

Γ' τάξη Τεχνολογικής Κατεύθυνσης Ενιαίου Λυκείου

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Συγγραφείς

Α. Βακάλη

Η. Γιαννόπουλος

Ν. Ιωαννίδης

Χ. Κοίλιας

Κ. Μάλαμας

Ι. Μανωλόπουλος

Π. Πολίτης

Διδάσκων: *Τσιωτάκης Παναγιώτης*





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



- ★ Η έννοια του προγράμματος

- ★ Ιστορική αναδρομή

- ★ Φυσικές και τεχνητές γλώσσες



- ★ Τεχνικές σχεδίασης προγραμμάτων

- ★ Αντικειμενοστραφής προγραμματισμός

- ★ Παράλληλος προγραμματισμός



- ★ Προγραμματιστικά περιβάλλοντα





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
- ★ Ο προγραμματισμός ασχολείται με το τρίτο αυτό στάδιο





Η έννοια του προγράμματος



- ★ Ο προγραμματισμός είναι αυτός που δίνει την εντύπωση ότι, οι υπολογιστές είναι έξυπνες μηχανές που επιλύουν τα πολύπλοκα προβλήματα
- ★ Ο υπολογιστής είναι μία μηχανή που καταλαβαίνει μόνο δύο καταστάσεις, το μηδέν και το ένα, τα ψηφία του δυαδικού συστήματος
- ★ Ο υπολογιστής μπορεί απλά να αποθηκεύει στη μνήμη τις ακολοιυθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ Η έννοια του προγράμματος

★ Ιστορική αναδρομή

★ Φυσικές και τεχνητές γλώσσες



★ Τεχνικές σχεδίασης προγραμμάτων

★ Αντικειμενοστραφής προγραμματισμός

★ Παράλληλος προγραμματισμός



★ Προγραμματιστικά περιβάλλοντα





Ιστορική αναδρομή

- ★ Οι πρώτοι υπολογιστές, τεράστιοι σε μέγεθος αλλά με πάρα πολύ περιορισμένες δυνατότητες και μικρές ταχύτητας επεξεργασίας
- ★ Εξελίχθηκαν σε πολύ μικρούς σε μέγεθος υπολογιστές με τεράστιες όμως δυνατότητες και ταχύτητες επεξεργασίας
- ★ Την ίδια αργή εξέλιξη ουσιαστικά έχουν και οι γλώσσες προγραμματισμού, οι οποίες αν και εξελίσσονται και συνεχώς εμπλουτίζονται με νέες δυνατότητες, τα χαρακτηριστικά τους και οι βασικές τους ιδιότητες ουσιαστικά παραμένουν τα ίδια





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Ιστορική αναδρομή

- Γλώσσες μηχανής



- Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

- Γλώσσες υψηλού επιπέδου

- Γλώσσες 4ης γενιάς

★ ...





Η έννοια του προγράμματος

- ★ Αρχικά έπρεπε να δοθούν κατευθείαν οι κατάλληλες ακολουθίες από 0 και 1, δηλαδή εντολές σε μορφή κατανοητή από τον υπολογιστή
- ★ Ελάχιστοι μπορούσαν να το κάνουν, αφού απαιτούσε βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή
- ★ Ο πρώτος υπολογιστής ο περίφημος ENIAC για να "προγραμματιστεί", ώστε να εκτελέσει κάποιους υπολογισμούς, έπρεπε να αλλάξουν θέση εκατοντάδες διακόπτες και να ρυθμιστούν αντίστοιχα όλες οι καλωδιώσεις





Η έννοια του προγράμματος

- ★ Οι εντολές ενός προγράμματος και σήμερα μετατρέπονται σε ακολουθίες που αποτελούνται από 0 και 1, τις εντολές σε **γλώσσα μηχανής**, όπως ονομάζονται, οι οποίες εκτελούνται από τον υπολογιστή

```
10101000 00001010
10001100 00000001
00111100
01010001 00000001
01000011 00000001
11000000 11111010
10001100 00000010
11111111
```





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Ιστορική αναδρομή

– Γλώσσες μηχανής

– Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

– Γλώσσες υψηλού επιπέδου

– Γλώσσες 4ης γενιάς



★ ...





Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου



- ★ Γίνονται προσπάθειες για τη δημιουργία μίας συμβολικής γλώσσας, η οποία ενώ θα έχει έννοια για τον άνθρωπο, θα μετατρέπεται εσωτερικά από τους υπολογιστές στις αντίστοιχες ακολουθίες από 0 και 1
- ★ Για παράδειγμα η λέξη ADD (πρόσθεσε) ακολουθούμενη από δύο αριθμούς, είναι κατανοητή και απομνημονεύεται. Η εντολή θα μεταφραστεί από τον υπολογιστή σε μία ακολουθία δυαδικών ψηφίων και στη συνέχεια μπορεί να εκτελεστεί
- ★ Το έργο της μετάφρασης το αναλαμβάνει ένα ειδικό πρόγραμμα, ο **συμβολομεταφραστής** (assembler)



Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

```
INDEX=$01  
SUM=$02  
LDA #10  
STA INDEX  
CLA  
LOOP  ADD  INDEX  
      DEC  INDEX  
      BNE  LOOP  
      STA  SUM  
      BRK
```

- ★ Ωστόσο παρέμεναν στενά συνδεδεμένες με την αρχιτεκτονική του κάθε υπολογιστή
- ★ Επίσης δεν διέθεταν εντολές πιο σύνθετων λειτουργιών οδηγώντας έτσι σε μακροσκελή προγράμματα, που ήταν δύσκολο να γραφούν και κύρια να συντηρηθούν
- ★ Οι γλώσσες αυτές ονομάζονται συμβολικές ή γλώσσες χαμηλού επιπέδου, αφού εξαρτώνται από την αρχιτεκτονική του υπολογιστή



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Ιστορική αναδρομή

- Γλώσσες μηχανής
- Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου
- Γλώσσες υψηλού επιπέδου
- Γλώσσες 4ης γενιάς



★ ...





Γλώσσες υψηλού επιπέδου



- ★ Οι παραπάνω ανεπάρκειες των συμβολικών γλωσσών οδήγησαν στα τέλη της δεκαετίας του 50 στην εμφάνιση των πρώτων γλωσσών προγραμματισμού υψηλού επιπέδου
- ★ Το 1957 η IBM ανέπτυξε την πρώτη γλώσσα υψηλού επιπέδου τη **FORTRAN**.
Το όνομα FORTRAN προέρχεται από τις λέξεις FORmula TRANslation, που σημαίνουν μετάφραση τύπων
- ★ Η FORTRAN αναπτύχθηκε ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων



Γλώσσες υψηλού επιπέδου

- ★ Το ίδιο πρόγραμμα **FORTRAN** μπορεί να εκτελεστεί σε οποιοδήποτε άλλο υπολογιστή, αρκεί να υπάρχει ο αντίστοιχος μεταγλωττιστής για τον υπολογιστή αυτό
- ★ Η γλώσσα FORTRAN μετά από πολλές αλλαγές, προσθήκες και βελτιώσεις χρησιμοποιείται ακόμη και σήμερα για επιστημονικές εφαρμογές





FORTRAN



```
C  PROGRAM EQUATION
      READ(*,1) A,B
1    FORMAT(F5.1)
      IF (A.EQ.0) GO TO 20
      X=(-1.)*B/A
      WRITE(*,2) X
2    FORMAT('X=',F10.2)
      GO TO 50
20   IF (B.EQ.0) WRITE(*,3)
      IF (B.NE.0) WRITE(*,4)
3    FORMAT('ΆϊñÉóôç')
4    FORMAT('ΆÄõíÁôç')
50   STOP
      END
```




Γλώσσες υψηλού επιπέδου



- ★ Το 1960 αναπτύχθηκε μία άλλη γλώσσα, σταθμός στον προγραμματισμό η γλώσσα COBOL
- ★ Η **COBOL** είναι κατάλληλη για ανάπτυξη εμπορικών εφαρμογών, και γενικότερα διαχειριστικών εφαρμογών, τομέας όπου η FORTRAN υστερούσε



Γλώσσες υψηλού επιπέδου



- ★ Μια από τις σημαντικότερες γλώσσα προγραμματισμού με ελάχιστη στη πρακτική εφαρμογή αλλά που επηρέασε ιδιαίτερα τον προγραμματισμό και τις επόμενες γλώσσες, είναι η **ALGOL**
- ★ Αναπτύχθηκε από Ευρωπαίους επιστήμονες, αρχικά το 1960, με σκοπό τη δημιουργία γενικής φύσης προγραμμάτων που να μη συνδέονται με συγκεκριμένες εφαρμογές



Η γλώσσα προγραμματισμού COBOL



```
IDENTIFICATION DIVISION.  
PROGRAM-ID. EQUATION.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
SOURCE-COMPUTER. IBM-PC.  
OBJECT-COMPUTER. IBM-PC.  
SPECIAL-NAMES. DECIMAL-POINT IS COMMA.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 X          PIC S9(6)V9.  
77 A          PIC S9(6).  
77 B          PIC S9(6).  
77 W-X        PIC -(6),-.  
PROCEDURE DIVISION.  
ARXH.  
    DISPLAY ' ÆÜóÅ Á'.  
    ACCEPT A.  
    DISPLAY ' ÆÜóÅ Â'.  
    ACCEPT B.  
    DISPLAY ' '.  
    IF A = 0 GO TO ROYT-1.  
    COMPUTE X = B * (- 1) / A.  
    MOVE X TO W-X.  
    DISPLAY ' Η ÆÖóÇ ÅÉíÁÉ : ' W-X.  
    STOP RUN.  
ROYT-1.  
    IF B = 0  
        DISPLAY ' ÅĬÑÉóôÇ'  
    ELSE  
        DISPLAY ' ÅÄöíÁôÇ'.  
    STOP RUN.
```



Γλώσσες υψηλού επιπέδου



- ★ Στα μέσα της δεκαετίας του 60 αναπτύχθηκε η γλώσσα **PL/1** που προσπάθησε, χωρίς επιτυχία να καλύψει όλους τους τομείς του προγραμματισμού, επιστημονικούς και εμπορικούς, αντικαθιστώντας τόσο τη FORTRAN όσο και την COBOL



- ★ Στο χώρο της Τεχνητής Νοημοσύνης αναπτύχθηκαν δύο γλώσσες αρκετά διαφορετικές από όλες τις άλλες. Στα μέσα του 60 αναπτύχθηκε στο MIT η **LISP**, γλώσσα η οποία προσανατολίζεται σε χειρισμό λιστών από σύμβολα και η **PROLOG** στις αρχές του 70





Γλώσσες υψηλού επιπέδου

- ★ Δύο σημαντικότερες γλώσσες γενικού σκοπού, οι οποίες αναπτύχθηκαν τη δεκαετία του 60 αλλά χρησιμοποιούνται πάρα πολύ στις ημέρες μας, είναι η BASIC και η PASCAL

```
10 REM ΑΔΕΞΘΟÇ ÅÎÉÓÛÓÇ Å' ÅÆÈÌÏÕ  
20 INPUT "A=",A  
30 INPUT "B=",B  
40 IF A=0 THEN 100  
50 X=-B/A  
60 PRINT "X=" ;X  
70 END  
100 IF B=0 THEN PRINT "Άϊñέόôç" ELSE PRINT "Άäõíáôç"  
110 END
```



Γλώσσες υψηλού επιπέδου



- ★ Η γλώσσα προγραμματισμού **BASIC** αρχικά αναπτύχθηκε ως γλώσσα για την εκπαίδευση αρχαρίων στον προγραμματισμό
- ★ Σχεδιάστηκε για να γράφονται σύντομα προγράμματα, τα οποία εκτελούνται με τη βοήθεια διερμηνευτή (interpreter)
- ★ Η ανάπτυξη όμως των μικροϋπολογιστών και οι συνεχείς εκδόσεις της γλώσσας βοήθησαν στην εξάπλωσή της, τόσο ώστε να γίνει ίσως η δημοφιλέστερη γλώσσα στους προσωπικούς υπολογιστές





Γλώσσες υψηλού επιπέδου



- ★ Η γλώσσα **PASCAL** έφερε μεγάλες αλλαγές στον προγραμματισμό
- ★ Παρουσιάστηκε το 1970 και στηρίχτηκε πάνω στην ALGOL
- ★ Είναι μία γλώσσα γενικής χρήσης, η οποία είναι κατάλληλη τόσο για την εκπαίδευση όσο και τη δημιουργία ισχυρών προγραμμάτων κάθε τύπου
- ★ Χαρακτηριστικό της γλώσσας είναι η καταλληλότητα για τη δημιουργία δομημένων προγραμμάτων



Γλώσσες υψηλού επιπέδου



- ★ Η PASCAL γνώρισε και συνεχίζει να γνωρίζει τεράστια εξάπλωση ειδικά στο χώρο των μικροϋπολογιστών και αποτέλεσε τη βάση για την ανάπτυξη άλλων ισχυρότερων γλωσσών όπως η ADA και η Modula-2
- ★ Στα μέσα του 1960 παρουσιάστηκε για πρώτη φορά μία τεχνική ή σχεδίασης προγραμμάτων που έμελλε να αλλάξει ριζικά τον τρόπο ανάπτυξης προγραμμάτων καθώς και τις ίδιες τις γλώσσες προγραμματισμού





Γλώσσες υψηλού επιπέδου



- ★ Η τεχνική του **δομημένου προγραμματισμού** η οποία εξασφαλίζει τη δημιουργία προγραμμάτων απλών στη συγγραφή και την κατανόηση και εύκολων στη διόρθωση
- ★ Ο δομημένος προγραμματισμός και τα χαρακτηριστικά του θα παρουσιαστούν εκτενώς σε επόμενη παράγραφο



Γλώσσες υψηλού επιπέδου

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    float a, b;

    printf("A = ");
    scanf("%f", &a);
    printf("B = ");
    scanf("%f", &b);
    if (a == 0) {
        if (b == 0) {
            printf("Άϊνστάϊν\n");
        }
        else {
            printf("Άϊνστάϊν\n");
        }
    }
    else {
        printf("X = %f\n", -b/a);
    }

    return 0;
}
```

- ★ Μία ακόμη γλώσσα που γνώρισε μεγάλη διάδοση είναι η γλώσσα **C**
- ★ Η C αναπτύχθηκε στα εργαστήρια της εταιρείας BELL και χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος Unix, γλώσσα με ισχυρά χαρακτηριστικά, μερικά από αυτά κοινά με την Pascal κατάλληλη για ανάπτυξη δομημένων εφαρμογών αλλά και με πολλές δυνατότητες γλωσσας χαμηλού επιπέδου
- ★ Η C εξελίχτηκε στη γλώσσα C++, που είναι αντικειμενοστραφής



Γλώσσες υψηλού επιπέδου



- ★ Τα τελευταία χρόνια χρησιμοποιείται ιδιαίτερα, ειδικά για προγραμματισμό στο Διαδίκτυο, η **JAVA**
- ★ Η JAVA είναι μία αντικειμενοστραφής γλώσσα που αναπτύχθηκε από την εταιρεία SUN με σκοπό την ανάπτυξη εφαρμογών, που θα εκτελούνται σε διαφορετικούς υπολογιστές οι οποίοι είναι συνδεδεμένοι στο Διαδίκτυο
- ★ Τα προγράμματα αυτά μπορούν να εκτελούνται από διαφορετικούς υπολογιστές, προσωπικούς ή μεγάλα συστήματα με διαφορετικά λειτουργικά συστήματα χωρίς αλλαγές



Γλώσσες υψηλού επιπέδου

- ★ Η εμφάνιση των γραφικών περιβαλλόντων εργασίας δημιούργησε την ανάγκη για ανάπτυξη αντιστοίχων προγραμμάτων
- ★ Έτσι εμφανίστηκαν γλώσσες ή νέες εκδόσεις των γλωσσών που υλοποιούσαν τις έννοιες του **οδηγούμενου από το γεγονός προγραμματισμού** (object driven programming) και του **οπτικού προγραμματισμού** (visual programming)





Γλώσσες υψηλού επιπέδου



- ★ Με τον όρο **οπτικό προγραμματισμό** εννοούμε τη δυνατότητα να δημιουργούμε γραφικά περιβάλλοντα
- ★ Με τον όρο **οδηγούμενο από το γεγονός προγραμματισμό** εννοούμε τη δυνατότητα να ενεργοποιούνται λειτουργίες του προγράμματος με την εκτέλεση ενός γεγονότος, για παράδειγμα την επιλογή μίας εντολής από ένα μενού ή το κλικ του ποντικιού
- ★ Οι πιο διαδεδομένες γλώσσες προγραμματισμού σε γραφικό περιβάλλον για προσωπικούς υπολογιστές είναι η Visual Basic, η Visual C++ και η Java



Πλεονεκτήματα των γλωσσών υψηλού επιπέδου



- ★ Ο φυσικότερος και πιο “ανθρώπινος” τρόπος έκφρασης των προβλημάτων
- ★ Η ανεξαρτησία από τον τύπο του υπολογιστή. Η δυνατότητα της **μ**
εταφερισιμότητας των προγραμμάτων είναι σημαντικό προσόν
- ★
Η ευκολία της εκμάθησης και εκπαίδευσης ως απόρροια των προηγούμενων
- ★
Η διόρθωση λαθών και η συντήρηση προγραμμάτων σε γλώσσα υψηλού επιπέδου είναι πολύ ευκολότερο έργο
- ★ Ελάττωσαν το χρόνο και το κόστος παραγωγής νέων προγραμμάτων



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Ιστορική αναδρομή

- Γλώσσες μηχανής
- Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου
- Γλώσσες υψηλού επιπέδου
- Γλώσσες 4ης γενιάς



★ ...





Γλώσσες 4ης γενιάς

- ★ Στις γλώσσες αυτές ο χρήστης ενός υπολογιστή έχει τη δυνατότητα να υποβάλει ερωτήσεις στο σύστημα ή να αναπτύσσει εφαρμογές που ανακτούν πληροφορίες από βάσεις δεδομένων και να καθορίζει τον ακριβή τρόπο εμφάνισης αυτών των πληροφοριών, π.χ.

```
SELECT ENAME, JOB, SAL
  FROM EMPLOYES
 WHERE DEPTNO=20
    AND SAL > 300000;
```

Η ερώτηση αυτή σε SQL εκτελεί αναζήτηση στη βάση δεδομένων EMPLOYES και επιστρέφει το όνομα, τη θέση και το μισθό των υπαλλήλων της διεύθυνσης 20 που κερδίζουν πάνω από 300.000 δρχ.



Ταξινόμηση γλωσσών προγραμματισμού



- ★ Όλες οι γλώσσες προγραμματισμού που έχουν αναπτυχθεί μέχρι σήμερα αντιπροσωπεύουν διάφορες ιδέες πάνω στον προγραμματισμό και η κάθε μία είναι συνήθως καλύτερα προσαρμοσμένη σε ορισμένες κατηγορίες προβλημάτων
- ★ Η μεγάλη πλειοψηφία των γλωσσών ανήκει στην κατηγορία των **διαδικασιακών** (procedural) γλωσσών
- ★ Είναι γνωστές επίσης και ως **αλγοριθμικές** γλώσσες, γιατί είναι σχεδιασμένες για να επιτρέπουν την υλοποίηση αλγορίθμων





Ταξινόμηση γλωσσών προγραμματισμού – Γενικά



- ★ **Αντικειμενοστραφείς γλώσσες**
(object - oriented languages)



- ★ **Συναρτησιακές γλώσσες**
(functional languages) π.χ. LISP



- ★ **Μη διαδικασιακές γλώσσες**
(non procedural languages) π.χ. PROLOG. Χαρακτηρίζονται επίσης και ως γλώσσες πολύ υψηλού επιπέδου

- ★ **Γλώσσες ερωταπαντήσεων**
(query languages) π.χ. SQL



Ταξινόμηση με βάση περιοχή χρήσης

★ Γλώσσες γενικής χρήσης.

Θεωρητικά κάθε γλώσσα γενικής χρήσης μπορεί να χρησιμοποιηθεί για την επίλυση οποιουδήποτε προβλήματος

★ Στην πράξη ωστόσο κάθε γλώσσα έχει σχεδιαστεί για να ανταποκρίνεται καλύτερα σε ορισμένη κατηγορία προβλημάτων:

- **Γλώσσες επιστημονικής κατεύθυνσης**
(science-oriented languages) π.χ. FORTRAN
- **Γλώσσες εμπορικής κατεύθυνσης**
(business-oriented languages) π.χ. COBOL



Ταξινόμηση με βάση περιοχή χρήσης



- ★ **Γλώσσες προγραμματισμού συστημάτων**
(system programming languages) π.χ. C



- ★ **Γλώσσες τεχνητής νοημοσύνης**
(artificial intelligence languages) π.χ. LISP, PROLOG



- ★ **Γλώσσες ειδικής χρήσης.**
Πρόκειται για γλώσσες που χρησιμοποιούνται σε ειδικές περιοχές εφαρμογών στη ρομποτική



Ποια είναι η καλύτερη γλώσσα προγραμματισμού



- ★ Υπάρχουν γλώσσες κατάλληλες για ανάπτυξη ειδικών εφαρμογών και άλλες κατάλληλες για γενική χρήση
- ★ Υπάρχουν γλώσσες κατάλληλες για εκπαίδευση και άλλες για ανάπτυξη εμπορικών εφαρμογών
- ★ Γλώσσες που επιτρέπουν την εύκολη ανάπτυξη εφαρμογών σε γραφικό περιβάλλον και άλλες που εκμεταλλεύονται τα παράλληλα συστήματα
- ★ Υπάρχουν γλώσσες ισχυρές και γλώσσες χωρίς μεγάλες δυνατότητες αλλά απλές και εύκολες στην εκμάθηση



Ποια είναι η καλύτερη γλώσσα προγραμματισμού



- ★ Ο προγραμματιστής καλείται να επιλέξει την “καλύτερη” γλώσσα για να υλοποιήσει το πρόγραμμα
- ★ **Μπορούμε να ισχυριστούμε με βεβαιότητα ότι μία γλώσσα προγραμματισμού που να είναι αντικειμενικά καλύτερη από τις άλλες δεν υπάρχει, ούτε πρόκειται να υπάρξει**
- ★ Η επιλογή της γλώσσας εξαρτάται από το είδος της εφαρμογής, το υπολογιστικό περιβάλλον στο οποίο θα εκτελεστεί, τα προγραμματιστικά περιβάλλοντα που διαθέτουμε και κυρίως τις γνώσεις του προγραμματιστή



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ Η έννοια του προγράμματος

★ Ιστορική αναδρομή

★ Φυσικές και τεχνητές γλώσσες



★ Τεχνικές σχεδίασης προγραμμάτων

★ Αντικειμενοστραφής προγραμματισμός

★ Παράλληλος προγραμματισμός



★ Προγραμματιστικά περιβάλλοντα





Το αλφάβητο



- ★ Αλφάβητο μίας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα
- ★ Π.χ. η ελληνική γλώσσα περιέχει τα εξής στοιχεία: Τα γράμματα του αλφαβήτου πεζά και κεφαλαία 48 δηλαδή χαρακτήρες (Α-Ω και α-ω), τα 10 ψηφία (0-9) και όλα τα σημεία στίξης
- ★ Αντίστοιχα η αγγλική γλώσσα περιλαμβάνει τα γράμματα του αγγλικού αλφαβήτου (Α-Z και α-z) καθώς και τα ψηφία και όλα τα σημεία στίξης που χρησιμοποιούνται



Το λεξιλόγιο



- ★ Το λεξιλόγιο αποτελείται από ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου, τις λέξεις που είναι δεκτές από την γλώσσα
- ★ Για παράδειγμα στην ελληνική γλώσσα η ακολουθία των γραμμάτων ΑΒΓΑ είναι δεκτή αφού αποτελεί λέξη, αλλά η ακολουθία ΑΒΓΔΑ δεν αποτελεί λέξη της ελληνικής γλώσσας, άρα δεν είναι δεκτή



Η γραμματική



- ★ Η Γραμματική αποτελείται από το **ΤΥΠΙΚΟ** ή **ΤΥΠΟΛΟΓΙΚΟ** (accidence) και το **ΣΥΝΤΑΚΤΙΚΟ** (syntax)



- ★ **ΤΥΠΙΚΟ**
είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή
- ★ Για παράδειγμα στην ελληνική γλώσσα οι λέξεις γλώσσα, γλώσσας, γλώσσες είναι δεκτές, ενώ η λέξη γλώσσατ δεν είναι αποδεκτή





Η γραμματική

★ **Συντακτικό**

είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη δημιουργία προτάσεων

★ Η γνώση του συντακτικού επιτρέπει τη δημιουργία σωστών προτάσεων στις φυσικές γλώσσες ενώ στις γλώσσες προγραμματισμού τη δημιουργία σωστών εντολών





Η σημασιολογία



- ★ Η σημασιολογία (Semantics) είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατά επέκταση των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μία γλώσσα
- ★ Στις γλώσσες προγραμματισμού οι οποίες είναι τεχνητές γλώσσες, ο δημιουργός της γλώσσας αποφασίζει τη σημασιολογία των λέξεων της γλώσσας





Διαφορές φυσικών και τεχνητών γλωσσών



- ★ Μία βασική διαφορά μεταξύ φυσικών και τεχνητών γλωσσών είναι η δυνατότητα εξέλιξής τους
- ★ Οι φυσικές γλώσσες εξελίσσονται συνεχώς, νέες λέξεις δημιουργούνται, κανόνες γραμματικής και σύνταξης αλλάζουν
- ★ Αντίθετα οι τεχνητές γλώσσες χαρακτηρίζονται από στασιμότητα, αφού κατασκευάζονται συνειδητά για ένα συγκεκριμένο σκοπό



Διαφορές φυσικών και τεχνητών γλωσσών



- ★ Ωστόσο συχνά οι γλώσσες προγραμματισμού βελτιώνονται και μεταβάλλονται από τους δημιουργούς τους, με σκοπό:
 - ← να διορθωθούν αδυναμίες
 - ← να καλύψουν μεγαλύτερο εύρος εφαρμογών
 - ← να ακολουθήσουν τις νέες εξελίξεις



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ Η έννοια του προγράμματος

★ Ιστορική αναδρομή

★ Φυσικές και τεχνητές γλώσσες



★ Τεχνικές σχεδίασης προγραμμάτων

★ Αντικειμενοστραφής προγραμματισμός

★ Παράλληλος προγραμματισμός



★ Προγραμματιστικά περιβάλλοντα





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι

Γίνονται συνεχείς προσπάθειες για ανάπτυξη μεθοδολογιών και τεχνικών προγραμματισμού, που θα εξασφαλίζουν τη δημιουργία απλών και κομψών προγραμμάτων

★ ...

★ Τεχνικές σχεδίασης προγραμμάτων

- Ιεραρχική σχεδίαση προγράμματος
- Τμηματικός προγραμματισμός
- Δομημένος προγραμματισμός

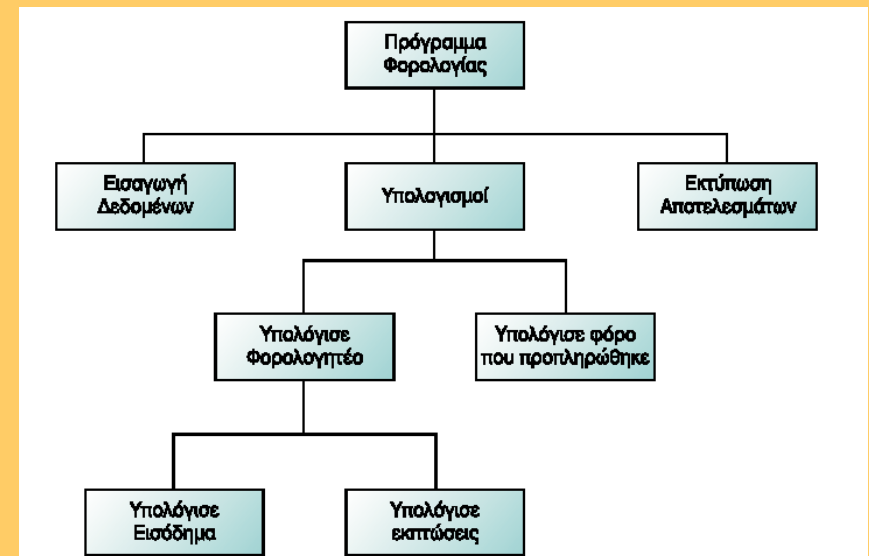
★ ...





Ιεραρχική σχεδίαση προγράμματος

- ★ Η τεχνική της ιεραρχικής σχεδίασης και επίλυσης ή η διαδικασία σχεδίασης "από επάνω προς τα κάτω" όπως συχνά ονομάζεται (top-down program design)
- ★ Σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε απλούστερα υποπροβλήματα, τα οποία να είναι εύκολο να επιλυθούν





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Τεχνικές σχεδίασης προγραμμάτων

– Ιεραρχική σχεδίαση προγράμματος

– Τμηματικός προγραμματισμός

– Δομημένος προγραμματισμός



★ ...





Τμηματικός προγραμματισμός

- ★ Η ιεραρχική σχεδίαση προγράμματος υλοποιείται με τον τμηματικό προγραμματισμό
- ★ Μετά την ανάλυση του προβλήματος σε αντίστοιχα υποπροβλήματα, κάθε υποπρόβλημα αποτελεί ανεξάρτητη **ενότητα** (module), που γράφεται ξεχωριστά από τα υπόλοιπα τμήματα προγράμματος
- ★ Ο τμηματικός προγραμματισμός διευκολύνει τη δημιουργία του προγράμματος, μειώνει τα λάθη και επιτρέπει την ευκολότερη παρακολούθηση, κατανόηση και συντήρηση του προγράμματος από τρίτους



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ ...

★ Τεχνικές σχεδίασης προγραμμάτων

– Ιεραρχική σχεδίαση προγράμματος

– Τμηματικός προγραμματισμός

– Δομημένος προγραμματισμός



★ ...





Δομημένος προγραμματισμός

- ★ Η μεθοδολογία που σήμερα έχει επικρατήσει απόλυτα και σχεδόν όλες οι σύγχρονες γλώσσες προγραμματισμού υποστηρίζουν, είναι ο δομημένος προγραμματισμός (structured programming)
- ★ Το 1968 ο καθηγητής Edsger Dijkstra δημοσίευσε ένα κείμενο με τίτλο "GO TO Statement Considered Harmful -η εντολή GOTO θεωρείται επιβλαβής" και θεμελίωσε το δομημένο προγραμματισμό
- ★ Βασικός λόγος για τα προβλήματα αυτά ήταν η αλόγιστη χρήση μίας εντολής, της εντολής GOTO που χρησιμοποιούμενη άλλαζε διαρκώς τη ροή του προγράμματος



Δομημένος προγραμματισμός

- ★ Ο δομημένος προγραμματισμός είναι μία μεθοδολογία σύνταξης προγραμμάτων που έχει σκοπό
 - ← να βοηθήσει τον προγραμματιστή στην ανάπτυξη σύνθετων προγραμμάτων
 - ← να μειώσει τα λάθη
 - ← να εξασφαλίσει την εύκολη κατανόηση των προγραμμάτων
 - ← να διευκολύνει τις διορθώσεις και τις αλλαγές σε αυτά



Δομημένος προγραμματισμός

- ★ Ο δομημένος προγραμματισμός στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών
 - ← τη δομή της ακολουθίας
 - ← τη δομή της επιλογής
 - ← και τη δομή της επανάληψης
- ★ Η εντολή GOTO έχει ως αποτέλεσμα την αλλαγή της ροής του προγράμματος, της διακλάδωσης σε μία άλλη εντολή του προγράμματος εκτός από την επόμενη



Δομημένος προγραμματισμός

- ★ Ο δομημένος προγραμματισμός προήλθε από την ανάγκη του περιορισμού της ανεξέλεγκτης χρήσης του GOTO
- ★ Η χρήση της εντολής αυτής θα παρουσιαστεί με ένα απλό παράδειγμα

```
AN Αριθμός>0 ΤΟΤΕ GOTO 1
AN Αριθμός=0 ΤΟΤΕ GOTO 2
ΓΡΑΨΕ "Αρνητικός"
GOTO 4
1: ΓΡΑΨΕ "Θετικός"
GOTO 4
2: ΓΡΑΨΕ "Μηδέν"
GOTO 4
4: τελος_αλγορίθμου
```

```
AN Αριθμός>0 ΤΟΤΕ
    ΓΡΑΨΕ "Θετικός"
ΑΛΛΙΩΣ_ΑΝ Αριθμός=0
    ΤΟΤΕ_
        ΓΡΑΨΕ "Μηδέν"
ΑΛΛΙΩΣ_ΓΡΑΨΕ "Αρνητικός"
ΤΕΛΟΣ_ΑΝ
```




Δομημένος προγραμματισμός

- ★ Η χρήση του GOTO κάνει ακόμα και αυτό το μικρό τμήμα προγράμματος δύσκολο στην κατανόηση του και στην παρακολούθησή του
- ★ Όλες οι σύγχρονες γλώσσες προγραμματισμού, υποστηρίζουν το δομημένο προγραμματισμό και διαθέτουν εντολές που καθιστούν τη χρήση του GOTO περιττή





Πλεονεκτήματα Δομημένου προγραμματισμού



- ★ Δημιουργία απλούστερων προγραμμάτων
- ★ Άμεση μεταφορά των αλγορίθμων σε προγράμματα
- ★ Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα
- ★ Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος
- ★ Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους
- ★ Ευκολότερη διόρθωση και συντήρηση



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ Η έννοια του προγράμματος

★ Ιστορική αναδρομή

★ Φυσικές και τεχνητές γλώσσες



★ Τεχνικές σχεδίασης προγραμμάτων

★ Αντικειμενοστραφής προγραμματισμός

★ Παράλληλος προγραμματισμός



★ Προγραμματιστικά περιβάλλοντα





Αντικειμενοστραφής προγραμματισμός

- ★ Πρόκειται για μια νέα τάση αντιμετώπισης προγραμματιστικών αντιλήψεων και δομών που ονομάζεται **αντικειμενοστραφής** (object-oriented) προγραμματισμός
- ★ Ένα πρόγραμμα περιγράφει "ενέργειες" (επεξεργασία) που εφαρμόζονται πάνω σε δεδομένα





Αντικειμενοστραφής προγραμματισμός



- ★ Η αντικειμενοστραφής σχεδίαση εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα **αντικείμενα** (objects)
- ★ Χρησιμοποιεί την ιεραρχική σχεδίαση, τον τμηματικό προγραμματισμό και ακολουθεί τις αρχές του δομημένου προγραμματισμού



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



- ★ Η έννοια του προγράμματος
- ★ Ιστορική αναδρομή
- ★ Φυσικές και τεχνητές γλώσσες
- ★ Τεχνικές σχεδίασης προγραμμάτων
- ★ Αντικειμενοστραφής προγραμματισμός
- ★ Παράλληλος προγραμματισμός
- ★ Προγραμματιστικά περιβάλλοντα





Παράλληλος προγραμματισμός

- ★ Σχετικά πρόσφατα εμφανίστηκαν υπολογιστές που ξεφεύγουν από την κλασική αρχιτεκτονική και διαθέτουν περισσότερους από έναν επεξεργαστές
- ★ Οι επεξεργαστές αυτοί μοιράζονται την ίδια μνήμη και λειτουργούν παράλληλα εκτελώντας διαφορετικές εντολές του ίδιου προγράμματος
- ★ Οι υπολογιστές αυτοί εμφανίζονται θεωρητικά να πετυχαίνουν ταχύτητες, που είναι ασύλληπτες για τους τυπικούς υπολογιστές με έναν επεξεργαστή



Παράλληλος προγραμματισμός



★ Για να εκμεταλλευτούμε όμως την ταχύτητα που προσφέρει η αρχιτεκτονική αυτή, πρέπει το πρόβλημα να διαιρεθεί σε τμήματα που εκτελούνται παράλληλα



★ Στη συνέχεια να προγραμματιστεί σε ένα προγραμματιστικό περιβάλλον που να επιτρέπει τον παράλληλο προγραμματισμό





Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



★ Η έννοια του προγράμματος

★ Ιστορική αναδρομή

★ Φυσικές και τεχνητές γλώσσες



★ Τεχνικές σχεδίασης προγραμμάτων

★ Αντικειμενοστραφής προγραμματισμός

★ Παράλληλος προγραμματισμός



★ Προγραμματιστικά περιβάλλοντα





Προγραμματιστικά περιβάλλοντα



- ★ Κάθε πρόγραμμα που γράφτηκε σε οποιαδήποτε γλώσσα προγραμματισμού, πρέπει να μετατραπεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή, δηλαδή σε εντολές γλώσσας μηχανής
- ★ Η μετατροπή αυτή επιτυγχάνεται με τη χρήση ειδικών μεταφραστικών προγραμμάτων
- ★ Υπάρχουν δύο μεγάλες κατηγορίες τέτοιων προγραμμάτων:
 - ← οι **μεταγλωττιστές** (compilers)
 - ← οι **διερμηνευτές** (interpreters)



Προγραμματιστικά περιβάλλοντα

- ★ Ο **μεταγλωττιστής** δέχεται στην είσοδο ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής
- ★ Το τελευταίο μπορεί να εκτελείται οποτεδήποτε από τον υπολογιστή και είναι τελείως ανεξάρτητο από το αρχικό πρόγραμμα
- ★ Το αρχικό πρόγραμμα λέγεται **πηγαίο** πρόγραμμα (source), ενώ το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται **αντικείμενο** πρόγραμμα (object)





Προγραμματιστικά περιβάλλοντα



- ★ Αντίθετα ο **διερμηνευτής** διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής
- ★ Το αντικείμενο πρόγραμμα είναι μεν σε μορφή κατανοητή από τον υπολογιστή, αλλά συνήθως δεν είναι σε θέση να εκτελεστεί
- ★ Χρειάζεται να συμπληρωθεί και να συνδεθεί με άλλα τμήματα προγράμματος απαραίτητα για την εκτέλεσή του, τμήματα που είτε τα γράφει ο προγραμματιστής είτε βρίσκονται στις **βιβλιοθήκες** (libraries) της γλώσσας



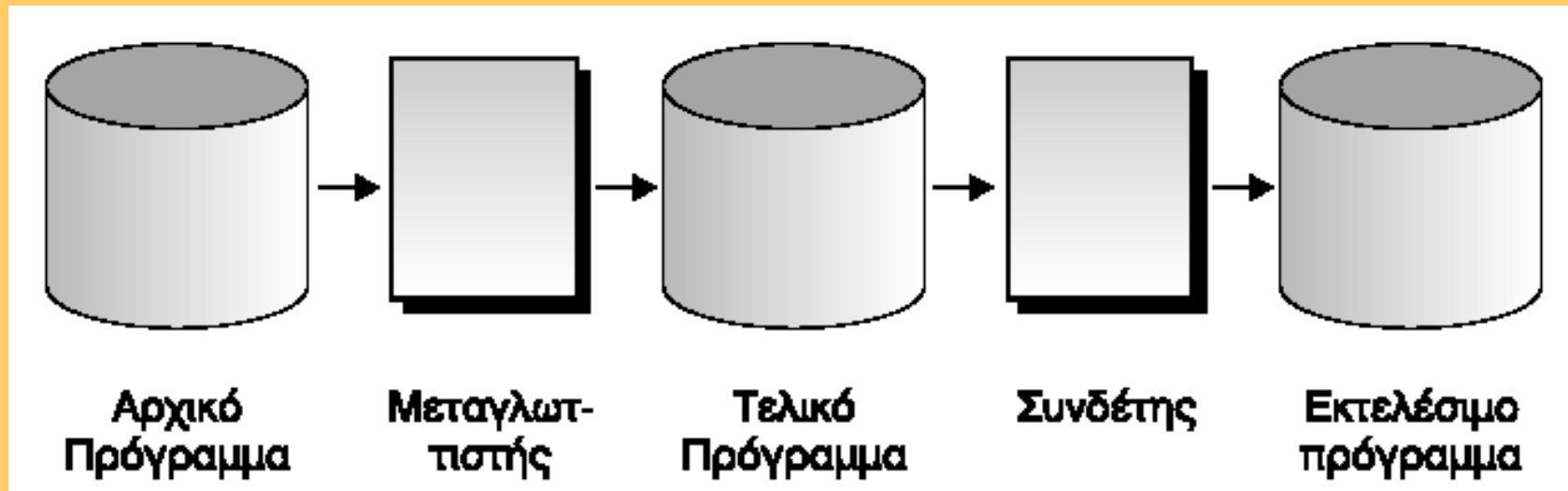
Προγραμματιστικά περιβάλλοντα



- ★ Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης – φορτωτής** (linkerloader)
- ★ Το αποτέλεσμα του συνδέτη είναι η παραγωγή του **εκτελέσιμου προγράμματος** (executable), το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή
- ★ Για το λόγο αυτό η συνολική διαδικασία αποκαλείται μεταγλώττιση και σύνδεση



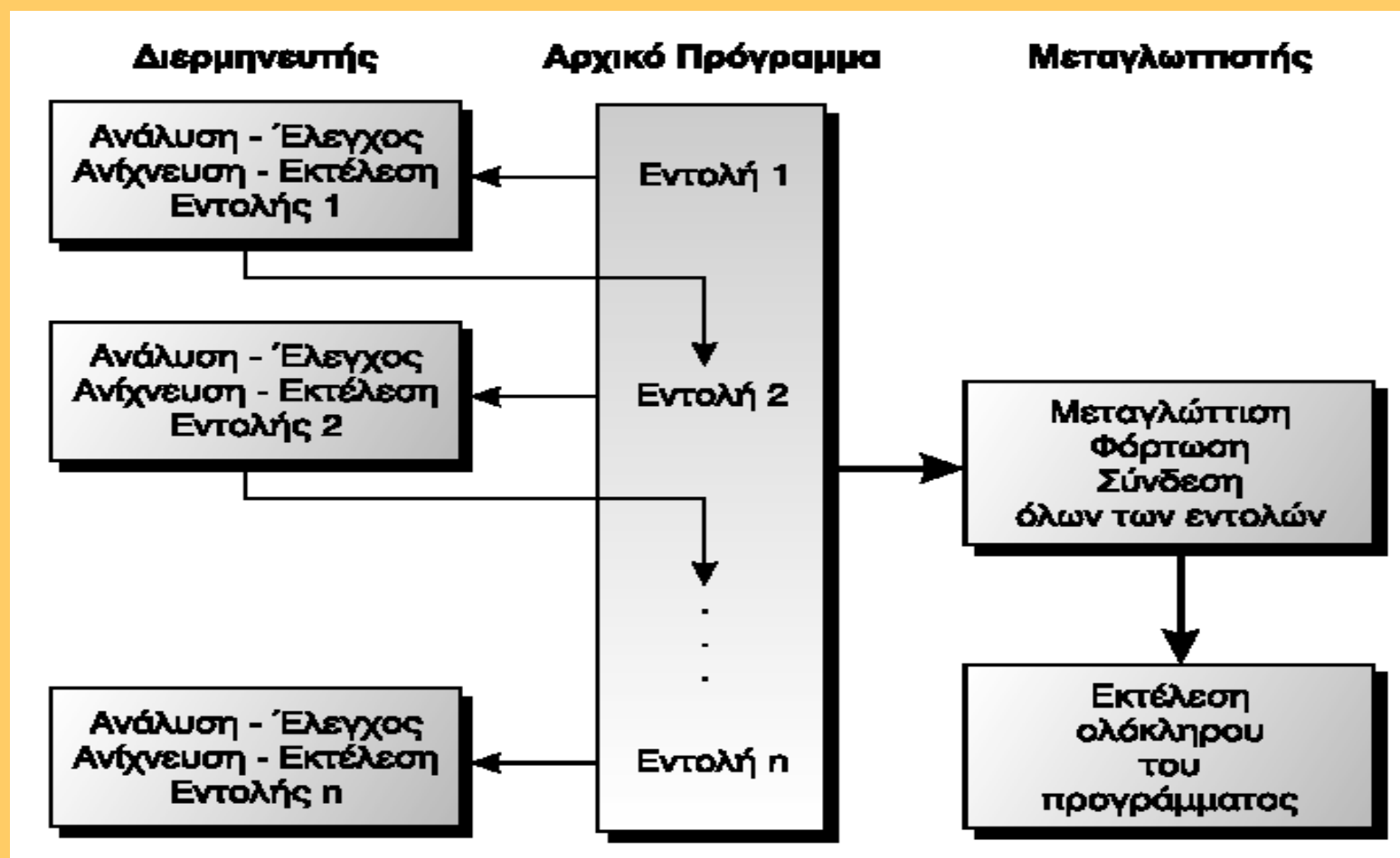
Προγραμματιστικά περιβάλλοντα



Μεταγλώττιση και σύνδεση προγράμματος



Προγραμματιστικά περιβάλλοντα



Διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος



Προγραμματιστικά περιβάλλοντα



- ★ Η δημιουργία του εκτελέσιμου προγράμματος γίνεται μόνο στην περίπτωση, που το αρχικό πρόγραμμα δεν περιέχει λάθη
- ★ Τις περισσότερες φορές κάθε πρόγραμμα αρχικά θα έχει λάθη
- ★ Τα λάθη του προγράμματος είναι γενικά δύο ειδών, λογικά και συντακτικά
- ★ Τα λογικά λάθη εμφανίζονται μόνο στην εκτέλεση, ενώ τα συντακτικά λάθη στο στάδιο της μεταγλώττισης





Προγραμματιστικά περιβάλλοντα



- ★ Τα λογικά λάθη που είναι τα πλέον σοβαρά και δύσκολα στη διόρθωση τους, οφείλονται σε σφάλματα κατά την υλοποίηση του αλγορίθμου
- ★ Τα συντακτικά οφείλονται σε αναγραμματισμούς ονομάτων εντολών, παρόληψη δήλωσης δεδομένων και πρέπει πάντα να διορθωθούν, ώστε να παραχθεί το τελικό εκτελέσιμο πρόγραμμα



Προγραμματιστικά περιβάλλοντα



- ★ Ο μεταγλωττιστής ή ο διερμηνευτής ανιχνεύει λοιπόν τα λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα
- ★ Το στάδιο που ακολουθεί είναι η διόρθωση των λαθών
- ★ Το διορθωμένο πρόγραμμα επαναυποβάλλεται για μεταγλώττιση και η διαδικασία αυτή επαναλαμβάνεται, μέχρις ότου εξαληφθούν πλήρως όλα τα λάθη





Προγραμματιστικά περιβάλλοντα

- ★ Η χρήση μεταγλωττιστή έχει το μειονέκτημα, ότι προτού χρησιμοποιηθεί ένα πρόγραμμα, πρέπει να περάσει από τη διαδικασία της μεταγλώττισης και σύνδεσης
- ★ Η χρήση διερμηνευτή έχει το πλεονέκτημα της άμεσης εκτέλεσης και συνεπώς και της άμεσης διόρθωσης
- ★ Όμως η εκτέλεση του προγράμματος καθίσταται πιο αργή, σημαντικά μερικές φορές, από εκείνη του ισοδύναμου εκτελέσιμου προγράμματος που παράγει ο μεταγλωττιστής





Προγραμματιστικά περιβάλλοντα

- ★ Πάντως τα σύγχρονα προγραμματιστικά περιβάλλοντα παρουσιάζονται συνήθως με μεικτές υλοποιήσεις, όπου χρησιμοποιείται διερμηνευτής κατά τη φάση δημιουργίας του προγράμματος και μεταγλωττιστής για την τελική έκδοση και εκμετάλλευση του προγράμματος





Προγραμματιστικά περιβάλλοντα



- ★ Για την αρχική σύνταξη των προγραμμάτων και τη διόρθωσή τους στη συνέχεια χρησιμοποιείται ένα ειδικό πρόγραμμα που ονομάζεται **συντάκτης** (editor)
- ★ Ο συντάκτης είναι ουσιαστικά ένας μικρός επεξεργαστής κειμένου, με δυνατότητες όμως που διευκολύνουν τη γρήγορη γραφή των εντολών των προγραμμάτων





Προγραμματιστικά περιβάλλοντα



- ★ Για τη δημιουργία, τη μετάφραση και την εκτέλεση ενός προγράμματος απαιτούνται τουλάχιστον τρία προγράμματα:

- ← ο συντάκτης

- ← ο μεταγλωττιστής

- ← ο συνδέτης

- ★ Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν αυτά τα προγράμματα με ενιαίο τρόπο



Κεφάλαιο 6 :

Δομές Δεδομένων και Αλγόριθμοι



Βιβλίο Μαθητή





Ερωτήσεις

Σωστό 1. Τα προγράμματα σε γλώσσες υψηλού επιπέδου είναι ανεξάρτητα του υπολογιστή που αναπτύχθηκαν

Λάθος 2. Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα

Λάθος 3. Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση

Σωστό 4. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα



Ερωτήσεις

5. Χαρακτηριστικό του οπτικού προγραμματισμού είναι

Α Επιτρέπει τη γραφική δημιουργία του περιβάλλοντος

Β. Επιτρέπει την ανάπτυξη του προγράμματος σε τμήματα

Γ. Είναι ταχύτερος στην εκτέλεση των προγραμμάτων

Δ. Επιτρέπει την διαγραμματική παράσταση της σχεδίασης του προγράμματος



6. Η Basic είναι

A. Κατάλληλη για εφαρμογές τεχνητής νοημοσύνης

B. Υποστηρίζει την ανάπτυξη παράλληλου προγραμματισμού

Γ. Μία γλώσσα γενικής χρήσης

Δ. Κατάλληλη μόνο για εκπαίδευση





Ερωτήσεις



7. Ο δομημένος προγραμματισμός στηρίζεται στη χρήση τριών στοιχειωδών δομών: της ακολουθίας
..... επιλογής και επανάληψης



8. Η μεταγλώττιση ενός προγράμματος γίνεται από τους
μεταγλωττιστές ως διερμηνευτές





Τεστ αυτοαξιολόγησης

Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει

1. Ο μεταγλωττιστής μεταγλωττίζει το πηγαίο πρόγραμμα σε αντικείμενο πρόγραμμα.
2. Ο τμηματικός προγραμματισμός υλοποιεί την ιεραρχική σχεδίαση του προγράμματος.
3. Οι γλώσσες που υλοποιούν τον οπτικό και τον από γεγονότα οδηγούμενο διευκολύνουν την ανάπτυξη εφαρμογών σε γραφικά περιβάλλοντα.



Τεστ αυτοαξιολόγησης

Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος

Σωστό

4. Η Visual Basic είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού.

Λάθος

5. Οι εντολές στις συμβολικές γλώσσες αποτελούνται από ακολουθίες 0 και 1.

Λάθος

6. Ο δομημένος προγραμματισμός εξασφαλίζει τη δημιουργία σωστών προγραμμάτων.

Λάθος

7. Οι γλώσσες 4^{ης} γενιάς είναι κατάλληλες για ανάπτυξη γενικών εφαρμογών.



Τεστ αυτοαξιολόγησης

Διάλεξε ένα μεταξύ των προτεινόμενων

8. Οι εντολές ενός προγράμματος γράφονται σε ένα πρόγραμμα που ονομάζεται:
- A. Συντάκτης
 - B. Μεταγλωττιστής
 - Γ. Διερμηνευτής
 - Δ. Συνδέτης
9. Η Pascal είναι μία γλώσσα:
- A. Μηχανής
 - B. Υψηλού επιπέδου
 - Γ. Συμβολική
 - Δ. 4ης γενιάς



Τεστ αυτοαξιολόγησης



10. Ο μεταγλωττιστής επισημαίνει:

A. Όλα τα λάθη του προγράμματος

B. Μόνο τα λογικά λάθη του προγράμματος

Γ. Μόνο τα συντακτικά λάθη του προγράμματος

Δ. Μόνο τα λάθη που προέρχονται από αναγραμματισμό των εντολών

11. Ο δομημένος προγραμματισμός είναι:

A. μία γενική μεθοδολογία ανάπτυξης προγραμμάτων

B. ένας τρόπος προγραμματισμού που εφαρμόζεται μόνο από τη γλώσσα Pascal

Γ. η εξέλιξη του τμηματικού προγραμματισμού

Δ. ένας τρόπος να εξαλείψουμε τις εντολές GOTO από ένα πρόγραμμα.



Τεστ αυτοαξιολόγησης

Διάλεξε όλα όσα χρειάζεται μεταξύ των προτεινόμενων

12. Ποια από τα παρακάτω είναι χαρακτηριστικά ενός δομημένου προγράμματος:

A. Δομικό στοιχείο είναι τα αντικείμενα.

B. Έχει μία είσοδο και μία έξοδο.

Γ. Χρησιμοποιεί τις τρεις δομές: της ακολουθίας, της επιλογής και της επανάληψης.

Δ. Μπορεί να εκμεταλλευτεί τους παράλληλους υπολογιστές.

13. Κάθε φυσική γλώσσα προσδιορίζεται από:

A. Το αλφάβητο της

B. Το λεξιλόγιο της

Γ. Τη γραμματική της

Δ. Τη σημασιολογία της





Τεστ αυτοαξιολόγησης



14. Ποιες από τις παρακάτω γλώσσες χρησιμοποιούνται για ανάπτυξη εφαρμογών τεχνητής νοημοσύνης:

A. LISP

B. FORTRAN

Γ. COBOL

Δ. PROLOG

E. JAVA

