

ΚΕΦΑΛΑΙΟ 2.3

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

1. Αναφορά σε γλώσσες προγραμματισμού

Πρόγραμμα είναι το σύνολο των εντολών που χρειάζεται να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί κάποιος αλγόριθμος για την επίλυση του προβλήματος. Το πρόγραμμα γράφεται σε κάποια **γλώσσα προγραμματισμού**, δηλ. σε μια γλώσσα κατανοητή από τον υπολογιστή.

Η εργασία σύνταξης των προγραμμάτων σε κάποια γλώσσα προγραμματισμού, ονομάζεται **προγραμματισμός**, ενώ τα άτομα που ασχολούνται με αυτό είναι οι **προγραμματιστές**.

Ο υπολογιστής εκτελεί στοιχειώδεις ενέργειες (πρόσθεση, μεταφορά και σύγκριση) μεταξύ **δυναδικών ψηφίων (0 και 1)**, αλλά τις ενέργειες αυτές τις εκτελεί ταχύτατα.

Γλώσσα μηχανής ή γλώσσα 1ης γενιάς: Τα προγράμματα γράφονται στη γλώσσα που καταλαβαίνει ο υπολογιστής δηλ. είναι μια ακολουθία από 0 και 1. Ο τρόπος αυτός είναι επίπονος και απαιτεί άριστη γνώση του υλικού του υπολογιστή.

Συμβολικές γλώσσες ή γλώσσα χαμηλού επιπέδου ή 2ης γενιάς: Χρήση συμβολικών ονομάτων για τις εντολές μηχανής και χρήση του δεκαδικού συστήματος αρίθμησης. Η μετάφραση των εντολών αυτών σε γλώσσα μηχανής γίνεται από τον **συμβολομεταφραστή (assembler)**. Παραμένουν συνδεδεμένες με την αρχιτεκτονική κάθε υπολογιστή.

Γλώσσες υψηλού επιπέδου ή 3ης γενιάς: Γλώσσες προγραμματισμού στις οποίες:

- Είναι φυσικότερος και πιο “ανθρώπινος” ο τρόπος έκφρασης των προβλημάτων
- Υπάρχει δυνατότητα μεταφοράς “μεταφερσιμότητα” των προγραμμάτων σε διαφορετικούς υπολογιστές.
- Εύκολη εκμάθηση, διόρθωση λαθών και συντήρηση προγραμμάτων.

Οι εντολές των γλωσσών υψηλού επιπέδου είναι συνδυασμοί **απλών αγγλικών λέξεων** και ακολουθούν **αυστηρούς κανόνες σύνταξης**. Τα προγράμματα που γράφονται σε γλώσσα υψηλού επιπέδου **μεταφράζονται** από τον υπολογιστή σε ακολουθίες εντολών μηχανής με τη βοήθεια ειδικών προγραμμάτων που ονομάζονται **μεταγλωττιστές (compilers)** και **διερμηνευτές (interpreters)**

Αρχικά για τη συγγραφή ενός προγράμματος υπήρχε ένα περιβάλλον **Γραμμής εντολών**, όπου ο χρήστης πληκτρολογεί τις εντολές, ενώ στα **γραφικά περιβάλλοντα** όλη η διαδικασία γίνεται με γραφικά και σχήματα.

Γλώσσα Προγραμματισμού	Βασικά Χαρακτηριστικά
FORTRAN	Μαθηματικές και επιστημονικές εφαρμογές.
COBOL	Εμπορικές και διαχειριστικές εφαρμογές. (Επιχειρήσεις και δημόσιους οργανισμούς).
LISP	Χειρισμός λιστών από σύμβολα εφαρμογές τεχνητής νοημοσύνης
PROLOG	Χρησιμοποιείται σε προβλήματα τεχνητή νοημοσύνης , π.χ. έμπειρα συστήματα.
BASIC	Δημοφιλής γλώσσα για πλήθος εφαρμογών και για εκπαίδευση αρχαρίων στον προγραμματισμό. Τα προγράμματα της εκτελούνται με τη βοήθεια διερμηνευτή .
PASCAL	Γενικής χρήσης, κατάλληλη για εκπαίδευση και για δημιουργία ισχυρών δομημένων προγραμμάτων κάθε τύπου.
C και C++	Χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος UNIX . Είναι κατάλληλη για ανάπτυξη δομημένων εφαρμογών Έχει πολλές δυνατότητες γλώσσας χαμηλού επιπέδου . Γλώσσα προγραμματισμού συστημάτων
Visual Γλώσσες	Αποτελούν εξέλιξη των παραπάνω γλωσσών, προκειμένου να ανταποκριθούν σε γραφικό περιβάλλον προγραμματισμού. (Visual Basic, Visual C κτλ)
JAVA	Κατάλληλη για την ανάπτυξη εφαρμογών που θα εκτελούνται σε καταναεμημένα περιβάλλοντα δηλ. στο Διαδίκτυο (Internet).
SCRATCH - Google AppInvertor	Οπτικές γλώσσες προγραμματισμού.
LOGO, GameMaker	Εκπαιδευτικούς σκοπούς

Γλώσσες 4ης Γενιάς: Ο απλός χρήστης μπορεί να επιλύει από μόνος του μικρά προβλήματα, χωρίς τη βοήθεια κάποιου προγραμματιστή. Βασικό στοιχείο των γλωσσών αυτών, όπως η **SQL**, είναι η ύπαρξη βάσεων δεδομένων από τις οποίες ο χρήστης μπορεί να αντλήσει συγκεκριμένες πληροφορίες και να τις εμφανίσει με συγκεκριμένο τρόπο.

2. Προγραμματιστικά υποδείγματα

Τα κυριότερα είδη προγραμματισμού είναι:

- Προστακτικός προγραμματισμός:** Χρήση απλών εντολών στην προστακτική (Διάβασε, Γράψε κτλ). Προγράμματα που προκύπτουν από γλώσσες όπως η PASCAL και η C.

- b. **Δηλωτικός προγραμματισμός:** Χρήση λογικών εκφράσεων και έλεγχος μέσω του προγράμματος για το ποιες ισχύουν. Γλώσσες: PROLOG και SQL.
- c. **Συναρτησιακός προγραμματισμός:** επίλυση του προβλήματος με τη χρήση μαθηματικών συναρτήσεων. Γλώσσα LISP
- d. **Αντικειμενοστραφής προγραμματισμός:** Χρήση αντικειμένων που περιέχουν τόσο τα δεδομένα όσο και τις λειτουργίες που επενεργούν σε αυτά.
- e. **Λογικός προγραμματισμός:** Προτάσεις σε μορφή λογικών εκφράσεων. PROLOG
- f. **Παράλληλος προγραμματισμός:** Προγράμματα που εκμεταλλεύονται την ύπαρξη πολλών επεξεργαστών στον ίδιο υπολογιστή.

3. Δομημένος προγραμματισμός

Ο δομημένος προγραμματισμός χρησιμοποιεί:

1. την **ιεραρχική σχεδίαση** για την ανάπτυξη του αλγορίθμου επίλυσης του προβλήματος. Με τον όρο αυτό εννοούμε την τεχνική διάσπασης του αρχικού προβλήματος σε επιμέρους υποπροβλήματα και αυτών σε απλούστερα μέχρι να φτάσουμε σε υποπροβλήματα πολύ απλά και εύκολα στην επίλυσή τους.
2. Τον **τμηματικό προγραμματισμό** για την υλοποίηση του προγράμματος. Ο τμηματικός προγραμματισμός υλοποιεί την ιεραρχική σχεδίαση, όπου κάθε υποπρόβλημα αποτελεί μια ανεξάρτητη ενότητα που ονομάζεται υποπρόγραμμα.
3. **τρεις βασικές συνιστώσες** για τη συγγραφή των προγραμμάτων. Οι τρεις αυτές συνιστώσες είναι η δομή της **ακολουθίας**, η δομή της **επιλογής** και η δομή της **επανάληψης**.

4. Σχεδίαση και συγγραφή κώδικα

Η διαδικασία συγγραφής και η παράγωγή του εκτελέσιμου προγράμματος περιλαμβάνει τα παρακάτω στάδια:

1. Στο **Συντάκτη (editor)** δημιουργείται το αρχικό πρόγραμμα, το οποίο ονομάζεται **πηγαίο πρόγραμμα (source program)**.
2. Επειδή το πρόγραμμα αυτό δεν είναι κατανοητό στον υπολογιστή πρέπει να μεταφραστεί σε κατάλληλη μορφή με τον μεταγλωττιστή. Το πρόγραμμα που παράγεται από τον μεταγλωττιστή ονομάζεται **αντικείμενο πρόγραμμα (object program)**.
3. Κατά τη διαδικασία παραγωγής του αντικείμενου προγράμματος, ο μεταγλωττιστής **ανιχνεύει** τα λάθη του προγράμματος (**μόνο τα συντακτικά λάθη** και όχι για τα **λογικά λάθη**) και εμφανίζει ενημερωτικά μηνύματα στον προγραμματιστή. Η διαδικασία της μεταγλώττισης **επαναλαμβάνεται** μέχρις ότου ο προγραμματιστής επιλύσει όλα τα συντακτικά λάθη.
4. Το αντικείμενο πρόγραμμα είναι σε **μορφή κατανοητή** από τον υπολογιστή, αλλά πρέπει να **συνδεθεί** με άλλα τμήματα προγράμματος απαραίτητα για την εκτέλεση του. Τα τμήματα αυτά τα γράφει ο προγραμματιστής ή βρίσκονται στις **βιβλιοθήκες** της γλώσσας προγραμματισμού.
5. Τη διαδικασία της σύνδεσης την αναλαμβάνει ο **συνδέτης η φορτωτής (linker – loader)**.

6. Το αποτέλεσμα του συνδέτη είναι η παραγωγή του **εκτελέσιμου (executable)** προγράμματος, το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή.
7. Τα **λογικά λάθη** δεν ανιχνεύονται κατά τη μεταγλώττιση αλλά κατά την εκτέλεση του προγράμματος και για αυτό οι περισσότερες γλώσσες περιλαμβάνουν εργαλεία που βοηθούν το χρήστη να παρακολουθεί την εκτέλεση του προγράμματος και τις τιμές που παίρνουν οι μεταβλητές σε κάθε βήμα.

5. Κύκλος ζωής εφαρμογής λογισμικού

Ένα πρόγραμμα ξεκινάει τη ζωή του από τη στιγμή που θα καθοριστούν οι απαιτήσεις του μέχρι τη στιγμή που σταματάει πλέον η συντήρησή του (προσθήκες, αλλαγές, βελτιώσεις). Ο **Κύκλος Ζωής** ενός προγράμματος περιλαμβάνει τις παρακάτω φάσεις:

- **Ανάλυση:** Προσδιορισμός του προβλήματος από τον πελάτη και αναλυτική καταγραφή των δεδομένων και των ζητούμενων του προβλήματος.
- **Σχεδίαση:** Καθορίζεται η δομή του προγράμματος, οι ενότητες (υποπρογράμματα) και αναζητούνται έτοιμες ενότητες από παλιότερα προγράμματα. Επιλέγονται ακόμη οι αλγόριθμοι και οι δομές δεδομένων που θα χρησιμοποιηθούν.
- **Υλοποίηση:** Χρήση κάποιας γλώσσας προγραμματισμού, συγγραφή του πηγαίου προγράμματος, διόρθωση συντακτικών λαθών και δημιουργία του εκτελέσιμου προγράμματος.
- **Λειτουργία και συντήρηση:** Προσαρμογές και βελτιώσεις που απαιτούνται προκειμένου το πρόγραμμα να συνεχίσει να χρησιμοποιείται. Οι ανάγκες αυτές προκύπτουν από λογικά λάθη του προγράμματος αλλά και από νέες προδιαγραφές που πρέπει να ενσωματωθούν. Για να γίνουν οι αλλαγές αυτές πρέπει να επαναληφθούν οι προηγούμενες φάσεις.

Δίπλα από κάθε λογισμικό υπάρχει και ένας αριθμός έκδοσης π.χ. Windows 8. Όταν οι αλλαγές είναι σημαντικές τότε ο αριθμός έκδοσης αυξάνεται κατά ακέραιο αριθμό (Windows 7.1, Windows 8.0), ενώ όταν γίνονται μικρές αλλαγές τότε η έκδοση αυάνεται κατά δέκατα ή εκατοστά (windows 8.0 windows 8.1 κλπ)

Οι εκδόσεις **beta** μιας εφαρμογής δίνονται για έλεγχο σε κάποιους τελικούς χρήστες. Οι χρήστες αυτοί **beta testers** παίρνουν την τελική έκδοση δωρεάν ή με μεγάλη έκπτωση.

Λογισμικό κλειστού κώδικα.

Λογισμικό που παράγεται από εταιρείες λογισμικού και δεν παρέχουν στο χρήστη τη δυνατότητα να δει ή να μεταβάλει τον κώδικα. Κάθε αλλαγή γίνεται από την εταιρεία που έχει παράγει τον κώδικα.

Ελεύθερο Λογισμικό/ Λογισμικό Ανοικτού Κώδικα.

Ο κώδικας είναι ελεύθερος και διαθέσιμος προς όλους. Ο καθένας μπορεί να κάνει τροποποιήσεις, βελτιώσεις, να αντιγράψει το κώδικα, να το μοιράσει ή να το πουλήσει. Το διαδίκτυο αποτελεί το βασικό μέσο συνεργασίας των προγραμματιστών για τη πρόσβαση στο διαθέσιμο Ελεύθερο Λογισμικό.

ΑΣΚΗΣΕΙΣ – ΕΡΩΤΗΣΕΙΣ**ΚΕΦΑΛΑΙΟΥ 2.3****ΕΡΩΤΗΣΕΙΣ ΘΕΩΡΙΑΣ**

- 2.1 Τι ονομάζουμε πρόγραμμα;
- 2.2 Τι εννοούμε με τον όρο μεταφερσιμότητα προγράμματος;
- 2.3 Να αναφέρετε τρεις τουλάχιστον γλώσσες προγραμματισμού υψηλού επιπέδου.
- 2.4 Να αναφέρετε δύο τουλάχιστον γλώσσες προγραμματισμού με τη χρήση γραφικού περιβάλλοντος.
- 2.5 Ποιες είναι οι βασικές συνιστώσες του δομημένου προγραμματισμού;
- 2.6 Τι ονομάζουμε ιεραρχική σχεδίαση;
- 2.7 Σε τι χρησιμεύουν τα μεταφραστικά προγράμματα;
- 2.8 Τι είναι τα συντακτικά λάθη και σε ποια φάση του προγραμματισμού τα ανακαλύπτουμε;
- 2.9 Τι είναι τα λογικά λάθη και πως μπορούν να διορθωθούν;
- 2.10 Τι είναι ο συντάκτης;
- 2.11 Ποιο είναι το πηγαίο, το αντικείμενο και το εκτελέσιμο πρόγραμμα;
- 2.12 Ποιες είναι οι φάσεις του κύκλου ζωής μιας εφαρμογής λογισμικού; Τι συμβαίνει σε κάθε μια από τις φάσεις αυτές;
- 2.13 Τι ονομάζουμε έκδοση ενός λογισμικού; Γιατί και πως μεταβάλλεται η έκδοση αυτή;
- 2.14 Τι είναι η έκδοση **beta** ενός λογισμικού;
- 2.15 Τι ονομάζουμε Ελεύθερο Λογισμικό/Λογισμικό Ανοικτού κώδικα;

ΕΡΩΤΗΣΕΙΣ ΣΩΣΤΟ ΛΑΘΟΣ

Σημειώστε *αν είναι σωστή ή λανθασμένη* καθεμιά από τις παρακάτω προτάσεις

- 2.16 Ένα πρόγραμμα σε γλώσσα μηχανής εκτελείται ταχύτερα από το ίδιο πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου.
- 2.17 Οι εντολές στις συμβολικές γλώσσες αποτελούνται από ακολουθίες 0 και 1
- 2.18 Ένα πρόγραμμα σε συμβολική γλώσσα εκτελείται μόνο σε συγκεκριμένους υπολογιστές.
- 2.19 Ο δομημένος προγραμματισμός υλοποιείται με τη χρήση δύο βασικών δομών.
- 2.20 Η μεταφερσιμότητα είναι χαρακτηριστικό των γλωσσών υψηλού επιπέδου.
- 2.21
- 2.22 Ο συνδέτης ενώνει το πηγαίο πρόγραμμα με το αντικείμενο πρόγραμμα
- 2.23 Το αντικείμενο πρόγραμμα δεν είναι εκτελέσιμο από τον υπολογιστή.
- 2.24 Ο συντάκτης χρησιμοποιείται για τη συγγραφή του εκτελέσιμου προγράμματος.
- 2.25 Τα λογικά λάθη τα διαπιστώνουμε κατά τη φάση εκτέλεσης του προγράμματος.
- 2.26 Τα συντακτικά λάθη τα ανακαλύπτουμε κατά τη φάση της μετάφρασης του πηγαίου προγράμματος σε αντικείμενο.
- 2.27 Κατά τη φάση της Υλοποίησης μια εφαρμογής βρίσκουμε και διορθώνουμε τα λογικά λάθη του προγράμματος.
- 2.28 Κατά τη φάση της Σχεδίασης μιας εφαρμογής επιλέγουμε τις δομές δεδομένων που θα χρησιμοποιήσουμε στην εφαρμογή.
- 2.29 Η έκδοση beta μιας εφαρμογής είναι δοκιμαστική έκδοση με στόχο την εύρεση λογικών λαθών της εφαρμογής.
- 2.30 Σε μια εφαρμογή κλειστού κώδικα μπορεί να επέμβει ο τελικός χρήστης.

ΕΡΩΤΗΣΕΙΣ ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΟΥ

Να συμπληρώσετε τα κενά των παρακάτω προτάσεων βάζοντας την κατάλληλη λέξη

- 2.31** Ο δομημένος προγραμματισμός υλοποιείται με τη χρήση τριών βασικών δομών. Τη δομή της, τη δομή της και τη δομή της
- 2.32** Ο χρησιμοποιείται για τη συγγραφή του προγράμματος.
- 2.33** Ο ενώνει το αντικείμενο πρόγραμμα με τις της γλώσσας και παράγει το πρόγραμμα.
- 2.34** Τα λάθη τα ανακαλύπτουμε κατά τη φάση της του προγράμματος ενώ τα κατά τη φάση της εκτέλεσης του.
- 2.35** Κατά τη φάση της ενός προγράμματος γίνεται ο του προβλήματος από τον πελάτη.
- 2.36** Κατά τη φάση της ενός προγράμματος επιλέγεται η κατάλληλη προγραμματισμού για την επίλυσή του.
- 2.37** Οι εκδόσεις μιας εφαρμογής δίνονται για έλεγχο σε κάποιους τελικούς χρήστες.

ΕΡΩΤΗΣΕΙΣ ΕΠΙΛΟΓΗΣ

Να επιλέξετε την απάντηση που αντιστοιχεί στην κάθε ερώτηση.

- 2.38** Οι εντολές ενός προγράμματος γράφονται σε ένα πρόγραμμα που ονομάζεται:
 Α) Συντάκτης Β) Μεταγλωτιστής
 Γ) Διερμηνευτής Δ) Συνδέτης
- 2.39** Ο μεταγλωτιστής βρίσκει
 Α) όλα τα λάθη του προγράμματος
 Β) μόνο τα λογικά λάθη του προγράμματος
 Γ) μόνο τα συντακτικά λάθη του προγράμματος
 Δ) μόνο λάθη που οφείλονται σε αναγραμματισμούς εντολών
- 2.40** Σε μια εφαρμογή ανοικτού κώδικα
 Α) δεν μπορεί ο χρήστης να προσπελάσει (διαβάσει) τον κώδικα
 Β) πρέπει ο χρήστης να περιμένει τις αναβαθμίσεις από την εταιρεία κατασκευής της.
 Γ) ο χρήστης δεν μπορεί να προσθέσει λειτουργίες.
 Δ) ο χρήστης μπορεί να δει τον κώδικα και να προσθέσει λειτουργίες

ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΙΣΗΣ

2.41 **Να αντιστοιχίσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β**

A	B (εξήγηση)
1 Μετάφραση προγράμματος	α. Συντάκτης
2 Συγγραφή προγράμματος	β. Συνδέτης
3 Σύνδεση με βιβλιοθήκες	γ. Μεταγλωτιστής
	δ. Εκτελέσιμο πρόγραμμα