

Κεφάλαιο 1 : Εισαγωγή στην Έννοια του Αλγορίθμου και στον Προγραμματισμό

1. Τι ονομάζουμε πρόβλημα;

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.

2. Ποιες οι κατηγορίες προβλημάτων;

- Προβλήματα που είναι σχετικά απλά και σύντομα βρίσκουμε τη λύση τους,
- Προβλήματα που είτε δεν μπορούμε να τα επιλύσουμε με τις μέχρι τώρα γνώσεις μας, όπως η ακριβής πρόβλεψη των σεισμών, η γήρανση του ανθρώπου, η ανακάλυψη εξωγήινων πολιτισμών και η επικοινωνία μαζί τους,
- Προβλήματα που είτε έχει αποδειχθεί ότι δεν μπορούμε να τα επιλύσουμε, όπως ο τετραγωνισμός του κύκλου με κανόνα και διαβήτη ή το ταξίδι στο παρελθόν.

3. Τι πρέπει να κάνουμε για να επιλύσουμε ένα πρόβλημα;

- Να καταγράψουμε τα **δεδομένα**, που είναι τα στοιχεία που μας είναι γνωστά και μπορούν να μας βοηθήσουν στη λύση του προβλήματος.
- Να εντοπίσουμε το ή τα **ζητούμενα**, που είναι η απάντηση σε μια ερώτηση, αυτό που ψάχνουμε.
- Και τέλος να **επιλύσουμε το πρόβλημα**, περιγράφοντας την διαδικασία μέσω της οποίας βρίσκουμε το ζητούμενο και επιτυγχάνουμε τον επιθυμητό στόχο.

4. Ας τα βάλουμε σε μια σειρά. Ποια βήματα ακολουθούμε για να επιλύσουμε ένα πρόβλημα;

1. Κατανόηση

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).

2. Ανάλυση

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

3. Επίλυση

Είναι η υλοποίηση της λύσης του προβλήματος με σαφείς και απλές στην διατύπωσή του εντολές.

5. Τι ονομάζουμε αλγόριθμο;

Οι οδηγίες που δίνουμε με λογική σειρά, ώστε να εκτελέσουμε μια εργασία ή να επιλύσουμε ένα πρόβλημα, συνθέτουν έναν **Αλγόριθμο**. Για παράδειγμα, οι οδηγίες για την κατασκευή ενός χαρταετού μπορεί να αποτελέσουν έναν αλγόριθμο.

Αλγόριθμο ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών - βημάτων, με σκοπό την επίλυση ενός προβλήματος.

6. Οι αλγόριθμοι εκτελούνται μόνο από τους υπολογιστές ;

Οι οδηγίες που δίνουμε με λογική σειρά, ώστε να εκτελέσουμε μια εργασία ή να επιλύσουμε ένα πρόβλημα, συνθέτουν έναν **Αλγόριθμο**. Για παράδειγμα, οι οδηγίες για την κατασκευή ενός χαρταετού μπορεί να αποτελέσουν έναν αλγόριθμο.

Επομένως, η απάντηση είναι όχι. Ένας αλγόριθμος μπορεί να εκτελεστεί από έναν άνθρωπο ή μια μηχανή (π.χ. φούρνος μικροκυμάτων).

7. Ιδιότητες ενός Αλγορίθμου

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.

Για έχουμε το επιθυμητό αποτέλεσμα ο αλγόριθμος πρέπει να έχει τις εξής ιδιότητες:

- 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
- Τέλος, πρέπει να είμαστε σίγουροι ότι, αν υλοποιήσουμε τον αλγόριθμο, κάποτε θα τελειώσει επιτυχάνοντας τον αρχικό σκοπό

8. Πρόγραμμα – προγραμματισμός – προγραμματιστής

➤ 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**.

➤ Τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**.

9. Τι είναι οι Γλώσσες Προγραμματισμού;

➤ Οι γλώσσες που «καταλαβαίνουν» οι υπολογιστές είναι τεχνητές γλώσσες που ονομάζονται **γλώσσες προγραμματισμού**.

- ☞ Όταν ο αλγόριθμος πρέπει να υλοποιηθεί από έναν υπολογιστή ο προγραμματιστής:
- μεταφράζει τον αλγόριθμο σε μία **γλώσσα προγραμματισμού**
 - κάθε **εντολή** του αλγόριθμου μεταφράζεται σε μία ή περισσότερες **εντολές** της γλώσσας προγραμματισμού που χρησιμοποιούμε.

10. Μερικά παραδείγματα Γλωσσών Προγραμματισμού

1. Γλώσσα μηχανής

Οι εντολές διατυπώνονται στην γλώσσα του υπολογιστή. Είναι δηλαδή, ακολουθίες από 0 και 1.

2. Σύγχρονες γλώσσες προγραμματισμού

Οι σύγχρονες γλώσσες προγραμματισμού είναι πιο κοντά στις φυσικές γλώσσες.

Μερικές γνωστές γλώσσες προγραμματισμού Visual Basic, Java, Pascal, C++, Fortran, Cobol, Logo κ.α.

11. Ποια είναι τα χαρακτηριστικά των Γλωσσών Προγραμματισμού;

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

1. το αλφάβητο,
2. το λεξιλόγιο και
3. το συντακτικό.

1. Το **αλφάβητο** μιας γλώσσας προγραμματισμού είναι το σύνολο των χαρακτήρων που χρησιμοποιούνται από τη γλώσσα. (A,...Z, a,...z, A...Ω, α...ω, 0-9, +, *, /, (,) κλπ)

2. 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), ώστε να μπορούμε να το μάθουμε εύκολα.
Π.Χ. Logo : μπροστά, πίσω, δείξε, κάνε κ.α.

3. 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.

12. Τι είναι το ολοκληρωμένο προγραμματιστικό περιβάλλον;

Είναι ένα φιλικό περιβάλλον που μάς προσφέρουν οι σύγχρονες γλώσσες προγραμματισμού, έτσι ώστε γρήγορα να αναπτύσσουμε τα προγράμματά μας. Ένα ολοκληρωμένο περιβάλλον προγραμματισμού αποτελείται από διάφορα εργαλεία που βοηθάνε

τον προγραμματιστή να γράψει και να διορθώσει το πρόγραμμά του.

13. Ποια εργαλεία περιλαμβάνει ένα ολοκληρωμένο προγραμματιστικό περιβάλλον;

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

14. Ποιες είναι οι κατηγορίες προγραμμάτων μεταφραστών;

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 από τον επεξεργαστή του υπολογιστή.
- ✓ Αντίθετα οι **διερμηνείς (interpreters)** ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία.

15. Ποια λάθη μπορεί να συμβούν κατά τον προγραμματισμό;

Δύο κατηγορίες λαθών γίνονται από τους προγραμματιστές στη διαδικασία ανάπτυξης ενός προγράμματος :

➤ Τα **συντακτικά λάθη**

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 και

➤ Τα **λογικά λάθη**

Δεν πρέπει να ξεχνάμε ότι ο υπολογιστής εκτελεί πιστά, όποιες συντακτικές ορθές εντολές και αν του δώσουμε. 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.