ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Τεχνολογία και Προγραμματισμός Υπολογιστών

Ενότητα 1: Εισαγωγή-Γραμματική-Συντακτικό Γλώσσας-Δομή Προγράμματος

Διδάσκων

Χρήστος Μακρής

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Πανεπιστήμιο Πατρών

Εισαγωγή-Γραμματική-Συντακτικό Γλώσσας-Δομή Προγράμματος

(οι διαφάνειες είναι βασισμένες στο βιβλίο Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C KERNIGHAN W. BRIAN, RITCHIE M. DENNIS, εκδόσεις Κλειδάριθμος δεύτερη έκδοση, 2011)

Γλώσσες Προγραμματισμού

Αλγόριθμος: Διαδικασία επίλυσης προβλήματος.

Γλώσσα Προγραμματισμού: Γλώσσα διατύπωσης αλγορίθμων.

Πρόγραμμα: Αλγόριθμος διατυπωμένος σε κάποια γλώσσα προγραμματισμού.

Γλώσσα Προγραμματισμού

- Λεξιλόγιο (σύνολο εντολών)
- Συντακτικό (κανόνες σύνταξης προτάσεων)
- Σημασιολογία (μοντέλο υπολογισμού)

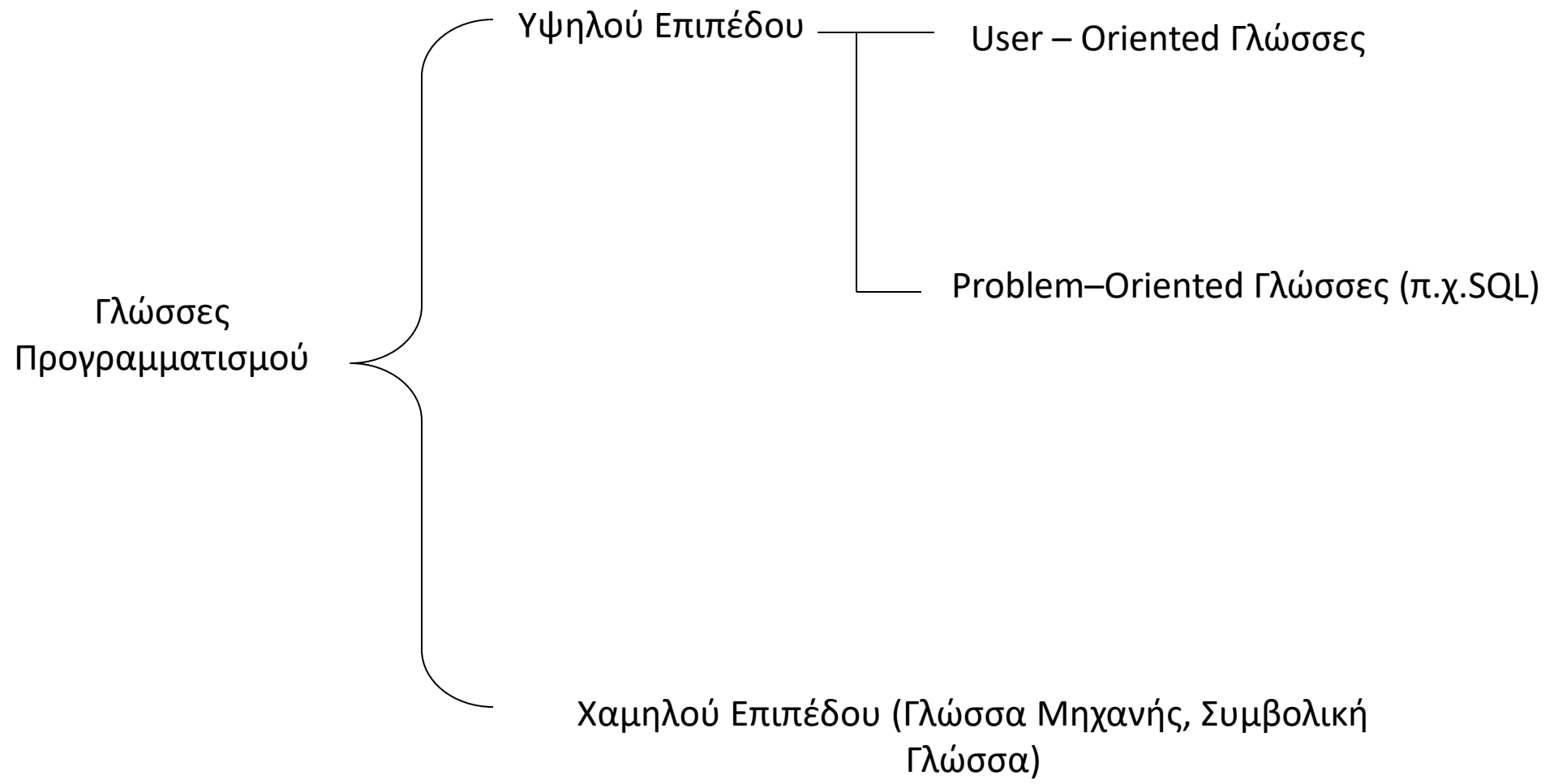
Χαρακτηριστικά Αλγορίθμου

- Αλγόριθμος (πεπερασμένος, καλά ορισμένος, είσοδος, έξοδος, αποτελεσματικός)
- Δομή Αλγορίθμου (κεφαλή (header), τμήμα δηλώσεων δεδομένων, κυρίως τμήμα αλγορίθμου)
- Τρόποι περιγραφής

  -- λεκτική (ψευδοκώδικας)

  -- διαγράμματα ροής προγράμματος

Είδη Γλωσσών Προγραμματισμού (1)



Είδη Γλωσσών Προγραμματισμού (2)

User Oriented Γλώσσες Προγραμματισμού

- ✓ Επιτακτικές ή Διαδικαστικές ή Προστακτικές Γλώσσες

  π.χ. FORTRAN, ALGOL,COBOL, BASIC, PL/1, C, ADA

  - Αντικειμενοστραφείς (C++, Java)
  - Οπτικές (Visual Basic)

- ✓ Εφαρμοστικές ή Δηλωτικές ή Περιγραφικές Γλώσσες

  - Συναρτησιακές (Lisp, Haskell)
  - Λογικές (Prolog)

Μοντέλα Εφαρμογών

Μοντέλα= διαμερίζουν την ανάπτυξη λογισμικού σε φάσεις και καθορίζουν:

- ✓τις δραστηριότητες που την απαρτίζουν
- ✓τα προϊόντα στα οποία καταλήγει
- ✓τις διαδικασίες επαλήθευσης των αποτελεσμάτων
- ✓τα κριτήρια ολοκλήρωσης της

Υπάρχουν αρκετά μοντέλα εφαρμογών, τα κυριότερα είναι:

- ✓το μοντέλο του καταρράκτη (waterfall model)
- ✓το μοντέλο του έλικα ή σπειροειδές μοντέλο (spiral model)

Μοντέλο Καταρράκτη

Η ανάπτυξη λογισμικού υποδιαιρείται στις φάσεις:

- ✓ανάλυσης απαιτήσεων
- ✓σχεδίασης
- ✓υλοποίησης
- ✓ολοκλήρωσης
- ✓λειτουργίας και συντήρησης

Κάθε εργασία υπόκειται σε διαδικασίες επικύρωσης (validation) και επαλήθευσης (verification) με στόχο:

- ✓ τη συνέπεια με τις απαιτήσεις του χρήστη
- ✓ τη συνοχή με το αποτέλεσμα των προηγουμένων φάσεων

Ανάλυση Απαιτήσεων

Η φάση της ανάλυσης είναι εκείνη στην οποία:

- ✓καθορίζονται οι στόχοι
- ✓διαμορφώνεται το πλάνο εργασίας
- ✓καταγράφονται και αναλύονται οι απαιτήσεις που υπάρχουν
- ✓υπολογίζεται το κόστος ανάπτυξης της εφαρμογής

Κύρια προϊόντα της φάσης αυτής είναι:

- ✓ ο καθορισμός των προδιαγραφών της εφαρμογής
- ✓ η δημιουργία ενός πλάνου εργασίας για τον τρόπο εκτέλεσης του έργου.

Σχεδίαση

Η φάση της *σχεδίασης* είναι εκείνη στην οποία βασιζόμενοι στα αποτελέσματα της προηγούμενης φάσης προβαίνουμε σε μια δομημένη λεπτομερή περιγραφή της εφαρμογής, τέτοια ώστε να επιτρέπει στην ομάδα ανάπτυξης να προχωρήσει στην υλοποίησή της.

Κύρια στάδια της φάσης αυτής είναι:

- ✓ Το στάδιο της *λειτουργικής σχεδίασης* όπου περιγράφεται η διεπαφή, οι λειτουργίες και το περιεχόμενο της εφαρμογής (εικονίδια, τρόπος πλοήγησης, χαρακτηριστικά περιεχομένου, κ.ά.)
- ✓ Το στάδιο της *τεχνικής σχεδίασης* όπου δημιουργείται ένα λεπτομερές σχέδιο της αρχιτεκτονικής της. Στο στάδιο αυτό λαμβάνονται αποφάσεις για την πλατφόρμα, το λογισμικό ανάπτυξης, τη μορφή των αρχείων κ.ά.

Υλοποίηση

Στη φάση υλοποίησης υλοποιούνται:

- ✓Η συγγραφή και η διόρθωση του κώδικα της εφαρμογής
- ✓Ο ποιοτικός έλεγχος της εφαρμογής, δηλαδή η διενέργεια κατάλληλων δοκιμών από την ομάδα ανάπτυξης και από επιλεγμένες ομάδες χρηστών.

Οι εργασίες γίνονται σε τέσσερα βασικά στάδια:

- ✓ Πρώτο Στάδιο (Έκδοση Άλφα): δημιουργία μιας πιλοτικής εφαρμογής με μικρό αριθμό αντιπροσωπευτικών δεδομένων.
- ✓ Δεύτερο Στάδιο (Έκδοση Βήτα): περιλαμβάνει την ολοκλήρωση της εφαρμογής.
- ✓ Τρίτο Στάδιο (Έκδοση Γάμμα): γίνονται δοκιμές και διορθώσεις προγραμματιστικών λαθών της εφαρμογής και η τελική επιμέλεια όλων των δεδομένων της.
- ✓ Τέταρτο Στάδιο (Χρυσή Έκδοση)

Ολοκλήρωση/Διανομή

Το στάδιο ολοκλήρωσης ασχολείται :

- ✓Με τρόπους προστασίας της εφαρμογής (νομική προστασία, προστασία από παράνομη αντιγραφή και χρήση)
- ✓Με τη δημιουργία προγράμματος εγκατάστασης – απεγκατάστασης (install – uninstall)
- ✓Με την προετοιμασία της συσκευασίας του προϊόντος
- ✓Με τη μαζική αναπαραγωγή του προϊόντος και του συνοδευτικού υλικού (εγχειρίδιο εγκατάστασης, συνοδευτικά φυλλάδια, κ.ά.).

Στο στάδιο διανομής οριστικοποιούνται οι στρατηγικές:

- ✓ Τιμολόγησης
- ✓ Προβολής και προώθησης προϊόντος στην αγορά.

Λειτουργία/Συντήρηση

Η Λειτουργία/Συντήρηση ασχολείται με :

- ✓Πραγματοποίηση αλλαγών στο σύστημα για να διορθωθούν λάθη που διαπιστώνονται κατά την πραγματική λειτουργία του και διέφυγαν κατά τη διαδικασία ανάπτυξης.
- ✓Πραγματοποίηση αλλαγών στο σύστημα με σκοπό την ενσωμάτωση νέων δυνατοτήτων
- ✓Αντιμετώπιση καθημερινών προβλημάτων

Πλεονεκτήματα/Μειονεκτήματα

- ✓ Το κύριο *πλεονέκτημα* του μοντέλου είναι η οργάνωση της διαδικασίας ανάπτυξης σε διακριτές φάσεις, που συμφωνούν με την ακολουθούμενη πρακτική ανάπτυξης εφαρμογών στην αγορά. Το συγκεκριμένο χαρακτηριστικό εξηγεί και τη μεγάλη δημοτικότητά του σε σχέση με κάθε άλλο μοντέλο ανάπτυξης λογισμικού που χρησιμοποιείται σήμερα.

- ✓ Το κύριο μειονέκτημα της χρήσης του συγκεκριμένου μοντέλου προκύπτει από την ανάγκη του ακριβούς καθορισμού προδιαγραφών του τελικού προϊόντος αρκετά νωρίς κατά τη διαδικασία ανάπτυξης και πιο συγκεκριμένα με την ολοκλήρωση του σταδίου της ανάλυσης.

Μοντέλο Έλικας

Το μοντέλο της έλικας υποστηρίζει μια εξελικτική διαδικασία δημιουργίας μιας εφαρμογής. Πιο συγκεκριμένα, η ανάπτυξη στο μοντέλο αυτό αποτελείται από την επαναληπτική εκτέλεση ενός κύκλου φάσεων. Κάθε φορά ο κύκλος παράγει μια ενδιάμεση έκδοση του τελικού προϊόντος η οποία βελτιώνεται κατά τον επόμενο κύκλο κ.ο.κ.

- ✓ Το κύριο *πλεονέκτημα* του μοντέλου είναι η δυνατότητα ελέγχου και αξιολόγησης από τους χρήστες κάθε ενδιάμεσου προϊόντος σε κάθε κύκλο ανάπτυξης. Με τον τρόπο αυτό διασφαλίζεται ότι το τελικό προϊόν θα ανταποκρίνεται όσο το δυνατόν καλύτερα στις απαιτήσεις των χρηστών.

- ✓ Το κύριο μειονέκτημα της χρήσης του συγκεκριμένου μοντέλου είναι οι αυξημένες απαιτήσεις σε χρόνο και πόρους που απαιτεί η υλοποίηση και αξιολόγηση μιας εφαρμογής σε διαδοχικούς κύκλους. Επίσης το συγκεκριμένο μοντέλο απαιτεί μια πολύπλοκη διαδικασία ανάπτυξης.

Πρακτικές Προγραμματισμού

- Κατά βήμα εκλέπτυνση (παραγωγή διαδοχικών «εκδόσεων» προγράμματος)

- Αρθρωτός προγραμματισμός (υλοποίηση διάσπασης προβλήματος σε απλούστερα)

Τεχνοτροπίες Προγραμματισμού

- Προγραμματισμός για επαναχρησιμοποίηση
- Προγραμματισμός με πλεονασμό (για ανοχή σε σφάλματα)
- Αμυντικός Προγραμματισμός

Δομημένη Υλοποίηση

1. Δομή Ακολουθίας Εντολών

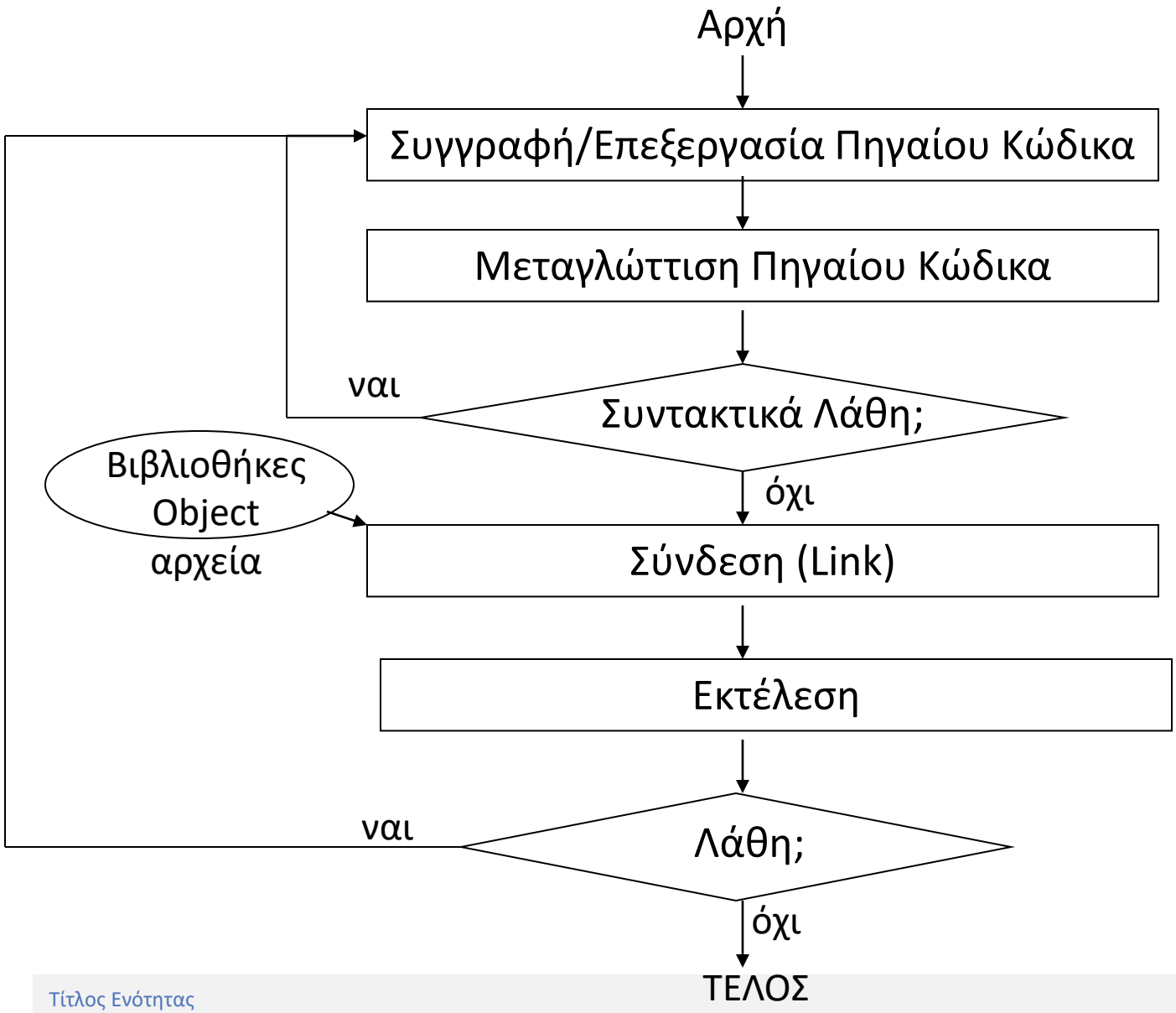
2. Δομή Απόφασης (επιλογή υπό συνθήκη)
3. Δομή Επανάληψης (επανάληψη υπό συνθήκη)

Απαραίτητη προϋπόθεση το να ισχύουν τα εξής:

-- σε κάθε δομή προγραμματισμού (και κατ' επέκταση στο πρόγραμμα) υπάρχει μόνο μία είσοδος (αρχή της ροής ελέγχου) και μία μόνο έξοδος (τέλος της ροής ελέγχου)

-- η ροή μεταξύ εισόδου και εξόδου σε κάθε δομή είναι ομαλή και απρόσκοπτη

Φάση Υλοποίησης



Ιστορία της C

1969-1973: AT&T Bell Labs από Dennis Ritchie ως συνέχεια των BCPL, B (ανάπτυξη νέας έκδοσης UNIX).

1978: "The C Programming Language" K&R: Kernigham & Ritchie.

1983: Σύσταση ANSI Standardization Commitee X3J11.

1989-1990: Αποδοχή ANSI/ISO Standard (ANSI C) C89

- ISO/IEC 9899:1990 Information Technology – Programming Language C.
- ISO/IEC 9899 AM1 – αποτελεί τροποποίηση του C πρωτοτύπου. Έγινε αποδεκτό το 1995.
- ISO/IEC 9899 TCOR1 – Technical Corrigendum 1. Έγινε αποδεκτό το 1995.
- ISO/IEC 9899 TCOR2 – Technical Corrigendum 2. Έγινε αποδεκτό το 1996.
- C99, C11, C17, C2x

Γιατί C;

- ✓ Είναι σχετικά μικρή και εύκολη στην εκμάθηση υποστηρίζοντας top down/modular σχεδιασμό και δομημένο προγραμματισμό
- ✓ Υπάρχει μεγάλη εγκατεστημένη βάση εφαρμογών που αναπτύχθηκαν με τη γλώσσα αυτή.
- ✓ Μπορεί να διαχειριστεί: δυαδικά ψηφία (bits), ψηφιολέξεις (bytes), συμβολοσειρές (words), δείκτες (pointers).
- ✓ Μπορεί να χρησιμοποιηθεί για χαμηλού επιπέδου προγραμματισμό επιτρέποντας άμεση πρόσβαση στους πόρους του υπολογιστή. Συνεπώς είναι κατάλληλη γλώσσα για ανάπτυξη προγραμμάτων συστήματος (systems programs) όπως είναι: λειτουργικά συστήματα (operating systems), διερμηνευτές (interpreters), συντάκτες (editors), συμβολομεταφραστές (assemblers), μεταγλωττιστές (compilers), διαχειριστές βάσεων δεδομένων (database managers).

Δομή Προγράμματος (1)

- Γλώσσα C
  - Εντολές προεπεξεργαστή
  - Δηλώσεις συναρτήσεων
  - Δηλώσεις μεταβλητών
  - Κυρίως πρόγραμμα →
  - Ορισμοί συναρτήσεων

```
main()
{
-δηλώσεις μεταβλητών
- προτάσεις γλώσσας
}
```

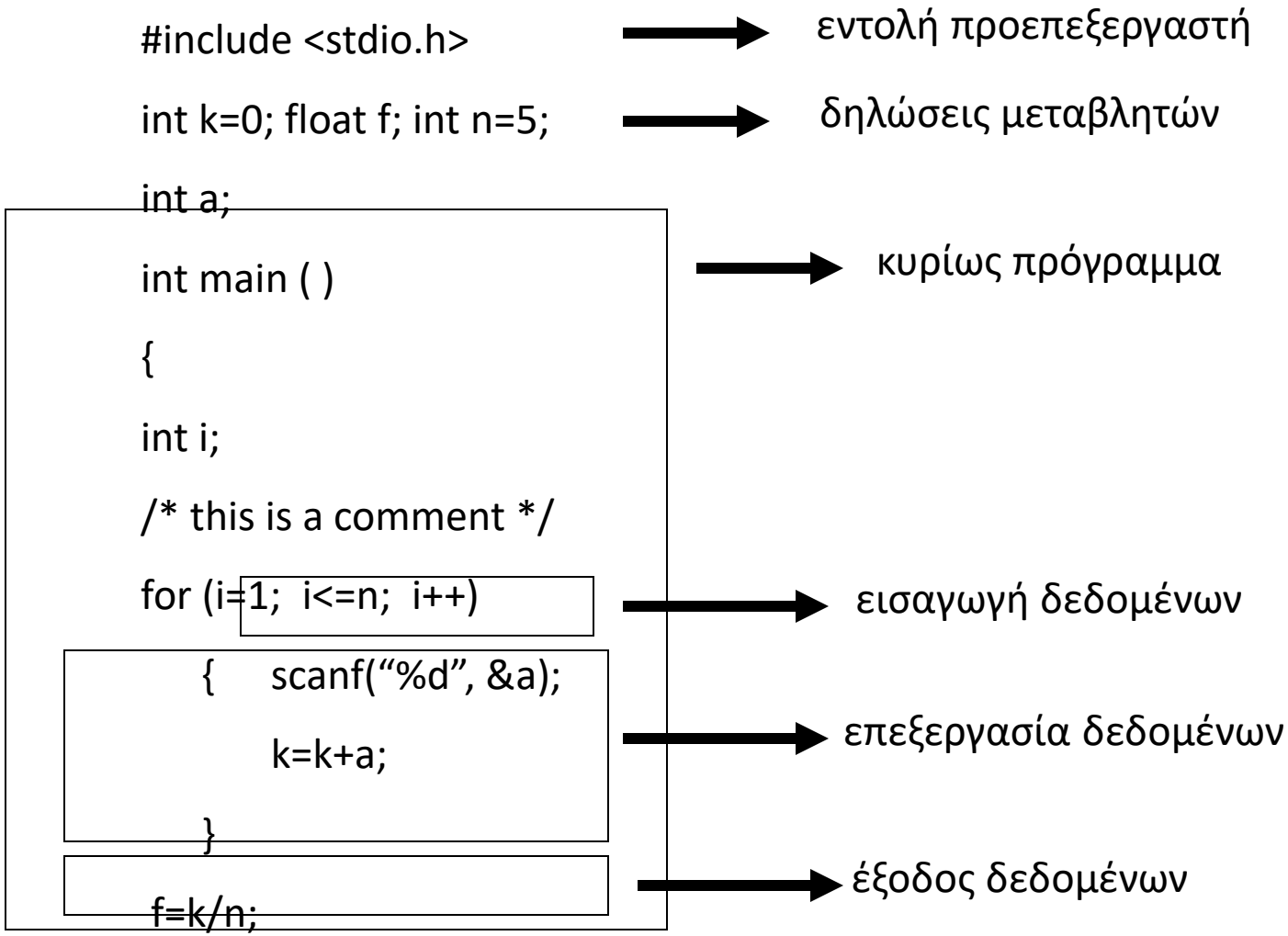
Δομή Προγράμματος (2)

- Κυρίως Πρόγραμμα

  - Είσοδος Δεδομένων
  - Επεξεργασία Δεδομένων
  - Έξοδος Αποτελεσμάτων

  } Προτάσεις Γλώσσας

Ένα Απλό Πρόγραμμα (test.c)



Βασικά Σημεία του Προγράμματος (1)

1. #include <stdio.h>

   Περιέχεται στην αρχή κάθε προγράμματος και περιλαμβάνει οδηγίες προς τον προεπεξεργαστή.

2. int main()

   Το σημείο έναρξης κάθε προγράμματος.

3. { }

   Δηλώνονται οι εντολές που είναι στη main και πρέπει να εκτελεστούν

Βασικά Σημεία του Προγράμματος (2)

4. printf(), scanf()

   Βασικές συναρτήσεις εισόδου/εξόδου της βασικής βιβλιοθήκης της C

5. \n

   Ακολουθία διαφυγής για την αναπαράσταση δύσκολων χαρακτήρων (π.χ. \t, \b, \\)

6. %f

   Δηλώνεται η φόρμα (format) εμφάνισης πραγματικών.

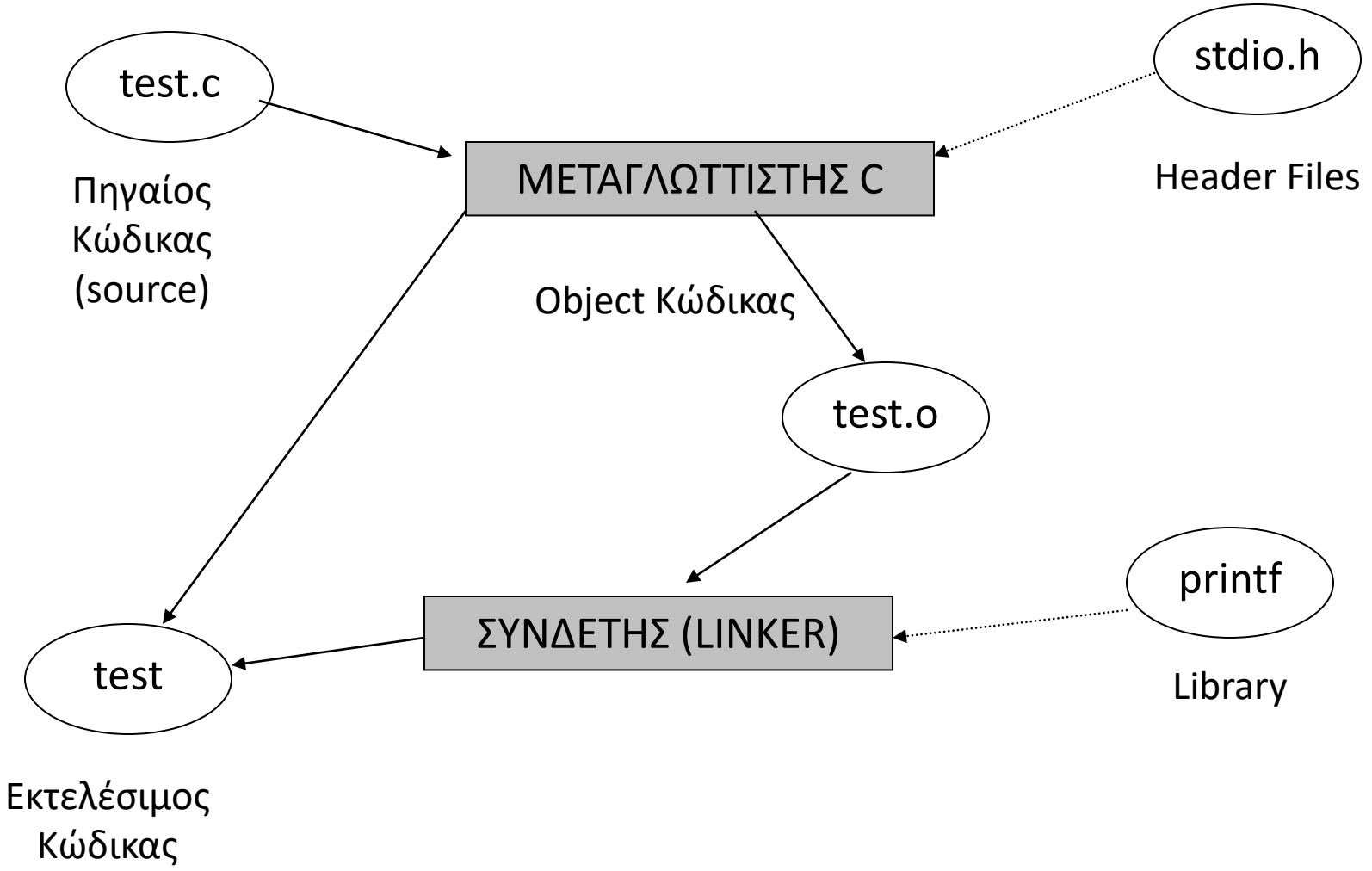
7. ;

   Όλες οι δηλώσεις των C-προγραμμάτων τελειώνουν σε ; (semicolon), που δηλώνει τέλος εντολής.

Χρήση Σχολίων σε C

```
#include <stdio.h>
main()
  {
        // This line is comments
        // This line also.
        // All those lines are NOT executed.
        /*
        Everything here is comments.
        we can place not needed lines of code here a=a+4;
        */
printf("This program  shows the use of comments \n");
   }
```

Διαδικασία Μεταγλώττισης



Στοιχεία Γλώσσας Προγραμματισμού

- Αλφάβητο
  - Οι χαρακτήρες από τους οποίους σχηματίζονται οι λέξεις της γλώσσας
- Λεξιλόγιο
  - Οι λέξεις που χρησιμοποιεί η γλώσσα
- Συντακτικό
  - Οι κανόνες σύνταξης των προτάσεων της γλώσσας
- Σημασιολογία
  - Οι κανόνες ερμηνείας των προτάσεων της γλώσσας

Αλφάβητο

- Αποτελείται από 96 χαρακτήρες

  -Χαρακτήρας κενού

  -Χαρακτήρας ελέγχου οριζοντίου στηλοθέτη (tab)
  -Χαρακτήρας ελέγχου καθέτου στηλοθέτη
  -Χαρακτήρας αλλαγής σελίδας (form feed)
  -Χαρακτήρας νέας γραμμής (new-line)

  a b c d e f g h i j k l m n o p q r s t u v w x y z
  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  0 1 2 3 4 5 6 7 8 9
  _ { } [ ] # ( ) < > % : ; . ? * + - / ^ & | ~ = ! , \ “ ‘

Λεξιλόγιο – Κατηγορίες Λέξεων

- Δεσμευμένες λέξεις (reserved words)
  - -Λέξεις κλειδιά (π.χ. *int*, *if*, *for*)
  - -Ονόματα συναρτήσεων βιβλιοθήκης (π.χ. *printf*(), *isdigit*())
  - -Ονόματα εντολών προεπεξεργαστή (π.χ. *#include*, *#define*)
- Αναγνωριστές (identifiers)
  - - Ονόματα μεταβλητών, σταθερών, συναρτήσεων, τύπων δεδομένων, τα οποία δίνονται από τον προγραμματιστή
- Τελεστές (operators)
  - - Ειδικά σύμβολα που παριστάνουν μία συγκεκριμένη στοιχειώδη διεργασία που εκτελείται σε δεδομένα

Κανόνες Δημιουργίας Αναγνωριστών (1)

- Αποτελούνται από γράμματα, αριθμούς και τον ειδικό χαρακτήρα “_”.
- Αρχίζουν πάντα από γράμμα ή “_”
  Δεν μπορεί να είναι ίδια με κάποια δεσμευμένη λέξη (C)
- Διάκριση μικρών και κεφαλαίων (case sensitive)
- Όριο μήκους (π.χ. 31, για ANSI C)

Παραδείγματα

~~2class~~ class1 ~~Class-1~~ Class_1 ~~Class#1~~ CLaSs1

Κανόνες Δημιουργίας Αναγνωριστών (2)

- Αποφύγετε ονόματα ενός χαρακτήρα όπως i, j ,x, y.
- Χρησιμοποιείστε εκφραστικά ονόματα.
- Χρησιμοποιείστε μικρά γράμματα, για ονόματα μεταβλητών. Κατά κανόνα χρησιμοποιούμε κεφαλαία γράμματα για μακροεντολές
- Διατηρείστε ενιαίο κανόνα όσον αφορά το διαχωριστικό για μεταβλητές που αποτελούνται από 2 ή περισσότερες λέξεις.

Αναγνωριστές - Αναγνωσιμότητα

```
x=20;
If ( x > y )
    function1( );
else
    function2( );
```

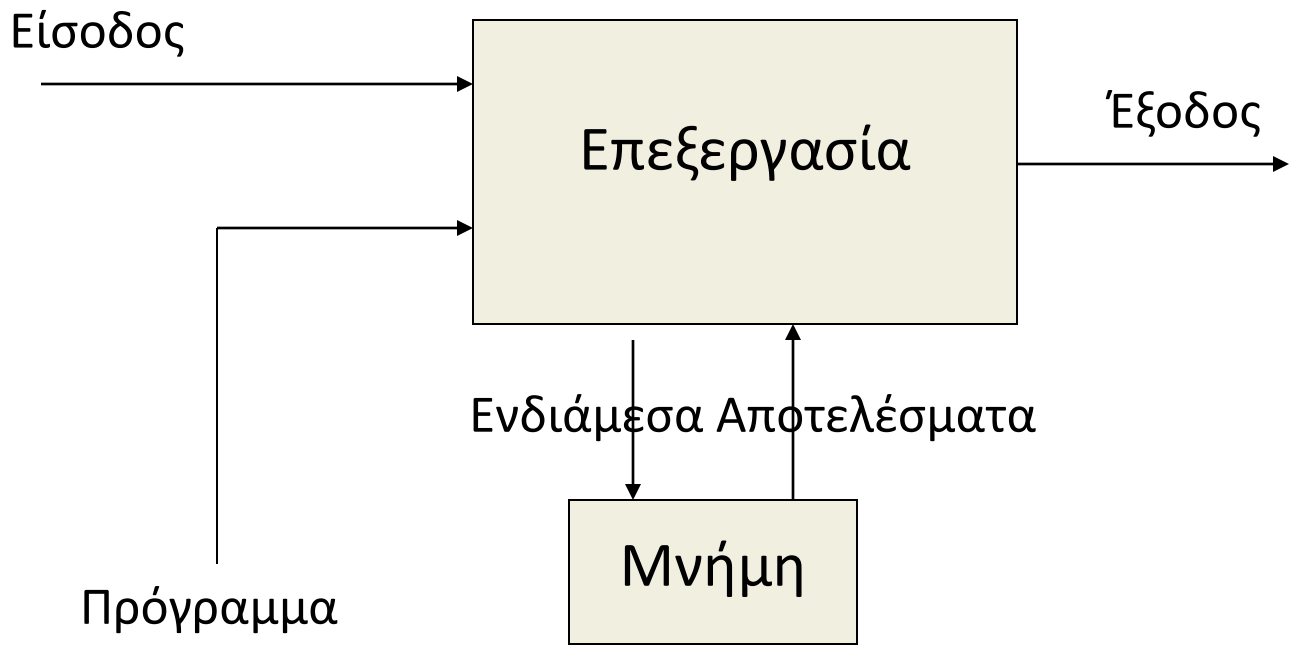
```
baros=20;
If ( baros > max_baros )
    ADD_baros( );
else
    DECREASE_baros( );
```

Δεδομένα

- Είναι η είσοδος, η ακατέργαστη γνωστή πληροφορία, με την επεξεργασία της οποίας θα παραχθούν τα αποτελέσματα (η λύση του προβλήματος)

- Δεδομένα=τιμές ή πλειάδες τιμών που σχετίζονται με τις οντότητες του προβλήματος.

Δεδομένα


Είσοδος
Επεξεργασία
Έξοδος
Ενδιάμεσα Αποτελέσματα
Μνήμη
Πρόγραμμα

Μεταβλητές (1)

- Είναι υπολογιστικές οντότητες μέσω των οποίων χρησιμοποιούνται τα δεδομένα.
- Το περιεχόμενο (τιμή) μίας μεταβλητής μπορεί να μεταβάλλεται κατά τη διάρκεια εκτέλεσης ενός προγράμματος
- Η χρήση των μεταβλητών πηγάζει από την ανάγκη για

  - αποθήκευση δεδομένων

  - ανάκληση αποθηκευμένων δεδομένων

Μεταβλητές (2)

- Οι μεταβλητές αποτελούν αναφορά
  - σε μία θέση μνήμης
  - στο περιεχόμενο μίας θέσης μνήμης
- Χαρακτηριστικά μεταβλητής
  - το όνομά της (θέση μνήμης)
  - η τιμή της (περιεχόμενο)
- Οι μεταβλητές παριστάνουν οντότητες του προβλήματος που επιλύει το πρόγραμμα

Π.χ. Πρόβλημα υπολογισμού της επιφάνειας ενός ορθογωνίου.
Οντότητες: μήκος, πλάτος, εμβαδόν
Μεταβλητές: *mikos*, *platos*, *embadon*

Σταθερές - Τιμές

- Σταθερές (constants) είναι υπολογιστικές οντότητες που έχουν σταθερή τιμή κατά την διάρκεια εκτέλεσης ενός προγράμματος (π.χ. PI για το 3.14)

- Τιμές (lirerals) είναι δεδομένα που δεν χρειάζονται αποθήκευση, χρησιμοποιούνται όπως είναι (π.χ. οι αριθμοί 2, 3.5)

Δήλωση Σταθερών

- #define <όνομα> <τιμή>

- ```
  #define    TRUE        1
  #define    FALSE       0
  #define    JANUARY     1
  #define    PI          3.1453
  ```
  ```

Σταθερά Απαρίθμησης

- enum boolean { NO, YES};
- enum boolean { TRUE, FALSE};
- enum months { JAN=1, FEB, MAR, APR, MAY, JUN,
  JUL, AUG, SEP, OCT, NOV, DEC}; 1

Τύποι Δεδομένων

- Τα δεδομένα δεν έχουν όλα τις ίδιες ιδιότητες, δεν είναι όλα του ίδιου τύπου.
- Τύπος δεδομένων = πεδίο τιμών + τρόποι διαχείρισης = πεδίο τιμών + πράξεις
- Τύποι δεδομένων
  - βαθμωτοί ή ατομικοί ή πρωτογενείς (*int*, *float*, *double*, *char*)
  - συναθροιστικοί ή σύνθετοι (πίνακες, δείκτες, ενώσεις, δομές)

  (- ενσωματωμένοι
  - Παραγόμενοι)

Πρωτογενείς Τύποι Δεδομένων

- Ακέραιος (int)
- Πραγματικός (float, double)
- Χαρακτήρας (char)
- Λογικός (η C δεν διαθέτει)

Γιατί τύποι δεδομένων;

- Οικονομία χώρου (λόγω διαφορετικής αποθήκευσης)

- Καλύτερος έλεγχος κατά τη μετάφραση και εκτέλεση του προγράμματος

Δήλωση Μεταβλητής

- <τύπος δεδομένων> <λίστα μεταβλητών>;

  ή

  <τύπος δεδομένων> <μεταβλητή>=<τιμή>;

  π.χ. float mikos, platos, embadon;

  int i=1;

- Μεταβλητές που δεν αλλάζουν τιμή (Σταθερές)

  π.χ. const float pi=3.14;

Βασικές Πράξεις

Τελεστής ανάθεσης: x=3;

Αριθμητικοί τελεστές (+, -, /, *, %)

x=5+3;

y=5*6;

Διαφορά στην ερμηνεία του τελεστή διαίρεσης μεταξύ ακεραίων και πραγματικών τελεστέων.

Τύπος «ακέραιος»

- Αναπαράσταση ακεραίων, θετικών ή αρνητικών αριθμών
- Λέξη κλειδί: *int*
- Εύρος: εξαρτάται από το μήκος λέξης του Η/Υ, π.χ. για λέξη 16-bit: -32768 έως 32767 ή 0 έως 65335
- Προσδιοριστές: short, long, unsigned
- Ο short έχει μήκος >=16 bit, ο long <=32 bit και ο int 16 ή 32 bits.
- Ακέραιες τιμές: τις χειρίζεται σαν ακεραίους αριθμούς ανάλογα με το μέγεθος, π.χ. την 125 την χειρίζεται σαν *int*, ενώ την 135840 σαν *long int* (μπορούμε όμως να επιβάλουμε τον τύπο χειρισμού, π.χ. 6524L)

Τύπος «πραγματικός»

- Αναπαράσταση πραγματικών αριθμών
- Εκφράσεις

  - Σταθερής υποδιαστολής: 120.52, 0.035

  - Εκθετικής ή επιστημονικής μορφής: 1.2052e+02, 3.5e-2

- Λέξεις κλειδιά

  - *float*: απλής ακρίβειας (6-7 δεκαδικά ψηφία)
  - *double*: διπλής ακρίβειας (14-15 δεκαδ. ψηφία)

- Προσδιοριστές: *long* για το double
- Πραγματικές τιμές: θεωρούνται σαν *double*

  π.χ. 0.13, 56.48, 8e-3, 15e05, 0.004e-0.4

Τύπος «χαρακτήρας»

- Αναπαράσταση απλών χαρακτήρων του αλφαβήτου της γλώσσας
- Λέξη κλειδί: char
- Η C χειρίζεται τους χαρακτήρες σαν ακεραίους. Κάθε χαρακτήρας θεωρείται σαν ακέραιος με τιμή τον αντίστοιχο κωδικό ASCII (στο δεκαδικό σύστημα, π.χ. 0 και 48)
- Οι τιμές σε μία τέτοια μεταβλητή αποδίδονται όταν ο χαρακτήρας εμπεριέχεται σε ′ ′. Για παράδειγμα:

  ′a′ είναι ο χαρακτήρας a

  ′b’ είναι ο χαρακτήρας b

  ‘c’ είναι ο χαρακτήρας c

Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την 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.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αδειοδότησης