

ΕΠΛ 002:
ΕΙΣΑΓΩΓΗ ΣΤΗΝ
ΕΠΙΣΤΗΜΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εισαγωγή

Στόχοι

1

- Να ορίσουμε τι είναι **υπολογιστικό σύστημα** και να απαριθμήσουμε τα **στρώματά** του.
- Να περιγράψουμε την έννοια της **αφαίρεσης** στη μελέτη των υπολογιστικών συστημάτων.
- Να κάνουμε μια ιστορική αναδρομή στην εξέλιξη του **υλικού** και του **λογισμικού** των υπολογιστών.
- Να περιγράψουμε πώς μεταβλήθηκε ο ρόλος του χρήστη των υπολογιστών.
- Να εξηγήσουμε τη διαφορά: **προγραμματιστές συστημάτων vs. προγραμματιστών εφαρμογών.**
- Να διαχωρίσουμε τον **υπολογισμό ως εργαλείο** από τον **υπολογισμό ως πεδίο μελέτης.**

Υπολογιστικά συστήματα

2

Υπολογιστικό σύστημα

- Ένα δυναμικό σύστημα.
- Χρησιμοποιείται στην επίλυση προβλημάτων.
- Αλληλεπιδρά με το περιβάλλον του:
 - Δέχεται δεδομένα εισόδου.
 - Παράγει δεδομένα εξόδου.
- Αποτελείται από **υλικό** και **λογισμικό**.

Υπολογιστικά συστήματα

3

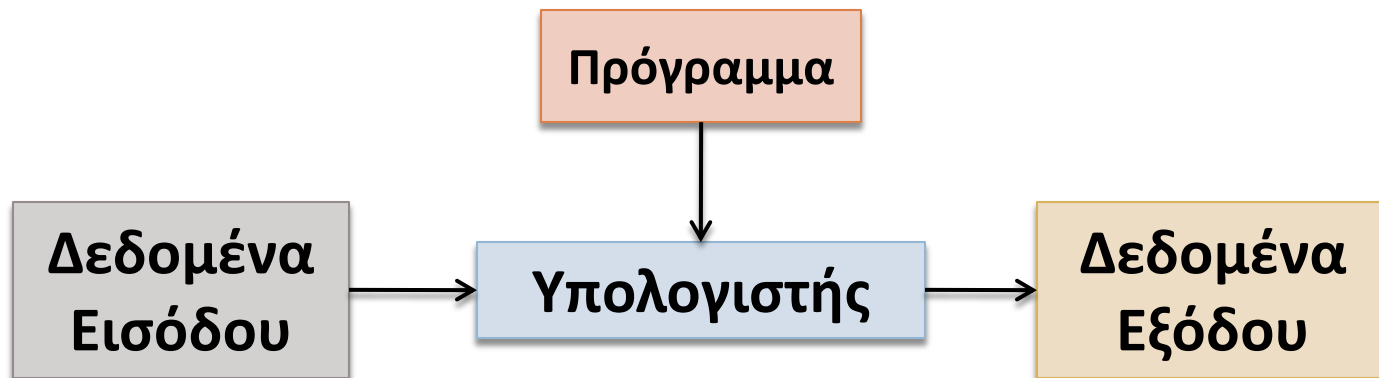
Επεξεργαστής Δεδομένων



Υπολογιστικά συστήματα

4

Προγραμματιζόμενος Επεξεργαστής Δεδομένων



Υπολογιστικά συστήματα

5

Υλικό:

Τα **ηλεκτρονικά, ηλεκτρικά, και μηχανικά μέρη** του υπολογιστικού συστήματος (π.χ., πληκτρολόγιο, ποντίκι, εκτυπωτής, καλώδια, πλακέτες κυκλωμάτων, επεξεργαστής, σκληρός δίσκος, κλπ.)

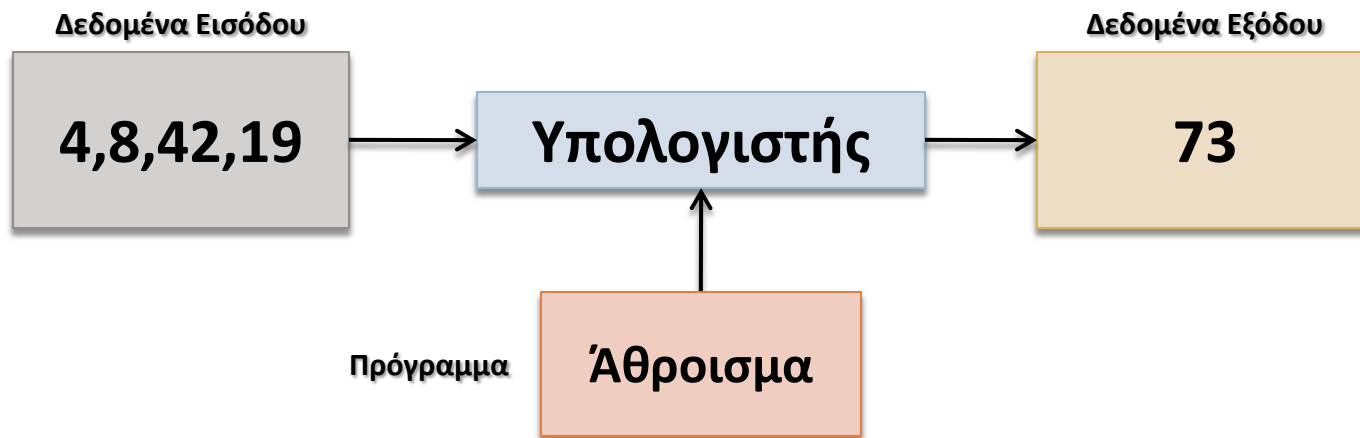
Λογισμικό:

Τα **προγράμματα** που εκτελεί ο υπολογιστής.

[**Πρόγραμμα:** Σειρά από **εντολές** (δηλ. οδηγίες) που πρέπει να ακολουθήσει το υπολογιστικό σύστημα για να εκτελέσει μια συγκεκριμένη εργασία]

Πρόγραμμα

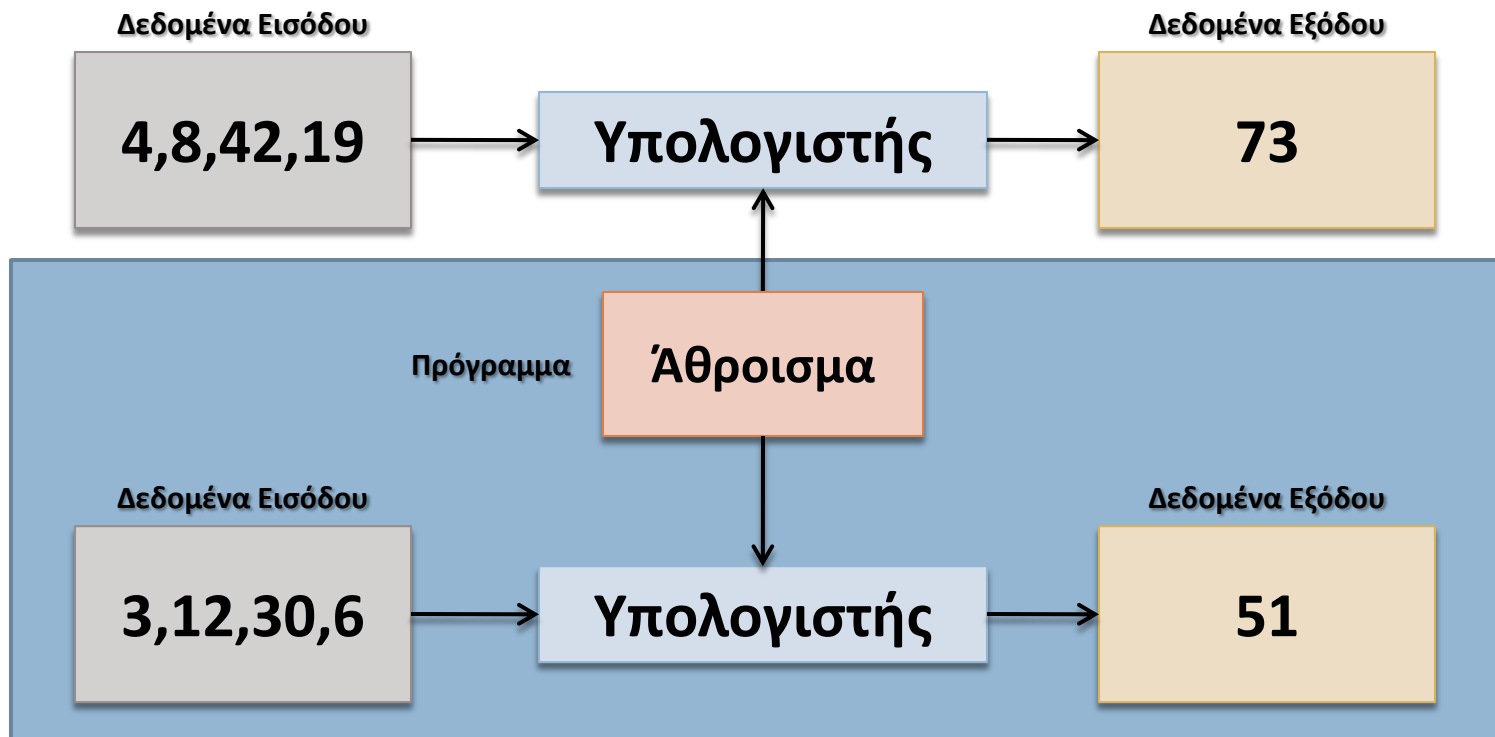
6



Πρόγραμμα

7

Ίδιο πρόγραμμα – διαφορετικά δεδομένα!!



Υπολογιστικά συστήματα: Στρώματα

8



Αφαίρεση

9

Βασική έννοια που χρησιμοποιούμε στην παρουσίαση των στρωμάτων: η διανοητική διεργασία κατά την οποία

- αγνοούμε τις περίπλοκες λεπτομέρειες και
- διατηρούμε μόνο όσες πληροφορίες είναι απολύτως απαραίτητες για να επιτύχουμε τον στόχο μας.

Αφαίρεση: ένα παράδειγμα

10

Δεν χρειάζεται να γνωρίζουμε πως ακριβώς δουλεύει η μηχανή ενός αυτοκινήτου για να το οδηγήσουμε!

Αρκεί να γνωρίζουμε τα βασικά τού πώς αλληλεπιδρούμε με αυτό: πώς λειτουργεί το τιμόνι, πως αλλάζουμε ταχύτητα, ποιο πετάλι επιβραδύνει, ποιο επιταχύνει, κτλ.

Ιστορική αναδρομή

11

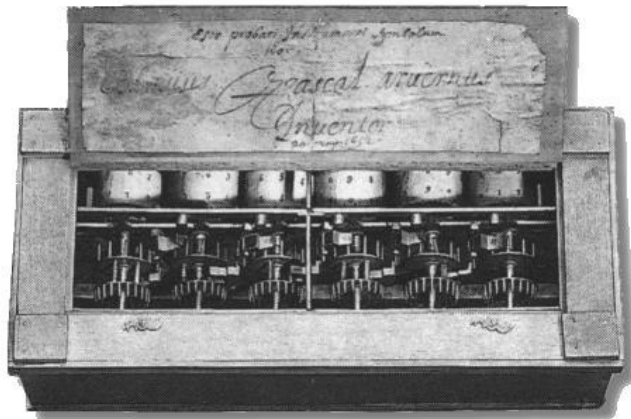
2500 π.Χ.: Εφευρίσκεται ο **άβακας**, η γνωστή συσκευή για την καταγραφή αριθμητικών τιμών (**αριθμητήριο**)



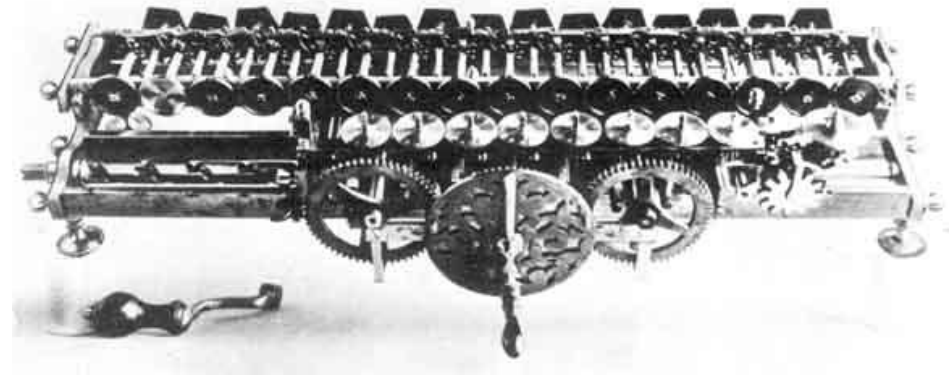
Ιστορική αναδρομή

12

- 1641 μ.Χ.: Ο **Pascal** εφευρίσκει την πρώτη μηχανή πρόσθεσης και αφαίρεσης (γνωστή ως Pascaline).
- 1673 μ.Χ.: Ο **Leibniz** εφευρίσκει μηχανή πρόσθεσης, αφαίρεσης, πολλαπλασιασμού, και διαίρεσης



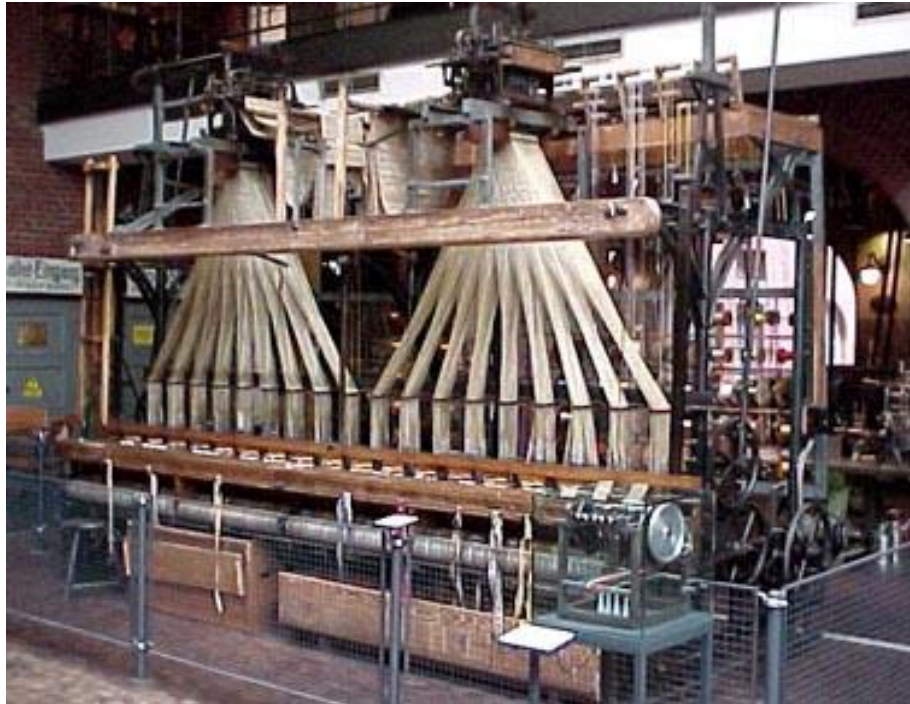
Pascaline



Ιστορική αναδρομή

13

Τέλη 18^{ου} αιώνα: ο **Jacquard** σχεδιάζει αργαλειό που χρησιμοποιεί διάτρητες κάρτες. Η πρώτη μηχανή που εφαρμόζει **αποθήκευση** και **προγραμματισμό**.

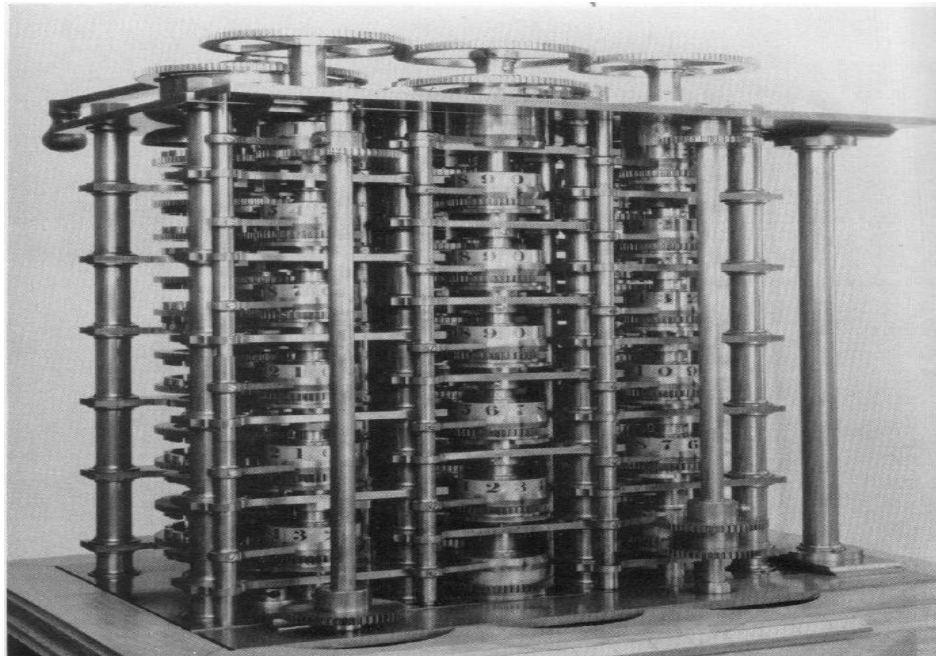


Ιστορική αναδρομή

14

1823: Ο **Babbage** σχεδιάζει τη **διαφορική μηχανή**, που λύνει πολυωνυμικές εξισώσεις.

Αργότερα: Σχεδιάζει την **αναλυτική μηχανή**, με πολλά χαρακτηριστικά σύγχρονων υπολογιστών: μνήμη, χειριστή, εισόδους-εξόδους, προγραμματισμό με διάτρητες κάρτες.



Ιστορική αναδρομή

19ος αιώνας: Η **Ada Augusta Byron** εκδίδει μελέτη με οδηγίες για το πώς η μηχανή του Babbage μπορεί να προγραμματιστεί για να εκτελέσει εργασίες πέραν των αριθμητικών πράξεων (σύνθεση μουσικής).

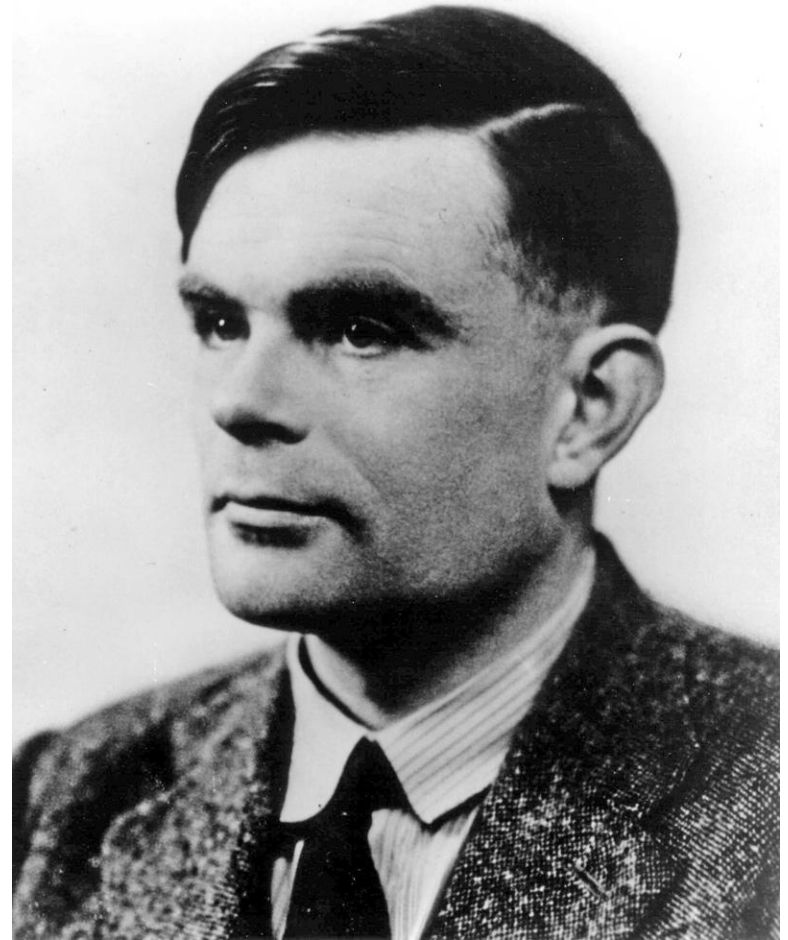
1890: Ο **Hollerith** κατασκευάζει προγραμματιζόμενη μηχανή που διαβάζει, καταμετρά, και ταξινομεί δεδομένα αποθηκευμένα σε διάτρητες κάρτες. Η μηχανή χρησιμοποιείται στην απογραφή πληθυσμού των Η.Π.Α. Η εργασία οδηγεί στην ίδρυση της **IBM**.

Ιστορική αναδρομή

16

1936: Ο **Alan Turing**

ανακοινώνει ένα θεωρητικό μοντέλο ικανό να αναπαραστήσει κάθε υπολογιστική συσκευή. Πρόκειται για την λεγόμενη **μηχανή Turing** που αποτελεί από τότε το βασικό μαθηματικό εργαλείο μελέτης της ισχύος των υπολογιστών.



Παρένθεση

17

- Τι είναι το Turing test?
 - ▣ Live example: <http://alice.pandorabots.com>
- Δοκιμή της δυνατότητας ενός μηχανήματος να επιδεικνύει ευφυή συμπεριφορά.

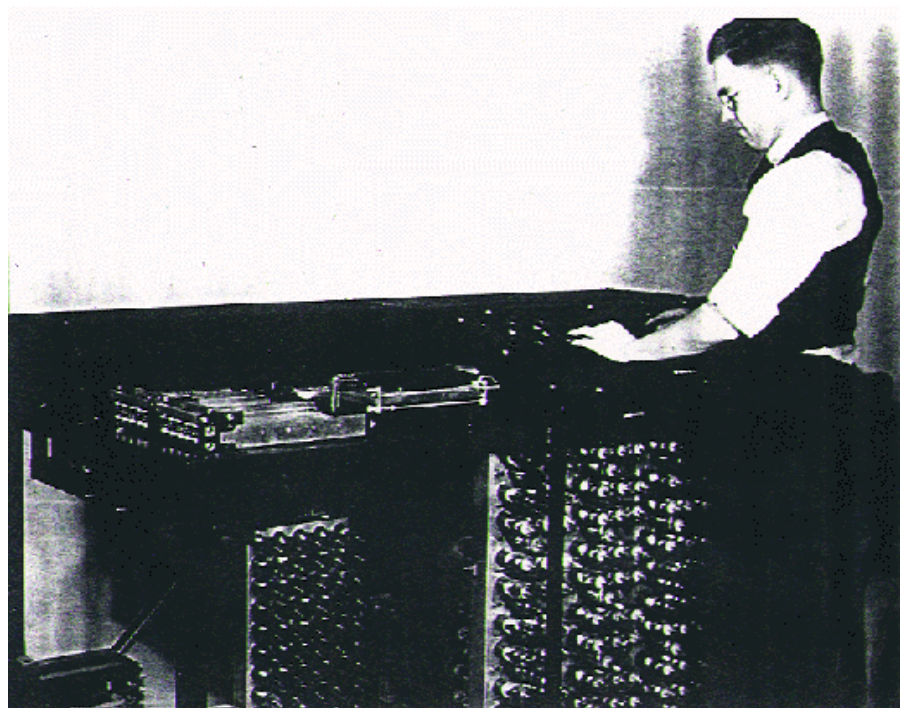
<http://xkcd.com/329>



Ιστορική αναδρομή

18

1939: Οι **Atanasoff** και **Berry** δημιουργούν στο Πανεπιστήμιο της Iowa τον **ABC**, τον πρώτο **ψηφιακό ηλεκτρονικό υπολογιστή ειδικής χρήσης** (επίλυση συστημάτων γραμμικών εξισώσεων).



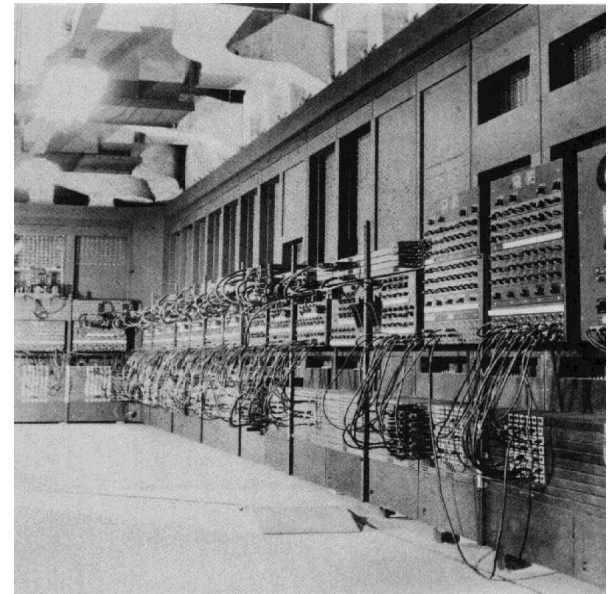
Ιστορική αναδρομή

19

*Β' Παγκόσμιος Πόλεμος: Ο **Turing** κατασκευάζει τον **ACE** (**A**utomatic **C**omputer **E**ngine) με δυνατότητα επεξεργασίας 25,000 χαρακτήρων/δευτερόλεπτο.*

*1946: Ολοκληρώνεται ο **ENIAC**, ο πρώτος **ηλεκτρονικός υπολογιστής γενικής χρήσης**.*

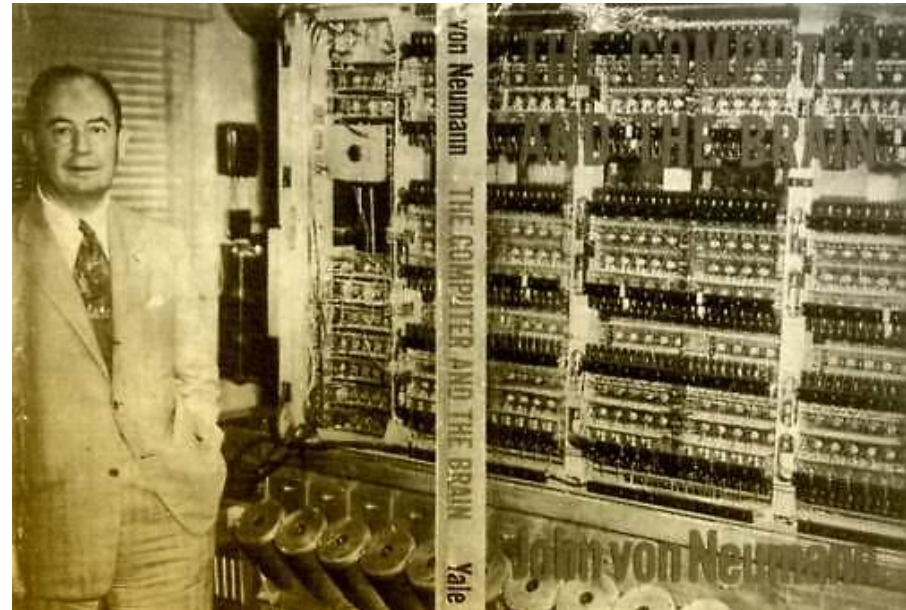
*30m μήκος
3m ύψος
30 τόνοι
18,000 λυχνίες*



Ιστορική αναδρομή

20

1950, Πανεπιστήμιο της Ρενσσυλβανία: Κατασκευάζεται ο **EDVAC**, ο πρώτος υπολογιστής βασισμένος στην **αρχιτεκτονική Eckert-von Neumann** (η μνήμη περιέχει όχι μόνο δεδομένα, αλλά και προγράμματα).



Παρένθεση

21

- Μοντέλο αρχιτεκτονικής υπολογιστή von Neumann
 - ▣ Ορίζει ότι το πρόγραμμα αποθηκεύεται στη μνήμη (όχι μόνο τα δεδομένα!)
 - ▣ Ακολουθιακή εκτέλεση εντολών
 - ▣ Έχει 4 υποσυστήματα
 - Μνήμη
 - Αριθμητική & λογική μονάδα (ΑΛΜ)
 - μονάδα ελέγχου
 - είσοδο/έξοδο

Ιστορική αναδρομή

22

1947: Ανακάλυψη του **τρανζίστορ**.

1959-1965: Εντυπωσιακή σμίκρυνση των Η/Υ και μείωση του κόστους τους.

1956-1975: Εμφάνιση **ολοκληρωμένων κυκλωμάτων**. Περαιτέρω μείωση κόστους και μεγέθους των Η/Υ.

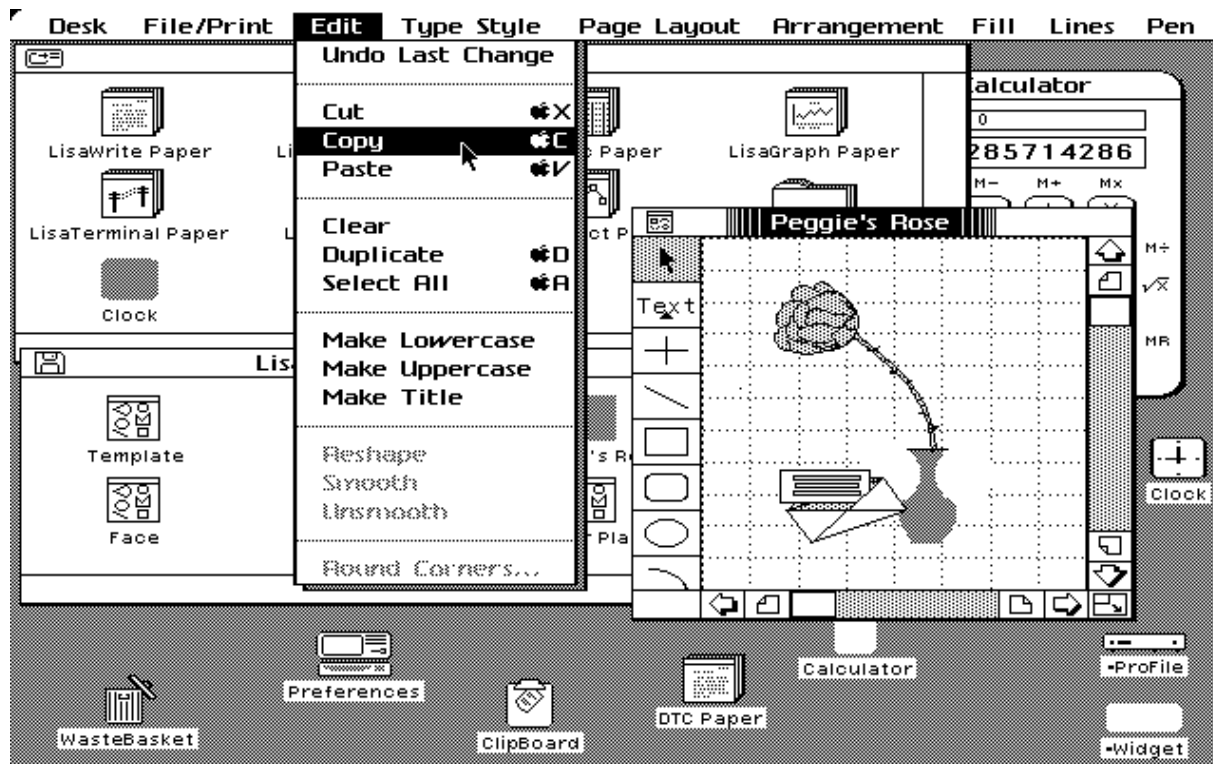
Δεκαετία '70: Εμφάνιση υπολογιστών για χομπίστες. Εμφάνιση υπερ-υπολογιστών. Ο **Seymour Cray** σχεδιάζει τον **CRAY-1**, που κατασκευάζεται το 1976.

Δεκαετία '80: Κατασκευή από την IBM του πρώτου προσωπικού υπολογιστή (PC, **P**ersonal **C**omputer).

Ιστορική αναδρομή

23

1983: Η **Apple** εκδίδει τον ηλεκτρονικό υπολογιστή **Lisa**, που περιλαμβάνει **γραφικό περιβάλλον** επικοινωνίας ανθρώπου –υπολογιστή.



Ιστορική αναδρομή

24

1989: Το Ευρωπαϊκό Κέντρο Πυρηνικών Ερευνών (**CERN**) ξεκινά την κατασκευή ενός πρωτοκόλλου για την ανταλλαγή εγγράφων μεταξύ υπολογιστών.

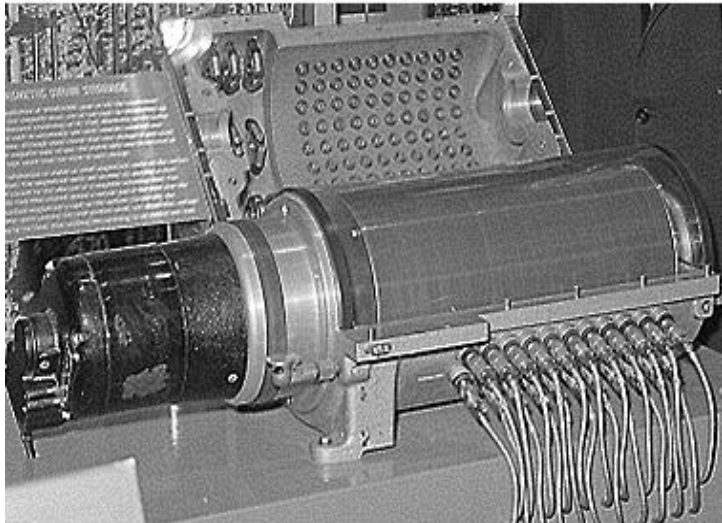
1989-1992: Εισάγεται το πρωτόκολλο **http**, που επιτρέπει την προσπέλαση εγγράφων με κείμενο και εικόνες από απομακρυσμένους υπολογιστές, ανεξαρτήτως κατασκευής και χρησιμοποιούμενου λογισμικού.

1994: **Παγκόσμιος Ιστός (World Wide Web)**

Υλικό 1ης γενιάς (1951-59)

25

Ηλεκτρονική λυχνία: Μεγάλο μέγεθος, μικρή αξιοπιστία, πολλή θερμότητα



Μαγνητικό τύμπανο: Συσκευή μνήμης που περιστρέφονταν κάτω από κεφαλή ανάγνωσης/εγγραφής

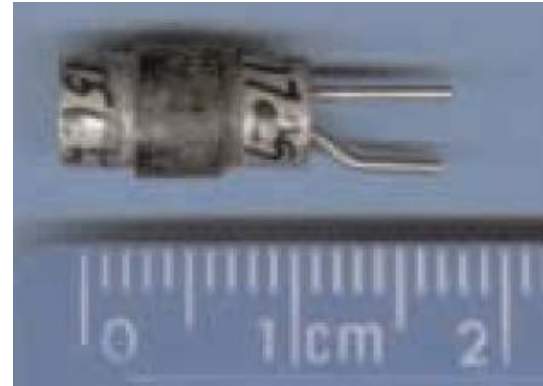
Μαγνητική ταινία: Σειριακή μνήμη



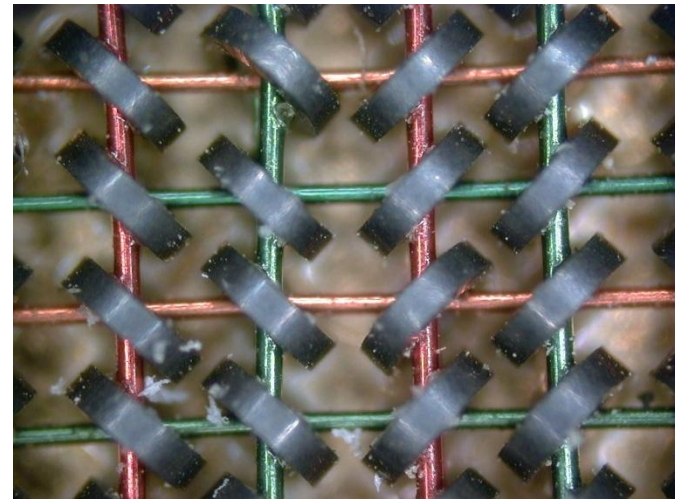
Υλικό 2ης γενιάς (1959-65)

26

Τρανζίστορ: Αντί λυχνίας.
Ταχύτερο, μικρότερο,
ανθεκτικότερο, φθηνότερο.
Σε πίνακες κυκλωμάτων.



Μαγνητικός πυρήνας:
Αντί μαγνητικού τυμπάνου.
Άμεση προσπέλαση δεδομένων.

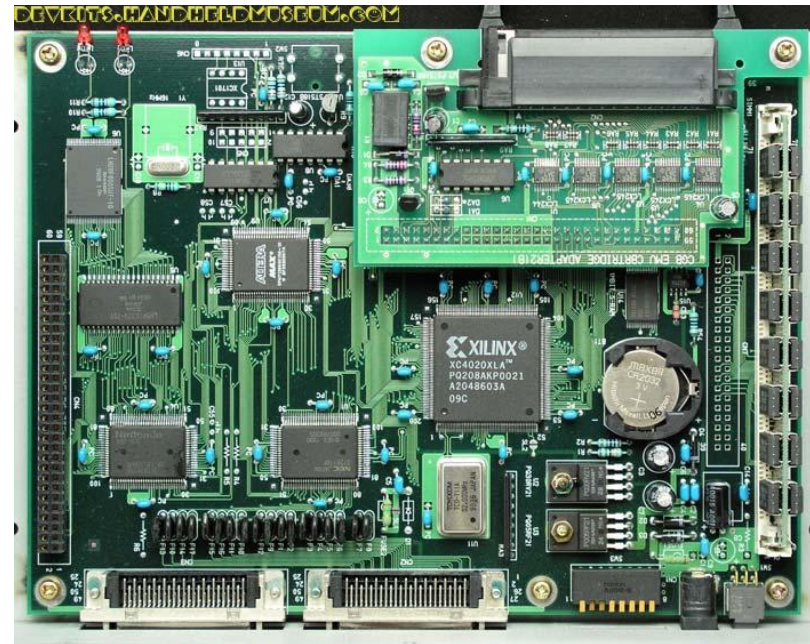


Μαγνητικός δίσκος:
Αντί μαγνητικής ταινίας.
Επίσης άμεση προσπέλαση δεδομένων.

Υλικό 3ης γενιάς (1965-71)

27

Ολοκληρωμένα κυκλώματα:
Αντί πινάκων κυκλωμάτων.
Συμπαγή κομμάτια σιλικόνης.
Μικρότερα, φθηνότερα,
ταχύτερα, πιο αξιόπιστα.

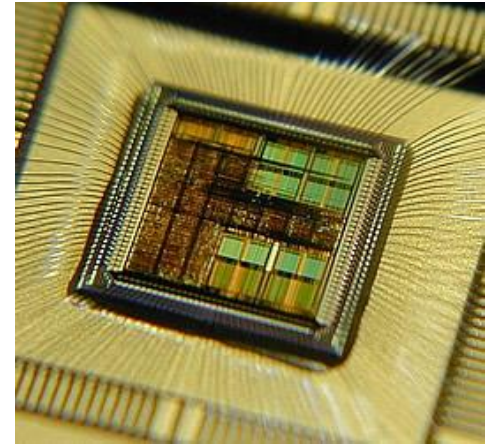


Τερματικά: Συσκευές
εισόδου/εξόδου με
πληκτρολόγιο και οθόνη.

Υλικό 4ης γενιάς (1971-σήμερα)

28

Κυκλώματα μεγάλης ολοκλήρωσης (Large-scale integration, LSI) και πολύ μεγάλης ολοκλήρωσης (very large-scale integration, VLSI):
Πρόοδος στη τεχνολογία του τσιπ (chip).



Προσωπικοί υπολογιστές (Personal Computers, PC), εμπορική αγορά:
Εταιρίες (π.χ., Apple, Atari, HP) κατασκευάζουν πλέον προσωπικούς υπολογιστές.



Νόμος του Moore

29

Κάθε 18 μήνες, η πυκνότητα των ολοκληρωμένων κυκλωμάτων διπλασιάζεται.

Gordon Moore (συνιδρυτής της εταιρείας Intel).

Παράλληλη επεξεργασία/δικτύωση

30

Παράλληλος υπολογισμός:

Πολλαπλοί επεξεργαστές αλληλεπιδρούν και συνεργάζονται για ταχύτερη επεξεργασία δεδομένων.

Δικτύωση:

Το **ethernet** επιτρέπει σε μικρούς υπολογιστές να συνδεθούν μεταξύ τους και να ανταλλάξουν δεδομένα.

→ **δίκτυα υπολογιστών**

ARPANET + Τοπικά δίκτυα υπολογιστών → Διαδίκτυο

(Advanced Research Projects Agency Network)

Λογισμικό 1ης γενιάς (1951-59)

31

Γλώσσα μηχανής:

Τα προγράμματα γράφονται στη δυαδική γλώσσα, όπου τα πάντα εκφράζονται ως ακολουθίες από 0 και 1.

Γλώσσα *assembly* και μεταγλωττιστής *assembly*:

Τα προγράμματα γράφονται σε **τεχνητή γλώσσα** ειδικών κωδικών. Έπειτα μεταγλωττίζονται σε **γλώσσα μηχανής**.

Οι προγραμματιστές αρχίζουν να διαχωρίζονται σε

- **προγραμματιστές συστημάτων**, και
- **προγραμματιστές εφαρμογών**.

Assembly

32

□ Motorola 6800

```
C000          ORG      ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START  LDS      #STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A

0013          RESETA EQU    %00010011
0011          CTLREG EQU    %00010001

C003 86 13          INITA  LDA A  #RESETA  RESET ACIA
C005 B7 80 04          STA A  ACIA
C008 86 11          LDA A  #CTLREG  SET 8 BITS AND 2 STOP
C00A B7 80 04          STA A  ACIA

C00D 7E C0 F1          JMP    SIGNON  GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04  INCH   LDA A  ACIA      GET STATUS
C013 47          ASR A          SHIFT RDRF FLAG INTO CARRY
C014 24 FA          BCC  INCH     RECIEVE NOT READY
C016 B6 80 05          LDA A  ACIA+1  GET CHAR
C019 84 7F          AND A  #$7F     MASK PARITY
C01B 7E C0 79          JMP    OUTCH   ECHO & RTS

*****
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: Digit in acc A
* CALLS: INCH
* DESTROYS: acc A
* Returns to monitor if not HEX input

C01E 8D F0          INHEX  BSR    INCH     GET A CHAR
C020 81 30          CMP A  #'0     ZERO
C022 2B 11          BMI    HEXERR  NOT HEX
C024 81 39          CMP A  #'9     NINE
C026 2F 0A          BLE    HEXRTS  GOOD HEX
C028 81 41          CMP A  #'A
C02A 2B 09          BMI    HEXERR  NOT HEX
C02C 81 46          CMP A  #'F
C02E 2E 05          BGT    HEXERR
C030 80 07          SUB A  #7     FIX A-F
C032 84 0F          HEXRTS AND A  #$0F   CONVERT ASCII TO DIGIT
C034 39          RTS

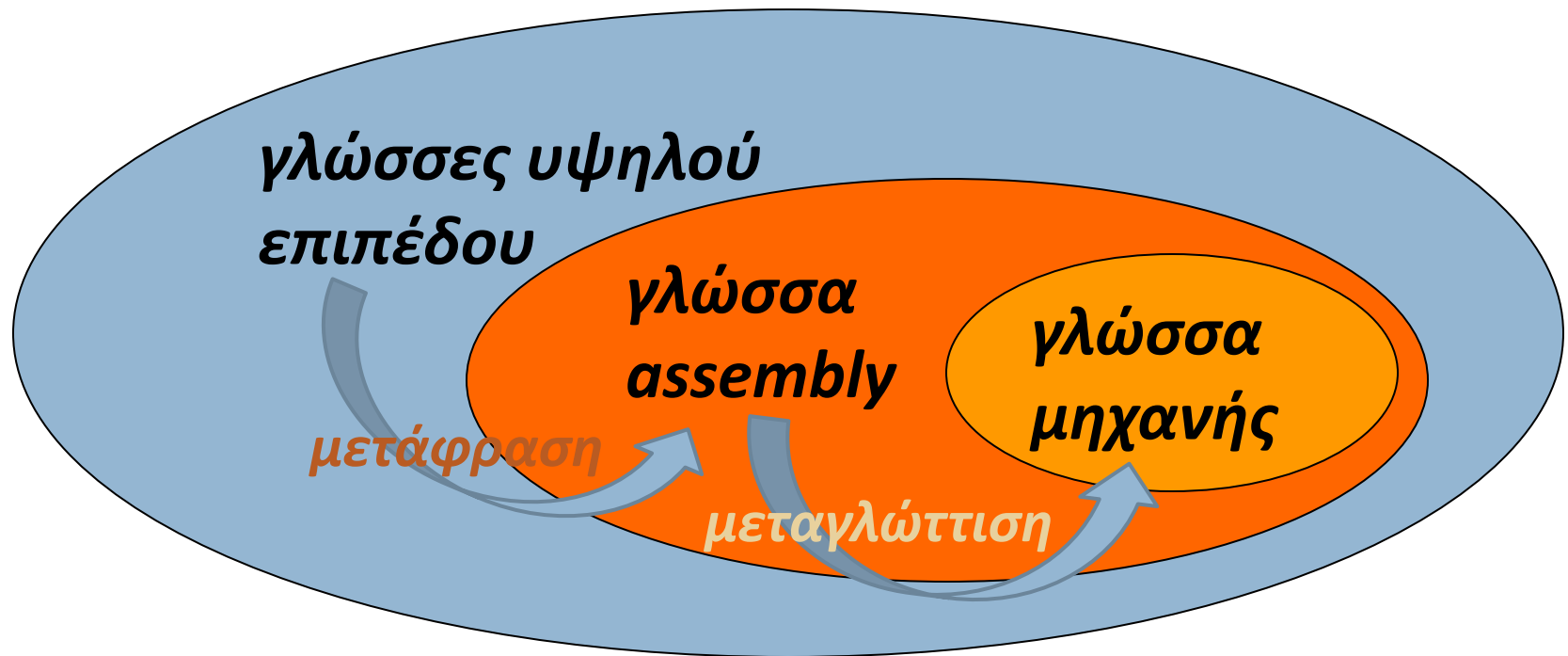
C035 7E C0 AF  HEXERR  JMP    CTRL    RETURN TO CONTROL LOOP
```

Λογισμικό 2ης γενιάς (1959-65)

33

Γλώσσες υψηλού επιπέδου και μεταφραστές:

Τα προγράμματα γράφονται σε γλώσσα (π.χ., FORTRAN, COBOL, LISP) που χρησιμοποιεί αγγλικές λέξεις/φράσεις. Έπειτα μεταφράζεται σε γλώσσα μηχανής.



Λογισμικό 3ης γενιάς (1965-71)

➤ Ακόμη μεγαλύτερη διαφοροποίηση ανάμεσα σε:

Λογισμικό συστημάτων: Λογισμικό βασικών χρήσεων.

- *Λειτουργικά συστήματα* (αναλαμβάνουν τις πολύ βασικές λειτουργίες, π.χ. ποια προγράμματα θα εκτελεστούν και πότε, διαχείριση πολλών χρηστών).
- Μεταφραστές, μεταγλωττιστές, κλπ.

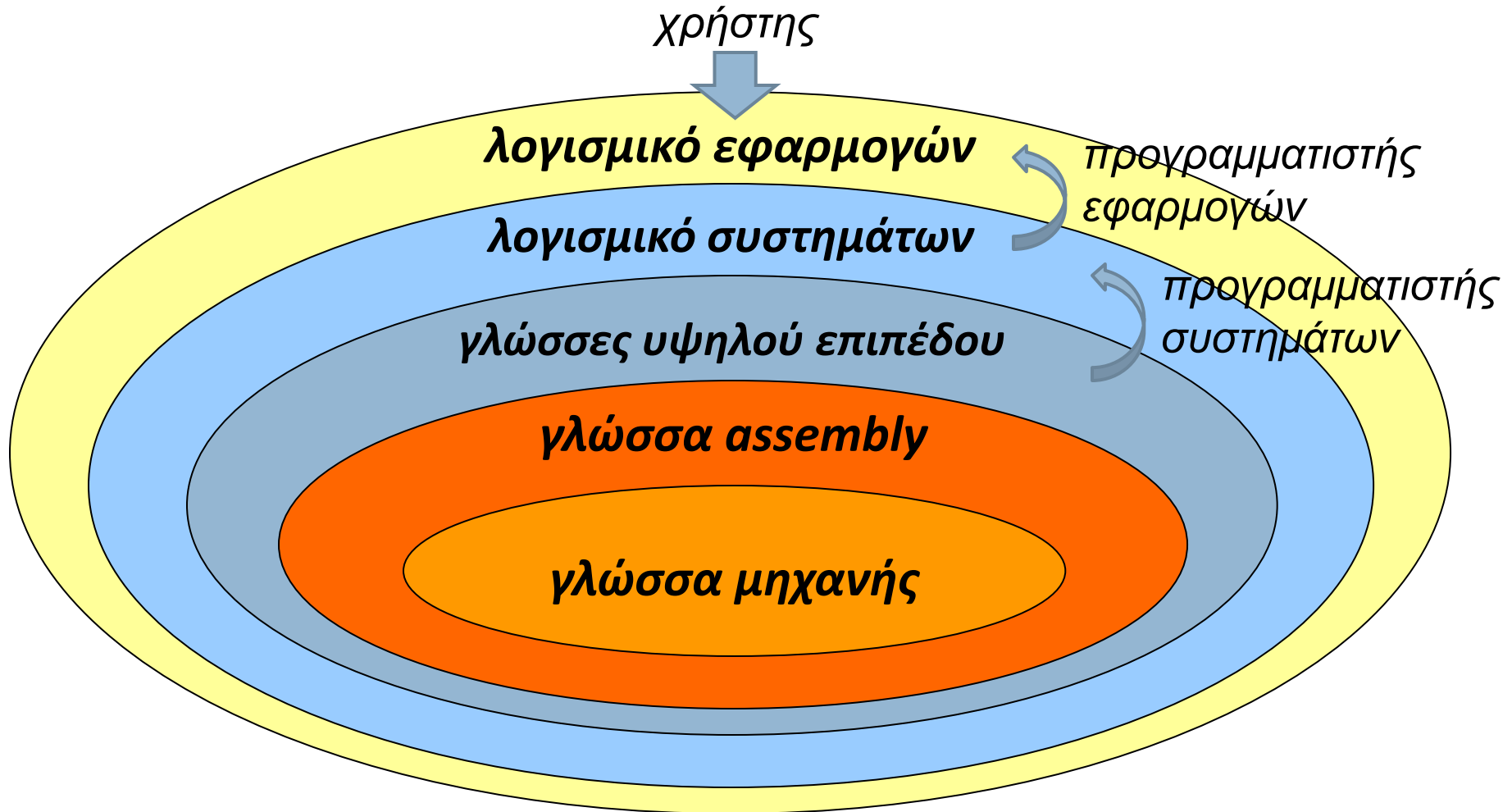
Λογισμικό εφαρμογών: Λογισμικό άλλων χρήσεων.

Οι προγραμματιστές αρχίζουν να γράφουν προγράμματα που θα χρησιμοποιηθούν από μη προγραμματιστές.

➤ Απομάκρυνση του χρήστη από το υλικό.

Λογισμικό 3ης γενιάς (1965-71)

35



Λογισμικό 4ης γενιάς (1971-89)

36

Δομημένος προγραμματισμός:

Γλώσσες που τον υποστηρίζουν: Pascal, C, QuickBASIC,...

Ισχυρότερα λειτουργικά συστήματα:

UNIX, MS-DOS, γραφικό περιβάλλον στα Macintosh

Νέο λογισμικό εφαρμογών:

- Λογιστικά φύλλα (Lotus)
- Επεξεργαστές κειμένου (Word Perfect)
- Συστήματα διαχείρισης βάσεων δεδομένων (dBASE)

Λογισμικό 5ης γενιάς (1990-σήμερα)

37

Microsoft: Το λειτουργικό σύστημα Windows και άλλα προγράμματα εφαρμογών της εταιρίας Microsoft (π.χ., MS Excel, MS Word, MS Access) κυριαρχούν στην αγορά.

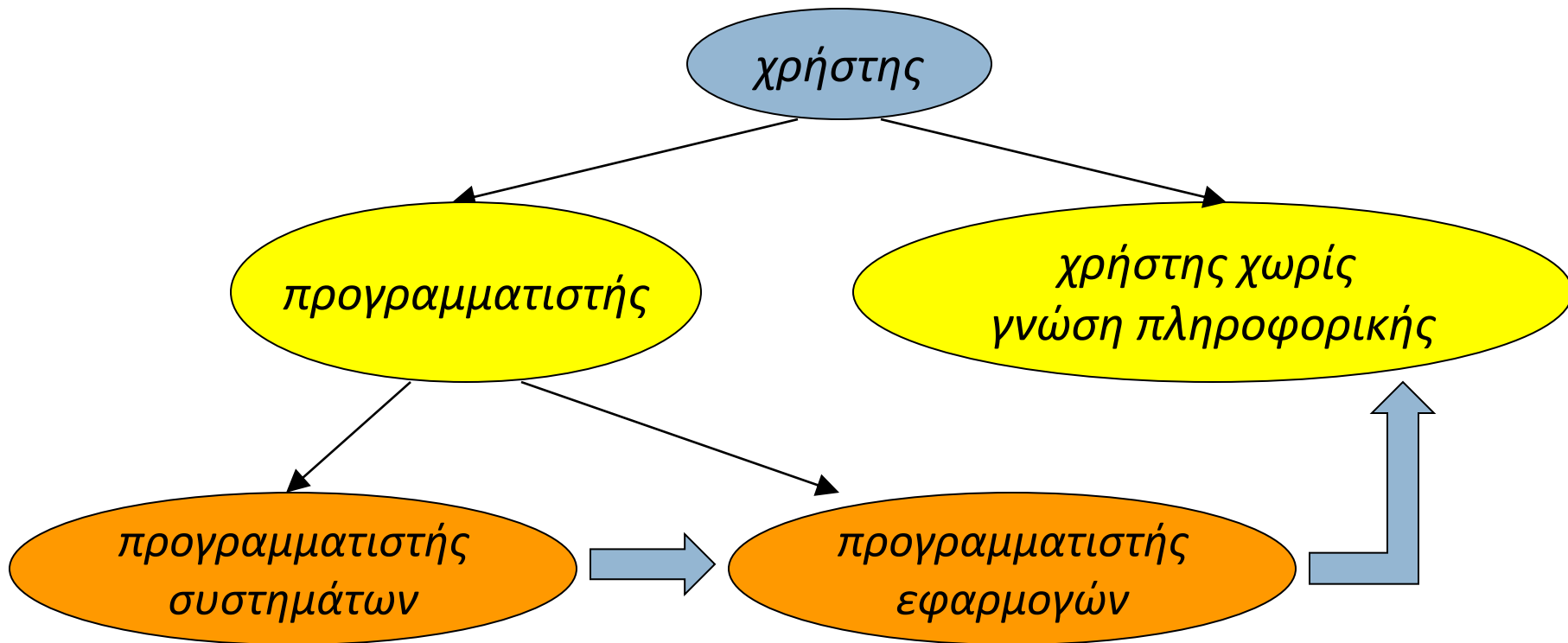
Αντικειμενοστρεφής σχεδιασμός: Βασίζεται σε ιεραρχία αντικειμένων. Γλώσσες που τον υποστηρίζουν: C++, Java.

Παγκόσμιος Ιστός (World Wide Web): Διευκολύνει την παγκόσμια επικοινωνία μέσω του Διαδικτύου.

Οι υπολογιστές χρησιμοποιούνται άνετα και από άτομα που δεν γνωρίζουν πληροφορική.

Οι υπολογιστές ως εργαλείο

38



Οι υπολογιστές ως πεδίο μελέτης

39

Ποιες εργασίες μπορούν να εκτελεστούν (γρήγορα) στον υπολογιστή;

Τέσσερα απαραίτητα προσόντα ενός «πληροφορικού»:

- 1. Αλγοριθμική σκέψη.***
- 2. Γνώση μεθόδων αναπαράστασης δεδομένων.***
- 3. Προγραμματιστική ικανότητα.***
- 4. Σχεδιαστική ικανότητα.***

Τι πιστεύετε: Η πληροφορική ανήκει στα μαθηματικά, στις φυσικές επιστήμες, ή στην τεχνολογία;

Οι υπολογιστές ως πεδίο μελέτης

40

A. Περιοχές της πληροφορικής **σχετικές με συστήματα:**

- *Αλγόριθμοι & δομές δεδομένων*
- *Γλώσσες προγραμματισμού*
- *Αρχιτεκτονική υπολογιστών*
- *Λειτουργικά συστήματα*
- *Τεχνολογία & μεθοδολογία λογισμικού*
- *Επικοινωνία ανθρώπου-υπολογιστή*

Οι υπολογιστές ως πεδίο μελέτης

41

B. Περιοχές της πληροφορικής **σχετικές με εφαρμογές:**

- *Αριθμητικός & συμβολικός υπολογισμός*
- *Βάσεις δεδομένων & ανάκτηση πληροφοριών*
- *Τεχνητή νοημοσύνη & ρομποτική*
- *Γραφικά υπολογιστών*
- *Βιοπληροφορική*

Συζήτηση

43

- Αν ήταν δυνατό, τι θα προτιμούσατε να έχει ένας υπολογιστής, απεριόριστη μνήμη ή απεριόριστη υπολογιστική ισχύ?
- Θέλουμε υπολογιστές με ευφυΐα?
- Είπαν
 - ▣ “I think there is a world market for maybe five computers.” --- Thomas Watson, πρόεδρος της IBM, 1943.
 - ▣ “There is no reason anyone would want a computer in their home.” ---Ken Olson, πρόεδρος της DEC, 1977.

Ερωτήσεις

44

