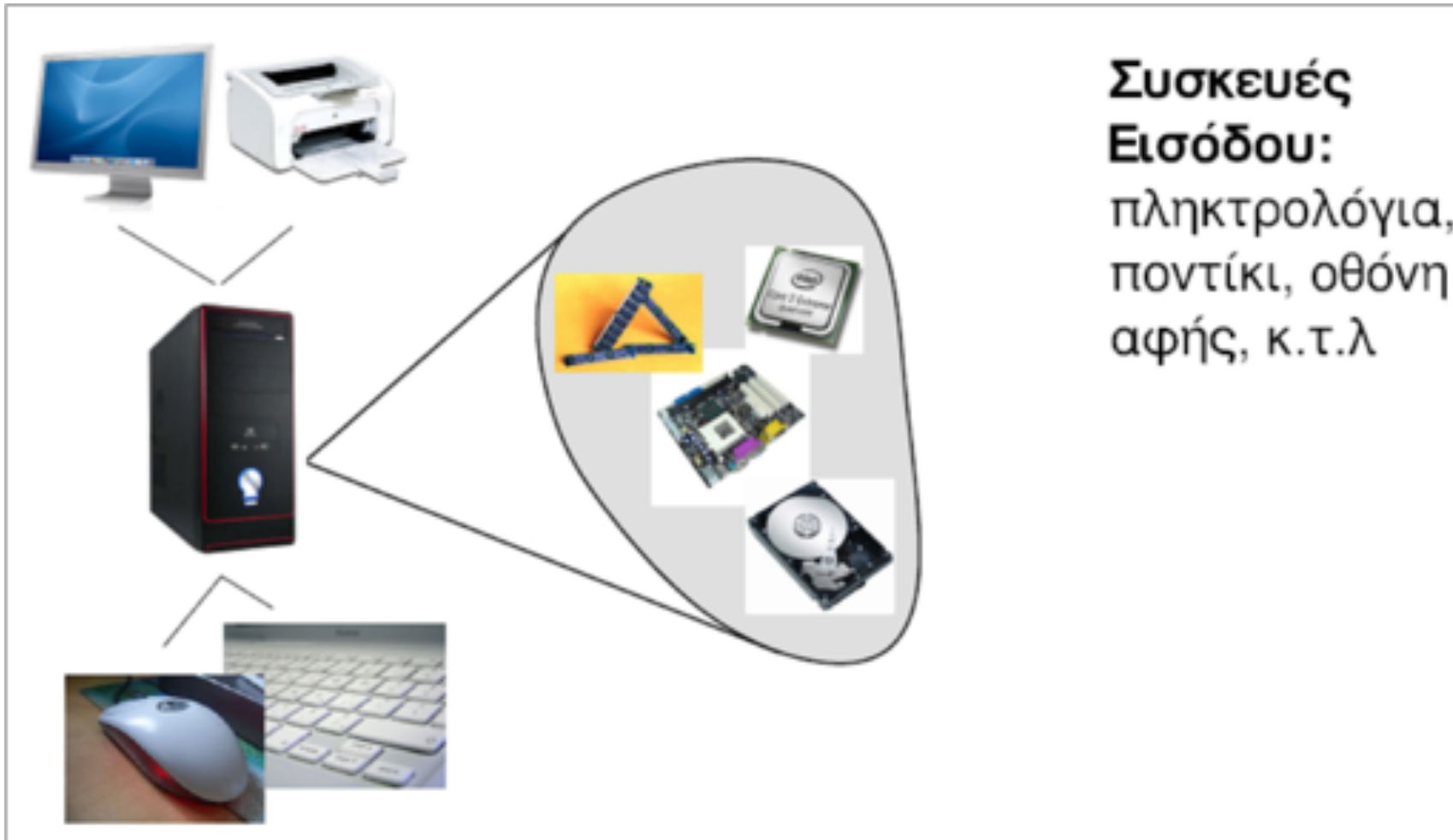


Τι είναι ο Η/Υ;

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**

Η ανατομία ενός Η/Υ



Η ανατομία ενός Η/Υ



Η ανατομία ενός Η/Υ



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

Πότε χρησιμοποιούμε υπολογιστή?

Πότε χρησιμοποιούμε υπολογιστή?

Χρήση υπολογιστή αν:

- Έχουμε πολλά στοιχεία να επεξεργαστούμε.
- Παράγονται πολλά αποτελέσματα.
- Μέθοδος επίλυσης εξαιρετικά πολύπλοκη για ένα άνθρωπο.
- Χρησιμοποιούμε την ίδια μέθοδο πολλές φορές.

Να θυμάστε:

- Ο Η/Υ εκτελεί «πράξεις» γρηγορότερα από το άνθρωπο.
- Ο Η/Υ μπορεί να «θυμάται» περισσότερα πράγματα από τον άνθρωπο.
- Ο Η/Υ μπορεί να εκτελέσει με «ακρίβεια» μια λογική σειρά εντολών.

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

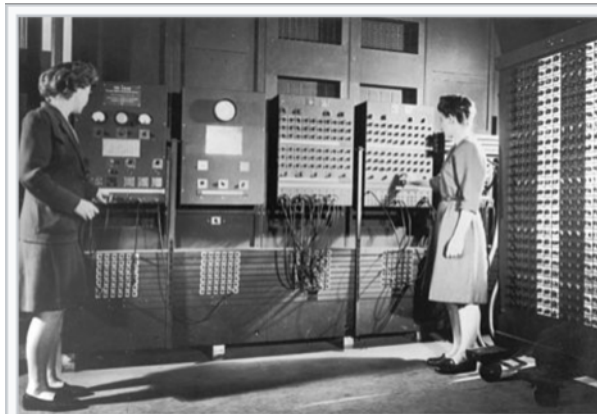
- **Πρόγραμμα** ενός Η/Υ είναι ένα σύνολο εντολών που καθοδηγούν τον υπολογιστή να εκτελέσει καθορισμένες ενέργειες
- **Προγραμματισμός** ενός Η/Υ είναι ο σαφής καθορισμός μιας διαδικασίας ως συνόλου εντολών (το πρόγραμμα) που περιγράφει λεπτομερώς τα βήματα που πρέπει να εκτελεστούν για να λυθεί ένα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

A **programmer**, **developer**, **dev**, **coder**, or **software engineer** is a person who creates **computer software**. The term *computer programmer* can refer to a specialist in one area of **computers** or to a generalist who writes code for many kinds of software. One who practices or professes a formal approach to programming may also be known as a **programmer analyst**.

Wikipedia



Two programmers working on an IBM Type 704 at [NACA](#), 1954



[Betty Jennings](#) and [Fran Bilas](#), part of the first [ENIAC](#) programming team.



[Trevor Parscal](#) coding in the Wikimedia Foundation offices.

Desired Skills and Qualities

- **Analytical Thinking:** Computer programmers need to understand, manipulate, and repair complex computer code.
- **Strong Attention to Detail:** Computer programmers need to pay attention to every line of code written. One wrong command and the entire program could malfunction.
- **Collaboration:** Computer programmers may need help from another department or colleague to fix a software issue. It is important they have a collaborative mindset.

Why Coding Is Still The Most Important Job Skill Of The Future

- Apart from companies in the technology sector, there are an increasing number of businesses relying on computer code. A software engineer could just as easily find themselves working at Apple, as they could in a hospital, or at an automotive manufacturer.

Seven million job openings in 2015 were in occupations that required coding skills, and programming jobs overall are growing 12% faster than the market average.

One major revelation: Half of all programming openings are in industries outside of technology. Among them:

- Finance
- Manufacturing
- Health Care

This is the skill you need today if you want a job in one of the highest paying industries

Ο κύκλος ανάπτυξης προγράμματος

Ο κύκλος ανάπτυξης προγράμματος αναλύεται σε έξι βασικά βήματα:

1. Περιγραφή του προβλήματος, καθορισμός απαιτήσεων.
2. Ανάλυση προβλήματος (inputs, outputs), προσδιορισμός της λύσης
3. Σχεδίαση της λύσης του προβλήματος
 - Ανάπτυξη αλγορίθμου
 - Σχεδιασμός διαγράμματος ροής
 - Δημιουργία ψευδοκώδικα
4. Κωδικοποίηση σε γλώσσα προγραμματισμού,
5. Έλεγχος, διόρθωση λαθών,
6. Συντήρηση προγράμματος, τεκμηρίωση.

Αλγόριθμος

- Πριν δημιουργήσουμε ένα πρόγραμμα για να λύσουμε ένα πρόβλημα **πρέπει να σχεδιάσουμε έναν αλγόριθμο** για αυτό το πρόβλημα
- Μία 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

Αλγόριθμοι

Παραδείγματα αλγορίθμων

- Οδηγίες πλυντηρίου
- Οδηγίες για συναρμολόγηση επίπλου

Στην Πληροφορική

- Ταξινόμηση Λίστας Αριθμών
- Εύρεση Μέσου Όρου Μιας Λίστας Αριθμών
-

Εντολές

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.

Αρχές Ανάπτυξης Προγραμμάτων

- **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

- **Μεταγλώττιση – Σύνδεση**

- Ο μεταφραστής ελέγχει το πηγαίο πρόγραμμα για γραμματικά λάθη (ορθογραφία εντολών) και συντακτικά λάθη (εντολές που λείπουν)
- Το πηγαίο πρόγραμμα μετατρέπεται σε γλώσσα μηχανής

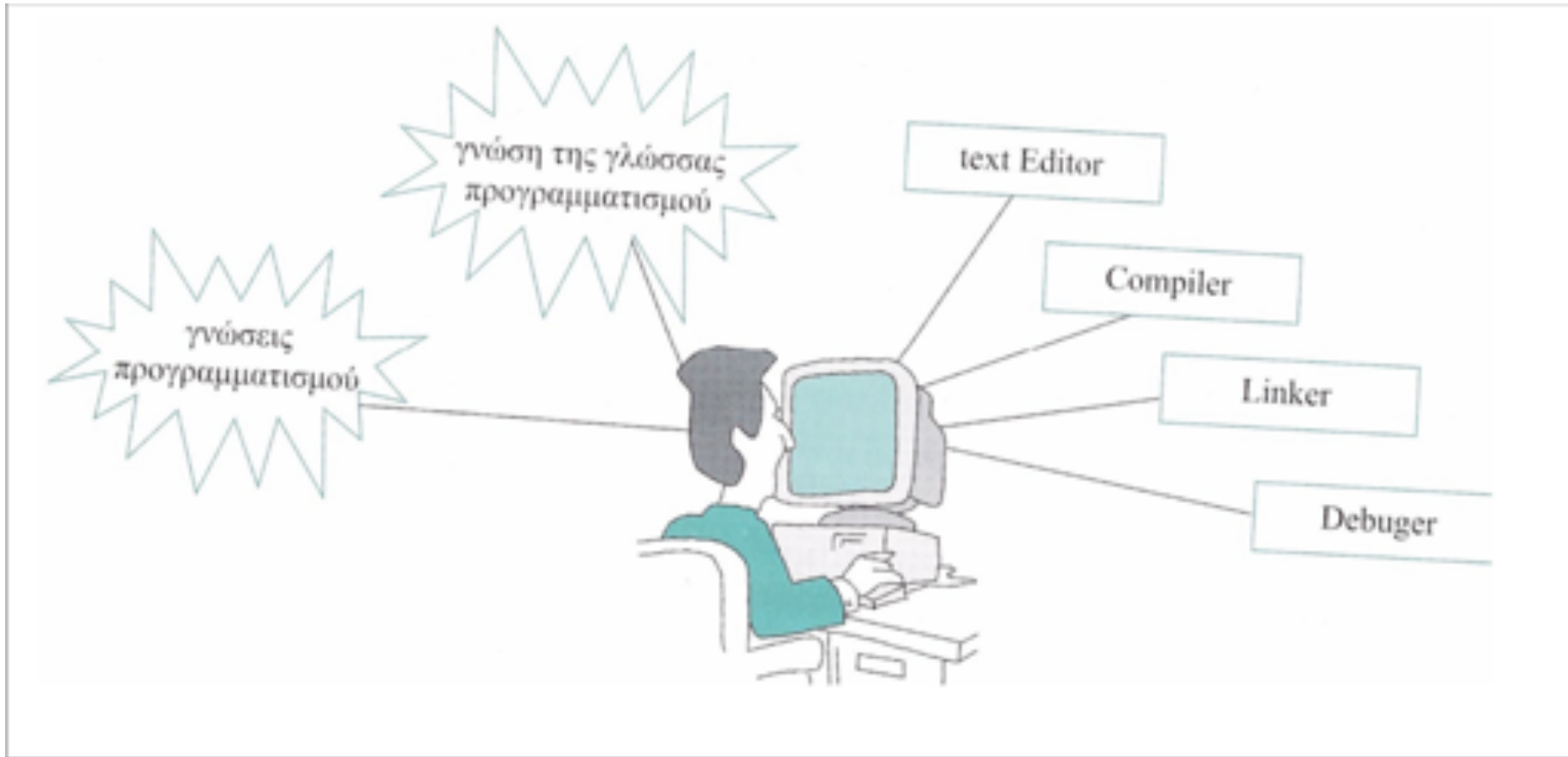
- **Διορθώσεις**

- 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 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

Γλώσσα Ψευδοκώδικα

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")

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

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

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







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

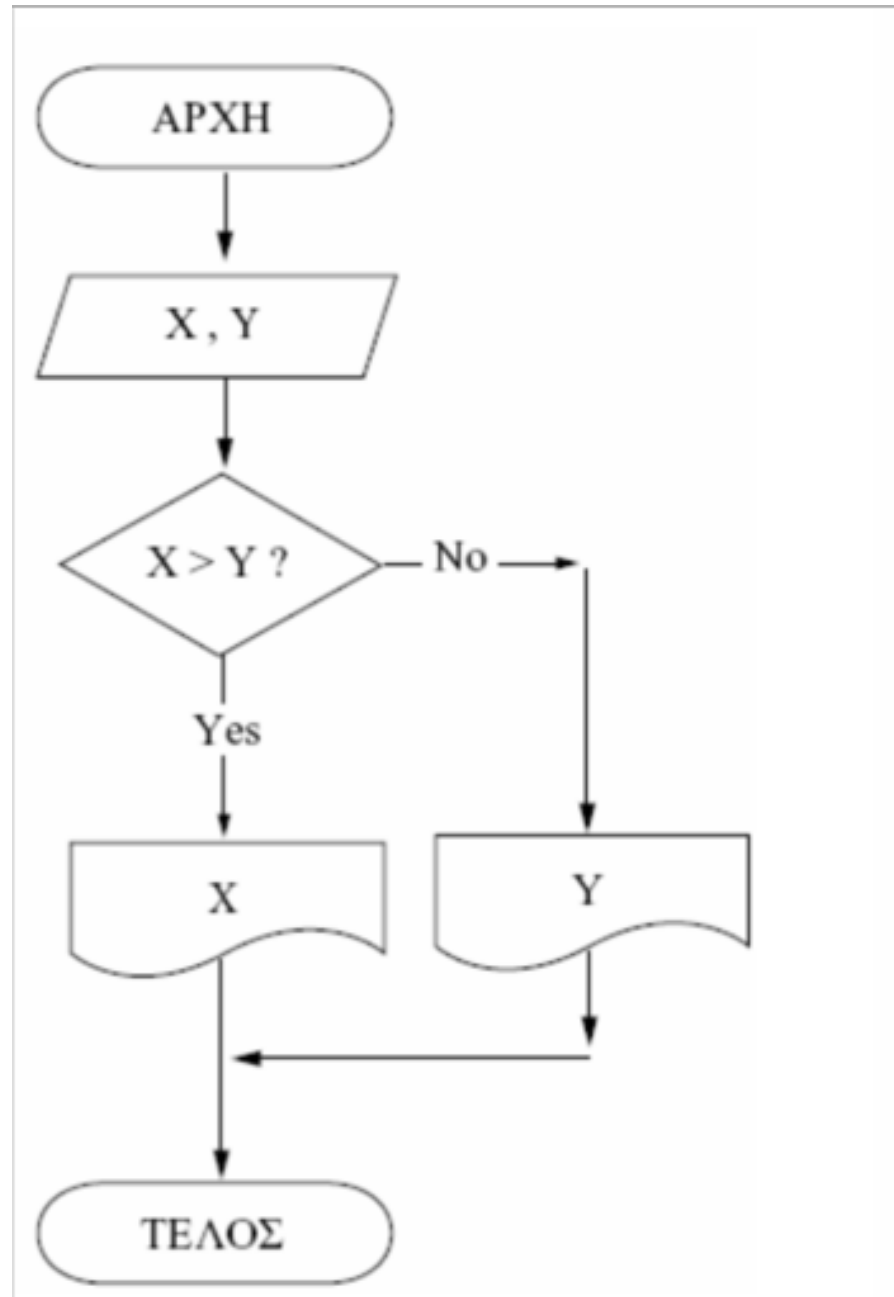
Διαγράμματα Ροής

- 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

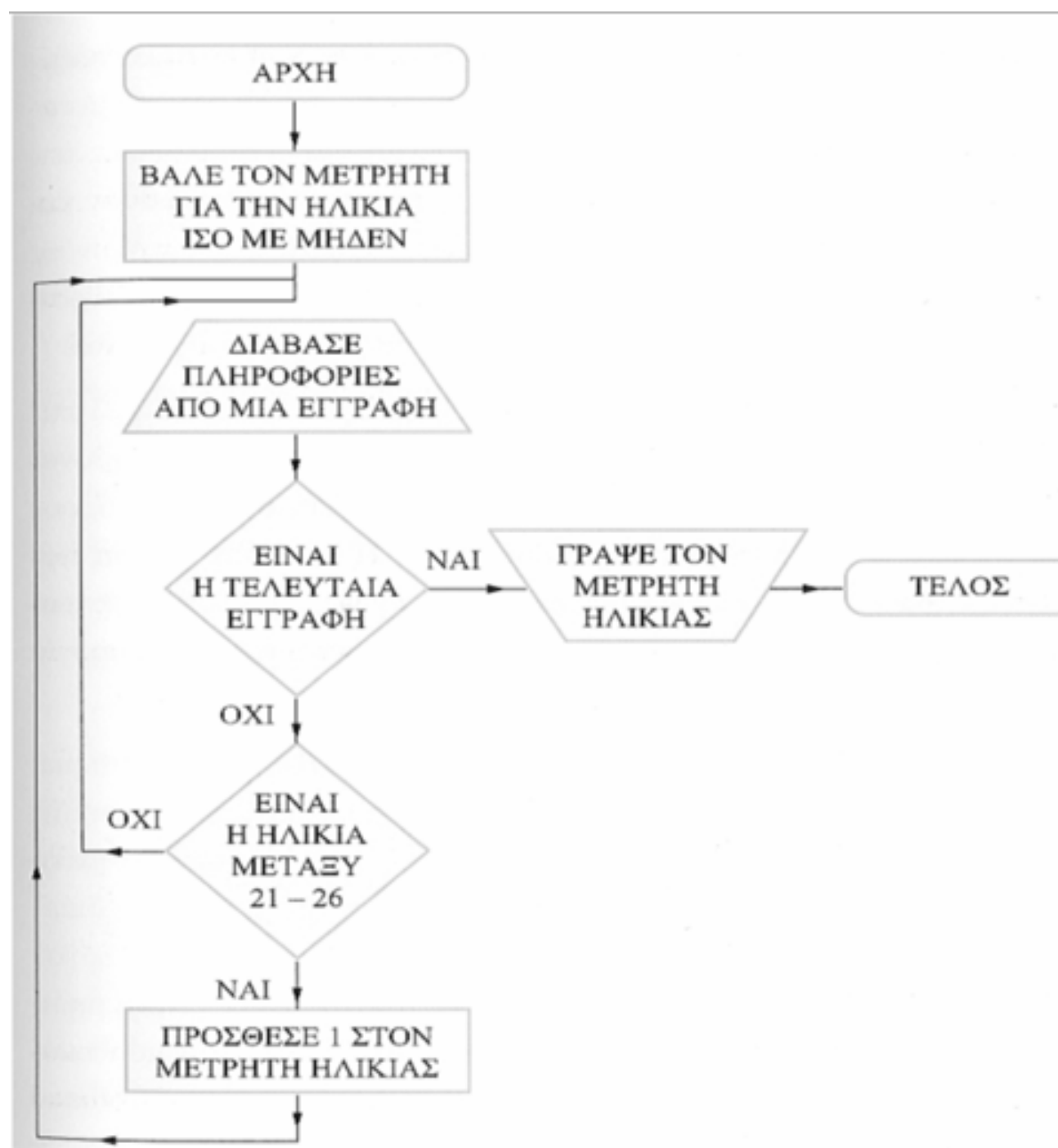
	Τερματισμός ή Αρχή BEGIN - END		Σύμβολο απόφασης, χρησιμοποιείται όταν υπάρχουν δύο εναλλακτικές διαδρομές
	Κατεύθυνση της ροής του προγράμματος		Σύμβολο εισόδου, εξόδου. Χρησιμοποιείται όταν διαβάζονται αποτελέσματα
	Γενικό σύμβολο που δηλώνει διεργασίες (μαθηματικές πράξεις, αρχικοποίηση μεταβλητών)		Σύμβολο επέκτασης (οθόνη ή εκτυπωτή)

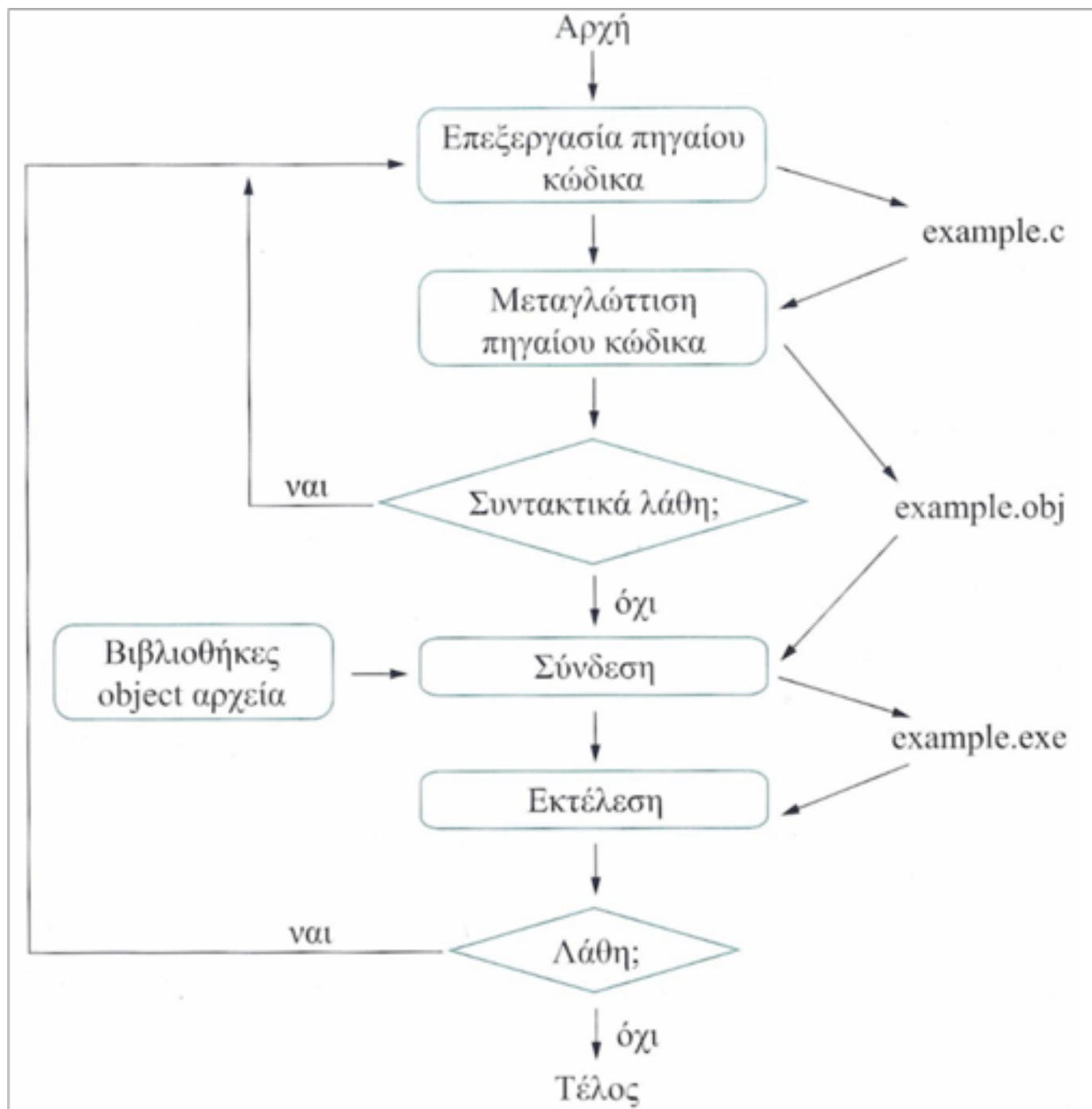
Παραδείγματα



Παράδειγμα

- 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





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)
- Στο στάδιο αυτό γίνεται ο έλεγχος συντακτικών λαθών.

Μεταγλωττιστές

- Μεταφράζουν μία συγκεκριμένη γλώσσα υψηλού επιπέδου στη γλώσσα μηχανής ενός συγκεκριμένου είδους
- Όταν τελειώσει η διαδικασία μεταγλώττισης είναι διαθέσιμο το εκτελέσιμο πρόγραμμα για τον χειρισμό των δεδομένων που δίνει τα αποτελέσματα στην έξοδο

Έλεγχος λαθών

Έλεγχος λαθών και διόρθωση προγράμματος:

– Διορθώνονται πιθανά λογικά σφάλματα (σφάλματα που σχετίζονται με τον σχεδιασμό της λύσης).

- Σφάλματα σύνταξης: Σφάλματα που σχετίζονται με το αν χρησιμοποιήσαμε σωστά τη γλώσσα προγραμματισμού στη διάρκεια της υλοποίησης

- Σφάλματα Run-time: Σφάλματα κατά τη διάρκεια εκτέλεσης του προγράμματος