



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Εισαγωγή στον δομημένο προγραμματισμό

Ενότητα 1^η: Εισαγωγή

Αν. καθηγητής Στεργίου Κώστας
e-mail: kstergiou@uowm.gr

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Στόχοι της διάλεξης

- Γενική περιγραφή του μαθήματος, του τρόπου λειτουργίας των εργαστηρίων, και του τρόπου βαθμολόγησης.
- Εισαγωγή στην έννοια του προγράμματος και του προγραμματισμού.
- Εισαγωγή στην έννοια του αλγορίθμου και στους βασικούς τρόπους περιγραφής αλγορίθμων.
- Επισκόπηση της ιστορίας των γλωσσών προγραμματισμού.



Περιεχόμενα

- Προγράμματα και προγραμματισμός.
- Εντολές σε γλώσσες προγραμματισμού.
- Αλγόριθμοι:
 - Ψευδοκώδικας.
 - Διάγραμμα ροής.
- Μεταγλώττιση.
- Γλώσσες προγραμματισμού:
 - Ιστορία και εξέλιξη.



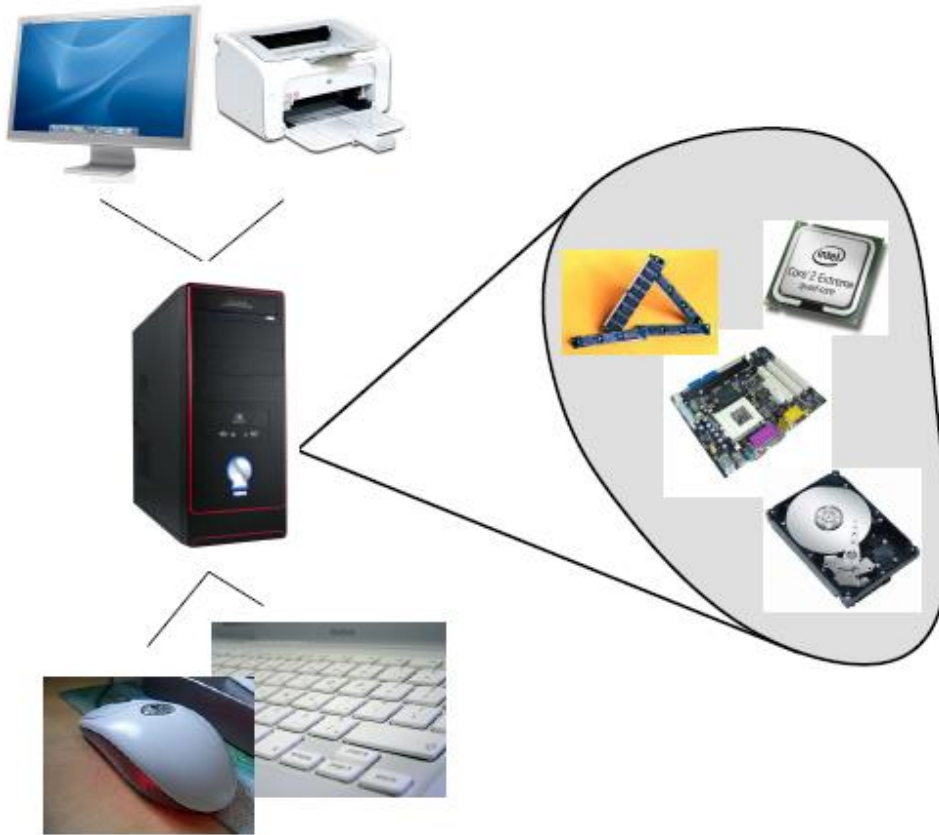
Τι είναι ο Η/Υ;

- Ένας ηλεκτρονικός υπολογιστής (Η/Υ) είναι μια συσκευή που μπορεί να εκτελέσει υπολογισμούς και να πάρει λογικές αποφάσεις με ταχύτητες δισεκατομμύρια φορές μεγαλύτερες από τους ανθρώπους:
 - Αλλά για να ξέρει τι να κάνει ο Η/Υ, πρέπει κάποιος να τον καθοδηγήσει, να του δώσει οδηγίες.

Στο μάθημα αυτό θα μάθουμε να δίνουμε οδηγίες σε έναν Η/Υ!

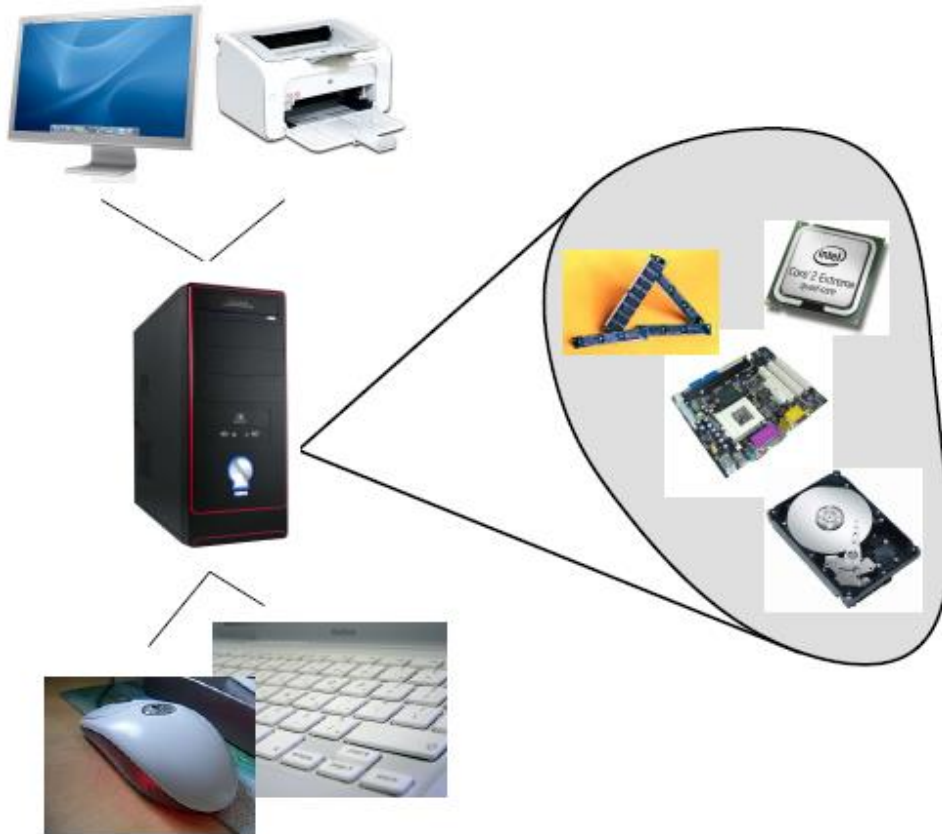


Η ανατομία ενός Η/Υ (1/3)



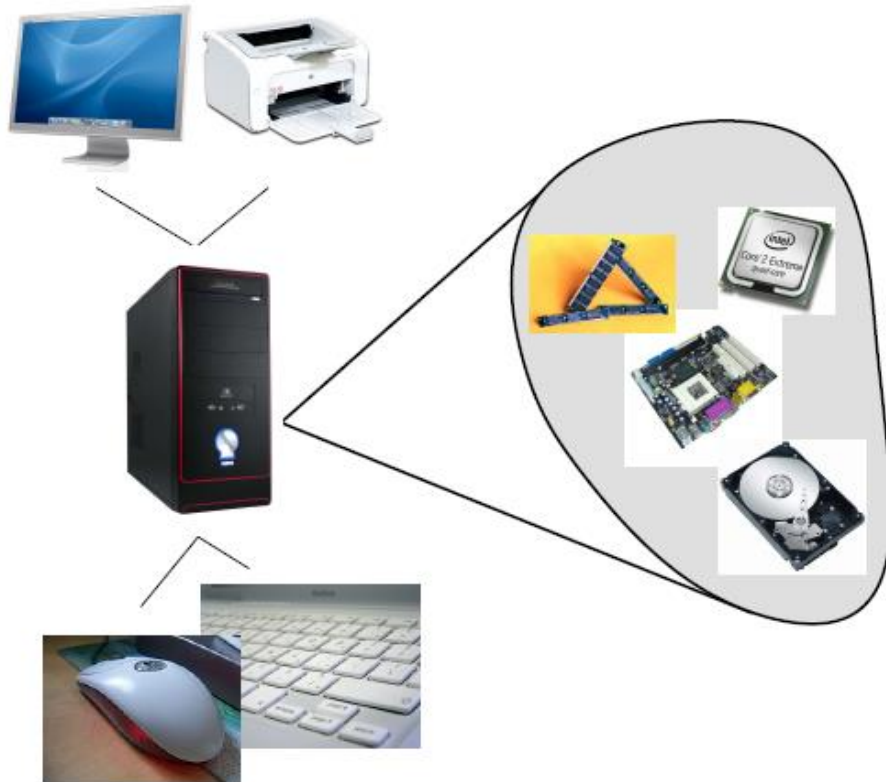
**Συσκευές
Εισόδου:**
πληκτρολόγιο,
ποντίκι, οθόνη
αφής, κ.τ.λ

Η ανατομία ενός Η/Υ (2/3)



**Συσκευές
Εξόδου:** οθόνη,
εκτυπωτής, κ.τ.λ

Η ανατομία ενός Η/Υ (3/3)



μονάδα υπολογιστή:
κεντρική μονάδα επεξεργασίας (CPU), μονάδα μνήμης, μονάδα δευτερεύουσας αποθήκευσης (πχ σκληρός δίσκος), αριθμητική και λογική μονάδα (συνήθως μέσα στη CPU), δίαυλος επικοινωνίας (bus) για επικοινωνία των διαφόρων μονάδων, κ.τ.λ

Λογισμικό (Software)

Ο υπολογιστής χρειάζεται και οδηγίες για το τι να κάνει. Αυτό το ρόλο παίζουν τα διάφορα προγράμματα:

- **BIOS:** πρόγραμμα που βοηθάει τον υπολογιστή να ξεκινήσει
- **λειτουργικό σύστημα:** πρόγραμμα που παίρνει τον έλεγχο μετά την εκκίνηση και διαχειρίζεται όλους τους πόρους του υπολογιστή, τις συσκευές εισόδου και εξόδου και γενικά φροντίζει για την εκτέλεση άλλων προγραμμάτων (πχ Windows, Linux, SunOS, MacOSX, κ.τ.λ)
- **εφαρμογές γενικής χρήσης:** προγράμματα φτιαγμένα με συγκεκριμένο σκοπό, πχ αναπαραγωγή ταινιών, επεξεργασία εικόνας, κειμενογράφοι, παιχνίδια, κ.τ.λ



Προγραμματισμός

- **Πρόγραμμα** ενός Η/Υ είναι ένα σύνολο **εντολών** που καθοδηγούν τον υπολογιστή να εκτελέσει καθορισμένες ενέργειες.
- **Προγραμματισμός** ενός Η/Υ είναι ο σαφής καθορισμός μιας διαδικασίας ως συνόλου εντολών (το πρόγραμμα) που περιγράφει λεπτομερώς τα βήματα που πρέπει να εκτελεστούν για να λυθεί ένα πρόβλημα.
- Κάθε πρόβλημα αναλύεται σε τέσσερις θεμελιώδεις λειτουργίες:
 - Λειτουργίες εισόδου/εξόδου πληροφοριών.
 - Αριθμητικές λειτουργίες.
 - Κίνηση πληροφοριών μέσα στη CPU.
 - Λογικές ή συγκριτικές λειτουργίες.



Προγραμματισμός στη γλώσσα C

- Το αντικείμενο του μαθήματος.
- Στόχος είναι στο τέλος του εξαμήνου:
 - Να γνωρίζετε τις αρχές προγραμματισμού με τη γλώσσα C.
 - Να μπορείτε να κατασκευάσετε απλά (και πιο προχωρημένα) προγράμματα στη γλώσσα C.
 - Να είστε σε θέση να μάθετε με σχετική ευκολία άλλες γλώσσες προγραμματισμού (π.χ. C++, Java).
 - Να γνωρίζετε βασικές αρχές τεχνολογίας λογισμικού.



Διδάσκοντες

Θεωρία και Εργαστήριο:

Στεργίου Κώστας (kstergiou@uowm.gr)



Βαθμολόγηση Μαθήματος

- Για να περάσει κάποιος το μάθημα πρέπει να βαθμολογηθεί με τουλάχιστον 50% τόσο στη γραπτή εξέταση του μαθήματος **όσο και** στο εργαστήριο.
- Ο τελικός βαθμός υπολογίζεται ως:
Τελικός Βαθμός = 0.7*βαθμός γραπτού + 0.3*ΒΕ
- Στο εργαστήριο θα δοθούν 7 εργασίες και θα γίνει τελική εξέταση:
 - Η κάθε εργασία θα μετράει 0.2 μονάδες (εκτός από μια που θα μετράει 0.3). Η εξέταση του εργαστηρίου θα μετράει 1.5 μονάδα.



Εργαστήριο (1/2)

- Όλες οι εργασίες (εκτός από την πρώτη) θα υλοποιούνται από ομάδες των 2 ατόμων.
 - Η πρώτη εργασία θα είναι ατομική.
- Δεδομένου ότι τα εργαστήρια Α και Β έχουν χωρητικότητα 25 περίπου ατόμων, θα γίνει διαχωρισμός σε 4-5 ομάδες.

Αυστηρά αλφαβητικά.

Περισσότερα στο επόμενο μάθημα.



Εργαστήριο (2/2)

- Κάθε δεύτερη εβδομάδα θα δίνεται μια εργασία που θα αποτελείται από δύο ασκήσεις.
- Η μια από τις ασκήσεις πρέπει να υλοποιείται κατά τη διάρκεια του συγκεκριμένου εργαστηρίου.
- Προθεσμία υλοποίησης της άλλης άσκησης: **2 εβδομάδες**,
 - μέχρι το μεθεπόμενο εργαστήριο.
- Η κάθε εργασία πρέπει να αποστέλλεται μέσω του e-class στον διδάσκοντα του εργαστηρίου (kstergiou@uowm.gr) αυστηρά μέχρι την προθεσμία.
- **Η παρακολούθηση των εργαστηρίων είναι υποχρεωτική!**
 - Αυτό ισχύει μόνο για τους πρωτοετείς φοιτητές:
 - Αλλά όσοι φοιτητές μεγαλύτερων ετών δεν έχουν παραδώσει εργασίες στο παρελθόν ή δεν έχουν περάσει το εργαστήριο, πρέπει να παραδώσουν εργασίες.



Διαφάνειες και λοιπό υλικό

- Οι διαφάνειες των διαλέξεων, οι εκφωνήσεις των εργαστηριακών ασκήσεων, και άλλο υλικό θα βρίσκονται στον χώρο του μαθήματος στην πλατφόρμα UOWM Open eClass (<http://eclass.uowm.gr/>).
- Λογαριασμό για το e-class θα λάβετε από το helpdesk (helpdesk@noc.uowm.gr).
- Ίδια με το email σας.

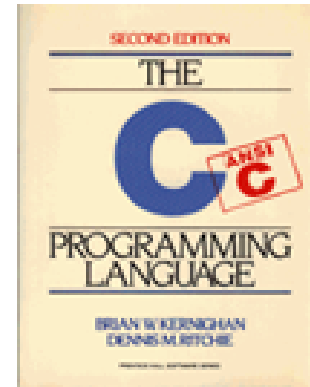


Βιβλία

- Η γλώσσα C σε βάθος
Νίκος Χατζηγιαννάκης



- Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C
Kernighan, Ritchie
- C ΓΙΑ ΑΡΧΑΡΙΟΥΣ
Σεφερίδης
- Υπάρχουν και πολλά άλλα



Αλγόριθμος

- Πριν δημιουργήσουμε ένα πρόγραμμα για να λύσουμε ένα πρόβλημα πρέπει να σχεδιάσουμε έναν **αλγόριθμο** για αυτό το πρόβλημα.
- Μια **σαφώς καθορισμένη** ακολουθία εντολών ή κανόνων που οδηγεί σε ένα προκαθορισμένο αποτέλεσμα (λύση ενός προβλήματος) μέσω ενός **πεπερασμένου** πλήθους βημάτων.
- Ένα πρόγραμμα είναι η διατύπωση ενός αλγορίθμου σε μια συγκεκριμένη γλώσσα προγραμματισμού.
 - Το πρόγραμμα είναι η **υλοποίηση** του αλγορίθμου.



Παράδειγμα

- Μέγιστος Κοινός Διαιρέτης

Βήμα 1. ΕΣΤΩ $z = \text{ΥΠΟΛΟΙΠΟ}(x/y)$
Βήμα 2. ΕΑΝ $z = 0$, ΤΟΤΕ
Βήμα 2.1 ΜΚΔ = y . ΤΕΡΜΑΤΙΣΜΟΣ
Βήμα 3. ΑΛΛΙΩΣ ΕΠΑΝΑΛΗΨΗ ΑΠΟ
Βήμα 1 ΜΕ ($x = y$ ΚΑΙ $y = z$)

Αλγόριθμος

```
mkd = 0;
while (mkd == 0) {
    z = x % y;
    if (z == 0) {
        mkd = y;
        return mkd;
    }
    x = y;
    y = z;
}
```

Πρόγραμμα

Εντολές



Εντολές

Εντολή είναι μια φράση της γλώσσας προγραμματισμού που αναγκάζει τον Η/Υ να **εκτελέσει** μια συγκεκριμένη λειτουργία.

- Μια αριθμητική πράξη, έναν έλεγχο, εκτύπωση στην οθόνη, κτλ.

Η διαδικασία εκτέλεσης μιας εντολής περιλαμβάνει τα εξής πέντε βήματα:

Βήμα 1. Ανάκληση επόμενης εντολής από τη μνήμη.

Βήμα 2. Ερμηνεία εντολής και ανάκληση δεδομένων.

Βήμα 3. Εκτέλεση εντολής.

Βήμα 4. Μεταφορά του αποτελέσματος της εκτέλεσης στη μνήμη.

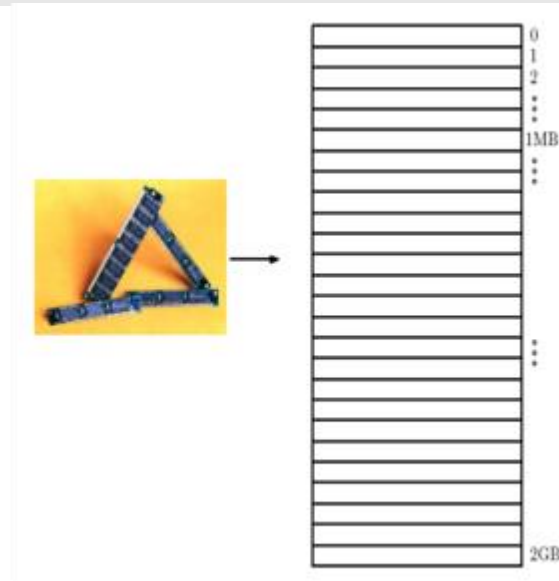
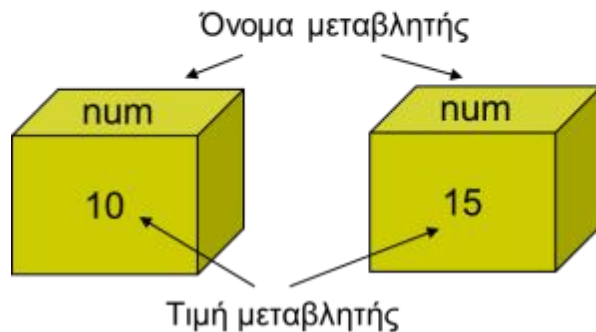
Βήμα 5. Εάν υπάρχουν και άλλες εντολές στο πρόγραμμα, επαναλαμβάνεται το Βήμα 1.

```
mkd = 0;
while (mkd == 0) {
    z = x % y;
    if (z == 0) {
        mkd = y;
        return mkd;
    }
    x = y;
    y = z;
}
```



Μεταβλητές

- Τα δεδομένα, οι ενδιάμεσες πράξεις, και τα αποτελέσματα ενός προγράμματος αποθηκεύονται σε θέσεις στην κεντρική μνήμη (RAM) του Η/Υ.
- Για να χρησιμοποιηθεί μια θέση μνήμης από μια γλώσσα προγραμματισμού, πρέπει να της δοθεί ένα όνομα:
 - Το περιεχόμενο της θέσης μνήμης είναι η τιμή της.
 - Με κατάλληλες εντολές μπορούμε να **αλλάζουμε** την τιμή.
- Κάθε όνομα θέσης μνήμης καλείται **μεταβλητή** (variable) σε μια γλώσσα προγραμματισμού.



Αρχές Ανάπτυξης Προγραμμάτων (1/2)

- **Αλγόριθμος:**

- Ο προγραμματιστής πρέπει να διατυπώσει τη λύση του προβλήματος σε γενικές γραμμές.

- **Ψευδοκώδικας / Διάγραμμα ροής:**

- Η περιγραφή γίνεται χρησιμοποιώντας ψευδοκώδικα ή διαγράμματα ροής.
- Έλεγχος της λογικής της προτεινόμενης λύσης χρησιμοποιώντας κατάλληλα δεδομένα.

- **Κωδικοποίηση σε γλώσσα υψηλού επιπέδου:**

- Η ακολουθία των λειτουργιών του ψευδοκώδικα ή του διαγράμματος ροής κωδικοποιείται σε γλώσσα προγραμματισμού.

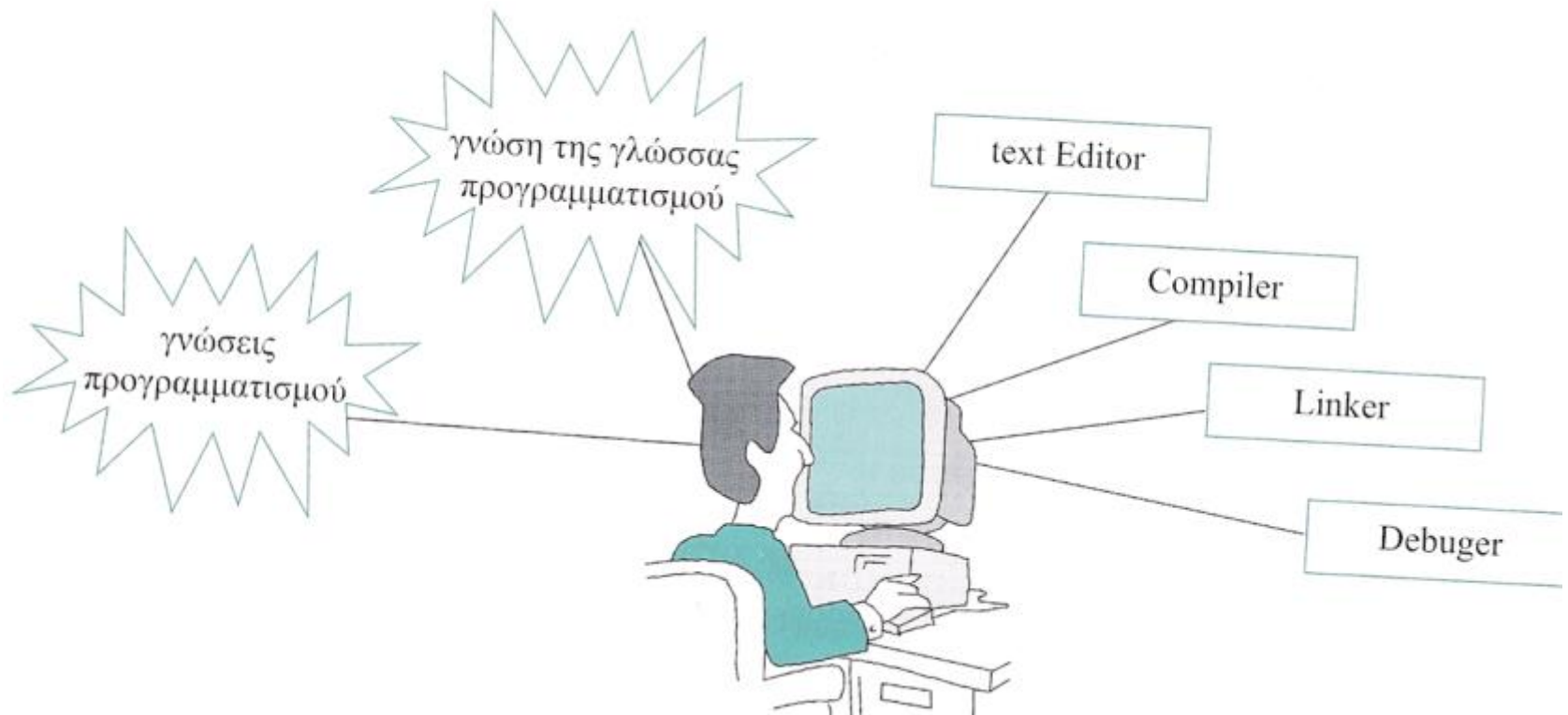


Αρχές Ανάπτυξης Προγραμμάτων (2/2)

- **Μεταγλώττιση – Σύνδεση:**
 - Ο μεταγλωττιστής ελέγχει το πηγαίο πρόγραμμα για γραμματικά λάθη (ορθογραφία εντολών) και συντακτικά λάθη (εντολές που λείπουν).
 - Το πηγαίο πρόγραμμα μετατρέπεται σε γλώσσα μηχανής.
- **Διορθώσεις:**
 - Τα γραμματικά και συντακτικά λάθη διορθώνονται από τον προγραμματιστή και επαναλαμβάνεται η μεταγλώττιση.
 - Διαδοχική απομάκρυνση των σφαλμάτων (debugging).
- **Δοκιμή:**
 - Λάθη λογικής που παραμένουν στο πρόγραμμα μπορούν να βρεθούν εξετάζοντας την έξοδο.
 - Εκτός από συνηθισμένα δεδομένα, δίνονται σκόπιμα λάθη για να διαπιστωθεί αν θα εντοπιστούν.



Υλοποίηση Προγραμμάτων



Περιγραφή Αλγορίθμων

Ως εργαλείο επικοινωνίας, ένας αλγόριθμος πρέπει:

- να έχει συγκεκριμένη δομή,
- να εκφράζεται σε μορφή κατανοητή από όλους,
- να εκφράζεται με καθορισμένη σύνταξη ώστε να μην είναι διφορούμενη η ερμηνεία του.

Διακρίνουμε δύο μεγάλες κατηγορίες εργαλείων περιγραφής αλγορίθμων:

- εργαλεία που βασίζονται στη γλώσσα (**ψευδοκώδικας**),
- εργαλεία που βασίζονται σε σχήματα (**διαγράμματα ροής**).



Ψευδοκώδικας

- Ο ψευδοκώδικας είναι κάτι που μοιάζει με κώδικα προγραμματισμού, αλλά δεν είναι.
- Ο ψευδοκώδικας είναι μια δομημένη γλώσσα που χρησιμοποιεί:
 - στοιχεία (συντακτικές δομές) από τις γλώσσες προγραμματισμού,
 - συμβολισμούς (σημασιολογικά στοιχεία) από τα Μαθηματικά και τη Μαθηματική Λογική,
 - στοιχεία (λεκτικές περιγραφές) από τη φυσική γλώσσα.



Χαρακτηριστικά ψευδοκώδικα

- Ο ψευδοκώδικας χρησιμοποιείται για την ανάπτυξη ενός εργαλείου περιγραφής αλγορίθμων που θα συνδυάζει ταυτόχρονα την αυστηρότητα και την ακρίβεια που έχουν οι γλώσσες προγραμματισμού με την εκφραστική δύναμη της φυσικής γλώσσας.
- Ο ψευδοκώδικας είναι ανεξάρτητος από οποιαδήποτε γλώσσα προγραμματισμού.
- Ένας αλγόριθμος που περιγράφεται με ψευδοκώδικα μπορεί άμεσα να «μεταφραστεί» σε ένα πρόγραμμα.
- Γίνεται κατανοητός τόσο από προγραμματιστές όσο και από μη προγραμματιστές.



Δομές Ελέγχου

Μια απλή δομημένη γλώσσα μπορεί να περιέχει τρεις δομές ελέγχου:

- ακολουθία,
- απόφαση,
- επανάληψη.



Γλώσσα Ψευδοκώδικα (1/2)

Αρχή ομάδας οδηγιών	ΑΡΧΗ
Τέλος ομάδας οδηγιών	ΤΕΛΟΣ
Καταχώρηση σε μεταβλητή	:=
Διαχωριστής εντολών	;
Είσοδος δεδομένων από συσκευή (προαιρετικά)	ΔΙΑΒΑΣΕ (λίστα ονομάτων μεταβλητών εισόδου) [ΑΠΟ πηγή δεδομένων]
Εξοδος δεδομένων στην οθόνη	ΤΥΠΩΣΕ (μεταβλητές εξόδου, “κείμενο”)
Εξοδος δεδομένων στον εκτυπωτή	ΕΚΤΥΠΩΣΕ (μεταβλητές, “κείμενο”)



Γλώσσα Ψευδοκώδικα (2/2)

Εξοδος δεδομένων σε άλλη συσκευή	ΓΡΑΨΕ (μεταβλητές, “κείμενο”) ΣΕ συσκευή
Απόφαση	ΕΑΝ (συνθήκη) ΤΟΤΕ οδηγία ή ομάδα οδηγιών [ΑΛΛΙΩΣ οδηγία ή ομάδα οδηγιών] ΕΑΝ-ΤΕΛΟΣ
Επανάληψη με μετρητή / δείκτη	ΓΙΑ μετρητής = αρχική τιμή ΕΩΣ τελική τιμή ΕΠΑΝΕΛΑΒΕ οδηγία ή ομάδα οδηγιών ΓΙΑ-ΤΕΛΟΣ
Επανάληψη με συνθήκη στην αρχή	ΕΝΟΣΩ (συνθήκη) ΕΠΑΝΕΛΑΒΕ οδηγία ή ομάδα οδηγιών ΕΝΟΣΩ-ΤΕΛΟΣ
Επανάληψη με συνθήκη στο τέλος	ΕΠΑΝΕΛΑΒΕ οδηγία ή ομάδα οδηγιών ΜΕΧΡΙ (συνθήκη)
Κλήση υποπρογράμματος με μεταβλητές εισόδου (προαιρετικά)	ΥΠΟΛΟΓΙΣΕ όνομα υποπρογράμματος [(λίστα μεταβλητών)]



Παράδειγμα (1/3)

ΑΛΓΟΡΙΘΜΟΣ MAX-XY

ΔΕΔΟΜΕΝΑ

X, Y, MAX: REAL;

ΑΡΧΗ

ΔΙΑΒΑΣΕ(X, Y);

ΕΑΝ (X > Y) ΤΟΤΕ

MAX:=X

ΑΛΛΙΩΣ

MAX:=Y

ΕΑΝ-ΤΕΛΟΣ;

ΤΥΠΩΣΕ(MAX)

ΤΕΛΟΣ



Παράδειγμα (2/3)

ΑΛΓΟΡΙΘΜΟΣ MAX-XYZ

ΔΕΔΟΜΕΝΑ

X,Y,Z: REAL;

ΑΡΧΗ

ΔΙΑΒΑΣΕ(X,Y,Z);

ΕΑΝ (X > Y) ΤΟΤΕ

ΕΑΝ (X > Z) ΤΟΤΕ

ΤΥΠΩΣΕ(X)

ΑΛΛΙΩΣ

ΤΥΠΩΣΕ(Z)

ΕΑΝ-ΤΕΛΟΣ

ΑΛΛΙΩΣ

ΕΑΝ (Y > Z) ΤΟΤΕ

ΤΥΠΩΣΕ(Y)

ΑΛΛΙΩΣ

ΤΥΠΩΣΕ(Z)

ΕΑΝ-ΤΕΛΟΣ

ΕΑΝ-ΤΕΛΟΣ

ΤΕΛΟΣ



Παράδειγμα (3/3)

ΑΛΓΟΡΙΘΜΟΣ ΜΚΔ

ΔΕΔΟΜΕΝΑ

X, Y, Z, MKD: INTEGER;

ΑΡΧΗ

ΔΙΑΒΑΣΕ (X,Y) ;

Z := X MOD Y ;

ΕΑΝ (Z <> 0) ΤΟΤΕ

 X := Y ;

 Y := Z

ΑΛΛΙΩΣ

 MKD := Y

ΕΑΝ-ΤΕΛΟΣ

ΤΕΛΟΣ



Διαγράμματα Ροής (Flow Charts)

- Επιδεικνύουν την ακολουθία των βημάτων που περιγράφει ο αλγόριθμος με ένα διαγραμματικό τρόπο.
- Δείχνει την γενική ακολουθία των γεγονότων με την οποία μπορεί να λυθεί το πρόβλημα.
- Δίνουν μια πολύ καλή προσέγγιση για το σχεδιασμό προγραμμάτων.
 - Απευθύνονται στους προγραμματιστές.
- Δεν θεωρούνται ο καλύτερος τρόπος για την τεκμηρίωση των προγραμμάτων.
 - Δεν απευθύνονται στους χρήστες.



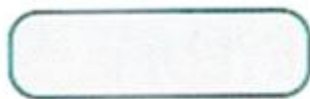
Διαγράμματα Ροής

- Χρησιμοποιούνται διάφορα γεωμετρικά σχήματα με ορισμένη σημασία.
- Τα σχήματα συνδέονται με συνεχείς γραμμές και χρησιμοποιούνται τόξα για να δείξουν την ροή των γεγονότων.
- Ένα διάγραμμα ροής δεν είναι ένα συμπληρωμένο πρόγραμμα, αλλά αποτελεί μια βοήθεια για τον προγραμματιστή.



Σχήματα Διαγραμμάτων Ροής

Τα βασικά σχήματα είναι τα παρακάτω



Αρχή ή Τέλος



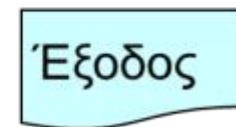
Εκτέλεση ενέργειας



Απόφαση (απάντηση: ναι / όχι)



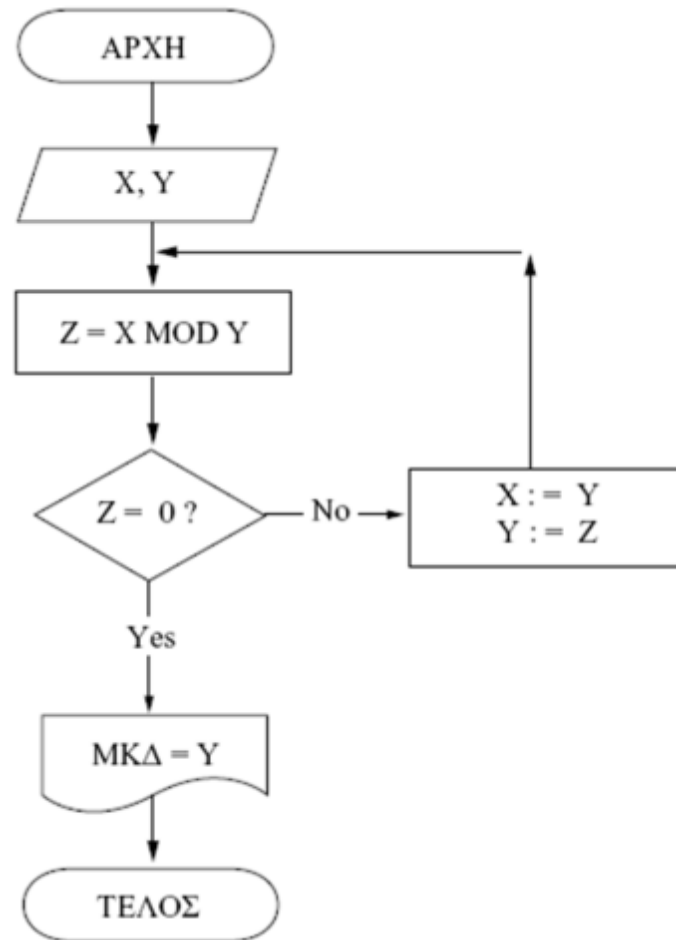
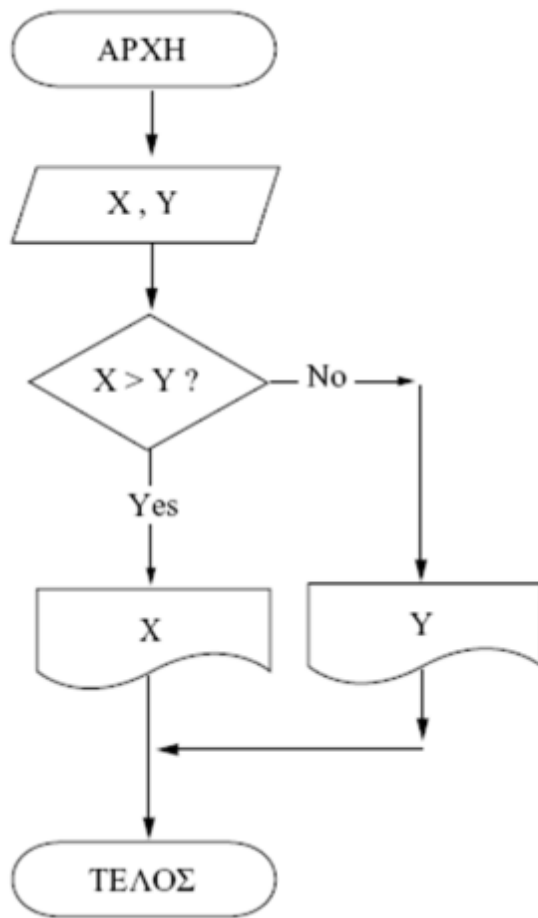
Είσοδος – έξοδος



Συνδετικό σύμβολο



Παραδείγματα

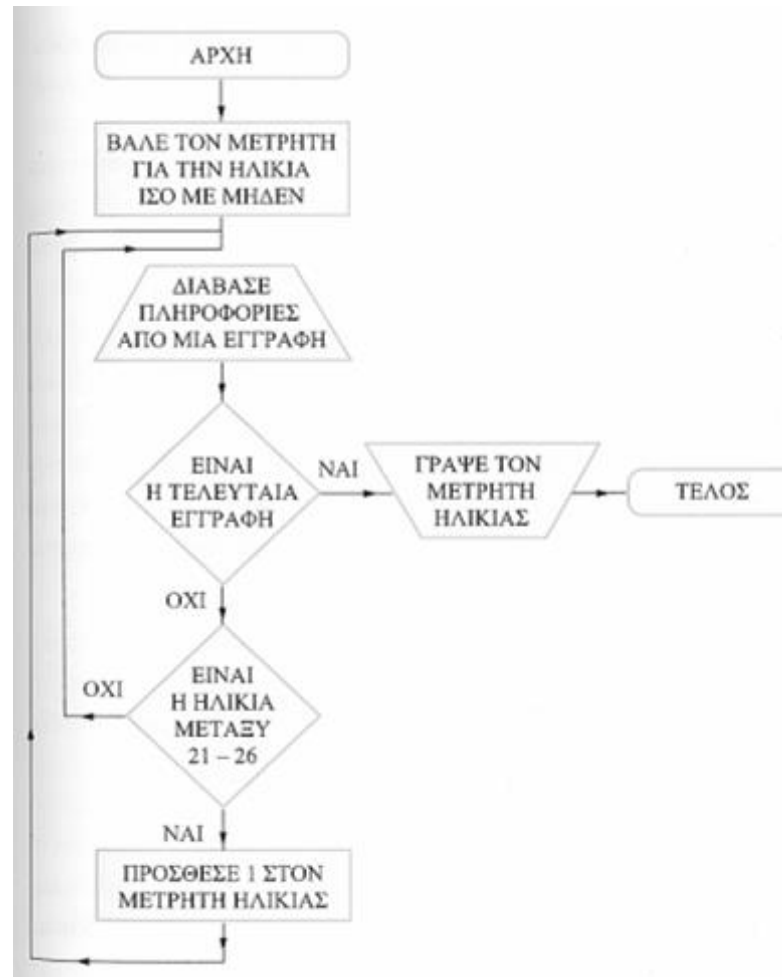


Παράδειγμα (Α)

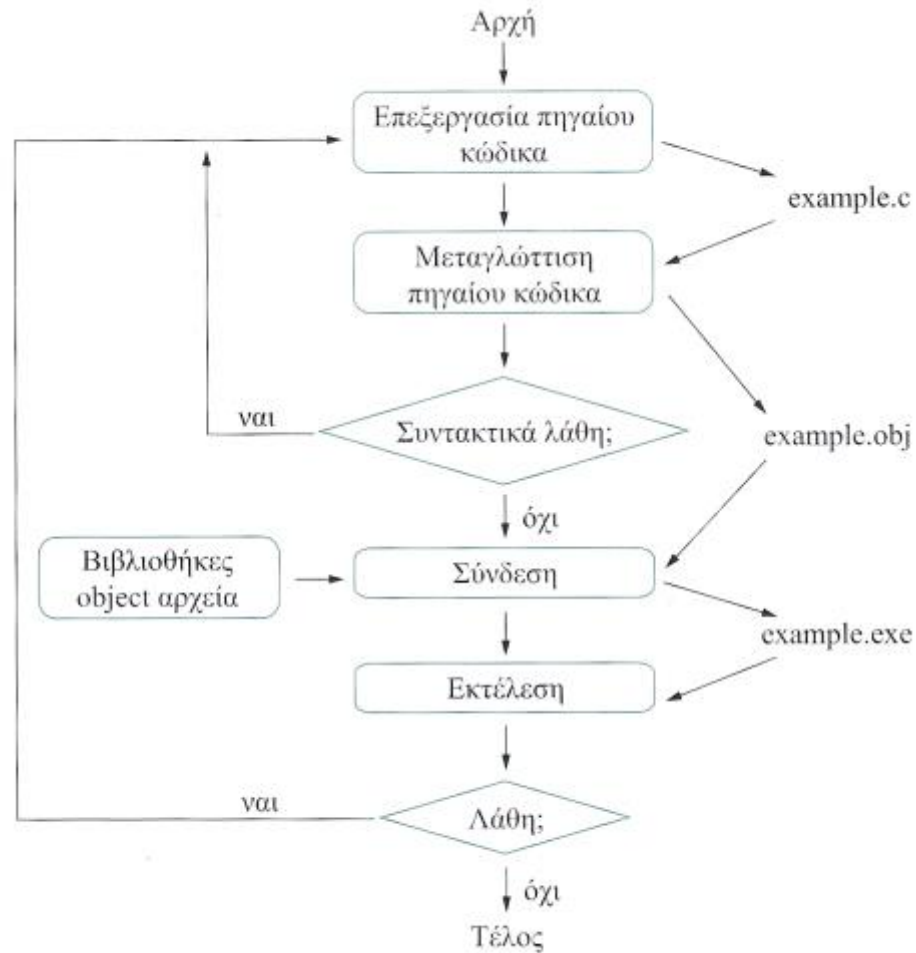
- Σε ένα αρχείο έχουμε αποθηκευμένες πληροφορίες για τους κατοίκους μιας περιοχής.
 - Όνομα, διεύθυνση, φύλο, ηλικία, επάγγελμα κλπ.
- Το ζητούμενο πρόβλημα είναι να βρούμε τον αριθμό των ατόμων με ηλικία μεταξύ 21 και 26 χρονών.



Παράδειγμα (B)



Παράδειγμα (Γ)



Γλώσσες

- Φυσικές γλώσσες (Ελληνικά, Αγγλικά):
 - Επικοινωνία μεταξύ ανθρώπων.
 - Λεξιλόγιο, γραμματική, συντακτικό.
 - Αμφισημία σε έννοιες λέξεων, σύνταξη, ύφος κτλ.
- Γλώσσες προγραμματισμού:
 - Επικοινωνία μεταξύ ανθρώπου και Η/Υ.
 - Καλά ορισμένες έννοιες και σύνταξη.



Τι καταλαβαίνει ένας Η/Υ;

Ένας υπολογιστής καταλαβαίνει από 0 και από 1

Αλλά ένας άνθρωπος δε μπορεί εύκολα να μιλάει στον υπολογιστή με 0 και 1

πχ

000000 00001 00010 00110 00000 100000

σε μία αρχιτεκτονική λέει στον υπολογιστή να προσθέσει τους καταχωρητές 1 και 2 και να βάλει το αποτέλεσμα τον καταχωρητή 6.



Γλώσσες Προγραμματισμού

- Γλώσσες μηχανής:
 - Άμεση επικοινωνία με το υλικό του υπολογιστή.
- Συμβολικές γλώσσες:
 - Χρήση συμβολικών ονομάτων για τις βασικές εντολές.
- Υψηλού επιπέδου:
 - Προσανατολισμένες στο πρόβλημα και όχι στον υπολογιστή.



Γλώσσες Μηχανής (Machine Language)

- Η μόνη γλώσσα που «καταλαβαίνει» άμεσα ο υπολογιστής.
- Κάθε υπολογιστής έχει τη δική του διαφορετική γλώσσα μηχανής.
- Κάθε κωδικός της γλώσσας μηχανής συμβολίζει μία συγκεκριμένη λειτουργία.
 - π.χ. Πολλαπλασιασμός = 1001
- Ο προγραμματισμός σε γλώσσα μηχανής είναι εξαιρετικά επίπονη διαδικασία.
- Ξένη προς τον τρόπο που σκέφτεται ο άνθρωπος.
- Μεγάλη ταχύτητα εκτέλεσης των προγραμμάτων.



Συμβολικές Γλώσσες (Assembly)

- Για την καλύτερη απομνημόνευση των κωδικών μηχανής χρησιμοποιούνται μνημονικοί κωδικοί (συντομεύσεις των Αγγλικών).
 - π.χ. Πολλαπλασιασμός = MULT = 1001.
 - π.χ. ADD32 6,1,2.
- Η μετάφραση από τα συμβολικά ονόματα στους αριθμητικούς κωδικούς της γλώσσας μηχανής γίνεται μέσω του συμβολομεταφραστή (assembler).



Γλώσσες Υψηλού Επιπέδου

- Δεν εξαρτώνται από την μηχανή αλλά από το πρόβλημα.
- Επιτρέπουν στον προγραμματιστή να γράψει εντολές χρησιμοποιώντας κωδικές λέξεις και συμβατικούς μαθηματικούς συμβολισμούς.
- Οι εντολές μοιάζουν σχεδόν όπως τα Αγγλικά και περιέχουν τα συνηθισμένα μαθηματικά σύμβολα:
 - π.χ. **if A then B.**
 - π.χ. $D = A * B + C.$

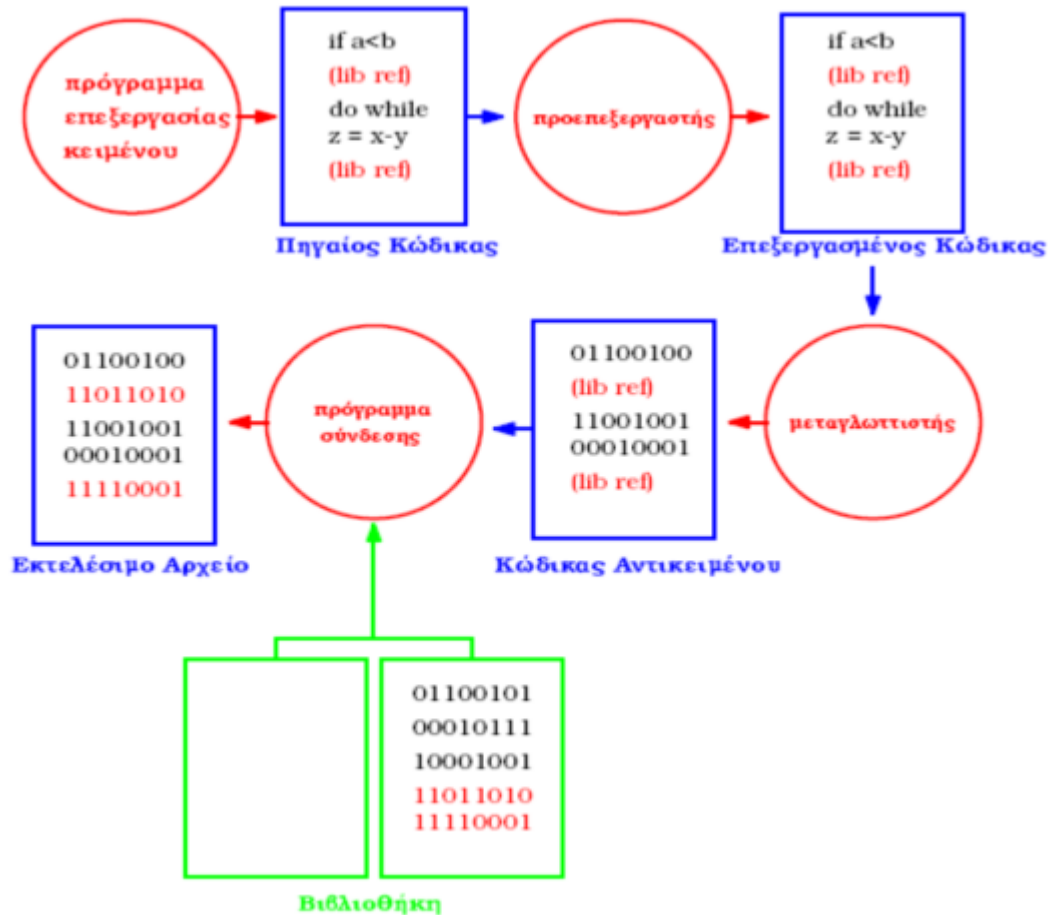


Μεταγλώττιση (Compilation)

- Ο υπολογιστής δεν «καταλαβαίνει» άμεσα μια γλώσσα υψηλού επιπέδου.
- Η μετάφραση από μία γλώσσα υψηλού επιπέδου στη γλώσσα μηχανής γίνεται μέσω των μεταγλωττιστών (compilers):
 - Οδηγίες του προγραμματιστή: αποθηκευμένες στο πηγαίο πρόγραμμα (source program).
 - Αντίστοιχες οδηγίες επιπέδου μηχανής: αποθηκευμένες στο αντικειμενικό ή τελικό πρόγραμμα (object program).
 - Σε ορισμένα μικρά υπολογιστικά συστήματα (π.χ. PC) το αντικειμενικό πρόγραμμα πρέπει να προετοιμαστεί για εκτέλεση από άλλο ειδικό πρόγραμμα, τον συνδέτη (linker).



Διαδικασία Μεταγλώττισης



Μεταγλωττιστές

- Μεταφράζουν μία συγκεκριμένη γλώσσα υψηλού επιπέδου στη γλώσσα μηχανής ενός συγκεκριμένου είδους.
- Μεγάλα προγράμματα που τοποθετούνται μόνιμα στη δευτερεύουσα μνήμη και όταν χρειαστεί μεταφέρονται στην κύρια μνήμη.
- Όταν τελειώσει η διαδικασία μεταγλώττισης είναι διαθέσιμο το εκτελέσιμο πρόγραμμα για τον χειρισμό των δεδομένων που δίνει τα αποτελέσματα στην έξοδο.



Είδη Προγραμματισμού

- Διαδικαστικός προγραμματισμός:
 - Pascal, Basic, C.
- Δηλωτικός προγραμματισμός:
 - Prolog.
- Αντικειμενοστρεφής προγραμματισμός:
 - C++, Java.
- Οπτικός προγραμματισμός:
 - Visual Basic, Visual C++, Visual Prolog.



Διαδικαστικός Προγραμματισμός (Procedural Programming)

- Κάθε πρόγραμμα είναι μία διαδικασία ή ένα σύνολο εργασιών που δίνονται με προκαθορισμένο τρόπο και σειρά στο πρόγραμμα.
- Οι περισσότερες γλώσσες προγραμματισμού είναι διαδικαστικές.
- Ευνοείται άμεσα η δημιουργία και εκτεταμένη χρήση βιβλιοθηκών προγραμμάτων που περιέχουν υπορουτίνες σχετικές με κοινά θέματα (π.χ. βιβλιοθήκη γραφικών).



Δηλωτικός Προγραμματισμός (Declarative Programming)

- Ένα πρόβλημα αποτελείται από σχέσεις που περιγράφουν τα συστατικά του προβλήματος και τους συσχετισμούς μεταξύ τους.
- Η λύση του προβλήματος προκύπτει από την εφαρμογή του μηχανισμού συμπερασμού της γλώσσας.
- Το πρόγραμμα τερματίζεται όταν βρεθούν όλες οι δυνατές λύσεις του προβλήματος.
- **Παραλληλισμός:** χάρτης \leftrightarrow σαφείς οδηγίες για την εύρεση της πορείας.



Αντικειμενοστρεφής Προγραμματισμός (Object-oriented Programming)

- Βασίζεται σε αντικείμενα (objects) διαφόρων κατηγοριών (classes), τα οποία αλληλεπιδρούν μεταξύ τους.
- Πρώτα ορίζονται τα κατάλληλα για το πρόβλημα αντικείμενα και στη συνέχεια χρησιμοποιούνται για να περιγραφεί βήμα προς βήμα η εξεύρεση της λύσης.



Οπτικός Προγραμματισμός (Visual Programming)

- Παρέχεται ευκολία ανάπτυξης ολοκληρωμένων εφαρμογών.
- Έτοιμες βιβλιοθήκες γραφικών που μπορούν να χρησιμοποιηθούν σε διαφορετικές εφαρμογές.
- Περισσότερο οικείος στους προγραμματιστές.



Εξέλιξη Γλωσσών Προγραμματισμού (1/3)

- FORTRAN (FORmula TRANslator):
 - J. Backus, 1954.
 - Η πρώτη ευρέως χρησιμοποιούμενη γλώσσα υψηλού επιπέδου.
 - Προορίζεται κυρίως για την επίλυση επιστημονικών και τεχνολογικών προβλημάτων.
 - FORTRAN-77, FORTRAN-90: οι πιο πετυχημένες εκδόσεις.



Εξέλιξη Γλωσσών Προγραμματισμού (2/3)

- COBOL (Common Business Oriented Language):
 - G. Hopper, 1961.
 - Προορίζεται κυρίως για την επίλυση οικονομικών και εμπορικών προβλημάτων που περιέχουν μεγάλο μεγέθους όγκου δεδομένων.
- BASIC (Beginners All-purpose Symbolic Instruction Code):
 - J. Kemmeny, T. Kurtz, 1961.
 - Ιδιαίτερα διαδεδομένη σε αρχάριους και απλούς χρήστες υπολογιστών.
 - Quick Basic, Visual Basic: οι πιο πετυχημένες εκδόσεις.
- PASCAL:
 - K. Jensen, N. Wirth, 1971.
 - Γενικής χρήσης.
 - Ακολουθεί τις αρχές του δομημένου προγραμματισμού.

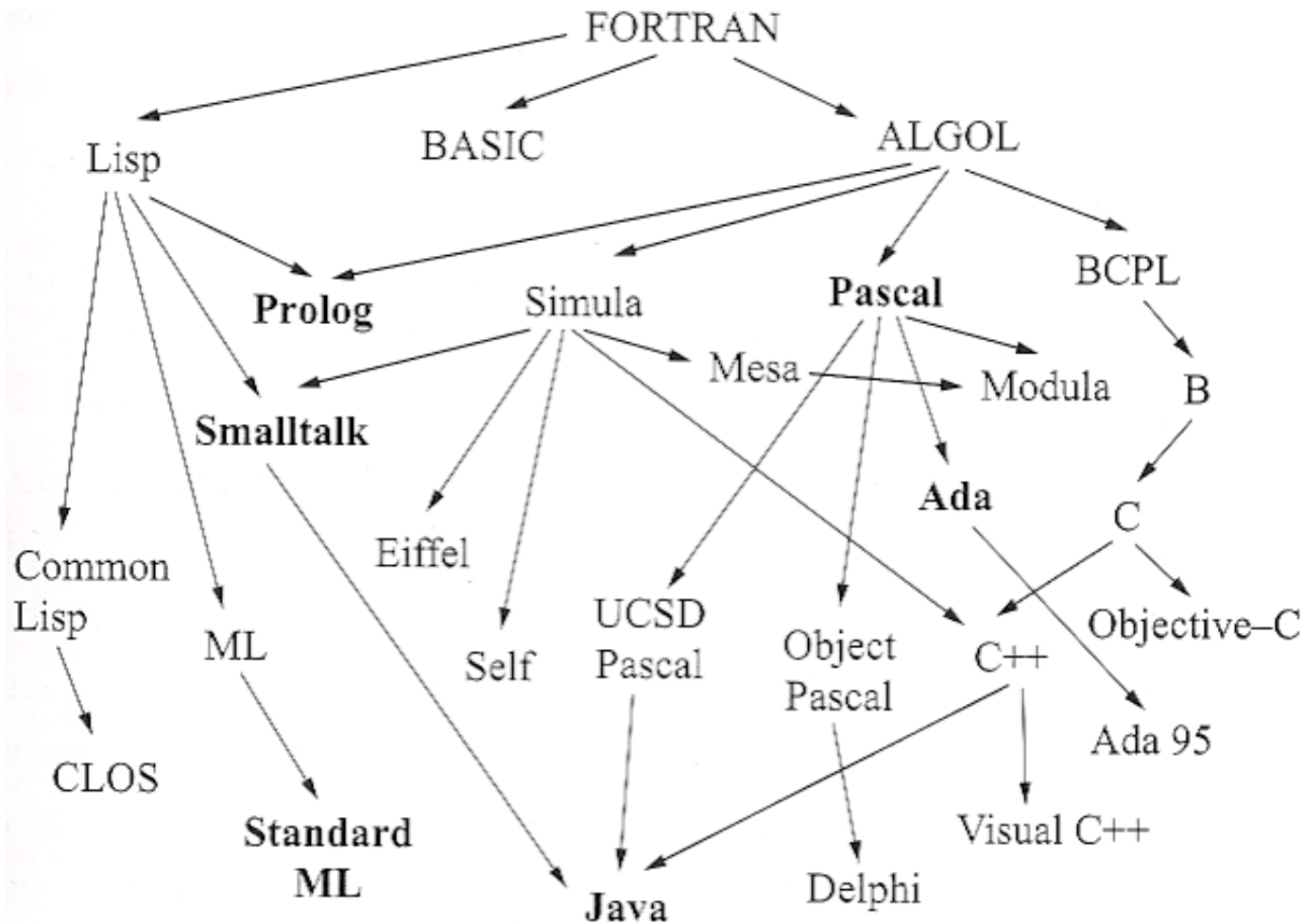


Εξέλιξη Γλωσσών Προγραμματισμού (3/3)

- PROLOG (PROgramming in LOGic):
 - 1970.
 - Λογικός προγραμματισμός, χειρισμός συμβόλων.
 - Βασίζεται στην μαθηματική λογική (κατηγορηματικός λογισμός πρώτης τάξης).
 - Χρησιμοποιείται σε εφαρμογές τεχνητής νοημοσύνης.
- C:
 - D. Ritchie, 1972.
 - Υποστήριξη πολύ εξειδικευμένων τελεστών.
 - Προσεγγίζει την ταχύτητα της γλώσσας Assembly.
 - Χρησιμοποιήθηκε για την κατασκευή του λειτουργικού συστήματος Unix.
 - C++: προγραμματισμός προσανατολισμένος σε αντικείμενα.



Γενεαλογικό Δένδρο Γλωσσών Προγραμματισμού



Ιστορία της C

- Σχεδιάστηκε και αναπτύχθηκε στα AT&T Labs από τον Dennis Ritchie.
- Βασίστηκε στις γλώσσες BCPL και B που χρησιμοποιούνταν στην ανάπτυξη του UNIX.
- Χρησιμοποιήθηκε το 1973 για να ξαναγραφεί το μεγαλύτερο τμήμα του UNIX.
- Πρώτη επίσημη τεκμηρίωση της γλώσσας το 1977.
- Αποδοχή του προτύπου ANSI C (1988).



Γιατί C; (1/2)

- Είναι σχετικά μικρή και εύκολη στην εκμάθηση.
- Υποστηρίζει top-down και bottom-up σχεδιασμό αλλά και δομημένο προγραμματισμό.
- Είναι αποτελεσματική (γρήγορα στην εκτέλεση προγράμματα).
- Είναι μεταφέρσιμη (portable) σε πολλά συστήματα.
- Δεν βάζει περιορισμούς.
- Αποτελεί με την C++ την πιο ευρύτερα χρησιμοποιούμενη γλώσσα σε ερευνητικά και αναπτυξιακά προγράμματα.



Γιατί C; (2/2)

- Υπάρχει μεγάλη βάση εφαρμογών που αναπτύχθηκαν με τη γλώσσα αυτή.
- Μπορεί να χρησιμοποιηθεί και ως γλώσσα χαμηλού επιπέδου (άμεση πρόσβαση στους πόρους) και ως γλώσσα υψηλού επιπέδου (διαθέσιμες βιβλιοθήκες).
- Η γνώση της C είναι καλό εφόδιο για την εκμάθηση πολλών άλλων γλωσσών (C++, Java, PHP).



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Στεργίου Κωνσταντίνος. «Εισαγωγή στον Δομημένο Προγραμματισμό». Έκδοση: 1.0. Κοζάνη 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.uowm.gr/courses/ICTE258/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Όχι Παράγωγα Έργα Μη Εμπορική Χρήση 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως Μη Εμπορική ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους
υπερσυνδέσμους.

