

**Ανάπτυξη Εφαρμογών  
σε Προγραμματιστικό Περιβάλλον**

**Τόμος 3ος**

**Ομάδα Συγγραφής**

**ΑΘΗΝΑ ΒΑΚΑΛΗ,   
Λέκτωρ Πληροφορικής ΑΠΘ**

**ΗΛΙΑΣ ΓΙΑΝΝΟΠΟΥΛΟΣ,   
Μηχανικός Πληροφορικής**

**ΝΕΣΤΩΡ ΙΩΑΝΝΙΔΗΣ,   
Επίκουρος Καθηγητής Τμήματος**

**ΧΡΗΣΤΟΣ ΚΟΙΛΙΑΣ,   
Επίκουρος Καθηγητής Τμήματος Πληροφορικής ΤΕΙ Αθήνας**

**ΚΩΝ/ΝΟΣ ΜΑΛΑΜΑΣ, M.Sc. Πληροφορικής, Σύμβουλος Επιχειρήσεων,**

**ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΠΟΥΛΟΣ, Αναπληρωτής Καθηγητής Τμήματος Πληροφορικής ΑΠΘ**

**ΠΑΝΑΓΙΩΤΗΣ ΠΟΛΙΤΗΣ,   
Δρ. Διδακτικής Πληροφορικής, Καθηγητής ΠΕ 19**

**Υπεύθυνος για το Παιδαγωγικό Ινστιτούτο  
ΓΕΩΡΓΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ, ΣΥΜΒΟΥΛΟΣ Π.Ι. (κατά τη συγγραφή)**

**Υπεύθυνος Μαθήματος  
ΑΔΑΜ Κ. ΑΓΓΕΛΗΣ,   
Πάρεδρος Πληροφορικής Π.Ι.**

**Επιτροπή Αξιολόγησης  
ΚΩΝ/ΝΟΣ ΓΙΑΛΟΥΡΗΣ,   
Καθηγητής ΠΕ 19   
ΕΥΑΓΓΕΛΙΑ ΓΟΥΛΗ,   
Καθηγήτρια ΠΕ 19   
ΔΗΜΗΤΡΙΟΣ ΔΕΣΠΟΤΗΣ, Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς   
ΚΩΝ/ΝΟΣ ΖΑΧΑΡΗΣ,   
ΠΛΗΝΕΤ Καρδίτσας   
ΑΛΕΞΑΝΔΡΟΣ ΣΙΔΕΡΙΔΗΣ, Καθηγητής Γεωπονικού Πανεπιστημίου Αθηνών**

**Εικονογράφηση   
ΑΓΓΕΛΟΣ ΑΓΙΟΣΤΡΑΤΙΤΗΣ**

**Ηλεκτρονική σελιδοποίηση   
ΑΝΝΑ ΑΡΓΥΡΟΠΟΥΛΟΥ - ΧΡΗΣΤΟΣ ΠΙΓΚΑΣ**

**Εξώφυλλο:   
ΣΠΥΡΟΣ ΣΙΑΚΑΣ - ΝΙΚΟΣ ΓΕΩΡΓΙΟΥ**

**Επιμέλεια:   
ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, Στουρνάρη 49Α, 106 82, Αθήνα, Τηλ. 38.45.594**

**Φορέας:**  
**ΕΛΛΗΝΙΚΗ ΕΤΑΙΡΙΑ ΕΠΙΣΤΗΜΟΝΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ (ΕΠΥ), Μαυρομιχάλη 16, Αθήνα,   
τηλ.: 3645274, e.mail:epy@epy.gr**

**Συντονιστές έργου:**

**Σπ. Μπακογιάννης, πρόεδρος Δ.σ. Βασ. Μιχαλακόπουλος, μέλος Δ.Σ.**

**Ενέργεια 1.1.α: «Προγράμματα Βιβλία»**

**Επιστημονικός Υπεύθυνος Ενέργειας**

**Θεόδωρος Γ. Εξαρχάκος,**

**Καθηγητής του Πανεπιστημίου**

**Αθηνών, Πρόεδρος του Παιδαγωγικού Ινστιτούτου**

**Έργο No 15: «Αναμόρφωση / εκ νέου σύνταξη και συγγραφή Προγραμμάτων Σπουδών και Σχολικών Βιβλίων για το Ενιαίο Λύκειο»**

**Επιστημονικός Υπεύθυνος Έργου**

**Γιάννης Σαλβαράς,**

**Επίκουρος Καθηγητής του**

**Δημοκρίτειου Πανεπιστημίου Θράκης**

**Καλλιτεχνικός Υπεύθυνος Έργου**

**Σπύρος I. Παπασπύρου**

**Καθηγητής Εφαρμογών του ΤΕΙ Ηπείρου**

**Η επανέκδοση του παρόντος βιβλίου πραγματοποιήθηκε από το Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος» μέσω ψηφιακής μακέτας, η οποία δημιουργήθηκε με χρηματοδότηση από το ΕΣΠΑ / ΕΠ «Εκπαίδευση & Διά Βίου Μάθηση» / Πράξη «ΣΤΗΡΙΖΩ».**

**Οι αλλαγές που ενσωματώθηκαν στην παρούσα επανέκδοση έγιναν με βάση τις διορθώσεις του Παιδαγωγικού Ινστιτούτου**

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ INΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ**

**Α. ΒΑΚΑΛΗ, Η. ΓΙΑΝΝΟΠΟΥΛΟΣ, Ν. ΙΩΑΝΝΙΔΗΣ, Χ. ΚΟΙΛΙΑΣ, Κ. ΜΑΛΑΜΑΣ, Ι. ΜΑΝΩΛΟΠΟΥΛΟΣ, Π. ΠΟΛΙΤΗΣ**

**Η συγγραφή και η επιμέλεια του βιβλίου πραγματοποιήθηκε υπό την αιγίδα του Παιδαγωγικού Ινστιτούτου**

**Ανάπτυξη Εφαρμογών**

**σε Προγραμματιστικό Περιβάλλον**

**Βιβλίο Μαθητή**

**Γ' Γενικού Λυκείου**

**(Τεχνολογικής Κατεύθυνσης)**

**ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ   
ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»**

**ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΒΙΒΛΙΟΥ ΓΙΑ ΜΑΘΗΤΕΣ ΜΕ ΜΕΙΩΜΕΝΗ ΟΡΑΣΗ**

**Ομάδα εργασίας για το Ινστιτούτου Εκπαιδευτικής**

**Πολιτικής**

**Προσαρμογή: Μαρία Γκλεζάκου, Εκπαιδευτικός**

**Eπιμέλεια: Αρετή Ανδριώτου, Εκπαιδευτικός**

**Επιστημονικός υπεύθυνος: Βασίλης Κουρμπέτης,**

**Σύμβουλος Α΄ του Υ.ΠΟ.ΠΑΙ.Θ**

**Υπεύθυνη του έργου: Μαρία Γελαστοπούλου,**

**M.Ed. Ειδικής Αγωγής**

**Τεχνική υποστήριξη: Κωνσταντίνος Γκυρτής,**

**Δρ. Πληροφορικής**

**Κεφάλαιο 10**

**Υποπρογράμματα**

**ΠΕΡΙΕΧΟΜΕΝΑ**

|  |  |
| --- | --- |
|  | **Τμηματικός προγραμματισμός** |
|  | **Χαρακτηριστικά των υποπρογραμμάτων** |
|  | **Πλεονεκτήματα του τμηματικού προγραμματισμού** |
|  | **Παράμετροι** |
|  | **Διαδικασίες και συναρτήσεις** |
|  | **Εμβέλεια μεταβλητών - σταθερών**  **5 / 169** |
|  | **Αναδρομή** |

**• Εισαγωγή**

**Η επίλυση ενός προβλήματος διευκολύνεται με τη διαίρεση του σε μικρότερα υποπροβλήματα. Η επίλυση των υποπροβλημάτων αυτών οδηγεί στην επίλυση του αρχικού προβλήματος. Ο τμηματικός προγραμματι-σμός, η διαίρεση δηλαδή ενός προγράμματος σε υποπρογράμματα υλοποιεί αυτήν την ιδέα στον προγραμματισμό. Το κεφάλαιο αυτό ασχολείται με τις αρχές του τμηματικού προγραμματισμού, τα είδη των υποπρογραμμάτων που υποστηρίζει η ΓΛΩΣΣΑ, τις διαδικασίες και τις συναρτήσεις καθώς και τον τρόπο που τα υποπρογράμματα αυτά επικοινωνούν μεταξύ τους. Τέλος παρουσιάζεται και αναλύεται ο τρόπος υλοποίησης αναδρομικών αλγορίθμων με χρήση αναδρομικών υποπρογραμμάτων.**

**• Διδακτικοί στόχοι**

**Να είναι σε θέση ο μαθητής:**

* **Να αναλύει ένα σύνθετο πρόγραμμα σε απλά υποπρογράμματα.**
* **Να διακρίνει τις συναρτήσεις από τις διαδικασίες και να επιλέγει τη χρήση διαδικασίας ή συνάρτησης για την υλοποίηση ενός υποπρογράμματος.**

**6 / 170**

* **Να περιγράφει τη δομή των υποπρογραμμάτων και να χρησιμοποιεί παραμέτρους για την επικοινωνία τους.**

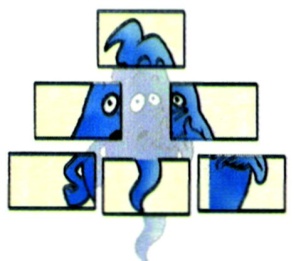
**7 / 170**

* **Να καθορίζει τις περιοχές εμβέλειας των παραμέ-τρων.**
* **Να συντάσσει αναδρομικά υποπρογράμματα και να συγκρίνει αναδρομικές και επαναληπτικές διαδι-κασίες.**

**• Προερωτήσεις**

* **Η δόμηση ενός προγράμματος με τη μορφή ενός συνόλου μικρότερων προγραμμάτων βοηθάει τον προγραμματιστή στην ανάπτυξη ενός σύνθετου προγράμματος;**
* **Νομίζετε ότι όλα τα είδη τμημάτων προγραμμάτων επιτελούν την ίδια εργασία;**
* **Πώς μπορεί να οργανώνεται ένα πρόγραμμα σε μικρότερα προγράμματα;**
* **τα επιμέρους προγράμματα να επικοινωνούν μεταξύ τους;**
* **Γνωρίζεις από την άλγεβρα τους αναδρομικούς τύπους; Ποια τα πλεονεκτήματα τους;**

|  |  |
| --- | --- |
| **10.1** | **Τμηματικός προγραμματισμός** |

****

**Τα προβλήματα που αντιμετωπίστηκαν στα προηγούμενα κεφάλαια, ήταν αρκετά απλά, ώστε να μπορούν να αναπτυχθούν σωστά σε ένα και μόνο πρόγραμμα. Όπως αναφέρθηκε στο κεφάλαιο 6, ο καλύτερος τρόπος για να αντιμετωπισθούν σύνθετα προβλήματα και να γραφούν τα αντίστοιχα, είναι η ιεραρχική προσέγγιση, η ανάπτυξη του προγράμματος από επάνω προς τα κάτω (top-down). Κάθε πρόβλημα διαιρείται σε μικρότερα επιμέρους προβλήματα και κάθε ένα από αυτά τα προ γράμματα διαιρείται σε ακόμα απλούστερα και μικρότερα. Στο τέλος τα επιμέρους υποπροβλήματα είναι αρκετά απλά, ώστε οι αντίστοιχοι αλγόριθμοι και τα αντίστοιχα τμήματα προγράμματος να μπορούν να σχεδιασθούν και να γραφούν εύκολα. Ο τελικός αλγόριθμος του προβλήματος ανάγεται σε πολλούς απλούστερους επιμέρους αλγόριθμους και το τελικό πρόγραμμα σε πολλά απλούστερα τμήματα προγράμματος.**

**Η τεχνική του τμηματικού προγραμματισμού είναι ένα από τα βασικότερα συστατικά του δομημένου προγραμματισμού, ο οποίος εξασφαλίζει σε μεγάλο βαθμό την επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων.**

**ΟΡΙΣΜΟΣ**

**Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.**

**8 / 171**

**ΠΑΡΑΔΕΙΓΜΑ 1**

**Ας μελετήσουμε κατ' αρχήν το πρόβλημα που μας απασχόλησε στο πρώτο κεφάλαιο του βιβλίου, την αξιολόγηση των αποτελεσμάτων των μαθητών Γ' Λυκείου στα μαθήματα ειδικότητας.**

**Το σύνθετο αυτό πρόβλημα για να αντιμετωπισθεί πιο εύκολα πρέπει να αναλυθεί σε επιμέρους μικρότερα προβλήματα.**

**Συγκεκριμένα τα τρία βασικά διαφορετικά τμήματα είναι:**

* + - **Εισαγωγή δεδομένων**
    - **Επεξεργασία δεδομένων**

**9 / 171**

* + - **Εκτύπωση αποτελεσμάτων**

**Τα τρία αυτά τμήματα μπορούν να αναλυθούν περισ-σότερο. Συγκεκριμένα:**

**Εισαγωγή δεδομένων**

* **Καταχώριση δεδομένων**
* **Έλεγχος δεδομένων**

**Επεξεργασία δεδομένων**

* **Υπολογισμός μέσης τιμής**
* **Υπολογισμός τυπικής απόκλισης**
* **Κατανομή συχνοτήτων**
* **Δημιουργία γραφικών παραστάσεων**

**Εκτύπωση αποτελεσμάτων**

* **Εκτύπωση πινάκων συχνοτήτων**
* **Εκτύπωση γραφικών παραστάσεων**

**Όπως φαίνεται το αρχικό πρόβλημα διασπάστηκε σε αρκετά απλούστερα υποπροβλήματα. Η δημιουργία λοιπόν του τελικού προγράμματος ανάγεται στη δημιουργία των επί μέρους τμημάτων προγραμμάτων ή ενοτήτων και τη σύνδεση αυτών μεταξύ τους. Μερικά από αυτά τα τμήματα όπως ο υπολογισμός της μέσης τιμής ή της τυπικής απόκλισης, έχουν ήδη αντιμετω-πιστεί στο προηγούμενο κεφάλαιο, που σημαίνει ότι μπορούμε να εκμεταλλευτούμε τα προγράμματα που ήδη έχουμε γράψει μειώνοντας έτσι την εργασία για την επίλυση του προβλήματος.**

**Η παράσταση της ανάλυσης του προβλήματος μπορεί να γίνει γραφικά με το διάγραμμα του σχήματος 10.1**

**Η έννοια του τμηματικού προγραμματισμού έχει ήδη αποτυπωθεί και σε προηγούμενα κεφάλαια. Για παράδειγμα στο πρόγραμμα υπολογισμού των θερμο-κρασιών του προηγούμενου κεφαλαίου (παράδειγμα 3) το τμήμα της εισαγωγής δεδομένων έχει ξεχωρίσει από το τμήμα υπολογισμών σε αντίθεση με τα παραδείγ-ματα 1 και 2 που είναι ενιαία.**

**10 / 172**

**Όταν ένα τμήμα προγράμματος επιτελεί ένα αυτόνομο έργο και έχει γραφεί χωριστά από το υπόλοιπο πρόγραμμα, τότε αναφερόμαστε σε υποπρόγραμμα (subprogram).**

**Σχ. 10.1 Διαγραμματική απεικόνιση της ανάλυσης προβλήματος**

**000 / 000**

**11 / 172**

|  |  |
| --- | --- |
| **10.2** | **Χαρακτηριστικά των υποπρογραμμάτων** |

**Ο χωρισμός ενός προγράμματος σε υποπρογράμματα προϋποθέτει την ανάλυση του αρχικού προβλήματος σε μικρότερα υποπροβλήματα, τα οποία να μπορούν να αντιμετωπισθούν ανεξάρτητα το ένα από το άλλο. Η ανάλυση όμως αυτή δεν είναι πάντα εύκολη και όπως ισχύει γενικά στον προγραμματισμό, δεν υπάρχουν συγκεκριμένοι κανόνες για την επιτυχή ανάλυση. Η δυσκολία δε αυξάνεται όσο πιο μεγάλο και πιο σύνθετο είναι το πρόβλημα. Η σωστή εφαρμογή του τμηματικού προγραμματισμού απαιτεί μελέτη στην ανάλυση του προβλήματος, εμπειρία στον προγραμματισμό, ταλέντο και φυσικά γνώσεις.**

**Υπάρχουν πάντως τρεις ιδιότητες που πρέπει να διακρίνουν τα υποπρογράμματα :**

1. **Κάθε υποπρόγραμμα έχει μόνο μία είσοδο και μία έξοδο. Στην πραγματικότητα κάθε υποπρόγραμμα ενεργοποιείται με την είσοδο σε αυτό που γίνεται πάντοτε από την αρχή του, εκτελεί ορισμένες ενέργειες, και απενεργοποιείται με την έξοδο από αυτό που γίνεται πάντοτε από το τέλος του.**
2. **Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα. Αυτό σημαίνει ότι κάθε υποπρό-γραμμα μπορεί να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί αυτόνομα χωρίς να επηρεαστούν άλλα υποπρογράμματα. Στην πράξη βέβαια η απόλυτη ανεξαρτησία είναι δύσκολο να επιτευχθεί.**

**12 / 173**

1. **Κάθε υποπρόγραμμα πρέπει να μην είναι πολύ μεγάλο. Η έννοια του μεγάλου προγράμματος είναι υποκειμενική, αλλά πρέπει κάθε υποπρόγραμμα να είναι τόσο, ώστε να είναι εύκολα κατανοητό για να μπορεί να ελέγχεται. Γενικά κάθε υποπρόγραμμα πρέπει να εκτελεί μόνο μία λειτουργία. Αν εκτελεί περισσότερες λειτουργίες, τότε συνήθως μπορεί και πρέπει να διασπαστεί σε ακόμη μικρότερα υποπρογράμματα.**

|  |  |
| --- | --- |
| **10.3** | **Πλεονεκτήματα του τμηματικού Προγραμματισμού** |

**Η χρήση υποπρογραμμάτων σε ένα πρόγραμμα παρουσιάζει πολλά πλεονεκτήματα που αναφέρθηκαν συνοπτικά στο κεφάλαιο 6. Η σωστή χρήση του τμηματικού προγραμματισμού, δηλαδή ο σωστός χωρισμός ενός σύνθετου προγράμματος σε υποπρογράμματα εξασφαλίζει τέσσερα βασικά χαρακτηριστικά του σωστού προγραμματισμού:**

1. **Διευκολύνει την ανάπτυξη του αλγορίθμου και του αντιστοίχου προγράμματος.**

**13 / 173**

**Επιτρέπει την εξέταση και την επίλυση απλών προ-βλημάτων και όχι στην αντιμετώπιση του συνολικού προβλήματος. Με τη σταδιακή επίλυση των υποπρο-βλημάτων και τη δημιουργία των αντιστοίχων υποπρογραμμάτων τελικά επιλύεται το συνολικό πρόβλημα.**

1. **Διευκολύνει την κατανόηση και διόρθωση του προγράμματος.**

**Ο χωρισμός του προγράμματος σε μικρότερα αυτοτελή τμήματα επιτρέπει τη γρήγορη διόρθωση ενός συγκεκριμένου τμήματός του χωρίς οι αλλαγές αυτές να επηρεάσουν όλο το υπόλοιπο πρόγραμμα. Επίσης διευκολύνει οποιονδήποτε χρειαστεί να διαβάσει και να κατανοήσει τον τρόπο που λει-τουργεί το πρόγραμμα. Όπως έχει πολλές φορές τονιστεί αυτό είναι πολύ σημαντικό χαρακτηριστικό του σωστού προγραμματισμού, αφού ένα μεγάλο πρόγραμμα στον κύκλο της ζωής του χρειάζεται να συντηρηθεί από διαφορετικούς προγραμματιστές.**

**14 / 174**

1. **Απαιτεί λιγότερο χρόνο και προσπάθεια στη συγγραφή του προγράμματος.**

**Πολύ συχνά χρειάζεται η ίδια λειτουργία σε διαφορετικά σημεία ενός προγράμματος. Από τη στιγμή που ένα υποπρόγραμμα έχει γραφεί, μπορεί το ίδιο να καλείται από πολλά σημεία του προγράμ-ματος. Έτσι μειώνονται το μέγεθος του προγράμ-ματος, ο χρόνος που απαιτείται για τη συγγραφή του και οι πιθανότητες λάθους, ενώ ταυτόχρονα το πρόγραμμα γίνεται πιο εύληπτο και κατανοητό.**

1. **Επεκτείνει τις δυνατότητες των γλωσσών προγραμματισμού.**

**Ένα υποπρόγραμμα που έχει γραφεί μπορεί να χρησιμοποιηθεί πολύ εύκολα και σε άλλα προ-γράμματα. Από τη στιγμή που έχει δημιουργηθεί, η χρήση του δεν διαφέρει από τη χρήση των ενσω-ματωμένων συναρτήσεων που παρέχει η γλώσσα προγραμματισμού, όπως για τον υπολογισμό του ημίτονου ή του συνημίτονου ή την εντολή με την οποία εκτελεί μία συγκεκριμένη διαδικασία, για παράδειγμα γράφει στην οθόνη (εντολή ΓΡΑΨΕ). Αν λοιπόν χρειάζεται συχνά κάποια λειτουργία που δεν υποστηρίζεται απευθείας από τη γλώσσα, όπως για παράδειγμα η εύρεση του μικρότερου δύο αριθμών, τότε μπορεί να γραφεί το αντίστοιχο υποπρόγραμμα. Η συγγραφή πολλών υποπρογραμμάτων και η δημιουργία βιβλιοθηκών με αυτά, ουσιαστικά επεκτείνουν την ίδια τη γλώσσα προγραμματισμού.**

|  |  |
| --- | --- |
| **10.4** | **Παράμετροι** |

**Τα υποπρογράμματα ενεργοποιούνται από κάποιο άλλο πρόγραμμα ή υποπρόγραμμα για να εκτελέσουν συγκεκριμένες λειτουργίες. Κάθε υποπρόγραμμα για να ενεργοποιηθεί καλείται, όπως λέγεται, από ένα άλλο υποπρόγραμμα ή το αρχικό πρόγραμμα, το οποίο ονομάζεται κύριο πρόγραμμα. Το υποπρόγραμμα είναι αυτόνομο και ανεξάρτητο τμήμα προγράμματος, αλλά συχνά πρέπει να επικοινωνεί με το υπόλοιπο πρόγραμμα. Συνήθως δέχεται τιμές από το τμήμα προγράμματος που το καλεί και μετά την εκτέλεση επιστρέφει σε αυτό νέες τιμές, αποτελέσματα.**

**Οι τιμές αυτές που περνούν από το ένα υποπρόγραμμα στο άλλο λέγονται παράμετροι.**

**Οι παράμετροι λοιπόν είναι σαν τις κοινές μεταβλητές ενός προγράμματος με μία ουσιώδη διαφορά, χρησιμο-ποιούνται για να περνούν τιμές στα υποπρογράμματα.**

**15 / 174**

**ΟΡΙΣΜΟΣ**

**Μία παράμετρος είναι μία μεταβλητή που επιτρέπει το πέρασμα της τιμής της από ένα τμήμα προγράμματος σε ένα άλλο.**

**Παραδείγματα παραμέτρων, καθώς και ο τρόπος που περνούν οι τιμές προς και από το υποπρόγραμμα, θα δοθούν στη συνέχεια.**

|  |  |
| --- | --- |
| **10.5** | **Διαδικασίες και συναρτήσεις** |

**Υπάρχουν δύο ειδών υποπρογράμματα, οι διαδικασίες και οι συναρτήσεις. Το είδος κάθε υποπρογράμματος καθορίζεται από το είδος της λειτουργίας που καλείται να επιτελέσει.**

**Οι διαδικασίες μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα. Να εισάγουν δεδομένα, να εκτελέσουν υπολογισμούς, να μεταβάλλουν τις τιμές των μετα-βλητών και να τυπώσουν αποτελέσματα. Με τη χρήση των παραμέτρων, αυτές τις τιμές μπορούν να τις μεταφέρουν και στα άλλα υποπρογράμματα.**

**Αντίθετα η λειτουργία των συναρτήσεων είναι πιο περιορισμένη. Οι συναρτήσεις υπολογίζουν μόνο μία τιμή, αριθμητική, χαρακτήρα ή λογική και μόνο αυτήν επιστρέφουν στο υποπρόγραμμα που την κάλεσε. Οι συναρτήσεις μοιάζουν με τις συναρτήσεις των μαθη-ματικών και η χρήση τους είναι όμοια με τη χρήση των ενσωματωμένων συναρτήσεων που υποστηρίζει η γλώσσα προγραμματισμού.**

**16 / 175**

**Ο τρόπος κλήσης καθώς και ο τρόπος σύνταξης των δύο αυτών τύπων των υποπρογραμμάτων είναι διαφορετικός. Τόσο οι συναρτήσεις όσο και οι διαδικασίες τοποθετούνται μετά το τέλος του κυρίου προγράμματος.**

**Οι συναρτήσεις εκτελούνται απλά με την εμφάνιση**

**του ονόματός τους σε οποιαδήποτε έκφραση, ενώ για να εκτελεστούν οι διαδικασίες χρησιμοποιείται η ειδική εντολή ΚΑΛΕΣΕ και το όνομα της διαδικασίας.**

**ΟΡΙΣΜΟΣ**

**Η συνάρτηση είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομά της (όπως οι μαθηματικές συναρτήσεις).**

**Η διαδικασία είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος.**

**Ας δούμε ένα απλό παράδειγμα χρήσης διαδικασιών και συναρτήσεων.**

**ΠΑΡΑΔΕΙΓΜΑ 2**

**Να γραφεί πρόγραμμα, το οποίο υπολογίζει το εμβαδό του κύκλου από την ακτίνα του.**

**Το πρόγραμμα εκτελεί τρεις συγκεκριμένες απλές λειτουργίες.**

**α) Διαβάζει τα δεδομένα, την ακτίνα η οποία πρέπει να είναι θετικός αριθμός**

**17 / 175-176**

**β) Υπολογίζει το εμβαδό (Ε=πr2)**

**γ) Τυπώνει το αποτέλεσμα, το εμβαδό, Ε**

**Αν και το πρόγραμμα είναι πολύ απλό και μπορεί κάλλιστα να γραφεί χωρίς τη χρήση υποπρογραμ-μάτων, ας το διασπάσουμε σε τρία υποπρογράμματα που εκτελούν τις τρεις παραπάνω λειτουργίες.**

**Τα πρώτο υποπρόγραμμα πρέπει να διαβάζει την ακτίνα και να την επιστρέφει στο κύριο πρόγραμμα. Αφού το υποπρόγραμμα πρέπει να διαβάζει δεδομένα, υλοποιείται με διαδικασία. Η διαδικασία αυτή που ονομάζεται Είσοδος\_δεδομένων, δέχεται από το πληκτρολόγιο την τιμή της ακτίνας που την καταχωρεί στη μεταβλητή Αριθμός και έχει ως εξής:**

ΔΙΑΔΙΚΑΣΙΑ  **Είσοδος\_δεδομένων(Αριθμός)**  
ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ:  **Αριθμός**

ΑΡΧΗ  
 ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ **‘Δώσε την ακτίνα’**

ΔΙΑΒΑΣΕ  **Αριθμός**

ΜΕΧΡΙΣ\_ΟΤΟΥ **Αριθμός > 0**  
ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

**Το δεύτερο πρέπει να υπολογίζει το εμβαδό και να επιστρέφει την τιμή στο κύριο πρόγραμμα. Το υποπρόγραμμα αυτό την τιμή της ακτίνας και επιστρέφει μόνο μία τιμή, την τιμή του Εμβαδού. Μπορεί λοιπόν να υλοποιηθεί με μία συνάρτηση, η οποία επιστρέφει έναν πραγματικό αριθμό.**

**18 / 176**

**Η συνάρτηση Εμβαδό\_κύκλου(R) δέχεται έναν πραγματικό αριθμό και υπολογίζει το εμβαδό που επίσης είναι ένας πραγματικός αριθμός. Το είδος της συνάρτησης, δηλαδή η τιμή που επιστρέφει, δηλώνεται στην αρχή της συνάρτησης.**

ΣΥΝΑΡΤΗΣΗ  **Εμβαδό\_κύκλου(R):** ΠΡΑΓΜΑΤΙΚΗ  
ΣΤΑΘΕΡΕΣ:

**Π=3.14**

ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ:  **R**

ΑΡΧΗ  
 **Εμβαδό\_κύκλου** 🡨 **Π\* R^2**

ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

**Το τρίτο υποπρόγραμμα τυπώνει το αποτέλεσμα. Εφό-σον απαιτείται από αυτό η εκτέλεση της λειτουργίας της εκτύπωσης, πρέπει να υλοποιηθεί με διαδικασία. Η διαδικασία Εκτύπωση δέχεται από το κύριο πρόγραμμα μια τιμή στη μεταβλητή Αποτέλεσμα και την εκτυπώνει.**

ΔΙΑΔΙΚΑΣΙΑ  **Εκτύπωση (Αποτέλεσμα)**  
ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ:  **Αποτέλεσμα**

ΑΡΧΗ  
 ΓΡΑΨΕ **‘Το εμβαδό του κύκλου είναι :’, Αποτέλεσμα**

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

**Το κύριο πρόγραμμα που καλεί όλα τα υποπρογράμ-ματα έχει ως εξής:**

ΠΡΟΓΡΑΜΜΑ **Παράδειγμα\_2**

ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ:  **R, Εμ**ΑΡΧΗ  
 ΚΑΛΕΣΕ **Είσοδος\_δεδομένων(R)**

**19 / 176-177**

**Εμ** 🡨 **Εμβαδό\_κύκλου (R)**

ΚΑΛΕΣΕ **Εκτύπωση (Εμ)**

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

**Το πρόγραμμα πλέον έχει ολοκληρωθεί. Όταν εκτελε-στεί, θα ζητήσει από το χρήστη να εισάγει μια τιμή για την ακτίνα και θα εμφανίσει το εμβαδό του κύκλου. Αν η εισαγόμενη τιμή για την ακτίνα είναι 10, τότε θα η οθόνη θα παρουσιάζει τα εξής:**

**Δώσε την ακτίνα**

**10**

**Το εμβαδό του κύκλου είναι : 314**

**20 / 177**

**Ωστόσο υπάρχουν μερικά λεπτά σημεία που αφορούν στο πέρασμα τιμών, τα οποία θα διευκρινιστούν στη συνέχεια.**

**10.5.1. Ορισμός και κλήση συναρτήσεων**

**Κάθε συνάρτηση έχει την ακόλουθη δομή.**

ΣΥΝΑΡΤΗΣΗ **όνομα(λίστα παραμέτρων) : τύπος συνάρτησης**

**Τμήμα δηλώσεων**

ΑΡΧΗ  
 . . .

όνομα🡨 **έκφραση**

. . .

ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

**Το όνομα της συνάρτησης είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ. Η λίστα παραμέτρων είναι μια λίστα μεταβλητών, των οποίων οι τιμές μεταβιβάζονται στη συνάρτηση κατά την κλήση ΠΡΑΓΜΑΤΙΚΗ, ΑΚΕΡΑΙΑ, ΧΑΡΑΚΤΗΡΑΣ, ΛΟΓΙΚΗ**

**Στις εντολές του σώματος της συνάρτησης πρέπει υποχρεωτικά να υπάρχει μία εντολή εκχώρησης τιμής στο όνομα της συνάρτησης, στο προηγούμενο παρά-δειγμα Εμβαδό\_κύκλου** 🡨 **Π \* R ^ 2.**

**Κάθε συνάρτηση εκτελείται, όπως ακριβώς εκτελούνται οι ενσωματωμένες συναρτήσεις της γλώσσας. Απλώς αναφέρεται το όνομά της σε μια έκφραση ή σε μία εντολή και επιστρέφεται η τιμή της. Στο παράδειγμα η συνάρτηση εκτελείται με την εντολή   
Εμ** 🡨 **Εμβαδό\_κύκλου (R).**

**Ο μηχανισμός που επιτυγχάνεται αυτό, είναι ο εξής:**

**Το κύριο πρόγραμμα πριν την κλήση της συνάρτησης γνωρίζει την τιμή της μεταβλητής R.**

**Κατά την κλήση μεταβιβάζεται αυτή η τιμή στην αντί-στοιχη μεταβλητή R της συνάρτησης. Η συνάρτηση υπολογίζει το εμβαδό του κύκλου και το αποτέλεσμα αυτό εκχωρείται στο όνομα της συνάρτησης. Με το τέλος της συνάρτησης γίνεται επιστροφή στο κύριο πρόγραμμα, όπου η τιμή του εμβαδού εκχωρείται στη μεταβλητή Εμ.**

**10.5.2. Ορισμός και κλήση διαδικασιών**

**Κάθε διαδικασία έχει την ακόλουθη δομή.**

ΔΙΑΔΙΚΑΣΙΑ  Όνομα **(λίστα παραμέτρων) Τμήμα δηλώσεων**

**21 / 177-178**

ΑΡΧΗ  
 **εντολές**

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

**Το όνομα της διαδικασίας είναι οποιοδήποτε έγκυρο όνομα της ΓΛΩΣΣΑΣ.Η λίστα παραμέτρων είναι μια λίστα μεταβλητών, των οποίων οι τιμές μεταβιβάζονται προς τη διαδικασία κατά την κλήση ή/και επιστρέ-φονται στο κύριο πρόγραμμα μετά το τέλος της διαδικασίας. Στο σώμα της διαδικασίας μπορούν να υπάρχουν οποιεσδήποτε εντολές της γλώσσας.**

**Κάθε διαδικασία εκτελείται όταν καλείται από το κύριο πρόγραμμα ή άλλη διαδικασία. Η κλήση σε διαδικασία πραγματοποιείται με την εντολή ΚΑΛΕΣΕ, που ακολου-θείται από το όνομα της διαδικασίας συνοδευόμενο μέσα σε παρενθέσεις με τη λίστα παραμέτρων. Η γενική μορφή της εντολής ΚΑΛΕΣΕ είναι:**

Σύνταξη

ΚΑΛΕΣΕ όνομα-διαδικασίας (λίστα-παραμέτρων)

Παράδειγμα

ΚΑΛΕΣΕ Πράξεις ( Α, Β, Διαφορά )

Λειτουργία

Η εκτέλεση του προγράμματος διακόπτεται και εκτε-λούνται οι εντολές της διαδικασίας που καλείται. Μετά το τέλος της διαδικασίας, η εκτέλεση του προγράμ-ματος συνεχίζεται από την εντολή που ακολουθεί. Η λίστα των παραμέτρων ορίζει τις τιμές που περνούν στη διαδικασία και τις τιμές που αυτή επιστρέφει. Η λίστα παραμέτρων δεν είναι υποχρεωτική.

**22 / 178**

**Στο προηγούμενο παράδειγμα η κλήση των δύο διαδι-κασιών έγινε με τις εντολές**

ΚΑΛΕΣΕ **Είσοδος\_δεδομένων (R)**

ΚΑΛΕΣΕ **Εκτύπωση (Εμ)**

**Σε κάθε περίπτωση κλήσης διαδικασίας μπορεί να γίνεται πέρασμα τιμών μέσω της λίστας παραμέτρων. Πιο συγκεκριμένα, στην περίπτωση της διαδικασίας Είσοδος\_δεδομένων γίνεται επιστροφή στο κύριο πρόγραμμα της τιμής της ακτίνας, ενώ στη διαδικασία Εκτύπωση γίνεται μεταβίβαση της τιμής του εμβαδού από το κύριο πρόγραμμα στη διαδικασία. Δηλαδή, η Είσοδος\_δεδομένων δέχεται μια τιμή από πληκτρο-λόγιο, την εκχωρεί στη μεταβλητή Αριθμός και κατά την επιστροφή (μετά το τέλος της διαδικασίας) γίνεται μεταβίβαση αυτής της τιμής στη μεταβλητή R του κύριου προγράμματος. Αντίθετα στη διαδικασία Εκτύπωση κατά την κλήση της μεταβιβάζεται η τιμή της μεταβλητής Ε του κύριου προγράμματος στη μεταβλητή Αποτέλεσμα της διαδικασίας.**

**Στο συγκεκριμένο παράδειγμα κάθε διαδικασία έχει από μία παράμετρο. Στη γενική περίπτωση μπορούν να υπάρχουν καμία, μία ή περισσότερες παράμετροι. Όταν υπάρχουν πολλές παράμετροι, τότε άλλες χρησιμο-ποιούνται για να μεταβιβάσουν τιμές στη διαδικασία και άλλες για να επιστρέψουν τιμές στο κύριο πρόγραμμα.**

**Κάθε διαδικασία ή συνάρτηση μπορεί να καλείται από το κύριο πρόγραμμα ή από άλλη διαδικασία ή συνάρ-τηση. Σε κάθε περίπτωση μετά το τέλος της εκτέλεσης της διαδικασίας ή της συνάρτησης γίνεται επιστροφή ακριβώς μετά το σημείο απ' όπου κλήθηκε.**

**23 / 178-179**

**Στη συνέχεια παρουσιάζεται το παράδειγμα 2 υλοποι-ημένο στις γλώσσες Pascal και Basic.**

|  |
| --- |
| Προγραμματιστικό περιβάλλον Pascal  PROGRAM example2;  VAR  r,e: REAL;  **FUNCTION area(r:REAL) :REAL;**  **BEGIN**  **area := pi \* sqr (r)**  **END;**  **PROCEDURE input (var x:REAL);**  **BEGIN**  **REPEAT**  **write** **(‘Δώσε την ακτίνα :’);**  **readln (x);**  **UNTIL x >0**  **END;** |
| **PROCEDURE output (result:REAL);**  **BEGIN**  **writeln** **(‘Το εμβαδό είναι :’); result:6:2)**  **END;** |
| **BEGIN**  **Input (r);**  **e:=area (r);**  **output (e);**  **END.** |

**24 / 179-180**

Προγραμματιστικό περιβάλλον Basic

‘ Παράδειγμα 3

DECLARE SUB Eisodos (nb!)

DECLARE SUB Ektypwsh (res!)

DECLARE SUB Emvado! (r!)

CLS

**CALL** Eisodos (r)

e = Emvado (r)

CALL Ektypwsh (e)

END

SUB Eisodos (nb)

DO

INPUT “Δώσε την ακτίνα : ”, nb

LOOP UNTIL nb >0

END SUB

**25 / 180**

SUB Ektypwsh (res)

PRINT “Το εμβαδό του κύκλου είναι: ”; res

END SUB

**FUNCTION Emvado (r)**

**pi =3.14**

**Emvado = pi \* r ^2**

**END FUNCTION**

**10.5.3. Πραγματικές και τυπικές παράμετροι**

**Η κατανόηση του τρόπου που γίνεται η ανταλλαγή των τιμών ανάμεσα στις παραμέτρους είναι ιδιαίτερα σημαντική και γι αυτό ας παρακολουθήσουμε το επόμενο παράδειγμα.**

**ΠΑΡΑΔΕΙΓΜΑ 3**

**Να γραφεί μια διαδικασία η οποία δέχεται στην είσοδο δυο τιμές και υπολογίζει και επιστρέφει το άθροισμα και τη διαφορά τους.**

****

**Οι μεταβλητές Α, Β, Διαφ1, Αθρ1, Διαφ2, Αθρ2 είναι μεταβλητές του προγράμματος Παράδειγμα\_3 και αποτελούν τις πραγματικές παραμέτρους, ενώ οι μεταβλητές Χ,Υ, Διαφορά, Άθροισμα είναι μεταβλητές της διαδικασίας Πράξεις, και ονομάζονται τυπικές παράμετροι.**

**26 / 180-181**

**Οι μεταβλητές Α, Β, Διαφ1 καθώς και όλες οι μεταβλητές του προγράμματος Παράδειγμα\_3 δεν είναι γνωστές στη διαδικασία Πράξεις και αντίστοιχα όλες οι μετα-βλητές της διαδικασίας Πράξεις είναι άγνωστες στο πρόγραμμα Παράδειγμα\_3. Τα ονόματα των τυπικών και των πραγματικών παραμέτρων μπορούν να είναι οποιαδήποτε. Αφού είναι ονόματα μεταβλητών σε δια-φορετικά τμήματα προγράμματος, είναι υποχρεωτικά διαφορετικές μεταβλητές, άσχετα αν έχουν το ίδιο όνομα.**

**Όλες οι μεταβλητές είναι γνωστές, έχουν ισχύ όπως λέγεται, μόνο για το τμήμα προγράμματος στο οποίο έχουν δηλωθεί, ισχύουν δηλαδή τοπικά για το συγκε-κριμένο υποπρόγραμμα ή κυρίως πρόγραμμα.**

**ΟΡΙΣΜΟΣ**

|  |
| --- |
| **Η λίστα των τυπικών παραμέτρων (formal parameter list) καθορίζει τις παραμέτρους στη δήλωση του υποπρογράμματος.**  **Η λίστα των πραγματικών παραμέτρων (actual parameter list) καθορίζει τις παραμέτρους στην κλήση του υποπρογράμματος.** |

|  |  |
| --- | --- |
| **τρομακτικο.jpg** | **Μερικές γλώσσες προγραμματισμού ονομάζουν ορίσματα τις τυπικές παραμέτρους και απλά παραμέτρους τις πραγματικές παραμέτρους** |

**27 / 181**

**Ας παρακολουθήσουμε πώς γίνεται η επικοινωνία ανάμεσα στο πρόγραμμα Παράδειγμα\_3 και τη διαδικασία Πράξεις.**

**Οι τιμές που υπάρχουν στις μεταβλητές του προ-γράμματος Α,Β, Διαφ1 και Αθρ1 δίνονται κατά την κλήση στις μεταβλητές της διαδικασίας Χ, Υ, Διαφορά, Άθροισμα.**

**Έτσι η μεταβλητή Χ παίρνει την τιμή 5 κι η Υ την τιμή 7. Οι μεταβλητές Διαφορά και Άθροισμα δεν παίρνουν καμία τιμή, αφού οι αντίστοιχες μεταβλητές Διαφ1 και Αθρ1 δεν έχουν συγκεκριμένη τιμή.**

**Μετά την εκτέλεση των εντολών της διαδικασίας, όταν εκτελεστεί η εντολή ΤΕΛΟΣ\_ΔIΑΔIΚΑΣIΑΣ, οι μεταβλη-τές της διαδικασίας που αναφέρονται στη δήλωση της διαδικασίας δίνουν τις τιμές που περιέχουν στις αντίστοιχες μεταβλητές που περιλαμβάνονται στην κλήση της διαδικασίας Πράξεις. Έτσι η Α παίρνει την τιμή της Χ ( = 5), η Β την τιμή της Υ (= 7), η Διαφ1 της Διαφορά (=-2) και η μεταβλητή Άθρ1 της Άθροισμα ( = 12). Με την επιστροφή στο κύριο πρόγραμμα όλες οι θέσεις μνήμης που είχαν δοθεί στη διαδικασία απελευθερώνονται.**

**Οι λίστες των παραμέτρων πρέπει να ακολουθούν τους εξής κανόνες:**

**28 / 181**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Α** | | **5** |  | | **Α** | | **5** | |  | | **5** | **Χ** | | |
| **Β** | | **7** |  | | **Β** | | **7** | |  | | **7** | **Υ** | | |
| **Διαφ1** | |  |  | | **Διαφ1** | |  | |  | |  | **Διαφορά** | | |
| **Άθρ1** | |  |  | | **Άθρ1** | |  | |  | |  | **Άθροισμα** | | |
|  | | **(α)** |  | |  | |  | | **(β)** | |  |  | | |  |
|  | |  |  | |  | |  | |  | |  |  | | |  |
| **5** | **Χ** | | | | | **Α** | **5** |  | | **5** | | | **Χ** | |
| **7** | **Υ** | | | | | **Β** | **7** |  | | **7** | | | **Υ** | |
| **-2** | **Διαφορά** | | | | | **Διαφ1** | **-2** |  | | **-2** | | | **Διαφορά** | |
| **12** | **Άθροισμα** | | | | | **Άθρ1** | **12** |  | | **12** | | | **Άθροισμα** | |
|  |  | | |  | |  |  |  | |  | | |  |  |
| **(γ)** |  | | |  | |  |  | **(δ)** | |  | | |  |  |

**Σχ. 10.2. Πέρασμα παραμέτρων κατά την κλήση διαδικασιών (α) Κατάσταση πριν την κλήση (β) Μεταβίβαση τιμών των μεταβλητών Α και Β στις Χ και Υ αντίστοιχα (γ) Στη διαδικασία εκχωρούνται τιμές στις μεταβλητές Διαφορά και Άθροισμα (δ) Οι τιμές των τελευταίων επιστρέφονται στις Διαφ1 και Αθρ1 μετά το τέλος της διαδικασίας** χρήση στ

**130 / 218**

**29 / 181**

**Η χρήση στοίβας στην κλήση διαδικασιών**

**Η έννοια της στοίβας είναι πολύ χρήσιμη στο ίδιο το λογισμικό των γλωσσών προγραμματισμού. Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται διεύθυνση επιστροφής (return address), αποθηκεύεται από το μεταφραστή σε μία στοίβα που ονομάζεται στοίβα χρόνου εκτέλεσης (execution time stack).Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης, η διεύθυνση επιστροφής απωθείται από τη στοίβα και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα. Η τεχνική αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μία διαδικασία ή συνάρτηση καλεί μία διαδικασία ή συνάρτηση. Για παράδειγμα, έστω ότι μία διαδικασία a καλεί τη διαδικασία b που με τη σειρά της καλεί τη διαδικασία c κοκ. Στην περίπτωση αυτή ο διευθύνσεις επιστροφής εμφανίζονται στη στοίβα με σειρά c, b, a. Μετά την εκτέλεση κάθε διαδικασίας, η διεύθυνση επιστροφής απωθείται από τη στοίβα και ο έλεγχος μεταβιβάζεται στη διεύθυνση αυτή. Το παράδειγμα αυτό δείχνει μία από τις πολλές χρησιμότητες της LIFO ιδιότητας της στοίβας.**

**30 / 182**



**Σχ. 10.3. Χρήση στοίβας από το μεταφραστή για το χειρισμό κλήσεων διαδικασιών και επιστροφών από αυτές.**

**31 / 182**

**Οι λίστες των παραμέτρων πρέπει να ακολουθούν τους εξής κανόνες:**

* **Ο αριθμός των πραγματικών και των τυπικών παραμέτρων πρέπει να είναι ίδιος.**
* **Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση. Για παράδειγμα η πρώτη της λίστας των τυπικών παραμέτρων στην πρώτη της λίστας των πραγματικών παραμέτρων κοκ.**
* **Η τυπική παράμετρος και η αντίστοιχή της πραγματική πρέπει να είναι του ιδίου τύπου.**

|  |  |
| --- | --- |
| **10.6** | **Εμβέλεια μεταβλητών-σταθερών** |

**Κάθε κύριο πρόγραμμα όπως και κάθε υποπρόγραμμα περιλαμβάνει τις δικές του μεταβλητές και σταθερές.**

**Οι μεταβλητές αυτές στη ΓΛΩΣΣΑ είναι γνωστές στο αντίστοιχο υποπρόγραμμα που δηλώνονται και μόνο σε αυτό. Όλες οι μεταβλητές (και οι σταθερές) είναι τοπικές στο συγκεκριμένο τμήμα προγράμματος.**

**Ο μόνος τρόπος για να περάσει μία τιμή από ένα υπο-πρόγραμμα σε ένα άλλο ή από το κυρίως πρόγραμμα σε ένα υποπρόγραμμα είναι διαμέσου των παραμέτρων κατά το στάδιο της κλήσης του υποπρογράμματος και μετά το τέλος της εκτέλεσης του υποπρογράμματος.**

**32 / 182-183**

**Ας δούμε τον παρακάτω σκελετό προγράμματος**

ΠΡΟΓΡΑΜΜΑ **Αρχικό**  
ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ: **Α, Β, Γ**   
ΑΡΧΗ  
. . .

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ **Πρώτη**

ΜΕΤΑΒΛΗΤΕΣ  
 ΠΡΑΓΜΑΤΙΚΕΣ: **Δ, Ε, Ζ, Η**

ΑΡΧΗ  
. . .

ΤΕΛΟΣ\_ ΔΙΑΔΙΚΑΣΙΑΣ   
ΔΙΑΔΙΚΑΣΙΑ **Δεύτερη**

ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ: **Γ, Θ, Ι**

ΑΡΧΗ  
. . .

ΤΕΛΟΣ\_ ΔΙΑΔΙΚΑΣΙΑΣ

**Οι μεταβλητές του προγράμματος Αρχικό με ονόματα Α, Β, Γ είναι γνωστές, ισχύουν μόνο για το πρόγραμμα. Έξω από το πρόγραμμα σε όλα τα υποπρογράμματα οι μεταβλητές αυτές δεν ισχύουν. Επίσης η διαδικασία Πρώτη έχει τις πραγματικές μεταβλητές Δ, Ε, Ζ, Η, οι οποίες ισχύουν μόνο για τη συγκεκριμένη διαδικασία και όχι για τα υπόλοιπα υποπρογράμματα ή το κύριο πρόγραμμα.**

**Η διαδικασία Δεύτερη έχει και αυτή τις δικές της μετα-βλητές Γ,Θ,Ι. Οι μεταβλητές αυτές ισχύουν μόνο για τη διαδικασία Δεύτερη. Η μεταβλητή με το όνομα Γ δεν έχει καμία σχέση με τη μεταβλητή Γ του κύριου προγράμματος, η μία είναι τύπου Ακεραίου και η άλλη τύπου Πραγματικού. Αφού όλες οι μεταβλητές είναι τοπικές, το ίδιο όνομα μεταβλητής μπορεί να εμφα-νίζεται σε διαφορετικά τμήματα προγράμματος, χωρίς να αντιστοιχεί στην ίδια μεταβλητή. Το ίδιο έγινε και με τη μεταβλητή R της ακτίνας του κύκλου στο παράδειγμα 2.**

**33 / 183**

**Ότι ισχύει για τις μεταβλητές ισχύει και για τις σταθερές.**

**Πολλές γλώσσες προγραμματισμού επιτρέπουν τη χρήση των μεταβλητών και των σταθερών, όχι μόνο στο τμήμα προγράμματος που δηλώνονται, αλλά και σε άλλα ή ακόμη και σε όλα τα υπόλοιπα υποπρογράμ-ματα. Αυτό που καθορίζει την περιοχή που ισχύουν οι μεταβλητές και οι σταθερές είναι η εμβέλεια των μεταβλητών της γλώσσας.**

**ΟΡΙΣΜΟΣ**

**Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται εμβέλεια (scope) μεταβλητών.**

**Απεριόριστη εμβέλεια.**

**Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμο-ποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν. Όλες οι μεταβλητές είναι καθολικές.**

**Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.**

**34 / 183-184**

**Περιορισμένη εμβέλεια**

**Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μετα-βλητές που χρησιμοποιούνται σε ένα τμήμα προ-γράμματος, να δηλώνονται σε αυτό το τμήμα. Όλες οι μεταβλητές είναι τοπικές, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν. Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια.**

**35 / 184**

**Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.**

**Μερικώς περιορισμένη εμβέλεια**

**Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι τοπικές και άλλες καθολικές. Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται οι μεταβλητές ως τοπικές ή καθολικές.**

**Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.**

|  |  |
| --- | --- |
| **10.7** | **Αναδρομή** |

**Ένα υποπρόγραμμα καλείται από το κύριο πρόγραμμα ή άλλο υποπρόγραμμα. Υπάρχει όμως και η δυνα-τότητα ένα υποπρόγραμμα να καλεί τον εαυτό του. Η δυνατότητα αυτή αποκαλείται αναδρομή.**

**Η έννοια της αναδρομής παρουσιάστηκε στο κεφάλαιο 3, όπου δόθηκαν παραδείγματα και έγινε μία πρώτη αναφορά στα πλεονεκτήματα, αλλά και τα προβλήματα που παρουσιάζει η χρήση αναδρομικών αλγορίθμων.**

**Η αναδρομή υλοποιείται στον προγραμματισμό με χρήση αναδρομικών υποπρογραμμάτων και επειδή τόσο η αναδρομή ως έννοια όσο και η λειτουργία της είναι αρκετά πολύπλοκες, θα παρουσιαστούν ανα-λυτικά στη συνέχεια.**

**Ο ορισμός κάθε αναδρομικού υποπρογράμματος έχει δύο τμήματα, την αναδρομική σχέση και την τιμή βάσης ή συνθήκη τερματισμού.**

**Ας εξετάσουμε το παράδειγμα υπολογισμού του παραγοντικού:**

**Το παραγοντικό ορίζεται αναδρομικά ως εξής:**

****

**Τιμή βάσης ή συνθήκη τερματισμού είναι η τιμή που ορίζεται για μία συγκεκριμένη τιμή της παραμέτρου της συνάρτησης, στο παράδειγμα η τιμή είναι 1 για n=0.**

**Αναδρομική σχέση είναι η n\*(n-1)!, όπου η τιμή της συνάρτησης υπολογίζεται με βάση τις προηγούμενες τιμές της συνάρτησης, οι οποίες πρέπει να υπολο-γιστούν.**

**36 / 184-185**

**Οι συναρτήσεις στη γλώσσα που υλοποιούν τον υπολογισμό του παραγοντικού τόσο αναδρομικά όσο και επαναληπτικά είναι οι εξής.**

ΣΥΝΑΡΤΗΣΗ  **Παραγοντικό (N):** ΑΚΕΡΑΙΑ  
**! Υπολογισμός του παραγοντικού με αναδρομική διαδικασία**

**37 / 185**

ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ:  **N**

ΑΡΧΗ

ΑΝ **N = 0** ΤΟΤΕ  
 **Παραγοντικό** 🡨 **1**

ΑΛΛΙΩΣ  
 **Παραγοντικό** 🡨 **N \* Παραγοντικό(N-1)**

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

ΣΥΝΑΡΤΗΣΗ  **Παραγοντικό (N):** ΑΚΕΡΑΙΑ  
**! Υπολογισμός του παραγοντικού με επαναληπτική διαδικασία**

ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ:  **i, N**

ΑΡΧΗ

Fact 🡨 **1**

ΓΙΑ **i** ΑΠΟ 2 ΜΕΧΡΙ **N**  
 Fact 🡨 Fact **\* i**

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ

**Και οι δύο αυτές συναρτήσεις, η πρώτη αναδρομική και η δεύτερη επαναληπτική έχουν το ίδιο αποτέλεσμα. Το αναδρομικό υποπρόγραμμα επειδή εκφράζει ακριβώς την αλγεβρική συνάρτηση υπολογισμού του παρα-γοντικού είναι πιο απλό στη διατύπωση, αλλά είναι πιο δύσκολο στην κατανόηση του τρόπου που λειτουργεί.**

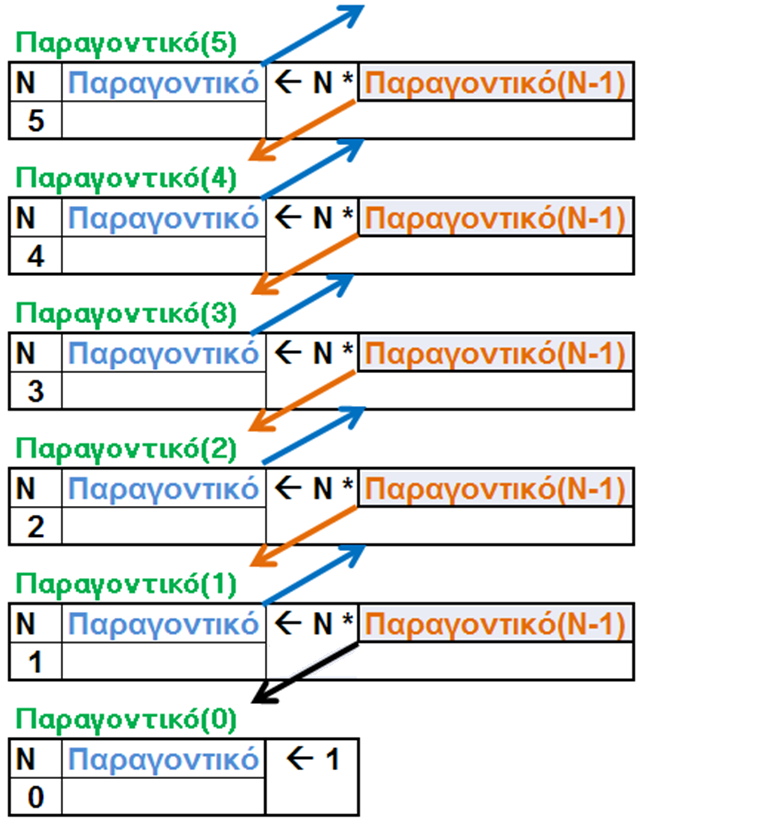
**Γενικά η έννοια της αναδρομής καθώς και ο τρόπος με τον οποίο εκτελείται μία αναδρομική διαδικασία από τον υπολογιστή παρουσιάζει στην αρχή δυσκολία στην κατανόηση της.**

**Ας παρακολουθήσουμε τον τρόπο με τον οποίο υπολο-γίζεται το 5! με τη χρήση της αναδρομικής συνάρτησης Παραγοντικό().**

**Εφόσον η παράμετρος της συνάρτησης είναι αρχικά 5, διάφορη δηλαδή του 0, καλείται η συνάρτηση Παραγο-ντικό(4).**

**Στην συνέχεια επειδή η παράμετρος είναι διάφορη του μηδενός, καλείται το Παραγοντικό(3) και η διαδικασία συνεχίζεται όπως δείχνει το σχήμα μέχρι την κλήση του Παραγοντικό(0).**

**38 / 185-186**

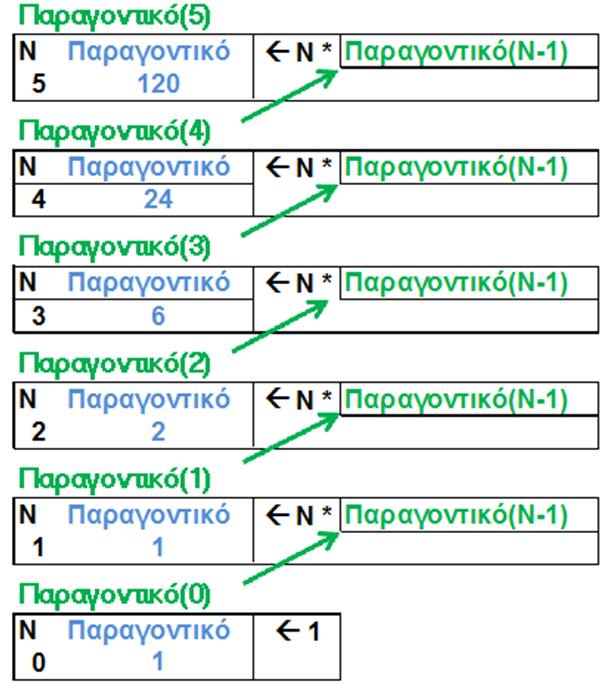
****

**39 / 186**

**Για τη τιμή της παραμέτρου Ν=0 το Παραγοντικό(0) έχει τιμή 1, την τιμή βάσης. Έτσι αρχίζει η αντίστροφη διαδικασία υπολογισμού διαδοχικά του υπολογισμού των συναρτήσεων Παραγοντικό(2)=1 \*2, Παραγοντικό(3) = 2\*3, Παραγοντικό(4) = 6\*4 όπως δείχνει το παρακάτω σχήμα.**

**Το τελευταίο βήμα είναι ο υπολογισμός του Παραγοντικό(5)=24\*5. Η τιμή αυτή δηλαδή 120 είναι η τιμή που η συνάρτηση τελικά επιστρέφει.**

**40 / 186**

****

**41 / 187**

**Πολύ συχνά προβλήματα τα οποία μπορούν να λυθούν με χρήση αναδρομής λύνονται και με απλούς επαναλη-πτικούς αλγόριθμους. Παραδείγματα παρουσιάστηκαν στο κεφάλαιο 3, όπου έγινε και σύγκριση μεταξύ των δύο μεθόδων. Εδώ επαναλαμβάνουμε ότι αν και η αναδρομή φαίνεται ως πιο φυσικός τρόπος προγραμ-ματισμού, πρέπει να χρησιμοποιείται με μέτρο, γιατί το κέρδος σε χρόνο προγραμματισμού δημιουργεί συχνά απώλεια σε χρόνο εκτέλεσης.**

|  |  |
| --- | --- |
| **πιν.jpg** | **Ο μηχανισμός διαχείρισης των αναδρο-μικών διαδικασιών από το μεταφραστή κάνει χρήση μιας στοίβας. Κατά την κλήση μιας αναδρομικής διαδικασίας από την ίδια, φυλάσσονται σε μια στοίβα όλες οι τιμές των μεταβλητών, που έχουν πριν την κλήση (λειτουργία ώθηση).Κατά την επιστροφή, ανασύρονται οι τιμές αυτές από τη στοίβα (λειτουργία απώθηση) και έτσι η διαδικασία μπορεί να συνεχίσει τη λειτουργία της με τις προηγούμενες τιμές. Προφανώς, αν υπάρχουν διαδοχι-κές κλήσεις της διαδικασίας, λόγω της ιδιότητας LIFO (Last In – First Out) με την οποία λειτουργεί η στοίβα, εξασφα-λίζεται ότι εξάγονται από αυτή, οι πλέον πρόσφατες τιμές των μεταβλητών, δηλαδή αυτές που αποθηκεύτηκαν κατά την τελευταία κλήση.** |

**42 / 187**

Προγραμματιστικό περιβάλλον Pascal

FUNCTION factorial (n:INTEGER) : INTEGER

**BEGIN**

IF n=0 **THEN**

factorial := 1

ELSE

factorial := n\* factorial (n-1);

**END;**

FUNCTION factorial (n:INTEGER) : INTEGER

**VAR**

**i, fact : integer;**

**BEGIN**

**fact:=1;**

**FOR i:= 2 TO n DO**

**fact:= i\* fact;**

**factorial:= fact;**

**END;**

Προγραμματιστικό περιβάλλον BASIC

FUNCTION Factorial (n)

IF n=0 **THEN**

Factorial = 1

ELSE

Factorial = n\* Factorial (n-1)

**END IF**

**END** FUNCTION

FUNCTION Factorial (n)

fact = 1

**FOR i = 2 TO n** fact = fact \* i

NEXT i

Factorial = fact

**END** FUNCTION

**43 / 188**

**• Ανακεφαλαίωση**

**Στο κεφάλαιο αυτό παρουσιάστηκαν οι αρχές και τα χαρακτηριστικά του τμηματικού προγραμματισμού και ο τρόπος που χειρίζεται τα υποπρογράμματα η ΓΛΩΣΣΑ.**

**Η χρήση υποπρογραμμάτων σε ένα πρόγραμμα παρουσιάζει πολλά πλεονεκτήματα, αφού διευκολύνει την ανάπτυξη των αλγόριθμων και την υλοποίηση του προγράμματος σε λιγότερο μάλιστα χρόνο και διευκο-λύνει την κατανόηση και τη διόρθωσή του. Υπάρχουν δύο ειδών υποπρογράμματα, οι συναρτήσεις και οι διαδικασίες. Η συνάρτηση παράγει ένα αποτέλεσμα και επιστρέφει μόνο μία τιμή, ενώ οι διαδικασίες μπορούν να εκτελούν όλες τις ενέργειες ενός προγράμματος. Τα υποπρογράμματα επικοινωνούν μεταξύ τους καθώς και με το κύριο πρόγραμμα με τη βοήθεια των παραμέ-τρων. Στη δήλωση του υποπρογράμματος βρίσκονται οι τυπικές παράμετροι, ενώ οι πραγματικές παράμετροι βρίσκονται στην εντολή κλήσης του υποπρογράμ-ματος. Οι διαδικασίες ενεργοποιούνται στη ΓΛΩΣΣΑ με την εντολή ΚΑΛΕΣΕ, ενώ οι συναρτήσεις μόνο με την αναγραφή του ονόματος τους μέσα σε μία έκφραση.**

**Η δυνατότητα ενός υποπρογράμματος να καλεί τον εαυτό του, ονομάζεται αναδρομή. Πολύ συχνά τα προ-βλήματα που λύνονται με αναδρομή μπορούν να λυ-θούν με χρήση απλής επανάληψης.**

**44 / 189**

**Λέξεις κλειδιά**

**Τμήμα προγράμματος, διαδικασία, συνάρτηση, παράμετρος, εμβέλεια παραμέτρων, αναδρομή.**

**• Ερωτήσεις Θέματα για συζήτηση**

1. **Πώς ορίζεται ο τμηματικός προγραμματισμός;**

**2. Ποια τα βασικά χαρακτηριστικά των υποπρογραμ-μάτων;**

**3. Ποια η διαφορά παραμέτρων και απλών μεταβλητών;**

**4. Ποια η βασική διαφορά διαδικασιών και συναρτή-σεων;**

**5. Πώς εκτελείται μία συνάρτηση;**

**6. Πώς καλείται μία διαδικασία;**

**7. Ποια η διαφορά τυπικών και πραγματικών παραμέ-τρων**;

**8. Πώς γίνεται η ανταλλαγή των τιμών ανάμεσα στις τυπικές και πραγματικές παραμέτρους; Δώσε ένα παράδειγμα ανταλλαγής παραμέτρων.**

**9. Τι ονομάζεται εμβέλεια μεταβλητών;**

**10. Τι είναι η αναδρομή;**

**45 / 189**

**11. Ποια είναι τα τμήματα που περιλαμβάνει κάθε αναδρομικός ορισμός; Δώσε ένα παράδειγμα.**

**• Βιβλιογραφία**

1. **Θ.Αλεβίζος, Α.Καμπουρέλης, Εισαγωγή με τη γλώσσα Pascal, Αθήνα, 1984.**
2. **Γ.Βουτυράς,Basic: Αλγόριθμοι και εφαρμογές, Κλεψύδρα, Αθήνα, 1991.**
3. **Χρ.Κοίλιας, Η QuickBasic και οι εφαρμογές της, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.**
4. **R.Shackelford, Introduction to Computing and Algorithms, Addison-Wesley, USA, 1998.**
5. **S.Leestma-L.Nyhoff, Turbo Pascal, Programming and Solving, McMillan, New York,1990.**

**46 / 190**

1. **N.Wirth, Systematic Programming: An introduction, Prentice Hall, 1973.**

**47 / 265**

* **ΠΑΡΑΡΤΗΜΑ**

**Πίνακας ASCII**

|  |
| --- |
| **48 / 266** |
|  |

**49 / 267**

|  |  |  |  |
| --- | --- | --- | --- |
| **DEC.** | **HEX.** | **CHAR.** |  |
| **0** | **00** |  | **NUL** |
| **1** | **01** | **☺** | **SOH** |
| **2** | **02** | **☻** | **STX** |
| **3** | **03** | **♥** | **ETX** |
| **4** | **04** | **♦** | **EOT** |
| **5** | **05** | **♣** | **ENQ** |
| **6** | **06** | **♠** | **ACK** |
| **7** | **07** | **•** | **BEL** |
| **8** | **08** | **◘** | **BS** |
| **9** | **09** | **○** | **HT** |
| **10** | **0A** | **◙** | **LF** |
| **11** | **0B** | **♂** | **VT** |
| **12** | **0C** | **♀** | **FF** |
| **13** | **0D** | **♪** | **CR** |
| **14** | **0E** | **♫** | **SO** |
| **15** | **0F** | **☼** | **SI** |
| **16** | **10** | **►** | **DLE** |
| **17** | **11** | **◄** | **DC1** |
| **18** | **12** | **↕** | **DC2** |
| **19** | **13** | **‼** | **DC3** |
| **20** | **14** | **¶** | **DC4** |
| **21** | **15** | **§** | **NAK** |
| **22** | **16** | **▬** | **SYN** |
| **23** | **17** | **↨** | **ETB** |
| **24** | **18** | **↑** | **CAN** |
| **25** | **19** | **↓** | **EM** |
| **26** | **1A** | **→** | **SUB** |
| **27** | **1B** | **←** | **ESC** |
| **28** | **1C** | **∟** | **FS** |
| **29** | **1D** | **↔** | **GS** |
| **30** | **1E** | **▲** | **RS** |
| **31** | **1F** | **▼** | **US** |



**50 / 267**

**51 / 267**

|  |  |
| --- | --- |
| **Ευρετήριο αλγορίθμων** | |
| **Πολλαπλασιασμός αλά ρωσικά** | **48** |
| **Σειριακή αναζήτηση** | **58** |
| **Μικρότερο στοιχείο πίνακα** | **59** |
| **Άθροισμα κατά γραμμή και στήλη δισδιάστατου πίνακα** | **63** |
| **Ταξινόμηση φυσαλίδας** | **66** |
| **Μέγιστος Κοινός Διαιρέτης** | **68,185** |
| **Παραγοντικό (επαναληπτικός)** | **68,185** |
| **Παραγοντικό (αναδρομικός)** | **68** |
| **Αλγόριθμος Ευκλείδη (επαναληπτικός)** | **69** |
| **Αλγόριθμος Ευκλείδη (αναδρομικός)** | **70** |
| **Αριθμοί Fibonacci** | **70** |
| **Δυαδική αναζήτηση** | **80** |
| **Δύναμη αριθμού** | **83** |
| **Υπολογισμός νομισμάτων** | **84** |
| **Μέσος όρος αριθμών** | **146,148** |
| **Προπαίδεια** | **151** |
| **Μέση τιμή, τυπική απόκλιση, διάμεσος** | **157** |
| **Συχνότητα εμφάνισης** | **244** |
| **Ταξινόμηση τριών αριθμών** | **247** |
| **Έλεγχος ημερομηνίας** | **251** |

**52 / 268**

|  |
| --- |
| **ΓΛΩΣΣΑΡΙΟ** |
|  |
| **αλγοριθμικές γλώσσες**  **algorithmic** **languagesΚατηγορία γλωσσών προγραμματισμού κατάλληλων για την περιγραφή προβλημάτων που μπορούν να λυθούν αλγοριθμικά.** |
| **αλγόριθμος** **algorithm** **Πεπερασμένο σύνολο σαφώς καθορισμένων κανόνων που βοηθούν στην επίλυση ενός προβλήματος μέσω ενός πεπερασμένου αριθμού βημάτων.** |
| **αλφαριθμητικό** **alphanumeric**  **Σύνολο χαρακτήρων που μπορεί να εμπεριέχει γράμματα, ψηφία και ειδικά σύμβολα όπως π.χ. σημεία στίξης.** |
| **αναδρομή** **recursion**  **Κατάσταση κατά την οποία μια διαδικασία ή συνάρτηση καλεί τον εαυτό της.** |
| **αναδρομική συνάρτηση** **recursive function**  **Μια συνάρτηση της οποίας οι τιμές είναι φυσικοί αριθμοί που παράγονται από φυσικούς αριθμούς με τύπους αντικατάστασης, στους οποίους η ίδια η συνάρτηση είναι ένας τελεστέος.** |
| **ανάλυση** **analysis**  **Η μεθοδική μελέτη ενός προβλήματος και η διαδικασία της διάσπασής του σε μικρότερες μονάδες για περαιτέρω έρευνα σε λεπτομέρεια.**  **53 / 269** |
| **αντικείμενο** **object***.*  **Στον αντικειμενοστραφή προγραμματισμό μια νοητική αφαίρεση που αποτελείται από δεδομένα και σχετικές λειτουργίες με αυτά.** |
| **αντικείμενο πρόγραμμα** **object** **program**  **Πρόγραμμα υπολογιστή σε μια τελική γλώσσα που έχει μεταφραστεί από πηγαίο πρόγραμμα.  αντικειμενοστραφής προγραμματισμός object** **oriented** **language**  **Μέθοδος για δόμηση προγράμματος σε ιεραρχικά οργανωμένες τάξεις, που περιγράφουν τα δεδομένα και τις λειτουργίες αντικειμένων που μπορούν να αλληλεπιδρούν με άλλα αντικείμενα.** |
|  |
| **αφαίρεση δεδομένων data** **abstraction  Η αντιμετώπιση των δεδομένων με βάση τις αφηρημένες ιδιότητές τους και όχι σε σύνδεση με κάποιο φορέα τους.** |
| **βρόχος loop  Σύνολο εντολών που μπορεί να εκτελεστεί επανειλημμένα, όσο ισχύει μια ορισμένη συνθήκη.** |
| **γεγονός event  Ενέργεια του χρήστη μιας εφαρμογής που αναγκάζει την εφαρμογή ή το λειτουργικό σύστημα να ανταποκριθεί. Τα γεγονότα συχνά σχετίζονται με κινήσεις του ποντικιού ή πληκτρολογήσεις.**  **54 / 269** |
| **γλώσσα μηχανής machine language  Γλώσσα χαμηλού επιπέδου που οι εντολές της αποτελούνται μόνο από τα δυαδικά ψηφία.** |
| **γλώσσα προγραμματισμού programming language  Τεχνητή γλώσσα σχεδιασμένη για να δημιουργεί ή να εκφράζει προγράμματα.** |
| **γράφος graph**  **Ένα σύνολο σημείων, κορυφών ή τόξων και ένα σύνολο ακμών, τόξων ή γραμμών που ενώνουν μερικά ή όλα από τα σημεία.** |
| **γράφω write  Κάνω μια μόνιμη ή παροδική καταχώριση δεδομένων σε μια μνήμη ή σε ένα μέσο αποθήκευσης δεδομένων.** |
| **δεδομένα data  Παράσταση γεγονότων, εννοιών ή εντολών σε τυποποιημένη μορφή που είναι κατάλληλη για επικοινωνία, ερμηνεία ή επεξεργασία από άνθρωπο ή αυτόματα μέσα.** |
| **δείκτης index  Στον προγραμματισμό, ένας ακέραιος που αναγνω-ρίζει τη θέση ενός στοιχείου δεδομένων σε μια ακο-λουθία στοιχείων δεδομένων.**  **55 / 269** |
| **δείκτης pointer  Στοιχείο δεδομένου που περιέχει τη διεύθυνση ενός άλλου στοιχείου δεδομένου.** |
| **δένδρο tree  Δομή δεδομένων που αποτελείται από κόμβους οι οποίο συνδέονται με ακμές. Σε κάθε κόμβο καταλήγει μια μόνο ακμή, αλλά μπορούν να ξεκινούν καμία, μία ή περισσότερες. Σε ένα μόνο κόμβο που αποκαλείται ρίζα, δεν καταλήγει καμία ακμή.  διαβάζω read  Παίρνω δεδομένα από μια μνήμη, ένα μέσο αποθή-κευσης ή από άλλη πηγή.** |
| **διαδικασιακή γλώσσα procedural language  Γλώσσα προσανατολισμένη στο πρόβλημα που διευκολύνει την έκφραση μιας διαδικασίας με τη μορφή ρητά εκφρασμένου αλγόριθμου.** |
| **διαλογικός interactive  Τρόπος λειτουργίας ενός προγράμματος κατά τον οποίο ένας χρήστης εισάγει δεδομένα ή εντολές ως απόκριση σε αντίστοιχες αιτήσεις (προτροπές) από το σύστημα.** |
| **διεπαφή interface  Κοινό σύνορο μεταξύ δύο λειτουργικών μονάδων, που ορίζεται από λειτουργικά χαρακτηριστικά, κοινά φυσικά χαρακτηριστικά διασύνδεσης, χαρακτηριστι-κά σημάτων και άλλα κατάλληλα χαρακτηριστικά.** |
| **διερμηνευτής interpreter Πρόγραμμα που μεταφράζει και εκτελεί κάθε εντολή μια γλώσσας προγραμματισμού υψηλού επιπέδου πριν τη μετάφραση και εκτέλεση της επόμενης.**  **56 / 269-270** |
| **δοκιμή και εκσφαλμάτωση προγράμματος  program testing and debugging  Διαδικασίες εντοπισμού και διόρθωσης σφαλμάτων.** |
| **δοκιμή προγράμματος program testing  Ένα βήμα της ανάπτυξης προγράμματος όπου ένα πλήρες πρόγραμμα δοκιμάζεται για σφάλματα. Μια δοκιμή προγράμματος εμπεριέχει την εκσφαλμά-τωση του προγράμματος.** |
| **δομημένος προγραμματισμός structure programming  Μέθοδος για την κατασκευή προγραμμάτων που χρησιμοποιεί μόνο ιεραρχικά εμπερικλειόμενα κατασκευάσματα, καθένα από τα οποία έχει ένα απλό σημείο εισόδου και ένα εξόδου. Τρεις τύποι ελέγχου χρησιμοποιούνται στο δομημένο προ-γραμματισμό: ακολουθιακός, υπό συνθήκη και επαναληπτικός.** |
| **εκσφαλμάτωση debugging**  **Έλεγχος της λογικής ενός προγράμματος για τον εντοπισμό και απομάκρυνση σφαλμάτων.**  **57 / 270** |
| **εκσφαλματωτής debugger**  **Βοηθητικό πρόγραμμα, που συνήθως παρέχεται με τις σύγχρονες γλώσσες προγραμματισμού, το οποίο καθιστά αποδοτική την έρευνα για τον εντοπισμό σφαλμάτων σε ένα πρόγραμμα.** |
| **εκχώρηση assignement**  **Μηχανισμός τιμοδότησης μιας μεταβλητής.** |
| **έλεγχος λογισμικού software testing  Φάση της ανάπτυξης λογισμικού κατά την οποία το προϊόν ελέγχεται ως προς την αναμενόμενη λειτουρ-γία του.** |
| **εντολή instruction  Σε μια γλώσσα προγραμματισμού μια έκφραση που έχει νόημα και η οποία καθορίζει μια πράξη και προσδιορίζει τους τελεστέους της, αν υπάρχουν.** |
| **επαλήθευση προγράμματος program verification  Διαδικασία που αποδεικνύει ότι ένα πρόγραμμα συμπεριφέρεται σύμφωνα με τις προδιαγραφές του.** |
| **επανάληψη iteration  Η διαδικασία επαναληπτικής εκτέλεσης ενός συνόλου εντολών μέχρι την ικανοποίηση κάποιας συνθήκης.**  **58 / 270** |
| **επεξεργασία δεδομένων data processing  Η συστηματική εκτέλεση πράξεων σε δεδομένα. Παραδείγματα: χειρισμός, συγχώνευση, ταξινόμηση, μεταγλώττιση κ.α.** |
| **εργαλεία λογισμικού software tools  Ειδικά προγράμματα που διευκολύνουν την συγκεκριμένες εργασίες της ανάπτυξης λογισμικού.** |
| **εφαρμογή application  Προγράμματα που γράφονται για την κάλυψη μια συγκεκριμένης ανάγκης.** |
| **κύκλος ζωής λογισμικού software life cycle  Ο κύκλος σχεδίασης, ανάπτυξης, εγκατάστασης και συντήρησης λογισμικού.** |
| **κώδικας code  Ένα ή περισσότερα προγράμματα ή τμήμα προγράμματος** |
| **λίστα (συνεζευγμένη) linked list  Μια λίστα στην οποία η μετάβαση από έναν κόμβο στον άλλο γίνεται με τη χρήση ενός δείκτη (pointer) που αποτελεί μέρος του κόμβου.**  **59 / 270** |
| **λογισμικό software  Πνευματική δημιουργία που περιλαμβάνει τα προγράμματα, τις διαδικασίες, τους κανόνες και οποιαδήποτε σχετική τεκμηρίωση που αναφέρεται στη λειτουργία ενός συστήματος επεξεργασίας δεδομένων.** |
| **μεταβλητή variable  Ένα όνομα που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου, του οποίου η τιμή μπορεί ν' αλλάζει κατά τη διάρκεια λειτουργίας του προγράμ-ματος.** |
| **μεταβλητή ελέγχου control variable  Μεταβλητή της οποίας η τιμή ελέγχει τον αριθμό εκτελέσεων ενός βρόχου.** |
| **μεταγλωττιστής compiler**  **Πρόγραμμα υπολογιστή που χρησιμοποιείται για τη μετάφραση σε γλώσσα χαμηλού επιπέδου ενός προ-γράμματος εκφρασμένου σε γλώσσα προσανατο-λισμένη στο πρόβλημα.** |
| **οδηγούμενο από γεγονότα event driven  Η ιδιότητα ενός λειτουργικού συστήματος ή περβάλ-λοντος κατά την οποία όταν συμβεί ένα γεγονός εκτελείται κατάλληλο τμήμα κώδικα για την εξυπηρέτησή του.** |
| **ορθότητα accuracy  1) Η ιδιότητα αυτού που είναι απαλλαγμένο από σφάλμα.  2) Ποιοτική αξιολόγηση της απαλλαγής από σφάλμα. Υψηλή ορθότητα αντιστοιχεί σε μικρό σφάλμα.** |
| **ουρά queue  Δομή δεδομένων με δύο άκρα στην οποία το πρώτο στοιχείο που εισάγεται είναι και το πρώτο που μπορεί να εξαχθεί.** |
| **παγίδευση σφάλματος error trapping  Μια διαδικασία με την οποία τα σφάλματα που δημιουργούνται κατά τη διάρκεια εκτέλεσης μιας εφαρμογής, οδηγούνται σε ειδικό τμήμα του κώδικα της εφαρμογής που καλείται χειριστής σφάλματος.  Ο τελευταίος εκτελεί κάποια προκαθορισμένη λειτουργία, όπως για παράδειγμα να αγνοήσει το σφάλμα.**  **60 / 270-271** |
| **πακέτο λογισμικού software package  Πλήρες και τεκμηριωμένο σύνολο προγραμμάτων που παρέχεται σε έναν αριθμό χρηστών για μια εφαρμογή.** |
| **πειρατεία λογισμικού software piracy  Παράνομη αναπαραγωγή προϊόντων λογισμικού.** |
| **περιβάλλον ανάπτυξης λογισμικού  software** **development** **environment  Σύνολο μεταφραστικών προγραμμάτων και άλλων εργαλείων ανάπτυξης λογισμικού που χρησιμοποι-ούνται στη δημιουργία προγραμμάτων εφαρμογών.** |
| **πίνακας table  Παράθεση δεδομένων καθένα από τα οποία μπορεί να προσδιοριστεί μονοσήμαντα μέσω μιας ή περισ-σότερων μεταβλητών.** |
| **πληροφορία (ες) Information  Γνώση που αφορά πράγματα όπως πράξεις, έννοιες, αντικείμενα, γεγονότα, ιδέες ή διεργασίες που μέσα σε συγκεκριμένο κείμενο έχουν μια ιδιαίτερη σημασία.** |
| **πρόγραμμα (υπολογιστή) program  Ακολουθία εντολών κατάλληλων για επεξεργασία. Η επεξεργασία περιλαμβάνει τη χρήση μεταφραστικού προγράμματος για να προετοιμάσει το πρόγραμμα για εκτέλεση, καθώς και την ίδια την εκτέλεση του προγράμματος.**  **61 / 271** |
| **προγραμματισμός programming  Η διαδικασία δημιουργίας προγραμμάτων υπολο-γιστή.** |
| **προγραμματιστής programmer**  **Πρόσωπο υπεύθυνο για το σχεδιασμό, εγγραφή, έλεγχο, διόρθωση, συντήρηση και τεκμηρίωση ενός προγράμματος.**  **62 / 271** |
| **προδιαγραφή προγράμματος program specification  Τεκμήριο που περιγράφει τη δομή και λειτουργίες ενός προγράμματος με επαρκή λεπτομέρεια, ώστε να επιτρέπει τον προγραμματισμό και να διευκολύνει τη συντήρηση.** |
| **προσάρτηση append Προσθήκη στοιχείων στο τέλος μιας δομής δεδομένων** |
| **προσέγγιση από κάτω προς τα πάνω bottom-up approach  Προσέγγιση στο σχεδιασμό συστημάτων η οποία αρχίζει με την αναγνώριση των βασικών συναλ-λαγών και αναγκών σε επεξεργασία πληροφοριών και στη συνέχεια την ολοκλήρωση τους σε όλα και υψηλότερο επίπεδο.** |
| **προσπέλαση access  Πρόσβαση σε δεδομένα με σκοπό την ανάγνωση ή μετακίνηση δεδομένων ή εντολών.** |
| **σημείο διακοπής breakpoint  Ένα σημείο σε ένα πρόγραμμα στο οποίο σταματάει η εκτέλεση του προγράμματος.** |
| **σταθερά constant  Γλωσσικό αντικείμενο που παίρνει μόνο μια ειδική τιμή.**  **63 / 271** |
| **στοίβα stack  Δομή δεδομένων με ένα άκρο στην οποία το τελευ-ταίο στοιχείο που εισάγεται είναι και το πρώτο που μπορεί να εξαχθεί.** |
| **στοιχειοσειρά string  Γραμμική ακολουθία στοιχείων όπως χαρακτήρες, δυαδικά ψηφία ή άλλα φυσικά στοιχεία.** |
| **συγχώνευση merging  Η διαδικασία συνδυασμού δύο ταξινομημένων συνό-λων δεδομένων για την παραγωγή ενός ταξινομη-μένου συνόλου.** |
| **συμβάν  Βλ. γεγονός** |
| **συμβολική γλώσσα assembly language  Γλώσσα χαμηλού επιπέδου εξαρτώμενη από το υλικό και η οποία έχει άμεση αντιστοιχία με τη γλώσσα μηχανής. Αποτελεί συμβολική αναπα-ράσταση του δυαδικού κώδικα της γλώσσας μηχανής και χρειάζεται συμβολομετάφραση.** |
| **συμβολομεταφραστής assembler Πρόγραμμα που μεταφράζει συμβολική γλώσσα σε γλώσσα μηχανής του δεδομένου υπολογιστή.**  **64 / 272** |
| **συνθήκη condition  Μια έκφραση σε πρόγραμμα ή διαδικασία που μπορεί να εκτιμηθεί είτε ως αληθής είτε ως ψευδής, όταν εκτελείται το πρόγραμμα ή η διαδικασία.** |
| **σφάλμα bug  Λάθος ή ελάττωμα προγράμματος.** |
| **σχεδίαση προγράμματος program design  Τα βήματα που αναγνωρίζουν τους πόρους υλικού και λογισμικού που απαιτούνται από το πρόγραμμα. Η σχεδίαση προγράμματος προσδιορίζει τη λογική που χρησιμοποιείται από το πρόγραμμα.** |
| **ταξινόμηση sorting  Η διαδικασία τοποθέτησης των στοιχείων δεδομένων σε μια δομή δεδομένων με αύξουσα ή φθίνουσα σειρά.** |
| **τεκμηρίωση προγράμματος program documentation  Έγγραφα ή άλλα μέσα που παρέχουν την περιγραφή του προγράμματος.** |
| **Τελεστέος operand  Μια οντότητα στην οποία εφαρμόζεται μια πράξη.** |
| **Τελεστής operator  Σύμβολο που παριστάνει τη φύση μιας πράξης που πρόκειται να εκτελεστεί.** |
| **τεχνητή νοημοσύνη artificial intelligence  Προσπάθειες ώστε να καταστεί ο υπολογιστής ικανός για λειτουργίες που αποδίδονται σε ανθρώπινη νοημοσύνη.** |
| **υποπρόγραμμα subprogram  Ένα πρόγραμμα καλούμενο από άλλο πρόγραμμα, σε αντίθεση με ένα κύριο πρόγραμμα.** |
| **φυσική γλώσσα natural language**  **Γλώσσα οι κανόνες της οποίας βασίζονται στην τρέ-χουσα χρήση, χωρίς να είναι αυστηρά προδιαγε-γραμμένοι.** |
| **φώλιασμα nesting  Η ένθεση βρόχου ή υπορουτινών μέσα σε άλλους βρόχους ή υπορουτίνες.** |
| **ψευδοκώδικας pseudocode  Τρόπος αποτύπωσης αλγορίθμων με χρήση προκα-θορισμένων λέξεων κλειδιών.** |

**65 / 272**

|  |
| --- |
| **Αγγλοελληνικό λεξικό όρων** |

|  |  |  |  |
| --- | --- | --- | --- |
| **a posteriori** | **εκ των υστέρων** | | |
| **a priori** | **εκ των προτέρων** | | |
| **abstraction** | **αφαίρεση (νοητική)** | | |
| **access** | **προσπέλαση** | | |
| **accessing** | **διαδικασία προσπέλασης** | | |
| **accidence** | **τυπικό, τυπολογικό** | | |
| **accuracy** | **ορθότητα** | | |
| **activation** | **ενεργοποίηση** | | |
| **actual** | **πραγματικός** | | |
| **aid** | **βοήθεια** | | |
| **algorithm** | **αλγόριθμος** | | |
| **alignment** | **ευθυγράμμιση, στοίχιση** | | |
| **alphanumeric** | **αλφαριθμητικό** | | |
| **alteration** | **εναλλαγή, μετατροπή, τροποποίηση** | | |
| **ambersand** | **σύμβολο &** | | |
| **ambiguious** | **διφopoύμεvoς, ασαφής** | | |
| **ambiguity** | **ασάφεια** | | |
| **analysis** | **ανάλυση** | | |
| **analyst** | **αναλυτής** | | |
| **analytic** | **αναλυτικός** | | |
| **animation** | **κίνηση** | | |
| **annotation** | **σχόλιο** | | |
| **append** | **προσάρτηση** | | |
| **application** | **εφαρμογή** | | |
| **approximate** | **προσεγγιστικός** | | |
| **argument** | **όρισμα** | | |
| **array** | **πίνακας** | | |
| **artificial intelligence** | **τεχνητή νοημοσύνη** | | |
| **assembler** | **συμβολομεταφραστής** | | |
| **assembly language**  **66 / 273** | **συμβολική γλώσσα** | | |
| **assignement** | **εκχώρηση** | | |
| **at sign** | **σύμβολο @** | | |
| **attribute** | **ιδιοχαρακτηριστικό** | | |
| **authentication** | **επαλήθευση ακεραιότητας (δεδομένων)** | | |
| **authoring languages** | **γλώσσες συγγραφής** | | |
| **automation** | **αυτοματισμός** | | |
| **auxiliary** | **βοηθητικός** | | |
| **background** | **παρασκήνιο, υπόβαθρο** | | |
| **backslash** | **ανάποδη κάθετος \** | | |
| **backtracking** | **οπισθοδρόμηση** | | |
| **backup** | **εφεδρεία** | | |
| **bar chart** | **ραβδοδιάγραμμα** | | |
| **bell** | **ηχητικό σήμα, κουδούνι** | | |
| **benchmark** | **δοκιμή επιδόσεων** | | |
| **beta test** | **έλεγχος βήτα** | | |
| **blank (character)** | **χαρακτήρας κενού** | | |
| **blinking** | **αναβόσβημα** | | |
| **block** | **ομάδα** | | |
| **bookmark** | **καταχωρώ σημάδι** | | |
| **boolean (data tγpe)** | **λογικός τύπος δεδομένου** | | |
| **box** | **πλαίσιο** | | |
| **braces** | **άγκιστρα {}** | | |
| **brackets** | **τετραγωνικά άγκιστρα[]** | | |
| **break** | **διακοπή** | | |
| **break key** | **πλήκτρο διακοπής** | | |
| **breakpoint** | **σημείο διακοπής** | | |
| **browse** | **ξεφυλλίζω, φυλλομετρώ** | | |
| **bubblesort** | **ταξινόμηση φυσαλίδας** | | |
| **bug** | **σφάλμα** | | |
| **build-in functions** | **ενσωματωμένες συναρτήσεις**  **67 / 273** | | |
| **business applications**  **68 / 273** | **επιχειρησιακές εφαρμογές** | | |
| **button** | **κουμπί** | | |
| **call** | **κλήση** | | |
| **call-by-reference** | **κλήση δι' αναφοράς** | | |
| **call-by-value** | **κλήση διά τιμής** | | |
| **cancel** | **ακύρωση** | | |
| **caret** | **σύμβολο ^** | | |
| **carriage return character** | **χαρακτήρας επιστροφής** | | |
| **catalog** | **κατάλογος** | | |
| **cell** | **κύτταρο, κελί** | | |
| **certification** | **πιστοποίηση** | | |
| **character code** | **κώδικας χαρακτήρων** | | |
| **character set** | **σύνολο χαρακτήρων** | | |
| **chart** | **διάγραμμα** | | |
| **character** | **χαρακτήρας** | | |
| **check digit** | **ψηφίο ελέγχου** | | |
| **check sum** | **άθροισμα ελέγχου** | | |
| **checking** | **έλεγχος** | | |
| **choice** | **επιλογή** | | |
| **class** | **κλάση, τάξη** | | |
| **clear** | **καθαρίζω** | | |
| **client** | **πελάτης** | | |
| **clipping** | **ψαλίδιση** | | |
| **code** | **κώδικας** | | |
| **coding** | **κωδικοποίηση** | | |
| **collate** | **διαταξινομώ** | | |
| **colon** | **χαρακτήρας :** | | |
| **column** | **στήλη** | | |
| **combination** | **συνδυασμός** | | |
| **combining** | **συνένωση, συνδυασμός** | | |
| **comma** | **κόμμα** | | |
| **command** | **εντολή, διαταγή** | | |
| **command driven** | **καθοδηγούμενο από εντολές** | | |
| **command language** | **γλώσσα εντολών** | | |
| **command line** | **γραμμή εντολών** | | |
| **comment** | **σχόλιο** | | |
| **compact** | **σύμπυκνος** | | |
| **compare** | **συγκρίνω** | | |
| **compatibility** | **συμβατότητα** | | |
| **compatible** | **συμβατός** | | |
| **compilation** | **μεταγλώττιση** | | |
| **compiler** | **μεταγλωττιστής** | | |
| **computability** | **δυνατότητα υπολογισμού** | | |
| **computational** | **υπολογιστικός** | | |
| **computational complexity** | **υπολογιστική πολυπλοκότητα** | | |
| **computer instruction set** | **σύνολο εντολών υπολογιστή** | | |
| **computer science** | **επιστήμη υπολογιστών** | | |
| **concatenation** | **συναλύσσωση** | | |
| **concept** | **έννοια** | | |
| **conception** | **ιδέα, σύλληψη** | | |
| **conceptual** | **εννοιολογικός** | | |
| **condition** | **συνθήκη** | | |
| **conditional branch** | **διακλάδωση υπό συνθήκη** | | |
| **conjuction** | **τομή, σύζευξη, πράξη ΚΑΙ, λογικός πολ/σμός** | | |
| **constant** | **σταθερά** | | |
| **constraint** | **περιορισμός** | | |
| **consultant** | **σύμβουλος** | | |
| **context** | **συμφραζόμενα** | | |
| **continue** | **συνεχίζω** | | |
| **control character**  **69 / 273-274** | **χαρακτήρας ελέγχου** | | |
| **control point** | **σημείο ελέγχου** | | |
| **control variable** | **μεταβλητή ελέγχου** | | |
| **convention** | **σύμβαση** | | |
| **conversational** | **διαλογικός** | | |
| **conversion** | **μετατροπή** | | |
| **copying** | **αντιγραφή** | | |
| **correction** | **διόρθωση** | | |
| **counter** | **απαριθμητής, μετρητής** | | |
| **create** | **δημιουργώ** | | |
| **cursor**  **70 / 274** | **δρομέας** | | |
| **customer** | **πελάτης** | | |
| **cut** | **κόβω, αποκοπή** | | |
| **cut-and-paste** | **αποκοπή και συγκόλληση** | | |
| **data** | **δεδομένα** | | |
| **data base** | **βάση δεδομένων** | | |
| **data capture** | **συλλογή δεδομένων** | | |
| **data compression** | **συμπίεση δεδομένων** | | |
| **data definition** | **ορισμός δεδομένων** | | |
| **data entry** | **εισαγωγή δεδομένων** | | |
| **data flow** | **ροή δεδομένων** | | |
| **data processing** | **επεξεργασία δεδομένων** | | |
| **data structure** | **δομή δεδομένων** | | |
| **data type** | **τόπος δεδομένου** | | |
| **date** | **ημερομηνία** | | |
| **debugger** | **εκσφαλματωτής, διορθωτής** | | |
| **debugging** | **εκσφαλμάτωση** | | |
| **decision** | **απόφαση** | | |
| **declaration** | **δήλωση** | | |
| **decrement** | **ελάττωση** | | |
| **default** | **προτερότιμος** | | |
| **default value** | **προτεροτιμή** | | |
| **definiteness** | **καθοριστικότητα** | | |
| **definition** | **ορισμός** | | |
| **delete** | **εξαλείφω, διαγράφω** | | |
| **deletion** | **διαγραφή** | | |
| **delimiter** | **διαχωριστής** | | |
| **demo (demostration)** | **επίδειξη** | | |
| **design** | **σχεδίαση** | | |
| **development** | **ανάπτυξη** | | |
| **diagnostics** | **διαγνωστικά (μηνύματα)** | | |
| **diagram** | **διάγραμμα** | | |
| **diagramming technique** | **διαγραμματική τεχνική** | | |
| **digit** | **ψηφίο** | | |
| **digital** | **ψηφιακός** | | |
| **dimension** | **διάσταση** | | |
| **directive** | **οδηγία** | | |
| **directory** | **ευρετήριο** | | |
| **discrete** | **διάκριτος** | | |
| **disjunction** | **διάζευξη, λογική πρόσθεση,πράξη Ή** | | |
| **display** | **oπτική παρουσίαση** | | |
| **divide and conquer** | **διαίρει και βασίλευε** | | |
| **document** | **έγγραφο, τεκμήριο** | | |
| **documentation** | **τεκμηρίωση** | | |
| **down load** | **μεταφορτώνω** | | |
| **dummy instruction** | **πλασματική εντολή** | | |
| **dump** | **αποτυπώνω** | | |
| **duplicate** | **αναπαράγω** | | |
| **dyadic operation** | **διτελής πράξη** | | |
| **dynamic** | **δυναμικός** | | |
| **editing** | **σύνταξη προγραμμάτων, δεδομένων)** | | |
| **editor** | **συντάκτης**  **71 / 274** | | |
| **effectiveness** | **αποτελεσματικότητα** | | |
| **efficiency** | **αποδοτικότητα, ικανότητα**  **72 / 274** | | |
| **element** | **στοιχείο, μέλος (συνόλου)** | | |
| **embedding** | **ενσωμάτωση** | | |
| **empirical** | **εμπειρικός** | | |
| **emulation** | **εξομοίωση** | | |
| **encapsulation** | **ενθυλάκωση** | | |
| **enhancements** | **βελτιώσεις, επαυξήσεις** | | |
| **entity** | **οντότητα** | | |
| **entry** | **είσοδος, εισαγωγή δεδομένων** | | |
| **environment** | **περιβάλλον** | | |
| **equal** | **ίσος** | | |
| **equation** | **εξίσωση** | | |
| **equivalence** | **ισοδυναμία** | | |
| **erase** | **σβήνω** | | |
| **error** | **σφάλμα** | | |
| **error trapping** | **παγίδευση σφάλματος** | | |
| **escape** | **διαφυγή** | | |
| **evaluation** | **αξιολόγηση, αποτίμηση** | | |
| **even** | **ζυγός, άρτιος** | | |
| **event** | **γεγονός, συμβάν** | | |
| **exclusion** | **αποκλεισμός** | | |
| **executable** | **εκτελέσιμος** | | |
| **execution** | **εκτέλεση** | | |
| **exit** | **έξοδος** | | |
| **expandability** | **επεκτασιμότητα** | | |
| **exponent** | **εκθέτης** | | |
| **expression** | **έκφραση** | | |
| **factorial** | **παραγοντικό** | | |
| **failure** | **αποτυχία, αστοχία** | | |
| **false** | **ψευδής** | | |
| **fault** | **ελάττωμα, σφάλμα** | | |
| **feature** | **χαρακτηριστικό** | | |
| **feed** | **τροφοδότηση** | | |
| **feedback** | **ανατροφοδότηση, ανάδραση** | | |
| **field** | **πεδίο** | | |
| **file** | **αρχείο** | | |
| **finiteness** | **περατότητα** | | |
| **flag** | **σημαία, ενδείκτης** | | |
| **flow chart** | **διάγραμμα ροής** | | |
| **font** | **γραμματοσειρά** | | |
| **foreground** | **προσκήνιο** | | |
| **form** | **έντυπο, μορφή, σχήμα, τύπος** | | |
| **formal** | **τυπικός** | | |
| **format** | **μορφότυπο** | | |
| **formatting** | **μορφοτύπηση** | | |
| **free text** | **ελεύθερο κείμενο** | | |
| **function** | **συνάρτηση** | | |
| **global** | **καθολικός** | | |
| **graph** | **γράφος** | | |
| **greedy method** | **άπληστη μέθοδος** | | |
| **hardware** | **υλικό** | | |
| **heap** | **σωρός** | | |
| **help** | **βοήθεια** | | |
| **heuristic** | **ευριστικός** | | |
| **hyphen** | **παύλα** | | |
| **hyphenation** | **υφενισμός, συλλαβισμός** | | |
| **icon** | **εικονίδιο** | | |
| **identifier** | **αναγνωριστικό (ταυτότητας)** | | |
| **ignore** | **αγνοώ** | | |
| **image**  **73 / 274-275** | **εικόνα, είδωλο** | | |
| **immediate** | **άμεσος** | | |
| **implication** | **συνεπαγωγή** | | |
| **implicit** | **εννooύμεvoς, αφανής** | | |
| **implied** | **υπονοούμενος** | | |
| **improvement** | **βελτίωση** | | |
| **incompatibility** | **ασυμβατότητα** | | |
| **indent** | **εσοχή** | | |
| **index** | **δείκτης** | | |
| **infinite lοορ** | **ατέρμονας βρόχος** | | |
| **information** | **πληροφορία** | | |
| **information processing** | **επεξεργασία πληροφοριών** | | |
| **inheritance** | **κληρονομικότητα** | | |
| **input** | **είσοδος** | | |
| **insertion** | **εισαγωγή, παρεμβολή** | | |
| **installation** | **εγκατάσταση** | | |
| **instance** | **στιγμιότυπο** | | |
| **instruction** | **εντολή** | | |
| **integer** | **ακέραιος** | | |
| **interaction** | **αλληλεπίδραση** | | |
| **interactive** | **διαλογικός** | | |
| **interface** | **διεπαφή** | | |
| **interpreter** | **διερμηνευτής** | | |
| **interrupt** | **διακοπή** | | |
| **intractable** | **δυσχείριστος** | | |
| **intricic** | **εσώτερος, ενυπάρχων, εγγενής** | | |
| **inverse** | **αντίστροφος** | | |
| **italics** | **πλάγιοι χαρακτήρες** | | |
| **item** | **στοιχείο δεδομένου** | | |
| **iteration** | **επανάληψη** | | |
| **job** | **εργασία**  **74 / 275** | | |
| **join** | **ενώνω, συνδέω, ένωση** | | |
| **jump** | **άλμα** | | |
| **justification** | **στοίχιση** | | |
| **key** | **κλειδί** | | |
| **keyword** | **λέξη κλειδί** | | |
| **label**  **75 / 275** | **ετικέτα** | | |
| **landscape** | **τοπίο** | | |
| **library** | **βιβλιοθήκη** | | |
| **licence** | **άδεια** | | |
| **life cycle** | **κύκλος ζωής** | | |
| **link** | **σύνδεσμος, σύνδεση, ζεύξη** | | |
| **linked list** | **συνεζευγμένη λίστα** | | |
| **linker** | **συvδέτης** | | |
| **linking** | **διασύνδεση** | | |
| **list** | **λίστα** | | |
| **listing** | **παράθεση, καταλογράφηση** | | |
| **literal** | **κυριολεκτική σταθερά** | | |
| **loader** | **φορτωτής** | | |
| **local** | **τοπικός** | | |
| **logic** | **λογική** | | |
| **logical** | **λογικός** | | |
| **loop** | **βρόχος** | | |
| **machine language** | **γλώσσα μηχανής** | | |
| **macro (instruction)** | **μακροεντολή** | | |
| **main program** | **κύριο πρόγραμμα** | | |
| **maintenance** | **συντήρηση** | | |
| **manipulation** | **χειρισμός** | | |
| **manual** | **εγχειρίδιο, χειροκίνητος** | | |
| **map** | **απεικόνιση** | | |
| **mark** | **σημάδι** | | |
| **mask** | **μάσκα** | | |
| **matching** | **ταίριασμα, ταύτιση** | | |
| **matrix** | **μήτρα** | | |
| **menu** | **κατάλογος επιλογών, μενού** | | |
| **merging** | **συγχώνευση** | | |
| **mirroring** | **καθρεπτισμός** | | |
| **mode** | **κατάσταση, τρόπος** | | |
| **modify**  **76 / 275** | **τpoπoπoιώ, μεταβάλω** | | |
| **modular** | **δομοστοιχειωτός, τμηματικός** | | |
| **module** | **δoμoστoιχείo, δομική ενότητας** | | |
| **monadic operation** | **μονοτελής πράξη** | | |
| **monitor** | **μηνύτορας, παρακολουθώ** | | |
| **move** | **μετακινώ** | | |
| **multimedia** | **πολυμέσα** | | |
| **natural language** | **φυσική γλώσσα** | | |
| **negation** | **άρνηση, λογική αντιστροφή, πράξη ΟΧΙ** | | |
| **nested** | **εμφωλευμένος** | | |
| **nesting** | **φώλιασμα** | | |
| **nomenclature** | **ονοματολογία** | | |
| **numeric** | **αριθμητικός** | | |
| **object** | **αντικείμενο** | | |
| **object driven** | **οδηγούμενος από το γεγονός** | | |
| **object oriented** | **αvτικειμεvoστpαφής** | | |
| **odd** | **μονός, περιττός** | | |
| **office** | **γραφείο** | | |
| **office automation** | **αυτοματισμός γραφείου** | | |
| **O-notation** | **συμβολισμός Ο** | | |
| **operand** | **τελεστέος** | | |
| **operation** | **πράξη** | | |
| **operator** | **τελεστής, χειριστής** | | |
| **optimal** | **άριστος** | | |
| **option** | **επιλογή** | | |
| **optional** | **προαιρετικός** | | |
| **order** | **διάταξη** | | |
| **ordering** | **διάταξη** | | |
| **output** | **έξοδος, έξοδος δεδομένων** | | |
| **overflow** | **υπερχείλιση** | | |
| **overlay**  **77 / 275** | **επικαλύπτω** | | |
| **package** | **πακέτο** | | |
| **parameter** | **παράμετρος** | | |
| **parity** | **ισοτιμία** | | |
| **parsing** | **συντακτική ανάλυση** | | |
| **password** | **σύνθημα** | | |
| **paste** | **κολλώ, συγκολλώ** | | |
| **patch** | **διορθώνω πρόχειρα** | | |
| **pattern** | **σχηματομορφή** | | |
| **performance** | **επίδοση** | | |
| **permission** | **δικαίωμα** | | |
| **permutation** | **μετάθεση** | | |
| **planning** | **σχεδιασμός, προγραμματισμός (έργων)** | | |
| **platform** | **πλατφόρμα, υποδομή** | | |
| **pointer** | **δείκτης** | | |
| **polymorphism** | **πολυμορφισμός** | | |
| **polynomial** | **πολυωνυμικός** | | |
| **pop** | **απώθηση** | | |
| **portability** | **φορητότητα** | | |
| **portrait** | **πορτρέτο** | | |
| **precision** | **ακρίβεια** | | |
| **primary key** | **πρωτεύον κλειδί** | | |
| **print** | **τυπώνω, εκτυπώνω** | | |
| **procedure** | **διαδικασία** | | |
| **processing** | **επεξεργασία** | | |
| **program** | **πρόγραμμα** | | |
| **prompt** | **προτροπή** | | |
| **property** | **ιδιότητα** | | |
| **pseudocode** | **ψευδοκώδικας** | | |
| **pull down menu** | **κατακόρυφος πίνακας επιλογών** | | |
| **push** | **ώθηση**  **78 / 276** | | |
| **quantity** | **ποσότητα** | | |
| **queue** | **ουρά** | | |
| **quit** | **εγκαταλείπω, αφήνω** | | |
| **quotation mark** | **διπλά εισαγωγικά** | | |
| **radian** | **ακτίνιο** | | |
| **rank** | **τάξη** | | |
| **read** | **διαβάζω** | | |
| **ready** | **έτοιμος** | | |
| **real** | **πραγματικός** | | |
| **record** | **εγγραφή** | | |
| **recursion** | **αναδρομή** | | |
| **reference** | **αναφορά, παραπομπή** | | |
| **relation** | **σχέση** | | |
| **relative** | **σχετικός** | | |
| **release** | **έκδοση (λογισμικού)** | | |
| **reliability** | **αξιοπιστία** | | |
| **replica** | **αντίγραφο** | | |
| **reply** | **απαντώ** | | |
| **report** | **αναφορά, έκθεση, κατάσταση (εκτυπωτή)** | | |
| **requirement** | **απαίτηση** | | |
| **rerun** | **επανεκτέλεση** | | |
| **reserved** | **δεσμευμένος** | | |
| **reserved word** | **δεσμευμένη λέξη** | | |
| **restart** | **επανεκκίνηση** | | |
| **retrieval** | **ανάκτηση** | | |
| **return** | **επιστροφή** | | |
| **row** | **γραμμή, σειρά** | | |
| **rule** | **κανόνας** | | |
| **run** | **εκτέλεση** | | |
| **save** | **φυλάσσω** | | |
| **SCALAR** | **βαθμωτός, μονόμετρος**  **79 / 276** | | |
| **score** | **εμβέλεια** | | |
| **scroll bar** | **ράβδος κύλισης** | | |
| **scrolling** | **κύλιση** | | |
| **searching** | **αναζήτηση** | | |
| **secondary key** | **δευτερεύον κλειδί** | | |
| **semantics** | **σημασιολογία** | | |
| **separation** | **διαχωρισμός** | | |
| **sequence** | **ακολουθία, διαδοχή, σειρά** | | |
| **sequential** | **ακολουθιακός** | | |
| **serial** | **σειριακός** | | |
| **set** | **σύνολο** | | |
| **shift** | **ολίσθηση, μετατόπιση** | | |
| **simulation** | **προσομοίωση** | | |
| **software** | **λογισμικό** | | |
| **sorting** | **ταξινόμηση** | | |
| **source** | **πηγή** | | |
| **space complexity** | **πoλυπλoκότητα χώρου** | | |
| **specification** | **προδιαγραφή** | | |
| **SQL (Structure Query Language)** | **δομημένη γλώσσα ερωτοαπαντήσεων** | | |
| **stack** | **στοίβα** | | |
| **standardization** | | **τυποποίηση** | |
| **statement** | **εντολή, πρόταση (προγράμματος)** | | |
| **static** | **στατικός** | | |
| **status** | **κατάσταση** | | |
| **step** | **βήμα** | | |
| **straight exchange sort** | **ταξινόμηση ευθείας ανταλλαγής** | | |
| **string** | **στοιχειοσειρά** | | |
| **structural** | **δομημένος** | | |
| **structure programming** | **δομημένος προγραμματισμός** | | |
| **subprogram** | **υποπρόγραμμα**  **80 / 276** | | |
| **subroutine** | **υπορουτίνα** | | |
| **sum** | **άθροισμα** | | |
| **syntax** | **συντακτικό** | | |
| **table** | **πίνακας** | | |
| **testing** | **δοκιμή** | | |
| **text** | **κείμενο** | | |
| **time complexity** | **χρονική πολυπλοκότητα** | | |
| **tool** | **εργαλείο** | | |
| **toolbar** | **γραμμή εργαλείων** | | |
| **toolbox** | **εργαλειοθήκη** | | |
| **tracing** | **ιχνηλάτηση** | | |
| **transform** | **μετασχηματίζω** | | |
| **translate** | **μεταφράζω, μεταφέρω** | | |
| **translator** | **μεταφραστής** | | |
| **trapping** | **παγίδευση** | | |
| **tree** | **δένδρο** | | |
| **trial** | **δοκιμή** | | |
| **unary** | **μοναδιαίος, μονοτελής** | | |
| **unconditional jump** | **άλμα χωρίς συνθήκη** | | |
| **underflow** | **υποχείλιση** | | |
| **underscore** | **χαρακτήρας \_** | | |
| **undo** | **ανατρέπω** | | |
| **update** | **ενημερώνω** | | |
| **upgrade** | **αναβάθμιση** | | |
| **user** | **χρήστης** | | |
| **utility program** | **βοηθητικό πρόγραμμα** | | |
| **validation** | **επικύρωση** | | |
| **variable** | **μεταβλητή** | | |
| **verification** | **επαλήθευση** | | |
| **verify** | **επαληθεύω** | | |
| **version** | **εκδοχή, παραλλαγή, έκδοση** | | |
| **virus** | **ιός**  **81 / 276** | | |
| **visual** | **οπτικός** | | |
| **watchpoint** | **σημείο παρατήρησης** | | |
| **window** | **παράθυρο** | | |
| **write** | **γράφω** | | |
| **Βιβλιογραφία** | | |
| **Χ.Κοίλιας-Δ.Παναγιωτάκος, Ερμηνευτικό Λεξικό Όρων Πληροφορικής, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1994** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ευρετήριο** | | | |
| **al Khowarizmi** | **33** | | |
| **ALGOL** | **109** | | |
| **Allen P** | **109** | | |
| **BASIC** | **109** | | |
| **Bolzano** | **81** | | |
| **Clipper** | **112** | | |
| **COBOL** | **108** | | |
| **dBASE** | **112** | | |
| **Dijkstra E** | **117** | | |
| **Fibonacci** | **70** | | |
| **FIFO** | **61,62** | | |
| **FORTRAN** | **107,108** | | |
| **Gates B** | **109** | | |
| **GOTO** | **118** | | |
| **HIPO** | **258** | | |
| **Hopper G.M** | **108** | | |
| **JAVA** | **111** | | |
| **Kemeny G**  **82 / 277** | **109** | | |
| **Kurtz T.** | **109** | | |
| **LIFO** | **60,62** | | |
| **LISP** | **109,110** | | |
| **LOGO** | **110** | | |
| **ΝΡ-πλήρες πρόβλημα** | **100** | | |
| **OCCAM** | **120** | | |
| **Papert S** | **110** | | |
| **PASCAL** | **110** | | |
| **PROLOG** | **109** | | |
| **Ritchie D.** | **111** | | |
| **SMALTALK** | **196** | | |
| **SQL** | **113** | | |
| **VTOC** | **258** | | |
| **Wirth N.** | **57,110** | | |
| **A** |  | | |
| **ακολουθία** | **36** | | |
| **αλγοριθμική γλώσσα** | **114** | | |
| **αλγόριθμος** | **33** | | |
| **αλφάβητο**  **83 / 277** | **115,127** | | |
| **άμεση εκτέλεση** | **237** | | |
| **αναδρομή** | **67,184** | | |
| **αναζήτηση** | **63** | | |
| **ανάλυση προβλήματος** | **25,77** | | |
| **ανάλυση-σχεδίαση** | **262** | | |
| **αντικείμενο** | **194** | | |
| **αντικείμενο πρόγραμμα** | **121** | | |
| **αντικειμενοστραφής προγραμματισμός** | **111,119,193** | | |
| **αξιολόγηση** | **243** | | |
| **αξιοπιστία** | **249** | | |
| **άπληστη μέθοδος** | **84** | | |
| **απλότητα** | **243** | | |
| **από επάνω προς τα κάτω** | **80** | | |
| **από κάτω προς τα επάνω** | **82** | | |
| **αποδοτικότητα** | **91** | | |
| **αποτελεσματικότητα** | **33** | | |
| **απώθηση** | **61** | | |
| **αριθμητικός τελεστής**  **84 / 277** | **131** | | |
| **αριθμοί Fibonacci** | **70** | | |
| **αρχείο δεδομένων** | **66** | | |
| **αφηρημένος τύπος** | **197** | | |
| **Β** |  | | |
| **βήμα προς βήμα εκτέλεση** | **236** | | |
| **βιβλιοθήκη** | **121** | | |
| **βρόχος** | **45** | | |
| **Γ** |  | | |
| **γεγονός** | **198** | | |
| **γλώσσα 4ης γενιάς** | **113** | | |
| **- ερωτοααπαντήσεων** | **114** | | |
| **- μηχανής** | **106** | | |
| **- υψηλού επιπέδου** | **107** | | |
| **- χαμηλού επιπέδου** | **106** | | |
| **γραμματική** | **115** | | |
| **γραφικό περιβάλλον** | **207** | | |
| **γράφος** | **73** | | |
| **Δ** |  | | |
| **δεδομένα**  **85 / 277** | **19** | | |
| **δείκτης** | **58,71,155** | | |
| **δένδρο** | **72** | | |
| **διάγραμμα ροής** | **36** | | |
| **διαγραμματική αναπαράσταση** | | | **20** |
| **διαγραμματική τεχνική** | **35** | | |
| **διαδικασιακή γλώσσα** | **114** | | |
| **διαδικασίες** | **175** | | |
| **διαίρει και βασίλευε** | **80** | | |
| **διασύνδεση** | **210** | | |
| **διεπαφή χρήστη** | **201,217** | | |
| **διερμηνευτής** | **120** | | |
| **διεύθυνση επιστροφής** | **182** | | |
| **δομή δεδομένων** | **55** | | |
| **δομή προβλήματος** | **19** | | |
| **δομή προγράμματος** | **134** | | |
| **δομημένος προγραμματισμός** | **110,117** | | |
| **δυαδική αναζήτηση** | **80** | | |
| **δυναμική δομή δεδομένων** | **72**  **86 / 277** | | |
| **δυναμικός προγραμματισμός** | | **82** | |
| **Ε** |  | | |
| **εγγραφή** | **66** | | |
| **είσοδος** | **33,133** | | |
| **έκδοση λογισμικού** | **260** | | |
| **εκσφαλμάτωση** | **235** | | |
| **εκτελέσιμο πρόγραμμα** | **121** | | |
| **έκφραση** | **37,131** | | |
| **έκφραση ελέγχου** | **235** | | |
| **εκχώρηση** | **132** | | |
| **έλεγχος** | **262** | | |
| **ελεύθερο κείμενο** | **35** | | |
| **εμβέλεια** | **182** | | |
| **εμφωλευμένη διαδικασία** | **42** | | |
| **ενθυλάκωση** | **197** | | |
| **ενσωμάτωση** | **211** | | |
| **έξοδος** | **33,133** | | |
| **επανάληψη** | **44,145** | | |
| **επεξεργασία δεδομένων**  **87 / 277** | **19** | | |
| **επίδοση αλγορίθμου** | **89** | | |
| **επιλογή** | **38,139** | | |
| **Ερατοσθένης** | **33** | | |
| **ευελιξία** | **246** | | |
| **Ευκλείδης** | **33,68** | | |
| **ευριστικός αλγόριθμος** | **101** | | |
| **Ι** |  | | |
| **ιδιότητα** | **196** | | |
| **ιεραρχία** | **132** | | |
| **ιεραρχική σχεδίαση** | **116** | | |
| **ιστορικό** | **236** | | |
| **ιχνηλάτηση** | **236** | | |
| **Κ** |  | | |
| **καθορισμός απαιτήσεων** | **21** | | |
| **καθοριστικότητα** | **33** | | |
| **κατανόηση προβλήματος** | **17** | | |
| **κλάση** | **196** | | |
| **κλειδί** | **66** | | |
| **κληρονομικότητα**  **88 / 277-278** | **197** | | |
| **κύκλος ζωής** | **260** | | |
| **κωδικοποίηση** | **35** | | |
| **Λ** |  | | |
| **λάθη** | **233** | | |
| **λεξιλόγιο** | **115** | | |
| **λίστα** | **71** | | |
| **λογικά λάθη** | **234** | | |
| **λογική έκφραση** | **139** | | |
| **Μ** |  | | |
| **μέγιστος κοινός διαιρέτης** | **68** | | |
| **μέθοδος** | **195** | | |
| **μέθοδος διχοτόμησης** | **81** | | |
| **μενού επιλογών** | **208** | | |
| **μεταβλητή** | **37, 129** | | |
| **μεταγλωττιστής** | **120** | | |
| **μεταφερσιμότητα** | **113** | | |
| **μη πολυωνυμικός αλγόριθμος** | **100** | | |
| **Ο**  **89 / 278** |  | | |
| **οδηγούμενος από το γεγονός προγραμματισμός** | **112, 198** | | |
| **ολίσθηση** | **48** | | |
| **ονόματα** | **129** | | |
| **οπτικός προγραμματισμός** | **112** | | |
| **ορθότητα** | **92** | | |
| **ουρά** | **61** | | |
| **Π** |  | | |
| **παραγοντικό** | **68, 185** | | |
| **παράλληλος αλγόριθμος** | **99** | | |
| **παράλληλος προγραμματισμός** | **120** | | |
| **παράμετροι** | **174** | | |
| **πεδίο** | **66** | | |
| **περατότητα** | **33** | | |
| **πηγαίο πρόγραμμα** | **121** | | |
| **πίνακας** | **58, 155** | | |
| **πίνακας ASCII** | **266** | | |
| **πληροφορία** | **19, 55** | | |
| **πολυμορφισμός**  **90 / 278** | **197** | | |
| **πολυπλοκότητα** | **95** | | |
| **πολυωνυμικός αλγόριθμος** | **99** | | |
| **πραγματικές παράμετροι** | **180** | | |
| **πρόβλημα** | **15** | | |
| **πρόβλημα έτους 2000** | **16, 234, 257** | | |
| **προβλήματα αδόμητα** | **26** | | |
| **- άλυτα** | **26** | | |
| **- ανοικτά** | **25** | | |
| **- απόφασης**  **91 / 278** | **26** | | |
| **- βελτιστοποίησης** | **27** | | |
| **- δομημένα** | **26** | | |
| **- δυσχείριστα** | **100** | | |
| **- επιλύσιμα** | **25** | | |
| **- ημιδομημένα** | **26** | | |
| **- υπολογιστικά** | **27** | | |
| **πρόγραμμα** | **105** | | |
| **προγραμματιστικό περιβάλλον** | **120** | | |
| **προσεγγιστικός αλγόριθμος** | **100** | | |
| **Σ** |  | | |
| **σειριακή αναζήτηση** | **63** | | |
| **σημασιολογία** | **115** | | |
| **σημείο διακοπής** | **236** | | |
| **σταθερά** | **37, 128** | | |
| **σταθμική μέση τιμή** | **63** | | |
| **στατική δομή** | **58** | | |
| **στοίβα** | **60, 182, 187** | | |
| **συμβολική γλώσσα** | **106** | | |
| **συμβολισμός Ο** | **95** | | |
| **συμβολομεταφραστής** | **106** | | |
| **συναρτήσεις** | **131, 175** | | |
| **συναρτησιακή γλώσσα**  **92 / 277** | **114** | | |
| **συνδέτης** | **121** | | |
| **σύνθετη έκφραση** | **133** | | |
| **συντάκτης** | **122** | | |
| **συντακτικό** | **115** | | |
| **συντήρηση** | **263** | | |
| **Τ** |  | | |
| **ταξινόμηση** | **65** | | |
| **ταξινόμηση φυσαλίδας** | **67** | | |
| **ταχύτητα** | **253** | | |
| **τεκμηρίωση** | **255** | | |
| **τελεστέος** | **37** | | |
| **τελεστής** | **37** | | |
| **τεχνητή γλώσσα** | **115** | | |
| **τμηματικός προγραμματισμός** | **116, 171** | | |
| **τυπικές παράμετροι** | **180** | | |
| **τυπικότητα** | **243** | | |
| **τύποι δεδομένων**  **93 / 277** | **128** | | |
| **τυπολογικό** | **115** | | |
| **Υ** |  | | |
| **υλοποίηση** | **262** | | |
| **υπολογιστική πολυπλοκότητα** | **99** | | |
| **υποπρόγραμμα** | **173** | | |
| **Φ** |  | | |
| **φορτωτής** | **121** | | |
| **φυσική γλώσσα** | **35,115** | | |
| **Ψ** |  | | |
| **ψευδογλώσσα** | **49** | | |
| **ψηφίο ελέγχου** | **255** | | |
| **Ω** |  | | |
| **ώθηση Κ** | **61** | | |

**94 / 277**

|  |  |  |
| --- | --- | --- |
| **Περιεχόμενα 3ου Τόμου** | | |
| **10.** | **Πίνακες** | **5** |
|  | **10.1. Τμηματικός προγραμματισμός** | **8** |
|  | **10.2. Χαρακτηριστικά των υποπρογραμμάτων** | **12** |
|  | **10.3. Πλεονεκτήματα του τμηματικού προγραμματισμού** | **13** |
|  | **10.4. Παράμετροι** | **15** |
|  | **10.5. Διαδικασίες και συναρτήσεις** | **16** |
|  | **10.5.1 Ορισμός και κλήση συναρτήσεων** | **20** |
|  | **10.5.2 Ορισμός και κλήση διαδικασιών** | **21** |
|  | **10.5.3 Πραγματικές και τυπικές παράμετροι** | **25** |
|  | **10.6. Εμβέλεια μεταβλητών - σταθερών** | **32** |
|  | **10.7. Αναδρομή** | **32** |
|  | **ΠΑΡΑΡΤΗΜΑ** | **53** |

**95**

**Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔIΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946, 108, Α').**

**Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήμα-τος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας και Θρησκευμάτων / IΤΥΕ - ΔΙΟΦΑΝΤΟΣ.**