε να μπορεί να εκτελεστεί από τον επεξεργαστή του υπολογιστή.


Αντίθετα οι **διερμηνείς** (interpreters) ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία. Η γλώσσα προγραμματισμού Logo, που θα δούμε στο επόμενο κεφάλαιο, χρησιμοποιεί διερμηνέα.

Αν σε κάποια οδηγία έχουμε κάνει λάθος στο αλφάβητο, στο λεξιλόγιο ή στο συντακτικό τότε το πρόγραμμα που μετατρέπει τις οδηγίες μας σε 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 λάθη**.

Αν το αποτέλεσμα, που τελικά προκύπτει από την εκτέλεση του προγράμματος, δεν είναι το αναμενόμενο, τότε το πρόβλημα δε βρίσκεται στον τρόπο εκτέλεσης, αλλά στον αλγόριθμο που κατασκευάσαμε για τη λύση του προβλήματός 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.

Στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε. του υπολογιστή



 Σχήμα 1.3. Στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε. του υπολογιστή