

ΕΥΑΓΓΕΛΟΣ Χ. ΖΙΟΥΛΑΣ

Καθηγητής Πληροφορικής



ΚΕΦΑΛΑΙΟ
1

Η ΕΝΝΟΙΑ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ



ΠΡΟΒΛΗΜΑ

Είναι ένα ζήτημα που τίθεται προς επίλυση, μια κατάσταση η οποία πρέπει να αντιμετωπιστεί. Η λύση του όμως δεν είναι γνωστή ούτε προφανής.

- Προβλήματα αντιμετωπίζουμε καθημερινά είτε στο προσωπικό είτε στον κοινωνικό μας χώρο π.χ.

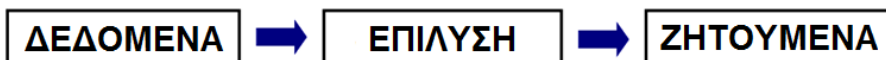
- η εύρεση της σύντομότερης διαδρομής για το σχολείο,
- η οργάνωση μιας σχολικής εκδρομής,
- η τακτοποίηση των βιβλίων μας σε μία βιβλιοθήκη
- η επίλυση μιας πρωτοβάθμιας ή δευτεροβάθμιας εξίσωσης
- η εύρεση της περιμέτρου και του εμβαδού ενός κύκλου

Τα προβλήματα που συναντάμε στο σχολείο είναι συνήθως **υπολογιστικά** και προκειμένου να λυθούν απαιτούν μια σειρά από **λογικές σκέψεις** (συγκρίσεις) ή **μαθηματικές πράξεις** (προσθέσεις, αφαιρέσεις, πολλαπλασιασμούς, διαιρέσεις).

- Ωστόσο, υπάρχουν και προβλήματα τα οποία είναι συνήθως πιο πολύπλοκα στην επίλυση τους π.χ.
 - Η ρύπανση της ατμόσφαιρας, η τρύπα του όζοντος
 - το ενεργειακό πρόβλημα, η εξερεύνηση του διαστήματος
 - ο υποσιτισμός, η ανεργία, η οικονομική κρίση κλπ.

ΣΤΟΙΧΕΙΑ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Κάθε πρόβλημα περιέχει τρία βασικά στοιχεία: τα **δεδομένα**, την **επίλυση** του και τα **ζητούμενα**.



Δεδομένα είναι όλα τα στοιχεία που είναι γνωστά και αδιαμφισβήτητα μέσα στο πρόβλημα και τα οποία αποτελούν την βάση επίλυσης του προβλήματος.

Επίλυση είναι η διαδικασία με την οποία ένας επιλυτής συγκεντρώνει τα δεδομένα, τα συνδυάζει και τα επεξεργάζεται με έναν κατάλληλο τρόπο έτσι ώστε να επιτύχει τον στόχο του προβλήματος, που είναι η εύρεση του ζητουμένου.

Ζητούμενα ή αλλιώς **Πληροφορίες** είναι τα στοιχεία που ψάχνουμε να βρούμε μέσα σε ένα πρόβλημα ώστε να βγούμε από την προβληματική κατάσταση. Είναι στοιχεία έρευνας και αναζήτησης. Οι πληροφορίες προκύπτουν από την επεξεργασία των δεδομένων.

ΣΤΑΔΙΑ ΑΝΤΙΜΕΤΩΠΙΣΗΣ ΠΡΟΒΛΗΜΑΤΟΣ

ΚΑΤΑΝΟΗΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ

Είναι η σωστή και πλήρης αποσαφήνιση των **δεδομένων** και των **ζητούμενων** μέσα σε ένα πρόβλημα, καθώς και του περιβάλλοντος στο οποίο εντάσσεται το πρόβλημα.

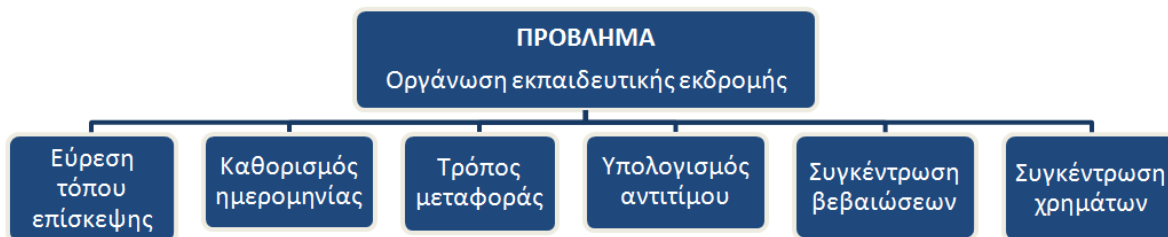
ΑΝΑΛΥΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ

Το αρχικό πρόβλημα λόγω της δυσκολίας του να αντιμετωπιστεί άμεσα, το διασπάμε σε **απλούστερα επιμέρους υποπροβλήματα**, που έχουν απλούστερη λύση.

ΕΠΙΛΥΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ

Είναι η **υλοποίηση της λύσης** του προβλήματος με σαφείς και απλές στην διατύπωσή τους **εντολές** (αλγόριθμος).

Παράδειγμα
Ανάλυσης
Προβλήματος



ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΒΛΗΜΑΤΩΝ

Η κατηγοριοποίηση των προβλημάτων γίνεται βάσει τριών κριτηρίων: τη **δυνατότητα επίλυσης**, τον **βαθμό δόμησης** και το **είδος επίλυσης**. Για κάθε ένα από τα τρία κριτήρια, διακρίνουμε τρεις επιμέρους κατηγορίες προβλημάτων.

Δυνατότητα Επίλυσης	Βαθμός Δόμησης	Είδος επίλυσης
<u>ΕΠΙΛΥΣΙΜΑ</u> Προβλήματα που η λύση τους είναι ήδη γνωστή και έχει διατυπωθεί. π.χ. ο υπολογισμός της περιμέτρου του κύκλου	<u>ΔΟΜΗΜΕΝΑ</u> Προβλήματα που η επίλυσή τους προέρχεται από μία αυτοματοποιημένη διαδικασία γνωστών βημάτων π.χ. η επίλυση μιας πρωτοβάθμιας ή δευτεροβάθμιας εξίσωσης	<u>ΑΠΟΦΑΣΗΣ</u> Προβλήματα των οποίων η απόφαση που πρόκειται να ληφθεί σαν λύση, απαντά σε κάποιο ερώτημα κλειστού τύπου π.χ είναι ο αριθμός 5 άρτιος; Ναι ή όχι;
<u>ΑΝΟΙΚΤΑ</u> Προβλήματα που η λύση τους δεν έχει ακόμα βρεθεί, αλλά δεν έχει αποδειχτεί και ότι δεν δέχονται λύση π.χ. ύπαρξη ζωής σε άλλους πλανήτες κλπ	<u>ΗΜΙΔΟΜΗΜΕΝΑ</u> Προβλήματα που η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επίλογής π.χ. η επιλογή του μεταφορικού μέσου για ένα ταξίδι	<u>ΥΠΟΛΟΓΙΣΤΙΚΑ</u> Προβλήματα στα οποία απαιτείται μια σειρά υπολογισμών προκειμένου να δοθεί σαν απάντηση στο πρόβλημα ένας αριθμός π.χ. ο υπολογισμός μιας μαθηματικής πράξης
<u>ΑΛΥΤΑ</u> Προβλήματα για τα οποία έχει αποδειχτεί ότι δεν έχουν λύση π.χ. τετραγωνισμός του κύκλου, η αποφυγή του γήρατος κλπ	<u>ΑΔΟΜΗΤΑ</u> Προβλήματα που η λύση τους βασίζεται στην ανθρώπινη διαίσθηση και δεν ακολουθούν αυτοματοποιημένη διαδικασία επίλυσης π.χ. ο τρόπος της διεξαγωγής ενός πάρτι	<u>ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ</u> Προβλήματα στα οποία ζητάμε να βρούμε το βέλτιστο αποτέλεσμα, δηλαδή αυτό που ικανοποιεί με το καλύτερο δυνατό τρόπο τα δεδομένα π.χ. εύρεση της ταχύτερης (ή αργότερης) διαδρομής για να φτάσουμε σε ένα σημείο

ΤΙ ΕΙΝΑΙ ΑΛΓΟΡΙΘΜΟΣ

Λέξη που προέρχεται από το όνομα του Πέρση μαθηματικού *Mohammed ibn-Musa al-Khuwarizmi* (750 – 850 μ.Χ)

Αλγόριθμος είναι ένα πεπερασμένο σύνολο ξεχωριστών εντολών (βημάτων), διατυπωμένων με ακρίβεια και σαφήνεια, που έχει σκοπό την επίλυση του προβλήματος.

- Ο αλγόριθμος δεν συνδέεται μόνο με επιστημονικά προβλήματα, αλλά και με προσωπικά καθημερινά προβλήματα π.χ. η συνταγή μιας μαγειρικής όπως και η επίλυση μιας μαθηματικής πράξης.
- Τα βήματα που αποτελούν τον αλγόριθμο, ονομάζονται **εντολές** ή **οδηγίες** και πρέπει πάντα να τοποθετούνται με λογική σειρά.
- Αυτός που εκτελεί τις εντολές ενός αλγορίθμου ονομάζεται **εκτελεστής** και μπορεί να είναι ο άνθρωπος ή ο υπολογιστής.
- Όταν ο αλγόριθμος εκφράζεται με εντολές που είναι κατανοητές στον υπολογιστή, τότε ονομάζεται **πρόγραμμα**.

Παράδειγμα αλγορίθμου Δημιουργία γεύματος

1. Συγκέντρωσε τα υλικά
2. Προετοίμασε τα σκεύη
3. Παρασκεύασε το φαγητό
4. Ετοίμασε τη σαλάτα
5. Στρώσε το τραπέζι
6. Γευμάτισε το φαί
7. Καθάρισε το τραπέζι
8. Πλύνε τα πιάτα
9. Καθάρισε την κουζίνα

ΙΔΙΟΤΗΤΕΣ ΑΛΓΟΡΙΘΜΟΥ

Ένας αλγόριθμος για να είναι πλήρης και ολοκληρωμένος οφείλει να έχει τις παρακάτω ιδιότητες:

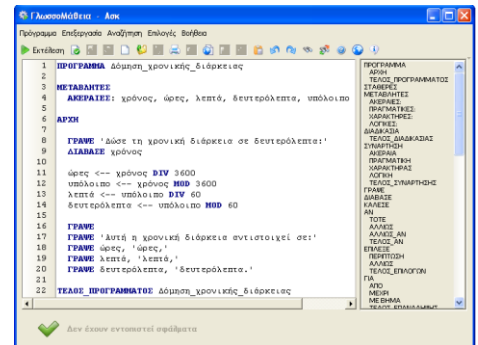
ΣΑΦΗΣ	Κανένα βήμα δεν αφήνει αμφιβολίες για την εκτέλεση του
ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΣ	Δίνει αποτελέσματα σε πεπερασμένο χρόνο
ΠΡΑΓΜΑΤΟΠΟΙΗΣΙΜΟΣ	Είναι εκτελέσιμος από τον αντίστοιχο εκτελεστή
ΤΑΧΥΣ	Χρησιμοποιεί όσο το δυνατό λιγότερες εντολές
ΟΙΚΟΝΟΜΙΚΟΣ	Απαιτεί όσο το δυνατό λιγότερα μέσα (πόρους)
ΓΕΝΙΚΟΣ	Μπορεί να επιλύει ένα σύνολο συγγενών προβλημάτων

ΠΡΟΓΡΑΜΜΑ

- Ονομάζεται ένας αλγόριθμος που είναι διατυπωμένος κατά τέτοιο τρόπο, ώστε οι **εντολές** του να είναι **κατανοητές από τον υπολογιστή**.
- Άρα, ενώ κάθε πρόγραμμα είναι ένας αλγόριθμος, κάθε αλγόριθμος δεν σημαίνει ότι αποτελεί αναγκαστικά και πρόγραμμα.
- Για να γίνει ένας αλγόριθμος κατανοητός από τον υπολογιστή, οφείλει να σέβεται μια σειρά από ορθογραφικούς και συντακτικούς κανόνες.

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Είναι μια **τεχνητή γλώσσα επικοινωνίας** που χρησιμοποιεί ο άνθρωπος για να επικοινωνήσει με τον υπολογιστή και να εκφράσει κατανοητά τους αλγορίθμους του.
- Όπως και κάθε φυσική γλώσσα, μια γλώσσα προγραμματισμού διαθέτει τους δικούς της **συντακτικούς** και **εννοιολογικούς κανόνες** για την σύνταξη των εντολών της.
- Κάθε γλώσσα έχει το δικό της **συντακτικό**, **λεξιλόγιο** και **αλφάβητο**.
- Σήμερα υπάρχουν εκατοντάδες γλώσσες προγραμματισμού, χωρίς αυτό να σημαίνει ότι όλες είναι αναγκαίες στον προγραμματιστή π.χ. *Pascal, Basic, Fortran, Cobol, Logo, C, C++, Java, Python, Perl, Lisp, Visual Basic, C#, SQL, Ada, Prolog* κλπ.
- Κάθε γλώσσα προγραμματισμού είναι φτιαγμένη για να υλοποιεί συγκεκριμένου τύπου εφαρμογές.



ΕΞΕΛΙΞΗ ΓΛΩΣΣΩΝ – ΚΑΤΗΓΟΡΙΕΣ

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

- Είναι η **μητρική γλώσσα** του υπολογιστή (αφού μόνο αυτήν καταλαβαίνει).
- Οι εντολές της γράφονται σε δυαδική μορφή ως ακολουθίες από **bits** (0 και 1).
- Αποτελεί αρκετά δύσκολο και σύνθετο τρόπο προγραμματισμού.
- Ένα πρόγραμμα σε δυαδική μορφή εκτελείται άμεσα από τον υπολογιστή χωρίς την ανάγκη ιδιαίτερης μετάφρασης.

```
10101000 00001010
10001100 00000001
00111100
01010001 00000001
01000011 00000001
11000000 11111010
10001100 00000100
11111111
```

ΣΥΜΒΟΛΙΚΗ ΓΛΩΣΣΑ (ASSEMBLY)

- Είναι πιο ανθρώπινη γλώσσα προγραμματισμού που χρησιμοποιεί λέξεις αντί αριθμούς.
- Η σύνταξη και ο έλεγχος των προγραμμάτων γίνεται ευκολότερα.
- Είναι επίσης γνωστή και ως **γλώσσα χαμηλού επιπέδου** εξαρτάται από τον τύπο (αρχιτεκτονική) του υπολογιστή.
- Αυτό σημαίνει ότι δεν έχει μεταφερσιμότητα και άρα ένα πρόγραμμα σε συμβολική γλώσσα είναι εκτελέσιμο μόνο στον υπολογιστή που γράφεται.
- Ένα πρόγραμμα Assembly για να γίνει κατανοητό στον υπολογιστή, πρέπει να πρώτα να μεταφραστεί σε γλώσσα μηχανής.
- Το πρόγραμμα που αναλαμβάνει την μετάφραση ονομάζεται **συμβολομεταφραστής (assembler)**.

```
INDEX=$01
SUM=$02
LDA #10
STA INDEX
CLA
LOOP ADD INDEX
DEC INDEX
BNE LOOP
STA SUM
BRK
```

ΓΛΩΣΣΕΣ ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ

- Μοιάζουν με τις ανθρώπινες γλώσσες και για αυτό είναι πιο οικίες στους προγραμματιστές.
- Διαθέτουν μικρό λεξιλόγιο από αγγλικές λέξεις, εύκολη σύνταξη και αρκετά μαθηματικά σύμβολα.
- Κάθε εντολή γλώσσας υψηλού επιπέδου αντιστοιχίζεται και μεταφράζεται σε πολλές εντολές γλώσσας μηχανής.
- Για την μετάφρασή τους σε γλώσσα μηχανής, οι γλώσσες υψηλού επιπέδου διαθέτουν μεταφραστικά προγράμματα όπως: **μεταγλωττιστές (compilers)** και **διερμηνευτές (interpreters)**.
- Σε αντίθεση με τις γλώσσες χαμηλού επιπέδου, οι γλώσσες αυτές έχουν **μεταφερσιμότητα**, δηλαδή είναι ανεξάρτητες από την αρχιτεκτονική του υπολογιστή.

```
sum = 0
FOR index=1 TO 10
  sum=sum+index
NEXT index
END
```

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Όπως όλες οι φυσικές γλώσσες, κάθε τεχνητή γλώσσα προγραμματισμού περιέχει:

ΑΛΦΑΒΗΤΟ	Είναι το σύνολο των χαρακτήρων – συμβόλων που χρησιμοποιεί η γλώσσα για να εκφράσει τις εντολές της (γράμματα, ψηφία, σημεία στίξης, μαθηματικά σύμβολα)
ΛΕΞΙΛΟΓΙΟ	Είναι το σύνολο των λέξεων που αναγνωρίζει η γλώσσα ως αποδεκτές εντολές της και έχουν κάποια καθορισμένη σημασία.
ΣΥΝΤΑΚΤΙΚΟ	Είναι το σύνολο των κανόνων της γλώσσας που καθορίζει την ορθογραφία και τον τρόπο σύνδεσης των λέξεων, ώστε ο υπολογιστής να τις αντιλαμβάνεται ως εντολές. Αποτελεί την γραμματική της γλώσσας.

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

- Είναι το λογισμικό περιβάλλον το οποίο επιτρέπει στον προγραμματιστή να συντάσσει, να μεταγλωττίζει και να εκτελεί τα προγράμματα του.
- Το προγραμματιστικό περιβάλλον παρέχει στον προγραμματιστή δύο βασικά βοηθητικά εργαλεία:
 - ▶ **ΚΕΙΜΕΝΟΓΡΑΦΟΣ (editor)**
Είναι το εργαλείο που χρησιμοποιούμε για να συντάξουμε και να διορθώσουμε τον κώδικα με τις εντολές του προγράμματος (λειτουργεί σαν επεξεργαστής κειμένου).
 - ▶ **ΜΕΤΑΦΡΑΣΤΗΣ**
Είναι το πρόγραμμα που μετατρέπει τις εντολές του προγράμματος από γλώσσα υψηλού επιπέδου (**πηγαίο πρόγραμμα**) σε γλώσσα μηχανής (**αντικείμενο πρόγραμμα**), κάνοντας ταυτόχρονα και έλεγχο σφαλμάτων.

ΚΑΤΗΓΟΡΙΕΣ ΜΕΤΑΦΡΑΣΤΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ (compilers)	ΔΙΕΡΜΗΝΕΥΤΕΣ (interpreters)
Ελέγχουν όλο το πρόγραμμα για λάθη κι αν δεν εντοπίσουν κανένα, τότε το μετατρέπουν σε γλώσσα μηχανής ώστε να εκτελεστεί. Σε περίπτωση που εντοπίσουν ένα συντακτικό λάθος, παρέχουν στον προγραμματιστή σχετικό μήνυμα λάθους που εξηγεί τη γραμμή εμφάνισης του λάθους και την αιτία πρόκλησης του προκειμένου να διορθωθεί.	Ελέγχουν μία – μία τις εντολές του προγράμματος κι αν είναι συντακτικά ορθή τότε την μετατρέπουν σε γλώσσα μηχανής και την εκτελούν. Σε περίπτωση που εντοπίσουν ένα συντακτικό λάθος, παρέχουν στον προγραμματιστή σχετικό μήνυμα λάθους και διακόπτουν την διαδικασία της μετάφρασης και εκτέλεσης του προγράμματος αυτόματα.

ΣΤΑΔΙΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΚΑΤΗΓΟΡΙΕΣ ΛΑΘΩΝ

Κατά τον προγραμματισμό εμφανίζονται δύο βασικές κατηγορίες λαθών που συνήθως κάνει ένας προγραμματιστής: τα συντακτικά λάθη και τα λογικά λάθη.

ΣΥΝΤΑΚΤΙΚΑ (Syntax Errors)

Είναι λάθη που **οφείλονται σε παραβίαση των γραμματικών κανόνων** της γλώσσας, δηλαδή λάθη που προκύπτουν από παραβίαση στη σύνταξη και την ορθογραφία των εντολών.

Τα συντακτικά λάθη **μπορούν να εντοπιστούν** από τον μεταγλωττιστή κατά την μετάφραση και ο χρήστης οφείλει να τα διορθώνει προκειμένου να εκτελέσει το πρόγραμμα.

ΛΟΓΙΚΑ (Runtime Errors)

Είναι λάθη που **οφείλονται σε εσφαλμένο αλγόριθμο** που κατασκεύασε ο προγραμματιστής για την λύση του προβλήματος.

Τα λογικά λάθη είναι πιο σύνθετα γιατί **δεν μπορούν να εντοπιστούν** από τον μεταγλωττιστή κατά τη φάση της μετάφρασης αλλά μόνο από τον ίδιο τον προγραμματιστή σε φάση εκτέλεσης.