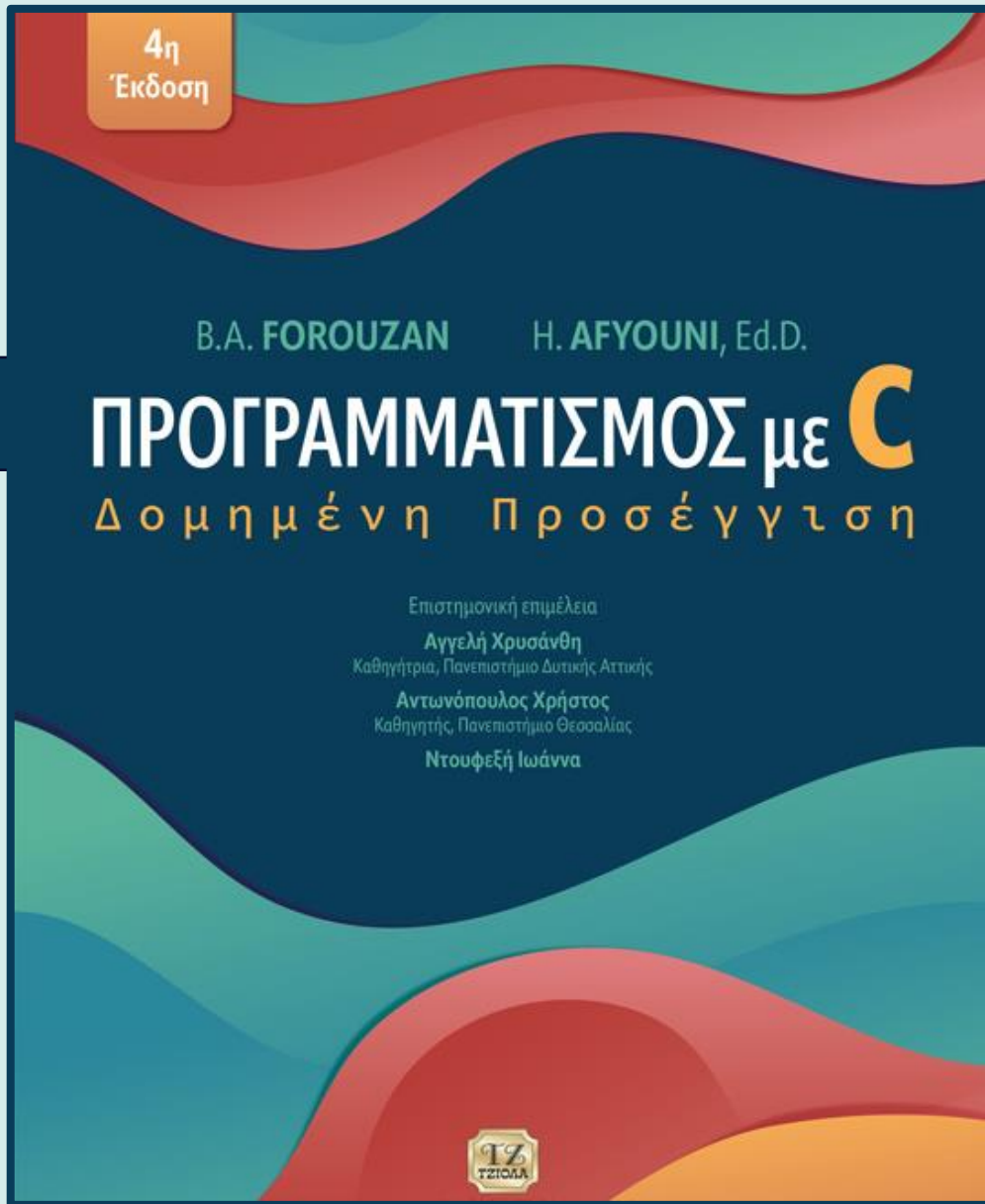




Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στους Υπολογιστές

Πρωτότυπο έργο: Computer Science: A Structured Programming Approach in C, 4th ed., H. Afyouni, B.A.Forouzan. Copyright 2000, 2007, 2023 Cengage Learning, Inc.
Αποκλειστικότητα για την ελληνική γλώσσα: Εκδόσεις TZIOΛA. Copyright 2024.
Με επιφύλαξη παντός νόμιμου δικαιώματος.

ΘΕΜΑΤΑ ΚΕΦΑΛΑΙΟΥ

Ολοκληρώνοντας αυτό το κεφάλαιο, θα είστε σε θέση:

- Να περιγράφετε τις βασικές έννοιες που χαρακτηρίζουν ένα σύστημα υπολογιστή
- Να αναγνωρίζετε/περιγράφετε τα διάφορα υπολογιστικά περιβάλλοντα και τα συστατικά τους
- Να αναφέρετε και να περιγράφετε τις κατηγορίες γλωσσών προγραμματισμού υπολογιστών
- Να προσδιορίζετε τα βήματα που απαιτούνται για την ανάπτυξη ενός προγράμματος υπολογιστή
- Να περιγράφετε τον κύκλο ζωής ενός συστήματος

1.1

Συστήματα υπολογιστών

Συστήματα υπολογιστών

- Σήμερα, τα συστήματα υπολογιστών βρίσκονται παντού
 - Αποτελούν ουσιαστικό μέρος της καθημερινής ζωής, διευκολύνοντας την επικοινωνία των χρηστών με την οικογένεια και τους φίλους τους, όπως επίσης και τη λειτουργία ολόκληρων κυβερνήσεων και επιχειρήσεων
- Ένας υπολογιστής είναι ένα σύστημα που αποτελείται από δύο βασικά συστατικά
 - Το **υλικό** (hardware)
 - Το **λογισμικό** (software)

Το υλικό των υπολογιστών (1/2)

- Το υλικό (hardware) ενός συστήματος υπολογιστή αποτελείται από πέντε μέρη:
 - Συσκευές εισόδου
 - Κεντρική μονάδα επεξεργασίας (CPU)
 - Κεντρική μνήμη
 - Συσκευές εξόδου
 - Περιφερειακή μνήμη (περιφερειακές συσκευές αποθήκευσης, όπως εσωτερικοί και εξωτερικοί σκληροί δίσκοι, μονάδες USB και εξωτερικές μονάδες ταινίας για λήψη αντιγράφων ασφαλείας)

Το υλικό των υπολογιστών (2/2)

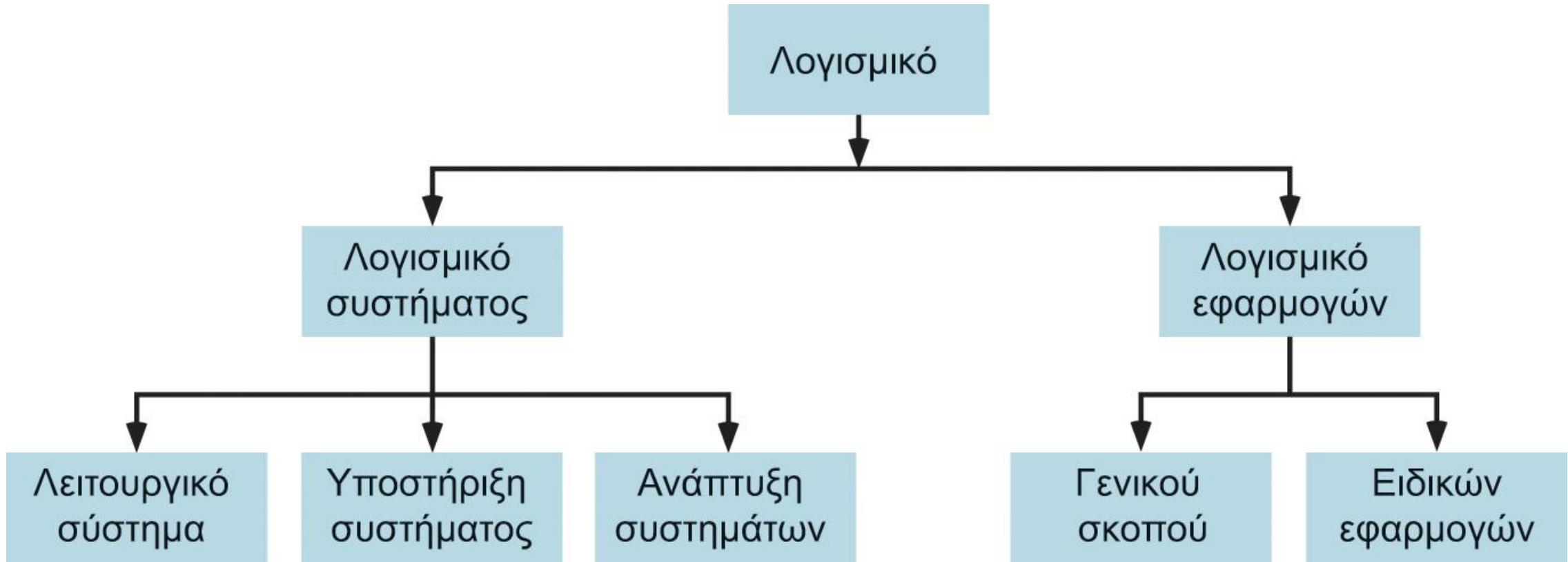


ΣΧΗΜΑ 1-2 Βασικά στοιχεία υλικού.

Το λογισμικό των υπολογιστών (1/2)

- Το λογισμικό (software) των υπολογιστών διακρίνεται σε δύο μεγάλες κατηγορίες:
 - Λογισμικό συστήματος
 - Λογισμικό εφαρμογών

Το λογισμικό των υπολογιστών (2/2)



ΣΧΗΜΑ 1-3 Τύποι λογισμικού.

Λογισμικό συστήματος

- Το λογισμικό συστήματος (system software) αποτελείται από προγράμματα που διαχειρίζονται τους πόρους υλικού ενός υπολογιστή και εκτελούν εργασίες επεξεργασίας δεδομένων
- Τα προγράμματα αυτά χωρίζονται σε τρεις κατηγορίες:
 - Λειτουργικό σύστημα
 - Εργαλεία υποστήριξης συστήματος
 - Εργαλεία ανάπτυξης συστήματος

Λογισμικό εφαρμογών

- Μπορούμε να διακρίνουμε το λογισμικό εφαρμογών (application software) σε δύο ευρείες κατηγορίες: λογισμικό γενικής χρήσης, ή «γενικού σκοπού», και εξειδικευμένο λογισμικό εφαρμογών
 - Το λογισμικό γενικής χρήσης (general-purpose software) αγοράζεται από έναν κατασκευαστή λογισμικού (ή προγραμματιστή) και μπορεί να χρησιμοποιηθεί για περισσότερες από μία εφαρμογές
 - Το εξειδικευμένο λογισμικό εφαρμογών (application-specific software) μπορεί να χρησιμοποιηθεί μόνο για τον σκοπό για τον οποίο προορίζεται

1.2 Υπολογιστικά περιβάλλοντα

Υπολογιστικά περιβάλλοντα

- Στις πρώτες μέρες των υπολογιστών, υπήρχε μόνο ένα περιβάλλον: ο κεντρικός (mainframe) υπολογιστής, κρυμμένος σε ένα κεντρικό τμήμα μηχανογράφησης
- Με την έλευση των μινι-υπολογιστών και των προσωπικών υπολογιστών, το περιβάλλον άλλαξε, με αποτέλεσμα να υπάρχουν υπολογιστές σχεδόν σε κάθε επιφάνεια εργασίας

Περιβάλλοντα προσωπικών υπολογιστών

- Ο πρώτος υπολογιστής-σε-ένα-ολοκληρωμένο κύκλωμα ήταν ο Intel 4004 και αποτέλεσε τον πρόγονο των μικροεπεξεργαστών που χρησιμοποιούνται σήμερα σε όλους τους υπολογιστές
 - Η ραγδαία ανάπτυξη της τεχνολογίας των ολοκληρωμένων κυκλωμάτων είχε ως αποτέλεσμα την παρουσίαση ολοένα και ισχυρότερων μικροεπεξεργαστών και οδήγησε, κατά τη δεκαετία του 1970, στη μετάβαση από τους μεγάλους mainframe υπολογιστές σε μία μικρότερη, αυτόνομη υπολογιστική μονάδα, η οποία είναι γνωστή σε όλους ως προσωπικός υπολογιστής ή PC (personal computer)

Περιβάλλοντα προσωπικών υπολογιστών (2/2)

- Δύο βασικοί τύποι προσωπικών υπολογιστών, βάσει λειτουργικού συστήματος
 - Microsoft DOS και, μεταγενέστερα, τα Microsoft Windows
 - υπολογιστές που κατασκευάζονταν αποκλειστικά από την εταιρεία Apple
- Σήμερα, οι προσωπικές υπολογιστικές συσκευές υπάρχουν σε διάφορες μορφές, συμπεριλαμβανομένων των έξυπνων τηλεφώνων, των ταμπλετών, των φορητών υπολογιστών, των επιτραπέζιων υπολογιστών που διαθέτουν μία οθόνη και ένα «κουτί» με το υλικό του συστήματος, και των επιτραπέζιων υπολογιστών «όλα σε ένα», οι οποίοι συνδυάζουν τη CPU και την οθόνη σε μία συσκευή

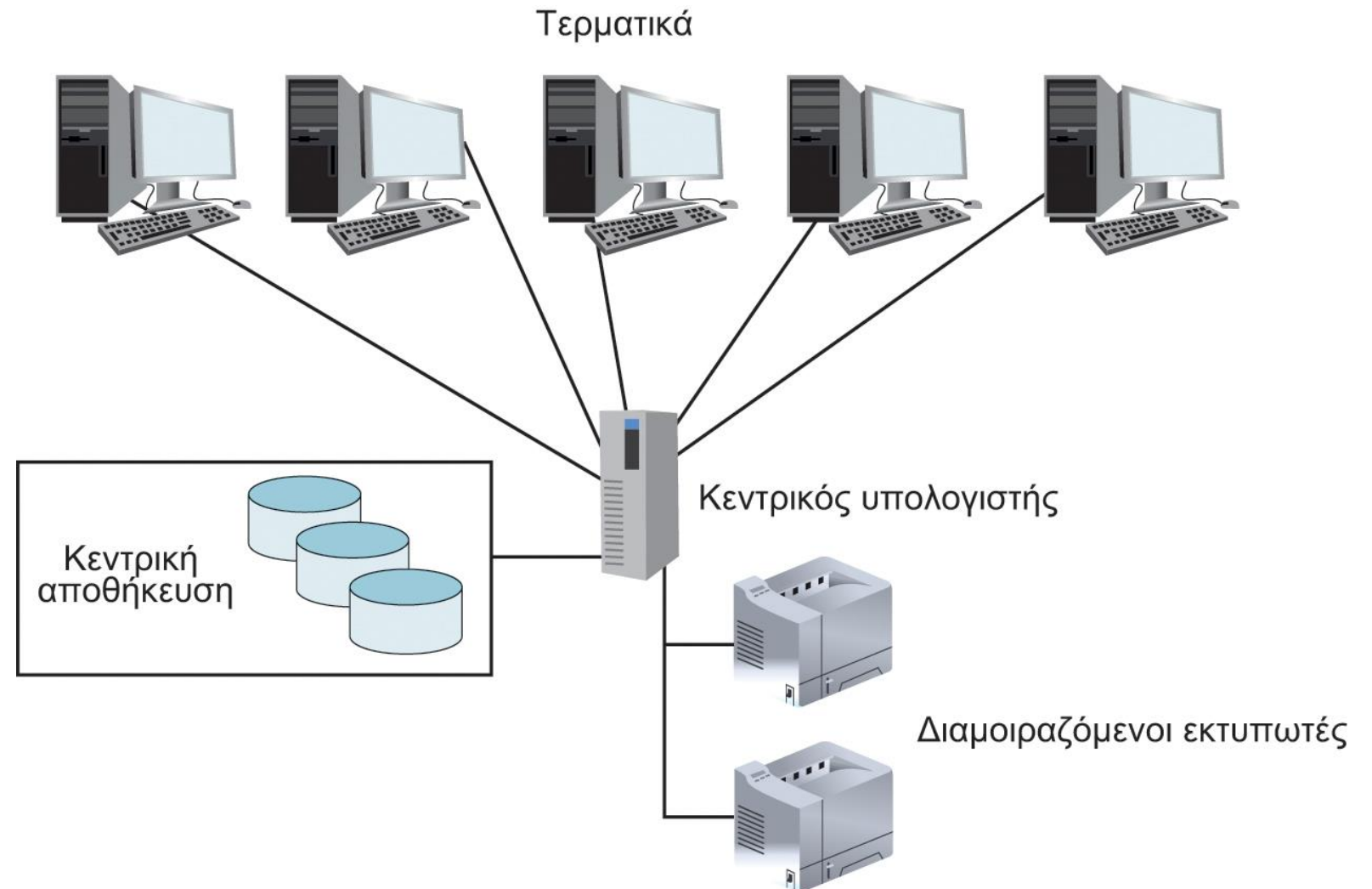
Περιβάλλοντα διαμοιρασμού χρόνου (1/2)

- Σε ένα υπολογιστικό περιβάλλον διαμοιρασμού χρόνου, πολλοί χρήστες συνδέονται, μέσω τερματικών, σε έναν ή περισσότερους κεντρικούς υπολογιστές—minicomputers ή mainframes
 - Όλοι οι υπολογισμοί πρέπει να γίνονται από τον κεντρικό υπολογιστή

Περιβάλλοντα διαμοιρασμού χρόνου (2/2)

ΣΧΗΜΑ 1-6

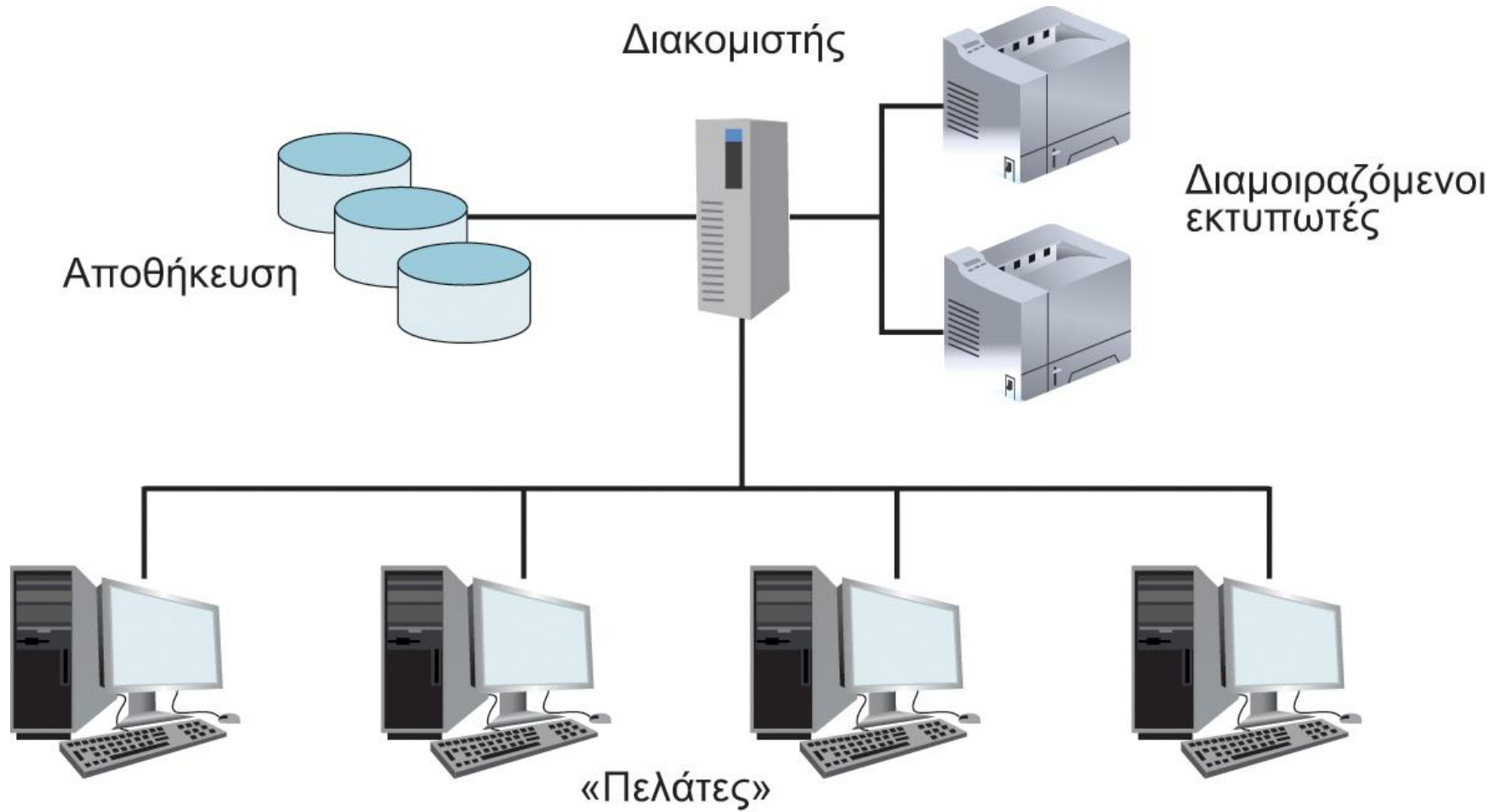
Υπολογιστικό περιβάλλον
διαμοιρασμού χρόνου.



Περιβάλλοντα πελάτη/διακομιστή (1/2)

- Ένα περιβάλλον πελάτη/διακομιστή (client/server) επιμερίζει τον φόρτο της υπολογιστικής εργασίας μεταξύ ενός κεντρικού υπολογιστή και των υπολογιστών των χρηστών
 - Οι σταθμοί εργασίας των χρηστών αποκαλούνται πελάτες (clients)
 - Ο κεντρικός υπολογιστής, ο οποίος μπορεί να είναι ένα μεγάλο υπολογιστικό σύστημα με ισχυρές δυνατότητες, αποκαλείται διακομιστής (server)

Περιβάλλοντα πελάτη/διακομιστή (2/2)

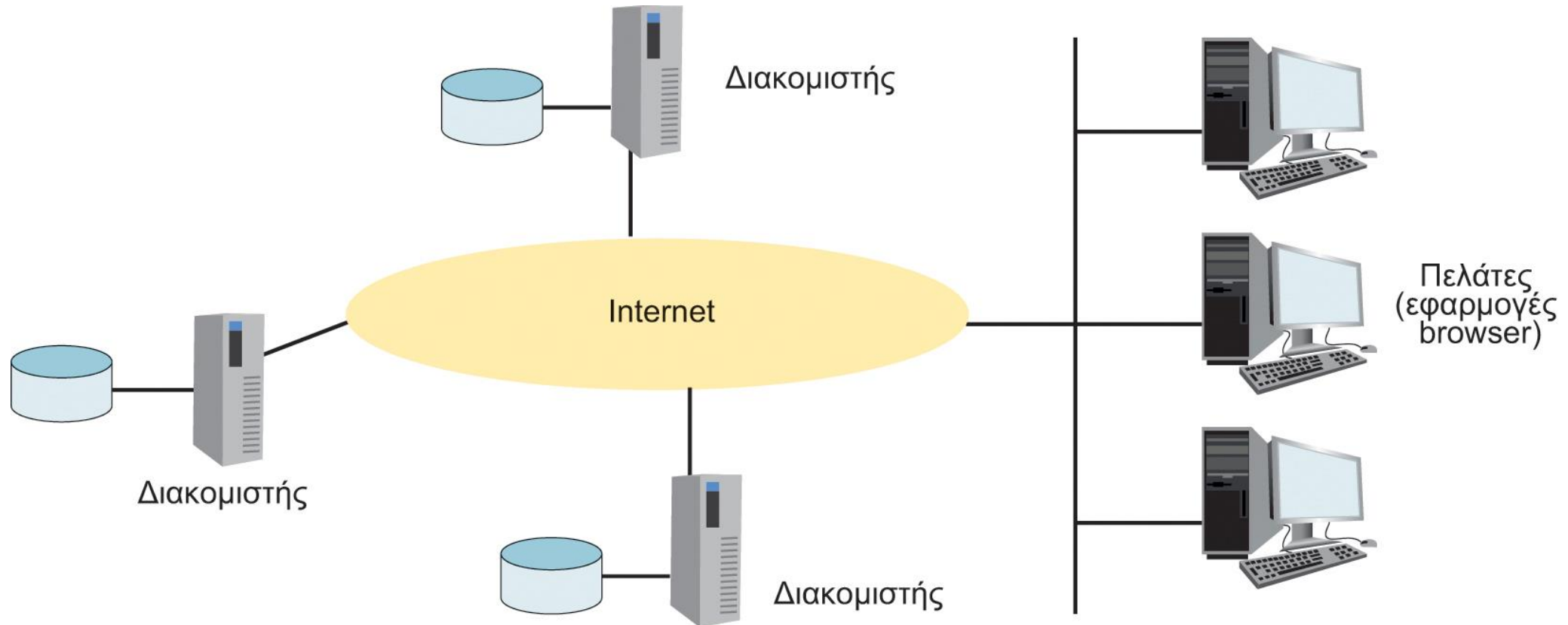


ΣΧΗΜΑ 1-7 Υπολογιστικό περιβάλλον πελάτη/διακομιστή.

Κατανεμημένα υπολογιστικά περιβάλλοντα (1/2)

- Μεγάλα δίκτυα πολλαπλών υπολογιστών—διακομιστών και πελατών—που είναι διασκορπισμένοι γεωγραφικά και διασυνδέονται μέσω τεχνολογιών επικοινωνίας ώστε να παρέχουν απρόσκοπτη ενοποίηση των υπολογιστικών λειτουργιών
 - Όλοι οι πόροι των διακομιστών, όπως η CPU και η μνήμη, συγκεντρώνονται σε μία «δεξαμενή υπολογιστικών πόρων», προκειμένου να παρέχουν υψηλή επεξεργαστική ισχύ για τις εφαρμογές
 - Αυτό το περιβάλλον παρέχει δυνατότητες που υποστηρίζουν αξιόπιστη λειτουργία δικτύων υψηλής διαθεσιμότητας με δυνατότητες επέκτασης σε μεγάλη κλίμακα

Κατανεμημένα υπολογιστικά περιβάλλοντα (2/2)



ΣΧΗΜΑ 1-8 Κατανεμημένο υπολογιστικό περιβάλλον.

Υπολογιστική νέφους

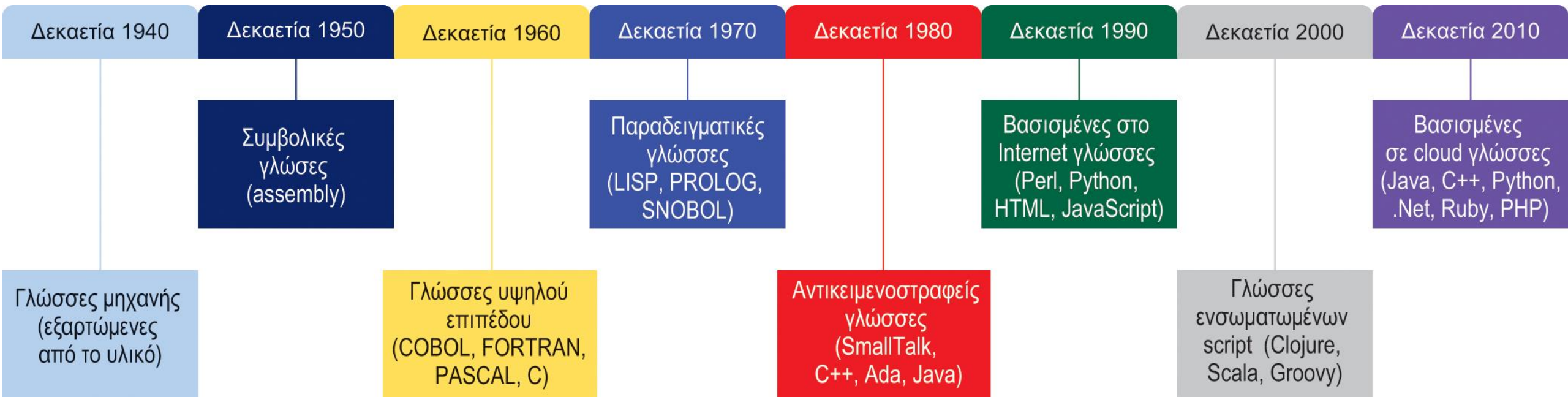
- Σε ένα περιβάλλον υπολογιστικής νέφους, οι διακομιστές και οι συσκευές αποθήκευσης κατανέμονται σε πολλές γεωγραφικές περιοχές και συνδέονται μέσω του Internet
- Τα περιβάλλοντα υπολογιστικής νέφους παρέχουν προηγμένες υπηρεσίες τεχνολογίας σε εκατομμύρια χρήστες—όπως, ενδεικτικά, λογισμικό ως υπηρεσία (SaaS), πλατφόρμα ως υπηρεσία (PaaS) και υποδομή ως υπηρεσία (IaaS)

1.3 Γλώσσες υπολογιστών

Γλώσσες υπολογιστών

- Για να γράψετε ένα πρόγραμμα για έναν υπολογιστή, πρέπει να χρησιμοποιήσετε μία γλώσσα υπολογιστή
- Στην πάροδο του χρόνου, οι γλώσσες υπολογιστών εξελίχθηκαν αξιολογώντας, από γλώσσες μηχανής σε γλώσσες υψηλού επιπέδου

Γλώσσες υπολογιστών



ΣΧΗΜΑ 1-10 Ιστορική εξέλιξη των γλωσσών υπολογιστών.

Γλώσσες μηχανής

- Στις πρώτες μέρες των υπολογιστών, οι μόνες διαθέσιμες γλώσσες προγραμματισμού ήταν οι γλώσσες μηχανής
 - Κάθε τύπος υπολογιστή έχει τη δική του **γλώσσα μηχανής** (machine language), η οποία κάνει τα πάντα χρησιμοποιώντας απλώς και μόνο ροές δυαδικών ψηφίων, 0 και 1

Γλώσσες μηχανής

ΠΡΟΓΡΑΜΜΑ 1-1 | Πρόγραμμα πολλαπλασιασμού σε γλώσσα μηχανής

1	00000000	00000100	0000000000000000
2	01011110	00001100	11000010 0000000000000010
3	11101111	00010110	00000000000000101
4	11101111	10011110	00000000000001011
5	11111000	10101101	11011111 0000000000010010
6	01100010	11011111	00000000000010101
7	11101111	00000010	11111011 0000000000010111
8	11110100	10101101	11011111 0000000000011110
9	00000011	10100010	11011111 0000000000100001
10	11101111	00000010	11111011 0000000000100100
11	01111110	11110100	10101101
12	11111000	10101110	11000101 0000000000101011
13	00000110	10100010	11111011 0000000000110001
14	11101111	00000010	11111011 0000000000110100
15	01010000	11010100	00000000000111011
16	00000100	00000000000111101	

Συμβολικές γλώσσες

- Οι πρώιμες γλώσσες προγραμματισμού απλώς προσομοίωναν τις γλώσσες μηχανής χρησιμοποιώντας σύμβολα ή μνημονικά ονόματα για την αναπαράσταση των εντολών της γλώσσας μηχανής.
- Επειδή χρησιμοποιούσαν σύμβολα, οι γλώσσες αυτές αποκαλούνταν **συμβολικές γλώσσες** (symbolic languages)
 - Το Πρόγραμμα 1-2 εκτελεί τον ίδιο πολλαπλασιασμό με το Πρόγραμμα 1-1, αλλά είναι διατυπωμένο σε συμβολική γλώσσα

Συμβολικές γλώσσες

ΠΡΟΓΡΑΜΜΑ 1-2 | Το πρόγραμμα πολλαπλασιασμού σε συμβολική γλώσσα

```
1      entry    main,<r2>
2      subl2    #12, sp
3      jsb      C$MAIN_ARGS
4      movab    $CHAR_STRING_CON
5      pushal   -8(fp)
6      pushal   (r2)
7      calls    #2,SCANF
8      pushal   -12(fp)
9      pushal   3(r2)
10     calls    #2,SCANF
11     mull3     -8(fp),-12(fp),-
12     pushal   6(r2)
13     calls    #2,PRINTF
14     clrl     r0
15     ret
```

Γλώσσες υψηλού επιπέδου

- Η ανάγκη (και επιθυμία) να βελτιωθεί η αποδοτικότητα των προγραμματιστών και να μεταφερθεί η εστίαση από τον υπολογιστή στο προς επίλυση πρόβλημα, οδήγησε στην ανάπτυξη των **γλωσσών υψηλού επιπέδου** (high-level languages)
- Οι γλώσσες υψηλού επιπέδου μοιράζονται ένα κοινό χαρακτηριστικό με τις συμβολικές γλώσσες: πρέπει να μετατραπούν σε γλώσσα μηχανής
 - Η διαδικασία μετατροπής τους αναφέρεται ως **μεταγλώττιση** (compilation)
 - Το Πρόγραμμα 1-3 εκτελεί τον ίδιο πολλαπλασιασμό, αλλά είναι διατυπωμένο στη γλώσσα προγραμματισμού C

Γλώσσες υψηλού επιπέδου

ΠΡΟΓΡΑΜΜΑ 1-3 | Το πρόγραμμα πολλαπλασιασμού σε C

```
1  /* Αυτό το πρόγραμμα διαβάζει δύο ακέραιους αριθμούς από το πληκτρολόγιο
2     και εκτυπώνει το γινόμενο τους.
3     Δημιουργός:
4     Ημερομηνία:
5  */
6  #include <stdio.h>
7
8  int main (void)
9  {
10 // Τοπικοί ορισμοί
11     int number1;
12     int number2;
13     int result;
14
15 // Εντολές
16     scanf ("%d", &number1);
17     scanf ("%d", &number2);
18     result = number1 * number2;
19     printf ("%d", result);
20     return 0;
21 } // τέλος main
```



1.4 Δημιουργία και εκτέλεση προγραμμάτων

Δημιουργία και εκτέλεση προγραμμάτων

- Είναι ευθύνη του προγραμματιστή να γράψει και να ελέγξει το πρόγραμμα.
- Υπάρχουν τέσσερα βήματα σε αυτή τη διαδικασία:
 - **Σύνταξη** του προγράμματος (συγγραφή και επεξεργασία)
 - **Μεταγλώττιση** του προγράμματος
 - **Διασύνδεση** του προγράμματος με τις απαιτούμενες βιβλιοθήκες
 - **Εκτέλεση** του προγράμματος

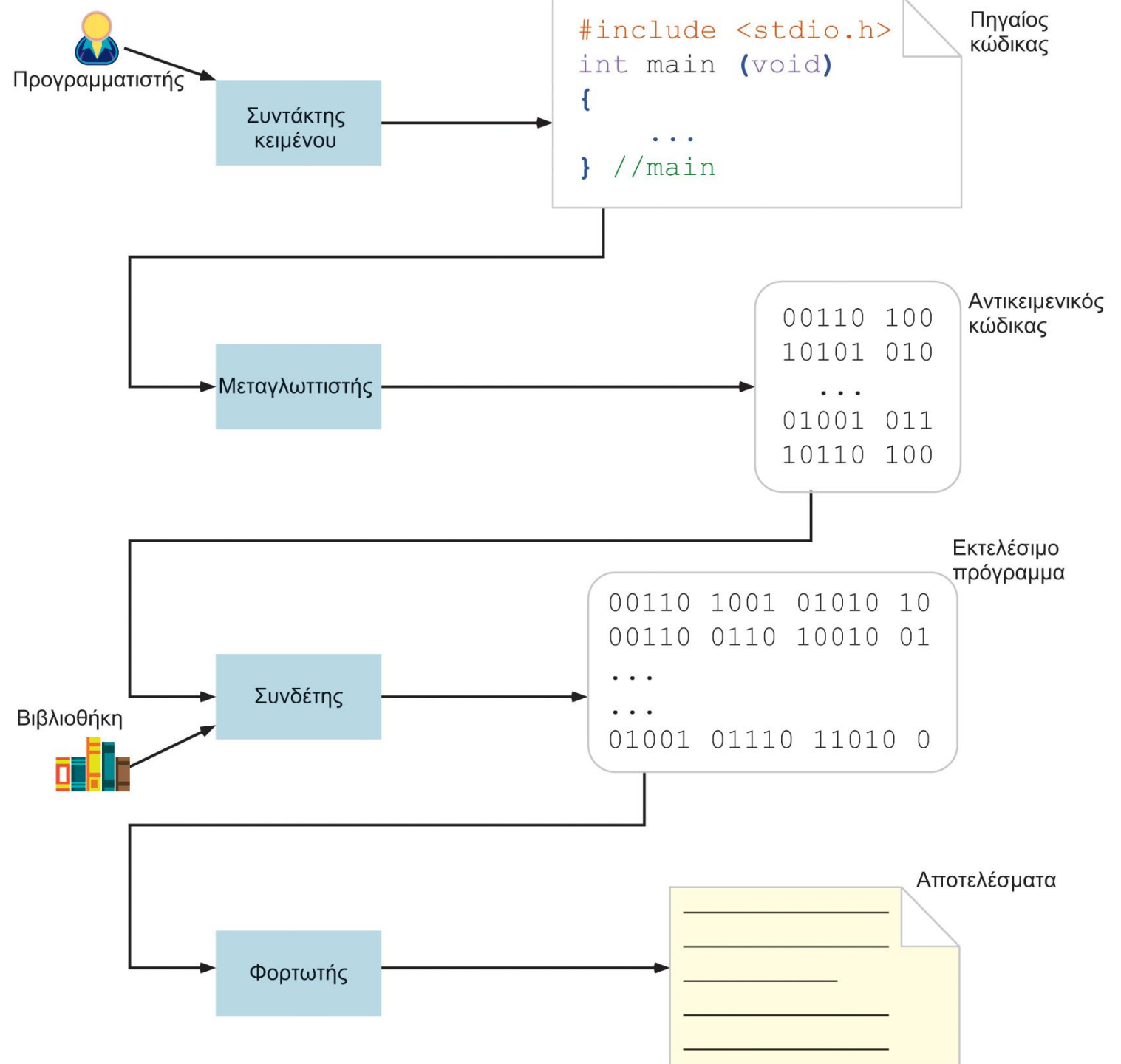
Σύνταξη προγραμμάτων

- Το λογισμικό που χρησιμοποιείται για τη συγγραφή του κώδικα των προγραμμάτων αναφέρεται ως **συντάκτης κειμένου** (text editor)
 - Σε έναν συντάκτη κειμένου, μπορείτε να εισάγετε, να αλλάζετε και να αποθηκεύετε δεδομένα χαρακτήρων
 - Διαθέτουν χαρακτηριστικά και δυνατότητες που διευκολύνουν τη συγγραφή κώδικα, συμπεριλαμβανομένης της εφαρμογής χρωμάτων σε συγκεκριμένα μέρη ενός προγράμματος
- Αφού ολοκληρωθεί η συγγραφή ενός προγράμματος, πρέπει να αποθηκευτεί σε αρχείο το οποίο εν συνεχεία εισάγεται στον μεταγλωττιστή
 - Αυτό το αρχείο αναφέρεται ως **αρχείο πηγαίου κώδικα** (source file)

Μεταγλώττιση προγραμμάτων (1/2)

- Ο **μεταγλωττιστής** (compiler) μεταφράζει ένα αρχείο πηγαίου κώδικα σε γλώσσα μηχανής προκειμένου να μετατραπεί σε εκτελέσιμο κώδικα
- Ο μεταγλωττιστής της C αποτελείται από δύο ξεχωριστά προγράμματα
 - Ο **προεπεξεργαστής** (preprocessor) διαβάζει τον πηγαίο κώδικα και τον προετοιμάζει για τον μεταφραστή
 - Το αποτέλεσμα της προεπεξεργασίας ονομάζεται **μονάδα μετάφρασης** (translation unit)
 - Ο **μεταφραστής** (translator) διαβάζει τη μονάδα μετάφρασης και παράγει μία **μονάδα αντικειμενικού κώδικα** (object module)—ένα αρχείο που μπορεί στη συνέχεια να συνδυαστεί με άλλες προμεταγλωττισμένες μονάδες προκειμένου να συγκροτηθεί το τελικό πρόγραμμα

Μεταγλώττιση προγραμμάτων (2/2)



ΣΧΗΜΑ 1-11
Δημιουργία ενός
προγράμματος C.

Διασύνδεση προγραμμάτων

- Καθώς γράφετε ένα πρόγραμμα C, ενδεχομένως να χρειαστεί να γράψετε συναρτήσεις για εκτέλεση συγκεκριμένων εργασιών
- Ωστόσο, ορισμένες συναρτήσεις που χειρίζονται βασικές λειτουργίες, όπως η είσοδος/έξοδος δεδομένων και η εκτέλεση μαθηματικών υπολογισμών, περιέχονται σε ξεχωριστά αρχεία (βιβλιοθήκες) που πρέπει να προσαρτηθούν στο πρόγραμμα
- Ο συνδέτης (linker) ενσωματώνει όλες τις απαραίτητες συναρτήσεις από εξωτερικά αρχεία (δικά σας ή βιβλιοθήκες του συστήματος), δημιουργώντας το τελικό **εκτελέσιμο πρόγραμμα** (executable program)

Εκτέλεση προγραμμάτων

- Αφού περάσει από τον συνδέτη, το πρόγραμμά σας είναι έτοιμο για εκτέλεση
- Για να εκτελέσετε ένα πρόγραμμα, χρησιμοποιείτε μία εντολή του λειτουργικού συστήματος, όπως η `run`, μέσω της οποίας το πρόγραμμα φορτώνεται στην κεντρική μνήμη
- Η μεταφορά του προγράμματός σας στη μνήμη είναι ευθύνη ενός προγράμματος του λειτουργικού συστήματος το οποίο ονομάζεται **φορτωτής** (loader)
 - Ο φορτωτής εντοπίζει το εκτελέσιμο πρόγραμμα και το διαβάζει στη μνήμη. Αφού φορτωθεί, το πρόγραμμα αναλαμβάνει τον έλεγχο και ξεκινά η λειτουργία του

1.5 Ανάπτυξη συστημάτων

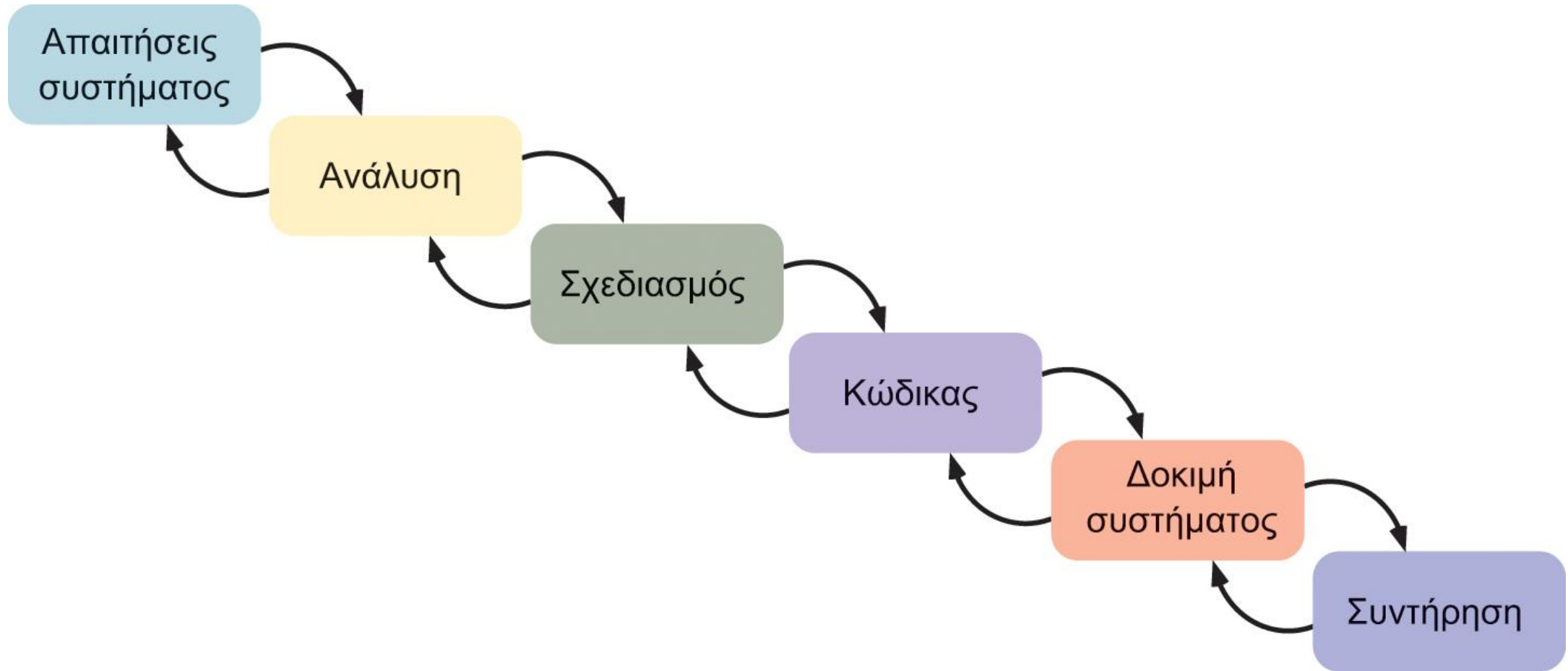
Ανάπτυξη συστημάτων

- Η διαδικασία ανάπτυξης ενός προγράμματος είναι μία πολύ σημαντική διαδικασία η οποία καθορίζει τη συνολική ποιότητα και επιτυχία του τελικού προγράμματος
- Ένα πρόγραμμα που σχεδιάζεται προσεκτικά, χρησιμοποιώντας καλά δομημένες τεχνικές ανάπτυξης, είναι
 - αποδοτικό ως προς τη λειτουργία του
 - απαλλαγμένο από λάθη
 - εύκολο στη συντήρηση

Ο κύκλος της ανάπτυξης συστημάτων (1/3)

- Τα σύγχρονα έργα προγραμματισμού εφαρμογών μεγάλης κλίμακας κατασκευάζονται βάσει μιας σειράς αλληλένδετων φάσεων
 - συνήθως αναφέρονται ως κύκλος της ανάπτυξης συστημάτων, ή κύκλος της ανάπτυξης λογισμικού
- Μία ιδιαίτερα δημοφιλή προσέγγιση είναι το μοντέλο καταρράκτη (waterfall model)

Ο κύκλος της ανάπτυξης συστημάτων (2/3)



ΣΧΗΜΑ 1-12 Μοντέλο καταρράκτη

Ο κύκλος της ανάπτυξης συστημάτων (3/3)

- Η μεθοδολογία **ευέλικτης ανάπτυξης** (agile development), υιοθετεί μία προσαρμοστική και επαναληπτική προσέγγιση στην ανάπτυξη λογισμικού
 - Στόχος η επιτάχυνση του κύκλου ζωής της ανάπτυξης και η συμμετοχή του χρήστη από νωρίς στη διαδικασία ανάπτυξης



Ανάπτυξη προγραμμάτων

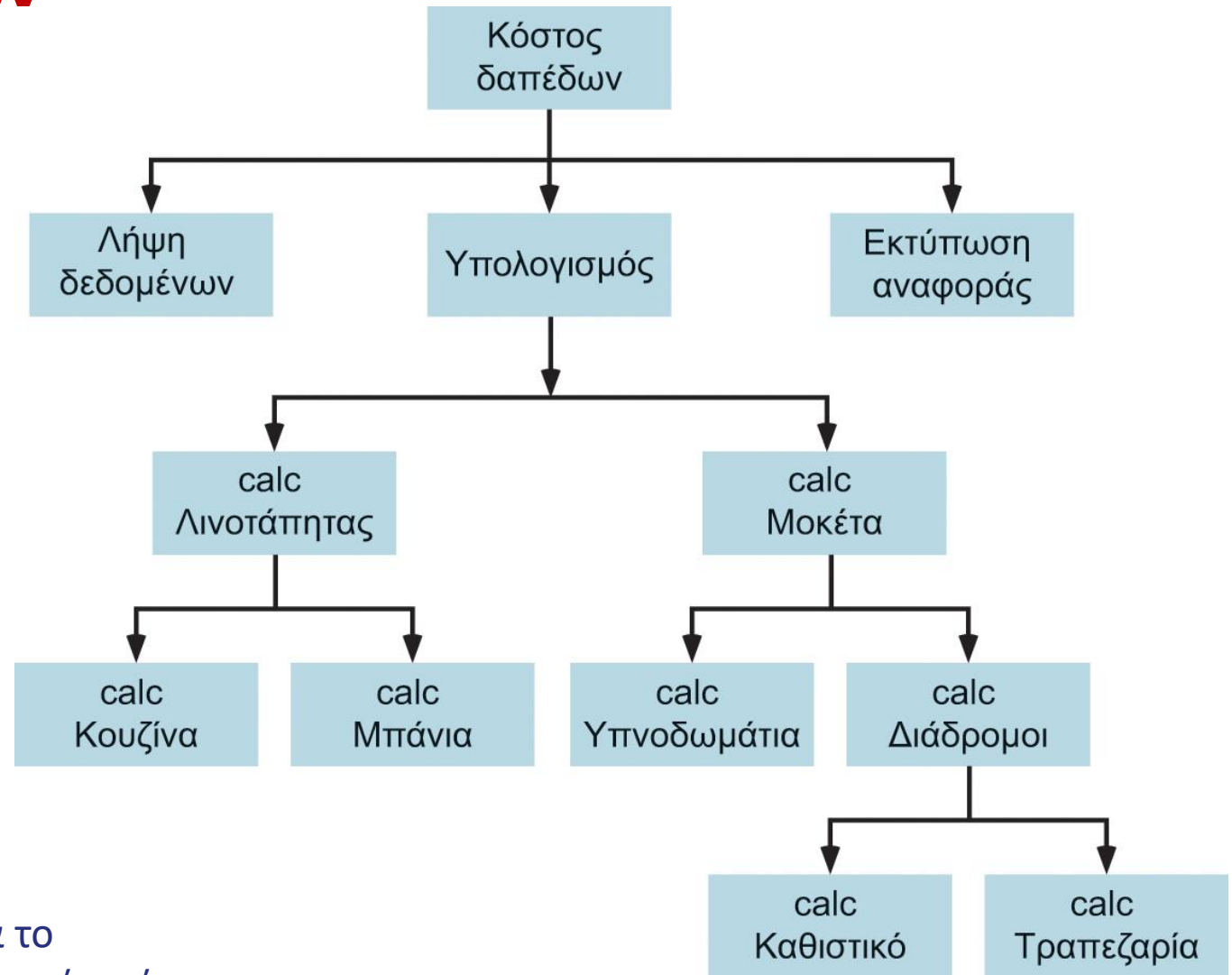
- Η ανάπτυξη ενός προγράμματος περιλαμβάνει τα εξής βασικά βήματα:
 - Κατανόηση του προβλήματος
 - Ξεκινήστε διαβάζοντας προσεκτικά τη δήλωση απαιτήσεων
 - Επανεξετάστε την εικόνα που σχηματίσατε για το πρόβλημα, συζητώντας το με τον χρήστη και με τον αναλυτή συστημάτων. Υποβάλλετε ερωτήσεις για να επιβεβαιώσετε ότι κατανοήσατε τα πάντα σωστά

Γνωμικό με μακρά ιστορία στον κόσμο του προγραμματισμού:
**Αντισταθείτε στον πειρασμό να ξεκινήσετε αμέσως το
γράψιμο κώδικα.**

Ανάπτυξη προγραμμάτων

- Ανάπτυξη της λύσης: τρία εργαλεία θα σας βοηθήσουν σε αυτό το έργο:
 - Το **διάγραμμα δομής** (structure chart): παρουσιάζει τη λειτουργική ροή του προγράμματος
 - Ο **ψευδοκώδικας** (pseudocode): σκοπός του είναι να περιγράψει, με επακριβή, αλγοριθμικού χαρακτήρα λεπτομέρεια, τι πρόκειται να κάνει το πρόγραμμα
 - Το **διάγραμμα ροής** (flowchart): ένα εργαλείο σχεδιαστικής αποτύπωσης της λογικής μιας μονάδας κώδικα (π.χ., συνάρτησης), το οποίο χρησιμοποιεί τυποποιημένα γραφικά σύμβολα για την αναπαράσταση της ροής των δεδομένων κατά την εκτέλεσή της

Ανάπτυξη προγραμμάτων



ΣΧΗΜΑ 1-14 Διάγραμμα δομής για το πρόγραμμα υπολογισμού τετραγωνικών μέτρων

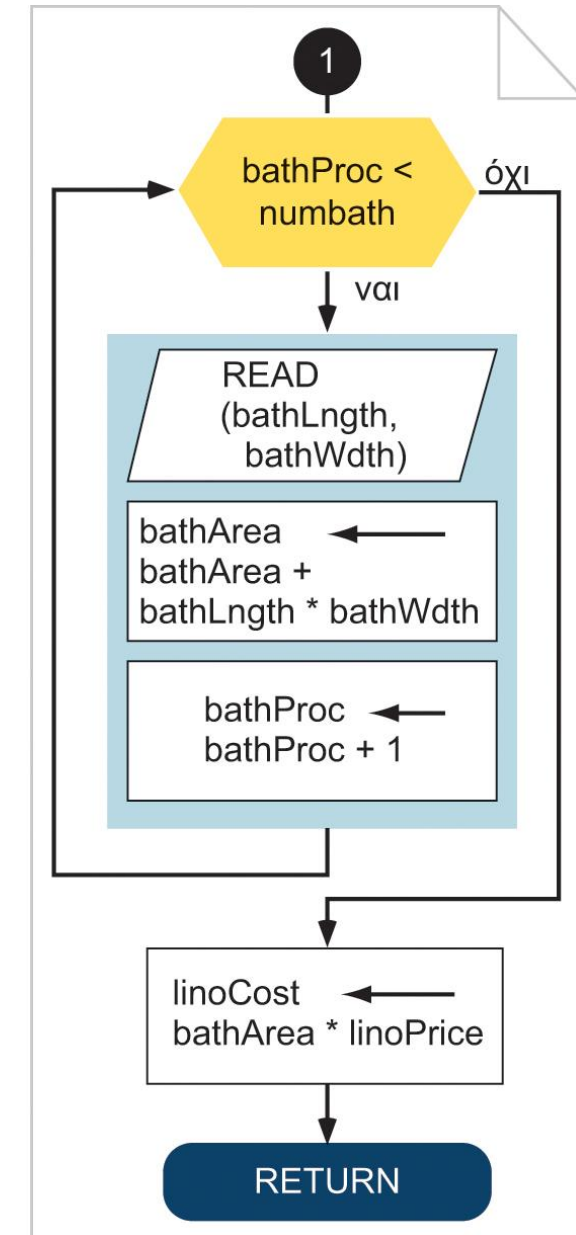
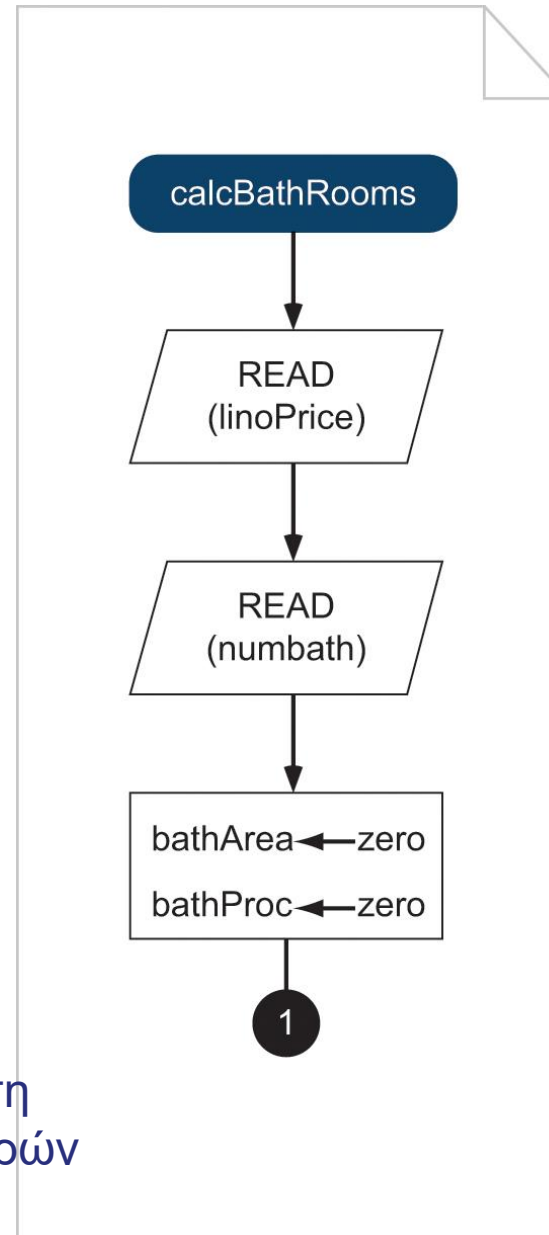
Ανάπτυξη προγραμμάτων

ΑΛΓΟΡΙΘΜΟΣ 1-1 Ψευδοκώδικας για το υποπρόγραμμα calcBathRooms

- 1 λήψη (ανάγνωση) τιμής λινόταπητα από τον χρήστη
- 2 λήψη (ανάγνωση) συνολικού αριθμού λουτρών από τον χρήστη
- 3 μηδενισμός μεταβλητών συνολικής επιφάνειας και αριθμού επεξεργασμένων λουτρών
- 4 while (αρ. επεξεργασμένων λουτρών < συνολικός αριθμός λουτρών)
 - 4.1 λήψη (ανάγνωση) μήκους και πλάτους λουτρού από τον χρήστη
 - 4.2 υπολογισμός: συνολική επιφάνεια λουτρού = μήκος λουτρού * πλάτος λουτρού
 - 4.3 πρόσθεση 1 στον αρ. επεξεργασμένων λουτρών
- 5 υπολογισμός: κόστος λουτρού = συνολική επιφάνεια λουτρού * τιμή λινόταπητα
- 6 επιστροφή κόστους κάλυψης λουτρού
- end αλγόριθμος calcBathRooms

Ανάπτυξη προγραμμάτων

ΣΧΗΜΑ 1-15 Διάγραμμα ροής για τη συνάρτηση υπολογισμού του κόστους κάλυψης των λουτρών



Ανάπτυξη προγραμμάτων

- Συγγραφή του προγράμματος
 - Όταν γράφετε ένα πρόγραμμα, ξεκινάτε με το κορυφαίο πλαίσιο του διαγράμματος δομής και προχωράτε προς τα κάτω.
 - Αυτό αναφέρεται ως υλοποίηση από πάνω προς τα κάτω (top-down implementation)

Ανάπτυξη προγραμμάτων

- Έλεγχος (δοκιμές για αξιολόγηση) του προγράμματος
 - Ο έλεγχος τύπου blackbox («μαύρου κουτιού») οφείλει το όνομά του στο γεγονός ότι το πρόγραμμα αντιμετωπίζεται ως ένα «μαύρο κουτί», υπό την έννοια ότι δεν γνωρίζουμε πώς λειτουργεί
 - πραγματοποιείται από τους μηχανικούς ελέγχου συστήματος και τους χρήστες
 - Ο έλεγχος τύπου whitebox («λευκού κουτιού») προϋποθέτει πλήρη γνώση του προγράμματος
 - είναι ευθύνη του προγραμματιστή

Σύνοψη

- Ένα σύστημα υπολογιστή αποτελείται από υλικό και λογισμικό
- Το υλικό του υπολογιστή αποτελείται από μία κεντρική μονάδα επεξεργασίας (CPU), κεντρική μνήμη, συσκευές εισόδου, συσκευές εξόδου και περιφερειακές μονάδες αποθήκευσης
- Το λογισμικό διακρίνεται σε δύο μεγάλες κατηγορίες: λογισμικό συστήματος και λογισμικό εφαρμογών
- Τα συστατικά στοιχεία του λογισμικού συστήματος είναι το λειτουργικό σύστημα, τα εργαλεία υποστήριξης του συστήματος και τα εργαλεία ανάπτυξης
- Το λογισμικό εφαρμογών διακρίνεται σε εφαρμογές γενικού σκοπού και σε λογισμικό ειδικών εφαρμογών
- Στην πάροδο του χρόνου, οι γλώσσες προγραμματισμού εξελίχθηκαν από τη γλώσσα μηχανής στις σύγχρονες γλώσσες υψηλού επιπέδου

Σύνοψη

- Η C είναι μία υψηλού-επιπέδου γλώσσα προγραμματισμού
- Το λογισμικό που χρησιμοποιείται για τη συγγραφή προγραμμάτων αναφέρεται ως συντάκτης κειμένου
- Το αρχείο που δημιουργείται από έναν συντάκτη κειμένου αναφέρεται ως αρχείο πηγαίου κώδικα
- Ο κώδικας σε ένα αρχείο πηγαίου κώδικα πρέπει να μεταφραστεί σε γλώσσα μηχανής χρησιμοποιώντας τον μεταγλωττιστή C, ο οποίος αποτελείται από δύο ξεχωριστά προγράμματα: τον προεπεξεργαστή και τον μεταφραστή

Σύνοψη

- Το αρχείο που δημιουργείται από τον μεταγλωττιστή αναφέρεται ως μονάδα αντικειμενικού κώδικα
- Μία μονάδα αντικειμενικού κώδικα συνδέεται με τις τυποποιημένες συναρτήσεις που είναι απαραίτητες για την εκτέλεση του προγράμματος από τον συνδέτη
- Το πρόγραμμα που παράγει ο συνδέτης εκτελείται με τη χρήση ενός φορτωτή
- Ο κύκλος της ανάπτυξης συστημάτων είναι μία σειρά αλληλένδετων βημάτων που παρέχουν μία αυστηρή και συστηματική προσέγγιση στην ανάπτυξη λογισμικού

Σύνοψη

- Η ανάπτυξη ενός πλάνου δοκιμών ξεκινά με τον σχεδιασμό του προγράμματος και συνεχίζεται σε όλα τα στάδια της ανάπτυξης του προγράμματος
- Οι δοκιμές τύπου blackbox («μαύρου κουτιού») βασίζονται κυρίως σε ελέγχους με βάση τις απαιτήσεις των χρηστών
- Ο έλεγχος τύπου whitebox («λευκού κουτιού») διενεργούνται από τον προγραμματιστή, ο οποίος ελέγχει το πρόγραμμα έχοντας πλήρη γνώση των λειτουργικών αδυναμιών του
- Ο έλεγχος είναι μία από τις σημαντικότερες πτυχές της δουλειάς σας ως προγραμματιστές. Εσείς είστε υπεύθυνοι για τον έλεγχο τύπου whitebox· οι αναλυτές συστημάτων και οι χρήστες είναι υπεύθυνοι για τον έλεγχο τύπου blackbox
- Η μηχανική λογισμικού συνίσταται στην εφαρμογή ορθών μεθόδων και αρχών μηχανικής κατά τον σχεδιασμό και την ανάπτυξη λογισμικού εφαρμογών



Το παρόν συνοδευτικό έργο **παρέχεται** αποκλειστικά και μόνο στους διδάσκοντες που έχουν επιλέξει το αντίστοιχο βιβλίο μέσω του συστήματος του **Ευδόξου** ως διδακτικό σύγγραμμα για τη διδασκαλία των μαθημάτων τους και την αξιολόγηση της εκμάθησης των φοιτητών και για όσο χρονικό διάστημα διατηρείται η επιλογή του συγκεκριμένου συγγράμματος. Η **διάδοση, δημοσίευση, αναπαραγωγή ή πώληση** οπουδήποτε μέρους αυτού του έργου με οποιοδήποτε μέσο καταστρέφει την ακεραιότητα του έργου **και για σκοπούς διαφορετικούς από αυτούς για τους οποίους έχει χορηγηθεί η σχετική άδεια δεν επιτρέπεται**. Το περιεχόμενο του έργου **δεν θα πρέπει να διατίθεται** στους φοιτητές παρά μόνο όταν αυτό αναφέρεται ρητά ότι μπορεί να γίνει. Η **χρήση** του παρόντος έργου γίνεται αποκλειστικά από τον διδάσκοντα στα πλαίσια των εκπαιδευτικών καθηκόντων του και με τη χρήση των μέσων που αυτός διαχειρίζεται και έχει στη διάθεσή του από τις **επίσημες υπηρεσίες του εκπαιδευτικού ιδρύματος** όπου ανήκει, διασφαλίζοντας τη μη περαιτέρω διάδοση, δημοσίευση και αναπαραγωγή του έργου προς τρίτους. Ο διδάσκων δύναται να **προσαρμόζει** μέρος του υλικού των διαφανειών σύμφωνα με τις διδακτικές του ανάγκες, αναφερόμενος όμως πάντοτε στην αρχική πηγή αναφοράς. Το παρόν έργο προστατεύεται από την ελληνική και ευρωπαϊκή νομοθεσία περί πνευματικής ιδιοκτησίας καθώς και από τη νομοθεσία του κράτους όπου ανήκει η ξενόγλωσση πρωτότυπη έκδοση του έργου.