Γ’ ΓΥΜΝΑΣΙΟΥ
Κεφάλαιο 1

Αλγόριθμοι - Προγραμματισμός

Σημειώσεις : Χρήστος Μουρατίδης

Κάντε κλικ για έναρξη…

Η έννοια του προβλήματος

Ορισμός

- **Πρόβλημα =** Μία κατάσταση που πρέπει να αντιμετωπίσουμε.

- Τα προβλήματα μπορούμε να τα χωρίσουμε σε δύο κατηγορίες <u>ανάλογα με την ποσότητα των υπολογισμών</u> που απαιτούνται για να φτάσουμε στη λύση τους:

Υπολογιστικά | **Μη Υπολογιστικά**

- **Υπολογιστικά =** Εννοούμε αυτά που για να φτάσουμε στη λύση τους κάνουμε μία σειρά αριθμητικών υπολογισμών (*π.χ. υπολογισμός εμβαδού τριγώνου*).
  Γι' αυτού του είδους τα προβλήματα οι Η/Υ είναι πολύ χρήσιμοι.

- **Μη Υπολογιστικά =** Εννοούμε αυτά που για να φτάσουμε στη λύση τους κάνουμε μία σειρά ενεργειών που εμπεριέχουν καθόλου ή σε μικρότερο βαθμό αριθμητικούς υπολογισμούς (π.χ. *οργάνωση μίας σχολικής εκδρομής, οργάνωση των ραντεβού ενός ιατρού*).
  Γι' αυτού του είδους τα προβλήματα οι Η/Υ μπορεί να είναι χρήσιμοι .

Είδη προβλημάτων

Ανάλογα με το βαθμό επίλυσής τους τα χωρίζουμε σε :

Επιλύσιμα

- Γι’ αυτά γνωρίζουμε ότι **λύνονται.** Η επίλυσή τους είναι εύκολη ή δύσκολη (π.χ. υπολογισμός εμβαδού κύκλου ή οργάνωση μίας σχολικής εκδρομής).

Ανοικτά

- Γι’ αυτά **δεν έχουμε βρει λύση αλλά παράλληλα δεν έχει αποδειχθεί ότι δεν λύνονται** (π.χ. ανακάλυψη εξωγήινου πολιτισμού, ανακάλυψη εμβολίου κατά του AIDS).

Άλυτα

- Γι’ **αυτά έχει αποδειχθεί ότι δεν υπάρχει λύση** (π.χ. ο τετραγωνισμός του κύκλου).

Κατανόηση προβλήματος - Επίλυση

Σημαίνει :

Να προσδιορίσουμε τα **δεδομένα** του προβλήματος

Να προσδιορίσουμε τα **ζητούμενα** του προβλήματος

Π.χ. Για τον υπολογισμό του εμβαδού ενός τριγώνου τα δεδομένα είναι η βάση και το ύψος. Το ζητούμενο είναι το εμβαδό.

Επίλυση = Η διαδικασία που ακολουθούμε ώστε ξεκινώντας από τα δεδομένα να φτάσουμε στη λύση (ζητούμενο) του προβλήματος.

Συνεπώς έχουμε :

Χώρος προβλήματος και Απλά-Σύνθετα προβλήματα

- Λέγεται και **«περιβάλλον προβλήματος»** = Είναι το πλαίσιο επίλυσης, δηλαδή ο χώρος στον οποίο ψάχνουμε να βρούμε τη λύση του.

  Π.χ. για την οργάνωση μίας σχολικής εκδρομής το πλαίσιο είναι το σχολικό περιβάλλον, για την ανακάλυψη εξωγήινου πολιτισμού το πλαίσιο είναι το διάστημα.

Τα προβλήματα μπορεί να είναι **απλά** ή **σύνθετα** ανάλογα με το <u>πόσο πολύπλοκη είναι η επίλυσή τους</u>

Απλά

- Η επίλυσή τους είναι απλή κι ενδεχομένως προφανής.

Σύνθετα

- Η επίλυσή τους δεν είναι γνωστή ή προφανής και πρέπει να ακολουθήσουμε μία αναλυτική διαδικασία για τη λύση τους.

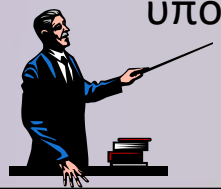
Ανάλυση προβλήματος

- Για **σύνθετα προβλήματα** πρέπει να ακολουθήσουμε μία **αναλυτική διαδικασία** για την επίλυσή τους.
- Αυτή περιλαμβάνει το «σπάσιμό» του σε μικρότερα κι απλούστερα **υπο-προβλήματα** που λύνονται το καθένα ευκολότερα.

- Π.χ. δείτε το παρακάτω παράδειγμα σχηματικά :

Ερωτήσεις κατανόησης

- Τί εννοούμε με τον όρο πρόβλημα;
- Ποιά θεωρούμε ως υπολογιστικά προβλήματα;
- Πόσα είδη προβλημάτων έχουμε ανάλογα με το βαθμό επίλυσης;
- Μπορείτε να αναφέρετε παραδείγματα από ανοικτά προβλήματα;
- Τί σημαίνει η κατανόηση του προβλήματος;
- Τί είναι το περιβάλλον του προβλήματος;
- Πώς επιλύουμε ένα σύνθετο πρόβλημα;
- Μπορείτε να αναλύσετε το πρόβλημα της διεξαγωγής των εκλογών 15μελούς σε υπο-προβλήματα;



Αλγόριθμοι - Προγραμματισμός

Ορισμός

- **Αλγόριθμος =** Ένα σύνολο οδηγιών (εντολών) που αν εκτελεστούν με ακρίβεια οδηγούν στην πραγματοποίηση μίας εργασίας ή τη λύση ενός προβλήματος.

- Παραδείγματα μπορεί να είναι :

  - Μία συνταγή μαγειρικής που περιγράφει τα βήματα που απαιτούνται για να φτιάξει κάποιος ένα φαγητό.
  - Η βήμα – βήμα λύση ενός μαθηματικού προβλήματος.
  - Οι οδηγίες που δίνουμε σε ένα ταξιδιώτη για να πάει κάπου.

Σε κάθε περίπτωση οι οδηγίες πρέπει να έχουν μία **λογική σειρά** ώστε να οδηγούν στη λύση του προβλήματος.

Ιδιότητες ενός αλγορίθμου

Εκτελεσιμότητα	Σαφήνεια	Πληρότητα	Αποτελεσματικότητα
• Οι **οδηγίες** πρέπει να είναι **εκφρασμένες με απλά λόγια** ώστε να είναι κατανοητές από αυτόν που θα τις εκτελέσει.	• Οι **οδηγίες** να είναι **σαφείς** και να μην επιδέχονται διπλή ερμηνεία.	• Να **προβλέπει κάθε πιθανό ενδεχόμενο** που μπορεί να προκύψει κατά την εκτέλεση των οδηγιών.	• Οι **οδηγίες να οδηγούν στο επιθυμητό αποτέλεσμα.**

Ιδιότητες ενός αλγορίθμου - Παράδειγμα

- Για παράδειγμα για να καθοδηγήσουμε κάποιον να εκτελέσει μία μαγειρική συνταγή μακαρονάδας πρέπει οι οδηγίες του αλγορίθμου μας :
  - Να είναι **απλές και κατανοητές** της μορφής :
    - Άνοιξε το μάτι της κουζίνας στο 2
    - Βάλε 3 λίτρα νερό στην κατσαρόλα κλπ…
  - Να είναι **σαφείς** : η οδηγία «βράσε τα μακαρόνια» είναι ασαφής. Αντίθετα, η οδηγία «βράσε για 10 λεπτά τα μακαρόνια» είναι σαφής.
  - Να είναι όσο το δυνατόν **πλήρεις** : Π.χ. να προβλέπει τι θα γίνει σε περίπτωση που κολλήσουν τα μακαρόνια…
  - Να είναι **αποτελεσματικές**, δηλαδή μετά το πέρας της εκτέλεσης των οδηγιών πρέπει να έχει φτιαχτεί η μακαρονάδα.

*Να έχετε υπόψη ότι οι οδηγίες πρέπει να είναι **πεπερασμένες,** δηλαδή συγκεκριμένες και να εκτελούνται μέσα σε συγκεκριμένα χρονικά όρια.*

Προγραμματισμός

Πρόγραμμα = Υλοποίηση του αλγορίθμου σε μία γλώσσα προγραμματισμού που είναι κατανοητή σε έναν υπολογιστή.

Δηλαδή, ένα **πρόγραμμα** περιλαμβάνει ένα **σύνολο εντολών** που δίνονται στον υπολογιστή με σκοπό να εκτελέσει μία συγκεκριμένη λειτουργία ή να υπολογίσει κάποιο επιθυμητό αποτέλεσμα.

- **Προγραμματισμός =** Είναι η εργασία σύνταξης ενός προγράμματος.

- **Προγραμματιστές =** Είναι τα άτομα που συντάσσουν ένα πρόγραμμα. Φυσικά, αυτά τα άτομα γνωρίζουν μία ή περισσότερες γλώσσες προγραμματισμού.

- Κάθε πρόγραμμα για να εκτελεστεί από τον υπολογιστή πρέπει πρώτα **οι εντολές του να φορτωθούν στη μνήμη RAM του**.
  Συνεπώς, όταν κάνετε διπλό κλικ σε ένα εικονίδιο προγράμματος ή το επιλέγετε από το μενού, αυτό καθυστερεί λίγο διότι «φορτώνονται» οι εντολές του στη μνήμη.

Γλώσσες προγραμματισμού

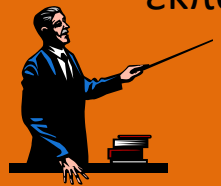
- Είναι **τεχνητές γλώσσες** που χρησιμοποιούμε για την επικοινωνία του ανθρώπου με τη μηχανή.

- Ενώ οι γλώσσες που μιλούμε μεταξύ μας οι άνθρωποι είναι **φυσικές γλώσσες** (ελληνικά, αγγλικά κλπ).

- Λέγονται και **γλώσσες προγραμματισμού υψηλού επιπέδου** διότι μοιάζουν με τη φυσική μας γλώσσα.

- Μερικά παραδείγματα τέτοιων γλωσσών είναι η **Basic**, **Pascal**, **Cobol**, **Fortran**, **C++**, **Java** , **Logo** κι αρκετές άλλες.

- Ο λόγος που υπάρχουν αρκετές γλώσσες οφείλεται στο γεγονός ότι **καθεμία έχει φτιαχτεί για να λύνει συγκεκριμένο τύπο προβλημάτων**. Πχ. η Fortran για επιστημονικά-μαθηματικά προβλήματα, η Cobol για εμπορικά-λογιστικά, η C++ κυρίως για λειτουργικά συστήματα, η Java για δικτυακές εφαρμογές, η Logo για εκπαίδευση-γραφικά. Κάποιες, μάλιστα, είναι **γενικής χρήσης** (Basic, Pascal).



Χαρακτηριστικά Γλωσσών προγραμματισμού

Όπως και οι φυσικές γλώσσες έχουν παρόμοια χαρακτηριστικά τα οποία είναι :

Αλφάβητο → Είναι το **σύνολο των χαρακτήρων** που χρησιμοποιεί η γλώσσα. (πχ. a, b. c. ! κ.α.)

Λεξιλόγιο → Είναι το **σύνολο των αποδεκτών λέξεων** που αναγνωρίζει η γλώσσα (πχ. στη Basic η λέξη print είναι αποδεκτή ενώ η rpint όχι),

Συντακτικό → Είναι το **σύνολο των κανόνων της γλώσσας** που καθορίζει πώς συνδέουμε τις λέξεις ώστε να σχηματιστεί μία σωστή πρόταση (πχ. στη Basic η εντολή print x είναι συντακτικά σωστή ενώ η print x y όχι διότι οι λέξεις print x και y δεν συνδέονται σωστά μεταξύ τους.

Άρα, για να μάθει κάποιος τις εντολές μίας γλώσσας πρέπει να γνωρίζει πώς αυτές συντάσσονται.

Παράδειγμα εντολών

- Στο παρακάτω παράδειγμα βλέπετε ένα πρόγραμμα που υπολογίζει το άθροισμα δύο αριθμών σε Basic και Pascal, αντίστοιχα:

```
Input "Δώσε τον πρώτο αριθμό''; a
Input "Δώσε το δεύτερο αριθμό''; b
SUM = a + b
Print ''Το άθροισμα είναι''; SUM
```

Basic

```
Read('Δώσε τον πρώτο αριθμό', a);
Read('Δώσε το δεύτερο αριθμό', b);
SUM := a + b
Write('Το άθροισμα είναι', SUM)
```

Pascal

- Παρατηρήστε πόσο κοντά στην αγγλική γλώσσα είναι οι εντολές.
- Στην πρώτη εντολή της BASIC αν ξεχάσουμε το ερωτηματικό ; θα έχουμε κάνει συντακτικό λάθος.

Γλώσσα μηχανής

- Στα αρχικά χρόνια των υπολογιστών (δεκαετία ‘40) οι άνθρωποι επικοινωνούσαν με τον υπολογιστή με μία γλώσσα που είχε ως **αλφάβητο το 0 και 1**.
  Όλες οι εντολές μετατρέπονταν σε 0 και 1.

- **Πρόγραμμα σε γλώσσα μηχανής =** Ένα πρόγραμμα που **οι εντολές του είναι σε δυαδική μορφή**, δηλαδή έχουν μετατραπεί σε 0 και 1.

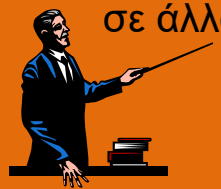
```
0000 0001
0000 0010
0000 0011
0000 0100
```

Εντολές σε γλώσσα μηχανής

- Η γλώσσα μηχανής είναι **γλώσσα χαμηλού επιπέδου** διότι είναι πιο κοντά στον υπολογιστή παρά στον άνθρωπο.

- **Πολύ δύσκολος ο προγραμματισμός** σε αυτό το επίπεδο.

- **Κάθε τύπος υπολογιστή** (με διαφορετικό επεξεργαστή) **έχει τη δική του γλώσσα μηχανής**. Έτσι, ένα πρόγραμμα που εκτελείται σε κάποιον Η/Υ πρέπει να ξαναγραφτεί για να εκτελεστεί σε άλλον.



Πηγαίο – Εκτελέσιμο - Μεταφραστής

Πηγαίο πρόγραμμα

Είναι το **πρόγραμμα που είναι γραμμένο σε γλώσσα προγραμματισμού υψηλού επιπέδου** π.χ. Basic, Logo.
Είναι κατανοητό από τον άνθρωπο κι όχι από τη μηχανή.

Εκτελέσιμο πρόγραμμα

Είναι το **πρόγραμμα που είναι σε μορφή 0 και 1, δηλαδή σε γλώσσα μηχανής.**
Είναι κατανοητό από τη μηχανή κι όχι από τον άνθρωπο.

Για να εκτελεστεί ένα πρόγραμμα από τον υπολογιστή πρέπει να μετατραπεί από πηγαίο σε εκτελέσιμο. Τη μετατροπή αυτή την κάνει ένα ειδικό πρόγραμμα που ονομάζεται **μεταφραστής.**

Πηγαίο → **Μεταφραστής** → **Εκτελέσιμο**

Συντακτικά – Λογικά λάθη

- **Ένα πρόγραμμα μπορεί να περιέχει λάθη**. Αυτά είναι δύο ειδών :

Συντακτικά λάθη = Πρόκειται για ***λάθη που παραβιάζουν τους κανόνες της γλώσσας (αλφάβητο, λεξιλόγιο, συντακτικό).*** Π.χ. κάπου ο προγραμματιστής ξέχασε να βάλει μία παρένθεση που απαιτείται.

Αυτού του είδους τα λάθη τα ανιχνεύει ο μεταφραστής κι ενημερώνει με κατάλληλα μηνύματα τον προγραμματιστή.

Λογικά λάθη = Πρόκειται για ***λάθη που αφορούν τη λογική επίλυσης του προβλήματος.*** Π.χ. το πρόγραμμα βγάζει άλλο αποτέλεσμα απ' αυτό που περιμένουμε.

Αυτού του είδους τα λάθη δεν ανιχνεύονται από το μεταφραστή. Την ανίχνευση πρέπει να την κάνει προσεκτικά ο προγραμματιστής και μερικές φορές είναι δύσκολη.

Είδη μεταφραστών

Μεταγλωττιστής (Compiler)

- Ο μεταγλωττιστής ελέγχει **συνολικά όλο το πηγαίο πρόγραμμα** για συντακτικά λάθη.
- Αν δεν βρει λάθη τότε παράγει το εκτελέσιμο.

Διερμηνέας (Interpreter)

- Ο διερμηνέας ελέγχει κι εκτελεί **μία – μία εντολή του πηγαίου προγράμματος.**
- Δηλαδή, κάθε εντολή μετατρέπεται σε **γλώσσα μηχανής (0 και 1) κι εκτελείται αμέσως εφόσον είναι σωστή.**

Προγραμματιστικό περιβάλλον

- Οι γλώσσες προγραμματισμού προσφέρουν ένα φιλικό περιβάλλον στο οποίο μπορούμε να συντάξουμε το πρόγραμμά μας.
- Τα **βασικά εργαλεία** που διαθέτει είναι :

Συντάκτης (editor)

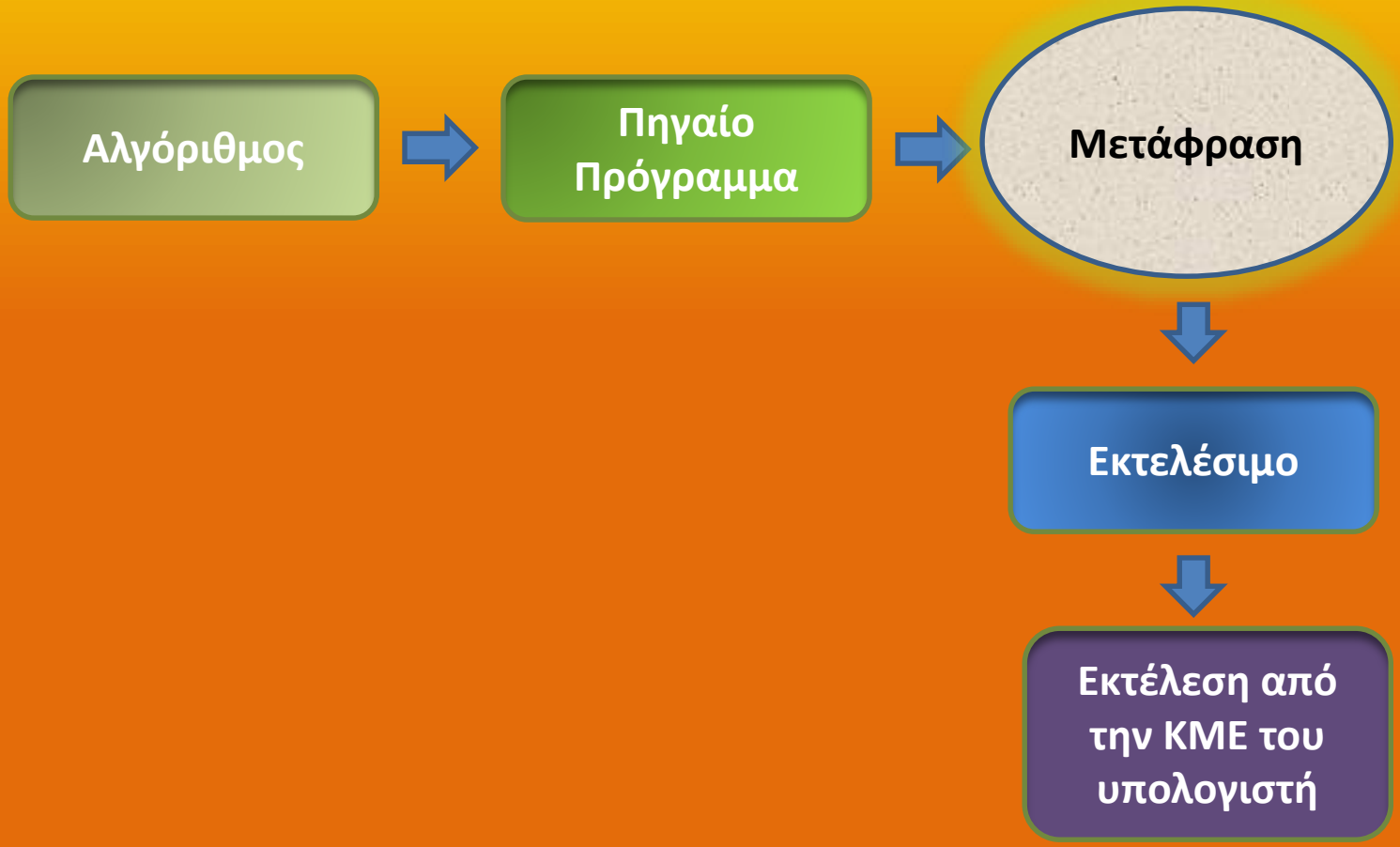
- Είναι ένας **εξειδικευμένος κειμενογράφος,** κατάλληλος για γραφή και διόρθωση του προγράμματός μας.

Μεταφραστής (compiler ή interpreter)

- Ειδικό πρόγραμμα που μετατρέπει τις εντολές μας σε μορφή που καταλαβαίνει ο επεξεργαστής, δηλαδή σε μία σειρά από 0 κα 1 (γλώσσα μηχανής).

Στάδια εκτέλεσης αλγορίθμου από την Κ.Μ.Ε.

- Μετά από όλα αυτά που είδαμε στα προηγούμενα συνοψίζουμε τη διαδικασία λύσης ενός προβλήματος μέσω του υπολογιστή :



Ερωτήσεις κατανόησης

- Τί είναι ο αλγόριθμος; Αναφέρατε 3 δικά σας παραδείγματα.
- Ποιές είναι οι ιδιότητες ενός αλγορίθμου;
- Τί είναι το πρόγραμμα;
- Τί είναι οι γλώσσες προγραμματισμού και ποιά τα χαρακτηριστικά τους;
- Τί είναι η γλώσσα μηχανής;
- Ποιό ονομάζουμε πηγαίο πρόγραμμα, ποιό εκτελέσιμο και τί ρόλο παίζει ο μεταφραστής;
- Ποιά είναι τα είδη των μεταφραστών και πού διαφέρουν;
- Τί είδους λάθη μπορεί να περιέχει ένα πρόγραμμα και ποιά είναι αυτά που απαιτούν περισσότερο κόπο να ανιχνευθούν;
- Ποιά είναι τα βασικά εργαλεία σε ένα προγραμματιστικό περιβάλλον;
- Ποιά τα στάδια εκτέλεσης ενός αλγορίθμου από τον επεξεργαστή (ΚΜΕ);