

Κεφάλαιο 6 :

Εισαγωγή στον προγραμματισμό

- ▶ 6.1 Η έννοια του προγράμματος
 - ▶ 6.3 Φυσικές και τεχνητές γλώσσες
 - ▶ 6.4 Τεχνικές σχεδίασης προγραμμάτων
 - ▶ 6.7 Προγραμματιστικά περιβάλλοντα
- 

Η έννοια του προγράμματος

- ▶ Η επίλυση ενός προβλήματος με τον υπολογιστή περιλαμβάνει, όπως έχει ήδη αναφερθεί, τρία εξίσου σημαντικά στάδια.
 1. Τον ακριβή προσδιορισμό του προβλήματος.
 2. Την ανάπτυξη του αντίστοιχου αλγορίθμου.
 3. Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή.
- ▶ Ο προγραμματισμός ασχολείται με το τρίτο αυτό στάδιο, τη δημιουργία του προγράμματος

Η έννοια του προγράμματος

- ▶ **Πρόγραμμα** είναι ένα σύνολο εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση ενός προβλήματος.
- ▶ Το πρόγραμμα, το οποίο γράφεται σε κάποια γλώσσα προγραμματισμού, δεν είναι απλά η υλοποίηση του αλγορίθμου, αλλά βασικό στοιχείο του είναι τα δεδομένα και οι δομές δεδομένων επί των οποίων ενεργεί.

Η έννοια του προγράμματος

- ▶ Ο προγραμματισμός είναι αυτός που δίνει την εντύπωση ότι οι υπολογιστές είναι έξυπνες μηχανές που επιλύουν τα πολύπλοκα προβλήματα.
- ▶ Η εντύπωση αυτή όμως είναι απλώς μία ψευδαίσθηση.
- ▶ Ο υπολογιστής, ως γνωστόν, είναι μία μηχανή που καταλαβαίνει μόνο δύο καταστάσεις, οι οποίες αντιπροσωπεύονται με δύο αριθμούς, το 0 και το 1, τα ψηφία του δυαδικού συστήματος

Η έννοια του προγράμματος

- ▶ Το μόνο πράγμα που κάνει ο υπολογιστής είναι στοιχειώδεις ενέργειες σε ακολουθίες αυτών των δύο ψηφίων, αλλά αυτές τις ενέργειες τις εκτελεί με ασύλληπτη ταχύτητα.
 - ▶ Ο υπολογιστής μπορεί απλά να αποθηκεύει στη μνήμη τις ακολουθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει.
- 

Κεφάλαιο 6 :

Εισαγωγή στον προγραμματισμό

- ▶ 6.1 Η έννοια του προγράμματος
 - ▶ **6.3 Φυσικές και τεχνητές γλώσσες**
 - ▶ 6.4 Τεχνικές σχεδίασης προγραμμάτων
 - ▶ 6.7 Προγραμματιστικά περιβάλλοντα
- 

Φυσικές & τεχνητές γλώσσες

- ★ Μία γλώσσα προσδιορίζεται από
 - ★ το Αλφάβητο
 - ★ το Λεξιλόγιο
 - ★ τη Γραμματική
 - ★ τη Σημασιολογία

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 **τυπικό** ή **τυπολογικό** (accidence) και το **συντακτικό** (syntax)
- ★ **Τυπικό** 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 & 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 & 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

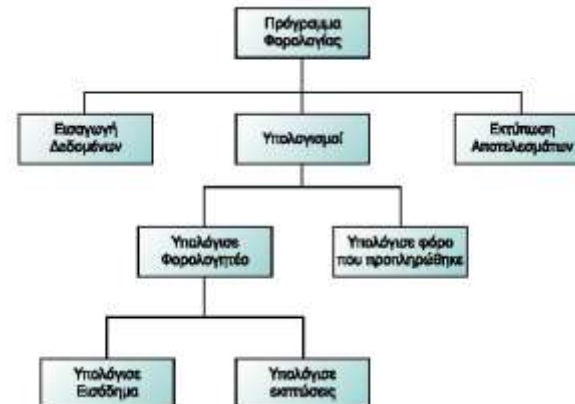
Κεφάλαιο 6 :

Εισαγωγή στον προγραμματισμό

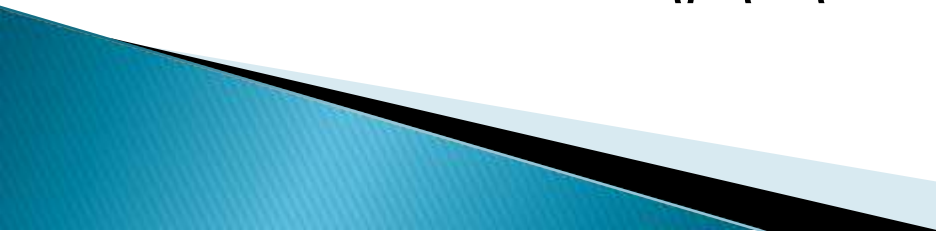
- ▶ 6.1 Η έννοια του προγράμματος
 - ▶ 6.3 Φυσικές και τεχνητές γλώσσες
 - ▶ **6.4 Τεχνικές σχεδίασης προγραμμάτων**
 - ▶ 6.7 Προγραμματιστικά περιβάλλοντα
- 

Ιεραρχική σχεδίαση προγράμματος

- ★ Τεχνική της ιεραρχικής σχεδίασης και επίλυσης ή 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
- ★ Για την υποβοήθηση της ιεραρχικής σχεδίασης χρησιμοποιούνται διάφορες διαγραμματικές τεχνικές



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
- 

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

```
ΑΝ Αριθμός>0 ΤΟΤΕ GOTO 1
ΑΝ Αριθμός=0 ΤΟΤΕ GOTO 2
ΓΡΑΨΕ “Αρνητικός”
GOTO 4
1: ΓΡΑΨΕ “Θετικός”
GOTO 4
2: ΓΡΑΨΕ “Μηδέν”
GOTO 4
4: τέλος_αλγορίθμου
```

```
ΑΝ 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

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

Πλεονεκτήματα Δομημένου προγραμματισμού

1. Δημιουργία απλούστερων προγραμμάτων
 2. Άμεση μεταφορά των αλγορίθμων σε προγράμματα
 3. Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα
 4. Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος
 5. Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους
 6. Ευκολότερη διόρθωση και συντήρηση
- 

Κεφάλαιο 6 :

Εισαγωγή στον προγραμματισμό

- ▶ 6.1 Η έννοια του προγράμματος
 - ▶ 6.3 Φυσικές και τεχνητές γλώσσες
 - ▶ 6.4 Τεχνικές σχεδίασης προγραμμάτων
 - ▶ **6.7 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)

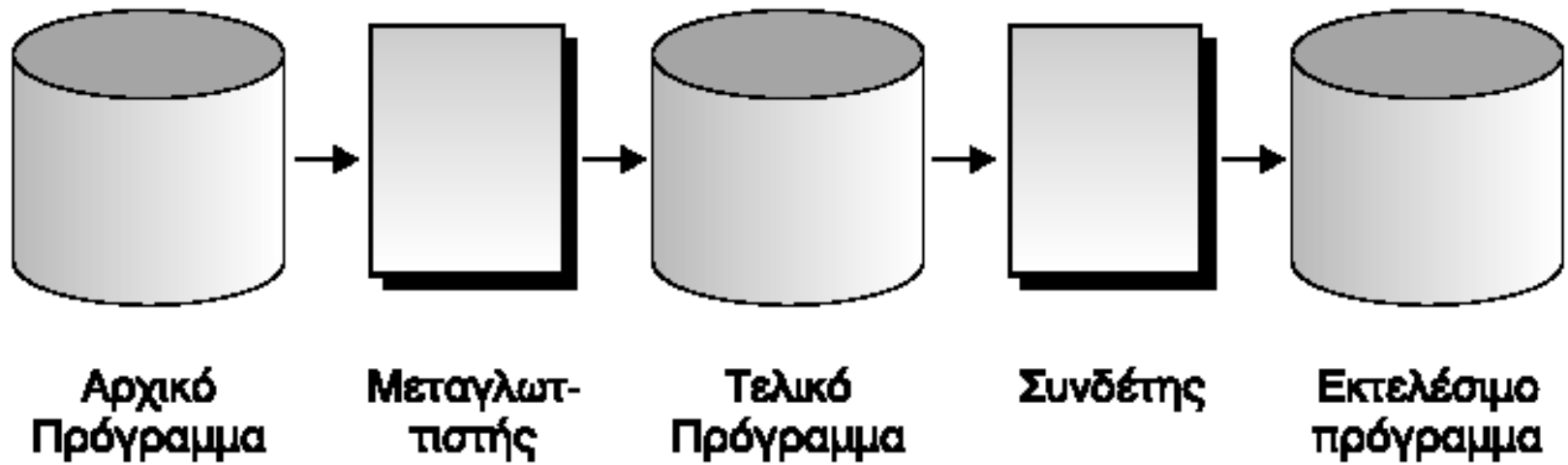
Προγραμματιστικά περιβάλλοντα

- ★ 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

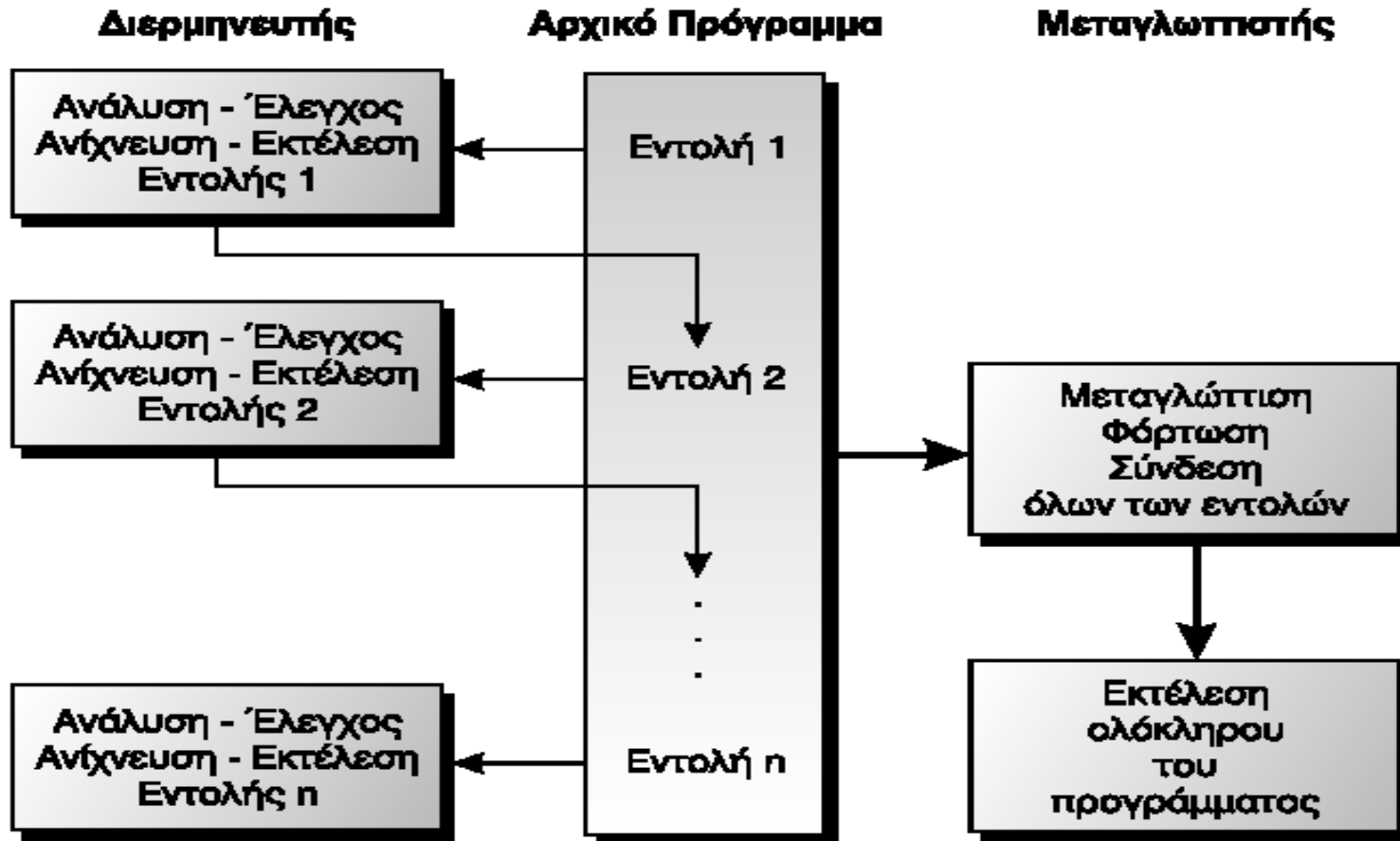
- ★ Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης – φορτωτής** (linkerloader)
- ★ Το αποτέλεσμα του συνδέτη είναι η παραγωγή του **εκτελέσιμου προγράμματος** (executable), το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή
- ★ Για το λόγο αυτό η συνολική διαδικασία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