

Κεφάλαιο 6

Εισαγωγή στον Προγραμματισμό

ΠΕΡΙΕΧΟΜΕΝΑ

Η έννοια του προγράμματος

Ιστορική αναδρομή

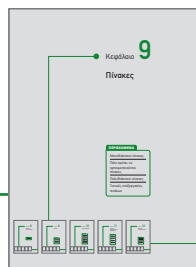
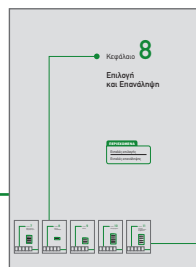
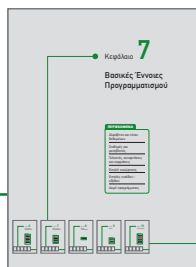
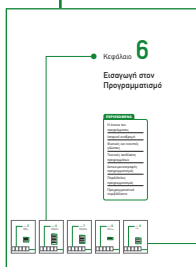
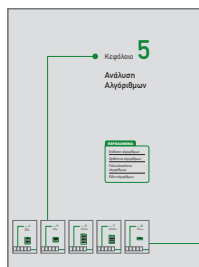
Φυσικές και τεχνητές γλώσσες

Τεχνικές σχεδίασης προγραμμάτων

Αντικειμενοστραφής προγραμματισμός

Παράλληλος προγραμματισμός

Προγραμματιστικά περιβάλλοντα



Εισαγωγή

Στα προηγούμενα κεφάλαια αναφερθήκαμε αναλυτικά στην ανάπτυξη των αλγορίθμων και των διαφόρων τεχνικών. Στα επόμενα κεφάλαια θα ασχοληθούμε με τον προγραμματισμό, δηλαδή τη διατύπωση των αλγορίθμων σε τέτοια μορφή, ώστε να μπορούν να υλοποιηθούν από τον υπολογιστή. Το κεφάλαιο αυτό ασχολείται με γενικές έννοιες που είναι απαραίτητες πριν από την ενασχόληση με τη διαδικασία του προγραμματισμού. Ορίζεται η έννοια του προγράμματος, παρουσιάζεται μία σύντομη ιστορία των γλωσσών προγραμματισμού και των ειδών προγραμματισμού, καθώς και οι τεχνικές που χρησιμοποιούνται για τη σωστή δημιουργία προγραμμάτων. Συγκεκριμένα παρουσιάζονται οι τεχνικές της ιεραρχικής σχεδίασης προγραμμάτων, του τμηματικού προγραμματισμού και κύρια οι αρχές του δομημένου προγραμματισμού. Επίσης περιγράφεται η διαδικασία που ακολουθείται από τη σύνταξη του προγράμματος μέχρι την τελική του εκτέλεση από τον υπολογιστή και τη λήψη των αποτελεσμάτων.

Διδακτικοί στόχοι

Στόχοι του κεφαλαίου αυτού είναι ο μαθητής:

- να ορίζει τι είναι πρόγραμμα και να κατατάσσει και να συγκρίνει τις γλώσσες προγραμματισμού,
- να αναγνωρίζει τα κυριότερα είδη προγραμματισμού και να περιγράφει τα βασικά χαρακτηριστικά των τεχνικών που χρησιμοποιούνται,
- να διατυπώνει τα πλεονεκτήματα του δομημένου προγραμματισμού,
- να περιγράφει τη διαδικασία εκτέλεσης ενός προγράμματος,
- να αναφέρει τα βασικά προγράμματα που περιέχει ένα προγραμματιστικό περιβάλλον.

Προερωτήσεις

- Η δημιουργία του αλγορίθμου αρκεί για να επιλύσουμε ένα πρόβλημα στον υπολογιστή;
- Πώς διαχειρίζεται τις πληροφορίες ο υπολογιστής;
- Γνωρίζεις κάποιες γλώσσες προγραμματισμού;
- Πώς και γιατί εξελίσσονται οι γλώσσες;
- Η ύπαρξη συγκεκριμένων μεθοδολογιών και τεχνικών βοηθάει στην επίλυση των προβλημάτων;

6.3 | Φυσικές και τεχνητές γλώσσες

Οι γλώσσες προγραμματισμού αναπτύχθηκαν για να μπορεί ο προγραμματιστής να δίνει τις εντολές που πρέπει να εκτελέσει ο υπολογιστής. Χρησιμοποιούνται δηλαδή για την επικοινωνία του ανθρώπου και της μηχανής, όπως αντίστοιχα οι φυσικές γλώσσες χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων. Οι γλώσσες προγραμματισμού, που είναι τεχνητές γλώσσες, ακολουθούν τις βασικές έννοιες και αρχές της γλωσσολογίας, επιστήμη που μελετά τις φυσικές γλώσσες.

Μία γλώσσα προσδιορίζεται από το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τέλος τη σημασιολογία της.

Το αλφάβητο

Αλφάβητο μίας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα.

Για παράδειγμα, 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 τη 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 γλώσσας δεν είναι αποδεκτή.

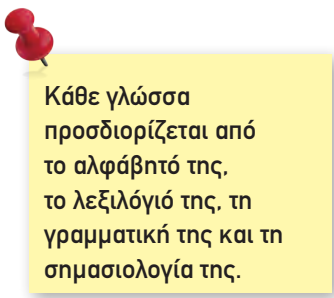
Συντακτικό είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη δημιουργία προτάσεων.

Η 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.



Κάθε γλώσσα προσδιορίζεται από το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τη σημασιολογία της.

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 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.

Ωστόσο συχνά οι γλώσσες προγραμματισμού βελτιώνονται και μεταβάλλονται από τους δημιουργούς τους, με σκοπό να διορθωθούν αδυναμίες ή να καλύψουν μεγαλύτερο εύρος εφαρμογών ή τέλος να ακολουθήσουν τις νέες εξελίξεις. Οι γλώσσες προγραμματισμού αλλάζουν σε επίπεδο διαλέκτου (για παράδειγμα GW-Basic και QuickBasic) ή σε επίπεδο επέκτασης (για παράδειγμα Basic και Visual Basic).

6.4 | Τεχνικές σχεδίασης προγραμμάτων

Από την αρχή της εμφάνισης των υπολογιστών γίνονται συνεχείς προσπάθειες ανάπτυξης μεθοδολογιών και τεχνικών προγραμματισμού, που θα εξασφαλίζουν τη δημιουργία απλών και κομψών προγραμμάτων, την εύκολη γραφή τους όσο και την κατανόησή τους.

6.4.1 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) 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 άπλες, ώστε να επιλυθούν εύκολα.

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.

Για την υποβοήθηση της ιεραρχικής σχεδίασης χρησιμοποιούνται διάφορες διαγραμματικές τεχνικές, όπως για παράδειγμα το διάγραμμα του σχήματος 6.9:

6.4.2 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.

Η σωστή διαίρεση του αρχικού προβλήματος σε υποπροβλήματα και κατά συνέπεια του αρχικού προγράμματος σε τμήματα προγράμματος

Η ιεραρχική σχεδίαση ή ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα.



Ανακεφαλαίωση

Δημιουργία προγράμματος είναι η μετατροπή του αλγορίθμου που επιλύει ένα πρόβλημα σε εντολές προγράμματος. Οι εντολές γράφονται σε κάποια από τις εκατοντάδες γλώσσες προγραμματισμού, που έχουν αναπτυχθεί με σκοπό να διευκολύνουν την επίλυση συγκεκριμένου τύπου προβλημάτων. Η επιλογή της καταλληλότερης γλώσσας εξαρτάται από το είδος της εφαρμογής, το υπολογιστικό περιβάλλον που θα εκτελεστεί, τις γνώσεις και τις προτιμήσεις του προγραμματιστή.

Οι τεχνικές που χρησιμοποιούνται για την ανάπτυξη προγραμμάτων είναι της ιεραρχικής σχεδίασης, του τμηματικού προγραμματισμού και του δομημένου προγραμματισμού, που επιτρέπουν τη δημιουργία προγραμμάτων κατανοητών και απλών, ενώ διευκολύνουν τη διόρθωση και τη συντήρηση των εφαρμογών. Ένα είδος προγραμματισμού που γνωρίζει ιδιαίτερη άνθηση τελευταία είναι ο αντικειμενοστραφής προγραμματισμός.

Κάθε πρόγραμμα για να εκτελεστεί από τον υπολογιστή χρειάζεται πρώτα να μετατραπεί σε μορφή κατανοητή από αυτόν. Η μετατροπή αυτή γίνεται από τους μεταγλωττιστές ή τους διερμηνευτές, οι οποίοι επισημαίνουν και τα συντακτικά λάθη, που έχει κάθε πρόγραμμα. Η σύνταξη του προγράμματος, η μετάφραση, η διόρθωση των λαθών και η εκτέλεσή του γίνεται με τα ολοκληρωμένα προγραμματιστικά περιβάλλοντα που διαθέτουν πολλά εργαλεία για την υποβοήθηση της ανάπτυξης των εφαρμογών.

Λέξεις-κλειδιά

Πρόγραμμα,
Γλώσσα μηχανής,
Συμβολική γλώσσα,
Γλώσσες υψηλού
επιπέδου,
Τμηματικός
προγραμματισμός,
Δομημένος
προγραμματισμός,
Αντικειμενοστραφής
προγραμματισμός,
Μεταγλωττιστής,
Διερμηνευτής,
Προγραμματιστικό
περιβάλλον

Ερωτήσεις - Θέματα για συζήτηση

- ~~1. Τι ονομάζεται πρόγραμμα;~~
- ~~2. Τι είναι οι γλώσσες μηχανής;~~
- ~~3. Ποιες οι διαφορές των γλωσσών υψηλού επιπέδου από αυτές χαμηλού επιπέδου;~~
- ~~4. Ποιες γλώσσες υψηλού επιπέδου γνωρίζεις;~~
- ~~5. Τι ονομάζουμε οπτικό προγραμματισμό και τι οδηγούμενο από τα γεγονότα;~~
6. Πώς προσδιορίζεται μία φυσική γλώσσα;
7. Ποιες οι κυριότερες διαφορές των φυσικών και των τεχνητών γλωσσών;
- ~~8. Πώς γίνεται η παράσταση της ιεραρχικής σχεδίασης προγράμματος;~~
- ~~9. Ποιες οι αρχές του δομημένου προγραμματισμού;~~
- ~~10. Ποια τα πλεονεκτήματα του δομημένου προγραμματισμού;~~
- ~~11. Τι ονομάζεται αντικειμενοστραφής προγραμματισμός;~~
- ~~12. Ποια η διαδικασία για τη μετάφραση και εκτέλεση ενός προγράμματος;~~
- ~~13. Ποιες οι διαφορές μεταγλωττιστή και διερμηνευτή;~~
- ~~14. Ποια προγράμματα και εργαλεία περιέχει ένα προγραμματιστικό περιβάλλον;~~

Βιβλιογραφία

1. Ph. Breton, *Ιστορία της Πληροφορικής*, Εκδόσεις Δίαυλος, Αθήνα.
2. Γ. Μπαμπινιώτης, *Θεωρητική Γλωσσολογία*, Αθήνα, 1986.
3. Χρ. Κοίλιας-Στρ. Καλαφατούδης, *Το πρώτο βιβλίο της Πληροφορικής*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
4. *Εγκυκλοπαίδεια Πληροφορικής και Τεχνολογίας Υπολογιστών*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1986.
5. Αθ. Τσουροπλής-Στ. Κλημόπουλος, *Από τη FORTRAN 77 στη FORTRAN 90*, Εκδόσεις Πελεκάνος, Αθήνα, 1995.
6. Χ. Κοίλιας-Στρ. Μαραγκός, *Η γλώσσα COBOL και οι εφαρμογές της*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
7. Κ. Μαρινάκης-Ν. Ιωαννίδης, *Structure & Advanced COBOL*, Εκδόσεις Έλιξ, Αθήνα, 1992.
8. Χ. Κοίλιας-Αλ. Τομαράς, *GWBASIC Θεωρία και Εφαρμογές*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
9. Μ. Κατζουράκη-Μ. Γεργατσούλης-Σ. Κόκκοτος, *ΠΡΟγραμματίζοντας στη ΛΟΓική*, Έκδοση ΕΠΥ, Αθήνα, 1991.
10. Αικ. Γεωργόπουλου, *LOGO Βήμα προς Βήμα*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1991.
11. Αλ. Τομαράς, *C Θεωρία και Πράξη*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1995.
12. Μ. Μαλιάππη, *SQL Περιβάλλοντα Ανάπτυξης Εφαρμογών 4ης Γενιάς*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1995.
13. E. Horowitz, *Βασικές αρχές γλωσσών προγραμματισμού*, Κλειδάριθμος, Αθήνα, 1995.
14. R. Shackelford, *Introduction to Computing and Algorithms*, Addison-Wesley, USA, 1998.
15. W. Hutching-H. Somers, *An Introduction to Machine Translation*, Academic Press, London, 1992.