

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ**

Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ

Β' ΛΥΚΕΙΟΥ

Τόμος 2ος

Αθήνα 2014

**ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»**

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Πρόεδρος: Σωτήριος Γκλαβάς

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β'

Προϊστάμενος: Παύλος Φ. Μάραντος

ΣΥΓΓΡΑΦΕΙΣ:

**Δρ. Σπυρίδων Δουκάκης, Πληροφορικός,
Μαθηματικός, PIERCE-Αμερικανικό Κολλέγιο Ελλάδος
Χρήστος Δουληγέρης, Καθηγητής Τμήματος**

Πληροφορικής Πανεπιστημίου Πειραιώς

Δρ. Θεόδωρος Καρβουνίδης, Εκπαιδευτικός ΠΕ19

**Χρήστος Κοΐλιας, Καθηγητής Τμήματος Μηχανικών
Πληροφορικής Τ.Ε. ΤΕΙ Αθήνας**

**Δρ. Αθανάσιος Πέρδος, Πληροφορικός,
Φυσικός, Ελληνογαλλική Σχολή Καλαμαρί**

ΣΥΝΤΟΝΙΣΤΗΣ: Χρήστος Κοΐλιας

ΣΥΛΛΟΓΗ - ΕΠΕΞΕΡΓΑΣΙΑ ΥΛΙΚΟΥ:

Δημήτριος Κοτσιφάκος, Εκπαιδευτικός, ΠΕ 1708

ΚΡΙΤΕΣ-ΑΞΙΟΛΟΓΗΤΕΣ:

Παναγιώτης Βαρζάκας, Μέλος ΔΕΠ (συντονιστής)

Σοφία Τζελέπη, Σχολική Σύμβουλος, ΠΕ19

Πέτρος Ματζάκος, Εκπαιδευτικός, ΠΕ19

ΦΙΛΟΛΟΓΙΚΗ ΕΠΙΜΕΛΕΙΑ: Μαρία Κοΐλια

ΕΞΩΦΥΛΛΟ: Γιώργος Σκούφος

ΣΕΛΙΔΟΠΟΙΗΣΗ: Γιώργος Σκούφος

ΑΝΑΔΟΧΟΣ: ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ



Στουρνάρη 49Α, 106 82, Αθήνα

Τηλ. 210-38.45.594

Fax: 210-38.08.009

E-mail: contact@newtech-publications.gr

URL: www.newtech-pub.com

«ΔΗΜΙΟΥΡΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΥΛΙΚΟΥ ΓΙΑ ΤΑ ΝΕΑ ΜΑΘΗΜΑΤΑ ΤΟΥ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ» της Πράξης «ΝΕΟ ΣΧΟΛΕΙΟ (ΣΧΟΛΕΙΟ 21ου αιώνα) - ΝΕΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ» ΜΕ ΚΩΔ. ΟΠΣ 295450, των Αξόνων Προτεραιότητας 1, 2 και 3 - ΟΡΙΖΟΝΤΙΑ ΠΡΑΞΗ του ΕΠΙΧΕΙΡΗΣΙΑΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ», που συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση - Ευρωπαϊκό Κοινωνικό Ταμείο και από Εθνικούς Πόρους (ΕΣΠΑ 2007 - 2013).



**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης**



Η αξιολόγηση, η κρίση των προσαρμογών και η επιστημονική επιμέλεια του προσαρμοσμένου βιβλίου πραγματοποιείται από τη Μονάδα Ειδικής Αγωγής του Ινστιτούτου Εκπαιδευτικής Πολιτικής.

Η προσαρμογή του βιβλίου για μαθητές με μειωμένη όραση από το ΙΤΥΕ – ΔΙΟΦΑΝΤΟΣ πραγματοποιείται με βάση τις προδιαγραφές που έχουν αναπτυχθεί από ειδικούς εμπειρογνώμονες για το ΙΕΠ

**ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΒΙΒΛΙΟΥ
ΓΙΑ ΜΑΘΗΤΕΣ ΜΕ ΜΕΙΩΜΕΝΗ ΟΡΑΣΗ**

ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ

Προγραμματισμός

Στόχοι του κεφαλαίου αυτού είναι να μπορούν οι μαθητές:

- ✓ να δημιουργούν ευκρινές γνωστικό και οργανωμένο νοητικό σχήμα που να περιλαμβάνει τα είδη και τεχνικές προγραμματισμού, με βάση την πρότερη εμπειρία τους.
- ✓ να συνδυάζουν αλγοριθμικές δομές και δεδομένα/δομές δεδομένων για να δημιουργούν κώδικα/πρόγραμμα.
- ✓ να διαπιστώνουν ότι οι σημερινές εφαρμογές είναι αρκετά πολύπλοκες και η δημιουργία τους ακολουθεί συγκεκριμένα μοντέλα ανάπτυξης εφαρμογών λογισμικού που εξελίσσονται σε συγκεκριμένες φάσεις.



Προερωτήσεις

- Η δημιουργία του αλγορίθμου αρκεί για να επιλύσεις ένα πρόβλημα στον υπολογιστή;
- Γνωρίζεις κάποιες γλώσσες προγραμματισμού;
- Έχεις ακολουθήσει κάποια μεθοδολογία ή τεχνική για να επιλύσεις ένα πρόβλημα;



Προερωτήσεις

- Ποια νομίζεις ότι είναι η δουλειά του προγραμματιστή υπολογιστών;

2.3.1 Αναφορά σε γλώσσες προγραμματισμού και «Προγραμματιστικά Υποδείγματα»

2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Για να αναπαρασταθούν οι αλγόριθμοι σε μορφή κατανοητή από τον υπολογιστή αναπτύσσονται προγράμματα.

Πρόγραμμα είναι το σύνολο των εντολών που χρειάζεται να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος.

Η εργασία σύνταξης των προγραμμάτων σε κάποια γλώσσα προγραμματισμού ονομάζεται **προγραμματισμός** και τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**. Βασικό στοιχείο του προγράμματος, εκτός από τον αλγόριθμο που υλοποιεί, είναι τα δεδομένα και οι δομές δεδομένων που επεξεργάζεται.

Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστή). Ο υπολογιστής κάνει στοιχειώδεις ενέργειες σε ακολουθίες των δύο ψηφίων 0 και 1 (δυναδικά ψηφία, bits), αλλά αυτές τις ενέργειες τις εκτελεί με ασύλληπτη ταχύτητα. Συγκεκριμένα μπορεί να αποθηκεύει στη μνήμη τις ακολουθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει.

Αρχικά τα προγράμματα γράφονταν σε γλώσσα μηχανής, δηλαδή ακολουθίες δυαδικών ψηφίων, που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Ο συγκεκριμένος τρόπος γραφής προγραμμάτων είναι επίπονος και ελάχιστοι μπορούν να τον κατανοήσουν και να τον υλοποιήσουν, αφού απαιτεί βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή.



Εντολές σε γλώσσα μηχανής που καταχωρούν το άθροισμα των τιμών δύο θέσεων μνήμης σε μία άλλη.

0000001001011010

0000101001011110

0000011011011110

Η περιγραφή των παραπάνω εντολών είναι η εξής:

- Μετάφερε στον καταχωρητή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011010.
- Πρόσθεσε στο περιεχόμενο του καταχωρητή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011110.
- Μετάφερε και αποθήκευσε το περιεχόμενο του καταχωρητή στη θέση μνήμης με διεύθυνση 11011110.

Στη συνέχεια αναπτύχθηκαν οι **συμβολικές γλώσσες** οι οποίες κάνουν χρήση εντολών που αποτελούνται από συμβολικά ονόματα τα οποία αντιστοιχούν σε εντολές της γλώσσας μηχανής. Το έργο της μετάφρασης των εντολών σε γλώσσα μηχανής το αναλαμβάνει ένα ειδικό πρόγραμμα, ο **συμβολομεταφραστής (assembler)**.

Οι συμβολικές γλώσσες ήταν σαφώς μια εξέλιξη αλλά παραμένουν στενά συνδεδεμένες με την αρχιτεκτονική του κάθε υπολογιστή. Επιπλέον η έλλειψη εντολών σύνθετων λειτουργιών στις παραπάνω γλώσσες οδηγεί σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και να συντηρηθούν. Ακόμη, δεν είναι δυνατό να μεταφερθούν και να εκτελεστούν σε υπολογιστή διαφορετικής αρχιτεκτονικής.



Εντολές σε συμβολική γλώσσα που καταχωρούν το άθροισμα των τιμών δύο θέσεων μνήμης σε μία άλλη.

LDA B

ADD C

STA A

Η περιγραφή των παραπάνω εντολών είναι η εξής:

- **Μετάφερε στον καταχωρητή το περιεχόμενο της θέσης μνήμης με όνομα B.**
- **Πρόσθεσε στο περιεχόμενο του καταχωρητή το περιεχόμενο της θέσης μνήμης με όνομα C.**
- **Μετάφερε και αποθήκευσε το περιεχόμενο του καταχωρητή στη θέση μνήμης με όνομα A.**

Οι παραπάνω ανεπάρκειες και η προσπάθεια για καλύτερη επικοινωνία ανθρώπου – μηχανής οδήγησαν στην εμφάνιση των γλωσσών υψηλού επιπέδου. Σε σχέση με τις συμβολικές γλώσσες στις γλώσσες υψηλού επιπέδου:

- **είναι φυσικότερος και πιο ανθρώπινος ο τρόπος έκφρασης των προβλημάτων.**
- **υπάρχει δυνατότητα μεταφοράς, «μεταφερσιμότητα» δηλαδή, εκτέλεσης των προγραμμάτων σε οποιοδήποτε υπολογιστή.**

- είναι εύκολη η εκμάθηση, η διόρθωση των λαθών και η συντήρηση των προγραμμάτων.

Έτσι αναπτύχθηκαν γλώσσες όπως οι ακόλουθες:

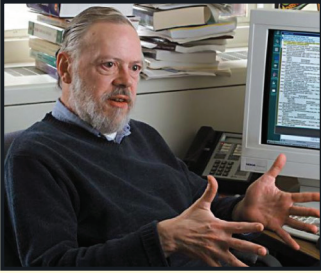
- **FORTRAN** (FORmula TRANslation, Μετάφραση Τύπων). Το 1957 η IBM ανέπτυξε την πρώτη γλώσσα υψηλού επιπέδου. Αναπτύχθηκε ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων.
- **COBOL** (COmmon Business Oriented Language, Κοινή Γλώσσα Προσανατολισμένη στις Επιχειρήσεις). Κατάλληλη για ανάπτυξη εμπορικών και γενικά διαχειριστικών εφαρμογών. Χρησιμοποιείται από επιχειρήσεις και από τη δημόσια διοίκηση.
- **LISP** (LISt Processor, Επεξεργαστής Λίστας). Συναρτησιακή γλώσσα η οποία προσανατολίζεται σε χειρισμό λιστών από σύμβολα. Χρησιμοποιείται στο χώρο της τεχνητής νοημοσύνης, σε έμπειρα συστήματα, παιχνίδια, επεξεργασία φυσικών γλωσσών.
- **PROLOG** (PROgramming in LOGic, Λογικός Προγραμματισμός). Η γλώσσα PROLOG χρησιμοποιεί μεθόδους λογικής για να αναπαραστήσει τη γνώση και να επιλύσει προβλήματα. Χρησιμοποιείται όπως και η LISP στο χώρο της τεχνητής νοημοσύνης, σε έμπειρα συστήματα, παιχνίδια, επεξεργασία φυσικών γλωσσών.
- **BASIC** (Beginner's All Purpose Symbolic Instruction Code, Συμβολικός Κώδικας Εντολών Γενικής Χρήσης για Αρχάριους). Γλώσσα που αναπτύχθηκε για την εκπαίδευση αρχαρίων στον προγραμματισμό.

Σχεδιάστηκε για να γράφονται σύντομα προγράμματα τα οποία εκτελούνται με τη βοήθεια διερμηνευτή. Η ανάπτυξη των μικροϋπολογιστών και η τυποποίησή της από τη Microsoft, την καθιέρωσε ως πρότυπο για ανάπτυξη εφαρμογών σε προσωπικούς υπολογιστές.

- **PASCAL.** Είναι μια γλώσσα γενικής χρήσης. Διέπεται από τις αρχές του δομημένου προγραμματισμού. Γνώρισε τεράστια εξάπλωση, και επηρέασε την ανάπτυξη άλλων γλωσσών όπως η ADA.
- **C** και η μετεξέλιξη της **C++**. Γλώσσα η οποία δημιουργήθηκε στα εργαστήρια BELL και χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος UNIX. Είναι γλώσσα με ισχυρά χαρακτηριστικά. Η **C++** είναι γλώσσα αντικειμενοστρεφούς προγραμματισμού.
- **JAVA.** Γλώσσα αντικειμενοστρεφής που αναπτύχθηκε από τη SUN με σκοπό την ανάπτυξη εφαρμογών για το διαδίκτυο.

Εντολή στη γλώσσα **BASIC** που καταχωρεί το άθροισμα των τιμών δύο μεταβλητών **B** και **C** στη μεταβλητή **A**.

A = B + C



Εικόνα 2.25

Ντένις Ρίτσι (Dennis Ritchie). Ο δημιουργός της γλώσσας προγραμματισμού C και ο βασικός συντελεστής στην ανάπτυξη του λειτουργικού συστήματος UNIX.

«Το UNIX είναι βασικά ένα απλό λειτουργικό σύστημα, πρέπει όμως να είσαι ιδιοφυΐα για να καταλάβεις την απλότητά του».

Η εμφάνιση των γραφικών περιβαλλόντων εργασίας δημιούργησε την ανάγκη για προγράμματα που να εκμεταλλεύονται τον γραφικό τρόπο επικοινωνίας χρήστη – υπολογιστή. Έτσι γλώσσες όπως η BASIC, η C++, η PASCAL που είναι μεν κειμενικές, εξελίχθηκαν (Visual Basic, Visual C++, Delphi) ώστε να διαθέτουν και οπτικό περιβάλλον προγραμματισμού.



Σε ένα οπτικό περιβάλλον προγραμματισμού υπάρχει η δυνατότητα να δημιουργείται γραφικά ολόκληρο το περιβάλλον της εφαρμογής, όπως για παράδειγμα τα πλαίσια διαλόγου ή τα μενού.

Αναπτύχθηκαν όμως και γλώσσες όπως η SCRATCH – BYOB και η Google AppInvertor που είναι αποκλειστικά οπτικές γλώσσες προγραμματισμού (Visual

Programming Languages, VPL). Οι συγκεκριμένες δίνουν τη δυνατότητα στον προγραμματιστή να δημιουργήσει προγράμματα μέσα από το γραφικό χειρισμό προγραμματιστικών στοιχείων (αντί κειμένου).

Αρχικά υπήρχαν τα περιβάλλοντα Γραμμής Εντολών, όπου ο χρήστης είχε τη δυνατότητα να πληκτρολογεί τις εντολές και να τις βλέπει στην οθόνη ενώ στα γραφικά περιβάλλοντα εργασίας τα προγράμματα και οι πληροφορίες εμφανίζονται στην οθόνη με γραφικά και σχήματα.



Εικόνα 2.26
Λογότυπο της γλώσσας
Google AppInvertor.

Η χρήση των υπολογιστών σχεδόν σε όλες τις εκφάνσεις της ανθρώπινης δραστηριότητας δημιούργησε την ανάγκη για γλώσσες κατάλληλες στην επίλυση συγκεκριμένων προβλημάτων. Έτσι αναπτύχθηκαν γλώσσες όπως η **LOGO** ή η **GameMaker** για εκπαιδευτικούς σκοπούς, η **LabView** που χρησιμοποιείται από τους επιστήμονες και τους μηχανικούς στο σχεδιασμό, τον έλεγχο και τη δοκιμή καταναλωτικών προϊόντων κ.ά..

Οι γλώσσες υψηλού επιπέδου χαρακτηρίζονται ως γλώσσες τρίτης γενιάς ενώ οι συμβολικές ως δεύτερης γενιάς ή χαμηλού επιπέδου. Από την άλλη ένα πρόγραμμα σε γλώσσα μηχανής είναι κωδικοποιημένο σε γλώσσα πρώτης γενιάς.

Οι παραπάνω γενιές γλωσσών προγραμματισμού απευθύνονται μόνο σε προγραμματιστές και ο χρήστης δεν έχει τη δυνατότητα να επιφέρει αλλαγές σε κάποιο πρόγραμμα, προκειμένου να ικανοποιήσει μια νέα ανάγκη του. Σταδιακά όμως πολλές γλώσσες εφοδιάστηκαν με εργαλεία προγραμματισμού που αποκρύπτουν πολλές λεπτομέρειες από τις τεχνικές υλοποίησης και με αυτά ο χρήστης μπορεί να επιλύει μόνος του μικρά προβλήματα εφαρμογών. Αυτή η αυξανόμενη τάση απόκρυψης της αρχιτεκτονικής του υλικού και της τεχνικής του προγραμματισμού οδήγησε στις γλώσσες τέταρτης γενιάς. Η **SQL (Structured Query Language, Δομημένη Γλώσσα Ερωτοαποκρίσεων)** είναι μία γλώσσα τέταρτης γενιάς η οποία χρησιμοποιείται για την ανάκτηση και τη διαχείριση δεδομένων καθώς και την παραγωγή πληροφοριών σε σχεσιακές βάσεις δεδομένων.

```
SELECT ENAME, JOB, SAL  
FROM EMPLOYEES  
WHERE DEPTNO = 20  
AND SAL > 1000;
```

Με την ερώτηση αυτή σε SQL εκτελείται αναζήτηση στη

βάση δεδομένων EMPLOYES και επιστρέφει το όνομα, τη θέση και τον μισθό των υπαλλήλων της διεύθυνσης 20 που κερδίζουν πάνω από 1000 ευρώ.

2.3.1.2 Προγραμματιστικά Υποδείγματα

Αναφέρθηκε προηγουμένως ότι κάποιες γλώσσες ακολουθούν τον αντικειμενοστρεφή προγραμματισμό και άλλες είναι συναρτησιακές ή χρησιμοποιούν μεθόδους λογικής για να επιλύσουν προβλήματα. Η ανάπτυξη λοιπόν ενός προγράμματος σε κάποια γλώσσα προγραμματισμού βασίζεται σε ένα πρότυπο ή μία καθορισμένη μεθοδολογία.

Ως «Προγραμματιστικό Υπόδειγμα» εννοείται ένα πρότυπο ανάπτυξης προγραμμάτων, δηλαδή μία καθορισμένη μεθοδολογία με βάση την οποία αναπτύσσονται η δομή και τα στοιχεία του προγράμματος.

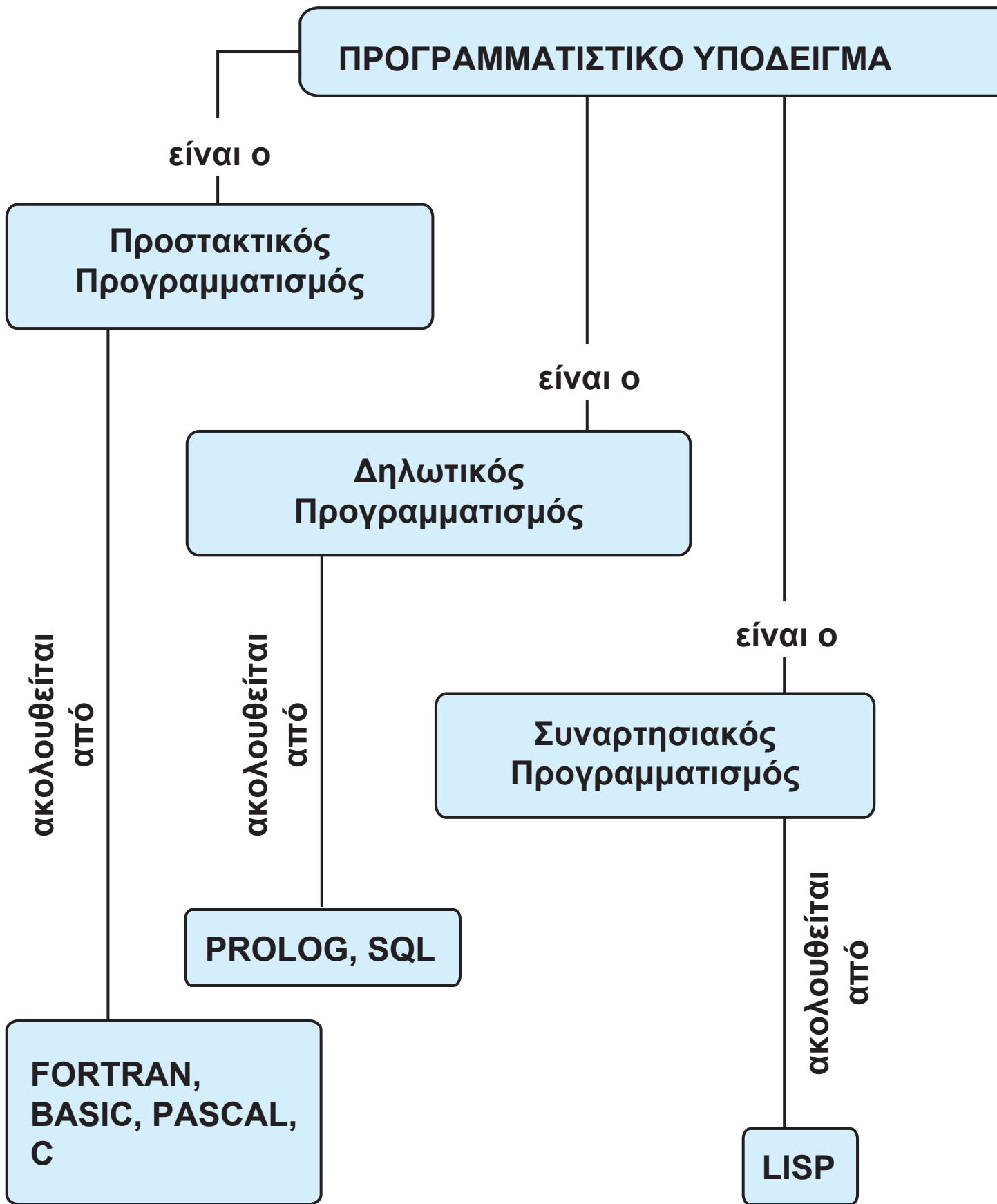
Οι δυνατότητες και οι μεθοδολογίες ανάπτυξης προγραμμάτων που παρέχει μία γλώσσα προγραμματισμού, καθορίζονται από το προγραμματιστικό υπόδειγμα που ακολουθεί. Υπάρχουν όμως γλώσσες που έχουν σχεδιαστεί να υποστηρίζουν περισσότερα από ένα υποδείγματα.

Τα κυριότερα προγραμματιστικά υποδείγματα είναι:

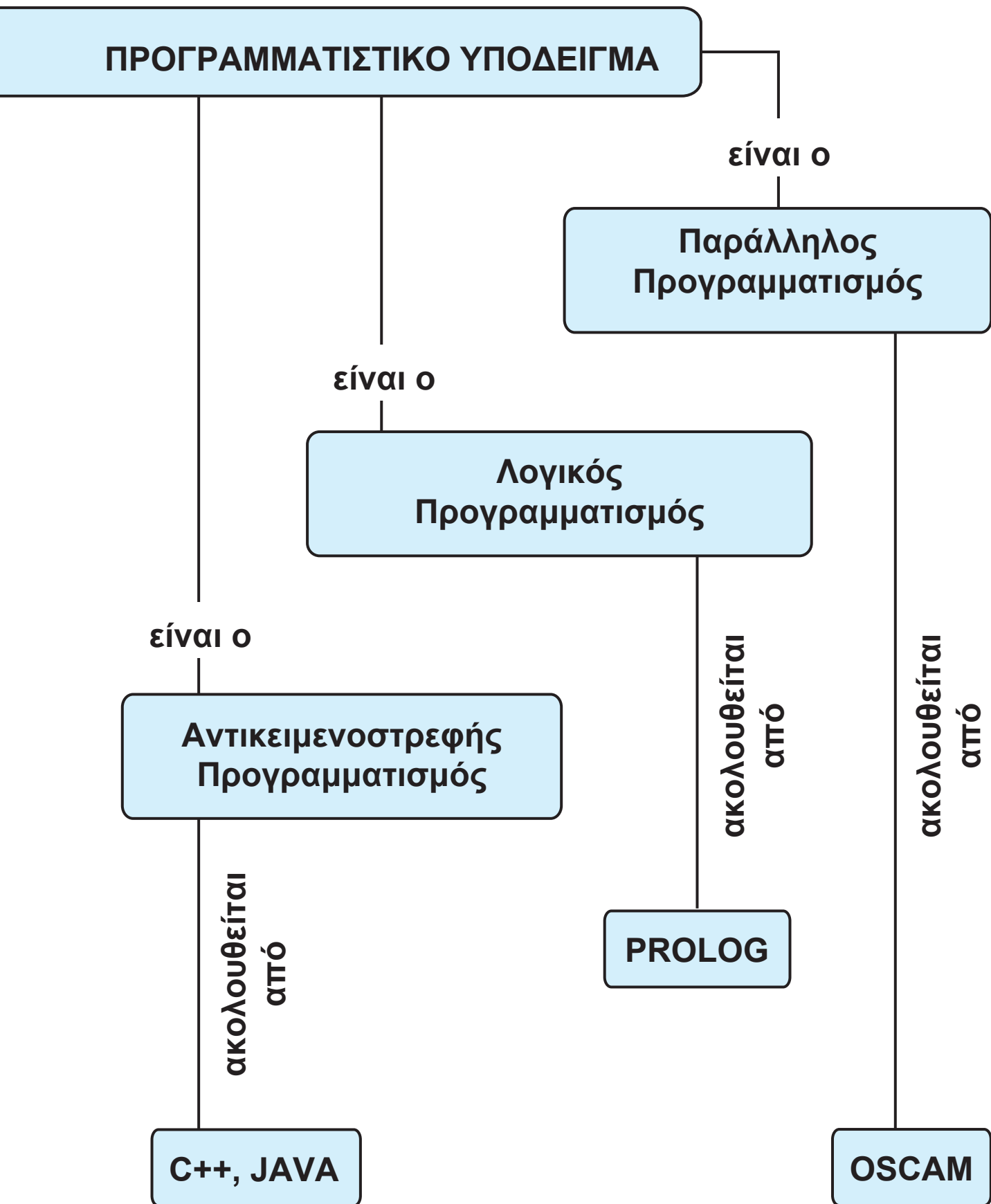
- Ο **προστακτικός προγραμματισμός** όπου τα προγράμματα αναπτύσσονται με απλές εντολές σε προστακτική (Διάβασε, Εμφάνισε, Επανάλαβε) που ζητούν από τον υπολογιστή να εκτελέσει συγκεκριμένες ενέργειες και να ακολουθήσει βήματα με μία λογική σειρά για να επιλύσει το πρόβλημα που έχει δοθεί. Γλώσσες, όπως η FORTRAN, η BASIC, η PASCAL, η C, ακολουθούν αυτό το υπόδειγμα.
- Ο **δηλωτικός προγραμματισμός** όπου, σε αντίθεση με τον προστακτικό προγραμματισμό, το πρόβλημα επιλύεται δηλώνοντας απλώς τις επιθυμητές ιδιότητες του αποτελέσματος. Το πρόγραμμα περιέχει λογικές εκφράσεις, ενώ κατά την εκτέλεσή του γίνεται έλεγχος για το ποιες ακριβώς ικανοποιούνται. Παραδείγματα γλωσσών που τον ακολουθούν είναι η PROLOG και η SQL.
- Ο **συναρτησιακός προγραμματισμός** επιλύει το πρόβλημα με τη χρήση μαθηματικών συναρτήσεων. Οι συναρτήσεις παράγουν αποτελέσματα με βάση τα δεδομένα εισόδου τους. Παράδειγμα συναρτησιακής γλώσσας είναι η LISP.
- Ο **αντικειμενοστρεφής προγραμματισμός** βασίζεται στην έννοια του αντικειμένου. Τα αντικείμενα δημιουργούνται από τις κλάσεις. Μία κλάση ορίζει τα χαρακτηριστικά και τη συμπεριφορά ενός τύπου αντικειμένου, λειτουργεί δηλαδή ως πρότυπο. Ένα αντικείμενο είναι μία δομή δεδομένων η οποία περιέχει τόσο τα δεδομένα (χαρακτηριστικά που την

περιγράφουν) όσο και τις διαδικασίες (μεθόδους) που επενεργούν σε αυτά. Τα αντικείμενα μπορούν να αλληλεπιδρούν μεταξύ τους. Αντικειμενοστραφείς γλώσσες είναι η C++ και η JAVA.

- Ο λογικός προγραμματισμός όπου τα προγράμματα είναι γραμμένα ως ένα σύνολο από προτάσεις σε μορφή λογικών εκφράσεων. Το συγκεκριμένο υπόδειγμα βασίζεται στα γεγονότα, στους κανόνες και στις ερωτήσεις και ακολουθείται κυρίως στο πεδίο της Τεχνητής Νοημοσύνης. Παράδειγμα γλώσσας που τον ακολουθεί είναι η PROLOG.
- Ο παράλληλος προγραμματισμός στον οποίο τα προγράμματα εκμεταλλεύονται την ύπαρξη υπολογιστών που διαθέτουν περισσότερους από έναν επεξεργαστές. Έτσι επιτυγχάνεται η αύξηση των υπολογιστικών επιδόσεων και η μείωση του χρόνου εκτέλεσης της εφαρμογής. Θα πρέπει όμως το πρόβλημα προς επίλυση να διαιρεθεί σε τμήματα που μπορούν να επιλυθούν παράλληλα. Μία γλώσσα που υποστηρίζει τον παράλληλο προγραμματισμό είναι η Occam.



Εικόνα 2.29 Προγραμματιστικά Υποδείγματα.



Εικόνα 2.29 Προγραμματιστικά Υποδείγματα.



Εικόνα 2.27

Η Άντα Λάβλεις (Ada Lovelace), κόρη του Λόρδου Βύρωνα, έγραψε το πρώτο πρόγραμμα υπολογιστή κατά τον 19ο αιώνα για την Αναλυτική Μηχανή του Τσαρλς Μπάμπατζ (Charles Babbage), πολύ πριν από την εμφάνιση ηλεκτρονικών υπολογιστών. Η γλώσσα προγραμματισμού ADA έχει ονομαστεί έτσι προς τιμήν της.



Εικόνα 2.28

Γκρέις Χόπερ (Grace Hopper). Αμερικανίδα καθηγήτρια μαθηματικών και αξιωματικός του αμερικανικού ναυτικού, η οποία ηγήθηκε της ομάδας που ανέπτυξε την πρώτη προηγμένη γλώσσα προγραμματισμού με προορισμό τον επιχειρηματικό κόσμο, την Common Business Oriented Language (COBOL)



Εικόνα 2.30

Νίκλαους Βιρθ (Niklaus Wirth)
Δημιουργός της γλώσσας PASCAL, η οποία ονομάστηκε έτσι προς τιμήν του Γάλλου επιστήμονα Μπλεζ Πασκάλ (Blaise Pascal).

2.3.1.3 Δομημένος Προγραμματισμός

Μία μεθοδολογία ανάλυσης, σχεδίασης και συγγραφής προγραμμάτων που αναπτύχθηκε και διαδόθηκε ευρύτατα στον χώρο της πληροφορικής είναι ο δομημένος προγραμματισμός (structured programming) ο οποίος χρησιμοποιεί

- την ιεραρχική σχεδίαση για την ανάπτυξη του αλγορίθμου που επιλύει το πρόβλημα.
- τον τμηματικό προγραμματισμό για τη σχεδίαση του προγράμματος και για τη δημιουργία των ενοτήτων του.
- τρεις βασικές συνιστώσες για τη συγγραφή των επιμέρους ενοτήτων που καθιστούν άσκοπη τη χρήση της εντολής GOTO (Πήγαινε).



Ο δομημένος προγραμματισμός προτάθηκε ως έννοια το 1966 από την ανάγκη να περιοριστεί η χρήση των εντολών GOTO (Πήγαινε).

Η ιεραρχική σχεδίαση ή ανάλυση «από πάνω προς τα κάτω» χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα. Αυτά είναι πιο εύκολο να επιλυθούν οδηγώντας τελικά στην επίλυση του αρχικού προβλήματος.

Ο τμηματικός προγραμματισμός υλοποιεί την ιεραρχική σχεδίαση όπου κάθε υποπρόβλημα αποτελεί ανεξάρτητη

ενότητα που ονομάζεται υποπρόγραμμα. Τα υποπρογράμματα πρέπει να χαρακτηρίζονται από όσο το δυνατόν μεγαλύτερο βαθμό ανεξαρτησίας και από υψηλό βαθμό συνεκτικότητας, δηλαδή να υλοποιούν μια μόνο συγκεκριμένη διαδικασία ή λειτουργία. Το πρόγραμμα τελικά είναι η σύνθεση όλων των υποπρογραμμάτων.



Υποπρόγραμμα είναι το τμήμα προγράμματος που επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα.

Τέλος, όλες οι ενότητες του προγράμματος μπορούν να αναπτυχθούν μόνο με τρεις βασικές συνιστώσες: τη **δομή ακολουθίας**, τη **δομή επιλογής** και τη **δομή επανάληψης** ή **συνδυασμό** τους. Οι τρεις αυτές συνιστώσες ομαδοποιούν τις εντολές και καθορίζουν τη σειρά εκτέλεσης τους. Οι εντολές λοιπόν του προγράμματος οργανώνονται σε μία δομή.

Σε αντίθεση, η χρήση της εντολής **GOTO** έχει ως αποτέλεσμα την αλλαγή της ροής του προγράμματος, δηλαδή της διακλάδωσης σε μία άλλη εντολή του προγράμματος εκτός από την επόμενη. Έτσι αυξάνεται η δυσκολία στην αρχική σχεδίαση της λύσης, στην παρακολούθηση και κατανόηση του προγράμματος και τέλος στη συντήρησή του.

Ο δομημένος προγραμματισμός αποτελεί λοιπόν μια μεθοδολογία σύνταξης προγραμμάτων που έχει σκοπό να βοηθήσει τον προγραμματιστή στην ανάπτυξη σύνθετων προγραμμάτων, να μειώσει τα λάθη, να διευκολύνει την κατανόηση των προγραμμάτων, τις διορθώσεις και τις αλλαγές σε αυτά.

Οι γλώσσες στις οποίες αναπτύσσονται προγράμματα με τη μεθοδολογία του δομημένου προγραμματισμού ακολουθούν κατά βάση το υπόδειγμα του προστακτικού προγραμματισμού.



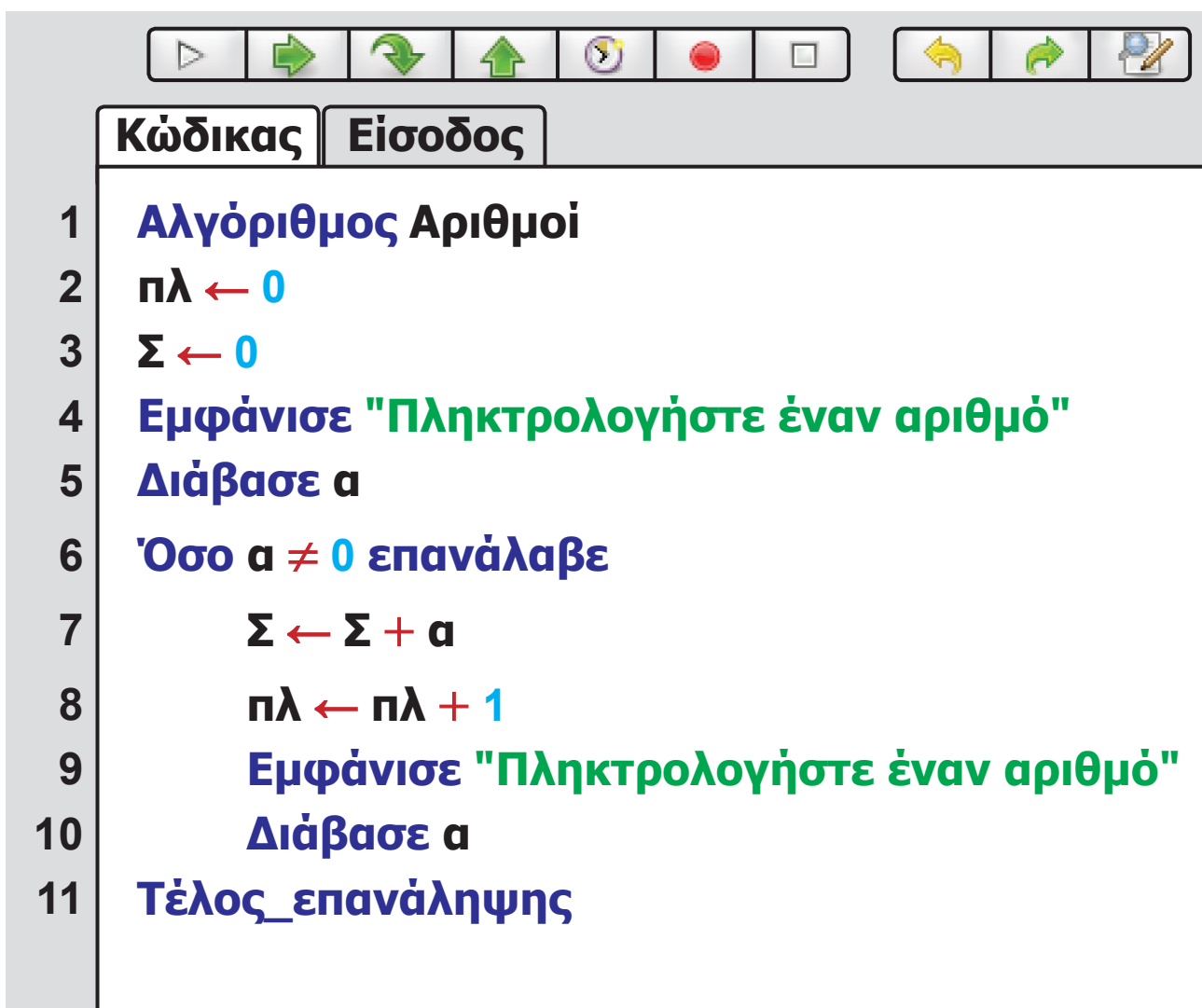
Στον δομημένο προγραμματισμό γίνεται χρήση υποπρογραμμάτων όπως οι διαδικασίες και οι συναρτήσεις.

Τα υποπρόγραμματα που αναπτύσσονται μπορούν να αποτελέσουν σύνθετες εντολές προς τον υπολογιστή.

Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μόνο μία έξοδο. Δηλαδή, όταν ενεργοποιείται, δέχεται κατά βάση κάποια δεδομένα και, αφού ολοκληρώσει τη λειτουργία του, επιστρέφει κάποια αποτελέσματα. Κατά τη διάρκεια όμως της εκτέλεσής του δεν εκτελείται καμία άλλη εντολή του προγράμματος πέρα από αυτές που συμπεριλαμβάνονται στο συγκεκριμένο υποπρόγραμμα.

Παράδειγμα 2.32. Να γραφεί πρόγραμμα το οποίο θα διαβάσει αριθμούς και θα υπολογίζει το μέσο όρο τους. Η διαδικασία εισαγωγής αριθμών θα σταματά αν διαβαστεί ο αριθμός 0, χωρίς να συνυπολογίζεται. Αν ο πρώτος αριθμός που θα διαβαστεί είναι το μηδέν, θα εκτυπώνει το μήνυμα «Δεν δόθηκε άλλο στοιχείο εκτός του μηδενός». Το πρόγραμμα να υλοποιηθεί:

1. στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL).
2. στη γλώσσα προγραμματισμού PASCAL.



The screenshot shows a programming environment with a toolbar at the top containing icons for play, step forward, step back, up, search, stop, and a square. Below the toolbar are two tabs: "Κώδικας" (Code) and "Είσοδος" (Input). The code editor displays the following pseudocode:

```
1  Αλγόριθμος Αριθμοί
2  πλ ← 0
3  Σ ← 0
4  Εμφάνισε "Πληκτρολογήστε έναν αριθμό"
5  Διάβασε α
6  Όσο α ≠ 0 επανάλαβε
7     Σ ← Σ + α
8     πλ ← πλ + 1
9     Εμφάνισε "Πληκτρολογήστε έναν αριθμό"
10    Διάβασε α
11  Τέλος_επανάληψης
```




12 **Αν** $πλ \neq 0$ **τότε**

13 **ΜΟ** $\leftarrow \Sigma / πλ$

14 **Εμφάνισε** "Ο μέσος όρος είναι:", **ΜΟ**

15 **αλλιώς**

16 **Εμφάνισε** "Δε δόθηκε άλλο στοιχείο εκτός του μηδενός"

17 **Τέλος_αν**

18 **Τέλος Αριθμοί**



Το προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) υποστηρίζει τις εντολές της ψευδογλώσσας που ορίστηκε στο προηγούμενο κεφάλαιο και ακολουθεί τις αρχές του δομημένου προγραμματισμού.



Στο πρόγραμμα αυτό θα εισαχθεί άγνωστο πλήθος αριθμών προς επεξεργασία. Αυτό θα επιτευχθεί με την εντολή Όσο. Επίσης είναι απαραίτητο να ελεγχθεί αν δόθηκε ένα τουλάχιστον στοιχείο διαφορετικό από το μηδέν ή όχι. Για το συγκεκριμένο έλεγχο θα χρησιμοποιηθεί η εντολή σύνθετης επιλογής.

```
1 Program arithmoi;  
2 VAR  
3 pl: INTEGER;  
4 a, s, mo: REAL;  
5 begin  
6 pl := 0; s := 0;  
7 writeln<'Πληκτρολογήστε έναν αριθμό'>;  
8 readln<a>;  
9 while a<> 0 do  
10     begin  
11         s:= s + a;  
12         pl := pl + 1;  
13         writeln<'Πληκτρολογήστε έναν αριθμό'>;  
14         readln<a>;  
15     end;  
16 if pl <> 0 then  
17     begin  
18         mo := s / pl;  
19         writeln<'Ο μέσος όρος είναι:', mo>;  
20     end  
21 else  
22     writeln<'Δε δόθηκε άλλο στοιχείο εκτός  
23         του μηδενός'>;  
24 end_.
```

1 File, 2 Edit, 3 Search, 4 Run, 5 Compile, 6 Debug
7 Tools, 8 Options, 9 Window, 10 Help



Το προγραμματιστικό περιβάλλον στο οποίο υλοποιήθηκε το πρόγραμμα στη γλώσσα PASCAL μπορεί να βρεθεί στη διεύθυνση

<http://www.freepascal.org/>.

Η εντολή Όσο υλοποιείται με την εντολή **While** και η σύνθετη εντολή επιλογής με την εντολή **If...else**

Η σχεδίαση προγραμμάτων στη γλώσσα PASCAL ακολουθεί τον δομημένο προγραμματισμό.

2.3.2 Σχεδίαση και συγγραφή κώδικα

Οι περισσότερες γλώσσες προγραμματισμού υποστηρίζονται από προγραμματιστικά περιβάλλοντα τα οποία παρέχουν εργαλεία που διευκολύνουν την εργασία του προγραμματιστή. Για τη σύνταξη του πηγαίου προγράμματος χρησιμοποιείται ένα ειδικό πρόγραμμα το οποίο ονομάζεται **συντάκτης (editor)**. Στη συνέχεια το πηγαίο πρόγραμμα πρέπει να μεταφραστεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή δηλαδή σε εντολές γλώσσας μηχανής. Το έργο της μετάφρασης το αναλαμβάνουν δύο προγράμματα ο **μεταγλωττιστής** ή ο **διερμηνευτής**.



Το πρόγραμμα που γράφεται σε κάποια γλώσσα προγραμματισμού ονομάζεται **πηγαίο πρόγραμμα (source program)**.

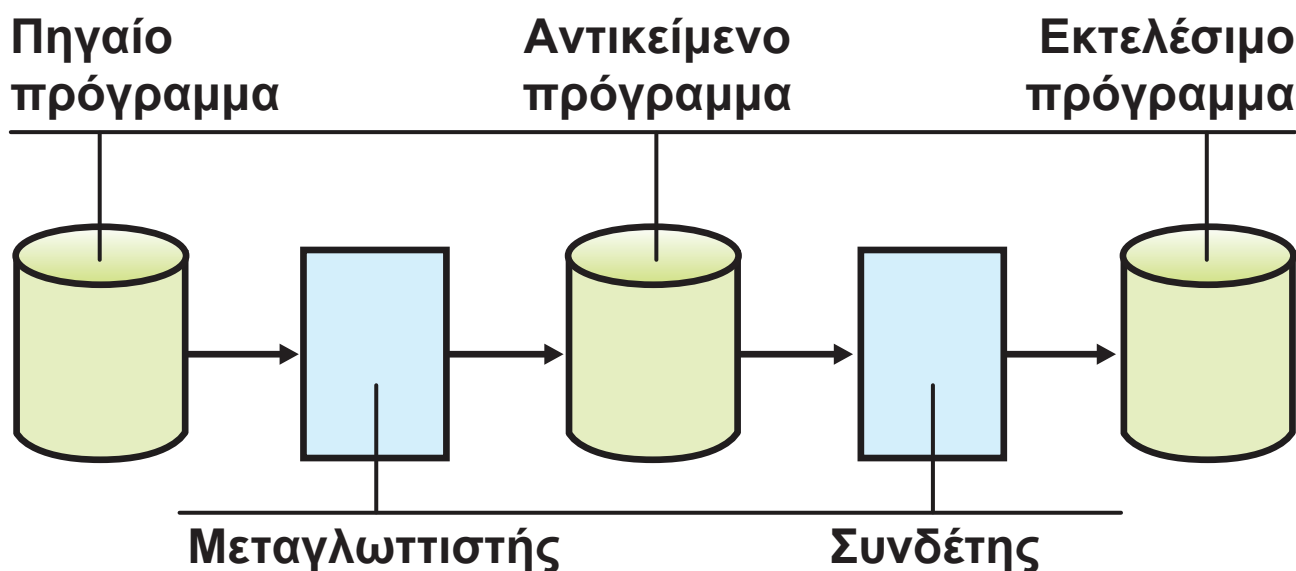


Οι μεταγλωττιστές (compilers) δέχονται στην είσοδο ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγουν ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. Το πρόγραμμα που παράγεται ονομάζεται **αντικείμενο (object) πρόγραμμα**. Το αντικείμενο πρόγραμμα δεν είναι σε θέση να εκτελεστεί. Χρειάζεται να συνδεθεί με άλλα τμήματα προγράμματος τα οποία είτε τα γράφει ο προγραμματιστής, είτε βρίσκονται στις βιβλιοθήκες (libraries) της γλώσσας. Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης – φορτωτής (linker - loader)**. Το αποτέλεσμα είναι η παραγωγή του **εκτελέσιμου (executable) προγράμματος (Εικόνα 2.31)**. Το τελευταίο μπορεί να εκτελείται οποτεδήποτε από τον υπολογιστή και είναι τελείως ανεξάρτητο από το πηγαίο πρόγραμμα.

Οι **διερμηνευτές (interpreters)** διαβάζουν μία προς μία τις εντολές του πηγαίου προγράμματος και για καθεμία εκτελούν αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

Για να μεταφραστεί το πηγαίο πρόγραμμα σε εντολές γλώσσας μηχανής δεν θα πρέπει να ανιχνευθούν λάθη. Τα λάθη που εμφανίζονται κατά τη μετάφραση ονομάζονται **συντακτικά**. Τα συντακτικά λάθη μπορεί να οφείλονται σε αναγραμματισμούς, σε λανθασμένη σύνταξη εντολών, παράλειψη δήλωσης μεταβλητών κ.ά..

Ο μεταφραστής ανιχνεύει τα λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα. Στη συνέχεια ακολουθεί η διόρθωσή τους από τον προγραμματιστή.

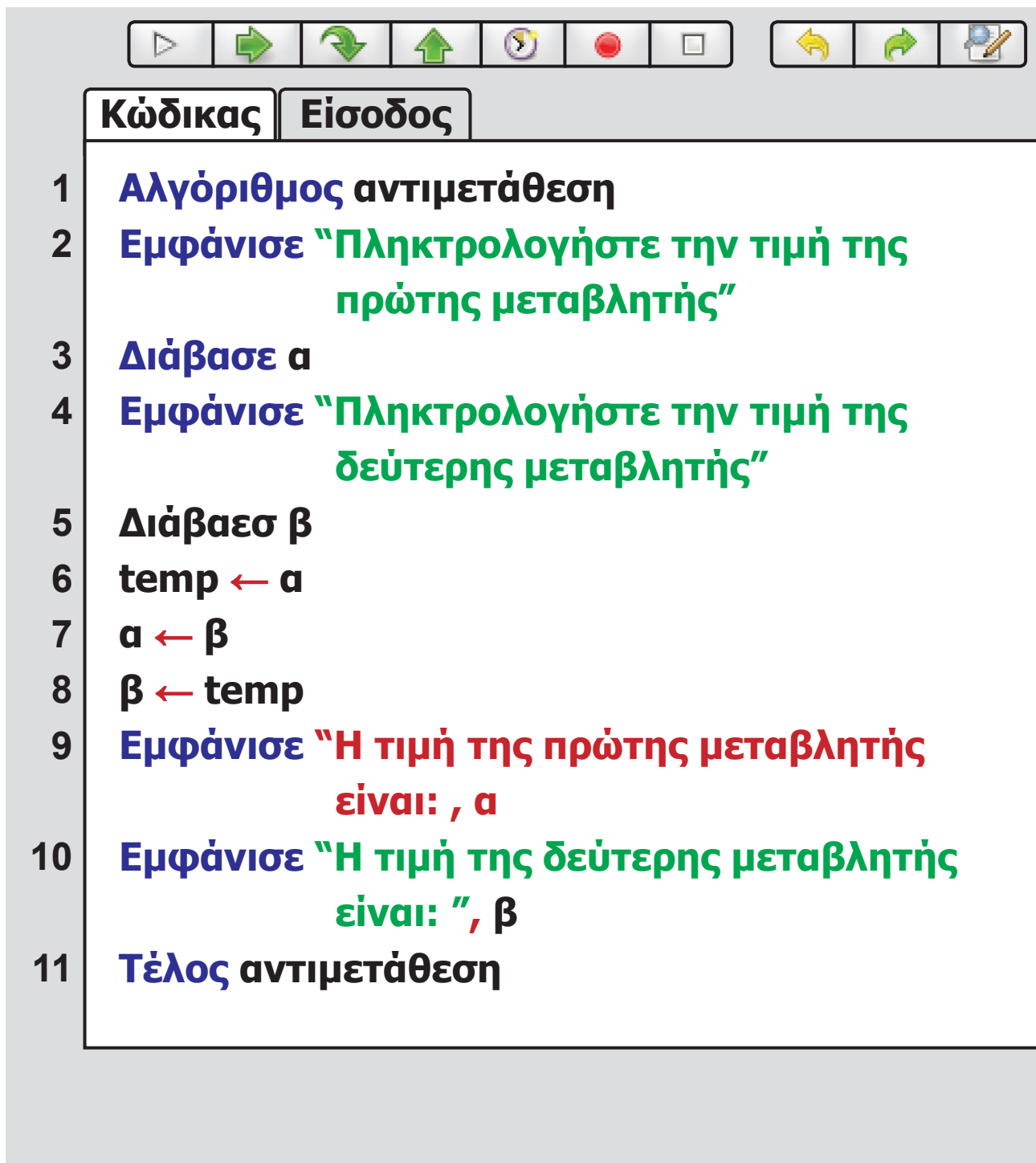


Εικόνα 2.31.

Μεταγλώττιση και Σύνδεση Προγράμματος.

Πέρα όμως από τα συντακτικά λάθη υπάρχουν και τα λογικά που δεν είναι δυνατό να ανιχνευθούν από τα μεταφραστικά προγράμματα. Τα περισσότερα όμως προγραμματιστικά περιβάλλοντα παρέχουν εργαλεία εκσφαλμάτωσης που βοηθούν τον προγραμματιστή να εκτελέσει το πρόγραμμα εντολή προς εντολή μέχρι συγκεκριμένο σημείο ή να παρακολουθεί τις τιμές των μεταβλητών έτσι ώστε να εντοπίσει τα λάθη στην υλοποίηση του αλγορίθμου.

Παράδειγμα 2.33. Με σκοπό να αναπτυχθεί πρόγραμμα το οποίο θα αντιμετωπίζει το περιεχόμενο δύο μεταβλητών γράφτηκαν οι ακόλουθες εντολές στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL).



The screenshot shows the ViALGOL programming environment. At the top, there is a toolbar with icons for execution (play, step, run, stop, quit) and editing (undo, redo, save). Below the toolbar are two tabs: "Κώδικας" (Code) and "Είσοδος" (Input). The code editor contains the following pseudocode:

```
1  Αλγόριθμος αντιμετάθεση
2  Εμφάνισε "Πληκτρολογήστε την τιμή της
    πρώτης μεταβλητής"
3  Διάβασε α
4  Εμφάνισε "Πληκτρολογήστε την τιμή της
    δεύτερης μεταβλητής"
5  Διάβασε β
6  temp ← α
7  α ← β
8  β ← temp
9  Εμφάνισε "Η τιμή της πρώτης μεταβλητής
    είναι: , α
10 Εμφάνισε "Η τιμή της δεύτερης μεταβλητής
    είναι: ", β
11 Τέλος αντιμετάθεση
```



Στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας, το χρώμα των γραμμών διαφοροποιείται σε μπλε για τις δεσμευμένες λέξεις, σε μαύρο για τις μεταβλητές και πράσινο για τις αλφαριθμητικές σταθερές ώστε να διακρίνονται εύκολα τα διάφορα στοιχεία του προγράμματος.

Τι θα συμβεί κατά τη μετάφραση του προγράμματος;

Απάντηση

Υπάρχουν δύο συντακτικά λάθη στις εντολές που έχουν γραφεί. Το πρώτο βρίσκεται στην πέμπτη γραμμή του προγράμματος και οφείλεται σε αναγραμματισμό της εντολής Διάβασε και το δεύτερο στην ένατη γραμμή και οφείλεται στην παράλειψη των διπλών εισαγωγικών στο τέλος της αλφαριθμητικής έκφρασης. Το προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) χρησιμοποιεί διερμηνευτή για τη μετάφραση του πηγαίου προγράμματος. Έτσι αρχικά θα εντοπιστεί το πρώτο συντακτικό λάθος με ένα μήνυμα της μορφής:

**Συντακτικό Λάθος - Περιμένα την εντολή εκχώρησης
Βρήκα: β**

Ο διερμηνευτής αναγνωρίζει τη λέξη Διάβασε ως το όνομα κάποιας μεταβλητής και περιμένει στη συνέχεια να δει το αριστερό βέλος της εντολής εκχώρησης, κάτι το οποίο όμως δεν συμβαίνει. Επίσης εντοπίζει μόνο το πρώτο συντακτικό λάθος. Εφόσον αυτό διορθωθεί,

ο διερμηνευτής βρίσκει το επόμενο λάθος και εμφανίζει ένα μήνυμα της μορφής:

Συντακτικό Λάθος - Περιμένα το χαρακτήρα. "



Το ερώτημα αν οι μηχανές μπορούν να σκέφτονται έχει την ίδια σημασία με το ερώτημα αν τα υποβρύχια μπορούν να κολυμπούν.

Έντσγκερ Ντάικστρα (Edsger Dijkstra),
Επιστήμονας Πληροφορικής

Παράδειγμα 2.34. Με σκοπό να αναπτυχθεί πρόγραμμα το οποίο θα αντιμετωπίζει το περιεχόμενο δύο μεταβλητών γράφτηκαν οι ακόλουθες εντολές στο προγραμματιστικό περιβάλλον FreePascal.



Η γλώσσα PASCAL είναι μία από τις γλώσσες για τη συγγραφή κώδικα στον Πανελλήνιο Διαγωνισμό Πληροφορικής. <http://www.pdp.gr>



Στη γλώσσα προγραμματισμού PASCAL απαιτείται η δήλωση του τύπου των μεταβλητών που χρησιμοποιούνται από το πρόγραμμα.


```
1 Program antimetathesi;  
2 VAR  
3 a, b: INTEGER;  
4 begin  
5 writeln<'Πληκτρολογήστε την τιμή της πρώτης  
6 μεταβλητής'>;  
7 readln<a>;  
8 writeln<'Πληκτρολογήστε την τιμή της δεύτερης  
9 μεταβλητής'>;  
10 readln<b>;  
c := a;  
a = b;  
b := c;  
writeln<'Η τιμή της πρώτης μεταβλητής είναι: ',  
a:5>;  
writeln<'Η τιμή της δεύτερης μεταβλητής είναι: ',  
b:5>;  
end.
```

1 File, 2 Edit, 3 Search, 4 Run, 5 Compile, 6 Debug
7 Tools, 8 Options, 9 Window, 10 Help

Τι θα συμβεί κατά τη μετάφραση του προγράμματος;

Απάντηση

Το μεταφραστικό πρόγραμμα στο συγκεκριμένο προγραμματιστικό περιβάλλον είναι μεταγλωττιστής. Έτσι θα εμφανιστεί το εξής μήνυμα.

antimetathesi.pas<12,3> Error: Identifier not found "c"

antinetathesi.pas <13,6> Error: Illegal expression

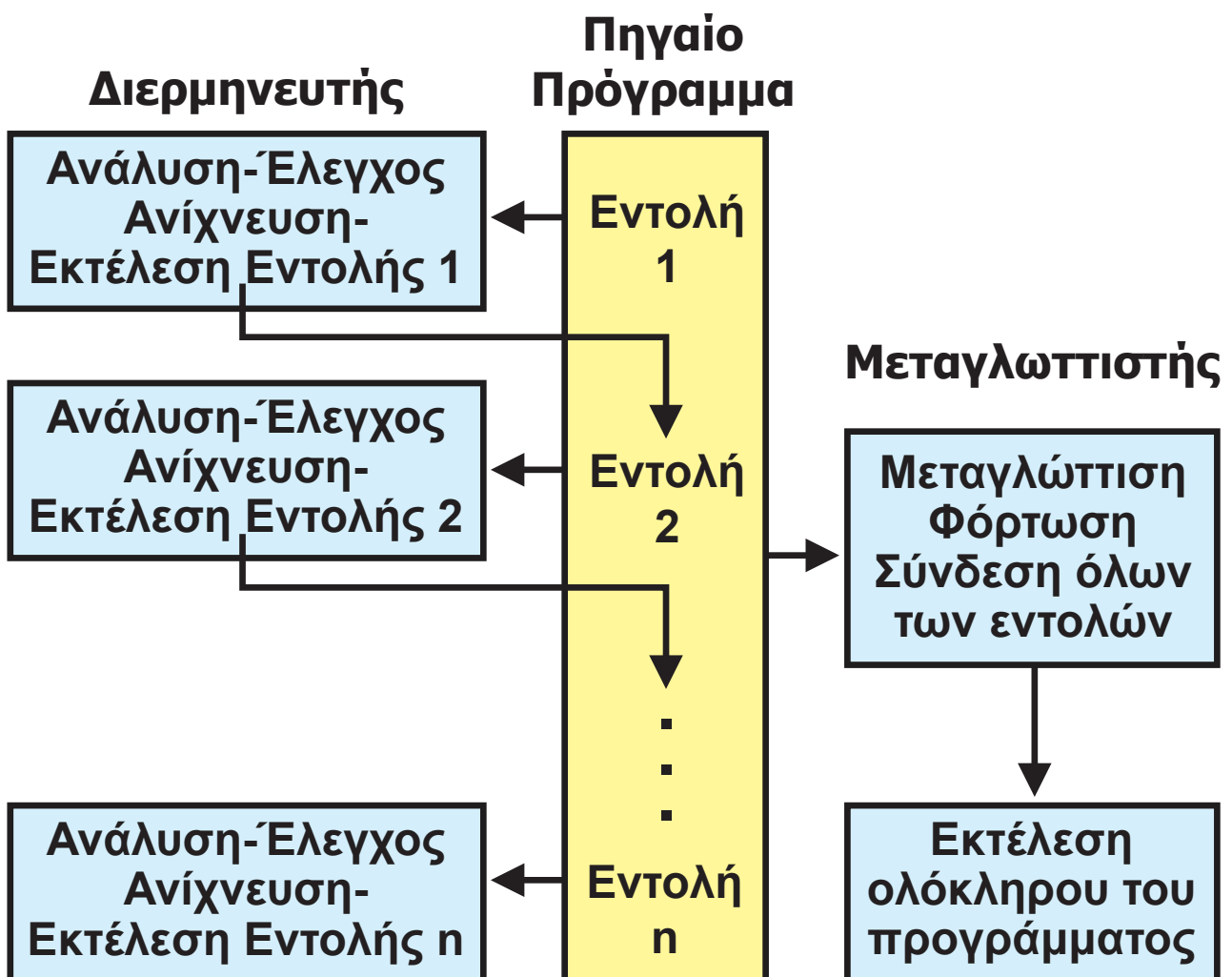
antimetathesi.pas<14,7> Error: Identifier not found "c"

antimetathesi.pas<20> Fatal: There were 3 errors compiling module,

antimetathesi.pas<0> Fatal: Compilation aborted

Εντοπίζονται όλα τα συντακτικά λάθη που υπάρχουν. Τα δύο οφείλονται σε παράλειψη δήλωσης της μεταβλητής c και το άλλο οφείλεται σε λανθασμένη σύνταξη της εντολής εκχώρησης της τιμής της μεταβλητής b στη μεταβλητή a .

Η διαδικασία μετάφρασης λοιπόν ενός προγράμματος σε προγραμματιστικά περιβάλλοντα που διαθέτουν είτε διερμηνευτή είτε μεταγλωττιστή φαίνεται στην εικόνα 2.32.



Εικόνα 2.32. Διαδικασία Μετάφρασης και Εκτέλεσης ενός προγράμματος.

Παράδειγμα 2.35. Να αναπτυχθεί πρόγραμμα το οποίο θα αντιμεταθέτει το περιεχόμενο δύο μεταβλητών στο προγραμματιστικό περιβάλλον SCRATCH.

Όταν στο  γίνει κλικ

ρώτησε

όρισε το σε

ρώτησε

όρισε το σε

όρισε το σε

όρισε το σε

όρισε το σε

πες

πες

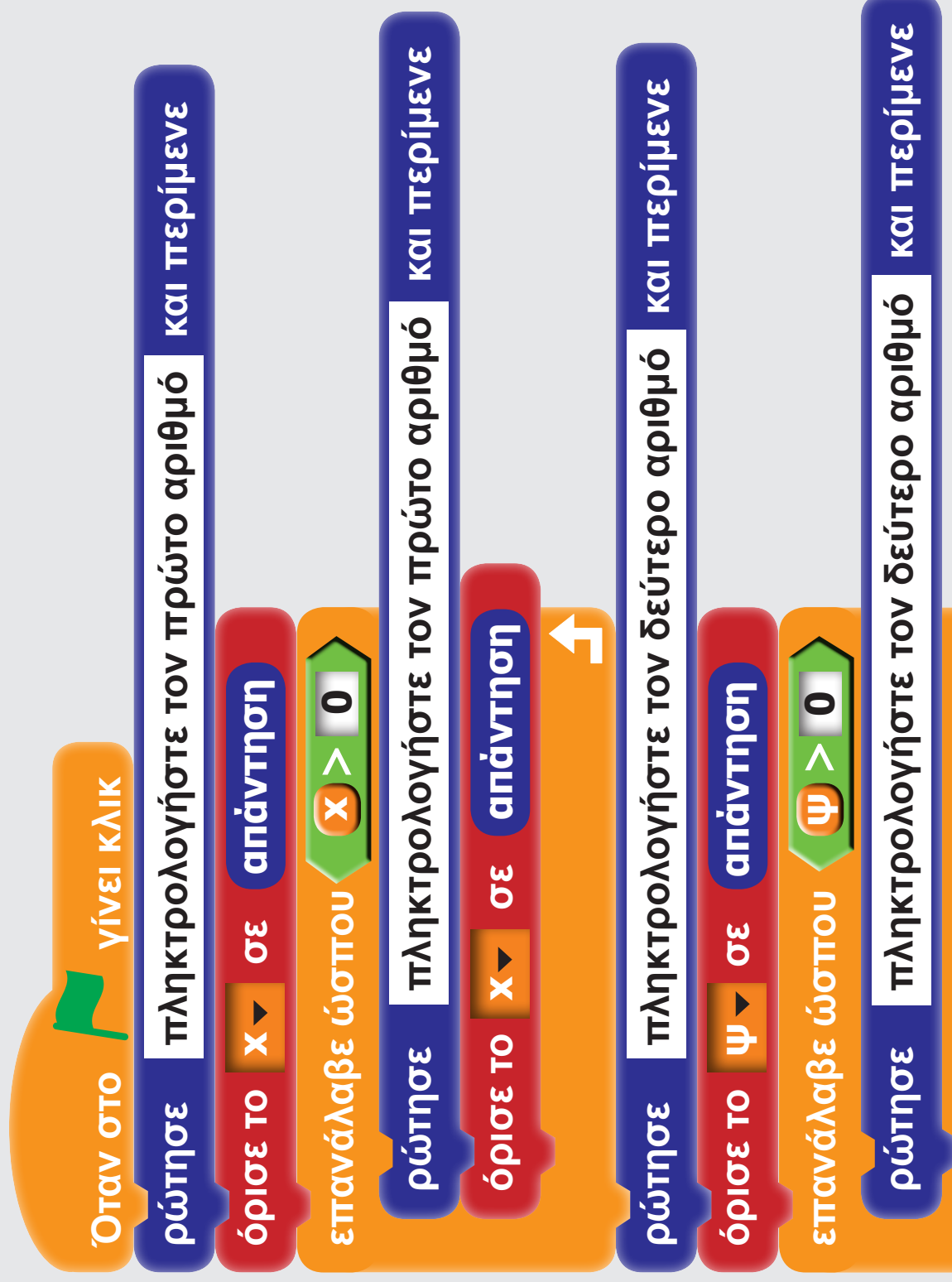
μεταβλητής και περίμενε

μεταβλητής και περίμενε

με το α για 5 δευτερόλεπτα

με το β για 5 δευτερόλεπτα

Παράδειγμα 2.36. Εύρεση μέγιστου κοινού διαιρέτη δύο θετικών ακέραιων αριθμών με τον επαναληπτικό αλγόριθμο του Ευκλείδη σε γλώσσα SCRATCH.



```

when green flag clicked
  say [ ] for 2 sec
  loop
    replace x with x mod y
    until x = 0
  say [Ο μέγιστος κοινός διαιρέτης είναι: y] for 2 sec

```

Η γλώσσα οπτικού προγραμματισμού SCRATCH έχει δημιουργηθεί στο MIT. <http://scratch.mit.edu>

Στη γλώσσα SCRATCH ο προγραμματισμός γίνεται με χρήση εντολών που μοιάζουν με κομμάτια ενός παζλ (Πλακίδια – blocks). Κάθε εντολή έχει το δικό της χαρακτηριστικό σχήμα και μπορεί να συνδυαστεί με άλλες μόνο με συγκεκριμένο τρόπο που αποκλείει τα συντακτικά λάθη.

Οι εντολές όταν συνδυάζονται δημιουργούν σενάρια ενεργειών που πρέπει να εκτελεστούν από αντικείμενα που λέγονται μορφές (sprites). Η πιο χαρακτηριστική μορφή είναι η γάτα.



Εικόνα 2.33. Η μορφή της γάτας.

Η εντολή Όσο της ψευδογλώσσας και η εντολή While της PASCAL δεν υπάρχει στη γλώσσα SCRATCH. Υπάρχει όμως η εντολή επανάλαβε ώσπου στην οποία οι εντολές εκτελούνται όσο η συνθήκη είναι ψευδής.

Το πρόγραμμα αρχικά εξασφαλίζει ότι οι τιμές που πληκτρολογούνται είναι θετικές.

Παράδειγμα 2.37. Να γραφεί πρόγραμμα για την εύρεση του μέγιστου κοινού διαιρέτη δύο θετικών ακεραίων αριθμών με τον αναδρομικό αλγόριθμο του

Ευκλείδη στη γλώσσα LOGO. Ο κώδικας να αναπτυχθεί ως διαδικασία στο περιβάλλον MicroWorlds Pro.

```
για μκδ :χ :ψ
ΑνΔιαφορετικά :ψ = 0
[
  κάνε "ζ :χ
  ανακοίνωση (φρ[ο μέγιστος κοινός διαιρέτης είναι] :ζ)
]
[
  κάνε "χ υπόλοιπο :χ :ψ
  μκδ :ψ :χ
]
τέλος
```



Αρκετές φορές οι αλγόριθμοι χρειάζεται να τροποποιηθούν ώστε να μπορούν να εκτελεστούν από κάποια γλώσσα προγραμματισμού.

Για να εκτελεστεί το διπλανό πρόγραμμα και να ανακοινώσει το μέγιστο κοινό διαιρέτη των αριθμών 27 και 6, αρκεί να πληκτρολογήσει ο χρήστης στο κέντρο εντολών του περιβάλλοντος την εντολή:

μκδ 27 6

Οι μεταβλητές χ και ψ ονομάζονται παράμετροι.

Η σύνθετη επιλογή υλοποιείται με την εντολή

ΑνΔιαφορετικά.

Πολλές φορές κάποια προγράμματα μπορούν να χρησιμοποιήσουν κώδικα που έχει γραφτεί προηγουμένως. Η επαναχρησιμοποίηση κώδικα είναι αρκετά συνηθισμένη πρακτική η οποία περιορίζει τα λάθη και μειώνει το χρόνο που απαιτείται για τη συγγραφή του προγράμματος. Προγραμματιστικά γίνεται συνήθως με τη δημιουργία κατάλληλων υποπρογραμμάτων, δηλαδή διαδικασιών ή συναρτήσεων ανάλογα με τη γλώσσα προγραμματισμού. Αρκετές φορές ο προγραμματιστής μπορεί να γράψει ή να χρησιμοποιήσει βιβλιοθήκες (libraries). Οι βιβλιοθήκες μιας γλώσσας προγραμματισμού είναι μία συλλογή από έτοιμα υποπρογράμματα που μπορούν να χρησιμοποιούνται κατά τη συγγραφή νέων προγραμμάτων. Μία βιβλιοθήκη μπορεί να χρησιμοποιηθεί τόσο από τον δημιουργό της όσο και από άλλους προγραμματιστές. Αποτελεί επίσης πρακτική πολλών προγραμματιστών να βελτιώνουν υπάρχουσες βιβλιοθήκες, ώστε να επεκτείνουν τις δυνατότητες των προγραμμάτων τους.

Παράδειγμα 2.38. Να γραφεί πρόγραμμα που θα δέχεται δύο θετικούς ακέραιους αριθμούς και θα εμφανίζει το μέγιστο κοινό διαιρέτη και το ελάχιστο κοινό πολλαπλάσιο τους. Η υλοποίηση να γίνει στο περιβάλλον MicroWorlds Pro.

```
για mkδ :χ :ψ
ΑνΔιαφορετικά :ψ = 0
[
  κάνε "ζ :χ
]
[
```

κάνε "χ υπόλοιπο :χ :ψ

μκδ :ψ :χ

]

τέλος

για εκπο :χ :ψ

μκδ :χ :ψ

κάνε "εκ :χ * :ψ / :ζ

τέλος

για υπολογισμός :χ :ψ

εκπο :χ :ψ

ανακοίνωση (φρ[ο μέγιστος κοινός διαιρέτης είναι] :ζ)

ανακοίνωση (φρ[το ελάχιστο κοινό πολλαπλάσιο είναι] :εκ)

τέλος



Αν χ, ψ είναι δύο θετικοί ακέραιοι τότε

$$\text{ΜΚΔ}(\chi, \psi) * \text{ΕΚΠ}(\chi, \psi) = \chi * \psi$$

Το πρόγραμμα στη γλώσσα LOGO περιλαμβάνει τρεις διαδικασίες: μία για τον υπολογισμό του μέγιστου κοινού διαιρέτη η οποία επαναχρησιμοποιείται, μία για τον υπολογισμό του ελάχιστου κοινού πολλαπλάσιου που καλεί τη διαδικασία του μέγιστου κοινού διαιρέτη και μία που καλεί τη διαδικασία υπολογισμού του ελάχιστου κοινού πολλαπλάσιου και εμφανίζει τα αποτελέσματα.

Οι τιμές των μεταβλητών στη LOGO είναι διαθέσιμες σε όλες τις διαδικασίες. Οι μεταβλητές αυτές ονομάζονται καθολικές. Σε άλλες γλώσσες προγραμματισμού υπάρχουν και οι τοπικές μεταβλητές, οι οποίες μπορούν να προσπελαστούν μόνο στο πρόγραμμα ή το υποπρόγραμμα το οποίο δηλώνονται.

Παράδειγμα 2.39. Να γραφεί πρόγραμμα σε γλώσσα SCRATCH το οποίο θα αναζητά αν υπάρχει ένα όνομα σε μία λίστα και θα εμφανίζει κατάλληλο μήνυμα σχετικά με την εύρεσή του.



αλλιώς

άλλαξε το **i** κατά **1**



εάν **found** = **1** ΤΟΤΕ

ΠΕΣ **ένωσε το** **το όνομα υπάρχει στη λίστα στη θέση** **index**

αλλιώς

ΠΕΣ **το όνομα δεν υπάρχει**



Ο τρόπος επικοινωνίας μεταξύ των ενοτήτων ή υποπρογραμμάτων (διαδικασιών ή συναρτήσεων) και των προγραμμάτων διαφέρει από γλώσσα σε γλώσσα προγραμματισμού. Κατά βάση αυτή η επικοινωνία επιτυγχάνεται με τη χρήση των παραμέτρων, δηλαδή μεταβλητών που επιτρέπουν το πέρασμα της τιμής τους από ένα τμήμα προγράμματος σε ένα άλλο.

Η δομή δεδομένων, που είναι αποθηκευμένα τα ονόματα στο παράδειγμα 2.39, είναι μία λίστα με το όνομα `names`.

Ο αλγόριθμος που κωδικοποιείται είναι αυτός της σειριακής αναζήτησης κατάλληλα τροποποιημένος.

Παράδειγμα 2.40. Σε ένα ραδιοφωνικό σταθμό το κόστος ενός διαφημιστικού μηνύματος σε σχέση με τα δευτερόλεπτα μετάδοσης, υπολογίζεται κλιμακωτά σύμφωνα με τον παρακάτω πίνακα.

Χρονική Διάρκεια διαφήμισης (δευτερόλεπτα)	Κόστος (€ / δευτερόλεπτο)
Μέχρι 15	90
Μέχρι 30	75
Πάνω από 30	50

- A. Να αναπτύξετε πρόγραμμα στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) το οποίο
- να διαβάζει το πλήθος των διαφορετικών μηνυμάτων που πρόκειται να μεταδώσει ο σταθμός την επόμενη εβδομάδα
 - να διαβάζει τη χρονική διάρκεια κάθε μηνύματος και να υπολογίζει καλώντας κατάλληλο υποπρόγραμμα το κόστος του
 - να εμφανίζει με κατάλληλο μήνυμα τα συνολικά έσοδα του σταθμού καθώς και το ποσοστό (%) των μηνυμάτων με χρονική διάρκεια άνω των 30 δευτερολέπτων.
- B. Να γράψετε τον αλγόριθμο για τον υπολογισμό του κόστους του κάθε μηνύματος.
- Ο κώδικας του προγράμματος είναι:

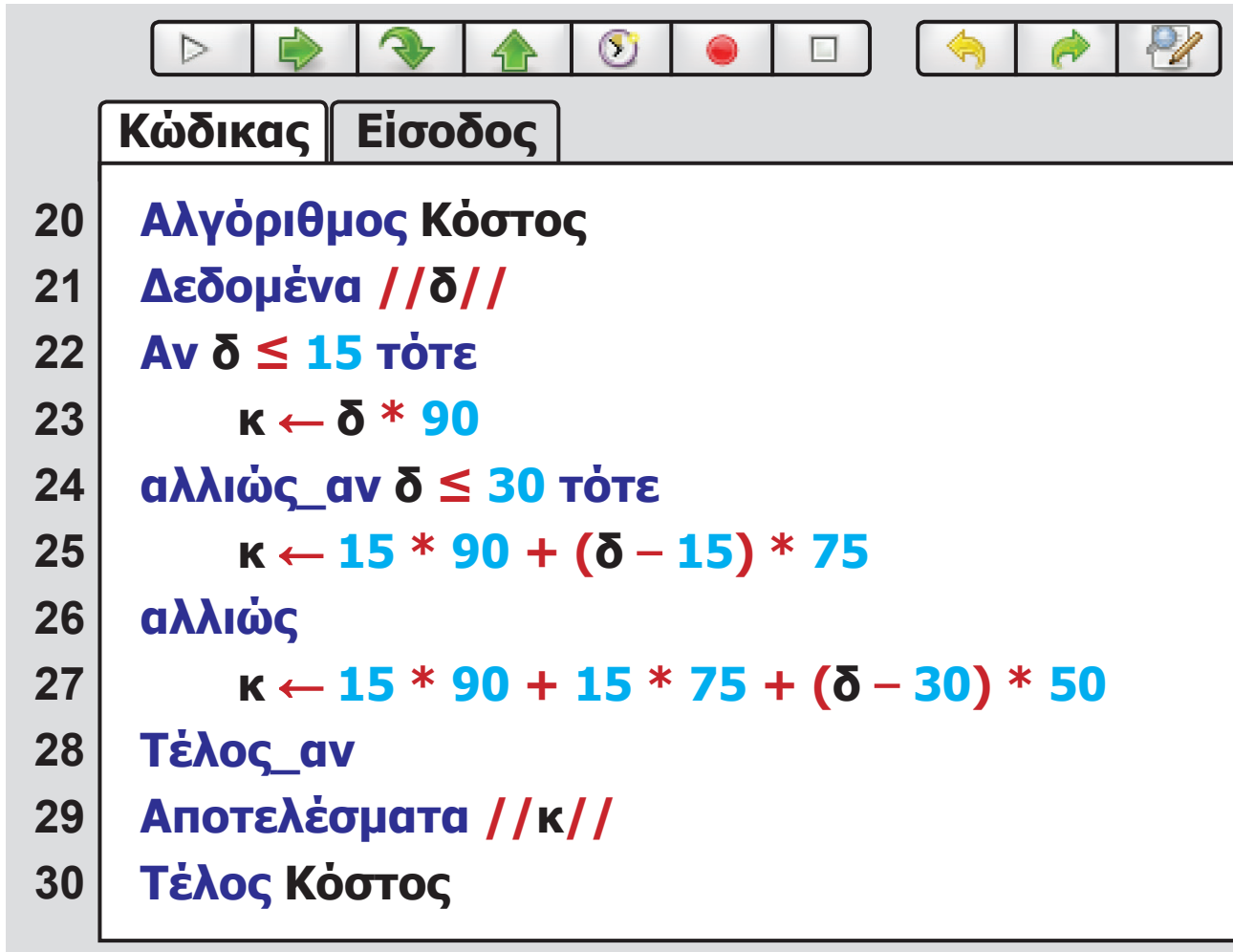


Κώδικας

Είσοδος

```
1  Αλγόριθμος Ραδιόφωνο
2  Σ ← 0
3  π30 ← 0
4  Εμφάνισε "Πληκτρολογήστε το πλήθος των
      μηνυμάτων"
5  Διάβασε n
6  Για i από 1 μέχρι n
7      Εμφάνισε "Πληκτρολογήστε τη διάρκεια
      του", i, "ου μηνύματος"
8      Διάβασε χδ
9      Κάλεσε Κόστος(χδ, κμ)
10     Σ ← Σ + κμ
11     Αν χδ > 30 τότε
12         π30 ← π30 + 1
13     Τέλος_αν
14 Τέλος_επανάληψης
15 Εμφάνισε "Τα συνολικά έσοδα του σταθμού
      είναι", Σ
16 ποσοστό ← π30 / n * 100
17 Εμφάνισε "Το ποσοστό είναι", ποσοστό, "%"
18 Τέλος Ραδιόφωνο
```


Ο κώδικας του υποπρογράμματος είναι:



The screenshot shows a code editor with a toolbar at the top containing icons for play, step forward, step back, up, search, stop, and a window icon. Below the toolbar are two tabs: 'Κώδικας' (Code) and 'Είσοδος' (Input). The code is as follows:

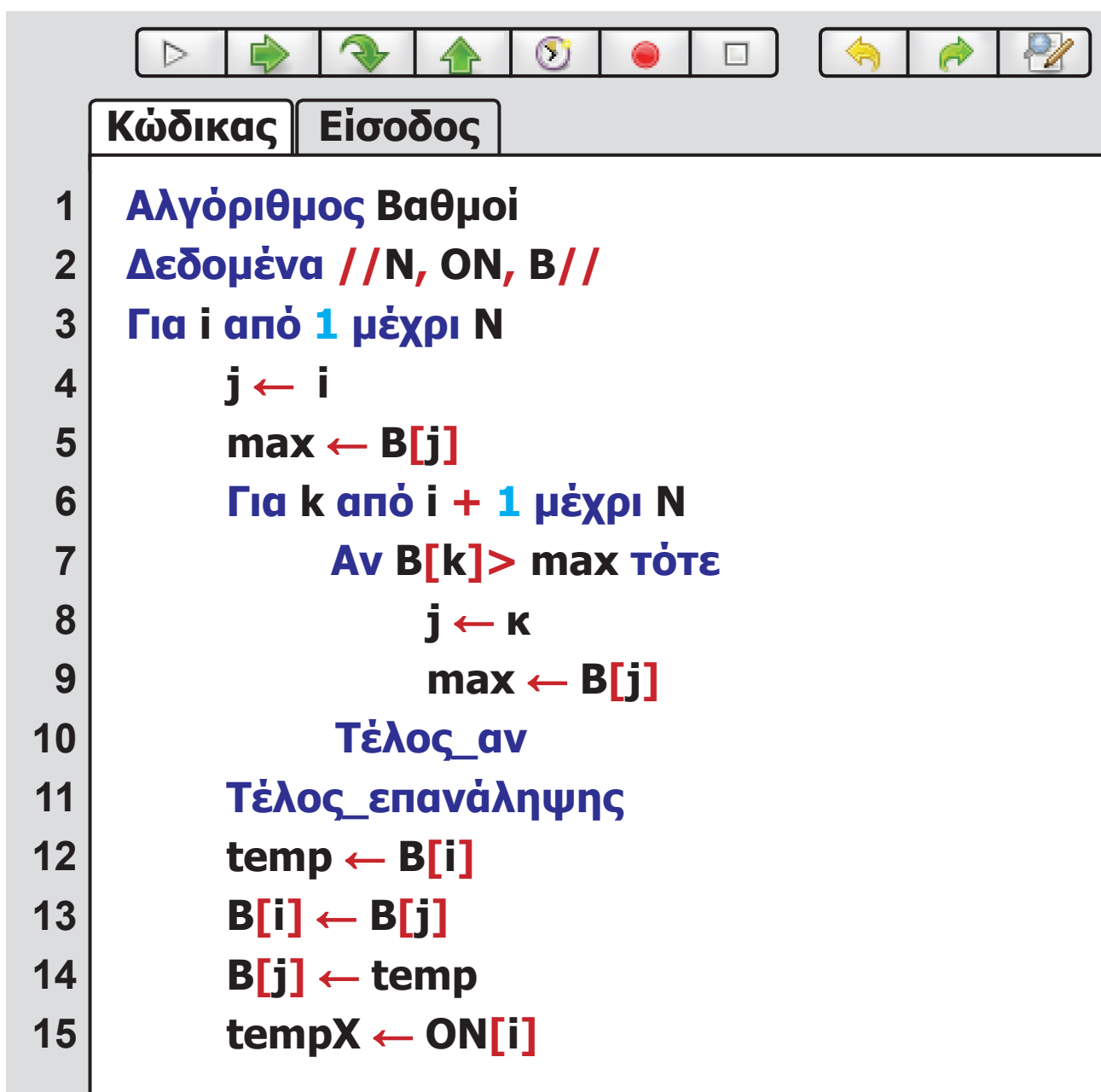
```
20 Αλγόριθμος Κόστος
21 Δεδομένα //δ//
22 Αν δ ≤ 15 τότε
23     κ ← δ * 90
24 αλλιώς_αν δ ≤ 30 τότε
25     κ ← 15 * 90 + (δ - 15) * 75
26 αλλιώς
27     κ ← 15 * 90 + 15 * 75 + (δ - 30) * 50
28 Τέλος_αν
29 Αποτελέσματα //κ//
30 Τέλος Κόστος
```



Η χρήση της δομής του πίνακα, όταν αυτή δεν κρίνεται απαραίτητη, επαφίεται στη λογική του προγραμματιστή. Κατά βάση όμως στις εφαρμογές λογισμικού τα δεδομένα αποθηκεύονται σε δομές δεδομένων.

Κατά την κλήση του αλγορίθμου Κόστος, η μεταβλητή $\chi\delta$ μεταβιβάζει την τιμή της στη μεταβλητή δ και όταν ολοκληρώνεται η εκτέλεση του αλγορίθμου η μεταβλητή κ μεταβιβάζει την τιμή της στη μεταβλητή $\kappa\mu$.

Παράδειγμα 2.41. Να γραφεί πρόγραμμα στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) το οποίο να δέχεται έναν πίνακα που περιέχει τα ονόματα των μαθητών ενός τμήματος και έναν παράλληλο πίνακα με το βαθμό απολυτηρίου τους. Το πρόγραμμα να ταξινομεί τον πίνακα με τα ονόματα σε φθίνουσα σειρά με βάση τους βαθμούς. Τέλος να εμφανίζει το όνομα του κάθε μαθητή και δίπλα τον βαθμό του.



The screenshot shows the ViALGOL programming environment. At the top, there is a toolbar with various icons for execution and editing. Below the toolbar, there are two tabs: "Κώδικας" (Code) and "Είσοδος" (Input). The "Κώδικας" tab is active, displaying the following pseudocode:

```
1  Αλγόριθμος Βαθμοί
2  Δεδομένα //N, ON, B//
3  Για i από 1 μέχρι N
4      j ← i
5      max ← B[j]
6      Για k από i + 1 μέχρι N
7          Αν B[k] > max τότε
8              j ← k
9              max ← B[j]
10     Τέλος_αν
11     Τέλος_επανάληψης
12     temp ← B[i]
13     B[i] ← B[j]
14     B[j] ← temp
15     tempX ← ON[i]
```



Κώδικας | **Είσοδος**

```
16      ON[i] ← ON[j]
17      ON[j] ← tempX
18  Τέλος_επανάληψης
19  Για i από 1 μέχρι N
20      Εμφάνισε ON[i], B[i]
21  Τέλος_επανάληψης
22  Τέλος Βαθμοί
```



Οι πίνακες στους οποίους χρησιμοποιούνται αντίστοιχοι δείκτες θέσης για την αποθήκευση συσχετιζόμενων τιμών ονομάζονται παράλληλοι πίνακες.



Με την εντολή
Δεδομένα //N, ON, B//

γνωστοποιούνται το μέγεθος και οι τιμές των πινάκων στο πρόγραμμα.

Στο πρόγραμμα του παραδείγματος 2.41 εφαρμόζεται ο αλγόριθμος της ταξινόμησης με επιλογή. Έχει τροποποιηθεί όμως σε σχέση με τον αλγόριθμο που παρουσιάστηκε στο προηγούμενο κεφάλαιο, ώστε να βρίσκει κάθε φορά το μέγιστο στοιχείο.

Παράδειγμα 2.42. Να αναπτυχθεί πρόγραμμα στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) το οποίο να διαβάζει τους βαθμούς των μαθητών ενός τμήματος και να εμφανίζει με κατάλληλα μηνύματα το μέσο όρο των βαθμών καθώς και πόσοι από αυτούς είναι μεγαλύτεροι από το μέσο όρο. Να γίνεται έλεγχος ότι οι βαθμοί που εισάγονται είναι ακέραιες τιμές μεταξύ του 0 και του 20. Μετά την εισαγωγή κάθε βαθμού το πρόγραμμα να ερωτά τον χρήστη αν θέλει να συνεχίσει την καταχώρηση εμφανίζοντας το μήνυμα «Θα συνεχίσετε; Πληκτρολογήστε Ν ή Ο» και ανάλογα με την απάντησή του (αποδεκτές τιμές μόνο τα γράμματα του ελληνικού αλφαβήτου «N», «n», «O» και «o»), να συνεχίζει την καταχώριση ή να εμφανίζει τα αποτελέσματα της επεξεργασίας.



Κατά την ανάπτυξη αυτού του προγράμματος θα χρησιμοποιηθεί η δομή δεδομένων του πίνακα, γιατί διαφορετικά δεν θα υπάρχει η δυνατότητα να γίνει η επεξεργασία για τον υπολογισμό του πλήθους αυτών που είναι πάνω από το μέσο όρο.



Γενικά η χρήση πινάκων είναι απαραίτητη αν τα δεδομένα που εισάγονται σε ένα πρόγραμμα πρέπει να διατηρούνται στη μνήμη μέχρι το τέλος της εκτέλεσης.



Κώδικας

Είσοδος

```
1  Αλγόριθμος Βαθμοί
2  i ← 0
3  Σ ← 0
4  Επανάλαβε
5      i ← i+1
6      Εμφάνισε "Πληκτρολογήστε τον", i,
7      "ο βαθμό"
8      Επανάλαβε
9          Διάβασε B[i]
10         Αν B[i] < 0 ή B[i] > 20 ή A_M(B[i]) ≠
11         B[i] τότε
12             Εμφάνισε "Λάθος βαθμός."
13             Εμφάνισε "Πληκτρολογήστε ξανά
14             τον", i, "ο βαθμό"
15         Τέλος_αν
16         Μέχρις_ότου B[i] ≥ 0 και B[i] ≤ 20 και
17         A_M(B[i]) = B[i]
18         Σ ← Σ + B[i]
19     Επανάλαβε
20     Εμφάνισε "Θέλετε να συνεχίσετε;
21     Ναι(N ή ν) – Όχι(O ή ο)"
22     Διάβασε απ
```



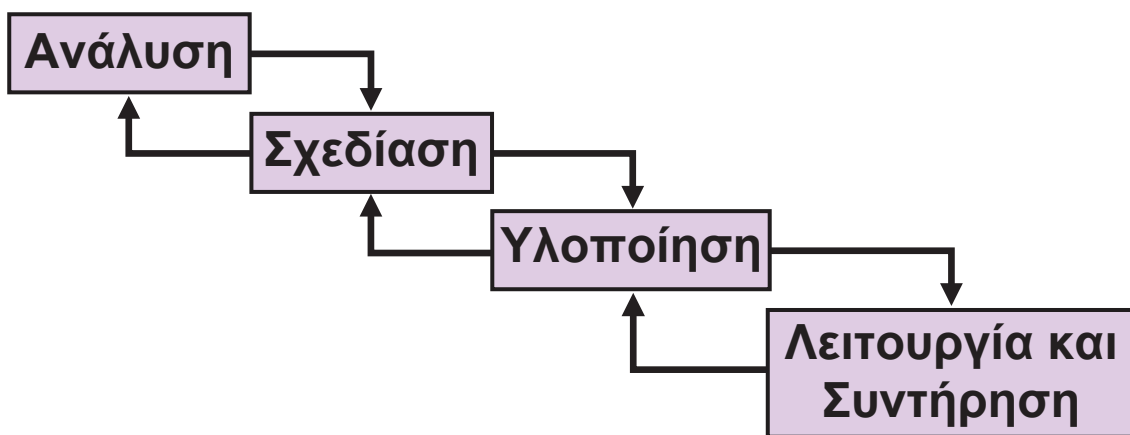
Κώδικας

Είσοδος

```
18     Μέχρις_ότου απ = "N" ή απ = "v" ή απ = "O"  
     ή απ = "o"  
19 Μέχρις_ότου απ = "O" ή απ = "o"  
20 N ← i  
21 MO ← Σ/N  
22 πλ ← 0  
23 Για i από 1 μέχρι N  
24     Αν B[i] > MO τότε  
25         πλ ← πλ + 1  
26     Τέλος_αν  
27 Τέλος_επανάληψης  
28 Εμφάνισε "Ο μέσος όρος είναι: ", MO  
29 Εμφάνισε "Υπάρχουν", πλ, "βαθμοί πάνω από το  
μέσο όρο"  
30 Τέλος Βαθμοί
```

2.3.3 Κύκλος ζωής εφαρμογής λογισμικού

Ένα πρόγραμμα αρχίζει την ζωή του από την στιγμή που θα καθοριστούν οι απαιτήσεις του, οι προδιαγραφές του και παύει να ζει όταν εξαντληθούν όλα τα περιθώρια συντήρησής του (προσθήκες, αλλαγές, βελτιώσεις). Ο Κύκλος Ζωής ενός προγράμματος περιλαμβάνει διάφορες φάσεις οι οποίες αλληλεπιδρούν μεταξύ τους. Οι φάσεις απεικονίζονται στο διάγραμμα της εικόνας 2.34.



Εικόνα 2.34. Κύκλος Ζωής Προγράμματος.

Στη φάση **Ανάλυσης** που ακολουθεί τον προσδιορισμό του προβλήματος από τον πελάτη, καταγράφονται αναλυτικά τα δεδομένα και τα ζητούμενα του προβλήματος και ζητούνται οι απαραίτητες διευκρινήσεις από τον πελάτη, σε όσα σημεία οι προδιαγραφές παρουσιάζουν ασάφεια.

Στη φάση της **Σχεδίασης** καθορίζεται η δομή του προγράμματος, οι ενότητες (υποπρογράμματα) από τις οποίες θα αποτελείται το πρόγραμμα και αναζητούνται έτοιμες ενότητες (modules) από παλιότερα προγράμματα που μπορούν να χρησιμοποιηθούν και σ' αυτό το πρόγραμμα. Ακόμη επιλέγονται οι αλγόριθμοι και οι δομές δεδομένων που θα χρησιμοποιηθούν σε κάθε ενότητα.

Στην επόμενη φάση της **Υλοποίησης** του προγράμματος επιλέγεται η κατάλληλη γλώσσα προγραμματισμού για το συγκεκριμένο πρόγραμμα όπου συντάσσεται το πηγαίο πρόγραμμα και μεταφράζεται από έναν μεταγλωττιστή, ώστε αυτό να γίνει κατανοητό από τον υπολογιστή. Η μετάφραση θα εντοπίσει πιθανά συντακτικά λάθη. Τα λάθη διορθώνονται και ακολουθεί ξανά μετάφραση του προγράμματος, έως την οριστική εξάλειψή τους. Το πρόγραμμα που προκύπτει είναι το εκτελέσιμο πρόγραμμα.

Στην επόμενη φάση της **Λειτουργίας και Συντήρησης** θα γίνουν όλες οι προσαρμογές και βελτιώσεις που χρειάζονται προκειμένου το πρόγραμμα να συνεχίσει να χρησιμοποιείται. Αυτές ζητούνται όταν διαφοροποιούνται τα δεδομένα του προβλήματος, όταν ο χρήστης ζητήσει νέες λειτουργίες ή προκύψουν κάποια λογικά λάθη που δεν είχαν διαπιστωθεί στον έλεγχο του αλγορίθμου και μπορεί, για παράδειγμα, να αφορούν την επικοινωνία των ενοτήτων. Για την υλοποίηση των αλλαγών μπορεί να επαναληφθεί η εκτέλεση της φάσης της ανάλυσης και σχεδίασης και άρα όλων των

υπολοίπων φάσεων. Έτσι πραγματοποιείται συνολικός έλεγχος του προγράμματος που έχει ως συνέπεια και την καταγραφή σχετικών σχολίων για την τεκμηρίωση.



Στα διάφορα πακέτα λογισμικού, είτε είναι γλώσσες προγραμματισμού, είτε πακέτα εφαρμογών, είτε λειτουργικά συστήματα, δίπλα στο εμπορικό όνομα του λογισμικού, συνηθίζεται να υπάρχει ο αριθμός της έκδοσής του. Ο αριθμός έκδοσης του πακέτου λογισμικού δείχνει ακριβώς τις αλλαγές που έχουν πραγματοποιηθεί από την αρχική του εμφάνιση.

Όταν οι αλλαγές είναι σημαντικές, δηλαδή έχουν προστεθεί νέες λειτουργίες, εντολές, προγράμματα, ο αριθμός έκδοσης αυξάνει κατά ακέραιο αριθμό (GeoGebra 3.0, GeoGebra 4.0), ενώ, όταν οι αλλαγές είναι μικρότερες, αυξάνεται κατά δέκατα ή εκατοστά (GeoGebra 4.1, GeoGebra 4.2).

<http://www.geogebra.org>

Οι εκδόσεις beta μιας εφαρμογής λογισμικού που αναπτύσσει μια εταιρεία δίνονται για έλεγχο συνήθως στους πελάτες της. Οι beta testers παίρνουν την τελική έκδοση δωρεάν ή με σημαντική έκπτωση.

Παράδειγμα 2.43. Ανάπτυξη μιας εφαρμογής αριθμομηχανής για κινητά τηλέφωνα με λειτουργικό Android.

Αρχικά στο στάδιο της ανάλυσης θα πρέπει να καταγραφούν οι απαιτήσεις από την αριθμομηχανή, δηλαδή τι είδους πράξεις θα είναι σε θέση να διεκπεραιώσει. Στη συνέχεια, στη φάση της σχεδίασης θα καθοριστούν οι ενότητες από τις οποίες θα αποτελείται η εφαρμογή, όπως ενότητα που θα υλοποιεί τις απλές αριθμητικές πράξεις ή ενότητα που θα υπολογίζει τριγωνομετρικούς αριθμούς ή τιμές μαθηματικών συναρτήσεων. Επίσης θα επιλεγούν οι αλγόριθμοι και οι δομές δεδομένων που θα χρησιμοποιηθούν σε κάθε ενότητα. Στη φάση της υλοποίησης θα πρέπει να επιλεγεί αρχικά η γλώσσα προγραμματισμού στην οποία θα συνταχθεί το πρόγραμμα. Μία γλώσσα οπτικού προγραμματισμού κατάλληλη για ανάπτυξη εφαρμογών για Android είναι η Google AppInventor. Αφού συνταχθούν όλες οι ενότητες, θα γίνει έλεγχος για τη σωστή τους επικοινωνία. Η αριθμομηχανή είναι έτοιμη για να δοθεί στον τελικό χρήστη. Σε περίπτωση που στη φάση της Λειτουργίας και της Συντήρησης κάποιος χρήστης ζητήσει την επέκταση των δυνατοτήτων της αριθμομηχανής, όπως να προστεθεί στατιστική επεξεργασία δεδομένων, τότε θα πρέπει ο προγραμματιστής πιθανότατα να επαναλάβει όλες τις φάσεις ώστε να καλύψει τις νέες απαιτήσεις.



ΑΝΔΡΟΙΔ

Εικόνα 2.35.

Λογότυπο του
Λειτουργικού Συστήματος
Android.

Στην ιστοσελίδα

<http://www.code.org>

υπάρχουν αρκετές δραστη-
ριότητες κατάλληλες για
την εκμάθηση οπτικού
προγραμματισμού από
μαθητές.

Ωστόσο μια εφαρμογή λογισμικού με σύνθετες λειτουργίες υλοποιείται κατά βάση από μία ομάδα προγραμματιστών που εργάζονται παράλληλα ώστε να μειωθεί ο χρόνος και το κόστος της υλοποίησης. Το κάθε μέλος της ομάδας ανάπτυξης του λογισμικού αναλαμβάνει τη συγγραφή συγκεκριμένων ενοτήτων. Υπάρχουν ταυτόχρονα και μέλη τα οποία είναι υπεύθυνα για το συντονισμό της διαδικασίας ανάπτυξης, τη συνένωση των ενοτήτων σε μία ενιαία εφαρμογή ή για τους ελέγχους του προϊόντος. Τα μεγάλα έργα πληροφορικής εμπλέκουν πολλά διαφορετικά και εξειδικευμένα πρόσωπα, ενώ αποτελούνται από δραστηριότητες των οποίων η σειρά είναι σημαντική.

Ο συντονιστής του έργου ανάπτυξης μιας εφαρμογής λογισμικού (project manager) δημιουργεί και ενημερώνει το πλάνο εργασίας, στελεχώνει το έργο, και παρακολουθεί και ελέγχει την πρόοδό του.

Οι εφαρμογές που αναπτύσσονται από εταιρείες λογισμικού είναι συνήθως **Κλειστού Κώδικα** (proprietary – closed source). Δεν υπάρχει η δυνατότητα για τον χρήστη να δει τον κώδικα ή να επιφέρει κάποια αλλαγή σε αυτόν. Οι βελτιώσεις στα κλειστά λογισμικά γίνονται μέσω αναβαθμίσεων που παρέχονται κατά καιρούς από τις εταιρείες κατασκευής τους, κυρίως μέσω διαδικτύου. Επίσης υπάρχει συνήθως κόστος απόκτησης και απαγόρευση της αναδιανομής της εφαρμογής.

Αντίθετα στις εφαρμογές **Ελεύθερου Λογισμικού / Λογισμικού Ανοικτού Κώδικα** (ΕΛ / ΛΑΚ free / open source) ο κώδικας είναι διαθέσιμος, συνεπώς ο καθένας μπορεί ελεύθερα να χρησιμοποιεί, να μελετά τον τρόπο λειτουργίας, να αντιγράψει, να διανέμει και να τροποποιεί την εφαρμογή προσθέτοντας δικές του βελτιώσεις ή νέες λειτουργίες. Με βάση αυτήν τη φιλοσοφία δημιουργούνται ομάδες προγραμματιστών που μοιράζονται τις αλλαγές που κάνουν στον κώδικα με σκοπό τη βελτίωσή του. Έτσι δημιουργείται ένα παγκόσμιο ανοικτό δίκτυο προγραμματιστών, οι οποίοι συνεργάζονται κυκλοφορώντας νέες εκδόσεις λογισμικού και συμβάλλοντας καθημερινά στη δημιουργία νέων κοινών αγαθών. Το Διαδίκτυο αποτελεί το βασικό μέσο συνεργασίας των προγραμματιστών αλλά και του τρόπου πρόσβασης στο διαθέσιμο Ελεύθερο Λογισμικό.

Πληροφορίες σχετικά με το Λογισμικό Ανοικτού Κώδικα μπορούν να βρεθούν στις διευθύνσεις:

<http://www.ellak.gr/>

<http://www.fsf.org/>

<http://opensource.org/>



Εικόνα 2.36.

**Λίνους Τόρβαλντς
(Linus Torvalds)**

Δημιουργός και συντονιστής της ομάδας εργασίας του λειτουργικού συστήματος ανοιχτού κώδικα Linux

<http://www.linux.org/>



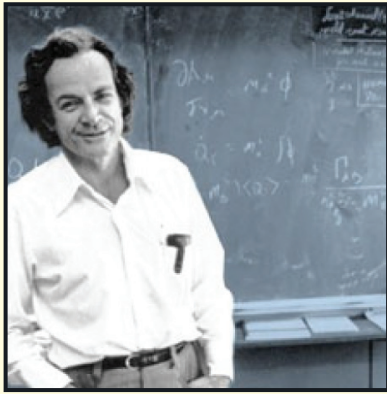
Ανακεφαλαίωση

Στην ενότητα αυτή επιχειρήθηκε να εξηγηθεί τι είναι πρόγραμμα και έγινε αναφορά στις διάφορες γενιές γλωσσών προγραμματισμού και στα βασικά Προγραμματιστικά Υποδείγματα. Επισημάνθηκαν οι αρχές του Δομημένου Προγραμματισμού και τα πλεονεκτήματα αυτής της μεθοδολογίας συγγραφής προγραμμάτων. Επίσης αναλύθηκε η διαδικασία μετάφρασης των προγραμμάτων και οι τρόποι εύρεσης και διόρθωσης συντακτικών λαθών. Με παραδείγματα ανάπτυξης προγραμμάτων σε διάφορα προγραμματιστικά περιβάλλοντα παρουσιάστηκε η επαναχρησιμοποίηση κώδικα μέσω διαδικασιών ή συναρτήσεων. Ακόμη χρησιμοποιήθηκαν βασικοί αλγόριθμοι του προηγούμενου κεφαλαίου οι οποίοι κωδικοποιήθηκαν σε προγράμματα. Τέλος περιγράφηκε ο Κύκλος Ζωής ενός προγράμματος και έγινε διάκριση μεταξύ των εφαρμογών κλειστού και ανοικτού κώδικα.



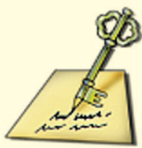
Εικόνα 2.37. Σέυμουρ Παπέρτ (Seymour Papert)

Επινόησε τη γλώσσα προγραμματισμού LOGO ως ένα τεχνολογικό εργαλείο που θα βελτιώνει τους τρόπους με τους οποίους τα παιδιά σκέπτονται και επιλύουν προβλήματα.



Εικόνα 2.38. Ρίσαρντ Φάινμαν (Richard Feynman) Βραβείο Νόμπελ Φυσικής.

«Η επιστήμη υπολογιστών δεν είναι τόσο παλιά όσο η φυσική, υστερεί χρονικά μερικούς αιώνες. Ωστόσο, αυτό δεν σημαίνει ότι υπάρχουν λιγότερα στο πιάτο του επιστήμονα των υπολογιστών απ' ό,τι σε αυτό του φυσικού: μπορεί να είναι νεότερη αλλά είχε μια πολύ πιο έντονη γέννηση!»



Λέξεις κλειδιά

Πρόγραμμα, Γλώσσες Προγραμματισμού, Προγραμματιστικό Υπόδειγμα, Δομημένος Προγραμματισμός, Συντάκτης, Μετάφραση Προγράμματος, Μεταγλωττιστής, Διερμηνευτής, Αντικείμενο Πρόγραμμα, Συνδέτης, Εκτελέσιμο Πρόγραμμα, Συντακτικά Λάθη, Διαδικασίες, Συναρτήσεις, Βιβλιοθήκες, Κύκλος Ζωής, Ανοικτό Λογισμικό

Ερωτήσεις - Θέματα προς συζήτηση - Δραστηριότητες

1. Τι είναι πρόγραμμα;
2. Ποια είναι τα πλεονεκτήματα των γλωσσών υψηλού επιπέδου σε σχέση με τις γλώσσες προηγούμενης γενιάς;
3. Ποια η διαφορά των γλωσσών του οπτικού προγραμματισμού με αυτές που διαθέτουν οπτικό περιβάλλον προγραμματισμού;
4. Τι εννοούμε με τη φράση «Προγραμματιστικό Υπόδειγμα»;
5. Τι είναι ο δομημένος προγραμματισμός και ποια είναι τα τρία κύρια χαρακτηριστικά του;
6. Ποια είναι τα πλεονεκτήματα του Δομημένου Προγραμματισμού;
7. Σε τι χρησιμεύουν τα μεταφραστικά προγράμματα;
8. Ποιες οι διαφορές του μεταγλωττιστή και του διερμηνευτή στην εύρεση των συντακτικών λαθών;
9. Τι εργαλεία πρέπει να περιέχει ένα προγραμματιστικό περιβάλλον;

- 10.** Πώς επιτυγχάνεται η επαναχρησιμοποίηση κώδικα και ποια τα πλεονεκτήματά της;
- 11.** Ποιες είναι φάσεις του Κύκλου Ζωής ενός προγράμματος;
- 12.** Ποιες δυνατότητες έχει ο χρήστης με τα Λογισμικά Ανοικτού Κώδικα;
- 13.** Να χαρακτηρίσετε με Σωστό ή Λάθος τις παρακάτω προτάσεις:
- A.** Οι εντολές στις συμβολικές γλώσσες αποτελούνται από ακολουθίες 0 και 1.
 - B.** Η μεταφερσιμότητα είναι χαρακτηριστικό των γλωσσών υψηλού επιπέδου.
 - Γ.** Η SQL αποκρύπτει τις τεχνικές του προγραμματισμού.
 - Δ.** Η γλώσσα LOGO αναπτύχθηκε για εκπαιδευτικού σκοπούς.
 - E.** Μία εφαρμογή λογισμικού με σύνθετες λειτουργίες υλοποιείται κατά βάση από μία ομάδα προγραμματιστών.
- 14.** Να επιλέξετε για κάθε μία από τις παρακάτω φράσεις το γράμμα που οδηγεί σε σωστή πρόταση:
- A.** Οι εντολές ενός προγράμματος γράφονται σε ένα πρόγραμμα που ονομάζεται:
 - i. Συντάκτης
 - ii. Μεταγλωττιστής
 - iii. Διερμηνευτής
 - iv. Συνδέτης
 - B.** Ο μεταγλωττιστής επισημαίνει:
 - i. όλα τα λάθη του προγράμματος
 - ii. μόνο τα λογικά λάθη του προγράμματος

- iii. μόνο τα συντακτικά λάθη του προγράμματος
- iv. μόνο τα λάθη που οφείλονται σε αναγραμματισμούς εντολών

Γ. Σε μία εφαρμογή ανοικτού κώδικα:

- i. δεν είναι δυνατόν να προσπελάσει ο χρήστης τον κώδικα
- ii. πρέπει ο χρήστης να περιμένει τις αναβαθμίσεις από την εταιρεία κατασκευής της.
- iii. ο χρήστης δεν μπορεί να προσθέσει λειτουργίες.
- iv. ο χρήστης μπορεί να δει τον κώδικα και να προσθέσει λειτουργίες.

15. Να πραγματοποιήσετε συζήτηση στην τάξη για το πώς θα μπορούσε να υλοποιηθεί προγραμματιστικά το υπολογιστικό πρόβλημα που περιγράφεται στο παράδειγμα 2.1. στο κεφάλαιο 2.1.

16. Να πραγματοποιήσετε συζήτηση στην τάξη σχετικά με το επάγγελμα του προγραμματιστή. Αναζητήστε πληροφορίες στη διεύθυνση <http://www.eoppep.gr>

17. Να αναπτύξετε πρόγραμμα στο προγραμματιστικό περιβάλλον Διερμηνευτής Ψευδογλώσσας (ViALGOL) το οποίο

- A.** να δέχεται τα ονόματα 20 εταιρειών καθώς και τα έσοδα τους για το προηγούμενο έτος.
- B.** να εκτυπώνει το μέσο όρο των εσόδων.
- Γ.** να εκτυπώνει το όνομα της εταιρείας με τα μεγαλύτερα έσοδα.

Δ. να διαβάζει το όνομα μιας εταιρείας και να εμφανίζει τα έσοδα της. Αν η εταιρεία δεν υπάρχει να εμφανίζει κατάλληλο μήνυμα.

18. Να αναπτύξετε κατάλληλα υποπρογράμματα για τα ερωτήματα Β και Γ της δραστηριότητας 17 και να γράψετε τον αντίστοιχο κώδικα του προγράμματος. Τα υποπρογράμματα θα πρέπει να επιστρέφουν το μέσο όρο και το όνομα της εταιρείας αντίστοιχα.



ΕΝΟΤΗΤΑ 3η

Θέματα Εφαρμοσμένης Επιστήμης των Υπολογιστών

ΚΕΦΑΛΑΙΑ

3.1. Λειτουργικά Συστήματα

3.2. Πληροφοριακά Συστήματα

3.3. Δίκτυα

3.4. Τεχνητή Νοημοσύνη

Λειτουργικά Συστήματα

Στόχος του κεφαλαίου αυτού είναι οι μαθητές να εντάξουν τις γνώσεις τους για τα Λειτουργικά Συστήματα στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών.



Προερωτήσεις

- Πώς συντονίζεται το υλικό σε ένα σύγχρονο υπολογιστικό σύστημα;
- Πώς γίνεται η διαχείριση των πληροφοριών, της μνήμης, της Κεντρικής Μονάδας Επεξεργασίας και των περιφερειακών συσκευών;
- Ποια είναι τα πιο γνωστά Λειτουργικά Συστήματα;
- Ποια είναι τα βασικά χαρακτηριστικά του Λειτουργικού Συστήματος που χρησιμοποιείτε συχνότερα;

3.1.1 Λογισμικό και Υπολογιστικό Σύστημα

Ένα σύγχρονο υπολογιστικό σύστημα αποτελείται από: το υλικό, τα ηλεκτρονικά μέρη του υπολογιστή και το λογισμικό, το σύνολο των προγραμμάτων που αξιοποιούν και διαχειρίζονται τις λειτουργίες του υλικού του υπολογιστή. Το λογισμικό χωρίζεται στο λειτουργικό σύστημα, το οποίο θα μελετηθεί στη συνέχεια και στο λογισμικό εφαρμογών, τα οποία αποτελούν το σύνολο των προγραμμάτων που επιλύουν τα προβλήματα των χρηστών.

3.1.2 Το Λειτουργικό Σύστημα και οι Αρμοδιότητές του

Λειτουργικό Σύστημα (Λ.Σ.) (Operating System – OS) είναι το σύνολο των προγραμμάτων ενός υπολογιστικού συστήματος το οποίο λειτουργεί ως σύνδεσμος ανάμεσα στα προγράμματα του χρήστη και το υλικό. Το Λ.Σ. είναι υπεύθυνο για τη δημιουργία του περιβάλλοντος επικοινωνίας του χρήστη με το σύστημα, τη διαχείριση και το συντονισμό των εργασιών του συστήματος, καθώς και για την κατανομή των διαθέσιμων πόρων.

Τα Λ.Σ. παρέχουν ένα περιβάλλον στο οποίο εκτελούνται διάφορα προγράμματα, τα οποία στοχεύουν στην ομαλή λειτουργία του υπολογιστικού συστήματος. Οι βασικές αρμοδιότητες ενός Λ.Σ. είναι να:

- **Λειτουργεί ως ενδιάμεσος μεταξύ του ανθρώπου και της μηχανής, μεταφέροντας εντολές ή απαιτήσεις του χρήστη στο υπολογιστικό σύστημα.**
- **Διαχειρίζεται τους διαθέσιμους πόρους και να τους κατανέμει στις διάφορες διεργασίες.**
- **Ελέγχει την εκτέλεση των προγραμμάτων.**
- **Διαχειρίζεται τη λειτουργία των συσκευών εισόδου και εξόδου και να ελέγχει τη ροή των δεδομένων και την έξοδο των πληροφοριών.**
- **Οργανώνει και να διαχειρίζεται τα αρχεία του συστήματος.**
- **Ανιχνεύει και να εντοπίζει πιθανά λάθη ή δυσλειτουργίες του υπολογιστικού συστήματος και να ενημερώνει τον χρήστη.**
- **Εφαρμόζει μηχανισμούς που βελτιώνουν την ασφάλεια του υπολογιστικού συστήματος από διάφορους κινδύνους.**

Κατά την εξέλιξή τους έχουν εμφανιστεί διάφορες κατηγορίες Λ.Σ.: Τα Λ.Σ. ομαδικής επεξεργασίας (1η γενιά - δεκαετία '50), πολυπρογραμματισμού (2η γενιά - δεκαετία '60), καταμερισμού χρόνου (3η γενιά - δεκαετία '70) και κατανεμημένης επεξεργασίας (4η γενιά - δεκαετία '80 μέχρι σήμερα).

3.1.3 Η Δομή και η Ιεραρχία του Λειτουργικού Συστήματος

Τα σύγχρονα Λ.Σ. είναι δομημένα σε ιεραρχικά τοποθετημένα επίπεδα (layers). Κάθε επίπεδο εκτελεί μια συγκεκριμένη εργασία και συνεργάζεται με τα δύο γειτονικά του. Στα κατώτερα επίπεδα γίνεται η διαχείριση της μνήμης και της επικοινωνίας με τις περιφερειακές συσκευές του υπολογιστή, ενώ στα ανώτερα γίνεται η διαχείριση των προγραμμάτων που εκτελούν οι χρήστες. Σε ένα Λ.Σ. υπάρχουν τα ακόλουθα επίπεδα (Εικόνα 3.1):



Εικόνα 3.1. Διαστρωμάτωση ενός Λ.Σ.

- Ο Πυρήνας (Kernel), βρίσκεται πλησιέστερα προς το υλικό και αποτελεί τον ενδιάμεσο για να επιτευχθεί η επικοινωνία των προγραμμάτων με το υλικό. Ο πυρήνας «φορτώνεται» πρώτος στην κύρια μνήμη όταν ξεκινάει ο υπολογιστής.
- Το Σύστημα Αρχείων (File System) διαχειρίζεται τα αρχεία (δίνοντάς τους ονομασία, καταχωρώντας τα, κτλ.) και φροντίζει για τη διάθεσή τους στους χρήστες.
- Ο Διερμηνευτής Εντολών (Command Interpreter) ή Φλοιός (Shell) είναι το σύνολο των προγραμμάτων, το οποίο επιτρέπει στο χρήστη και τις εφαρμογές του να επικοινωνεί με το Λ.Σ. Η επικοινωνία γίνεται είτε με απευθείας εντολές (command mode - Εικόνα 3.2) είτε μέσω ενός γραφικού περιβάλλοντος διεπαφής (GUI - Graphical User Interface – Εικόνα 3.3).

```

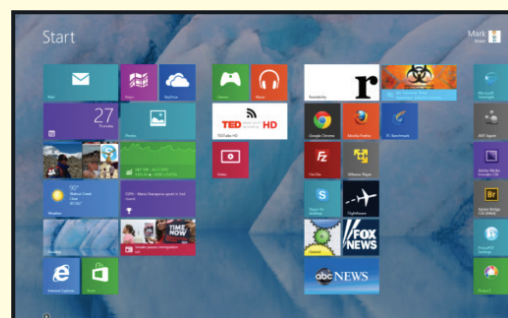
C:\WINDOWS\system32>dir \test
Volume in drive C has no label.
Volume Serial Number is 1643-0CD7

Directory of C:\test

17/10/2004  08:06 PM  <DIR>          .
17/10/2004  08:06 PM  <DIR>          ..
17/10/2004  08:08 PM                4,096 a.txt
17/10/2004  08:07 PM                 27 b.txt
17/10/2004  08:07 PM                1,826 c.txt
17/10/2004  08:10 PM               66,126 d.txt
               4 File(s)              72,075 bytes
               2 Dir(s)    11,792,990,208 bytes free

C:\WINDOWS\system32>_

```



Εικόνα 3.2.

Επικοινωνία με απευθείας εντολές (Λ.Σ. MS DOS)

Εικόνα 3.3.

Επικοινωνία με γραφικό περιβάλλον Διεπαφής (Λ.Σ. MS Windows)

3.1.4 Βασικές Εργασίες ενός Λ.Σ.

Εργασίες ενός Λ.Σ. αποτελούν η Διαχείριση της Κεντρικής Μονάδας Επεξεργασίας (Κ.Μ.Ε.), η Διαχείριση της Κεντρικής Μνήμης, η Διαχείριση του Συστήματος Αρχείων και η Διαχείριση των Λειτουργιών Εισόδου/ Εξόδου.

3.1.4.1 Διαχείριση της Κ.Μ.Ε.

Το χαρακτηριστικό των σύγχρονων Λ.Σ. είναι ο πολυπρογραμματισμός (multiprogramming) και η πολυδιεργασία (multitasking). Στην πρώτη περίπτωση το Λ.Σ. μπορεί να εκτελεί ταυτόχρονα περισσότερα από ένα προγράμματα (για παράδειγμα, την ώρα που η Κ.Μ.Ε. περιμένει απάντηση από μια περιφερειακή συσκευή, αυτός ο χρόνος «αναμονής» μπορεί να αξιοποιηθεί από ένα άλλο πρόγραμμα που είναι φορτωμένο στην κύρια μνήμη), ενώ στη δεύτερη περίπτωση το Λ.Σ. μπορεί να εκτελεί ταυτόχρονα περισσότερες από μία εργασίες (για παράδειγμα, μπορεί να γίνονται παράλληλα εκτυπώσεις και υπολογισμοί). Η όλη διαδικασία βασίζεται σε έναν αλγόριθμο χρονοπρογραμματισμού, ο οποίος στοχεύει στη μεγιστοποίηση της αποδοτικότητας και της «δίκαιης» χρήσης της Κ.Μ.Ε. από το μέγιστο αριθμό των επεξεργαζόμενων διεργασιών.



Μια από τις βασικές έννοιες στα Λ.Σ. είναι η έννοια της **διεργασίας** (process).

Μια διεργασία είναι ένα πρόγραμμα ή ένα αυτόνομο τμήμα προγράμματος υπό εκτέλεση από το σύστημα.

Το πρόγραμμα διαφέρει από τη διεργασία στο ότι είναι μικρότερο, μιας και η διεργασία περιέχει, εκτός από το πρόγραμμα, και στοιχεία συστήματος για την εκτέλεση του κώδικα.

3.1.4.2 Διαχείριση της Μνήμης

Η πολυδιεργασία προϋποθέτει ότι στην κεντρική μνήμη είναι φορτωμένα περισσότερα του ενός προγράμματα προς εκτέλεση από την Κ.Μ.Ε. Θα πρέπει, λοιπόν, να γίνει η διαχείριση της κύριας μνήμης με τέτοιο τρόπο, ώστε να επιτευχθεί ο αποτελεσματικός διαμοιρασμός της μεταξύ των διαφόρων προγραμμάτων. Το τμήμα του Λ.Σ. που διαχειρίζεται την κύρια μνήμη είναι ο **διαχειριστής μνήμης** (memory manager). Οι εργασίες που επιτελεί ο διαχειριστής μνήμης είναι:

- Η διάθεση τμημάτων μνήμης σε διεργασίες.
- Η παρακολούθηση της κατάστασης χρήσης της μνήμης, ώστε να γνωρίζει τα ελεύθερα ή μη τμήματα κάθε στιγμή και να τα διανέμει σε διεργασίες.
- Η ελευθέρωση μνήμης από διεργασίες που δεν τη χρειάζονται.

- Η ανταλλαγή (swapping) δεδομένων μεταξύ της κύριας μνήμης και της περιοχής του δίσκου (περιφερειακή μνήμη) που χρησιμοποιείται ως βοηθητική περιοχή της κύριας μνήμης.



Κάθε πρόγραμμα προτού εκτελεστεί από την κεντρική μνήμη βρίσκεται στην περιφερειακή μνήμη (σκληρός δίσκος), ενώ κατά τη διάρκεια εκτέλεσής του στην κεντρική μνήμη το πρόγραμμα διαβάζει από αυτήν και γράφει δεδομένα σε αυτήν.

3.1.4.3 Διαχείριση του Συστήματος Αρχείων

Το σύστημα αρχείων είναι το μέρος του Λ.Σ. με το οποίο ο χρήστης έρχεται σε άμεση επαφή. Το Λ.Σ. συνήθως οργανώνει τα αρχεία του σε καταλόγους ή φακέλους (directories ή folders). Κάθε κατάλογος αποτελείται από αρχεία, υποκαταλόγους ή υποφακέλους, δημιουργώντας μία δενδροειδή μορφή.

Ένα σύστημα αρχείων του Λ.Σ. προσφέρει στον χρήστη ένα εικονικό περιβάλλον διαχείρισης, το οποίο του δίνει τη δυνατότητα να εκτελεί μία σειρά από πράξεις όπως η δημιουργία (με προσδιορισμό ονόματος και τύπου), η διαγραφή, η μετονομασία, η αντιγραφή και το κλείσιμο αρχείων. Επιπλέον, είναι δυνατή η τροποποίηση του περιεχομένου ή η αντιγραφή του περιεχομένου ενός αρχείου σε ένα άλλο.



Σ' έναν υπολογιστή οι πληροφορίες αποθηκεύονται σε περιφερειακές μονάδες (π.χ. σε σκληρούς δίσκους) όπου τα στοιχεία διατηρούνται και μετά τη διακοπή της παροχής ρεύματος. Εκεί αποθηκεύονται τα στοιχεία ως συλλογές δεδομένων (αρχεία-files), τη διαχείριση των οποίων αναλαμβάνει μέρος του Λ.Σ. που καλείται **Σύστημα Αρχείων (file system)**.

3.1.4.4 Διαχείριση Λειτουργιών Εισόδου/Εξόδου

Το τμήμα του Λ.Σ. το οποίο ασχολείται με τις διαδικασίες εισόδου/εξόδου μεταξύ του κεντρικού μέρους και των εξωτερικών προς αυτό συσκευών εξασφαλίζει τη διασύνδεση των συσκευών με το κεντρικό σύστημα και την ομαλή επικοινωνία. Επίσης, αναλαμβάνει να διαχειριστεί τις εντολές που εκτελούνται και τα σφάλματα που παρουσιάζονται.

Στις αρμοδιότητες του συγκεκριμένου μέρους του Λ.Σ. είναι η αποδοτική διαχείριση των περιφερειακών μονάδων και ο ορισμός της σειράς ικανοποίησης των διαφόρων δραστηριοτήτων, όπως των αιτημάτων εγγραφής ή ανάγνωσης. Έτσι, οι συσκευές διακρίνονται σε **διαμοιραζόμενες (shared)** και **αποκλειστικές (dedicated)**. Τις διαμοιραζόμενες συσκευές μπορούν να τις χρησιμοποιούν πολλοί χρήστες ταυτόχρονα, όπως οι δίσκοι,

ενώ τις αποκλειστικές μπορεί να τις χρησιμοποιεί ένας χρήστης κάθε στιγμή, όπως οι εκτυπωτές οι οποίοι κατά τη διάρκεια της χρήσης τους δεν επιτρέπεται να χρησιμοποιούνται από άλλο πρόγραμμα.



Με τον όρο **Είσοδος (Input)** αναφερόμαστε στη ροή δεδομένων προς την Κ.Μ.Ε. ενώ με τον όρο **Έξοδος (Output)** αναφερόμαστε στη ροή δεδομένων από την Κ.Μ.Ε. προς τις περιφερειακές συσκευές.

3.1.5 Γνωστά Λειτουργικά Συστήματα

Έχουν αναπτυχθεί διάφορα Λ.Σ. τα οποία αξιοποιούν κατάλληλα την αρχιτεκτονική των επεξεργαστών. Στη συνέχεια παρατίθενται τα πιο γνωστά από αυτά.

Το **Unix** αναπτύχθηκε στα εργαστήρια Bell Labs της AT&T το 1969. Το μεγαλύτερο μέρος του έχει αναπτυχθεί σε γλώσσα προγραμματισμού C. Το Unix αποτελεί ένα Λ.Σ. πολυδιεργασίας, καταμερισμού χρόνου (timesharing), πολλών χρηστών (multiuser), και με φορητότητα (portability) - δηλαδή εύκολη εγκατάσταση σε διαφορετικού τύπου συστήματα.

Το **MS-DOS** (Microsoft Disk Operating System) είναι Λ.Σ. στο οποίο οι εντολές του χρήστη πληκτρολογούνται σε μία γραμμή. Αναπτύχθηκε από την εταιρεία Microsoft το 1981 για λογαριασμό της εταιρείας IBM (Εικόνα 3.2).

Τα **MS Windows** της Microsoft διαθέτουν γραφικό περιβάλλον διεπαφής χρήστη παραθυρικού τύπου (Εικόνα 3.3). Πρόκειται για Λ.Σ. πολλαπλών διεργασιών και πολλαπλών χρηστών. Από το 1985 μέχρι σήμερα έχουν αναπτυχθεί διάφορες εκδόσεις των Windows (π.χ. 95, 98, XP, 7, 8.1 κ.α.).

Το **Linux** είναι ένα λειτουργικό σύστημα τύπου Unix, το οποίο δημιουργήθηκε από τον Linus Torvalds το 1991. Ο πηγαίος κώδικάς του είναι «ανοικτός», με αποτέλεσμα να αναπτύσσεται συνεχώς από μια μεγάλη κοινότητα χρηστών (Εικόνα 3.4)

Το **Mac OS X** έχει ως βάση το Λ.Σ. Unix και διανέμεται αποκλειστικά για χρήση με τους υπολογιστές Mac της εταιρείας Apple. Το Mac OS X είναι η εξέλιξη του Mac OS το οποίο ήταν το αρχικό Λ.Σ. της Apple (1984-1999). Τα OS X διακρίνονται για τα πρωτότυπα γραφικά περιβάλλοντα διεπαφής τους (Εικόνα 3.5).

Το **Android** χρησιμοποιείται κυρίως σε συσκευές κινητής τηλεφωνίας και tablets και τρέχει τον πυρήνα του Λ.Σ. Linux. Αρχικά αναπτύχθηκε από την Google (2007) και αργότερα από την Open Handset Alliance (Εικόνα 3.6).



Εικόνα 3.4.

Το Λ.Σ. Linux είναι ΕΛ/ΛΑΚ και μπορεί να τρέξει μέσα από οπτικό δίσκο (LiveCD) χωρίς να απαιτείται μόνιμη εγκατάσταση.



Εικόνα 3.5.
Λ.Σ. OS X της Apple.



Εικόνα 3.6.
Λ.Σ. Android της Google



Ανακεφαλαίωση

Ένα Λ.Σ αποτελείται από ένα σύνολο προγραμμάτων τα οποία ελέγχουν ένα υπολογιστικό σύστημα, και λειτουργεί ως ενδιάμεσος ανάμεσα στο σύστημα και τον χρήστη. Τα χαρακτηριστικά των συγχρόνων Λ.Σ. είναι ο πολυπρογραμματισμός και η πολυδιεργασία. Έχουν

αναπτυχθεί διάφορα Λ.Σ. τα οποία εξελίσσονται παράλληλα με την εξέλιξη της αρχιτεκτονικής των σύγχρονων επεξεργαστών.

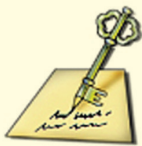
Χρήσιμοι Υπερσύνδεσμοι

<http://www.android.com>

Διαδικτυακός τόπος για το Λ.Σ. Android.

<http://www.linux.gr/news>

Διαδικτυακός τόπος της ελληνικής κοινότητας Linux.



Λέξεις κλειδιά

Διεπαφή, Σύστημα Αρχείων, Διαχείριση Κ.Μ.Ε., Είσοδος/Εξοδος, Πυρήνας.

Ερωτήσεις - Δραστηριότητες - Θέματα προς συζήτηση

1. Ποιο ρόλο επιτελεί το Λειτουργικό Σύστημα σε έναν υπολογιστή;
2. Να αναφέρετε ονομαστικά τις μεθόδους διαχείρισης των περιφερειακών συσκευών από το Λ.Σ.

- 3. Δείτε ποια Λ.Σ. διαθέτει το εργαστήριο υπολογιστών του σχολείου σας και εντοπίστε ομοιότητες και διαφορές.**
- 4. Αν στην τάξη σας βρίσκονται μαθητές με διαφορετική καταγωγή, μελετήστε τις ασυμβατότητες των Λ.Σ. μεταξύ διαφορετικών γλωσσών.**
- 5. Αναζητήστε το LiveCD του Linux και δοκιμάστε το σε κάποιους υπολογιστές του σχολικού εργαστηρίου.**
- 6. Αναζητήστε πληροφορίες για Λ.Σ. τα οποία είναι φιλικά στη χρήση τους από Άτομα με Αναπηρία (ΑμεΑ).**
- 7. Ποια από τα παρακάτω αποτελούν εργασίες ενός Λ.Σ.;**
 - A. Εκτύπωση**
 - B. Διαχείριση μνήμης**
 - Γ. Επεξεργασία κειμένου**
 - Δ. Διαχείριση Κ.Μ.Ε.**

Πληροφοριακά Συστήματα

Στόχοι του κεφαλαίου είναι οι μαθητές:

- ✓ Να εντάξουν τις γνώσεις τους για θέματα σχετικά με τη διαχείριση δεδομένων, τη δημιουργία, την αποθήκευση και την ανάκτηση πληροφοριών στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών.
- ✓ Να αιτιολογούν ότι τα δεδομένα αποθηκεύονται σε οργανωμένες δομές και ανακτώνται μέσω συγκεκριμένων συστημάτων και μεθοδολογιών.



Προερωτήσεις

- Ποια είναι τα στάδια υλοποίησης μιας παραγγελίας μέσω διαδικτύου;
- Πώς επιδρά η εξάπλωση του Διαδικτύου και των δικτύων στη λειτουργία ενός οργανισμού ή μιας επιχείρησης;
- Πώς εκτελούνται οι λειτουργίες παραγγελιών, παραγωγής και μεταφοράς προϊόντων σε συνθήκες παγκοσμιοποίησης;

3.2.1 Τι είναι τα Πληροφοριακά Συστήματα

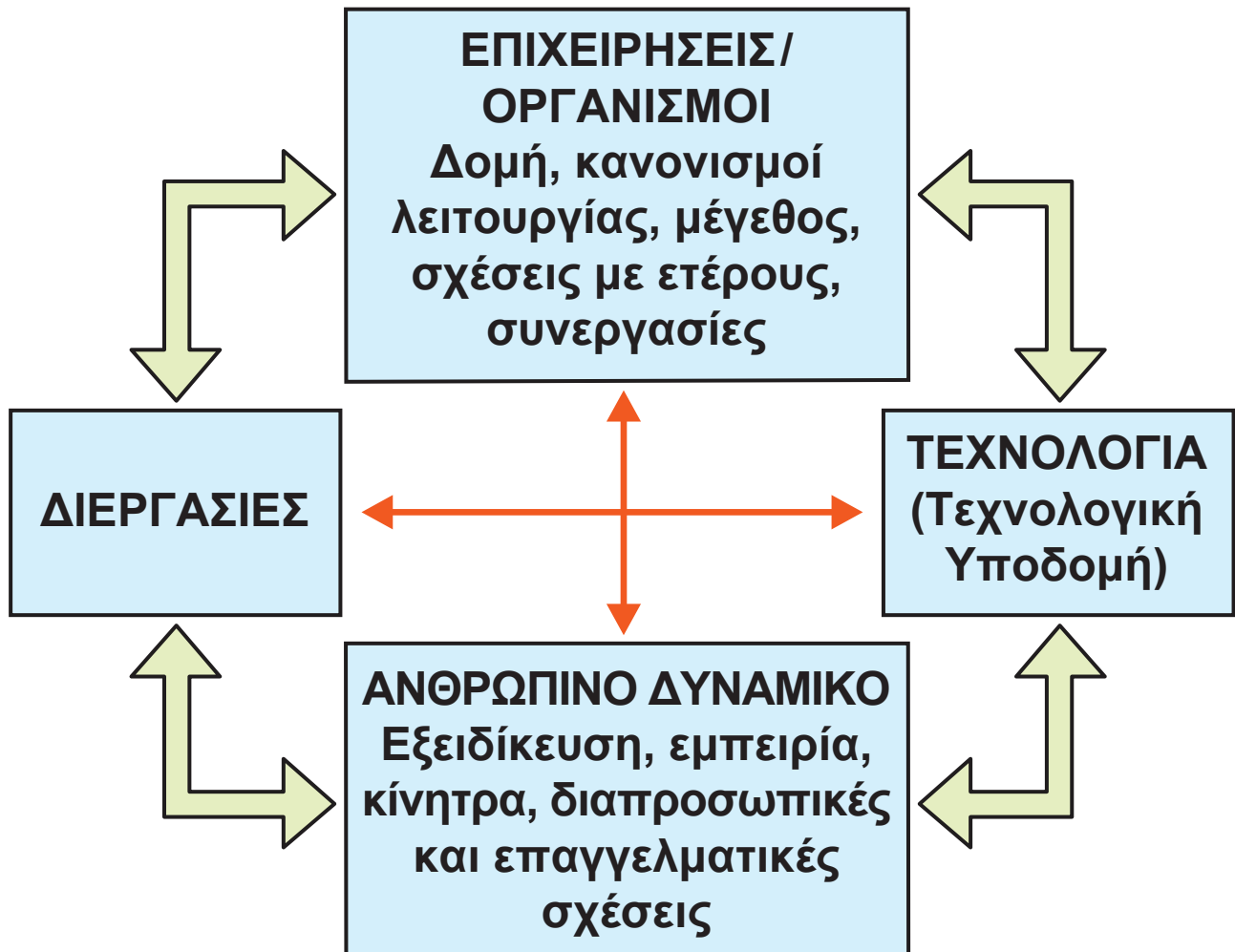
Τα τελευταία χρόνια έχουν διαμορφωθεί νέες απαιτήσεις για θέματα που αφορούν τη διαχείριση των δεδομένων, τη δημιουργία, αποθήκευση, ανάκτηση και χρήση της πληροφορίας. Οι απαιτήσεις αυτές οδηγούν στην ανάπτυξη βάσεων δεδομένων και νέων πληροφοριακών αρχιτεκτονικών. Τα Πληροφοριακά Συστήματα και οι Βάσεις Δεδομένων εντάσσονται στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών.

Πληροφοριακό σύστημα (Π.Σ.) ονομάζεται ένα σύνολο αλληλοσυσχετιζόμενων και αλληλεπιδρώντων οντοτήτων που συλλέγουν, επεξεργάζονται, αποθηκεύουν, ανακτούν και διανέμουν πληροφορίες για την υποστήριξη των αποφάσεων και ελέγχου σε μια επιχείρηση ή σε έναν οργανισμό.



Η οντότητα (**entity**) είναι μία αυτόνομη μονάδα του φυσικού κόσμου, για παράδειγμα ένα αντικείμενο, ένα πρόσωπο, μία κατάσταση και γενικά οτιδήποτε μπορεί να προσδιοριστεί ως ανεξάρτητη ύπαρξη. Για παράδειγμα, σε ένα σχολείο, οντότητες μπορεί να είναι τα ονόματα ή τα στοιχεία των μαθητών, των καθηγητών, οι αίθουσες διδασκαλίας, τα βιβλία κάθε τάξης, οι βαθμολογίες κ.ά.

Ένα Π.Σ. λειτουργεί μέσα σε ένα περιβάλλον, επηρεάζεται από αυτό και το επηρεάζει. Μέσα σ' αυτό λειτουργεί και το ανθρώπινο δυναμικό με την τεχνολογική υποδομή που διαθέτει (Εικόνα 3.7).

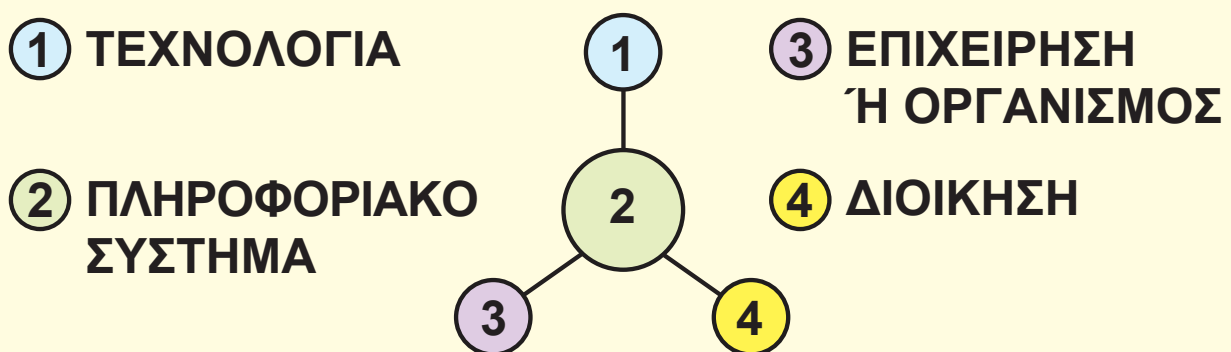


Εικόνα 3.7. Το μοντέλο αλληλεπίδρασης ενός Π. Σ.



Για μια επιχείρηση που διανέμει διάφορα είδη προϊόντων ο προγραμματισμός των εργασιών της, η παράδοση των αγαθών και τα έξοδα διανομής καθορίζουν σε μεγάλο βαθμό την κερδοφορία της. Έτσι, η εφοδιαστική αλυσίδα (**logistics**) παίζει σπουδαίο ρόλο στις δραστηριότητές της. Η αξιοποίηση ενός Π.Σ. γύρω από αυτά τα ζητήματα είναι καθοριστικής σημασίας για την επιχείρηση.

Κάθε Π.Σ. αποτελείται από **Δεδομένα** ή **Διαδικασίες** που αφορούν την επιχείρηση ή τον οργανισμό, **Τεχνολογίες Πληροφορικής** (υλικό και λογισμικό) και **Ανθρώπους** (ανθρώπινο δυναμικό και διοίκηση) (Εικόνα 3.8).



Εικόνα 3.8.
Τα συστατικά μέρη ενός Π.Σ.

Σε ένα Π.Σ. συντελείται: **Συλλογή Δεδομένων, Αποθήκευση Δεδομένων, Επεξεργασία Δεδομένων και Παρουσίαση της Πληροφορίας**. Η διαχείριση των δεδομένων ενός Π.Σ. είναι σημαντική για έναν οργανισμό ή μια επιχείρηση. Για το λόγο αυτό απαιτείται η δημιουργία ειδικής υπηρεσίας που θα καθορίζει α) τις ανάγκες και β) την δυνατότητα πρόσβασης σε δεδομένα και πληροφορίες.

3.2.2 Αρχιτεκτονικές Αποθήκευσης

Κάθε επιχείρηση ακολουθεί διαφορετικούς τρόπους αποθήκευσης των δεδομένων σύμφωνα με τις ανάγκες της και τις τεχνολογίες που μπορεί να υποστηρίξει. Οι αρχιτεκτονικές αποθήκευσης κατηγοριοποιούνται ανάλογα με τις μεθόδους που χρησιμοποιούν. Στις **αρχιτεκτονικές αποθήκευσης άμεσης πρόσβασης** είναι δυνατή η άμεση πρόσβαση στα δεδομένα ή τα εμπορεύματα μέσω συσκευών εισόδου. Οι καταχωρήσεις στο σύστημα αποθήκευσης μπορούν να γίνονται με συσκευές αναγνώρισης (Εικόνα 3.9). Στις **Αρχιτεκτονικές αποθήκευσης βασισμένες σε δίκτυο** η πρόσβαση στα δεδομένα γίνεται μέσω ενός δικτύου υπολογιστών, όπου τα δεδομένα αποθηκεύονται είτε τοπικά είτε σε κάποια άλλη τοποθεσία της επιχείρησης. Έτσι δίνεται η δυνατότητα απομακρυσμένης πρόσβασης σε όλα τα δεδομένα. Οι **Αρχιτεκτονικές αποθήκευσης βασισμένες στο σύννεφο** στηρίζονται στο Διαδίκτυο και χρησιμοποιούν τις υποδομές ενός παρόχου υπηρεσιών σύννεφου (**Cloud Service Provider, CSP**). Περιλαμβάνουν λειτουργικό σύστημα για υπηρεσίες σύννεφου και τις απαραίτητες

διαδικτυακές υπηρεσίες που εξασφαλίζουν τη διασυνδεσιμότητα και τη διαλειτουργικότητα. Για τη συγκεκριμένη αρχιτεκτονική υπάρχει η δυνατότητα αύξησης σε πραγματικό χρόνο του αποθηκευτικού χώρου και διατίθενται υπηρεσίες που επιτρέπουν την αποθήκευση μεγάλων ποσοτήτων μη δομημένης πληροφορίας, η οποία είναι προσβάσιμη μέσω του διαδικτύου, χωρίς την ανάγκη τοπικής αποθήκευσης και επεξεργασίας.



Εικόνα 3.9.

Συσκευή γραμμοκώδικα με χρήση πλακέτας Arduino.

Οι σύγχρονοι εξυπηρετητές διαθέτουν πλατφόρμα δεδομένων άμεσης πρόσβασης για διαχείριση και δόμηση επιχειρησιακών λύσεων βασισμένες στο σύννεφο.

Σήμερα, κάθε λογαριασμός μπορεί να περιέχει έως και 100TB από Blobs (Binary Large Object). Ένα Blob μπορεί να έχει μέγεθος εκατοντάδων gigabyte.

3.2.3 Βάσεις Δεδομένων

Η αποτελεσματική και αποδοτική διαχείριση των δεδομένων ενός Π.Σ. απαιτεί την οργάνωσή τους με δομημένο τρόπο. Για το σκοπό αυτό χρησιμοποιούνται **Βάσεις Δεδομένων (Data Bases, D.B.)** που αποτελούν το λειτουργικό πυρήνα του Π.Σ.. Σε μία Β.Δ. καταγράφονται ολοκληρωμένες συλλογές δεδομένων που συσχετίζονται (Εικόνα 3.10).



Εικόνα 3.10.
Μία Β.Δ. με τρεις συλλογές δεδομένων.

Βάση Δεδομένων (Β.Δ.) είναι μία οργανωμένη συλλογή από συσχετιζόμενα δεδομένα, επεξεργασμένα και αποθηκευμένα με τέτοιο τρόπο, ώστε να μπορούν να χρησιμοποιούνται σε όλες τις εφαρμογές ενός οργανισμού ή μιας επιχείρησης.

Η πιο συνηθισμένη μορφή Β.Δ. είναι οι **Σχισιακές Βάσεις Δεδομένων (Σ.Β.Δ.)**, όπου τα δεδομένα οργανώνονται σε συσχετιζόμενους πίνακες με βάση το **Σχισιακό Μοντέλο Δεδομένων**. Για παράδειγμα, στην εικόνα 3.11 το κοινό στοιχείο των 2 πινάκων είναι ο κωδικός του μαθήματος.

Στιγμιότυπο Πίνακα Μαθημάτων

Κωδικός_Μαθήματος	Όνομα_Μαθήματος	Διδακτικές_Ώρες
20	Εφαρμογές Πληροφορικής	2
30	Εισαγωγή στην Επιστήμη των Η/Υ	1
...

Στιγμιότυπο Πίνακα Καθηγητών

Αριθμός_Μητρώου	Επώνυμο	Όνομα	...	Κώδικας_Μαθήματος
7329	Επώνυμο1	Όνομα1		20
7499	Επώνυμο2	Όνομα2		30
7384	Επώνυμο3	Όνομα3		20
...

Εικόνα 3.11. Σχεσιακό Μοντέλο Δεδομένων
με ένα κοινό στοιχείο.

Σχεσιακό Μοντέλο Δεδομένων (Relation Data Model, RDM) είναι το είδος του λογικού μοντέλου βάσεων δεδομένων που μπορεί να συνδυάζει τα δεδομένα ενός πίνακα με τα δεδομένα ενός άλλου, αρκεί οι δυο πίνακες να έχουν ένα κοινό στοιχείο δεδομένων.

Η διαχείριση των συσχετιζόμενων δεδομένων μίας Β.Δ. γίνεται με ένα σύνολο κατάλληλων προγραμμάτων το οποίο καλείται Σύστημα Διαχείρισης της Βάσης Δεδομένων - ΣΔΒΔ (Database Management System, DBMS) (Εικόνα 3.12).



Εικόνα 3.12. Δεδομένα και Σ.Δ.Β.Δ.

Τα συστήματα διαχείρισης βάσεων δεδομένων αλληλεπιδρούν με τον χρήστη μέσα από γλώσσες ερωταποκρίσεων.

```
SELECT ENAME, JOB, SAL  
FROM EMPLOYES WHERE  
DEPTNO = 20  
AND SAL > 1000;
```

Με την ερώτηση αυτή σε SQL εκτελείται αναζήτηση στη βάση δεδομένων EMPLOYES και επιστρέφει το όνομα, τη θέση και τον μισθό των υπαλλήλων της διεύθυνσης 20 που κερδίζουν πάνω από 1000 ευρώ.

3.2.4 Γλώσσες Ερωταποκρίσεων (SQL, XML)

Μία από τις δημοφιλέστερες γλώσσες ερωταποκρίσεων είναι η **SQL (Structured Query Language, Δομημένη Γλώσσα Ερωταποκρίσεων)**, η οποία αναπτύχθηκε το 1971 από την IBM.

Άλλη μια σημαντική γλώσσα ερωταποκρίσεων είναι η **XML (Extensible Markup Language, Επεκτάσιμη Γλώσσα Σήμανσης)**. Η XML χρησιμοποιώντας ένα σύνολο κανόνων καθιστά εφικτή τη δημιουργία οποιασδήποτε ετικέτας απαιτείται για την περιγραφή των δεδομένων και της δομής τους (Εικόνα 3.13).

```
<?xml version="1.0"encoding="UTF-8"?>
<library>
<book>
<title>Networks</title>
<copies>15</copies>
</book>
<book>
<title>Algorithms</title>
<copies>8</copies>
</book>
</library>
```

Εικόνα 3.13. Τμήμα XML κειμένου για μία δανειστική βιβλιοθήκη. //library/book[copies>10]/title

Με αυτή την έκφραση διαδρομής επιλέγονται και επιστρέφονται οι τίτλοι των βιβλίων της βιβλιοθήκης με περισσότερα από 10 αντίτυπα. Από την παραπάνω βιβλιοθήκη θα επιστραφεί ο τίτλος Networks.

Η δομή των δεδομένων όπως αυτά αναπαρίστανται από την XML, οπτικά έχει τη μορφή ενός ανάποδου δέντρου το οποίο ξεκινάει από μια ρίζα και εκτείνεται προς τα κάτω με πολλαπλά κλαδιά. Η XML περιλαμβάνει τη γλώσσα ερωταποκρίσεων “XPath”, η οποία επιτρέπει την διατύπωση ερωτημάτων σε μορφή έκφρασης διαδρομής ή μονοπατιού (path expression) και αποκρίνεται επιστρέφοντας τα δεδομένα τα οποία πληρούν τα κριτήρια που έχει θέσει ο χρήστης.



Ανακεφαλαίωση

Στα Π.Σ. η οργάνωση και η διαχείριση δεδομένων σε επιχειρησιακή βάση έχει εμπλουτιστεί με τη χρήση βάσεων δεδομένων και τεχνικές βασισμένες στο δίκτυο. Η αξιοποίηση των δυνατοτήτων του σύννεφου (cloud) επιτρέπει την ανάπτυξη ενός οργανισμού ή μιας επιχείρησης χωρίς επενδύσεις σε υπολογιστική υποδομή. Οι γλώσσες ερωταποκρίσεων ενός συστήματος διαχείρισης βάσεων δεδομένων όπως η SQL και η XML αξιοποιούνται σε ένα ευρύ φάσμα επιχειρησιακών εφαρμογών.

Χρήσιμοι Υπερσύνδεσμοι

<http://www.w3.org/standards/xml/core>

Διαδικτυακός τόπος για τις προδιαγραφές xml.

<http://www.w3schools.com/XPath/>

Διαδικτυακός τόπος των προδιαγραφών Xpath.



Λέξεις κλειδιά

Αρχιτεκτονικές Αποθήκευσης, Γλώσσες Ερωταποκρίσεων, Συστήματα αποθήκευσης πληροφοριών, Γλώσσες Σήμανσης, Δικτυακές Σχεσιακές Βάσεις Δεδομένων.

Ερωτήσεις - Θέματα προς συζήτηση - Δραστηριότητες

1. Οργανώστε δεδομένα και πληροφορίες που αφορούν την τάξη σας ή το σχολείο σας (πόσοι μαθητές, πόσες μαθήτριες, πόσες τάξεις) και διαμορφώστε ένα μικρό αληθινό Π.Σ. που να αφορά τη ζωή στο σχολείο.
2. Ενημερωθείτε από συμμαθητές σας για τα συστήματα οργάνωσης των δημοσίων οργανισμών άλλων χωρών.
3. Ποια τα πλεονεκτήματα της χρήσης και της αξιοποίησης της τεχνικής αποθήκευσης δεδομένων στο σύννεφο;

- 4. Με ποιες γλώσσες προγραμματίζονται τα σύγχρονα περιβάλλοντα των Πληροφοριακών Συστημάτων;**
- 5. Επισκεφτείτε μια επιχείρηση της περιοχής σας και ενημερώστε τους συμμαθητές σας για τη διαδικασία οργάνωσής της. Πώς οργανώνονται οι προμήθειές της, πώς γίνεται η διακίνηση των εμπορευμάτων ή των υπηρεσιών που παρέχει και πώς έχει οργανωθεί το τμήμα του λογιστηρίου.**
- 6. Να χαρακτηρίσετε με Σωστό ή Λάθος τις παρακάτω προτάσεις:**
 - A. Ένα Π.Σ. λειτουργεί μέσα σε ένα περιβάλλον που το επηρεάζει και από το οποίο επηρεάζεται.**
 - B. Στις αρχιτεκτονικές αποθήκευσης βασισμένες στο σύννεφο είναι αναγκαία η τοπική αποθήκευση των δεδομένων.**
 - Γ. Η διαχείριση των συσχετιζόμενων δεδομένων μίας Β.Δ. γίνεται μέσω του συστήματος διαχείρισης της Β.Δ..**
 - Δ. Μέσω του σχεσιακού μοντέλου δεδομένων συνδυάζονται τα δεδομένα ενός πίνακα με τα δεδομένα ενός άλλου, αρκεί να έχουν ένα κοινό στοιχείο δεδομένων.**
 - Ε. Τα Π.Σ. αναπτύσσονται με σκοπό την υποστήριξη των αποφάσεων και τον έλεγχο μιας επιχείρησης.**



Στόχοι του κεφαλαίου είναι:

- ✓ Οι μαθητές να εντάξουν τις γνώσεις τους για θέματα επικοινωνίας και δικτύωσης συστημάτων στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών.
- ✓ Οι μαθητές να οργανώσουν σε νοητικό μοντέλο τα βασικά θέματα που αφορούν τα δίκτυα επικοινωνίας.



Προερωτήσεις

- Πώς επιτυγχάνεται η επικοινωνία των υπολογιστικών συστημάτων;
- Ποια υποδομή απαιτείται για να επιτευχθεί η επικοινωνία μεταξύ των διαφόρων τμημάτων σε ένα Πληροφοριακό Σύστημα;
- Ποιες συσκευές χρησιμοποιούνται στη δόμηση ενός δικτύου;
- Ποιες είναι οι σύγχρονες υπηρεσίες των δικτύων υπολογιστών;

3.3.1 Τι είναι ένα Δίκτυο Υπολογιστών

Η ραγδαία εξέλιξη καθώς και η σύγκλιση των υπολογιστικών με τα τηλεπικοινωνιακά συστήματα έχει επιφέρει σημαντικές αλλαγές στον τρόπο με τον οποίο επικοινωνούν και συναλλάσσονται οι άνθρωποι. Η απανταχού πρόσβαση σε άμεσα διαθέσιμους υπολογιστικούς πόρους και ασύρματες συσκευές, καθώς και η ανάπτυξη συνδέσεων μεγάλης ταχύτητας διευκολύνει τη συνεργασία και τις κοινωνικές αλληλεπιδράσεις. Τα δίκτυα υπολογιστών, με κύριο εκπρόσωπο το Διαδίκτυο (Internet) παρέχουν την απαραίτητη υποδομή για την επίτευξη της Κοινωνίας της Πληροφορίας.

Δίκτυο Υπολογιστών ή Δίκτυο Επικοινωνιών είναι ένα σύνολο συνδεδεμένων μεταξύ τους συσκευών με φυσικές συνδέσεις (κανάλια επικοινωνίας), οι οποίες μπορούν να παράγουν, να στέλνουν, να προωθούν και να λαμβάνουν πληροφορίες (απλά δεδομένα, ήχο, εικόνα και βίντεο).

Οι υπολογιστές και οι συσκευές (εκτυπωτές, τερματικά, δρομολογητές και δορυφόροι) που συνδέονται σε ένα δίκτυο ονομάζονται **σταθμοί ή τερματικές συσκευές**. Χαρακτηριστικές εφαρμογές δικτύων υπολογιστών αποτελούν οι τηλεπικοινωνίες (σταθερή/κινητή τηλεφωνία), η καλωδιακή τηλεόραση, το ηλεκτρονικό ταχυδρομείο, η τηλεγραφία, η τηλεδιάσκεψη), οι οικονομικές υπηρεσίες (π.χ. ηλεκτρονική συναλλαγή με την εφορία), οι πωλήσεις, το μάρκετινγκ και η διαφήμιση.

3.3.2 Στοιχεία δικτύων

Οι διάφορες συσκευές ενός δικτύου συνδέονται μεταξύ τους με γραμμές/ζεύξεις μετάδοσης (links).

Σε ένα δίκτυο υπάρχουν υπολογιστές υποδοχής, γραμμές μετάδοσης και στοιχεία μεταγωγής.

Οι Υπολογιστές Υποδοχής (hosts) παίζουν το ρόλο του πομπού ή του δέκτη και μπορεί να είναι κινητές συσκευές, προσωπικοί υπολογιστές, ή κεντρικά ισχυρά υπολογιστικά συστήματα (Mainframes).

Οι Γραμμές Μετάδοσης (Transmission Lines, Links) αποτελούν τα φυσικά μονοπάτια επικοινωνίας (κανάλια ή δίαυλους) διαμέσου των οποίων μεταφέρονται τα δεδομένα από τη μία συσκευή στην άλλη. Η επικοινωνία γίνεται με ασύρματο και ενσύρματο τρόπο.

Τα Στοιχεία Μεταγωγής (Switching Elements) είναι οι ενδιάμεσες συσκευές που συνδέουν τις γραμμές μετάδοσης και επιφορτίζονται με το έργο μεταφοράς των δεδομένων από την μία πλευρά στην άλλη μέσω διαδικασιών γνωστών ως δρομολόγηση και/ή μεταγωγή (Εικόνα 3.14).



Εικόνα 3.14.

Στοιχεία Μεταγωγής. Σε αυτή την κατηγορία ανήκουν οι Γέφυρες (Bridges), οι Μεταγωγείς (Switches), οι Δρομολογητές (Routers) και οι Πύλες (Gateways).

3.3.3 Κατηγοριοποίηση δικτύων

Ένα δίκτυο μπορεί να ανήκει σε διάφορες κατηγορίες ανάλογα με τη γεωγραφική του έκταση, το μέγεθός του και την τεχνολογία μετάδοσης και προώθησης που χρησιμοποιεί.

3.3.3.1 Είδη δικτύων ανάλογα με την τεχνολογία μετάδοσης

Στα **Δίκτυα σημείου προς σημείο** (point to point) δημιουργούνται συνδέσεις μεταξύ δύο κόμβων, οπότε ένα μήνυμα διατρέχοντας πολλές τέτοιες συνδέσεις, φθάνει στον προορισμό του. Στα **δίκτυα εκπομπής**, όλα τα μέλη του δικτύου μοιράζονται έναν κοινό δίαυλο. Στην **πολυεκπομπή** (multicasting) το μήνυμα λαμβάνεται από συγκεκριμένους παραλήπτες ενώ στην **καθολική εκπομπή** (broadcasting) από όλους.

3.3.3.2 Είδη δικτύων ανάλογα με την τεχνολογία προώθησης της πληροφορίας

Στα **δίκτυα μεταγωγής κυκλώματος** (circuit switching networks) η μετάδοση επιτυγχάνεται μέσα από μια αποκλειστικά χρησιμοποιούμενη φυσική σύνδεση (κύκλωμα), στην οποία τα δεδομένα δεν υποβάλλονται σε καμία επεξεργασία κατά τη διέλευσή τους από το δίκτυο. Η τεχνολογία μεταγωγής κυκλώματος χρησιμοποιείται κυρίως στο τηλεφωνικό δίκτυο.

Στα δίκτυα μεταγωγής πακέτου (packet switching networks) τα δεδομένα χωρίζονται σε μικρότερα τμήματα, τα πακέτα. Κάθε πακέτο πέραν του τμήματος προς αποστολή, περιέχει και πληροφορίες ελέγχου οι οποίες διασφαλίζουν τη σωστή δρομολόγηση του πακέτου μέσα στο δίκτυο. Στη συνέχεια αυτά μέσω ενδιάμεσων κόμβων ή στοιχείων μεταγωγής, φθάνουν στον τελικό παραλήπτη που τα συναρμολογεί και δημιουργεί το αρχικό μήνυμα. Τα πακέτα μπορούν να ακολουθούν την ίδια διαδρομή ή κάθε πακέτο μπορεί να ακολουθεί τη δική του διαδρομή.

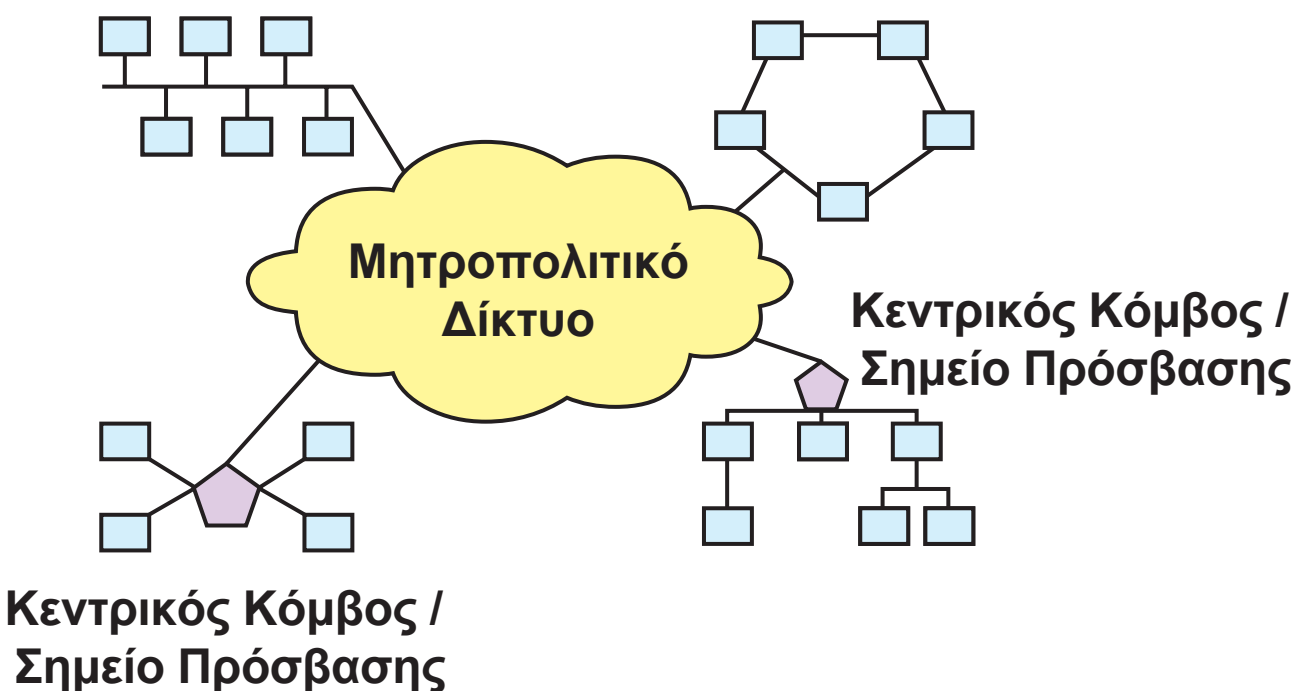


Το λογισμικό το οποίο πραγματοποιεί τις διάφορες λειτουργίες ενός δικτύου (όπως δρομολόγηση, έλεγχο σφαλμάτων, τμηματοποίηση πληροφορίας, κρυπτογράφηση πληροφορίας) είναι οργανωμένο με βάση κανόνες, οι οποίοι ονομάζονται πρωτόκολλα.

3.3.3.3 Είδη δικτύων βάσει περιοχής που καλύπτουν

Τα Τοπικά Δίκτυα (LAN – Local Area Networks) καλύπτουν μία μικρή έκταση, συνδέοντας συσκευές που βρίσκονται σε ένα δωμάτιο ή σε ένα κτίριο ή συγκρότημα κτιρίων.

Τα Μητροπολιτικά Δίκτυα (MAN – Metropolitan Area Networks) εκτείνονται στο περιβάλλον μιας ολόκληρης πόλης και χρησιμοποιούνται για την διασύνδεση δικτύων LAN ή σαν δίκτυα κορμού. Παραδείγματα τέτοιων δικτύων είναι ένα τηλεπικοινωνιακό δίκτυο υποδομής και ένα δίκτυο καλωδιακής τηλεόρασης (Εικόνα 3.15).



Εικόνα 3.15.

Παράδειγμα ενός Μητροπολιτικού δικτύου που αποτελείται από 4 δίκτυα.

Τα δίκτυα **Ευρείας Περιοχής (WAN – Wide Area Networks)** επεκτείνονται σε μεγάλες γεωγραφικές περιοχές που αποτελούνται από διάφορες χώρες ή ακόμα και ηπείρους. Χρησιμοποιούν πλήθος ενδιάμεσων συσκευών, όπως π.χ. δορυφόρους.



Διαδίκτυο (Internet ή internetwork) είναι ένα σύνολο από δύο ή περισσότερα δίκτυα (LAN, MAN, WAN κλπ.) που συνδέονται μεταξύ τους με κατάλληλες συσκευές. Σήμερα με τον όρο Διαδίκτυο αναφερόμαστε στην τεχνολογία που στηρίζεται στα πρωτόκολλα TCP και IP και αποτελεί τη βάση για τις εφαρμογές στον Παγκόσμιο Ιστό (World Wide Web, WWW).

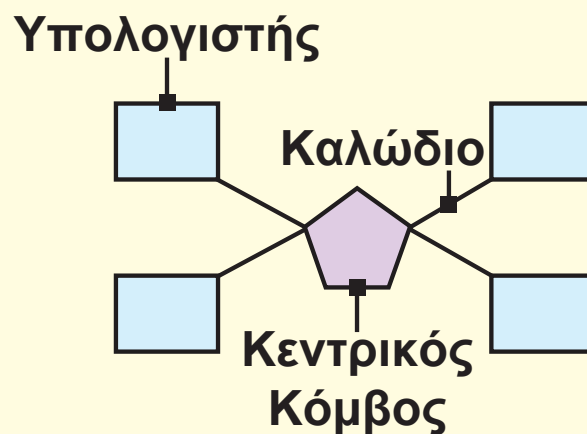
Το **TCP/IP** (Transmission Control Protocol/ Internet Protocol=Πρωτόκολλο Ελέγχου Μετάδοσης / Πρωτόκολλο Διαδικτύου) είναι μια συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το Διαδίκτυο.

3.3.4 Τοπολογίες δικτύων

Ο τρόπος με τον οποίο συνδέονται οι σταθμοί σε ένα δίκτυο ονομάζεται **τοπολογία δικτύου**.

Υπάρχουν τρεις βασικές τοπολογίες δικτύων, οι οποίες διαφέρουν ως προς την αξιοπιστία, το κόστος, την ανοχή τους σε σφάλματα και την ταχύτητα ανάνηψης μετά από κατάρρευση.

Τοπολογία Αστέρα: Υπάρχει ένας κεντρικός κόμβος για τον έλεγχο της κυκλοφορίας και όλες οι συσκευές συνδέονται με αυτόν με μία φυσική σύνδεση (Εικόνα 3.16).

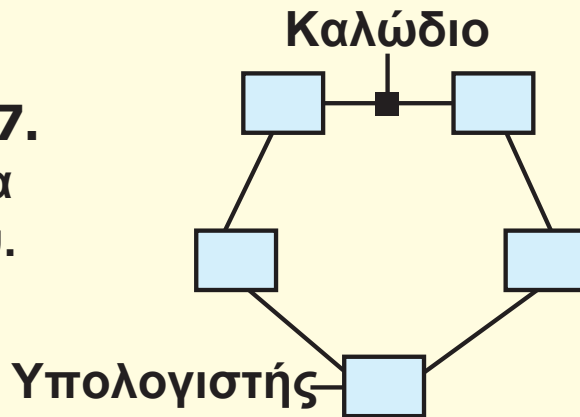


Εικόνα 3.16. Τοπολογία Αστέρα.

Τοπολογία Δακτυλίου: Σε αυτήν την τοπολογία, η κάθε συσκευή συνδέεται με μια γραμμή (point-to-point link) με τις δύο διπλανές συσκευές, δημιουργώντας ένα δακτύλιο. Ένα μήνυμα μεταφέρεται από τον κάθε κόμβο στον διπλανό του προς την ίδια κατεύθυνση μέχρι να

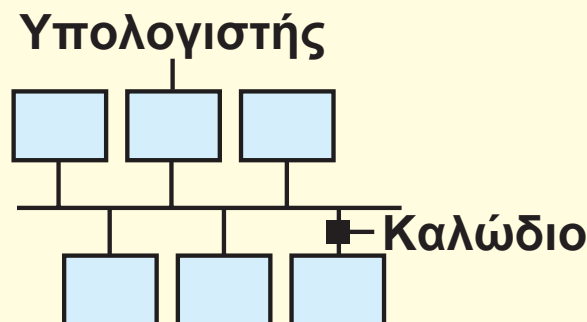
φθάσει στον προορισμό του (εικόνα 3.17). Η τοπολογία αυτή χρησιμοποιείται τόσο σε τοπικά όσο και ευρείας περιοχής δίκτυα.

Εικόνα 3.17.
Τοπολογία
Δακτυλίου.



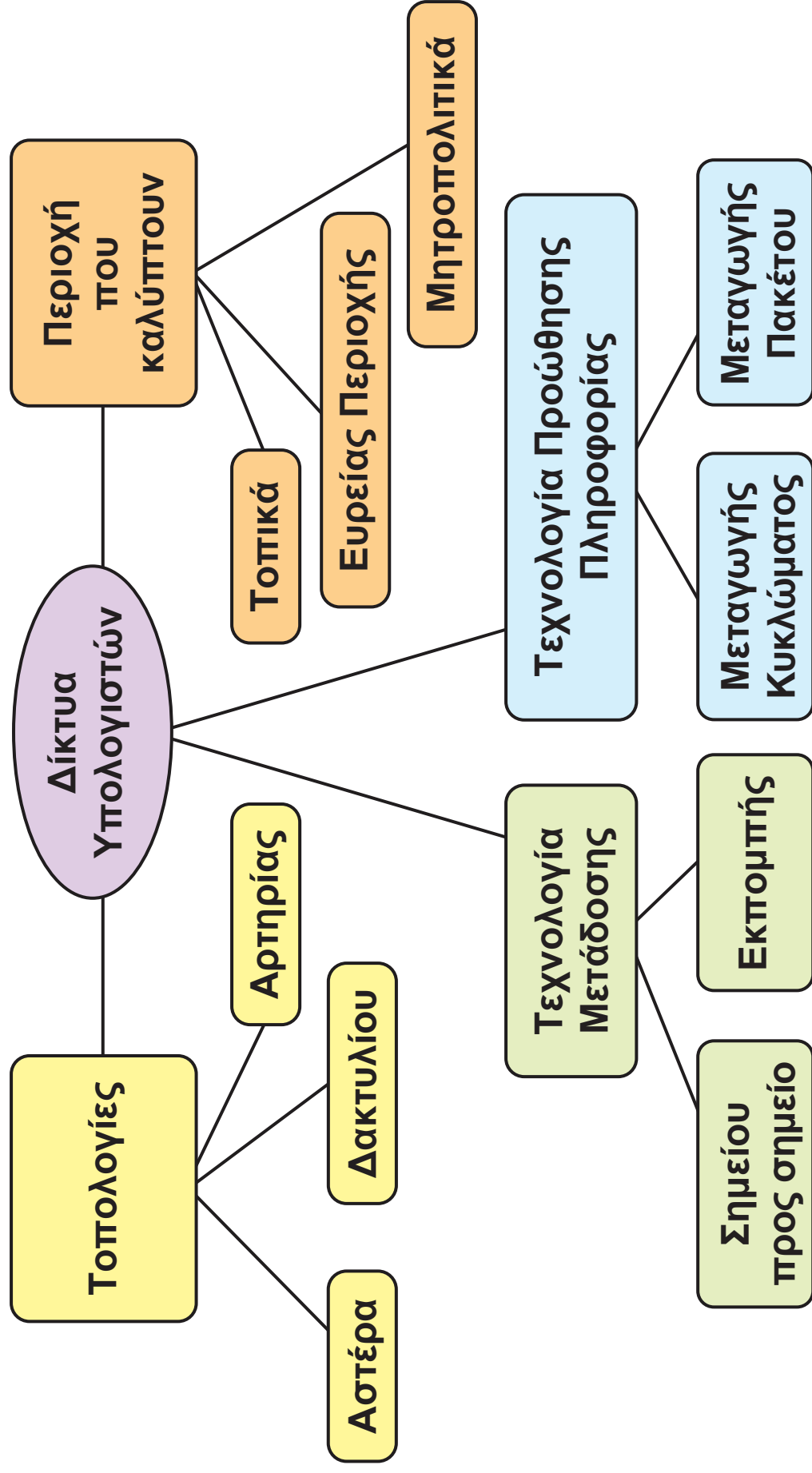
Τοπολογία Αρτηρίας: Στην τοπολογία αυτή, υπάρχει μια γραμμή (καλώδιο) που αποτελεί τη ραχοκοκαλιά του δικτύου και όλες οι συσκευές είναι συνδεδεμένες σε αυτήν (Εικόνα 3.18). Η τοπολογία αυτή συναντάται κυρίως σε τοπικά δίκτυα.

Εικόνα 3.18.
Τοπολογία
Αρτηρίας.



Εκτός από τις τρεις αυτές βασικές τοπολογίες υπάρχουν και άλλες οι οποίες αποτελούν παραλλαγές/συνδυασμούς των τριών βασικών, όπως οι τοπολογίες πλέγματος και τοπολογίες δένδρου.

Όλες οι κατηγοριοποιήσεις που έχουν αναφερθεί μέχρι τώρα συνοψίζονται στον παρακάτω νοητικό χάρτη (Εικόνα 3.19):



Εικόνα 3.19. Νοητικός χάρτης κατηγοριοποίησης των Δικτύων Υπολογιστών.

3.3.5 Σύγχρονες υπηρεσίες δικτύων

Οι σύγχρονες υπηρεσίες των δικτύων υπολογιστών στηρίζονται κυρίως στην αλματώδη ανάπτυξη του Διαδικτύου, το οποίο από ένα ρόλο παρουσίασης πληροφοριών μεταλλάχθηκε σε ένα εργαλείο κοινωνικής δικτύωσης, αναμορφώνοντας τον τρόπο επικοινωνίας και συνεργασίας μεταξύ ανθρώπων.

Ο Παγκόσμιος Ιστός (World Wide Web ή απλά Web), δίνει στο χρήστη τη δυνατότητα της πρόσβασης σε μεγάλη ποικιλία από πληροφορίες που είναι αποθηκευμένες στο διαδίκτυο υπό μορφή ιστοσελίδων.



Εικόνα 3.20.
Sir Tim Berners Lee. Ο δημιουργός του Παγκόσμιου Ιστού.

Η τηλεφωνία μέσω διαδικτύου (Voice over IP ή VoIP) προσφέρει φωνητική συνομιλία σε πραγματικό χρόνο με καλή ποιότητα και με μηδενικό κόστος.

Οι τεχνολογίες DSL (Digital Subscriber Line) παρέχουν πρόσβαση υψηλών ταχυτήτων στο διαδίκτυο και με χρήση των υπαρχουσών χάλκινων τηλεφωνικών γραμμών.

Το Υπολογιστικό Νέφος ή σύννεφο (Cloud Computing) παρέχει υπολογιστικούς πόρους (όπως διάφορες εφαρμογές, βάσεις δεδομένων, υπηρεσίες αρχείων, ηλεκτρονικό ταχυδρομείο κ.α.) μέσω ενός δικτύου υπολογιστών. Πρόκειται για μία παγκόσμια τεχνολογική υποδομή (global technological infrastructure), στην οποία ο χρήστης ενός υπολογιστή έχει πρόσβαση και χρησιμοποιεί λογισμικό και δεδομένα τα οποία είναι εγκατεστημένα ή βρίσκονται εκτός του προσωπικού του υπολογιστικού συστήματος.



Πολλά Πληροφοριακά Συστήματα χρησιμοποιούν Αρχιτεκτονικές αποθήκευσης βασισμένες στο δίκτυο και στο σύννεφο.



Ανακεφαλαίωση

Δίκτυα υπολογιστών δημιουργούνται με τη διασύνδεση υπολογιστών και περιφερειακών συσκευών με διάφορους τρόπους. Οι υπηρεσίες των δικτύων έχουν εφαρμογή στους περισσότερους τομείς της καθημερινότητάς μας. Αναδύονται καινούριες εφαρμογές σε πολύ συχνή βάση.

Χρήσιμοι Υπερσύνδεσμοι

<http://www.w3.org>

Ο οργανισμός World Wide Web Consortium.

<http://www.eett.gr/opencms/opencms/EETT/index.html>

Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων.

<http://www.yme.gr>

Υπουργείο Υποδομών Μεταφορών και Δικτύων



Λέξεις κλειδιά

Δίκτυο, μεταγωγή, μετάδοση, τοπολογίες, διαδίκτυο, LAN, WAN, υπολογιστικό νέφος.

Ερωτήσεις - Δραστηριότητες - Θέματα προς συζήτηση

1. Τι είναι ένα δίκτυο υπολογιστών;
2. Αναζητήστε στο διαδίκτυο πληροφορίες για τη δομή του δικτύου της κινητής τηλεφωνίας. Αν στην τάξη σας βρίσκονται μαθητές με διαφορετική καταγωγή, συγκρίνετε τα σχετικά δίκτυα μεταξύ τους από πλευράς δομής τους (ομαδική εργασία).
3. Αναζητήστε πληροφορίες στο διαδίκτυο για τα κύρια χαρακτηριστικά δύο ενσύρματων και 2 ασυρμάτων καναλιών επικοινωνίας.

- 4. Αναζητήστε στο διαδίκτυο πληροφορίες για υπηρεσίες υπολογιστικού νέφους.**
- 5. Αναζητήστε πληροφορίες για την συμβολή των δικτύων υπολογιστών στην υποστήριξη ΑμεΑ.**
- 6. Ποιές από τις παρακάτω προτάσεις είναι σωστές;**
 - A. Οι Υπολογιστές Υποδοχής μπορεί να είναι κινητές συσκευές, προσωπικοί υπολογιστές.**
 - B. Στα Δίκτυα Μεταγωγής Κυκλώματος τα δεδομένα υποβάλλονται σε επεξεργασία κατά την διέλευσή τους απο το δίκτυο.**
 - Γ. Τα Μητροπολιτικά Δίκτυα είναι μεγαλύτερα από τα Δίκτυα Ευρείας Περιοχής.**
 - Δ. Η Τοπολογία Δακτυλίου χρησιμοποιείται τόσο σε τοπικά όσο και ευρείας περιοχής δίκτυα.**

Τεχνητή Νοημοσύνη

Στόχοι του κεφαλαίου είναι οι μαθητές:

- ✓ Να εντάξουν την Τεχνητή Νοημοσύνη στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών
- ✓ Να γνωρίσουν τις επιστημονικές περιοχές εφαρμογής της Τεχνητής Νοημοσύνης
- ✓ Να μπορούν να αναφέρουν τομείς στους οποίους έχει εφαρμογή η Τεχνητή Νοημοσύνη.



Προερωτήσεις

- Σε έναν αγώνα σκάκι μεταξύ ανθρώπου και υπολογιστή πώς σχεδιάζονται οι πιθανές κινήσεις της μηχανής;
- Πώς λειτουργεί ένα σύστημα διόρθωσης κειμενογράφου;
- Πώς γίνεται η αναζήτηση πληροφοριών στο διαδίκτυο;
- Με ποιον τρόπο θα δίνετε οδηγίες σε ένα αυτοκινούμενο όχημα ώστε να διανύσει μια διαδρομή με εμπόδια;

3.4.1 Τι είναι η Τεχνητή Νοημοσύνη

Τα τελευταία χρόνια η Τεχνητή Νοημοσύνη έχει ενταχθεί στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών λόγω των θεμάτων που πραγματεύεται και των δεξιοτήτων που αναπτύσσονται στους χώρους εφαρμογής της.

Τεχνητή Νοημοσύνη (Τ.Ν.) είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών υπολογιστικών συστημάτων, δηλαδή συστημάτων ικανών για λειτουργίες που αποδίδονται σε ανθρώπινη νοημοσύνη.

Η Τ.Ν. εμφανίζεται σε διάφορα πεδία, όπως

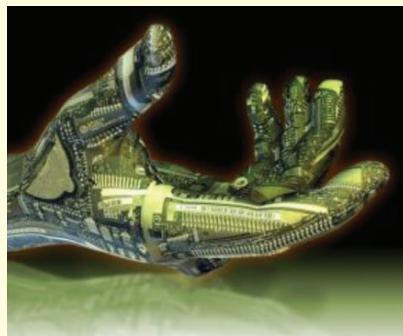
- στον προγραμματισμό ηλεκτρονικών παιχνιδιών
- στα έμπειρα συστήματα τα οποία χρησιμοποιούνται για την λήψη αποφάσεων γύρω από ζητήματα πραγματικής ζωής, διάγνωση και θεραπεία ασθενειών
- στην επεξεργασία φυσικών γλωσσών που βοηθά στην κατανόηση των νοημάτων μίας γλώσσας με τη χρήση μηχανών
- στον προγραμματισμό νοήμονος συμπεριφοράς σε μηχανές με τη χρήση νευρωνικών δικτύων τα οποία προσομοιώνουν τις φυσικές συνδέσεις ενός ανθρώπινου εγκεφάλου

- στη ρομποτική όπου προγραμματίζονται υπολογιστικά συστήματα ώστε να «βλέπουν», να «ακούν» και να «αντιδρούν» με βάση τα ανθρώπινα πρότυπα συμπεριφοράς (Εικόνα 3.21).



- 1.** Το ρομπότ δεν θα κάνει κακό σε άνθρωπο, ούτε με την αδράνειά του θα επιτρέψει να βλαφτεί ανθρώπινο όν.
- 2.** Το ρομπότ πρέπει να υπακούει τις διαταγές που του δίνουν οι άνθρωποι, εκτός αν αυτές οι διαταγές έρχονται σε αντίθεση με τον πρώτο νόμο.
- 3.** Το ρομπότ οφείλει να προστατεύει την ύπαρξή του, εφόσον αυτό δεν συγκρούεται με τον πρώτο και τον δεύτερο νόμο.

(Ισαάκ Ασίμωφ,
Isaac Asimov, 1942)



Εικόνα 3.21. Χέρι ρομπότ.

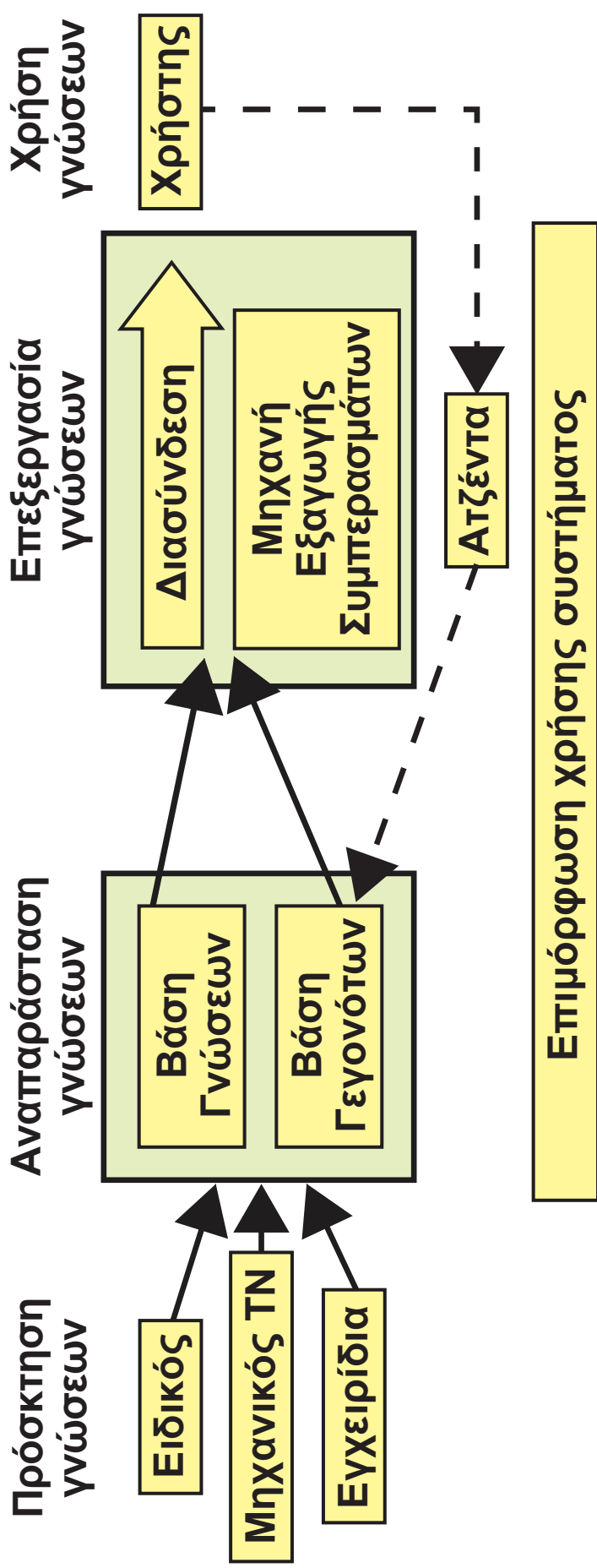
Κάθε πρόβλημα στην T.N. διακρίνεται σε δύο τμήματα. Το πρώτο τμήμα αφορά στην αναπαράσταση των υποθέσεων και των συμπερασμάτων, ενώ το δεύτερο αφορά στον ορισμό και την παραγωγή των λογικών σχέσεων μεταξύ των υποθέσεων και των συμπερασμάτων. Η T.N. χρησιμοποιεί λογική για να αναπαραστήσει τη γνώση και το μηχανισμό των αποδείξεων και των συμπερασμάτων για να την επεξεργαστεί. Ένα τυπικό σύστημα T.N. αποτελείται από τρία μέρη: το στάδιο εισερχομένων, το στάδιο επεξεργασίας και το στάδιο εξερχομένων. Με παρόμοιο τρόπο στην T.N. κάθε μορφή γνώσης μπορεί να αναπαρασταθεί σε τρία αντίστοιχα στάδια (Εικόνα 3.22). Το πρώτο αφορά στον τρόπο με τον οποίο η γνώση γίνεται μεταδότημη (Πρόσκτηση γνώσεων) μέσω ειδικών, εγχειριδίων ή μηχανικών της T.N. οι οποίοι τροφοδοτούν μια βάση γνώσης.

Το δεύτερο στάδιο (Επεξεργασία γνώσεων) αφορά στην επεξεργασία αυτών που έχουν εισαχθεί και στο τρίτο στάδιο (Χρήση γνώσεων), καταγράφεται το πώς αξιοποιούνται τα τελικά συμπεράσματα. Τα τρία αυτά στάδια μπορεί να βρίσκονται σε κατάσταση αλληλεπίδρασης.



Κάθε μορφή γνώσης μπορεί να διακριθεί σε τρία στάδια:

- Πρόσκτηση γνώσεων
- Επεξεργασία γνώσεων
- Χρήση γνώσεων

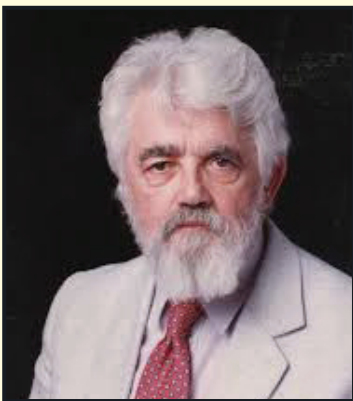


Εικόνα 3.22.

Αναπαράσταση Γνώσεων από την μεριά της Τεχνητής Νοημοσύνης.

Τεστ Turing

Η ιδέα της Τ.Ν. προήλθε από ένα «παιχνίδι μίμησης» που σήμερα είναι γνωστό ως Τεστ Turing. Ήταν ένα παιχνίδι ανάμεσα σε έναν ερωτώντα, μία γυναίκα, και έναν ψηφιακό υπολογιστή. Μεταξύ τους επικοινωνούσαν μέσω ηλεκτρονικού ταχυδρομείου (τηλε-εκτυπωτής Turing). Ο ερωτώντας μπορούσε να θέσει οποιαδήποτε ερώτηση είτε στην γυναίκα είτε στον υπολογιστή με τελικό στόχο να ανακαλύψει τη σωστή ταυτότητα του συνομιλητή του. Οι ερωτήσεις αποτελούσαν τα εισερχόμενα, οι απαντήσεις της γυναίκας και του υπολογιστή τα εξερχόμενα του συστήματος και το τεστ στόχευε να δείξει τη δυνατότητα ύπαρξης μηχανών με νοημοσύνη.



Εικόνα 3.23. Ο Τζων ΜακΚάρθου, (John McCarthy) έλαβε το 1971 το Βραβείο Turing για τη συνεισφορά του στον τομέα της Τ.Ν.

3.4.2 Εξέλιξη της Τεχνητής Νοημοσύνης

Το 1956 σε ένα ερευνητικό πρόγραμμα Τ.Ν. ο Τζων ΜακΚάρθου (Εικόνα 3.23) ανέπτυξε μια αλγεβρική γλώσσα επεξεργασίας λιστών που έδωσαν ώθηση στην έρευνα των **Γενετικών Αλγορίθμων (Genetic Algorithms)**. Οι γενετικοί αλγόριθμοι είναι αλγόριθμοι οι οποίοι αναζητούν μέσα από ένα χώρο υποψηφίων λύσεων, την

πιο κατάλληλη με βάση κάποιο συγκεκριμένο κριτήριο. Σημαντικά προβλήματα που αντιμετωπίζονται με αυτόν τον τρόπο είναι η εύρεση μεγίστου συνάρτησης μιας μεταβλητής, το πρόβλημα εύρεσης συντομότερης διαδρομής, η σχεδίαση κυκλωμάτων VLSI και ο σχεδιασμός ενός ωρολογίου προγράμματος. Την δεκαετία του '70 εμφανίζονται τα πρώτα έμπειρα συστήματα (Expert Systems) ως εμπορικές εφαρμογές και την δεκαετία του '80 γίνεται προσπάθεια για ενοποίηση των διαλέκτων της γλώσσας συναρτησιακού προγραμματισμού (Functional Programming) LISP κάτω από το πρότυπο της Common LISP. Η LISP βασίζεται στην αποτίμηση συναρτήσεων αντί για την εκτέλεση εντολών. Προγράμματα βασισμένα στο συναρτησιακό προγραμματισμό είναι γραμμένα με βάση συναρτήσεις που αποτιμούν βασικές τιμές.

Παράλληλα υλοποιήθηκαν τα πρώτα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Nets). Το αντικείμενο των τεχνητών νευρωνικών δικτύων αποτελεί ένα ευρύ και αυτόνομο επιστημονικό πεδίο, που σχετίζεται με το γενικότερο πλαίσιο των ευφυών συστημάτων και δικτύων. Μπορεί να σκεφτεί κανείς μια σειρά επιτυχημένων εφαρμογών όπως τα έξυπνα φίλτρα ηλεκτρονικού ταχυδρομείου, οι υψηλών ταχυτήτων και αποτελεσματικοί δρομολογητές, και η αυτόματη αναγνώριση πληροφοριών που έχουν σημασία και συνάφεια.

Ένα ακόμα βήμα στην εξέλιξη της Τ.Ν. ήταν η σύνδεσή της με την θεωρία της **Ασαφούς Λογικής (Fuzzy Logic)**. Η θεωρία της Ασαφούς Λογικής μέσω κατάλληλων αλγορίθμων έχει προσαρμοσθεί σε συστήματα ώστε, μέσω της υπάρχουσας γνώσης μιας ασαφούς κατάστασης να προκύπτουν διαχειρίσιμα και συγκεκριμένα συμπεράσματα. Βασικός εισηγητής της είναι ο καθηγητής **Λότφι Ζάντεχ (Εικόνα 3.24)**. Σχετικές εφαρμογές υπάρχουν σε συστήματα ταξινόμησης και σε αυτόματους διορθωτές κειμένων.



Εικόνα 3.24.
Λότφι Ζάντεχ (Lotfi Zadeh).
Βασικός εισηγητής της Ασαφούς
Λογικής το 1965.

Τη δεκαετία του '90 με την ανάπτυξη και τη διάδοση του διαδικτύου εμφανίζονται οι πρώτοι **ευφυείς πράκτορες** (intelligent agents), ανεξάρτητα προγράμματα τα οποία λαμβάνουν αποφάσεις και αλληλεπιδρούν με τα διάφορα συστήματα, και τα διαδικτυακά ρομπότ. Ακολούθησαν ο **Εξελικτικός Υπολογισμός** (Evolutionary Computation), η **Νοημοσύνη Σμηνών** (Swarm Intelligence), η αξιοποίηση της **Μηχανικής Μάθησης** (Machine Learning) και η **Ανακάλυψη Γνώσης σε Βάσεις Δεδομένων** (Knowledge Discovery in Databases) ως κατηγορίες της **Υπολογιστικής Νοημοσύνης**.



Η **Μηχανική Μάθηση** πραγματοποιείται όταν ένα υπολογιστικό σύστημα δημιουργεί νέα μοντέλα ή πρότυπα μέσα από ένα σύνολο δεδομένων.

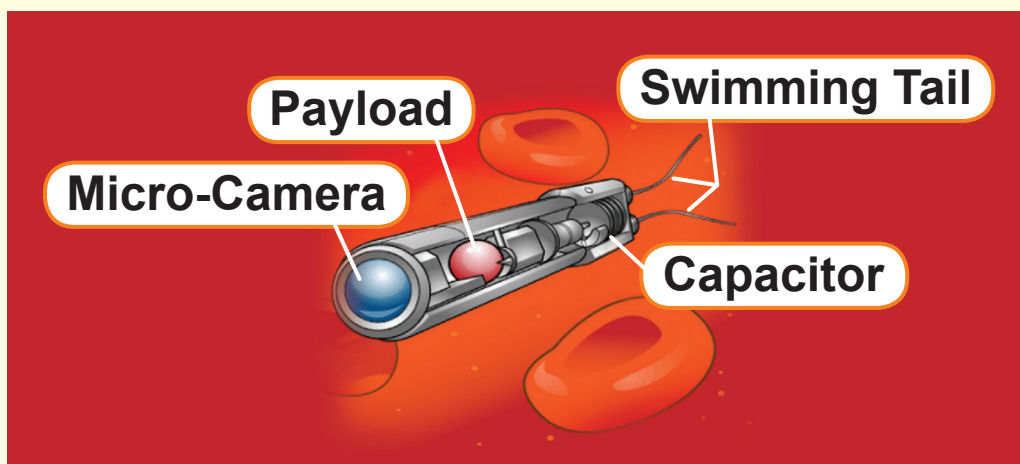


Ο **Εξελικτικός Υπολογισμός** και η **Νοημοσύνη Σμηνών** είναι μέθοδοι διαδικασιών βελτιστοποίησης υπολογισμών που προέκυψαν παρατηρώντας πρότυπα που υπάρχουν στη φύση. Σε αυτά γίνεται χρήση μαθηματικών εργαλείων από την **Θεωρία Πιθανοτήτων** και τα **Δυναμικά Συστήματα**.

Η Ανακάλυψη Γνώσης σε Βάσεις Δεδομένων είναι μία σύνθετη διαδικασία η οποία προσδιορίζει έγκυρες, νέες, χρήσιμες και κατανοητές σχέσεις - πρότυπα που υπάρχουν σε μία βάση δεδομένων.

3.4.3 Τομείς εφαρμογών της Τεχνητής Νοημοσύνης

Η Τ.Ν. επωφελήθηκε από την πρόοδο της Ηλεκτρονικής, της μικροηλεκτρονικής και της νανοτεχνολογίας (Εικόνα 3.25). Η ανάπτυξη και η εξέλιξη της Τ.Ν. στηρίζεται στην αξιοποίηση γνώσεων από περιοχές που εκτείνονται από τη μηχανική έως τη θεωρητική πληροφορική.



Εικόνα 3.25. Ιατρικά μικρορομπότ

Η Τ.Ν. παρέχει την δυνατότητα επεξεργασίας σε χώρους που δεν μπορεί να έχει πρόσβαση ο άνθρωπος, όπως το διάστημα (Εικόνα 3.26), τα βάθη των ωκεανών, οι μολυσμένες περιοχές.



Εικόνα 3.26. Ο «ρομποναύτης» της NASA κατασκευάστηκε για την συντήρηση του διαστημικού τηλεσκοπίου HUBBLE.

Η T.N. εφαρμόζεται σε ένα πλήθος πεδίων, όπως η συμπερασματολογία, ο αυτοματισμός, η ρομποτική, η αναγνώριση φωνής, η ευφυής αναζήτηση στο διαδίκτυο και τα συστήματα ελέγχου. Ο ρόλος της T.N. είναι σημαντικός στην επίλυση σύνθετων προβλημάτων, στο σχεδιασμό της συμπεριφοράς μηχανών, τη χρήση βιομηχανικών ρομπότ και ειδικότερα όπου χρειάζεται να γίνει ανάλυση μέσων και σκοπών. Σημαντικά πεδία εφαρμογής επίσης είναι η οπτική - ακουστική αντίληψη και η αναγνώριση προτύπων (pattern recognition).

Για παράδειγμα, στην περίπτωση ενός πυραύλου «αναγνώριση προτύπων» σημαίνει «εξατομίκευση στόχου» και «όραση υπολογιστή», σημαίνει δηλαδή την κατασκευή ενός συστήματος που «βλέπει» και «αντιδρά» εξίσου καλά με τον άνθρωπο. Άλλο πεδίο επιστημονικής εφαρμογής αποτελεί η έμπειρη ανάλυση και η παροχή συμβουλών στην ιατρική διάγνωση. Στην ιατρική έχουν αναπτυχθεί συστήματα που αναλύουν τα συμπτώματα μιας ασθένειας, το ιατρικό ιστορικό, τα αποτελέσματα των εργαστηριακών εξετάσεων ενός ασθενή και στη συνέχεια προτείνουν στο θεράποντα ιατρό μία ποικιλία δυνατών διαγνώσεων.

3.4.4 Γλώσσες προγραμματισμού που χρησιμοποιούνται στην T.N.

Η **Prolog** είναι μια γλώσσα λογικού προγραμματισμού, η οποία κυριαρχεί στην περιοχή των εφαρμογών της T.N.. Ένα πρόγραμμα της **Prolog** είναι μία συλλογή από γεγονότα και κανόνες που χρησιμοποιούνται για να αποδειχθούν κάποιες προτάσεις. Αρχικά ο χρήστης εισάγει τα δεδομένα ενός προβλήματος στο περιβάλλον της **Prolog**. Κατόπιν θέτει ερωτήματα γύρω από αυτά ή τις σχέσεις τους και λαμβάνει απαντήσεις. Οι απαντήσεις που δίνονται σε κάθε ερώτημα αξιοποιούν αποκλειστικά τα γεγονότα και τους κανόνες που έχουν εισαχθεί στο εκάστοτε πρόγραμμα (εικόνα 3.27). Μέσα από συσχετισμούς και συλλογισμούς, το περιβάλλον της **Prolog** καταλήγει να αποδείξει το αληθές μιας πρότασης ή την προβολή του αποτελέσματος που θα προκύψει.

man(giannis).
man(giorgos).
woman(aggeliki).

Εικόνα 3.27. Πρόγραμμα σε Prolog.

Αφού γραφεί το πρόγραμμα και εκτελεστεί, θα εμφανιστεί:
?-

όπου περιμένει ερωτήσεις. Π.χ.

?- man(giorgos).

και απαντά

Yes

?- woman(maria).

και απαντά

No



Ανακεφαλαίωση

Η Τ.Ν. είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται, εκτός των άλλων, με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά. Τα προβλήματα που επιλύονται αφορούν την αντίληψη μέσω της όρασης, την μηχανική μάθηση, την εξαγωγή συμπερασμάτων από βάσεις δεδομένων κλπ. Η βασική γλώσσα προγραμματισμού της είναι η Prolog.

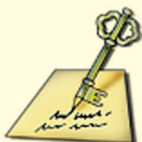
Χρήσιμοι Υπερσύνδεσμοι

<http://www.eccai.org>

Ευρωπαϊκή Συντονιστική Επιτροπή για Τ.Ν.

<http://www.eetn.gr/>

Ελληνική Εταιρεία Τεχνητής Νοημοσύνης



Λέξεις κλειδιά

Υπολογιστικά μοντέλα ανθρώπινης γνώσης, Νευρωνικά Δίκτυα, Έμπειρα Συστήματα, Ασαφής Λογική, Γενετικοί Αλγόριθμοι.

Ερωτήσεις - Θέματα προς συζήτηση - Δραστηριότητες

- 1.** Σχολιάστε και προτείνετε ταινίες επιστημονικής φαντασίας με κύρια αναφορά στην Τ.Ν. Εξηγήστε γιατί σας έκαναν εντύπωση.
- 2.** Να διερευνήσετε τι σημαίνει για τον άνθρωπο, τους ηθικούς του κώδικες και την εξέλιξή του η αποδοχή βοήθειας ή συμβουλών από μία μηχανή ή ένα υπολογιστικό σύστημα;
- 3.** Αναζητήστε αναφορές στην Τ.Ν. από την αρχαία ελληνική σκέψη, τη μυθολογία, την ιστορία ή τη λογοτεχνία των νεότερων χρόνων.

Βιβλιογραφία

- Comer, D. E. (2014). Δίκτυα και Διαδίκτυα Υπολογιστών. Αθήνα: Κλειδάριθμος.
- Floridi, L. (2008). Εισαγωγή στη φιλοσοφία της Πληροφορικής. Αθήνα: Νήσος.
- Forouzan, B. A. (2006). Πρωτόκολλο TCP/IP. Αθήνα: Γκιούρδας.
- Laudon, K. C., & Laudon, J. P. (2009). Πληροφοριακά Συστήματα Διοίκησης. Αθήνα: Κλειδάριθμος.
- Russell, S., & Norvig, P. (2004). Τεχνητή νοημοσύνη: Μια σύγχρονη προσέγγιση. Αθήνα: Κλειδάριθμος.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). Λειτουργικά Συστήματα. Αθήνα: Γκιούρδας.
- Tanenbaum, A. S. (2009). Σύγχρονα Λειτουργικά Συστήματα. Αθήνα: Κλειδάριθμος.
- Tanenbaum, A. S. (2012). Δίκτυα Υπολογιστών. Αθήνα: Κλειδάριθμος.
- Webber, A. B. (2009). Σύγχρονες γλώσσες προγραμματισμού. Πανεπιστημιακές Εκδόσεις Κρήτης.
- Αλεξανδρής, Ν., Μπελεσιώτης, Β. Σ., & Παναγιωτόπουλος, Θ. (2002). Εισαγωγή στην Επιστήμη των Υπολογιστών. Αθήνα: Εκδόσεις Βαρβαρήγου.
- Βακάλη, Α., Γιαννόπουλος, Η., Ιωαννίδης, Ν., Κοίλιας, Χ., Μάλαμας, Κ. Μανωλόπουλος, Ι., & Πολίτης, Π.

(1999). Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον. Αθήνα: ΥΠΕΠΘ.

Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Φ., & Σακελλαρίου, Η. (2011). Τεχνητή Νοημοσύνη. Εκδόσεις Πανεπιστημίου Μακεδονίας.

Βογιατζής, Ι., Ιωαννίδης, Ν., Κοίλιας, Χ., Μελετίου, Γ., & Μόρμορης, Μ. (2010). Εισαγωγή στην Αλγοριθμική. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Γεωργόπουλος, Ν., Κοπανάκη, Ε., Πανταζή, Μ., Νικολαράκος, Χ., & Βαγγελάτος, Ι. (2013). Ηλεκτρονικό Επιχειρείν. Αθήνα: Εκδόσεις Σταμούλης.

Δημητριάδης, Α. (2007). Διοίκηση-Διαχείριση Πληροφοριακών Συστημάτων. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Δουληγέρης, Χ. (2014). Σύγχρονα Τηλεπικοινωνιακά και Διαδικτυακά Πρωτόκολλα. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Καλαφατούδης, Σ., Δροσίτης, Ι., & Κοίλιας, Χ. (2012). Εισαγωγή στις Τεχνολογίες Πληροφορίας και Επικοινωνίας. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Κοίλιας, Χ. (2004). Δομές Δεδομένων και Οργανώσεις Αρχείων. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Κοίλιας, Χ., & Παναγιωτάκος, Δ. (1994). Ερμηνευτικό Λεξικό Όρων Πληροφορικής. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Μανωλόπουλος, Ι., & Παπαδόπουλος, Α. (2006). Συστήματα Βάσεων Δεδομένων. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Γλωσσάρι

AND operation = πράξη ΚΑΙ, τομή, σύζευξη, λογικός πολλαπλασιασμός. Η πράξη άλγεβρας Boole της οποίας το αποτέλεσμα έχει την τιμή 1, αν και μόνο εάν κάθε τελεστέος έχει την τιμή 1 (ΕΛΟΤ).

Ada. Γενικού σκοπού υψηλού επιπέδου διαδικασιακή γλώσσα προγραμματισμού, που αναπτύχθηκε αρχικά υπό την αιγίδα του αμερικανικού Υπουργείου Άμυνας για την υλοποίηση πολύ μεγάλων έργων λογισμικού.

access = προσπέλαση. Πρόσβαση σε δεδομένα με σκοπό την ανάγνωση, εγγραφή ή μετακίνηση δεδομένων ή εντολών.

access method = μέθοδος προσπέλασης. 1) Στα τοπικά δίκτυα πρωτόκολλο που καθορίζει ποια συσκευή θα έχει προσπέλαση στο μέσο μεταφοράς κάθε στιγμή. 2) Η τεχνική ή και τα προγράμματα για τη μετακίνηση δεδομένων μεταξύ συσκευών εισόδου/εξόδου και κύριας μνήμης.

access right = δικαίωμα προσπέλασης. Δικαίωμα που παραχωρείται σε ένα χρήστη να έχει πρόσβαση σε ορισμένα δεδομένα ή προγράμματα και να τα χρησιμοποιεί κατά ένα συγκεκριμένο τρόπο (ISO).

access time = χρόνος προσπέλασης. Ο χρόνος που χρειάζεται για την ανεύρεση και ανάκτηση καταχωρημένης πληροφορίας σε κάποιο είδος μνήμης. Συνήθως αναφέρεται στο μέγιστο απαιτούμενο χρόνο.

algorithm = αλγόριθμος. Πεπερασμένο σύνολο σαφώς καθορισμένων κανόνων που βοηθούν στην επίλυση ενός προβλήματος μέσω ενός πεπερασμένου αριθμού βημάτων (ΕΛΟΤ).

algorithmic languages = αλγοριθμικές γλώσσες. Κατηγορία γλωσσών προγραμματισμού κατάλληλων για την περιγραφή προβλημάτων που μπορούν να λυθούν αλγοριθμικά.

alphabetic data = αλφαβητικά δεδομένα. Δεδομένα που αποτελούνται από γράμματα μόνο ή από γράμματα και διάκενα.

alphanumeric = αλφαριθμητικό. Σύνολο χαρακτήρων που μπορεί να εμπεριέχει γράμματα, ψηφία και ειδικά σύμβολα όπως π.χ. σημεία στίξης.

analysis = ανάλυση. 1) Η φάση της ανάπτυξης ενός συστήματος πληροφορικής κατά την οποία αναλύεται η λειτουργία ενός οργανισμού και γίνεται ο καθορισμός των απαιτήσεων και προδιαγραφών των προγραμμάτων. 2) Η μεθοδική μελέτη ενός προβλήματος και η διαδικασία της διάσπασής του σε μικρότερες μονάδες για περαιτέρω έρευνα σε λεπτομέρεια.

analyst = αναλυτής. Πρόσωπο με ειδικότητα τον καθορισμό των απαιτήσεων και τη σύνταξη των προδιαγραφών των προγραμμάτων για μια συγκεκριμένη εφαρμογή.

append = προσάρτηση. Προσθήκη στοιχείων στο τέλος αρχείου.

application = εφαρμογή. Προγράμματα που γράφονται για την κάλυψη συγκεκριμένης ανάγκης π.χ. μιας επιχείρησης σε επεξεργασία δεδομένων.

applications software = λογισμικό εφαρμογών. Λογισμικό που γράφεται για να καλύψει ανάγκες εφαρμογών.

arithmetic = αριθμητικός. Τύπος δεδομένων που υποστηρίζουν όλες οι σύγχρονες γλώσσες προγραμματισμού. Με στοιχεία του αριθμητικού τύπου δεδομένων γίνονται αριθμητικές πράξεις.

arithmetic expression = αριθμητική έκφραση. Μία έκφραση που περιλαμβάνει αριθμητικές πράξεις και τελεστές και η οποία μπορεί να μετατραπεί σε μια απλή αριθμητική τιμή.

arithmetic operator = αριθμητικός τελεστής. Σύμβολο μιας γλώσσας προγραμματισμού, που καθορίζει την αριθμητική πράξη που πρέπει να γίνει μεταξύ δύο τελεστών, π.χ. +, -, *, /, ^.

array = πίνακας. Διάταξη δεδομένων μιας ή περισσότερων διαστάσεων.

array element = στοιχείο πίνακα. Ένα απλό προσπελάσιμο στοιχείο δεδομένου σε έναν πίνακα.

array index = δείκτης πίνακα. Σύνολο μιας ή περισσότερων τιμών που χρησιμοποιούνται για να προσπελαστεί ένα στοιχείο πίνακα.

artificial intelligence (AI) = τεχνητή νοημοσύνη. Ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, ώστε να καταστεί ο υπολογιστής ικανός για λειτουργίες που αποδίδονται σε ανθρώπινη νοημοσύνη.

artificial neural nets = τεχνητά νευρωνικά δίκτυα που έχουν δυνατότητα μάθησης μετασχηματίζοντας την εσωτερική τους δομή.

artificial intelligence languages = γλώσσες τεχνητής νοημοσύνης. Γλώσσες με δομή κατάλληλη για ανάπτυξη προγραμμάτων τεχνητής νοημοσύνης.

assembler = συμβολομεταφραστής. Πρόγραμμα που μεταφράζει συμβολική γλώσσα σε γλώσσα μηχανής του δεδομένου υπολογιστή.

assembly language = συμβολική γλώσσα. Γλώσσα χαμηλού επιπέδου εξαρτώμενη από το υλικό και η οποία έχει άμεση αντιστοιχία με τη γλώσσα μηχανής. Αποτελεί συμβολική αναπαράσταση του δυαδικού κώδικα της γλώσσας μηχανής και χρειάζεται συμβολομετάφραση.

assignment = εκχώρηση. Μηχανισμός τιμοδότησης μιας μεταβλητής (ISO).

BASIC (Beginners All-purpose Symbolic Instruction Code). Δημοφιλής γλώσσα προγραμματισμού. Τυπικά ασχολείται με αυτήν όποιος βρίσκεται -τουλάχιστον- στα πρώτα του βήματα στον προγραμματισμό.

backup = εφεδρεία. Δημιουργία αντιγράφων ασφαλείας που μπορούν να χρησιμοποιηθούν σε περίπτωση απώλειας των αρχικών.

backup copy = αντίγραφο εφεδρείας. Ο όρος χρησιμοποιείται για το αντίγραφο ενός αρχείου ή προγράμματος που τηρείται στην περίπτωση καταστροφής του αρχικού.

binary search = δυαδική αναζήτηση. Μέθοδος αναζήτησης κατά την οποία τα δεδομένα διαιρούνται διαδοχικά σε δύο ίσα τμήματα. Ένα από τα δύο τμήματα απαλλάσσεται από περαιτέρω έρευνα, γιατί είναι γνωστό ότι δεν περιέχει το προς αναζήτηση δεδομένο.

boolean (data type) = λογικός τύπος δεδομένου. Τύπος δεδομένου με δύο τιμές: αληθής (true) - ψευδής (false).

bridge = γέφυρα. Μια ηλεκτρονική συσκευή ή λογισμικό που χρησιμοποιείται για τη σύνδεση ενός τύπου ή πρωτοκόλλου δικτύου με ένα άλλο.

bubble sort = ταξινόμηση φουσαλίδας. Απλός αλγόριθμος ταξινόμησης κατά τον οποίο διαδοχικά στοιχεία συγκρίνονται μεταξύ τους και εάν δεν είναι τοποθετημένα σωστά, αντιμετατίθενται. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου καμία άλλη αντιμετάθεση δεν μπορεί να γίνει.

business applications = επιχειρησιακές εφαρμογές. Κατηγορία εφαρμογών προσανατολισμένων στις ανάγκες επιχειρήσεων. Συχνά αποκαλούνται και εμπορικές εφαρμογές.

C Language = γλώσσα C. Γλώσσα υψηλού επιπέδου ιδιαίτερα κατάλληλη για τη δημιουργία λογισμικού συστήματος.

C++ Language = γλώσσα C++. Αντικειμενοστρεφής έκδοση της C.

COBOL (COmmon Business Oriented Language). Γλώσσα προγραμματισμού υψηλού επιπέδου, βασισμένη στην Αγγλική, που χρησιμοποιείται κυρίως για διαχειριστικές εφαρμογές.

call = κλήση. Η ενεργοποίηση ενός προγράμματος υπολογιστή, μιας ρουτίνας ή μιας υπορουτίνας, συνήθως με καθορισμό των συνθηκών εισόδου και εκτέλεση άλματος προς ένα σημείο εισόδου (ISO).

cloud computing = Υπολογιστικό Νέφος ή σύννεφο. Παρέχει υπολογιστικούς πόρους (όπως διάφορες εφαρμογές, βάσεις δεδομένων, υπηρεσίες αρχείων, ηλεκτρονικό ταχυδρομείο κ.α.) μέσω ενός δικτύου υπολογιστών. Πρόκειται για μία παγκόσμια τεχνολογική υποδομή στην οποία ο χρήστης ενός υπολογιστή έχει πρόσβαση και χρησιμοποιεί λογισμικό και δεδομένα, τα οποία είναι εγκατεστημένα ή βρίσκονται εκτός του προσωπικού του υπολογιστικού συστήματος.

Cloud Service Provider (CSP) = Πάροχος υπηρεσιών σύννεφου.

code = κώδικας. 1) Σύνολο μη διαφορούμενων κανόνων που καθορίζουν τον τρόπο με τον οποίο δεδομένα μπορούν να παρασταθούν με διάκριτη μορφή (ISO). 2) Ένα ή περισσότερα προγράμματα ή τμήμα προγράμματος.

combination = συνδυασμός. Δεδομένο πλήθος διαφορετικών στοιχείων επιλεγμένων από ένα σύνολο χωρίς να λαμβάνεται υπόψη η διάταξη στην οποία βρίσκονται τα στοιχεία αυτά (ΕΛΟΤ).

command line = γραμμή εντολών. Μια γραμμή οθόνης, συνήθως η τελευταία, από την οποία μπορούν να δοθούν εντολές προς το λειτουργικό σύστημα.

comment = σχόλιο. Γλωσσικό δόμημα που επιτρέπει την εισαγωγή κειμένου σε ένα πρόγραμμα, χωρίς να υπάρχει αντίκτυπος στην εκτέλεση του προγράμματος (ISO).

communication = επικοινωνία. Μια επικοινωνία περιλαμβάνει τέσσερα βασικά στοιχεία: μια πηγή, ένα κανάλι επικοινωνίας, έναν προορισμό και ένα μήνυμα.

communications network = δίκτυο επικοινωνιών. Μια συλλογή διασυνδεδεμένων μονάδων που παρέχουν υπηρεσίες επικοινωνίας δεδομένων στους σταθμούς που είναι συνδεδεμένοι με το δίκτυο.

communications nodes = επικοινωνιακοί κόμβοι. Οι υπολογιστές που συνδέονται σε ένα δίκτυο.

compile = μεταγλωττίζω. Μεταφράζω σε γλώσσα χαμηλού επιπέδου πρόγραμμα υπολογιστή εκφρασμένο σε γλώσσα προανατολισμένη στο πρόβλημα (ISO).

compiler = μεταγλωττιστής. Πρόγραμμα υπολογιστή που χρησιμοποιείται για να μεταγλωττίζει (ISO).

computational complexity = υπολογιστική πολυπλοκότητα.

computational theory = θεωρία υπολογισμού.

computer program = πρόγραμμα υπολογιστή. Ακολουθία εντολών κατάλληλων για επεξεργασία. Η επεξεργασία μπορεί να περιλαμβάνει τη χρήση ενός συμβολομεταφραστή, ενός μεταγλωττιστή, ενός διερμηνευτή ή άλλου μεταφραστή, για να προετοιμάσει το πρόγραμμα για εκτέλεση καθώς και την ίδια την εκτέλεση του προγράμματος (ΕΛΟΤ).

computer science = επιστήμη των υπολογιστών, πληροφορική. Ο κλάδος της επιστήμης και τεχνολογίας που ασχολείται με μεθόδους και τεχνικές αναφερόμενες στην επεξεργασία δεδομένων που εκτελούνται με αυτόματα μέσα (ΕΛΟΤ).

condition = συνθήκη. 1) Μια έκφραση σε πρόγραμμα ή διαδικασία που μπορεί να εκτιμηθεί είτε ως αληθής είτε ως ψευδής, όταν εκτελείται το πρόγραμμα ή η διαδικασία. 2) Μια τιμή από ένα σύνολο συγκεκριμένων τιμών που μπορεί να λάβει ένα στοιχείο δεδομένου.

cryptography = κρυπτογραφία. Σύνολο τεχνικών μετατροπής ενός μηνύματος σε κωδικοποιημένη μορφή ώστε να μην γίνεται αντιληπτό από τρίτους.

DBMS – (Database Management System) = Σύστημα Διαχείρισης Βάσεων Δεδομένων – ΣΔΒΔ. Είναι αυτοτελής συλλογή από τμήματα λογισμικού (προγράμματα) για τη δημιουργία, επεξεργασία και συντήρηση των βάσεων δεδομένων.

data = δεδομένα. Παράσταση γεγονότων, εννοιών ή εντολών σε τυποποιημένη μορφή που είναι κατάλληλη για επικοινωνία, ερμηνεία ή επεξεργασία από άνθρωπο ή αυτόματα μέσα (ΕΛΟΤ).

data base = βάση δεδομένων. 1) Συλλογή δεδομένων με συγκεκριμένη δομή για αποδοχή, αποθήκευση και παροχή δεδομένων για πολλούς χρήστες σε κάθε ζήτηση. 2) Συλλογή διασυνδεδεμένων δεδομένων οργανωμένων σύμφωνα προς ένα σχήμα βάσης δεδομένων για την εξυπηρέτηση μιας ή περισσότερων εφαρμογών.

data dictionary = λεξικό δεδομένων. Ένα λεξικό των στοιχείων δεδομένων που χρησιμοποιούνται σε μια βάση δεδομένων ή σε κάποιο έργο λογισμικού μαζί με τις συσχετίσεις τους με άλλα δεδομένα και προγράμματα που τα χρησιμοποιούν.

data entry = εισαγωγή δεδομένων. Διαδικασία εισαγωγής από ειδικό προσωπικό των δεδομένων στον υπολογιστή για επεξεργασία.

data processing = επεξεργασία δεδομένων. Η συστηματική εκτέλεση πράξεων σε δεδομένα. Παραδείγματα: χειρισμός, συγχώνευση, ταξινόμηση, συμβολομετάφραση, μεταγλώττιση (ΕΛΟΤ).

data structure = δομή δεδομένων. Ένα σύνολο δεδομένων μαζί με τις επιτρεπτές λειτουργίες σε αυτά.

data transmission = μετάδοση δεδομένων. Η κάθε είδους μεταφορά δεδομένων σε απόσταση.

data type = τύπος δεδομένου. Στις γλώσσες προγραμματισμού, ένα σύνολο τιμών μαζί με ένα σύνολο επιτρεπόμενων πράξεων (ISO).

data validation = επικύρωση δεδομένων. Διεργασία που χρησιμοποιείται για να διαπιστώνεται η ορθότητα, η πληρότητα ή η λογικότητα των δεδομένων (ISO).

data value = τιμή δεδομένου.

debug = αίρω σφάλματα (σε προγραμματισμό). Ανιχνεύω, εντοπίζω και εξαλείφω σφάλματα σε προγράμματα υπολογιστή ή σε άλλο λογισμικό (ISO).

debugger = αποσφαλματωτής. Βοηθητικό πρόγραμμα -που συνήθως παρέχεται με τις σύγχρονες γλώσσες προγραμματισμού- και καθιστά αποδοτική την έρευνα για τον εντοπισμό σφαλμάτων σε ένα πρόγραμμα.

debugging = αποσφαλμάτωση, εκσφαλμάτωση. Έλεγχος της λογικής ενός προγράμματος για τον εντοπισμό και απομάκρυνση σφαλμάτων.

dimension = διάσταση. Ο αριθμός των δεικτών σε έναν πίνακα. Γενικότερα ο μέγιστος αριθμός και τάξη μιας σειράς από συσχετιζόμενα στοιχεία.

distributed data processing = κατανεμημένη επεξεργασία δεδομένων. Επεξεργασία δεδομένων στην οποία μερικές ή όλες από τις λειτουργίες επεξεργασίας, αποθήκευσης και ελέγχου, εκτός από τις λειτουργίες εισόδου/εξόδου, είναι διασπαρμένες μεταξύ των σταθμών επεξεργασίας δεδομένων (ISO).

documentation = τεκμηρίωση. 1) Διαχείριση τεκμηρίων. Η διαχείριση τεκμηρίων που μπορεί να περιλαμβάνει τις ενέργειες προσδιορισμού, απόκτησης, επεξεργασίας, αποθήκευσης και διανομής τους. 2) Συλλογή τεκμηρίων. Συλλογή τεκμηρίων σ' ένα δεδομένο θέμα (ISO).

editing = σύνταξη (προγραμμάτων, δεδομένων). 1) Μετασχηματισμός τιμών σε παραστάσεις που προδιαγράφονται από δεδομένο μορφότυπο. 2) Διαδικασία εγγραφής-διόρθωσης ενός προγράμματος ή προετοιμασίας δεδομένων με χρήση κατάλληλου προγράμματος.

editor = συντάκτης. Βοηθητικό πρόγραμμα υπολογιστή σχεδιασμένο για να εκτελεί λειτουργίες όπως εισαγωγή, επαναδιευθέτηση, τροποποίηση και διαγραφή δεδομένων σύμφωνα με προδιαγραφμένους κανόνες.

entity = οντότητα. Οποιοδήποτε συγκεκριμένο ή αφηρημένο αντικείμενο ενδιαφέροντος, συμπεριλαμβανομένων και των συσχετίσεων μεταξύ αντικειμένων. Για παράδειγμα ένα πρόσωπο, ένα αντικείμενο, ένα συμβάν ή μια διεργασία ενδιαφέροντος και τα σχετικά δεδομένα που πρέπει να αποθηκεύονται σε μια βάση δεδομένων.

evolutionary computation = εξελικτικός υπολογισμός. Είναι μια διαδικασία που ακολουθεί κάποιο καλά ορισμένο μοντέλο, το οποίο είναι κατανοητό και μπορεί να εκφραστεί με έναν αλγόριθμο, ένα πρωτόκολλο ή μια τοπολογία δικτύου.

executable = εκτελέσιμος. Κώδικας συχνά υπό μορφή αρχείου δίσκου τον οποίο μπορεί να εκτελέσει το εν χρήσει λειτουργικό σύστημα.

executable code = εκτελέσιμος κώδικας. Ένα σύνολο εντολών γλώσσας μηχανής για δεδομένο υπολογιστή ή μικροεπεξεργαστή, που μπορεί να εκτελεστεί απ' ευθείας χωρίς να χρειάζεται μεταγλώττιση.

expert system = έμπειρο σύστημα. Πρόγραμμα που επιδεικνύει νοήμονα συμπεριφορά σε συγκεκριμένους τομείς και διαδικασίες, ανάλογη ενός ανθρώπου εμπειρογνώμονα με ειδικότητα στον ίδιο τομέα.

expression = έκφραση. Γλωσσικό δόμημα για τον υπολογισμό μιας τιμής από έναν ή περισσότερους τελεστέους. Τελεστέοι μπορεί να είναι κυριολεκτικές σταθερές, αναγνωριστικά (ταυτότητας), αναφορές σε πίνακα, κλήσεις συναρτήσεων κλπ (ISO).

FIFO (First In-First Out) = πρώτος μέσα-πρώτος έξω. Τρόπος λειτουργίας μιας ουράς, κατά τον οποίο το πρώτο στοιχείο που εισάγεται είναι και το πρώτο που μπορεί να εξαχθεί.

FORTRAN (FORmula TRANslation). Η πρώτη γλώσσα προγραμματισμού υψηλού επιπέδου προσανατολισμένη σε επιστημονικά προβλήματα.

FTP (File Transfer Protocol) = πρωτόκολλο μεταφοράς αρχείου. Ένα πρωτόκολλο επικοινωνίας του Internet για τη διεκπεραίωση λειτουργιών μεταφοράς αρχείων.

factorial = παραγοντικό. Το γινόμενο των φυσικών αριθμών 1,2,3,... μέχρι ένα δεδομένο φυσικό ακέραιο (που συμπεριλαμβάνεται) (ΕΛΟΤ).

false = ψευδής. Λογική τιμή που μπορεί να αποδοθεί σε μεταβλητές λογικού τύπου.

field = πεδίο. Προκαθορισμένο τμήμα μιας εγγραφής το οποίο χρησιμοποιείται για κάποια ιδιαίτερη κατηγορία δεδομένων (ISO).

file = αρχείο. Σύνολο από συναφείς εγγραφές που υφίστανται επεξεργασία ως μια μονάδα. Βλ. και record.

function = συνάρτηση, λειτουργία. 1) Μαθηματική οντότητα της οποίας η τιμή εξαρτάται από τις τιμές μιας ή περισσότερων ανεξαρτήτων μεταβλητών κατά τρόπον που μια μοναδική τιμή της εξηρημένης μεταβλητής, δηλαδή της ίδιας της συνάρτησης, αντιστοιχεί σε κάθε επιτρεπόμενο συνδυασμό των τιμών καθορισμένων περιοχών των ανεξάρτητων μεταβλητών (ISO). 2) Σε γλώσσες προγραμματισμού, διαδικασία η

οποία όταν εκτελείται παρέχει μια τιμή και η κλήση της οποίας μπορεί να χρησιμοποιηθεί ως τελεστής σε μια έκφραση. Παράδειγμα: η συνάρτηση SIN παρέχει την τιμή του ημχ, όταν καλείται με SIN(X) (ISO).

functional programming = συναρτησιακός προγραμματισμός. Προγραμματισμός ο οποίος έχει ως βασικές του παραμέτρους συναρτήσεις

fuzzy logic = ασαφής λογική. Λογική η οποία εξάγεται με βάση τις εμπειρίες και τα συμπεράσματά μας.

gateway = πύλη (δικτύων). Λειτουργική μονάδα που διασυνδέει δύο δίκτυα υπολογιστών με διαφορετικές αρχιτεκτονικές δικτύου (ISO).

GIGO (Garbage In-Garbage Out) = άχρηστα στην είσοδο - άχρηστα στην έξοδο. Όρος που χρησιμοποιείται για να δηλώσει το γεγονός, ότι αν τα δεδομένα εισόδου είναι ακατάλληλα, τότε και τα αποτελέσματα θα είναι επίσης ακατάλληλα.

general-purpose language = γλώσσα γενικής χρήσης. Γλώσσα προγραμματισμού κατάλληλη για μια ευρεία περιοχή εφαρμογών.

genetic algorithm = γενετικός αλγόριθμος. Αλγόριθμος που χρησιμοποιεί την ιδέα της εξέλιξης μέσα από φυσικές επιλογές και διασταυρώσεις, ώστε να βρει λύσεις σε προβλήματα που μπορούν να περιγραφούν με αυτόν τον τρόπο.

graph = γράφος, γράφημα. Ένα σύνολο σημείων, κορυφών ή κόμβων και ένα σύνολο ακμών, τόξων ή γραμμών που ενώνουν μερικά ή όλα τα σημεία του.

hacker = (πληροφορικός) πειρατής. Ενθουσιώδης της πληροφορικής που χρησιμοποιεί τη γνώση του και τα μέσα για να αποκτήσει ανεξουσιοδότητη πρόσβαση σε προστατευμένους πόρους.

host computer = υπολογιστής υποδοχής. Σε δίκτυο υπολογιστών, κάθε υπολογιστής που παρέχει στους τελικούς χρήστες υπηρεσίες, όπως υπολογισμούς και πρόσβαση σε βάσεις δεδομένων και που μπορεί να επιτελεί λειτουργίες ελέγχου του δικτύου (ISO).

high-level language = γλώσσα υψηλού επιπέδου. Γλώσσα προγραμματισμού που προσομοιάζει της φυσικής γλώσσας και που χρειάζεται ένα μεταγλωττιστή ή ένα διερμηνευτή π.χ. Pascal, COBOL, FORTRAN, PROLOG, BASIC κλπ.

index = δείκτης, ευρετήριο. 1) Κατάλογος περιεχομένων ενός αρχείου ή ενός εγγράφου μαζί με τα κλειδιά ή τις παραπομπές για εντοπισμό των περιεχομένων. 2) Στον προγραμματισμό, ένας ακέραιος που αναγνωρίζει τη θέση ενός στοιχείου δεδομένων σε μια ακολουθία στοιχείων δεδομένων. 3) Σύμβολο ή αριθμός που χρησιμοποιείται για να αναγνωρίζει συγκεκριμένη ποσότητα σε πίνακα ομοίων ποσοτήτων. 4) Πίνακας που χρησιμοποιείται για να εντοπίζονται εγγραφές σε ένα σύνολο σειριακών δεδομένων με δείκτες ή σε ένα αρχείο με δείκτες.

informatics = πληροφορική.

information = πληροφορία (ες). Στην επεξεργασία πληροφοριών, γνώση που αφορά πράγματα όπως πράξεις, έννοιες, αντικείμενα, γεγονότα, ιδέες και διεργασίες, που μέσα σε συγκεκριμένο κείμενο έχουν μια ιδιαίτερη σημασία.

information processing = επεξεργασία πληροφοριών. Η συστηματική εκτέλεση πράξεων σε δεδομένα. Παραδείγματα: χειρισμός, συγχώνευση, ταξινόμηση, υπολογισμός, συμβολομετάφραση, μεταγλώττιση (ΕΛΟΤ).

information system = πληροφοριακό σύστημα. Σύστημα υπολογιστή που επεξεργάζεται δεδομένα, ώστε να προκύψουν πληροφορίες που μπορούν να χρησιμοποιηθούν στη λήψη αποφάσεων.

instruction = εντολή. Σε μια γλώσσα προγραμματισμού μια έκφραση που έχει νόημα και η οποία καθορίζει μια πράξη και προσδιορίζει τους τελεστές της, αν υπάρχουν (ISO).

instruction set = σύνολο εντολών. Το σύνολο των εντολών μηχανής ενός συγκεκριμένου υπολογιστή.

integer = ακέραιος. Ένας από τους αριθμούς 0, +1, -1, +2, -2, ...

interpreter = διερμηνευτής (σε προγραμματισμό).

1) Πρόγραμμα υπολογιστή που χρησιμοποιείται για να διερμηνεύει (ISO). 2) Πρόγραμμα που μεταφράζει και εκτελεί κάθε εντολή μιας γλώσσας υψηλού επιπέδου πριν τη μετάφραση και εκτέλεση της επόμενης εντολής.

iterative = επαναληπτικός.

kernel = πυρήνας. 1) Το μέρος ενός λειτουργικού συστήματος που εκτελεί βασικές λειτουργίες, όπως εκχώρηση πόρων υλικού. Συν. του nucleus. 2) Πρόγραμμα που εκτελείται σε διαφορετικά περιβάλλοντα λειτουργικού συστήματος. 3) Μέρος προγράμματος που πρέπει να είναι στην κύρια μνήμη για να φορτώνει άλλα μέρη του προγράμματος.

knowledge-based system = σύστημα βασισμένο στη γνώση. Σύστημα που εξαγεί λογικά συμπεράσματα από ένα σύνολο γνώσεων.

LIFO (Last-In-First-Out) = τελευταίος μέσα-πρώτος έξω. Τρόπος λειτουργίας σε μια δομή δεδομένων, κατά την οποία το στοιχείο που εισήχθη τελευταίο είναι και το πρώτο που αποσύρεται.

LISP (LISt Processing) = γλώσσα LISP. Γλώσσα προγραμματισμού σχεδιασμένη για επεξεργασία λιστών που χρησιμοποιείται εκτενώς για προβλήματα τεχνητής νοημοσύνης.

LOGO. Γλώσσα προγραμματισμού υψηλού επιπέδου κατάλληλη για μικρές ηλικίες.

language = γλώσσα. Σύνολο χαρακτήρων, συμβατικών παραδοχών και κανόνων που χρησιμοποιούνται για να διαβάσουν πληροφορίες (ISO).

language processor = επεξεργαστής γλώσσας. Πρόγραμμα υπολογιστή που εκτελεί λειτουργίες όπως μετάφραση, διερμηνεία και άλλες εργασίες που απαιτούνται για την επεξεργασία μιας ορισμένης γλώσσας προγραμματισμού (ISO).

library = βιβλιοθήκη. 1) Αρχείο ή σύνολο συσχετιζόμενων αρχείων. 2) Συλλογή συναρτήσεων, κλήσεων, υπορουτινών ή άλλων δεδομένων. 3) Αρχείο δεδομένων το οποίο περιέχει αρχεία και πληροφορίες ελέγχου που τους επιτρέπεται να προσπελαστούν μεμονωμένα.

line editor = συντάκτης γραμμής. Τύπος συντάκτη στον οποίο παρέχεται η δυνατότητα διόρθωσης μόνο σε επίπεδο γραμμής.

linkage editor = συνδέτης. Πρόγραμμα υπολογιστή που χρησιμοποιείται για να δημιουργεί μια φορτώσιμη ενότητα από μια ή περισσότερες αντικειμενικές ενότητες, που έχουν μεταφραστεί ανεξάρτητα ή από φορτώσιμες ενότητες με συμπλήρωση των αντιστοιχιών μεταξύ των κοινών αναφορών που χρησιμοποιούνται από διάφορες αντικειμενικές ενότητες και ενδεχόμενα με μεταθέσεις στοιχείων στη μνήμη (ISO).

linked list = συνδεσμική λίστα. Μια λίστα στην οποία η μετάβαση από έναν κόμβο στον επόμενο γίνεται με τη χρήση ενός δείκτη (pointer), που αποτελεί μέρος του κόμβου.

linker = συνδέτης. Βλ. linkage editor.

linking loader = συνδετικός φορτωτής. Βοηθητικός επεξεργαστής που εκτελεί σύνδεση και φόρτωση ενός μεταφρασμένου προγράμματος με σκοπό την εκτέλεσή του.

list = λίστα. Διαταγμένο σύνολο στοιχείων (ISO).

loader = φορτωτής. Πρόγραμμα υπολογιστή που μεταφέρει δεδομένα στην κύρια μνήμη.

local area network (LAN) = τοπικό δίκτυο. Δίκτυο υπολογιστών τοποθετημένο στο χώρο ενός χρήστη σε περιορισμένη γεωγραφική περιοχή (ISO).

local variable = τοπική μεταβλητή. Μεταβλητή της οποίας το όνομα και η τιμή μπορούν να χρησιμοποιηθούν μόνο μέσα στην ενότητα του προγράμματος όπου ορίζεται.

logic programming = λογικός προγραμματισμός. Μέθοδος κατασκευής προγραμμάτων ως σύνολα λογικών κανόνων με προκαθορισμένους αλγόριθμους για την επεξεργασία δεδομένων εισόδου.

logical constant = λογική σταθερά. Σταθερά με τιμή αληθής ή ψευδής.

logical expression = λογική έκφραση. Έκφραση που περιέχει λογικούς τελεστές και τελεστέους και που μπορεί να αντικατασταθεί με την τιμή αληθής ή ψευδής.

loop = βρόχος. Σύνολο εντολών που μπορεί να εκτελεστεί επανειλημμένα, όσο ισχύει μια ορισμένη συνθήκη (ISO).

machine language = γλώσσα μηχανής. Γλώσσα χαμηλού επιπέδου που οι εντολές της αποτελούνται μόνο από εντολές μηχανής (ISO).

machine learning = μηχανική μάθηση. Είναι η μάθηση σε ένα γνωστικό σύστημα όπως γίνεται αντιληπτή στην καθημερινή ζωή και συνδέεται με δύο βασικές ιδιότητες α) την ικανότητα στην πρόσκτηση γνώσης κατά την αλληλεπίδραση με το περιβάλλον και β) την ικανότητα να βελτιώνει με την επανάληψη τον τρόπο εκτέλεσης μία ενέργειας.

menu (list of options) = κατάλογος επιλογών. Κατάλογος δυνατών εργασιών που παρέχει σε μια συγκεκριμένη φάση εργασίας ένα προϊόν λογισμικού.

multiprocessing = πολυεπεξεργασία. Τρόπος εκμετάλλευσης που επιτρέπει παράλληλη επεξεργασία από δύο ή περισσότερους επεξεργαστές ενός πολυεπεξεργαστή (ISO).

multiprogramming = πολυπρογραμματισμός. Τρόπος εκμετάλλευσης που επιτρέπει διεμπλεκόμενη εκτέλεση δύο ή περισσότερων προγραμμάτων υπολογιστή από έναν και μόνο επεξεργαστή (ISO).

multitasking = πολυδιεργασία. Τρόπος εκμετάλλευσης που επιτρέπει τη συντρέχουσα διεξαγωγή ή τη διεμπλεκόμενη εκτέλεση δύο ή περισσότερων στοιχειωδών εργασιών (ISO).

multiuser system = σύστημα πολλών χρηστών. Σύστημα με δυνατότητα πρακτικά ταυτόχρονης υποστήριξης πολλών χρηστών.

NOT operation = πράξη ΟΧΙ, άρνηση, λογική αντιστροφή. Η μοναδιαία πράξη άλγεβρας Boole της οποίας το αποτέλεσμα έχει τιμή αντίθετη από αυτήν του τελεστέου (ISO).

network = δίκτυο. Ένα σύνολο κόμβων μαζί με τους κλάδους διασύνδεσής τους (ISO).

network topology = τοπολογία δικτύου. Η διάταξη των συνδέσεων και κόμβων σε ένα δίκτυο.

networking = δικτύωση. Η σύνδεση γεωγραφικά χωρισμένων υπολογιστών μέσω γραμμών επικοινωνίας.

node = κόμβος. 1) Σε δίκτυο δεδομένων, σημείο όπου μία ή περισσότερες λειτουργικές μονάδες διασυνδέουν

κανάλια ή κυκλώματα δεδομένων (ISO). 2) Η παράσταση μιας κατάστασης ή ενός γεγονότος με τη χρήση ενός σημείου σε ένα διάγραμμα. 3) Τα στοιχεία μιας δομής δένδρου ή συνδεδεμένης λίστας.

nucleus = πυρήνας. Το τμήμα προγράμματος ελέγχου που διαμένει στην κύρια μνήμη (ISO).

numeric character = αριθμητικός χαρακτήρας. Χαρακτήρας που ανήκει στο σύνολο των ψηφίων 0 έως 9.

OSI (Open Systems Interconnection) = διασύνδεση ανοικτών συστημάτων. Η διασύνδεση συστημάτων υπολογιστών σύμφωνα με τα πρότυπα ISO και τις συστάσεις CCITT για την ανταλλαγή δεδομένων (ISO).

object oriented language = αντικειμενοστρεφής γλώσσα. Γλώσσα που αντανακλά τις έννοιες του αντικειμενοστρεφούς προγραμματισμού.

object oriented programming = αντικειμενοστρεφής προγραμματισμός. Μέθοδος για δόμηση προγραμμάτων ως ιεραρχικά οργανωμένες τάξεις, που περιγράφουν τα δεδομένα και τις λειτουργίες αντικειμένων, που μπορούν να αλληλεπιδρούν με άλλα αντικείμενα.

object program = πρόγραμμα αντικειμένου, τελικό πρόγραμμα. Πρόγραμμα υπολογιστή σε μια τελική γλώσσα που έχει μεταφραστεί από μια γλώσσα πηγής (ISO).

operand = τελεστέος. Μια οντότητα στην οποία εφαρμόζεται μια πράξη (ΕΛΟΤ).

operating system = λειτουργικό σύστημα. Λογισμικό που ελέγχει την εκτέλεση των προγραμμάτων και που μπορεί να παρέχει υπηρεσίες όπως εκχώρηση πόρων, χρονοπρογραμματισμό, έλεγχο εισόδου/εξόδου και διαχείριση δεδομένων. Αν και τα λειτουργικά συστήματα είναι κατά το μέγιστο μέρος λογισμικό, είναι δυνατές μερικές ή ολικές υλοποιήσεις με υλικό (ΕΛΟΤ).

operation = πράξη. Μια σαφώς ορισμένη ενέργεια, η οποία όταν εφαρμόζεται σε οποιοδήποτε επιτρεπτό συνδυασμό γνωστών οντοτήτων, παράγει μια νέα οντότητα. Για παράδειγμα, η διεργασία της πρόσθεσης στην αριθμητική, όταν προστίθονται το ένα και το δύο με αποτέλεσμα τρία, οι αριθμοί ένα και δύο είναι οι τελεστέοι, ο αριθμός τρία είναι το αποτέλεσμα και το σημείο συν είναι ο τελεστής που δείχνει ότι η πράξη που εκτελείται είναι η πρόσθεση (ΕΛΟΤ).

operator = τελεστής, χειριστής. 1) Σύμβολο που παριστάνει τη φύση μιας πράξης που πρόκειται να εκτελεστεί (ΕΛΟΤ). 2) Πρόσωπο που χειρίζεται μια συσκευή.

parallel programming = παράλληλος προγραμματισμός. Η ανάπτυξη εφαρμογών που εκμεταλλεύονται ταυτόχρονα την ύπαρξη πολλαπλών επεξεργαστών σε ένα υπολογιστικό σύστημα με σκοπό την αύξηση των υπολογιστικών επιδόσεων και τη μείωση του χρόνου εκτέλεσης της εφαρμογής.

Pascal. Δημοφιλής γλώσσα τρίτης γενιάς που εισήχθη το 1971.

pattern recognition = αναγνώριση προτύπων. Η αναγνώριση ταυτότητας σχημάτων, μορφών ή διατάξεων με αυτόματα μέσα.

pointer = δείκτης. 1) Στοιχείο δεδομένου που περιέχει τη διεύθυνση ενός άλλου στοιχείου δεδομένου. 2) Οπτικά παρουσιαζόμενο σύμβολο στην οθόνη το οποίο ένας χρήστης μετακινεί με μια συσκευή στίξης, όπως ένα ποντίκι.

procedural = διαδικασιακός, διαδικαστικός.

procedural language = διαδικασιακή γλώσσα. Γλώσσα προσανατολισμένη στο πρόβλημα που διευκολύνει την έκφραση μιας διαδικασίας με τη μορφή ρητά εκφρασμένου αλγόριθμου (ISO).

procedure = διαδικασία. Στις γλώσσες προγραμματισμού, μια ομάδα εντολών με ή χωρίς τυπικές παραμέτρους, η εκτέλεση της οποίας προκαλείται μέσω μιας κλήσης διαδικασίας.

processing = επεξεργασία. Η εκτέλεση λογικών πράξεων και υπολογισμών σε δεδομένα συμπεριλαμβανομένης και της προσωρινής διατήρησης των δεδομένων στη μνήμη του επεξεργαστή.

program = πρόγραμμα (υπολογιστή). Ακολουθία εντολών κατάλληλων για επεξεργασία. Η επεξεργασία μπορεί να περιλαμβάνει τη χρήση ενός συμβολομεταφραστή, ενός μεταγλωττιστή, ενός διερμηνευτή ή άλλου μεταφραστή, για να προετοιμάσει το πρόγραμμα για εκτέλεση καθώς και την ίδια την εκτέλεση του προγράμματος (ΕΛΟΤ).

program documentation = τεκμηρίωση προγράμματος. Έγγραφα ή άλλα μέσα που παρέχουν την περιγραφή του προγράμματος. Η τεκμηρίωση προγράμματος κανονικά περιλαμβάνει ένα αντίγραφο λίστας προγράμματος, αποτελέσματα δοκιμών, ιστορικό τροποποιήσεων στο πρόγραμμα και άλλες πληροφορίες.

program specification = προδιαγραφή προγράμματος. Τεκμήριο που περιγράφει τη δομή και λειτουργίες ενός προγράμματος με επαρκή λεπτομέρεια, ώστε να επιτρέπει τον προγραμματισμό και να διευκολύνει τη συντήρηση (ISO).

program testing = δοκιμή προγράμματος. Ένα βήμα της ανάπτυξης προγράμματος όπου ένα πλήρες πρόγραμμα δοκιμάζεται για σφάλματα. Μια δοκιμή προγράμματος εμπεριέχει την αποσφαλμάτωση του προγράμματος.

program testing and debugging = δοκιμή και αποσφαλμάτωση προγράμματος. Διαδικασίες εντοπισμού και διόρθωσης τυχόν σφαλμάτων ενός προγράμματος.

program verification = επαλήθευση προγράμματος. Αποδεικνύει ότι ένα πρόγραμμα συμπεριφέρεται σύμφωνα προς τις προδιαγραφές του.

programmer = προγραμματιστής. Πρόσωπο υπεύθυνο για το σχεδιασμό, εγγραφή, έλεγχο, διόρθωση, συντήρηση και τεκμηρίωση ενός προγράμματος.

programming = προγραμματισμός. Η διαδικασία δημιουργίας προγραμμάτων υπολογιστή.

programming language = γλώσσα προγραμματισμού. Τεχνητή γλώσσα σχεδιασμένη για να δημιουργεί ή να εκφράζει προγράμματα (ISO).

protocol = πρωτόκολλο. Σύνολο κανόνων της σημασιολογίας και του συντακτικού που καθορίζει τη συμπεριφορά των λειτουργικών μονάδων στην επίτευξη επικοινωνίας (ISO).

pseudocode = ψευδοκώδικας. 1) Κώδικας που απαιτεί μετάφραση πριν από την εκτέλεση (ISO). 2) Τρόπος αποτύπωσης αλγορίθμων με χρήση προκαθορισμένων λέξεων κλειδιών.

push = ωθώ, σπρώχνω. Εισάγω ένα στοιχείο δεδομένων στην κορυφή μιας στοίβας.

query = ερώτηση. Μια αίτηση για δεδομένα από βάση δεδομένων που βασίζεται σε ειδικές συνθήκες.

Παράδειγμα, η ερώτηση για διαθεσιμότητα μιας θέσης σε ένα σύστημα δέσμευσης πτήσης.

query language = γλώσσα ερωταποκρίσεων. Μια γλώσσα χειρισμού δεδομένων για τελικούς χρήστες προκειμένου να ανακαλούν ή να τροποποιούν δεδομένα σε μια βάση δεδομένων

queue = ουρά. Δομή δεδομένων με δύο άκρα στην οποία νέα στοιχεία εισάγονται από το ένα άκρο και υπάρχοντα στοιχεία εξέρχονται από το άλλο άκρο. Βλ. και FIFO.

quicksort = γρήγορη ταξινόμηση. Αλγόριθμος ταξινόμησης που βασίζεται στο διαμερισμό του συνόλου των στοιχείων σε δύο μέρη, στην ανταλλαγή των πιο απομακρυσμένων στοιχείων και στην αναδρομική κλήση του για καθένα από τα δύο μέρη. Η μέθοδος αυτή είναι η καλύτερη γνωστή μέχρι σήμερα για εσωτερική ταξινόμηση.

relation data model (RDM) = σχεσιακό μοντέλο δεδομένων. Είναι το είδος του λογικού μοντέλου βάσεων δεδομένων που μπορεί να συνδυάζει τα δεδομένα ενός πίνακα με τα δεδομένα ενός άλλου, αρκεί οι δυο πίνακες να έχουν ένα κοινό στοιχείο δεδομένων.

record = εγγραφή. Στις γλώσσες προγραμματισμού, συνάθροισμα που αποτελείται από αντικείμενα σχετικά με δεδομένα με πιθανόν διαφορετικά ιδιοχαρακτηριστικά,

στα οποία αντικείμενα έχουν συνήθως προσαρτηθεί αναγνωριστικά ταυτότητας (ISO).

recursion = αναδρομή. Κατάσταση κατά την οποία μια διαδικασία ή συνάρτηση καλεί τον εαυτό της.

recursive function = αναδρομική συνάρτηση. Μια συνάρτηση της οποίας οι τιμές είναι φυσικοί αριθμοί που παράγονται από φυσικούς αριθμούς με τύπους αντικατάστασης, στους οποίους η ίδια η συνάρτηση είναι ένας τελεστής (ΕΛΟΤ).

repeater = επαναλήπτης. Σε κόμβο τοπικού δικτύου, μια συσκευή που αναγεννά σήματα για να επεκτείνει την εμβέλεια μετάδοσης μεταξύ σταθμών δεδομένων ή για να διασυνδέει δύο κλάδους.

reserved word = δεσμευμένη λέξη. Λέξη γλώσσας πηγής η σημασία της οποίας καθορίζεται από τους κανόνες αυτής της γλώσσας και δεν μπορεί να αλλάξει για την ευκολία οποιουδήποτε προγράμματος εκφρασμένου σε αυτή τη γλώσσα. Μάλιστα στις περισσότερες περιπτώσεις προγραμμάτων εκφρασμένων στη γλώσσα πηγής, μπορεί να απαγορεύεται η χρήση τέτοιων λέξεων σε άλλα συμφραζόμενα του προγράμματος. Παραδείγματα: 1. Η λέξη SIN μπορεί να είναι μια δεσμευμένη λέξη για την κλήση μιας υπορουτίνας υπολογισμού του ημιτόνου. 2. Λέξεις της COBOL όπως OCCURS, MOVE, COMPUTE κ.α.

retrieval = ανάκτηση. Διαδικασία αναζήτησης ενός στοιχείου με σκοπό τη λήψη (ανάγνωση) του περιεχομένου του.

robot = ρομπότ. Μια μηχανική συσκευή η οποία μπορεί να προγραμματιστεί για να εκτελεί κάποια εργασία χειρισμού ή μετακίνηση από ένα τόπο σε άλλο υπό συνθήκες αυτομάτου ελέγχου.

robotics = ρομποτική. Οι τεχνικές που χρησιμοποιούνται για σχεδιασμό, κατασκευή και χρήση ρομπότ.

router = δρομολογητής. Υπολογιστής που προσδιορίζει τη διαδρομή κυκλοφορίας δικτύου. Η επιλογή διαδρομής γίνεται από αρκετές διαδρομές που βασίζονται σε πληροφορίες, οι οποίες λαμβάνονται από ειδικά πρωτόκολλα και αλγορίθμους, που προσπαθούν να αναγνωρίζουν τη συντομότερη ή καλύτερη διαδρομή και άλλα κριτήρια.

routine = ρουτίνα. Ένα πρόγραμμα καλούμενο από άλλο πρόγραμμα, που μπορεί να έχει γενική ή συχνή χρήση (ΕΛΟΤ).

running time = χρόνος εκτέλεσης. Ο χρόνος που απαιτείται για την εκτέλεση ενός προγράμματος.

SQL (Structured Query Language) = δομημένη γλώσσα ερωταπαντήσεων. Γλώσσα ερωταπαντήσεων πολύ υψηλού επιπέδου που χρησιμοποιείται για αναζητήσεις σε βάσεις δεδομένων.

search = αναζητώ, ψάχνω. Εξετάζω σύνολο στοιχείων δεδομένων για να εντοπίσω ένα ή περισσότερα στοιχεία που έχουν μια δεδομένη ιδιότητα (ISO).

searching = αναζήτηση, ψάξιμο.

searching techniques = τεχνικές αναζήτησης. Σύνολο τεχνικών με βάση τις οποίες επιτυγχάνεται η λειτουργία της αναζήτησης.

selection sort = ταξινόμηση με επιλογή. Τεχνική ταξινόμησης κατά την οποία βρίσκεται το ελάχιστο στοιχείο μιας ομάδας και αντιμετωπίζεται με το πρώτο, στη συνέχεια βρίσκεται το ελάχιστο στοιχείο των υπολοίπων και αντιμετωπίζεται με το δεύτερο κ.ο.κ.

sequential = ακολουθιακός. Τρόπος επεξεργασίας κατά τον οποίο δύο ή περισσότερες πράξεις εκτελούνται η μία μετά την άλλη (ΕΛΟΤ).

sequential search = ακολουθιακή ή σειριακή αναζήτηση.

simplex line = ημιαμφίδρομη γραμμή, μονόδρομη γραμμή. Μια γραμμή επικοινωνίας που χρησιμοποιείται για τη μετάδοση δεδομένων κατά μία μόνο κατεύθυνση.

software = λογισμικό. Πνευματική δημιουργία που περιλαμβάνει τα προγράμματα, τις διαδικασίες, τους

κανόνες και οποιαδήποτε σχετική τεκμηρίωση που αναφέρεται στη λειτουργία ενός συστήματος επεξεργασίας δεδομένων (ΕΛΟΤ).

software development environment = περιβάλλον ανάπτυξης λογισμικού. Σύνολο μεταφραστικών προγραμμάτων και άλλων εργαλείων ανάπτυξης λογισμικού που χρησιμοποιούνται στη δημιουργία προγραμμάτων εφαρμογών.

software development tool = εργαλείο ανάπτυξης λογισμικού. Ένα σύνολο προγραμμάτων που βοηθά στην ανάπτυξη ορισμένων τύπων λογισμικού εφαρμογών.

software engineer = μηχανικός λογισμικού. Πρόσωπο που σχεδιάζει και γράφει προγράμματα υπολογιστών σύμφωνα με τις αρχές της τεχνολογίας λογισμικού.

software life cycle = κύκλος ζωής λογισμικού. Ο κύκλος σχεδίασης, ανάπτυξης, εγκατάστασης και συντήρησης λογισμικού.

software package = πακέτο λογισμικού. Πλήρες και τεκμηριωμένο σύνολο προγραμμάτων που παρέχεται σε έναν αριθμό χρηστών για μια εφαρμογή (ISO).

software piracy = πειρατεία λογισμικού. Παράνομη αναπαραγωγή προϊόντων λογισμικού.

software reusability = δυνατότητα επαναχρήσης λογισμικού, επαναχρηστικότητα λογισμικού. Επιδιωκόμενη ιδιότητα ενός προϊόντος λογισμικού σύμφωνα με την οποία μπορεί το εν λόγω προϊόν να χρησιμοποιηθεί και για τη σύσταση ενός άλλου.

software testing = έλεγχος λογισμικού. Φάση της ανάπτυξης ενός προϊόντος λογισμικού κατά την οποία το προϊόν ελέγχεται ως προς την αναμενόμενη λειτουργία του.

software tools = εργαλεία λογισμικού. Ειδικά προγράμματα που διευκολύνουν συγκεκριμένες εργασίες της ανάπτυξης λογισμικού.

sorting = ταξινόμηση. Η διαδικασία τοποθέτησης των στοιχείων δεδομένων σε μια δομή δεδομένων με αύξουσα ή φθίνουσα σειρά.

source code = πηγαίος κώδικας. Η είσοδος σε μεταγλωττιστή ή συμβολομεταφραστή που έχει γραφεί σε μια γλώσσα πηγής.

source program = πηγαίο πρόγραμμα, αρχικό πρόγραμμα. Πρόγραμμα υπολογιστή εκφρασμένο σε μια γλώσσα πηγής (ISO).

specification = προδιαγραφή. Λεπτομερής διατύπωση υπό μορφή τεκμηρίου, που παρέχει οριστική περιγραφή ενός συστήματος με σκοπό την ανάπτυξη ή επικύρωση του συστήματος (ISO).

square root = τετραγωνική ρίζα.

stack = στοίβα. Δομή δεδομένων με ένα μόνο άκρο, στην οποία το τελευταίο στοιχείο που μπήκε είναι και το πρώτο που μπορεί να εξαχθεί. Βλ. και LIFO.

statement = εντολή, πρόταση (προγράμματος). Γλωσσικό δόμημα που παριστάνει ένα βήμα σε μια ακολουθία ενεργειών ή σε ένα σύνολο δηλώσεων (ISO).

structured programming = δομημένος προγραμματισμός. Μέθοδος για την κατασκευή προγραμμάτων που χρησιμοποιεί μόνο ιεραρχικά εμπερικλειόμενα κατασκευάσματα, καθένα από τα οποία έχει ένα απλό σημείο εισόδου και ένα εξόδου. Τρεις τύποι ελέγχου χρησιμοποιούνται στο δομημένο προγραμματισμό: ακολουθία, επιλογή και επανάληψη.

swarm intelligence = νοημοσύνη σμήνους. Μέθοδος διαδικασιών βελτιστοποίησης υπολογισμών που προέκυψαν παρατηρώντας πρότυπα που υπάρχουν στη φύση.

subprogram = υποπρόγραμμα. Ένα πρόγραμμα καλούμενο από άλλο πρόγραμμα, σε αντίθεση με ένα κύριο πρόγραμμα.

subroutine = υπορουτίνα. Ακολουθιακό σύνολο εντολών ή προτάσεων που μπορούν να χρησιμοποιηθούν σε ένα ή περισσότερα σημεία ενός προγράμματος υπολογιστή.

switching elements = στοιχεία μεταγωγής. Πρόκειται για τις ενδιάμεσες συσκευές που συνδέουν τις γραμμές μετάδοσης και επιφορτίζονται με το έργο της δρομολόγησης των δεδομένων από τη μια γραμμή στην άλλη ή από το ένα δίκτυο στο άλλο.

symbolic language = συμβολική γλώσσα.

symbolic logic = συμβολική λογική. Ο επιστημονικός κλάδος στον οποίο οι συλλογισμοί και οι πράξεις αντιμετωπίζονται με τη χρήση μιας τεχνητής γλώσσας σχεδιασμένης έτσι, ώστε να αποφεύγονται τα διαφορούμενα και η λογική ανεπάρκεια των φυσικών γλωσσών (ΕΛΟΤ).

system analysis = ανάλυση συστήματος. Συστηματική έρευνα ενός πραγματικού ή σχεδιασμένου συστήματος για να προσδιοριστούν οι απαιτήσεις πληροφοριών και οι διεργασίες του συστήματος και πώς αυτά συνδέονται μεταξύ τους και με οποιοδήποτε άλλο σύστημα (ISO).

system software = λογισμικό συστήματος. Λογισμικό ανεξάρτητο της εφαρμογής, που υποστηρίζει την εκτέλεση του λογισμικού εφαρμογής (ISO).

table = πίνακας. 1) Παράθεση δεδομένων καθένα από τα οποία μπορεί να προσδιοριστεί μονοσήμαντα μέσω μιας ή περισσότερων μεταβλητών (ISO). 2) Ένα αντικείμενο βάσης δεδομένων που αποτελείται από μια ομάδα γραμμών (εγγραφές), οι οποίες χωρίζονται σε στήλες (πεδία) που περιέχουν δεδομένα.

telecommunication = τηλεπικοινωνία. 1) Η μετάδοση σημάτων ελέγχου και πληροφοριών μεταξύ δύο ή περισσοτέρων θέσεων, όπως από τηλέγραφο, ραδιόφωνο ή τηλεόραση. 2) Η μετάδοση δεδομένων μεταξύ συστημάτων υπολογιστών από γραμμές τηλεπικοινωνίας και μεταξύ ενός συστήματος υπολογιστή και απομακρυσμένων συσκευών.

test = δοκιμή. Η λειτουργία μιας μονάδας και η σύγκριση του επιτευχθέντος αποτελέσματος με το προβλεπόμενο, προκειμένου να γίνει αποδεκτή ή μη η μονάδα. Π.χ. δοκιμή συσκευής, δοκιμή προγράμματος.

test data = δεδομένα δοκιμής. Τα δεδομένα που χρησιμοποιούνται για ένα ελεγκτικό πρόβλημα (ISO).

text editor = συντάκτης κειμένου. Πρόγραμμα υπολογιστή που δίνει στο χρήστη τη δυνατότητα να δημιουργεί και να αναθεωρεί κείμενα. Ο συντάκτης κειμένου, αν και επιτελεί το ίδιο έργο με τον επεξεργαστή κειμένου, συνήθως έχει μικρότερες δυνατότητες από τον τελευταίο.

third generation language = γλώσσα τρίτης γενεάς. Η γενιά των γλωσσών προγραμματισμού που εμφανίστηκαν τη δεκαετία του 60 και ακολούθησαν τις γλώσσες μηχανής και τις συμβολικές γλώσσες. Είναι γλώσσες υψηλού επιπέδου που χρησιμοποιούν κάποιο μεταφραστικό πρόγραμμα.

topology = τοπολογία. Ο χάρτης ή το σχέδιο ενός δικτύου. Η φυσική τοπολογία περιγράφει τη διάταξη των καλωδίων και η λογική ή ηλεκτρική τοπολογία περιγράφει τη ροή των μηνυμάτων.

translate = μεταφράζω, μεταφέρω. 1) Σε γλώσσες προγραμματισμού, ο μετασχηματισμός όλου ή μέρους ενός προγράμματος που εκφράζεται σε μια γλώσσα προγραμματισμού, από τη γλώσσα αυτή σε μια άλλη γλώσσα ή σε μια γλώσσα μηχανής κατάλληλη για εκτέλεση. 2) Σε γραφικά υπολογιστή, η μετατόπιση μιας εικόνας σε χώρο οπτικής παρουσίασης από μια θέση στην άλλη χωρίς την περιστροφή της.

translator = μεταφραστής. Πρόγραμμα υπολογιστή που μεταφράζει από μια γλώσσα σε άλλη γλώσσα και ιδιαίτερα από μια γλώσσα προγραμματισμού σε άλλη γλώσσα προγραμματισμού (ISO).

transmission = εκπομπή. Η αποστολή δεδομένων από ένα μέρος για λήψη οπουδήποτε.

transmission medium = μέσο εκπομπής. Το φυσικό μέσο που μεταφέρει σήματα μεταξύ σταθμών δεδομένων π.χ. σύρμα συνεστραμένου ζεύγους, οπτική ίνα, ομοαξονικό καλώδιο.

transmit = εκπέμπω.

transmitter = πομπός.

tree = δένδρο. Δομή δεδομένων που αποτελείται από κόμβους, οι οποίοι συνδέονται με ακμές. Σε κάθε κόμβο καταλήγει μία μόνο ακμή, αλλά μπορεί να ξεκινούν καμία, μία ή περισσότερες ακμές. Σε ένα μόνο κόμβο που αποκαλείται ρίζα, δεν καταλήγει καμία ακμή.

UNIX operating system = λειτουργικό σύστημα UNIX. Λειτουργικό σύστημα που αναπτύχθηκε από τα εργαστήρια BELL και το χαρακτηρίζει ο πολυπρογραμματισμός σε περιβάλλον πολλαπλών χρηστών. Το UNIX αναπτύχθηκε αρχικά για χρήση σε μινι-υπολογιστές αλλά προσαρμόστηκε για μεγάλους υπολογιστές καθώς και για μικρο-υπολογιστές.

unary operation = μοναδιαία πράξη, μονοτελής πράξη. Μια πράξη με έναν και μόνον έναν τελεστέο. Παράδειγμα: η άρνηση (ΕΛΟΤ).

user = χρήστης. Πρόσωπο που χρησιμοποιεί το σύστημα του υπολογιστή.

user friendly system = σύστημα φιλικό προς το χρήστη. Χαρακτηριστικό ενός συστήματος ή εφαρμογής που δηλώνει ευκολία στη χρήση του.

user interface = διεπαφή χρήστη. Υλικό, λογισμικό ή και τα δύο, με τα οποία επιτρέπεται να αλληλεπιδρά ο χρήστης και να εκτελεί λειτουργίες στο σύστημα, στο πρόγραμμα ή στη συσκευή.

user manual = εγχειρίδιο χρήστη

validation = επικύρωση. Διεργασία που χρησιμοποιείται για να διαπιστώνεται η ορθότητα, η πληρότητα ή λογικότητα των δεδομένων (ISO).

validity check = έλεγχος εγκυρότητας. Έλεγχος για να προσδιορίζει, αν μια ομάδα κώδικα είναι πραγματικά χαρακτήρες του δεδομένου κώδικα σε χρήση.

variable = μεταβλητή. 1) Οντότητα της οποίας η τιμή μπορεί να είναι απροσδιόριστη -ή απροσδιόριστη μέσα σε γνωστά όρια- μέχρι να αποδοθεί μια πραγματική τιμή σε αυτή σε δεδομένη εφαρμογή (ΕΛΟΤ). 2) Στις γλώσσες προγραμματισμού, ένα γλωσσικό αντικείμενο που μπορεί να λάβει διάφορες τιμές, μία κάθε φορά. Οι τιμές μιας μεταβλητής περιορίζονται συνήθως σε έναν τύπο δεδομένων. 3) Ένα όνομα που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου, του οποίου η τιμή μπορεί ν' αλλάζει κατά τη διάρκεια λειτουργίας του προγράμματος.

verification = επαλήθευση, επιβεβαίωση. Η ενέργεια του προσδιορισμού αν μια λειτουργία έχει επιτελεσθεί ορθά.

version = εκδοχή, παραλλαγή, έκδοση. Όρος που χρησιμοποιείται για την αναγνώριση ενός συγκεκριμένου προϊόντος λογισμικού. Όταν το προϊόν τίθεται σε κυκλοφορία ή πωλείται για πρώτη φορά, αποκαλείται πρώτη έκδοση. Καθώς το προϊόν τροποποιείται ή βελτιώνεται, γίνονται διαθέσιμες μεταγενέστερες εκδόσεις.

virus = ιός. Ένα αυτοαναπαραγόμενο πρόγραμμα το οποίο «μολύνει» και μπορεί να καταστρέψει άλλα προγράμματα και δεδομένα.

WYSIWYG (What You See Is What You Get) = ό,τι βλέπεις είναι αυτό που παίρνεις. Όρος που χρησιμοποιείται για να περιγράψει συστήματα, στα οποία υπάρχει η δυνατότητα να αποτυπωθεί στον εκτυπωτή ό,τι ακριβώς φαίνεται στην οθόνη (προφέρεται γουί-σγουινγκ).

wide area network (WAN) = δίκτυο ευρείας περιοχής. Δίκτυο που παρέχει υπηρεσίες επικοινωνιών σε γεωγραφική περιοχή ευρύτερη από αυτή που εξυπηρετείται από τοπικό δίκτυο ή το μητροπολιτικό δίκτυο και που μπορεί να χρησιμοποιεί ή να παρέχει δημόσιες ευκολίες επικοινωνίας.

window = παράθυρο. 1) Τμήμα μιας επιφάνειας οπτικής παρουσίασης στην οποία μπορούν να παρουσιαστούν εικόνες που ανήκουν σε ιδιαίτερη εφαρμογή. Διαφορετικές εφαρμογές μπορούν να παρουσιαστούν οπτικά ταυτόχρονα σε διαφορετικά παράθυρα. 2) Περιοχή της οθόνης με ορατά σύνορα μέσα στην οποία παρουσιάζονται οπτικά πληροφορίες. Ένα παράθυρο μπορεί να είναι μικρότερο ή του ίδιου μεγέθους με την οθόνη.

XML - (Extensible Markup Language). Γλώσσα η οποία χρησιμοποιώντας ένα σύνολο κανόνων επιτρέπει την αναπαράσταση δεδομένων με τρόπο τέτοιο που να γίνεται κατανοητός όχι μόνο από τους ανθρώπους αλλά και από τους υπολογιστές.

**Σημ. ISO = Διεθνής Οργανισμός Τυποποίησης,
ΕΛΟΤ = Ελληνικός Οργανισμός Τυποποίησης**

Περιεχόμενα

Κεφάλαιο 2.3. Προγραμματισμός	5
2.3.1. Αναφορά σε γλώσσες προγραμματισμού και «Προγραμματιστικά Υποδείγματα»	6
2.3.1.1. Πρόγραμμα και Γλώσσες Προγραμματισμού	6
2.3.1.2. Προγραμματιστικά Υποδείγματα	15
2.3.1.3. Δομημένος Προγραμματισμός	21
2.3.2. Σχεδίαση και συγγραφή κώδικα	27
2.3.3. Κύκλος ζωής εφαρμογής λογισμικού	55

ΕΝΟΤΗΤΑ 3. ΘΕΜΑΤΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΕΠΙΣΤΗΜΗΣ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Κεφάλαιο 3.1. Λειτουργικά Συστήματα	71
3.1.1. Λογισμικό και Υπολογιστικό Σύστημα	72
3.1.2. Το Λειτουργικό Σύστημα και οι Αρμοδιότητές του	72
3.1.3. Η Δομή και η Ιεραρχία ενός Λειτουργικού Συστήματος	74
3.1.4. Βασικές Εργασίες του Λ.Σ.	76
3.1.4.1. Διαχείριση της ΚΜΕ	76
3.1.4.2. Διαχείριση της Μνήμης	77
3.1.4.3. Διαχείριση του Συστήματος Αρχείων	78
3.1.4.4. Διαχείριση Λειτουργιών Εισόδου /Εξόδου	79
3.1.5. Γνωστά Λειτουργικά Συστήματα	80

Κεφάλαιο 3.2. Πληροφοριακά Συστήματα	85
3.2.1. Τι είναι τα Πληροφοριακά Συστήματα	86
3.2.2. Αρχιτεκτονικές Αποθήκευσης	89
3.2.3. Βάσεις Δεδομένων	91
3.2.4. Γλώσσες Ερωτοαποκρίσεων (SQL, XML)	95
Κεφάλαιο 3.3. Δίκτυα	99
3.3.1. Τι είναι ένα Δίκτυο Υπολογιστών	100
3.3.2. Στοιχεία δικτύων	101
3.3.3. Κατηγορίες δικτύων	103
3.3.3.1. Είδη δικτύων ανάλογα με την τεχνολογία μετάδοσης	103
3.3.3.2. Είδη δικτύων ανάλογα με την τεχνολογία προώθησης της πληροφορίας	103
3.3.3.3. Είδη δικτύων βάσει περιοχής που καλύπτουν	105
3.3.4. Τοπολογίες Δικτύων	106
3.3.5. Σύγχρονες υπηρεσίες δικτύων	110
Κεφάλαιο 3.4. Τεχνητή Νοημοσύνη	115
3.4.1. Τι είναι η Τεχνητή Νοημοσύνη	116
3.4.2. Εξέλιξη της Τεχνητής Νοημοσύνης	120
3.4.3. Τομείς εφαρμογών της Τεχνητής Νοημοσύνης	124
3.4.4. Γλώσσες προγραμματισμού που χρησιμοποιούνται στην Τ.Ν.	126
Βιβλιογραφία	129
Γλωσσάριο	131

Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').

Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων / ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.