Κεφάλαιο 6

Εισαγωγή στον Προγραμματισμό

6.1 Η έννοια του προγράμματος

Η επίλυση ενός προβλήματος με τον υπολογιστή περιλαμβάνει τρία στάδια

- Τον ακριβή προσδιορισμό του προβλήματος
- Την ανάπτυξη του αντίστοιχου αλγορίθμου
- Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή

Προγραμματισμός είναι δημιουργία προγραμμάτων δηλαδή, δηλαδή το σύνολο των εντολών που πρέπει να δοθούν στον υπολογιστή,, ώστε να υλοποιηθεί ο αλγόριθμος για τη επίλυση του προβλήματος

Φυσικές και τεχνητές γλώσσες

Αλφάβητο

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρστυφχψω
0123456789
.,-;!

Λεξιλόγιο

ΓΡΑΨΕ ≠ ΡΓΕΨΑ

Φυσικές και τεχνητές γλώσσες

Γραμματική

Τυπικό ή τυπολογικό (accidence): σύνολο κανόνων που ορίζει τις μορφές που μια λέξη είναι αποδεκτή

Συντακτικό (syntax): σύνολο κανόνων που ορίζει τη νομιμότητα της διάταξης και σύνδεση των λέξεων

Φυσικές και τεχνητές γλώσσες

Σημασιολογία (semantics): σύνολο κανόνων που ορίζει το νόημα των λέξεων και των εκφράσεων Διαφορές φυσικών και τεχνητών γλωσσών

ΦΥΣΙΚΕΣ & ΤΕΧΝΗΤΕΣ ΓΛΩΣΣΕΣ

ΟΙ ΔΙΑΦΟΡΕΣ ΤΟΥΣ

Φυσικές γλώσσες

- Χρησιμεύουν στην επικοινωνία μεταξύ ανθρώπων
- Εξελίσσονται συνεχώς (νέες λέξεις δημιουργούνται, αλλάζουν οι κανόνες σύνταξης ή γραμματικής, κλπ.)

- **Τεχνητές γλώσσες**
  - Χρησιμεύουν για την επικοινωνία ανθρώπου-μηχανής
  - Συνήθως στάσιμες αφού κατασκευάζονται για ένα συγκεκριμένο σκοπό
  - Βελτιώνονται σε μικρό σχετικά εύρος, από τους δημιουργούς τους για να ακολουθήσουν νέες εξελίξεις στα είδη εφαρμογών

6.4 ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

Είναι συνεχής η προσπάθεια εξεύρεσης μεθοδολογιών και τεχνικών ανάπτυξης λογισμικού

Κύριοι λόγοι είναι :

- Η απλοποίηση της οργάνωσης του κώδικα
- Η ευκολία διόρθωσης και γενικά η συντήρησή του από τους προγραμματιστές
- Η μείωση του χρόνου υλοποίησης των εφαρμογών

Ακολουθούν οι κυριότερες τεχνικές σχεδίασης

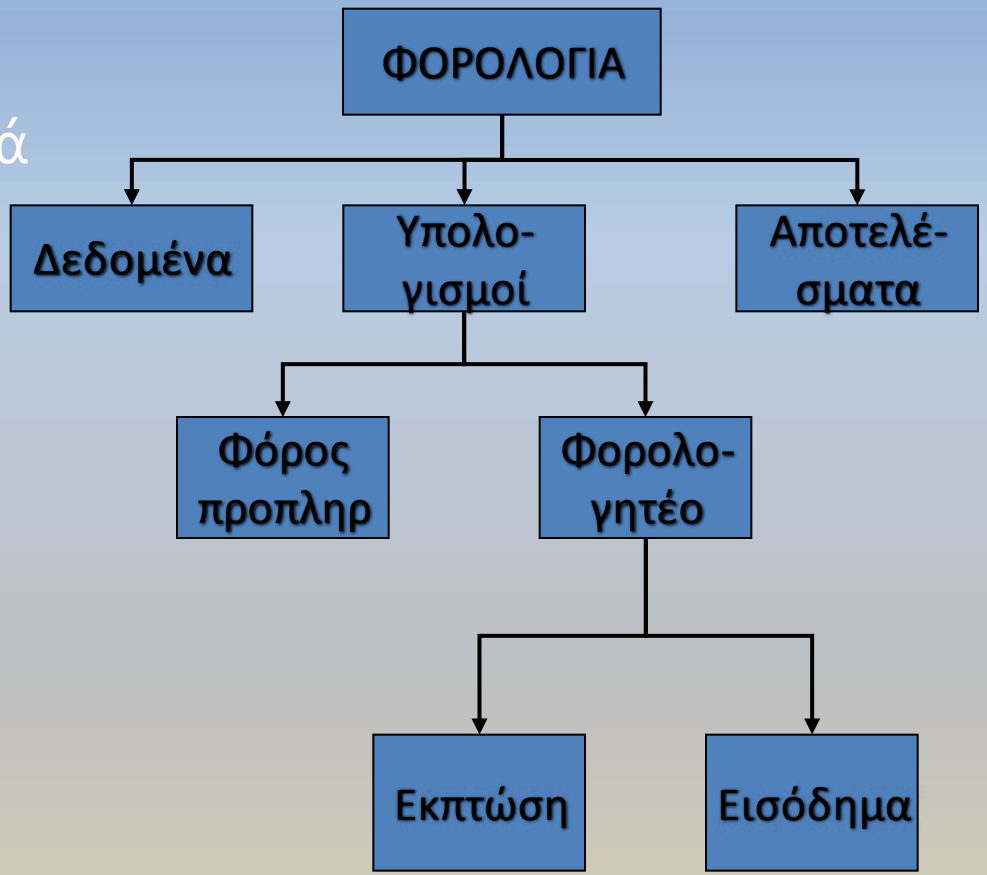
ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΙΕΡΑΡΧΙΚΗ ΣΧΕΔΙΑΣΗ

Διάσπαση του προβλήματος σε μικρότερα, απλούστερα και κατά το δυνατόν ανεξάρτητα μέρη

Λέγεται και προγραμματισμός «από επάνω προς τα κάτω» (top down program design)

Χρήση ειδικών διαγραμμάτων



ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Υλοποιεί την ιεραρχική σχεδίαση
- Κάθε υποπρόβλημα αποτελεί ξεχωριστή ενότητα (module) και γράφεται ανεξάρτητα
- Ο συνδυασμός των ενοτήτων συμβάλλει στην επίλυση του προβλήματος
- Χρειάζεται προσοχή για τη σωστή διάσπαση σε ενότητες
- Μειώνονται τα λάθη
- Καλύτερη παρακολούθηση και κατανόηση

ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Είναι ένας τρόπος οργάνωσης του κώδικα του προγράμματος

Παρουσιάστηκε από το 1960 και πλέον υποστηρίζεται σχεδόν από όλες τις γλώσσες προγραμματισμού

Θεμελιώθηκε το 1968, στη μελέτη του Edger Dijkstra με τίτλο : «**GOTO** Statement Considered Harmful»

Μέχρι τότε, δεν υπήρχε κάποια μεθοδολογία σχεδίασης, τα προγράμματα ήταν μπερδεμένα και η εντολή GOTO, που άλλαζε τη ροή εκτέλεσης, χρησιμοποιούνταν αλόγιστα (spagetti code)

Χρειάστηκαν αρκετά χρόνια για την πλήρη αποδοχή του

Στηρίζεται στις δομές ακολουθίας, επιλογής και επανάληψης και στο ότι κάθε υποενότητα πρέπει να έχει μια είσοδο και μια έξοδο

Περιλαμβάνει : ιεραρχική σχεδίαση και τμηματικό προγραμματισμό

ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

Παράδειγμα της «κακής» εντολή GO TO, που οδήγησε στο δομημένο προγραμματισμό

Αυτή η απόλυτη ελευθερία να μεταφέρεται η ροή του προγράμ-ματος σε διάφορα σημεία, παρήγαγε κώδικα με την επωνυμία «spaghetti code»

```
90   PRINT «Νέα συναλλαγή; (Κ/ατάθεση, Χ/ρέωση, Τ/έλος)"
100 T$=INPUT$(1)
110 IF T$ <> «Κ» AND T$<> «κ» GOTO 210
120    INPUT «Ποσό κατάθεσης:";DEPOSIT
130    BALANCE = BALANCE + DEPOSIT
140    PRINT USING «Νέο υπόλοιπο: $#####.##";BALANCE
150    GOTO 90
210 IF T$ <> «Χ» AND T$<> «χ» GOTO 300
220    INPUT «Ποσό χρέωσης";CHECK
230    BALANCE = BALANCE - CHECK
240    PRINT USING «Νέο υπόλοιπο: $#####.##";BALANCE
250    GOTO 90
300 IF T$ <> «Τ» AND T$<> «τ» GOTO 90
410 PRINT USING “Υπόλοιπο λογ/σμού: $#####.##";BALANCE
430 END
```

ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΔΟΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Απλούστερα προγράμματα

Άμεση μεταφορά των αλγορίθμων σε προγράμματα

Εύκολη ανάλυση του προγράμματος σε τμήματα

Περιορισμός λαθών

Εύκολη κατανόηση από τρίτους προγραμματιστές

Εύκολη διόρθωση και συντήρηση

Γρηγορότερη υλοποίηση των εφαρμογών

6.7 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

Για τη δημιουργία προγραμμάτων χρειάζονται τουλάχιστον τρία προγράμματα:

Ο συντάκτης (Editor)

Ο μεταγλωττιστής ή ο διερμηνευτής (Compiler ή Interpreter)

Ο συνδέτης (Linker)

Παρουσιάζονται αναλυτικότερα παρακάτω :

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

Ο συντάκτης (Editor)

Είναι ένας ανεξάρτητος επεξεργαστής απλού κειμένου.

Δεν είναι κάποια ειδική εφαρμογή. Ο προγραμματιστής μπορεί να επιλέξει οποιοδήποτε τρόπο συγγραφής, π.χ. το απλό "σημειωματάριο" του Η/Υ

Με τον καιρό, ενσωματώθηκε στα ολοκληρωμένα περιβάλλοντα προγραμματισμού

Απέκτησε λειτουργίες διευκόλυνσης συγγραφής κώδικα, π.χ. χρωματισμό εντολών, εύκολη διόρθωση λαθών, αυτόματη εισαγωγή ολόκληρων εντολών μετά από πληκτρολόγηση των αρχικών γραμμάτων τους, αντιγραφή, επικόλληση κλπ.

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ & ΔΙΕΡΜΗΝΕΥΤΕΣ

Ο μεταγλωττιστής και ο διερμηνευτής (Compiler και Interpreter)

Κάθε πρόγραμμα πρέπει να μετατραπεί σε μορφή αναγνωρίσιμη από τον Η/Υ, δηλ. σε γλώσσα μηχανής. Αυτό αναλαμβάνει ο **μεταγλωττιστής** και ο **διερμηνευτής**

Ο **μεταγλωττιστής**, δέχεται ως είσοδο όλο το αρχικό κείμενο (πηγαίος κώδικας - source code) του προγράμματος και παράγει το αντίστοιχο πρόγραμμα σε γλώσσα μηχανής (αντικείμενο πρόγραμμα - object)

Ο **διερμηνευτής** από την άλλη (π.χ. της ΓΛΩΣΣΑΣ), δέχεται μία προς μία τις εντολές του πηγαίου κώδικα, τις μεταφράζει σε γλώσσα μηχανής κάθε φορά (π.χ. στις επαναλήψεις) και τις εκτελεί. Έχουμε πιο αργή εκτέλεση, σε σχέση με τη μεταγλώττιση, η οποία γίνεται μία φορά για όλο τον κώδικα.

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ & ΔΙΕΡΜΗΝΕΥΤΕΣ

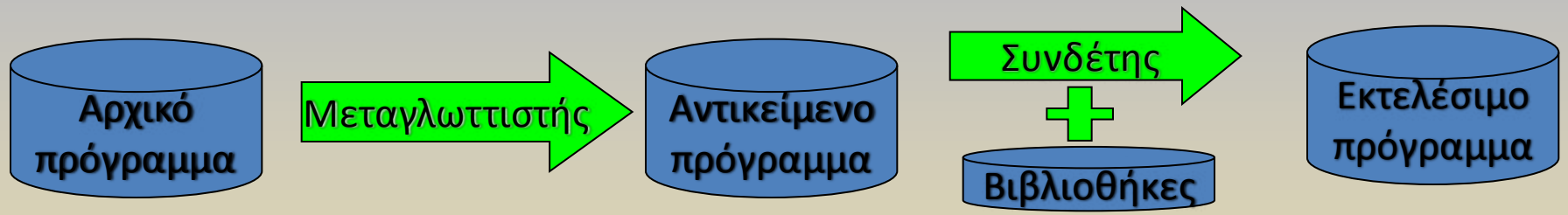
Ο συνδέτης (Linker)

Το αντικείμενο (object) πρόγραμμα που παράγει ο μεταγλωττιστής δεν μπορεί να εκτελεστεί αμέσως.

Χρειάζεται να συνδεθεί με άλλα τμήματα προγράμματος (βιβλιοθήκες – libraries) τα οποία, είτε έχει γράψει ο προγραμματιστής, είτε είναι μέρος της γλώσσας προγραμματισμού, ώστε να αξιοποιηθούν και χαρακτηριστικά του λειτουργικού συστήματος

Τη δουλειά αυτή αναλαμβάνει ο συνδέτης (linker), οποίος τελικά παράγει το τελικό εκτελέσιμο πρόγραμμα (executable)

Συνολικά η διαδικασία αποκαλείται «**μεταγλώττιση και σύνδεση**» (compiling and linking) ή «**χτίσιμο**» (build)



ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

Παλαιότερα, ο προγραμματιστής αναλάμβανε να εκτελέσει ξεχωριστά, κάθε ένα από τα προηγούμενα προγράμματα

Σήμερα, και τα τρία (συντάκτης, μεταγλωττιστής/διερμηνευτής και συνδέτης), είναι πλέον ενοποιημένα σε ειδικά ολοκληρωμένα περιβάλλοντα εργασίας, σε όλες τις γλώσσες προγραμματισμού

Χρησιμοποιείται γραφικό περιβάλλον εργασίας (Graphical User Interface – GUI)

Εύκολη συγγραφή κώδικα, άμεση μεταγλώττιση σε γλώσσα μηχανής και διασύνδεση με τις βιβλιοθήκες και το λειτουργικό σύστημα

Άμεση εμφάνιση συντακτικών λαθών

Βοηθητικές λειτουργίες, π.χ. λήψη εφεδρικών αντιγράφων, εκτυπώσεις, αλλαγή στην εμφάνιση και στη λειτουργία κατά τις προτιμήσεις του προγραμματιστή, εισαγωγή κειμένου από άλλα προγράμματα, κ.ά.

ΧΕΙΡΟΚΙΝΗΤΗ ΧΡΗΣΗ ΤΡΙΩΝ ΠΡΟΓΡΑΜΜΑΤΩΝ

1. Συντάκτης

- **Γλώσσα : C**
- **Αποθηκεύει το : main.c** (απλό κείμενο με τη σύνταξη της γλώσσας)

C:\WINDOWS\system32\edit.com

File Edit Search View Options Help C:\main.c

```
#include <stdio.h>
 int main(void)
 {
   int x=0;
     x ++;
     printf("hello, world ", x);
     return 0;
 }
```

2. Μεταγλωττιστής

- **Διαβάζει το : main.c**
- **Παράγει το : main.obj** (Αντικείμενο πρόγραμμα. Δεν εκτελείται ακόμη)

C:\WINDOWS\system32\cmd.exe

```
C:\>cc main.c
C:\>dir /b
main.c
main.obj
C:\>
```

Εντολή MS-DOS για λίστα αρχείων στο δίσκο (ορίστε ;)

3. Συνδέτης

- **Διαβάζει το: main.obj**
- **Παράγει το: main.exe** (εκτελέσιμο στον Η/Υ)

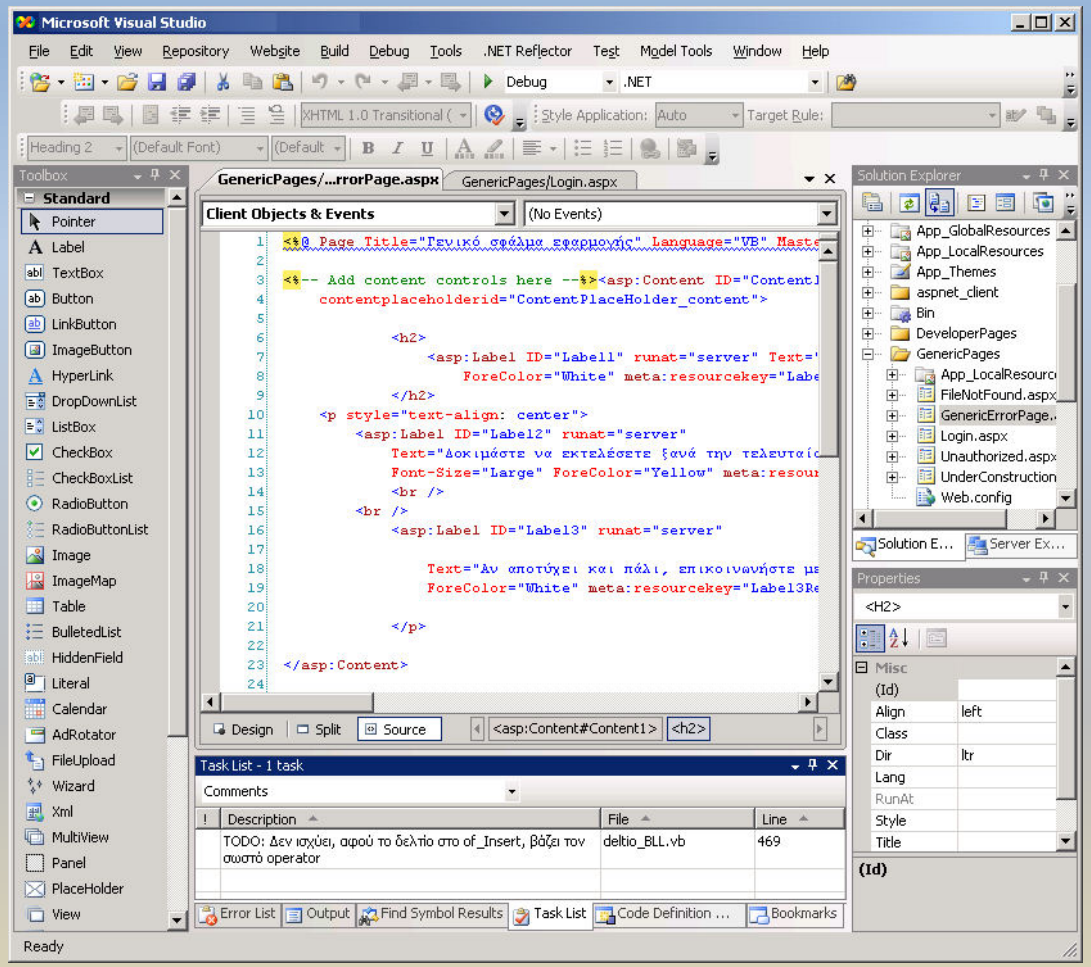
C:\WINDOWS\system32\cmd.exe

```
C:\>link main.obj /OUT: main.exe
C:\>dir /b
main.c
main.obj
main.exe
C:\>
```

ΟΛΟΚΛΗΡΩΜΕΝΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ

Integrated Development Environments (IDE)

- Περιλαμβάνεται αναβαθμισμένος συντάκτης κειμένου
- Εύκολος εντοπισμός συντακτικών λαθών
- Ηλεκτρονικό εγχειρίδιο βοήθειας για τη γλώσσα προγραμματισμού
- Γραφική σύνθεση των παραθύρων της εφαρμογής
- Πολλαπλές εκδόσεις
- Το ίδιο το περιβάλλον εργασίας αναλαμβάνει τη μεταγλώττιση και σύνδεση



ΕΙΔΗ ΛΑΘΩΝ ΣΤΑ ΠΡΟΓΡΑΜΜΑΤΑ

Συντακτικά λάθη (syntax errors)

- Οφείλονται στη λανθασμένη χρήση ή ορθογραφία των εντολών από τον προγραμματιστή
- Εντοπίζονται αμέσως από το περιβάλλον εργασίας κατά τη μεταγλώττιση
- Ο προγραμματιστής πρέπει να τα διορθώσει, για να μεταγλωττιστεί το πρόγραμμα

Λογικά (logical errors ή bugs)

- Είναι τα πιο επικίνδυνα, αφού οφείλονται σε κακή υλοποίηση του αλγορίθμου (είναι λάθη λογικής), π.χ. αντί μιας πρόσθεσης, προγραμματίστηκε μια αφαίρεση.
- Δύσκολα στην ανεύρεσή τους
- Μπορεί να οδηγήσουν σε σφάλματα εκτέλεσης (τερματισμό) του προγράμματος, π.χ. διαίρεση με το 0, ή άνοιγμα αρχείου που δεν υπάρχει, κλπ.