

ΘΕΩΡΙΑ § 4.1

Ποιος ο ρόλος των διαφορετικών προσεγγίσεων στη λύση των προβλημάτων;

Η λύση σε ένα πρόβλημα μπορεί να προέλθει από ποικίλες και **διαφορετικές προσεγγίσεις**, τεχνικές και μεθόδους. Έτσι, είναι απαραίτητο να γίνεται μία **καλή ανάλυση** του κάθε προβλήματος και να προτείνεται συγκεκριμένη μεθοδολογία και ακολουθία βημάτων. Βασικός στόχος μας είναι η πρόταση έξυπνων και αποδοτικών λύσεων.

Ποια **βήματα** περιλαμβάνει η ανάλυση ενός προβλήματος σε ένα σύγχρονο υπολογιστικό περιβάλλον;

- την **καταγραφή της υπάρχουσας πληροφορίας** για το πρόβλημα,
- την **αναγνώριση των ιδιοτήτων** του προβλήματος,
- την **αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησής του** και στη συνέχεια:
- την **πρόταση επίλυσης με χρήση κάποιας μεθόδου**,
- την **τελική επίλυση με χρήση υπολογιστικών συστημάτων**.

Γιατί είναι **απαραίτητη η ανάλυση** του προβλήματος ;

Η ανάλυση κάθε προβλήματος είναι απαραίτητη, έτσι ώστε να αναζητηθεί η **πλέον κατάλληλη μέθοδος** που να παρέχει τη ζητούμενη λύση, όσο γίνεται **ταχύτερα** και με το λιγότερο δυνατό **κόστος σε υπολογιστικούς πόρους**.

Γιατί οι **μέθοδοι ανάλυσης και επίλυσης των προβλημάτων παρουσιάζουν ιδιαίτερο ενδιαφέρον**

- παρέχουν ένα γενικό πρότυπο κατάλληλο για την επίλυση προβλημάτων ευρείας κλίμακας,
- μπορούν να αναπαρασταθούν με κοινές δομές δεδομένων και ελέγχου (που υποστηρίζονται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού),
- παρέχουν τη δυνατότητα καταγραφής των χρονικών και "χωρικών" απαιτήσεων της μεθόδου επίλυσης, έτσι ώστε να μπορεί να γίνει επακριβής εκτίμηση των αποτελεσμάτων.

ΘΕΩΡΙΑ § 6.1, 6.3, 6.4, 6.7

Ποια είναι τα στάδια για την **επίλυση ενός προβλήματος με τον υπολογιστή** ;

- Ο ακριβής προσδιορισμός του προβλήματος.
- Η ανάπτυξη του αντίστοιχου αλγορίθμου.
- Η διατύπωση του αλγορίθμου σε καταννητή μορφή από τον υπολογιστή.

Προγραμματισμός είναι η διατύπωση των αλγορίθμων σε τέτοια μορφή, ώστε να μπορούν να υλοποιηθούν από τον υπολογιστή, δηλαδή τη δημιουργία του προγράμματος

Οι **γλώσσες προγραμματισμού** αναπτύχθηκαν με **σκοπό** την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστή).

Ο υπολογιστής εκτελεί τις ακόλουθες απλές ενέργειες με μεγάλη ταχύτητα:

- **αποθηκεύει** στη **μνήμη** τις ακολουθίες των δυαδικών ψηφίων, να τις **ανακτά**,
- να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές
- και να τις συγκρίνει.

Ποια είναι τα στοιχεία που προσδιορίζουν μία **γλώσσα**; (σ.115)

αλφάβητο, το λεξιλόγιο, η γραμματική και η σημασιολογία μιας γλώσσας

Τι είναι το αλφάβητο, το λεξιλόγιο, η γραμματική και η σημασιολογία μιας γλώσσας προγραμματισμού;

- **Αλφάβητο** μίας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα.
- Το **λεξιλόγιο** αποτελείται από ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου (λέξεις) που είναι δεκτές από την γλώσσα.
- Η **Γραμματική** αποτελείται από το τυπικό και το συντακτικό (syntax).
 - Τυπικό είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή.
 - Συντακτικό είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη δημιουργία προτάσεων.
- 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.

Ποιες είναι οι διαφορές **φυσικών** και **τεχνητών** γλωσσών; (σ.116)

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 με την πάροδο του χρόνου και αυτό γιατί η γλώσσα χρησιμοποιείται για την επικοινωνία μεταξύ

ανθρώπων, που εξελίσσονται και αλλάζουν ανάλογα με τις εποχές και τον κοινωνικό περίγυρο. Αντίθετα οι τεχνητές γλώσσες χαρακτηρίζονται από **στασιμότητα**, αφού κατασκευάζονται συνειδητά για ένα συγκεκριμένο σκοπό.

Τι είναι **ιεραρχική σχεδίαση**; (σ.116) (σχεδίαση από επάνω προς τα κάτω, top-down, λύνουμε το γενικό πρόβλημα με συνεχή διάσπαση σε κατώτερα επίπεδα)

Η τεχνική της ιεραρχικής σχεδίασης και επίλυσης ή η διαδικασία σχεδίασης “**από επάνω προς τα κάτω**” όπως συχνά ονομάζεται (top-down program design) **περιλαμβάνει** :

- τον καθορισμό των βασικών λειτουργιών ενός προγράμματος, σε ανώτερο επίπεδο,
- και στη συνέχεια τη διάσπαση των λειτουργιών αυτών σε όλο και μικρότερες λειτουργίες,
- μέχρι το τελευταίο επίπεδο που οι λειτουργίες είναι πολύ απλές, ώστε να επιλυθούν εύκολα.

Ποιος ο **σκοπός της ιεραρχικής σχεδίασης** (σ.116)

Σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση λοιπόν του προβλήματος σε μια σειρά από απλούστερα υποπροβλήματα, τα οποία να είναι εύκολο να επιλυθούν οδηγώντας στην επίλυση του αρχικού προβλήματος.

Τι είναι **τμηματικός προγραμματισμός**; (σ.116)

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων και υλοποιεί την ιεραρχική σχεδίαση.

Μετά την ανάλυση του προβλήματος σε αντίστοιχα υποπροβλήματα, κάθε υποπρόβλημα αποτελεί ανεξάρτητη ενότητα (module), που γράφεται ξεχωριστά από τα υπόλοιπα τμήματα προγράμματος.

Ποια τα **πλεονεκτήματα του τμηματικού προγραμματισμού**; (σ.117)

Ο τμηματικός προγραμματισμός

- * διευκολύνει τη δημιουργία του προγράμματος,
- * μειώνει τα λάθη και
- * επιτρέπει την ευκολότερη παρακολούθηση, κατανόηση και
- * συντήρηση του προγράμματος από τρίτους.

Τι είναι **δομημένος προγραμματισμός**; (σ.119) (Ποιες είναι οι **αρχές** του;)(σ135,208) (μεθοδολογία σύνταξης και ανάπτυξης προγραμμάτων που περιέχει την ιεραρχική σχεδίαση και τον τμηματικό προγραμματισμό, χρησιμοποιεί τις δομές ακολουθίας , επιλογής και επανάληψης αλλά όχι το go to. Έχει μια είσοδο, στην αρχή και ένα σημείο τερματισμού, στο τέλος

Ποιος είναι ο **σκοπός του δομημένου προγραμματισμού**; (σ.119)

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 και
- να διευκολύνει τις διορθώσεις και τις αλλαγές σε αυτά.

Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος χρειάζεται στο δομημένο προγραμματισμό ή είναι περιττή; (σ.119)

Ποια τα **πλεονεκτήματα του δομημένου προγραμματισμού**; (σ.119)

- 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.

Τι κάνουν οι **μεταγλωττιστές και οι διερμηνευτές**; (σ.121) Ποιες οι διαφορές τους; Ποια τα πλεονεκτήματα-μειονεκτήματα του καθενός;

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. Ποιο γρήγορο στην εκτέλεση του προγράμματος. Έχει το μειονέκτημα, ότι προτού χρησιμοποιηθεί ένα πρόγραμμα, πρέπει να περάσει από τη διαδικασία της μεταγλώττισης και σύνδεσης.

Ο **διερμηνευτής** διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής. Πιο γρήγορο στην υλοποίηση του προγράμματος. Έχει το πλεονέκτημα της άμεσης εκτέλεσης και συνεπώς και της άμεσης διόρθωσης.

Αναλύστε τα **στάδια για τη δημιουργία ενός εκτελέσιμου προγράμματος**; (σ.121)



Περιγράψτε τα **στάδια μεταγλώττισης και σύνδεσης** ενός προγράμματος από το αρχικό πρόγραμμα ως το εκτελέσιμο.

Το αρχικό πρόγραμμα λέγεται **πηγαίο πρόγραμμα** (source), ενώ το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται **αντικείμενο πρόγραμμα** (object).

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. Το πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται **συνδέτης – φορτωτής** (linker-loader).

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.

Περιγράψτε τη διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος σε έναν **διερμηνευτή**

Ο διερμηνευτής διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής

Τι είναι ο **συντάκτης** και τι ο **συνδέτης**;

Ο **συντάκτης** είναι ένας μικρός επεξεργαστής κειμένου, με δυνατότητες όμως που διευκολύνουν τη γρήγορη γραφή των εντολών των προγραμμάτων τόσο στην αρχική σύνταξη των προγραμμάτων όσο και στη διόρθωσή τους.

Ο **συνδέτης** είναι ένα πρόγραμμα που δέχεται το αντικείμενο πρόγραμμα (το οποίο είναι σε γλώσσα μηχανής) αλλά συνήθως δεν είναι σε θέση να εκτελεστεί) και το μετατρέπει σε εκτελέσιμο συνδέοντάς το με άλλα τμήματα προγράμματος, που είναι απαραίτητα για την εκτέλεσή του, που είτε τα γράφει ο προγραμματιστής είτε βρίσκονται στις βιβλιοθήκες (libraries) της γλώσσας.

Ποιες είναι οι **κατηγορίες λαθών** και πότε ανιχνεύονται;

Τα **λογικά** λάθη που είναι τα πλέον σοβαρά και **δύσκολα** στη διόρθωσή 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.

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.

Ποια προγράμματα και εργαλεία πρέπει να περιέχει ένα **προγραμματιστικό περιβάλλον**;

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. Για τη δημιουργία, τη μετάφραση και την εκτέλεση ενός προγράμματος απαιτούνται τουλάχιστον τρία προγράμματα: ο **συντάκτης**, ο **μεταγλωττιστής** και ο **συνδέτης**. Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν αυτά τα προγράμματα με ενιαίο τρόπο

Ποια επιπλέον προγράμματα και εργαλεία πρέπει να περιέχει ένα **σύγχρονο προγραμματιστικό περιβάλλον**;

Τα σύγχρονα ολοκληρωμένα προγραμματιστικά περιβάλλοντα δεν παρέχουν απλώς ένα μεταφραστή μιας γλώσσας προγραμματισμού. Περιέχουν όλα τα προγράμματα και τα **εργαλεία** που απαιτούνται και βοηθούν τη **συγγραφή**, την **εκτέλεση** και κύρια τη **διόρθωση** των προγραμμάτων.