ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Γ΄ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ

Τεχνολογικής Κατεύθυνσης

Βιβλίο Καθηγητή



ΔΙΑΝΕΜΕΤΑΙ ΔΩΡΕΑΝ

Ενέργεια 1.1.α: **«Προγράμματα Βιβλία»**
– Επιστημονικός Υπεύθυνος Ενέργειας: *Θεόδωρος Γ. Εξαρχάκος,*
Καθηγητής του Πανεπιστημίου Αθηνών,
Πρόεδρος του Παιδαγωγικού Ινστιτούτου

Έργο Νο 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 Ι. Παπασπύρου,*
Καθηγητής Εφαρμογών του ΤΕΙ Ηπείρου

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Ομάδα Συγγραφής

ΑΘΗΝΑ ΒΑΚΑΛΗ, Λέκτωρ Πληροφορικής ΑΠΘ

ΗΛΙΑΣ ΓΙΑΝΝΟΠΟΥΛΟΣ, Μηχανικός Πληροφορικής

ΝΕΣΤΩΡ ΙΩΑΝΝΙΔΗΣ, Επίκουρος Καθηγητής Τμήματος Πληροφορικής ΤΕΙ Αθήνας

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

ΓΙΩΡΓΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ (κατά τη συγγραφή)

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.

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

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βιβλίο Καθηγητή

Γ΄ Γενικού Λυκείου

(Τεχνολογικής Κατεύθυνσης)

Περιεχόμενα

Εισαγωγικό σημείωμα 9
Γενικός σκοπός μαθήματος 13
1. Ανάλυση Προβλήματος 25
2. Βασικές έννοιες αλγορίθμων 39
3. Δομές Δεδομένων και Αλγόριθμοι 75
4. Τεχνικές Σχεδίασης Αλγορίθμων 109
5. Ανάλυση αλγορίθμων 129
6. Εισαγωγή στον προγραμματισμό 145
7. Βασικές έννοιες προγραμματισμού 155
8. Επιλογή και επανάληψη 165
9. Πίνακες 181
10. Υποπρογράμματα 193
11. Σύγχρονα προγραμματιστικά περιβάλλοντα 207
12. Σχεδίαση διεπαφής χρήστη 223
13. Εκσφαλμάτωση προγράμματος 233
14. Αξιολόγηση - Τεκμηρίωση 241
ΠΑΡΑΡΤΗΜΑ 255

Εισαγωγικό σημείωμα

Αγαπητέ συνάδελφε,

Το βιβλίο που κρατάς στα χέρια σου, το βιβλίο καθηγητή, αποτελεί ένα από τα τρία βιβλία που θα συναποτελούν πλέον το διδακτικό πακέτο καθενός από τα μαθήματα του προγράμματος σπουδών της Δευτεροβάθμιας Εκπαίδευσης. Το βιβλίο καθηγητή, μαζί με το τετράδιο εργασιών για το μαθητή, εμφανίζονται για πρώτη φορά στη σχολική κοινότητα και συνιστούν μια πρωτόγνωρη εμπειρία.

Σκοπός του βιβλίου καθηγητή είναι να διευκολύνει το εκπαιδευτικό σου έργο, μέσα και έξω από την τάξη, παρέχοντας μια σειρά από προτάσεις σε θέματα που αφορούν στην οργάνωση του μαθήματος, στην παρουσίαση της διδακτέας ύλης, σε πηγές πληροφόρησης, σε θέματα αξιολόγησης της επίδοσης των μαθητών. Το βιβλίο αυτό σε καμιά περίπτωση δεν αποτελεί πανάκεια για τη διεξαγωγή του μαθήματος. Για να μπορέσει να φανεί χρήσιμο και αποδοτικό, απαραίτητη προϋπόθεση είναι η αυτενέργειά σου, η αξιοποίηση με το δικό σου προσωπικό τρόπο του υλικού που βρίσκεται στις σελίδες του. Θα πρέπει πάντως να σημειώσουμε, ότι το βιβλίο καθηγητή δεν αποτελεί υποχρεωτικό υλικό για την οργάνωση και διεξαγωγή του μαθήματος. Η χρησιμοποίησή του ή όχι επαφίεται σε σένα. Αποτελεί όμως σίγουρα ένα σημείο αναφοράς, που μπορείς να συμβουλευτείς για να προετοιμάσεις καλύτερα, εγκυρότερα και ευκολότερα τις παρουσιάσεις, και όχι μόνο, των μαθημάτων σου.

Στο βιβλίο αυτό θα βρεις κατ' αρχήν ένα εισαγωγικό, γενικό κεφάλαιο που αναφέρεται στους σκοπούς του μαθήματος, στη γενική και ειδική διδακτική μεθοδολογία που θα πρέπει να ακολουθήσεις, ώστε να προσεγγίσεις σωστά το μάθημα, στο προτεινόμενο από τη μεριά μας χρονοδιάγραμμα διδασκαλίας των ενοτήτων του και στις προτεινόμενες μορφές και μεθοδολογίες αξιολόγησης της επίδοσης των μαθητών.

Στη συνέχεια υπάρχουν 14 κεφάλαια, καθένα από τα οποία αναφέρεται στο αντίστοιχο κεφάλαιο του βιβλίου μαθητή και είναι δομημένα με τον ίδιο τρόπο. Κάθε κεφάλαιο περιλαμβάνει:

- το γενικό και τους ειδικούς διδακτικούς σκοπούς του κεφαλαίου

- οδηγίες και επισημάνσεις αναφορικά με σημεία της διδασκαλίας και της διδακτέας ύλης, που κατά τη γνώμη μας θα πρέπει να προσεχθούν και να τονιστούν στους μαθητές
- προγραμματισμό των μαθημάτων του κεφαλαίου με αναλυτικά σχέδια μαθήματος που περιλαμβάνουν : διδακτικούς στόχους του συγκεκριμένου μαθήματος, προτεινόμενα υλικά και εποπτικά μέσα, αντίστοιχη ύλη από το βιβλίο μαθητή που θα πρέπει να διδαχθεί, δραστηριότητες από το τετράδιο εργασιών μαθητή που θα πρέπει να αντιμετωπιστούν και έτοιμα τεστ αξιολόγησης επίδοσης μαθητή για κάθε μάθημα
- προβληματισμούς και θέματα προς συζήτηση αναφορικά με το μάθημα
- επιπρόσθετες πηγές πληροφόρησης, είτε βιβλιογραφικές, είτε μέσω Διαδικτύου
- απαντήσεις τόσο στις ερωτήσεις που περιλαμβάνονται στο τέλος κάθε κεφαλαίου του βιβλίου μαθητή, όσο και στις προτεινόμενες δραστηριότητες που υπάρχουν στο τετράδιο εργασίας.

Επιπλέον, στο τέλος ορισμένων κεφαλαίων υπάρχει ένα τμήμα της διδακτέας ύλης που καταρχήν είχε γραφτεί στο βιβλίο μαθητή, αλλά στη συνέχεια κρίθηκε αρκετά προχωρημένο και δύσκολο και μεταφέρθηκε στο βιβλίο καθηγητή, υπό την έννοια της προαιρετικής χρήσης, αν και εφόσον το επιτρέπουν οι αντικειμενικές συνθήκες της κάθε τάξης.

Στο τέλος του βιβλίου καθηγητή παρατίθεται παράρτημα σχετικό με τη διαχείριση αρχείων δεδομένων, του οποίου η ύλη δεν συμπεριλαμβάνεται στο αναλυτικό πρόγραμμα. Το παράρτημα αυτό συμπεριλήφθηκε μετά από υπόδειξη του ΠΙ.

Ακόμα, δεν θα πρέπει να παραλείψουμε να αναφέρουμε ότι το βιβλίο καθηγητή συνοδεύεται από ένα αποθηκευτικό μέσο, το οποίο περιλαμβάνει όλα τα απαραίτητα αρχεία εργασίας για τις προτεινόμενες δραστηριότητες των μαθητών, όσο και τον κώδικα των δραστηριοτήτων/προγραμμάτων που προτείνονται, εκφρασμένο σε διάφορες γλώσσες προγραμματισμού.

Ολοκληρώνοντας το σύντομο αυτό εισαγωγικό σημείωμα, θα θέλαμε να τονίσουμε και πάλι τη βαρύτητα της δικής σου ενεργής συμμετοχής στην λειτουργικότητα και αποδοτικότητα αυτού του βιβλίου, τόσο με την ενεργοποίηση και αυτενέργειά σου, όσο και με τις υποδείξεις και παρατηρήσεις σου σχετικά με οτιδήπο-

τε κατά τη γνώμη σου θα μπορούσε να το καταστήσει σε μεγαλύτερο βαθμό υποβοηθητικό στο εκπαιδευτικό σου έργο.

Τέλος θα πρέπει να σημειωθεί ότι, παρ' όλο που η συγγραφική ομάδα διαθέτει μια πολύ μεγάλη συγγραφική εμπειρία, αθροιστικά πάνω από 50 βιβλία, σε καμία περίπτωση δεν τεκμηριώνει το αλάθητο. Με δεδομένα ότι το μάθημα αυτό γίνεται για πρώτη φορά, ότι για πρώτη επίσης φορά γράφεται τριάδα βιβλίων για το ίδιο μάθημα και ακόμη ότι υπήρχε ημερομηνία λήξης στην παράδοση, είναι πολύ πιθανό να υπάρχουν κάποια λάθη ή παραλείψεις, ελπίζουμε όχι σημαντικά. Για το λόγο αυτό θεωρούμε ότι η συμβολή σου με σχόλια και παρατηρήσεις κατά τη διάρκεια της εφαρμογής του βιβλίου στην εκπαιδευτική διαδικασία, θα είναι καταλυτική για τη βελτίωση του περιεχομένου του για τις επόμενες σχολικές χρονιές. Κατά τη διάρκεια της πρώτης σχολικής χρονιάς ήταν φυσικό και αναμενόμενο να διαπιστωθούν μια σειρά από λάθη (κυρίως τυπογραφικά, κάποια εκφραστικά και πολύ λίγα επιστημονικά). Τα πολλά μηνύματα που στάλθηκαν από καθηγητές Πληροφορικής στο Παιδαγωγικό Ινστιτούτο, υποδεικνύοντας οποιοδήποτε είδος λάθους, βοήθησαν πολύ τη συγγραφική ομάδα στο να διορθώσει τα λάθη αυτά και να προχωρήσει μαζί με το ΠΙ στη δεύτερη έκδοση του βιβλίου.

Θα θέλαμε να ευχαριστήσουμε όλους εκείνους τους συναδέλφους - και ιδιαίτερα τους ***Γιάννη Οικονόμου*** και ***Ιωάννη Φαμέλη*** - που με τα μηνύματά τους ανατροφοδότησαν τη διαδικασία έλεγχου και διόρθωσης του βιβλίου, ευελπιστώντας ότι και αυτή τη φορά όλοι θα εντρυφήσουν στο περιεχόμενό του με την ίδια διάθεση και ζήλο. Η αποστολή μηνυμάτων από τη μεριά των καθηγητών θα θέλαμε να συνεχιστεί και στο μέλλον, πιστεύοντας ότι υπάρχουν πάντοτε περιθώρια περαιτέρω βελτίωσης.

Τέλος, διορθώνοντας παράλειψή μας από την πρώτη έκδοση του βιβλίου, θα θέλαμε να ευχαριστήσουμε το συνάδελφο ***Νίκο Κασιμάτη*** για τη συνεισφορά του στην παιδαγωγική προσέγγιση του περιεχομένου του βιβλίου.

Οι συγγραφείς

Γενικός σκοπός μαθήματος

Ο γενικός σκοπός του μαθήματος *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον* της Γ΄ τάξης του κύκλου Πληροφορικής και Υπηρεσιών της Τεχνολογικής Κατεύθυνσης του Ενιαίου Λυκείου, δεν είναι αυτός στον οποίο πιθανά θα κατέληγε κάποιος διαβάζοντας απλά και μόνο τον τίτλο του μαθήματος. Το μάθημα έχει σαν πρωταρχικό στόχο την ανάπτυξη δεξιοτήτων και ικανοτήτων σχετικών με την αλγοριθμική και την ορθολογική χρήση τους στην καθημερινή ζωή. Πολλές βασικές έννοιες αλγοριθμικής, αλλά και προγραμματισμού (π.χ. συνθήκες ελέγχου, λογικές προτάσεις και συμπεράσματα, κ.α.), συνιστούν αναπόσπαστο τμήμα των γενικών γνώσεων και δεξιοτήτων που πρέπει να αποκτήσει ο μαθητής, οι οποίες δεξιότητες και γνώσεις - στην πλειονότητά τους - δεν προσεγγίζονται από άλλα μαθήματα.

Το μάθημα *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον* δεν έχει σαν στόχο τη διδαχή και την εκμάθηση κάποιου συγκεκριμένου προγραμματιστικού περιβάλλοντος, ούτε την καλλιέργεια προγραμματιστικών δεξιοτήτων από τη μεριά των μαθητών. Δεν αποσκοπεί στη λεπτομερειακή εξέταση της δομής, του ρεπερτορίου και των συντακτικών κανόνων κάποιας γλώσσας προγραμματισμού. Δεν προτίθεται να επιχειρήσει να δημιουργήσει προγραμματιστές. Το μάθημα δεν αφορά την εκμάθηση εξεζητημένων τεχνικών προγραμματισμού, αλλά ως εργαλείο δόμησης της σκέψης πρέπει να εστιάζει στις προσεγγίσεις και στις τεχνικές επίλυσης προβλημάτων.

Το μάθημα *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, έχει σαν γενικό σκοπό οι μαθητές :

- να καλλιεργήσουν αναλυτική σκέψη και συνθετική ικανότητα,
- να αναπτύξουν τη δημιουργικότητα, τη φαντασία στο σχεδιασμό,
- να καλλιεργήσουν και να εθιστούν στην αυστηρότητα και σαφήνεια της έκφρασης και της διατύπωσης,
- να αναπτύξουν ικανότητες μεθοδολογικού χαρακτήρα,
- να αποκτήσουν δεξιότητες αλγοριθμικής προσέγγισης,
- να καταστούν ικανοί να υλοποιούν τις λύσεις απλών προβλημάτων με χρήση βασικών προγραμματιστικών γνώσεων.

Ειδικοί σκοποί μαθήματος

Έμφαση δίνεται στις ενότητες του μαθήματος που πραγματεύονται θέματα ανάλυσης προβλήματος και σχεδίασης αλγορίθμου. Μέσα από τις δύο αυτές θεματικές ενότητες πηγάζουν οι περισσότεροι από τους ειδικούς σκοπούς του μαθήματος. Συγκεκριμένα:

Οι μαθητές θα πρέπει:

- να κατανοούν πλήρως τα προβλήματα που τους τίθενται,
- να μπορούν να ανιχνεύουν και να διακρίνουν τα μέρη ενός προβλήματος,
- να μπορούν να αναλύουν τις εργασίες που πρέπει να γίνουν για την επίλυση ενός προβλήματος,

- να μπορούν να προσδιορίζουν και να αναφέρουν με ακρίβεια και σαφήνεια τα δεδομένα και τα ζητούμενα ενός προβλήματος,
- να είναι σε θέση να διακρίνουν ανάμεσα στη σημασία των λέξεων σε μια φυσική γλώσσα και σε μια γλώσσα προγραμματισμού,
- να μπορούν να διακρίνουν ποια προβλήματα/καταστάσεις μπορούν ή πρέπει να αντιμετωπίζονται μέσα από προγραμματιστικό περιβάλλον,
- να κατανοήσουν ότι η έννοια του αλγορίθμου δεν συνιστά αποκλειστικό χαρακτηριστικό της επιστήμης των υπολογιστών και ότι υπάρχουν αλγόριθμοι που περιγράφουν κάθε είδος καθημερινής διεργασίας,
- να συνειδητοποιήσουν τη θεμελιώδη σπουδαιότητα των αλγορίθμων, απαραίτητη προϋπόθεση για να υπάρξει πρόγραμμα και να εκτελεστεί μια εργασία με υπολογιστή,
- να κατανοήσουν ότι ο αλγόριθμος συνιστά την ενοποιό έννοια για όλες τις δραστηριότητες της επιστήμης των υπολογιστών,
- να συνάγουν ότι ο αλγόριθμος είναι ανεξάρτητος της γλώσσας προγραμματισμού στην οποία εκφράζεται και του υπολογιστικού συστήματος που τον εκτελεί,
- να είναι ικανοί να εφαρμόζουν τους κανόνες σχεδίασης αλγορίθμων,
- να είναι ικανοί να γράφουν έναν αλγόριθμο για ένα πρόβλημα που τους δίδεται,
- να γνωρίσουν και να επιλέγουν την κατάλληλη δομή για την επίλυση ενός προβλήματος,
- να μπορούν να αποφαίνονται για την ορθότητα ενός αλγορίθμου,
- να αναπτύξουν δεξιότητες άτυπης σύγκρισης αλγορίθμων.

Στις υπόλοιπες θεματικές ενότητες το μάθημα πραγματεύεται υλοποίηση των αλγορίθμων σε προγραμματιστικό περιβάλλον. Για τις δύο αυτές ενότητες οι ειδικοί σκοποί ορίζονται ως εξής.

Οι μαθητές θα πρέπει:

- να γνωρίσουν τα κύρια είδη προγραμματισμού και να μπορούν να αναφέρουν τα βασικά χαρακτηριστικά τους,

- να μπορούν να μετατρέπουν έναν αλγόριθμο επίλυσης ενός προβλήματος σε απλό πρόγραμμα με τη χρήση δομημένης γλώσσας προγραμματισμού,
- να μπορούν να ολοκληρώνουν απλές εφαρμογές με τη χρήση αντικειμενοστραφούς, οδηγούμενης από γεγονότα γλώσσας προγραμματισμού,
- να αναπτύξουν δεξιότητες βασικού σχεδιασμού και υλοποίησης περιβάλλοντος διεπαφής,
- να προχωρούν σε εκσφαλμάτωση και βελτιστοποίηση του προγράμματος τους,
- να αποκτήσουν κριτήρια αντικειμενικής αξιολόγησης του προγράμματός τους, αλλά και των προγραμμάτων των συμμαθητών τους,
- να αναπτύξουν ικανότητες αναζήτησης εναλλακτικών λύσεων,
- να κατανοήσουν τη σημασία και την αναγκαιότητα της τεκμηρίωσης και να είναι σε θέση να τεκμηριώνουν τα προγράμματά τους.

Γενική διδακτική μεθοδολογία μαθήματος

Η γενική μεθοδολογία διδασκαλίας του μαθήματος θα πρέπει να προωθεί, να ενισχύει και να ενθαρρύνει:

- τη δημιουργική δράση του μαθητή μέσω της ενεργοποίησής του,
- τη συνεργατική μάθηση σε ομαδικό περιβάλλον,
- την ανάπτυξη δεξιοτήτων μεθοδολογικού χαρακτήρα,
- την ανάπτυξη κριτικής σκέψης μέσω της συζήτησης και του προβληματισμού,
- την αποτύπωση και κατάθεση ελεύθερης σκέψης και έκφρασης.

Στο εργαστηριακό μέρος του μαθήματος, χρησιμοποιώντας υπολογιστικά εργαλεία και τεχνικές, οι μαθητές εμπλέκονται σε ποικίλες δραστηριότητες οι οποίες:

- διευκολύνουν την ανάπτυξη της δημιουργικής ικανότητας,
- υποστηρίζουν το συμμετοχικό και συνεργατικό χαρακτήρα της μάθησης,

- καλλιεργούν την ανάπτυξη δεξιοτήτων μοντελοποίησης και επίλυσης προβλημάτων,
- παρέχουν ευχέρεια στη χρήση συμβολικών μέσων έκφρασης και διερεύνησης,
- ευνοούν την αναλυτική και τη συνθετική σκέψη.

Έμφαση θα πρέπει να δίνεται στην ανάλυση και στο σχεδιασμό της λύσης των προβλημάτων και όχι σε αυτήν την ίδια τη λύση. Η βάση των εργασιών και των δραστηριοτήτων, πρέπει να είναι η ανάλυση και ο σχεδιασμός, που κατά κανόνα θα γίνονται στην αίθουσα με την καθοδήγηση και υποβοήθηση του διδάσκοντα.

Οι δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές θα πρέπει κατά την ώρα της ανάθεσης να επεξηγούνται στην τάξη. Οι μαθητές θα μπορούν να διατυπώνουν τις απορίες τους τόσο σε σχέση με τη διατύπωση της δραστηριότητας, όσο και σε σχέση με το περιεχόμενό της και τα ζητούμενα. Ο διδάσκων θα πρέπει να παρέχει τις απαραίτητες διευκρινήσεις.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους αναθέτονται για το σπίτι, αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους. Θα πρέπει να αφιερώνεται χρόνος από τη διδακτική ώρα έτσι ώστε να παρουσιάζονται οι περισσότερες δυνατές απαντήσεις των μαθητών και να σχολιάζονται και να κρίνονται από τους συμμαθητές τους και τον διδάσκοντα, δίνοντας έτσι η δυνατότητα ανάπτυξης διαλόγου.

Η πρακτική αυτή έχει σαν αποτέλεσμα:

- να ενισχύει την ανάπτυξη της κριτικής ικανότητας και της επιχειρηματολογίας των μαθητών,
- να καλλιεργεί την ιδέα του εποικοδομητικού και γόνιμου διαλόγου,
- να τροφοδοτεί με νέες σκέψεις και απόψεις που τυχόν δεν έχουν επισημανθεί,
- να λειτουργεί σαν κίνητρο για τους μαθητές που βλέπουν την προσπάθειά τους να αποτελεί θέμα συζήτησης και κέντρο ενδιαφέροντος για όλους, ακόμα και στην περίπτωση όπου κριτικάρεται,
- να αναπτύσσει τον υγιώς εννοούμενο ανταγωνισμό και την ευγενή άμιλλα,
- να μπορεί να περάσει στο μαθητή της αίσθηση της επιβράβευσης της προσπάθειας του και τον παρακινεί προς την κατεύθυνση της μεγιστοποίησής της.

Θα πρέπει πάντοτε να δίνεται μεγάλη προσοχή στον τρόπο που γίνεται η κριτική στις απαντήσεις των μαθητών, τόσο από τον διδάσκοντα, όσο και από τους συμμαθητές του.

Ο διδάσκων θα πρέπει:

- να παρακινήσει τους μαθητές να ασκούν καλόπιστη και καλοπροαίρετη κριτική,
- να εμφυσήσει στους μαθητές το μήνυμα της συνεργατικότητας,
- να αποτρέψει συνειδητές ανταγωνιστικές διαθέσεις και δημιουργία αισθήματος αντιπαλότητας μεταξύ τους,
- να καλλιεργήσει το πνεύμα ομαδικότητας που αποτελεί πρωταρχικής σημασίας παιδαγωγικό στόχο.

Στην αίθουσα θα γίνεται επίσης, η εισαγωγή και η ανάπτυξη των διαφόρων εννοιών, παράλληλα με την πρακτική άσκηση στο εργαστήριο. Κατ' αυτήν την έννοια θεωρία και πρακτική θα πρέπει να βαδίζουν και να εξελίσσονται παράλληλα (βλέπε σχετικό πίνακα στο τέλος του κεφαλαίου *Ειδική διδακτική μεθοδολογία μαθήματος*).

Οι μαθητές θα πρέπει να εισάγονται στις καινούριες έννοιες μέσω αναφορών στις προηγούμενες γνώσεις και εμπειρίες τους. Είναι ιδιαίτερα σημαντικό να κατέχουν οι μαθητές εμπειρική γνώση πριν την εισαγωγή νέων εννοιών και θεματολογίας. Με τη χρήση κατάλληλων παραδειγμάτων αναδεικνύεται η αναγκαιότητα της εισαγωγής των εννοιών που πρόκειται να διδαχθούν και έτσι ο μαθητής μπορεί και εντάσσει σταδιακά τις καινούργιες έννοιες στις προϋπάρχουσες γνώσεις και εμπειρίες του.

Ειδική διδακτική μεθοδολογία μαθήματος

Σαν μεθοδολογία διδασκαλίας του μαθήματος προτείνεται η ***σπειροειδής προσέγγιση*** των ενοτήτων του. Μέσα από το πρίσμα της σπειροειδούς προσέγγισης του μαθήματος οι έννοιες προσεγγίζονται επαναληπτικά με διαφορετικό τρόπο και/ή βάθος ανά περίπτωση. Με βάση κάθε φορά προκαθορισμένο προς επίλυση πρόβλημα, ο διδάσκων επανέρχεται σε θεματολογία και έννοιες που έχουν ήδη εν μέρει αναπτυχθεί. Δεν αποκλείονται οι επαναλήψεις περιεχομένων, με έμφαση κατά την πρώτη φορά παρουσίασης.

Επιπρόσθετο λόγο για αυτού του είδους την προσέγγιση, αποτελεί η ανάγκη επανάληψης θεμάτων και εννοιών (π.χ. δομές ελέγχου, κ.α.), λόγω διττής αναφοράς τους, καταρχήν σε επίπεδο αλγοριθμικής προσέγγισης και εν συνεχεία σε επίπεδο προγραμματιστικού περιβάλλοντος. Θα πρέπει να επισημανθεί πως δεν ενδείκνυται η σειριακή διδασκαλία των ενοτήτων του αναλυτικού προγράμματος του μαθήματος.

Στον πίνακα που ακολουθεί παρατίθεται προτεινόμενο πρόγραμμα διδασκαλίας των κεφαλαίων του βιβλίου. Εκτιμάται ότι η διδασκαλία των ενοτήτων που παραθέτονται στις δύο στήλες του πίνακα θα πρέπει να εκτυλίσσονται παράλληλα. Για τη σύνταξη των περιεχομένων του πίνακα ελήφθησαν υπόψη:

- η σωστή επιστημολογικά προσέγγιση των θεματικών ενοτήτων,
- η τήρηση στο μεγαλύτερο δυνατό βαθμό της σειριακής προσπέλασης των περιεχομένων του προγράμματος σπουδών,
- η μικτή (θεωρητική και εργαστηριακή) φύση του μαθήματος.

Τάξη	Εργαστήριο
Ανάλυση προβλήματος	
	Εισαγωγή στον προγραμματισμό
Βασικές έννοιες αλγορίθμων (μέχρι και δομή ακολουθίας)	Βασικές έννοιες προγραμματισμού
Βασικές έννοιες αλγορίθμων (δομή επιλογής, πολλαπλών επιλογών)	Επιλογή και επανάληψη (εντολές επιλογής)
Βασικές έννοιες αλγορίθμων (δομή επανάληψης)	Επιλογή και επανάληψη (εντολές επανάληψης)
Δομές δεδομένων και αλγόριθμοι	Πίνακες
Τεχνικές σχεδίασης αλγορίθμων	Υποπρογράμματα
	Σύγχρονα προγραμματιστικά περιβάλλοντα
Ανάλυση αλγορίθμων	Σχεδίαση διεπαφής χρήστη
	Εκσφαλμάτωση προγράμματος
	Αξιολόγηση-Τεκμηρίωση

Στη συνέχεια του βιβλίου παραθέτονται οδηγίες σχετικές με διδασκαλία καθενός από τα κεφάλαια του βιβλίου. Θα πρέπει να σημειωθεί πως για τη διαμόρφωση των οδηγιών αυτών ελήφθησαν υπόψη οι προτεινόμενες συνολικά ώρες διδασκαλίας του μαθήματος που αναφέρονται στο αναλυτικό πρόγραμμα σπουδών.

Συγκεκριμένα, προβλέπεται η ολοκλήρωση της διδασκαλίας του μαθήματος σε 58 διδακτικές ώρες (29 εβδομάδες). Μεταξύ των τεσσάρων ενοτήτων του μαθήματος πραγματοποιήθηκαν μικρές, απαραίτητες κατά την κρίση των συγγραφέων, μετακινήσεις διδακτικών ωρών.

Τρόποι αξιολόγησης και κριτήρια μέτρησης επίδοσης μαθητή

Για να μπορέσει η αξιολόγηση να επιτύχει το σκοπό της, θα πρέπει καταρχήν να οριστεί με ακρίβεια τι είναι αυτό που θα μετρηθεί κατά την εξέταση, με ποιο τρόπο θα μετρηθεί και πως θα αξιολογηθεί.

Αυτό που μπορεί να μετρηθεί ως ένα βαθμό, είναι οι γνώσεις, οι ικανότητες, και οι δεξιότητες του μαθητή. Προηγούμενα θα πρέπει οι σκοποί της διδασκαλίας να έχουν διατυπωθεί αναλυτικά, με σαφήνεια και ακρίβεια.

Τα διαθέσιμα μέσα για τη μέτρηση της επίδοσης του μαθητή είναι οι παραδοσιακοί τρόποι ελεύθερων απαντήσεων του εξεταζόμενου καθώς και τα τεστ επίδοσης κλειστού τύπου. Τα μέσα αυτά πρέπει να εκπληρώνουν ορισμένες προϋποθέσεις, ώστε η μέτρηση να είναι, όσο το δυνατόν, αντικειμενική, αξιόπιστη και έγκυρη.

Οι *παραδοσιακοί τρόποι αξιολόγησης* δέχονται αυστηρή κριτική κυρίως γιατί:

- δεν εξασφαλίζουν αντικειμενική βαθμολογία,
- δεν είναι αξιόπιστοι, δηλαδή η ίδια επίδοση βαθμολογείται από τους ίδιους κριτές με διαφορετικό βαθμό σε διαφορετικά χρονικά διαστήματα,
- δεν παράγουν έγκυρη βαθμολογία, δηλαδή δε μετρούν πραγματικά αυτό που υποτίθεται ότι μετρούν,
- δεν πληροφορούν το διδάσκοντα για την ποιότητα της διδασκαλίας του,
- δεν παρέχουν τη δυνατότητα στο μαθητή να ελέγχει την εξέλιξη της επίδοσής του,
- καταπονούν τους μαθητές,
- αφαιρούν πολύτιμο χρόνο από τη διδασκαλία.

Αντίθετα τα *τεστ επίδοσης* βρίσκουν όλο και πιο πολλούς υποστηρικτές αφού θεωρείται ότι μπορούν να εξακριβώσουν την πνευματική ικανότητα του ατόμου. Το τεστ επίδοσης είναι ένα ψυχολογικό πείραμα με το οποίο μετριέται η επίδοση του μαθητή σ' ένα αντικείμενο ή περιοχή γνώσης που έχει διδαχτεί.

Τα τεστ επίδοσης για να είναι επιστημονικά έγκυρα, πρέπει να εκπληρώνουν τα εξής κριτήρια :

- Αντικειμενικότητα: η μέτρηση και η αξιολόγηση της επίδοσης του εξεταζόμενου είναι ανεξάρτητες από την υποκειμενική κρίση του εξεταστή, δηλαδή διαφορετικοί βαθμολογητές αξιολογούν με τον ίδιο βαθμό το αποτέλεσμα του τεστ.
- Αξιοπιστία: μια επανεξέταση, και υπό τους αυτούς όρους των ίδιων ατόμων, δίνει το ίδιο αποτέλεσμα.
- Εγκυρότητα: το τεστ μετράει πραγματικά αυτό που υποτίθεται ότι μετράει και όχι κάτι άλλο.

Τρία είναι τα είδη των τεστ επίδοσης : σταθμισμένα, άτυπα και κριτηρίων.

- Σταθμισμένο είναι εκείνο το τεστ που έχει δοκιμαστεί σε μια ομάδα μαθητών η οποία αποτελεί το δείγμα ενός ευρύτερου μαθητικού πληθυσμού. Η στάθμιση των τεστ απαιτεί τη χρήση πολύπλοκων στατιστικών μεθόδων.
- Άτυπα τεστ είναι εκείνα τα οποία καθορίζονται από τους σκοπούς της διδασκαλίας που ορίζει το αναλυτικό πρόγραμμα μιας τάξης. Η σύνταξη και η εφαρμογή των άτυπων τεστ επίδοσης στηρίζεται στην αρχή ότι, εφόσον οι σκοποί της διδασκαλίας είναι αναλυτικά διατυπωμένοι στα αναλυτικά προγράμματα, τα τεστ αυτά μπορούν εύκολα να κατασκευαστούν και να χρησιμοποιηθούν αποτελεσματικά για την αξιολόγηση της διδασκαλίας και τη διάγνωση των δυσκολιών που συναντούν οι μαθητές. Πράγματι, τα άτυπα τεστ επίδοσης έχουν το πλεονέκτημα ότι προσαρμόζονται στους σκοπούς και τις ανάγκες της διδασκαλίας, όπως οι σκοποί αυτοί και οι ανάγκες διαφοροποιούνται από τάξη σε τάξη και από σχολείο σε σχολείο και είναι στα χέρια του δασκάλου ένα εύχρηστο και αποτελεσματικό μέσο για τη διάγνωση των δυσκολιών μάθησης, για την εξακρίβωση της διαφοροποίησης που παρουσιάζει η τάξη και για την αξιολόγηση, τυπική ή μη, των μαθητών. Επίσης, αφού η εκτίμηση του αποτελέσματος των άτυπων τεστ γίνεται με κριτήριο τους σκοπούς της διδασκαλίας, μπορεί τα τεστ αυτά εφόσον οι σκοποί της διδασκαλίας είναι οι

σκοποί που ορίζει το πρόγραμμα, να αποτελέσουν τη βάση για την αξιολόγηση του αναλυτικού προγράμματος.

- Τεστ κριτηρίων. Τα τεστ κριτηρίων καθορίζονται με βάση τους σκοπούς της διδασκαλίας, οι οποίοι όμως είναι διατυπωμένοι με τη μορφή εξειδικευμένων στόχων μάθησης στο αναλυτικό πρόγραμμα. Η σύνταξη, δηλαδή, των τεστ αυτών προϋποθέτει αναλυτικό πρόγραμμα οργανωμένο με βάση τους στόχους μάθησης. Στα τεστ κριτηρίων η μέτρηση της επίδοσης των μαθητών αποτελεί κριτήριο για την εκτίμηση κατά πόσο πέτυχε ή όχι η διδασκαλία. Πρωταρχικός σκοπός των τεστ κριτηρίων είναι συνεπώς η αξιολόγηση της διδασκαλίας, ώστε με βάση τα αποτελέσματα της αξιολόγησης αυτής να λαμβάνονται τα αναγκαία μέτρα για τη βελτίωση της.

Τα τεστ επίδοσης, ανάλογα με τη μορφή που έχουν τα ερωτήματα, διακρίνονται σε:

- Τεστ συμπλήρωσης. Στο τεστ συμπλήρωσης παραλείπονται σκόπιμα ορισμένα στοιχεία των ερωτήσεων ή του κειμένου, τα οποία πρέπει να συμπληρώσει ο μαθητής.

- Τεστ σωστού-λάθους. Το τεστ αυτό αποτελείται από μία πρόταση, την οποία πρέπει να αναγνωρίσει ο μαθητής ως ορθή ή εσφαλμένη. Μειονεκτεί όμως γιατί το ποσοστό των απαντήσεων, που είναι δυνατόν να απαντηθούν τυχαία από το μαθητή, μπορεί να είναι σημαντικό.

- Τεστ σύζευξης. Το τεστ σύζευξης αποτελείται από πολλές ερωτήσεις, ώστε να δίνεται η δυνατότητα πολλαπλής εκλογής. Με το τεστ αυτό ελέγχεται η ικανότητα του μαθητή να αναγνωρίζει και να συνδυάζει τις γνώσεις που διδάχτηκε. Οι πιθανότητες να απαντηθεί σωστά, αλλά τυχαία από το μαθητή είναι λίγες.

- Τεστ πολλαπλής επιλογής. Το τεστ επιλογής αποτελείται από περισσότερες μερικότερες απαντήσεις για ένα θέμα ή πρόβλημα. Παρέχονται δύο τύποι:

  - της μοναδικής απάντησης, όπου μόνο μια από τις προτεινόμενες απαντήσεις είναι η ορθή,

  - της συνδυασμένης απάντησης, όπου η ολοκληρωμένη σωστή απάντηση της ερώτησης μορφοποιείται από τη σύνδεση πολλών από τις προτεινόμενες μερικές απαντήσεις.

Το τεστ πολλαπλής επιλογής επιτρέπει περισσότερο την αντικειμενική εκτίμηση της αξιολόγησης του μαθητή. Ελέγχει την ικανότητα του μαθητή να ερμηνεύει, να διακρίνει, να εφαρμόζει γνώσεις και να καταλήγει με βάση συγκεκριμένα δεδομένα σε λογικά συμπεράσματα. Επίσης, ο παράγοντας τύχη υπεισέρχεται λίγο κατά την εφαρμογή του. Παρ' όλα αυτά και το τεστ αυτό πολύ λίγο εξετάζει της δημιουργικές ικανότητες του παιδιού, αφού αποκλείει την προσωπική του έκφραση και δημιουργικότητα.

Σε αυτό το βιβλίο παραθέτονται και προτείνονται έτοιμα τεστ αξιολόγησης της επίδοσης των μαθητών. Για κάθε μάθημα κάθε κεφαλαίου, συμπεριλαμβάνεται και από ένα τεστ επίδοσης. Η δημιουργία τους έγινε με βάση τους διατυπωμένους από το πρόγραμμα σπουδών γενικούς και ειδικούς σκοπούς. Οι απαντήσεις των τεστ αξιολόγησης περιλαμβάνονται επίσης στο βιβλίο αυτό.

Αποτελεί πεποίθηση των συγγραφέων ότι η αξιολόγηση της επίδοσης των μαθητών δεν μπορεί να ολοκληρωθεί αποκλειστικά και μόνο από τα προτεινόμενα τεστ, δεδομένου ότι τα τεστ ελέγχου κλειστού τύπου δεν παρέχουν στο μαθητή τη δυνατότητα κατάθεσης και έκφρασης της δημιουργικής διάθεσής του, της αναλυτικής του σκέψης και συνθετικής ικανότητας, στοιχεία τα οποία αποτελούν βασικό διδακτικό σκοπό του μαθήματος. Κατά συνέπεια η αξιολόγηση του μαθητή για να είναι ολοκληρωμένη θα πρέπει να γίνεται και σε διαρκή βάση από τη μεριά του διδάσκοντα μέσα από την παρατήρηση και επισκόπηση της καθημερινής πρακτικής και δράσης του μαθητή στο σχολικό περιβάλλον.

Κεφάλαιο 1

Ανάλυση Προβλήματος

1.1. Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου αυτού είναι οι μαθητές να αναπτύξουν αναλυτικές και συνθετικές ικανότητες στην αντιμετώπιση προβλημάτων και να διατυπώνουν με σαφήνεια τις σκέψεις τους.

1.2. Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να κατανοούν πλήρως τα προβλήματα που τους τίθενται,
- να προσδιορίζουν τα συστατικά μέρη ενός προβλήματος,
- να αναλύουν ένα πρόβλημα σε άλλα απλούστερα,
- να προσδιορίζουν τα δεδομένα που παρέχονται για την αντιμετώπιση του προβλήματος,
- να προσδιορίζουν τα ζητούμενα αποτελέσματα και τη μορφή απόδοσής τους,
- να θέτουν οι ίδιοι προβλήματα διατυπώνοντάς τα με ακρίβεια και πληρότητα.

1.3. Οδηγίες – επισημάνσεις

Επισημαίνουμε ότι η προσοχή του διδάσκοντα θα πρέπει να επικεντρωθεί και στα παρακάτω σημεία:

1. Να γίνει σαφές στους μαθητές ότι τα προβλήματα δεν απαιτούν για τη λύση τους την παρουσία υπολογιστών. Οι υπολογιστές έκαναν την εμφάνισή τους πολύ αργότερα από την εμφάνιση των προβλημάτων. Ο άνθρωπος αντιμετώπιζε και συνεχίζει καθημερινά να αντιμετωπίζει πλήθος από προβλήματα χρησιμοποιώντας το μυαλό του.

2. Θα πρέπει να γίνει συνείδηση στους μαθητές ότι με τον όρο πρόβλημα δεν αναφερόμαστε σε μαθηματικού τύπου προβλήματα, τα οποία αποτελούν ένα μόνο είδος προβλημάτων. Προβλήματα υπάρχουν και θέτονται σε όλες τις επιστήμες και τους επιστημονικούς κλάδους, αλλά και σε κοινωνικό, πολιτισμικό, εκπαιδευτικό και σε κάθε άλλο επίπεδο της ανθρώπινης δραστηριότητας, καθώς επίσης και σε καταστάσεις της καθημερινής ζωής. Η προσέγγιση της έννοιας του προβλήματος γίνεται μέσα από μια γενική και πλατιά θεώρηση, χωρίς μαθηματικοκεντρική προδιάθεση.

3. Ένα "γνωσιακό ποσό" δεν μπορεί να χαρακτηριστεί αυτόματα σαν δεδομένο ή πληροφορία. Το στοιχείο εκείνο που θα μπορέσει να προσδώσει το χαρα-

κτηρισμό δεδομένο ή πληροφορία σε ένα "γνωσιακό ποσό" είναι το αν αυτό προέρχεται από διαδικασία επεξεργασίας ή αν πρόκειται να υποβληθεί σε επεξεργασία. Κατά συνέπεια το ίδιο "γνωσιακό πόσο" μπορεί για μία διαδικασία να αποτελεί πληροφορία και για μια άλλη επόμενη να αποτελεί δεδομένο.

Σχετική με την παρατήρηση αυτή είναι η ερώτηση α) της δραστηριότητας ΔΤ1.

1.4. Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του 1ου μαθήματος είναι οι μαθητές να μπορούν:

- να κατανοούν πλήρως τα προβλήματα που τους τίθενται,
- να προσδιορίζουν τα συστατικά μέρη ενός προβλήματος,
- να αναλύουν ένα πρόβλημα σε άλλα απλούστερα

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

πίνακας, διαφάνειες

Περιεχόμενα θεωρητικής παρουσίασης (παράγραφοι από το βιβλίο του μαθητή)

- Η έννοια πρόβλημα
- Κατανόηση προβλήματος (σαφήνεια διατύπωσης, δεδομένα, πληροφορίες, επεξεργασία δεδομένων)
- Δομή προβλήματος

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να παρουσιαστεί και να αναλυθεί, από το βιβλίο του μαθητή, το παράδειγμα που αναφέρεται στη σαφήνεια διατύπωσης (παράγραφος 1.2.1).

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη, με προτεινόμενη σειρά :

- η δραστηριότητα ΔΤ5
- μία από τις δραστηριότητες ΔΤ1 και ΔΤ2.

Να δοθούν στους μαθητές για αντιμετώπιση στο σπίτι :

- οι δραστηριότητες ΔΣ1 και ΔΣ2.

Η δραστηριότητα ΔΣ1 είναι αρκετά σύνθετη. Θα πρέπει να δοθούν κατευθύνσεις στους μαθητές (βλέπε παράγραφο παρόντος *απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή*).

Για τις δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές να δίνονται τυχόν διευκρινήσεις που απαιτούνται από την πλευρά των μαθητών.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους ανατίθενται στο σπίτι αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους.

Τεστ αξιολόγησης επίδοσης

Α. Συμπληρώστε με σωστό ή λάθος

1. Πρόβλημα είναι μια μαθηματική κατάσταση που πρέπει να αντιμετωπίσουμε.
2. Τα δεδομένα υποβαλλόμενα σε επεξεργασία παρέχουν πληροφορίες.
3. Ο υπολογιστής και το πρόβλημα είναι έννοιες που εξαρτώνται άμεσα η μια από την άλλη.
4. Ένα οποιοδήποτε πρόβλημα μπορεί να αναπαρασταθεί είτε διαγραμματικά, είτε φραστικά, είτε αλγεβρικά.
5. Ένα πρόβλημα μπορεί να αναλυθεί σε πολλά επιμέρους προβλήματα

Β. Συμπληρώστε τα κενά με τη λέξη που λείπει

1. Η ..... είναι η βάση της επίλυσης ενός προβλήματος.
2. Σημαντικός παράγοντας στην κατανόηση ενός προβλήματος είναι η .....του.
3. Τα συστατικά μέρη που αποτελούν ένα πρόβλημα προσδιορίζουν τη ..... του.
4. Τα δεδομένα μπορούν να παρέχουν πληροφορίες όταν υποβάλλονται σε .....

Απαντήσεις τεστ αξιολόγησης επίδοσης

Α1 : λάθος	Α2 : σωστό	Α3 : λάθος	Α4 : σωστό	Α5 : σωστό
Β1: κατανόηση	Β2: διατύπωσή	Β3: δομή	Β4: επεξεργασία	

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του 2ου μαθήματος είναι οι μαθητές να μπορούν :

- προσδιορίζουν τα δεδομένα που τους παρέχονται για την αντιμετώπιση του προβλήματος
- να προσδιορίζουν τα ζητούμενα αποτελέσματα και τη μορφή απόδοσής τους

- να θέτουν οι ίδιοι προβλήματα διατυπώνοντάς τα με ακρίβεια και πληρότητα
- να καταγοριοποιούν τα διάφορα προβλήματα

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

πίνακας, διαφάνειες

Περιεχόμενα θεωρητικής παρουσίασης (παράγραφοι από το βιβλίο του μαθητή)

- Καθορισμός απαιτήσεων
- Κατηγορίες προβλημάτων
- Πρόβλημα και υπολογιστής

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να παρουσιαστεί και να αναλυθεί, από το βιβλίο του μαθητή, το παράδειγμα που αναφέρεται στον καθορισμό απαιτήσεων (παράγραφος 1.4.).

Παραδείγματα από το τετράδιο του μαθητή

Να παρουσιαστεί και να αναλυθεί, από το τετράδιο του μαθητή το παράδειγμα 1, το οποίο αποτελεί επέκταση του παραδείγματος που αφορά τον καθορισμό απαιτήσεων και που αναφέρεται στο βιβλίο του μαθητή (παράγραφος 1.4.).

Να δοθεί προσοχή στο γεγονός ότι ο έλεγχος των δεδομένων μπορεί να οδηγήσει και πάλι στην είσοδο.

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη, με προτεινόμενη σειρά :

- η δραστηριότητα ΔΤ3
- η δραστηριότητα ΔΤ4 ή η ΔΣ5 (αν υπάρχει διαθέσιμος χρόνος).

Η δραστηριότητα ΔΣ5 μπορεί να υλοποιηθεί εν μέρει στην τάξη με τη συμβολή του διδάσκοντα σε επίπεδο αναζήτησης και διάθεσης στοιχείων.

Να δοθούν στους μαθητές για αντιμετώπιση στο σπίτι :

- οι δραστηριότητες ΔΣ3 και ΔΣ4.

Τεστ αξιολόγησης επίδοσης

Α. Συμπληρώστε με σωστό ή λάθος

1. Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων.
2. Ο ταχύτερος μηχανισμός επεξεργασίας δεδομένων είναι ο υπολογιστής.
3. Ο έλεγχος των δεδομένων μπορεί να οδηγήσει και πάλι στην είσοδο.

Β. Επιλέξτε όσα χρειάζονται μεταξύ των προτεινόμενων

1. Οι λόγοι που οδηγούν στη χρησιμοποίηση υπολογιστή είναι σχετικοί με:

   α) την ταχύτητα επεξεργασίας

   β) την ευφυία υπολογιστή

   γ) την πολυπλοκότητα δεδομένων

   δ) το μεγάλο πλήθος των δεδομένων

2. Τα στάδια αντιμετώπιση ενός προβλήματος είναι

   α) η δόμηση

   β) η κατανόηση

   γ) η επίλυση

   δ) η ανάλυση

Γ. Συμπληρώστε τα κενά με τη λέξη που λείπει

1. Για να μπορέσουμε να επιλύσουμε ένα πρόβλημα θα πρέπει να γίνει ο καθορισμός.....
2. Η ..... προηγείται της επίλυσης και έπεται της κατανόησης ενός προβλήματος.

Απαντήσεις τεστ αξιολόγησης επίδοσης

Α1 : σωστό	Α2 : σωστό	Α3 : σωστό
Β1 : α, γ, δ	Β2 : β, γ, δ	
Γ1: απαιτήσεων	Γ2: ανάλυση	

1.5. Προβληματισμοί και θέματα προς συζήτηση

Θέματα προβληματισμού και συζήτησης μπορούν να προκύψουν από όλες τις δραστηριότητες που προτείνονται στο συγκεκριμένο κεφάλαιο. Συγκεκριμένα μπορούν να προταθούν συζητήσεις σχετικά με :

- Την αντιμετώπιση προβλημάτων με τη χρήση υπολογιστών.
- Τη δημιουργία προβλημάτων εξ αιτίας της χρήσης υπολογιστών.
- Μέτρο συσχέτισης λύσης προβλημάτων και υπολογιστών .

1.6. Προτεινόμενες πηγές πληροφόρησης

Βιβλιογραφία

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον διευθύνσεις στο Διαδίκτυο

- http://members.tripod.com/~ideas/workshop.html
- http://www.wcape.school.za/malati/Probsolve.htm

- http://www.viabilitygroup.com/Pages/ProbSolvOTL.htm
- http://www.stemnet.nf.ca/Curriculum/Math/Intermediate/problems.htm
- http://www.roma.unisa.edu.au/11571/module3_11571/index.htm
- http://www.hsdg.com/ps.htm

1.7. Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

1. Οι παρακάτω τομείς - με αναφορά σε ένα χαρακτηριστικό παράδειγμα – είναι τομείς στους οποίους η συμβολή των υπολογιστών υπήρξε θετική: υγεία (αξονικός τομογράφος), επικοινωνία (δορυφορικές υπηρεσίες), μεταφορές (έλεγχος εναέριας κυκλοφορίας), εκπαίδευση (εξ αποστάσεως), υποστήριξη ατόμων με ειδικές ανάγκες κλπ
2. Σαν σπουδαιότερα κοινωνικά προβλήματα από τη χρήση της Πληροφορικής θα μπορούσαν να χαρακτηριστούν οι επιπτώσεις στην εργασία (αύξηση της ανεργίας;), στο προσωπικό απόρρητο πληροφοριών, εθισμού και εξάρτησης.
3. Οι απαντήσεις των μαθητών είναι ελεύθερες και θα πρέπει να αιτιολογηθούν από τους ίδιους και να κριθούν από τους συμμαθητές τους αναπτύσσοντας με αυτόν τον τρόπο σχετική με το θέμα συζήτηση.
4. Δες παράγραφο 1.2.
5. Διαγραμματική αναπαράσταση.
6. Δες παράγραφο 1.5.
7. Δες παράγραφο 1.6.
8. Φραστικός, διαγραμματικός.
9. **Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή**

Σημείωση: Σε πολλές από τις προτεινόμενες δραστηριότητες οι απαντήσεις είναι ελευθέρου τύπου και ταυτόχρονα σε μεγάλο βαθμό υποκειμενικές. Κατά συνέπεια δεν προτείνονται συγκεκριμένες απαντήσεις, παρά μόνο αναφέρονται ενδει-

κτικά στοιχεία. Στόχος των συγκεκριμένων δραστηριοτήτων είναι πέρα από το αναπτύξουν οι μαθητές δεξιότητες σχετικές με τη διδακτική ενότητα (αναλυτική σκέψη, συνθετική ικανότητα, σαφήνεια διατύπωσης), να ευαισθητοποιηθούν σε κοινωνικά, πολιτισμικά και πολιτιστικά θέματα και να πληροφορηθούν στοιχεί από τρέχοντα θέματα της σύγχρονης κοινωνίας. Η ανάπτυξη αυτού του είδους των θεμάτων επαφίεται στο διδάσκοντα. Οι σχολικές τάξεις, ανεξάρτητα από το μάθημα που διδάσκεται κάθε διδακτική ώρα, θα πρέπει να είναι κυψέλες παιδαγώγησης και διάπλασης αυριανών πολιτών.

1.8. Απαντήσεις δραστηριοτήτων

➤ *Στην τάξη*

ΔΤ1.

Η δραστηριότητα στοχεύει ως προς την πρώτη ερώτηση, στην κατανόηση των εννοιών δεδομένο και πληροφορία, και ως προς την δεύτερη ερώτηση στην ανάπτυξη της αναλυτικής ικανότητας των μαθητών.

α) Τα στοιχεία του πίνακα που παραθέτεται αποτελούν πληροφορίες για το επίπεδο της χώρας μας, αφού προέρχονται από επεξεργασία δεδομένων (των απαντήσεων των πολιτών). Αντίθετα τα ίδια στοιχεία για το επίπεδο των 15 χωρών αποτελούν δεδομένα, αφού μαζί με αντίστοιχα στοιχεία από άλλες χώρες θα υποβληθούν σε επεξεργασία για προκύψει ένας αντίστοιχος πίνακας αναφορικά και με τις 15 χώρες. Το στοιχείο προς σχολιασμό είναι ακριβώς αυτό, το ότι το ίδιο γνωσιακό στοιχείο μπορεί να είναι πληροφορία σε μια κατάσταση και δεδομένο σε κάποια άλλη.

β) Η απάντηση είναι υποκειμενική. Ενδεικτικά προτείνονται σαν γενεσιουργές αιτίες η ανεργία, η απουσία πληροφόρησης, ο υπερεθνικισμός.

ΔΤ2.

Η δραστηριότητα στοχεύει στην καλλιέργεια της δυνατότητας ανάλυσης προβλήματος και διατύπωσής της φραστικά και διαγραμματικά. Η απάντηση είναι υποκειμενική. Ενδεικτικά προτείνεται η ανάλυση του προβλήματος σε: πληροφόρηση, παροχή κινήτρων. Το τελευταίο μπορεί να αναλυθεί σε αύξηση επιτοκίων καταθέσεων, μείωση φορολογίας κλπ.

ΔΤ3.

Η δραστηριότητα στοχεύει στο να καλλιεργήσουν οι μαθητές τη δυνατότητα ανάλυσης προβλήματος

Προτείνονται: α) όνομα, επώνυμο, όνομα πατέρα, όνομα μητέρας, τόπος γέννησης, ημερομηνία γέννησης β) όνομα, επώνυμο, πατρώνυμο, διεύθυνση κατοικίας, ΑΦΜ.

ΔΤ4.

Η δραστηριότητα είναι, στα πλαίσια του ορθολογικού, τελείως υποκειμενική από την πλευρά των μαθητών.

ΔΤ5.

Με αυτήν την δραστηριότητα στόχος είναι η επισήμανση της σοβαρότητας που έχει η σαφήνεια της διατύπωσης ενός προβλήματος στην αντιμετώπισή του.

Το κρίσιμο σημείο της διατύπωσης είναι ο λογικός τελεστής (συζευκτικός σύνδεσμος) ΚΑΙ που χρησιμοποιείται. Όλοι θα καταλάβαιναν τι ακριβώς θέλουμε να πούμε ακούγοντας τη διατύπωση. Όμως υπάρχει σοβαρό λάθος. Ο σύνδεσμός που θα έπρεπε να χρησιμοποιηθεί είναι ο Ή και όχι ο ΚΑΙ. Χρησιμοποιώντας τον ΚΑΙ ουσιαστικά ζητάμε να βρούμε όσους λέγονται Γιώργος και Γεωργία ταυτόχρονα. Η σωστή διατύπωση της πρότασης θα ήταν "Να ψάξω να βρω όλους όσους λέγονται Γιώργος ή Γεωργία" ή "Να ψάξω να βρω όλους όσους λέγονται Γιώργος και όλους όσους λέγονται Γεωργία". Φυσικά αν δίναμε στον υπολογιστή να εκτελέσει ένα ερώτημα (query) όπως ακριβώς είναι διατυπωμένο στο τετράδιο του μαθητή, το αποτέλεσμα της έρευνας θα ήταν κενό.

ΔΤ6.

Στόχος αυτής της δραστηριότητας είναι η ανάπτυξη της ικανότητας ανάλυσης προβλήματος εκ μέρους των μαθητών.

Η απάντηση είναι υποκειμενική. Μπορεί κάλλιστα να αποτελέσει θέμα συζήτησης και προβληματισμού. Θα πρέπει η εξέταση του θέματος να εστιάσει στην παράμετρο χρήση. Μπορούν να αναφερθούν πολλαπλά παρόμοια παραδείγματα, από τα πιο απλά μέχρι τα πιο σύνθετα, "καλής" και "κακής" χρήσης. Για παράδειγμα μπορούν να αναφερθούν : μαχαίρι (χρήσιμο εργαλείο – φονικό όργανο), ατομική ενέργεια (παροχή ενέργειας – πυρηνική βόμβα), υπολογιστής (ταχύτητα επεξεργασίας – εθισμός), διαδίκτυο (επιστημονική χρήση – hackers).

➤ Στο σπίτι

ΔΣ1.

Η δραστηριότητα στοχεύει στο να αναπτύξουν οι μαθητές δυνατότητα ανάλυσης προβλήματος.

Το συγκεκριμένο πρόβλημα είναι, και σε πραγματικό επίπεδο, δύσκολο να αντιμετωπιστεί. Η προσπάθεια αντιμετώπισής του θα μπορούσε να γίνει σε τρία επίπεδα : τεχνικό, νομικό και ηθικό. Η παραπέρα ανάλυση του προβλήματος σε τεχνικό επίπεδο μας δίνει δύο συνιστώσες, υλικό και λογισμικό. Το υλικό μπορεί να διακριθεί σε τοπικό (όπου βρίσκεται ο υπολογιστής με τα εν λόγω στοιχεία) και απομακρυσμένο (όπου βρίσκεται αυτός που επιχειρεί την πρόσβαση) . Το λογισμικό μπορεί να διακριθεί σε κωδικό πρόσβασης και κρυπτογράφηση των φυλασσόμενων στοιχείων. Η ανάλυση σε νομικό επίπεδο μπορεί να διακριθεί σε ποινικές και αστικές ευθύνες και κατά συνέπεια επιπτώσεις. Στις αστικές ευθύνες μπορεί να συμπεριλαμβάνεται στέρηση δικαιώματος πρόσβασης στο διαδίκτυο.

ΔΣ2.

Η δραστηριότητα στοχεύει στην καλλιέργεια της δυνατότητας ανάλυσης προβλήματος.

Η απάντηση είναι υποκειμενική. Ενδεικτικά προτείνεται ότι για την ανάλυση του προβλήματος θα μπορούσαν να ληφθούν υπόψη ψυχολογικοί, πολιτισμικοί και κοινωνικοί παράγοντες. Τους κοινωνικούς παράγοντες μπορούμε να τους διακρίνουμε σε σταδιακή εξαφάνιση της γειτονιάς και της παρέας, εξατομίκευση του σύγχρονου ανθρώπου κλπ. Στους πολιτισμικούς παράγοντες μπορούμε να διακρίνουμε εισβολή της τεχνολογίας στην καθημερινή μας ζωή, τεράστια αύξηση της επιρροής των ΜΜΕ, κλπ. Οι ψυχολογικοί παράγοντες θα μπορούσαν να αναλυθούν σε αυτούς που σχετίζονται με το νεαρό της ηλικίας, τη γοητεία της ανακάλυψης μέσω του διαδικτύου, τη ψυχολογία του θεατή στην κοινωνία του θεάματος κλπ.

ΔΣ3.

Η δραστηριότητα στοχεύει στο να μπορούν οι μαθητές να προσδιορίζουν τα ζητούμενα αποτελέσματα ενός προβλήματος.

Προτείνονται : α) για το εργαστήριο πληροφορικής: ισχυρός server, υπολογιστές, κάρτες δικτύου, εκτυπωτής β) για το εργαστήριο καλλιτεχνικών σπουδών: υπολογιστής, scanner, videocamera, κάρτα ήχου, μικρόφωνο, ηχεία γ) για τη γραμ-

ματεία: υπολογιστής, κάρτα fax/modem, εκτυπωτής δ) για την αίθουσα εκδηλώσεων/παρουσιάσεων: υπολογιστής, barco ή overview display.

ΔΣ4.

Η δραστηριότητα στοχεύει στο να καλλιεργήσουν οι μαθητές την ικανότητά τους στη διατύπωση προβλήματος.

Η δραστηριότητα είναι, στα πλαίσια του ορθολογικού, τελείως υποκειμενική από την πλευρά των μαθητών.

ΔΣ5.

Η δραστηριότητα στοχεύει στην εξάσκηση των μαθητών στην μεθοδολογία καταγραφής των δεδομένων και δημιουργίας των ζητουμένων αποτελεσμάτων ενός προβλήματος.

Να δοθεί στους μαθητές η δυνατότητα απόκτησης των περιγραφόμενων στοιχείων, να δημιουργηθεί ο πίνακας συχνοτήτων και να βρεθεί η μέση τιμή και η τυπική απόκλιση με βάση τα στοιχεία αυτά. Οι μαθητές προηγούμενων ετών, των οποίων τα στοιχεία βαθμολογίας προτείνεται να χρησιμοποιηθούν, να μην υπερβαίνουν τους 20.

Η δραστηριότητα αυτή εισάγει τους μαθητές σε μια μεθοδολογία έρευνας, στοιχείο ιδιαίτερα χρηστικό σε επιστημονικό, ερευνητικό αλλά και επαγγελματικό επίπεδο.

Κεφάλαιο 2

Βασικές έννοιες αλγορίθμων

2.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να κατανοήσουν οι μαθητές την έννοια του αλγορίθμου, να αναγνωρίσουν τη σπουδαιότητα των αλγορίθμων και να εκτιμήσουν την αναγκαιότητα της αλγοριθμικής προσέγγισης για την επίλυση προβλημάτων με σταδιακή προσέγγιση των αλγοριθμικών εννοιών χρησιμοποιώντας συγκεκριμένες τεχνικές και συνιστώσες επίλυσης προβλημάτων.

2.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να αναγνωρίζουν τη σπουδαιότητα και την αναγκαιότητα των αλγορίθμων.
- να αποτυπώνουν ένα πρόβλημα σε βήματα αλγορίθμου με συγκεκριμένη δομή
- να διακρίνουν τα είδη των αλγοριθμικών συνιστωσών που χρησιμοποιούνται για την επίλυση προβλημάτων
- να χρησιμοποιούν σωστά τις συνιστώσες της ακολουθίας, της επιλογής καθώς και των πολλαπλών επιλογών
- να εφαρμόζουν στις κατάλληλες περιπτώσεις τα είδη των συνιστωσών της επανάληψης
- να χρησιμοποιούν εμφωλευμένες αλγοριθμικές δομές και να συνδυάζουν διάφορες συνιστώσες μεταξύ τους
- να κατανοούν τον αλγόριθμο του πολλαπλασιασμού αλά ρωσικά και της διαφορετικότητας του σε σχέση με τον κλασικό πολλαπλασιασμό

2.3 Οδηγίες – Επισημάνσεις

Ιδιαίτερη έμφαση και προσοχή πρέπει να δοθεί στα παρακάτω θεματικά αντικείμενα:

Δομές ακολουθίας, επιλογής και επανάληψης.

Είναι απαραίτητο να γίνουν αρκετά παραδείγματα για κάθε μία από τις συνιστώσες αυτές και να δοθεί έμφαση στη συνιστώσα της επιλογής και της επανάληψης. Θα πρέπει να τονισθεί ιδιαίτερα η αναγκαιότητα ύπαρξης διαφορετικών δομών επανάληψης και να δοθούν αντιπροσωπευτικά παραδείγματα για κάθε περίπτωση.

Συνδυασμός αλγοριθμικών δομών.

Είναι αναγκαίο να αναλυθεί και να εξηγηθεί με χρήση παραδειγμάτων η χρησιμότητα του συνδυασμού των αλγοριθμικών δομών για την επίλυση προβλημάτων. Έμφαση θα πρέπει να δοθεί στη συχνότητα συνδυασμού των συνιστωσών της επιλογής και της επανάληψης για την αλγοριθμική επίλυση ακόμα και απλών προβλημάτων.

Εναλλακτικές διδακτικές προσεγγίσεις παρουσιάζονται στην παρ. 2.9.

2.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

τρία (3) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- να αναγνωρίζουν τη σπουδαιότητα και την αναγκαιότητα των αλγορίθμων.
- να αποτυπώνουν ένα πρόβλημα σε βήματα αλγορίθμου με συγκεκριμένη δομή
- να χρησιμοποιούν σωστά τις συνιστώσες της ακολουθίας και της επιλογής

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Τι είναι αλγόριθμος:** εισαγωγή στην έννοια του αλγορίθμου, ορισμοί.
- **Σπουδαιότητα αλγορίθμων:** τεκμηρίωση της σπουδαιότητας των αλγορίθμων και συσχέτιση της επιστήμης της Πληροφορικής με τη μελέτη των αλγορίθμων.
- **Περιγραφή και αναπαράσταση αλγορίθμων:** αποτύπωση και κατηγοριοποίηση των διαφόρων τρόπων αναπαράστασης αλγορίθμων.
- **Ρεπερτόριο συμβόλων διαγράμματος ροής και εντολών ψευδογλώσσας:** καταγραφή των κυριοτέρων δομικών στοιχείων ενός διαγράμματος ροής και παρουσίαση των εντολών της ψευδογλώσσας που θα υιοθετηθεί για την καταγραφή και την αποτύπωση των αλγορίθμων.
- **Η δομή της ακολουθίας:** εισαγωγή στη δομή της ακολουθίας με χρήση απλών παραδειγμάτων.

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν οι παράγραφοι 2.1 έως και 2.4.2 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή. Να απαντηθούν τουλάχιστον 5 από τις ερωτήσεις 1-8 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα 1 και 2 του αντίστοιχου κεφαλαίου από το τετράδιο του μαθητή.

Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ1 ή ΔΤ2 στην τάξη, και μία από τις δραστηριότητες ΔΣ1 ή ΔΣ2 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Να απαντηθούν 10 από τις 15 ερωτήσεις του τεστ αξιολόγησης

Συμπληρώστε με σωστό ή λάθος

1. Ο αλγόριθμος είναι απαραίτητος μόνο για την επίλυση προβλημάτων Πληροφορικής.
2. Ο αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο εντολών.
3. Ο αλγόριθμος μπορεί να περιλαμβάνει και εντολές που δεν είναι σαφείς.
4. Η Πληροφορική μελετά τους αλγορίθμους μόνο από το πρίσμα των γλωσσών προγραμματισμού.
5. Η αναπαράσταση των αλγορίθμων μπορεί να γίνει μόνο με χρήση ελεύθερου κειμένου και φυσικής γλώσσας.
6. Τα κυριότερα σύμβολα των διαγραμμάτων ροής είναι η έλλειψη, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο.
7. Οι κυριότερες εντολές ψευδογλώσσας των αλγορίθμων είναι οι αριθμητικές και αλφαριθμητικές αναθέσεις τιμών σε μεταβλητές.
8. Η ακολουθιακή δομή εντολών χρησιμοποιείται για την επίλυση απλών προβλημάτων με δεδομένη τη σειρά εκτέλεσης ενός συνόλου ενεργειών.
9. Η δομή της ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση πολύπλοκων προβλημάτων.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση

10. Κάθε αλγόριθμος πρέπει να ικανοποιεί το κριτήριο της:

    α) επιλογής β) ακολουθίας

    γ) ανάθεσης δ) περατότητας

11. Η επιστήμη της Πληροφορικής περιλαμβάνει τη μελέτη των αλγορίθμων μεταξύ άλλων και από τη σκοπιά:

    α) υλικού και λογισμικού β) ελεύθερου κειμένου

    γ) αποτελεσματικότητας δ) ανάγνωσης /εκτύπωσης

12. Ενας από τους τρόπους αναπαράστασης των αλγορίθμων είναι:

    α) γλώσσα προγραμματισμού β) θεωρητική τυποποίηση

γ) διαγραμματικές τεχνικές δ) αριθμητικές πράξεις

13. Ποιά από τις παρακάτω αναπαραστάσεις εκχωρεί στη μεταβλητή Α την τιμή 138

α) A=138 β) A=:138 γ) A:=138 δ) A ← 138

14. Ποιο από τα παρακάτω αποτελεί σύμβολο για τα διαγράμματα ροής:

α) έλλειψη β) τραπέζιο

γ) κύκλος δ) τετράγωνο

15. Ποιά από τα παρακάτω αποτελούν εντολές της ψευδογλώσσας των αλγορίθμων:

α) A+B = 10 β) A ← B*3

γ) A+B ← 12 δ) A ← 2*B ← 22

Απαντήσεις τεστ αξιολόγησης επίδοσης

1: λάθος	6: σωστό	11: α
2: σωστό	7: λάθος	12: γ
3: λάθος	8: σωστό	13: δ
4: λάθος	9: λάθος	14: α
5: λάθος	10: δ	15: β

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να εμβαθύνουν και να κατανοήσουν τη δομή της επιλογής καθώς και των πολλαπλών επιλογών
- να συνδυάζουν και να εφαρμόζουν στις κατάλληλες περιπτώσεις τα είδη των συνιστωσών της επιλογής
- να κατανοήσουν τη χρήση βασικών δομών επανάληψης

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Δομή Επιλογής.** Επεξήγηση της αναγκαιότητας και σπουδαιότητας της δομής της επιλογής. Απλή περίπτωση μίας περίπτωσης επιλογής και πλήρης επιλογή για κάθε περίπτωση περιπτώσεων.
- **Διαδικασίες πολλαπλών επιλογών.** Παρουσίαση της δομής των πολλαπλών επιλογών και τυποποίηση της δομής με χρήση λογικού διαγράμματος.
- **Εμφωλευμένες Διαδικασίες.** Επεξήγηση της αναγκαιότητας χρήσης αλγοριθμικών δομών όπου η μία εντάσσεται στα όρια της άλλης και συνδυάζεται η χρήση τους για επίλυση προβλημάτων με απαιτήσεις.
- **Δομή Επανάληψης**. Εισαγωγή και γενική επεξήγηση της διαδικασίας της επανάληψης και αναφορά στο είδος επανάληψης «όσο... επανάλαβε».

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν οι παράγραφοι από 2.4.3 έως και την αρχή της παραγράφου 2.4.5 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή. Να απαντηθούν οι ερωτήσεις 10, 11 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα 3 και 5 του αντίστοιχου κεφαλαίου από το τετράδιο του μαθητή.

Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση δύο από τις δραστηριότητες ΔΤ4, ΔΤ6 και ΔΤ8 στην τάξη, και μία από τις δραστηριότητες ΔΣ5, ΣΔ7 ή ΣΔ9 στο σπίτι.

Τεστ αξιολόγησης επίδοσης

Να απαντηθούν 8 από τις 12 ερωτήσεις του τέστ αξιολόγησης.

Συμπληρώστε με σωστό ή λάθος

1. Η δομή της επιλογής χρησιμοποιείταιι στις περιπτώσεις όπου υπάρχει μία συγκεκριμένη σειρά βημάτων για την επίλυση ενός προβλήματος.
2. Οταν χρειάζεται να υπάρξει απόφαση με βάση κάποιο κριτήριο, τότε χρησιμοποιείται η δομή της επιλογής.
3. Η δομή της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής).
4. Οι διαδικασίες των πολλαπλών επιλογών εφαρμόζονται στα προβλήματα όπου πάντοτε λαμβάνεται η ίδια απόφαση ανάλογα με την τιμή που παίρνει μία μεταβλητή.
5. Μία εμφωλευμένη δομή μπορεί να συμπεριλαμβάνει μόνο την πράξη της ανάθεσης τιμών.
6. Μία εντολή «Αν...τότε» δεν μπορεί να περιληφθεί στα όρια κάποιας άλλης εντολής «Αν...τότε».

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση

7. Μία εντολή «Αν...τότε» περιλαμβάνει κάποια:

   α) συνθήκη β) ακολουθία

   γ) ανάθεση δ) επανάληψη

8. Οι εμφωλευμένες δομές περιλαμβάνουν συνδυασμό:

   α) συνθήκης και εκτύπωσης β) διαφόρων αλγοριθμικών δομών

   γ) συνθήκης και ανάγνωσης δ) ανάγνωσης και εκτύπωσης

9. Μία εμφωλευμένη δομή χρησιμοποιείται όταν χρειάζεται:

α) μία ενέργεια να περιληφθεί μέσα σε άλλη ενέργεια

β) να υπάρχει επανάληψη τυποποιημένων ενεργειών

γ) να υπάρχει εκτύπωση και ανάγνωση τιμών

δ) να επαναληφθεί μία ενέργεια πολλές φορές

10. Η λογική πράξη ***ή*** μεταξύ 2 προτάσεων είναι αληθής όταν:

α) οποιαδήποτε από τις δύο προτάσεις είναι αληθής.

β) η πρώτη πρόταση είναι ψευδής.

γ) η δεύτερη πρόταση είναι ψευδής.

δ) και οι δύο προτάσεις είναι αληθής.

11. Η λογική πράξη ***και*** μεταξύ 2 προτάσεων είναι αληθής όταν:

α) οποιαδήποτε από τις δύο προτάσεις είναι αληθής.

β) η πρώτη πρόταση είναι αληθής.

γ) η δεύτερη πρόταση είναι αληθής.

δ) και οι δύο προτάσεις είναι αληθείς.

12. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις όπου:

α) μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε δύο περιπτώσεις

β) μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων

γ) υπάρχει απαίτηση να ληφθεί μία απόφαση με βάση κάποια συνθήκη

δ) υπάρχουν δύο συνθήκες που πρέπει να ισχύουν η μία μετά την άλλη.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1: λάθος	7: α
2: σωστό	8: β
3: σωστό	9: α
4: λάθος	10: α
5: λάθος	11: δ
6: λάθος	12: β

Σχέδιο 3ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να χρησιμοποιούν σωστά τα διαφορετικά είδη των συνιστωσών επανάληψης.
- να χρησιμοποιούν εμφωλευμένες αλγοριθμικές δομές και να συνδυάζουν διάφορες συνιστώσες μεταξύ τους.
- να κατανοήσουν τον αλγόριθμο του πολλαπλασιασμού αλά ρωσικά και τη διαφορετικότητά του ως προς τον κλασικό πολλαπλασιασμό.

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Δομές Επανάληψης:** Επεξηγείται και δίνεται έμφαση στην αναγκαιότητα των διαδικασιών επανάληψης.

- **Είδη επαναληπτικών δομών:** Ανάλυση και τυποποίηση των διαφορετικών επαναληπτικών δομών:

  Α) **'Οσο** <συνθήκη> **επανάλαβε**

  Διαδικασία

  Τέλος_επανάληψης

  Β) **Αρχή_επανάληψης**

  Διαδικασία

  Μέχρις_ότου <συνθήκη>

  Γ) **Για** μεταβλητή **από** τ1 **μέχρι** τ2 **με_βήμα** β

  Διαδικασία

  Τέλος_επανάληψης

- **Πολλαπλασιασμός αλά ρωσικά.** Επεξήγηση, τεκμηρίωση χρησιμότητας και ιδιαιτερότητας του πολλαπλασιασμού αλά ρωσικά.
- **Επανάληψη των εννοιών που διδάχθηκαν – Ανακεφαλαίωση**

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν οι παράγραφοι 2.4.5 έως και 2.5 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή. Η παράγραφος 2.4.5 πρέπει να ολοκληρωθεί σε αυτό το μάθημα καθώς αποτελεί συνέχεια του προηγουμένου μαθήματος. Να απαντηθούν δύο από τις ερωτήσεις 9, 12, 13 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα 5 και 6 του αντίστοιχου κεφαλαίου από το τετράδιο του μαθητή.

Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση δύο από τις δραστηριότητες ΔΤ3, ΔΤ5 και ΔΤ7 στην τάξη, και μία από τις δραστηριότητες ΔΣ3, ΔΣ7 και ΔΣ8 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Να απαντηθούν 8 από τις 12 ερωτήσεις του τεστ αξιολόγησης.

Συμπληρώστε με σωστό ή λάθος

1. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό.

2. Οι επαναληπτικές διαδικασίες εφαρμόζονται όταν μία ακολουθία εντολών χρειάζεται να εκτελεσθεί περισσότερες από μία φορές.

3. Οι επαναληπτικές διαδικασίες εμπεριέχουν πάντοτε συνθήκες επιλογών.

4. Με χρήση της εντολής "Όσο...επανάλαβε" επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη.

5. Με την εντολή «Αρχή_επανάληψης...Μέχρις_ότου...» υπάρχει ένας βρόχος που θα εκτελεσθεί οπωσδήποτε τουλάχιστον μία φορά.

6. Η εντολή "Για i από .. μέχρι .. με_βήμα .." πρέπει να περιλαμβάνει για βήμα πάντοτε ένα θετικό αριθμό.

Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει

7. Η διαδικασία της _____ είναι ιδιαίτερα συχνή, για πλήθος προβλημάτων μπορεί να επιλυθεί με κατάλληλες επαναληπτικές διαδικασίες.

8. Η επαναληπτική δομή «Όσο ... επανάλαβε» περιλαμβάνει κάποια(ες) διαδικασίες και λήγει με τη φράση _____

9. Η επαναληπτική δομή που περιλαμβάνει έλεγχο επανάληψης στο τέλος της διαδικασίας ξεκινά με τη φράση «Αρχή_επανάληψης » και λήγει με τη φράση _____

10. Η δομή «_____ από τ1 μέχρι τ2 με_βήμα β » αποτελεί ένα επαναληπτικό σχήμα ορισμένων φορών επανάληψης.

11. Ο πολλαπλασιασμός _____ απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση.

12. Ο αλγόριθμος που δεν διαθέτει τρόπο τερματισμού χαρακτηρίζεται ως _____ βρόχος..

Απαντήσεις τεστ αξιολόγησης επίδοσης

1: σωστό	7: Επανάληψης
2: σωστό	8: Τέλος_Επανάληψης
3: λάθος	9: Μέχρις ότου
4: σωστό	10: Για μεταβλητή
5: σωστό	11: αλά ρωσικά
6: λάθος	12: Ατέρμων

2.5 Προβληματισμοί και θέματα προς συζήτηση

- Είναι χρήσιμο να τονισθεί η ιδιαίτερη χρησιμότητα των αλγορίθμων και να ζητηθεί από τους μαθητές να αναφερθούν σε παραδείγματα από την καθημερινή τους ζωή στα οποία θα μπορούσαν να χρησιμοποιήσουν αλγοριθμική προσέγγιση.
- Οι επαναληπτικές δομές και η δομή της επιλογής θα πρέπει να συζητηθούν και να δοθεί έμφαση στη χρησιμότητα τους.

2.6 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

2.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

1. Να δοθεί ο ορισμός του όρου "αλγόριθμος". Δες παράγραφο 2.1 του βιβλίου.
2. Ποιά είναι τα κριτήρια που πρέπει να ικανοποιεί κάθε αλγόριθμος. Δες παράγραφο 2.1 του βιβλίου.

3. Υπό ποία πρίσματα η Πληροφορική επιστήμη μελετά τους αλγορίθμους. Δες παράγραφο 2.2 του βιβλίου.

4. Ποιά η διαφορά της θεωρητικής από την αναλυτική προσέγγιση στην επίλυση ενός προβλήματος με χρήση αλγορίθμου; Η θεωρητική προσέγγιση προσδιορίζει τα όρια της λύσης ενός αλγορίθμου που θα βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα, ενώ η αναλυτική προσέγγιση μελετά τους υπολογιστικούς πόρους που απαιτούνται από έναν αλγόριθμο.

5. Περιγράψτε τους τρόπους περιγραφής και αναπαράστασης των αλγορίθμων. Δες παράγραφο 2.3 του βιβλίου.

6. Ποιοί είναι οι βασικοί τύποι συνιστωσών/εντολών ενός αλγορίθμου; Υπάρχουν οι δομές ακολουθίας (σειριακών εντολών, αναθέσεων τιμών), επιλογής με βάση κριτήρια, διαδικασιών επανάληψης, ενεργειών πολλαπλών επιλογών καθώς και συνδυασμούς εμφωλευμένων περιπτώσεων.

7. Να περιγραφεί η δομή της ακολουθίας και να δοθεί σε διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης. Δες παράγραφο 2.4.1 του βιβλίου.

8. Να περιγραφεί η δομή της επιλογής και να δοθεί με ακολουθία βημάτων ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης. Δες παράγραφο 2.4.2 του βιβλίου

9. Να περιγραφεί η δομή των επαναληπτικών διαδικασιών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης. Δες παράγραφο 2.4.5 του βιβλίου.

10. Να περιγραφεί η δομή των διαδικασιών πολλαπλών επιλογών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης. Δες παράγραφο 2.4.3 του βιβλίου.

11. Να περιγραφεί η δομή των εμφωλευμένων διαδικασιών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης. Δες παράγραφο 2.4.4 του βιβλίου.

12. Να περιγραφεί με ακολουθία βημάτων το πρόβλημα του "πολλαπλασιασμού αλά ρωσικά". Δες Παράδειγμα 12 του βιβλίου

13. Ποιά η πρακτική σημασία του αλγορίθμου του "πολλαπλασιασμού αλά ρωσικά". Πότε γίνεται χρήση αυτού του τρόπου πολλαπλασιασμού δύο ακεραίων; Δες Παράδειγμα 12 του βιβλίου.

2.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

ΔΤ1.

```
Αλγόριθμος εκκρεμές
Διάβασε L, g
T ← 2 * 3.14 * Ρίζα(L/g)
Εκτύπωσε T
Τέλος εκκρεμές
```

ΔΤ2 .

```
Αλγόριθμος συνάλλαγμα
Euro ← 330
lira ← 550
dollar ← 280
marko ← 100
synolo ← 1025*lira+2234*dollar+3459*marko
Εκτύπωσε synolo
Τέλος συνάλλαγμα
```

Σχόλιο: η διατήρηση διαφορετικών μεταβλητών για κάθε νόμισμα είναι ιδιαίτερα χρήσιμη γιατί σε περίπτωση ενημέρωσης νέων τιμών συναλλάγματος, αλλάζουν μόνο οι μεταβλητές και όχι ο τύπος για τον υπολογισμό του συνόλου.

ΔΤ3.

1.

```
Αλγόριθμος Μέσος_Ορος
ATHR ← 0
Για i από 1 μέχρι 100
```

```
    Διάβασε HLIKIA
    ATHR ← ATHR+ HLIKIA
Τέλος_επανάληψης
MO ← ATHR/100
Αποτελέσματα // MO //
Τέλος Μέσος_Ορος
```

2. Η άσκηση υλοποιείται με δεδομένες τις βαθμολογίες 5 ομάδων.

```
Αλγόριθμος Ομάδες
ATHR ← 0
Για i από 1 μέχρι 5
    Διάβασε VATHMOS
    Αν VATHMOS>100 τότε ATHR ← ATHR+VATHMOS
Τέλος_επανάληψης
Αποτελέσματα // ATHR //
Τέλος Ομάδες
```

ΔΤ4.

1. Επαναληπτική δομή.
2. Επιλογή
3. Επανάληψη και επιλογή.
4. Ανάθεση και ακολουθία.
5. Επανάληψη και ακολουθία.
6. Ανάθεση.

ΔΤ5.

```
Αλγόριθμος Μόλυνση
Δεδομένα // AN, AZ //
Αν AN<0.35 τότε
    Εκτύπωσε "καθαρή"
Αλλιώς
    Εκτύπωσε "μολυσμένη"
Τέλος_αν
```

```
Αν AZ<0.17 τότε
    Εκτύπωσε "διαυγής"
Αλλιώς
    Εκτύπωσε "αδιαυγής"
Τέλος_αν
Τέλος Μόλυνση
```

ΔΤ6.

Ο αλγόριθμος του μέσου όρου που χρησιμοποιήθηκε για τις ηλικίες στη Δραστηριότητα ΔΤ3 (ερώτημα 1) χρειάζεται να εφαρμοσθεί για να υπολογισθεί ο μέσος όρος σύμφωνα με τα δεδομένα της τάξης. Στη συνέχεια πρέπει να δοθεί το ακόλουθο τμήμα αλγορίθμου:

```
Αν ΜΟ>18 τότε
    Εκτύπωσε "ΣΥΜΜΕΤΟΧΗ"
Αλλιώς
    Εκτύπωσε "ΜΗ-ΣΥΜΜΕΤΟΧΗ"
Τέλος_αν
```

ΔΤ7.

```
Αλγόριθμος εισφορές
Δεδομένα // MISTHOS//
Αν MISTHOS<150000 τότε
    EISF1 ← 0.05*MISTHOS
    EISF2 ← 0.04*MISTHOS
Αλλιώς_αν MISTHOS<250000  τότε
   EISF1 ← 0.075*MISTHOS
   EISF2 ← 0.06*MISTHOS
Αλλιώς_αν MISTHOS<400000 τότε
   EISF1 ← 0.095*MISTHOS
   EISF2 ← 0.08*MISTHOS
Αλλιώς
   EISF1 ← 0.12*MISTHOS
   EISF2 ← 0.11*MISTHOS
Τέλος_αν
Εκτύπωσε ESIF1, EISF2, MISTHOS-(EISF1+EISHF2)
```

```
Τέλος εισφορές
```

ΔΤ8.

Το πρόβλημα αυτό ακολουθεί τον αλγόριθμο Ελάχιστη_Μέγιστη1 που δόθηκε στο Παράδειγμα 3 (τιμές θερμοκρασίας από Μετεωρολογικό Κέντρο) του Τετραδίου του Μαθητή. Χρειάζεται μόνο να δοθεί η κατάλληλη τιμή στις μεταβλητές MIN και MAX σε σχέση με την εκφώνηση του προβλήματος. Επομένως εάν δοθούν αρχικά

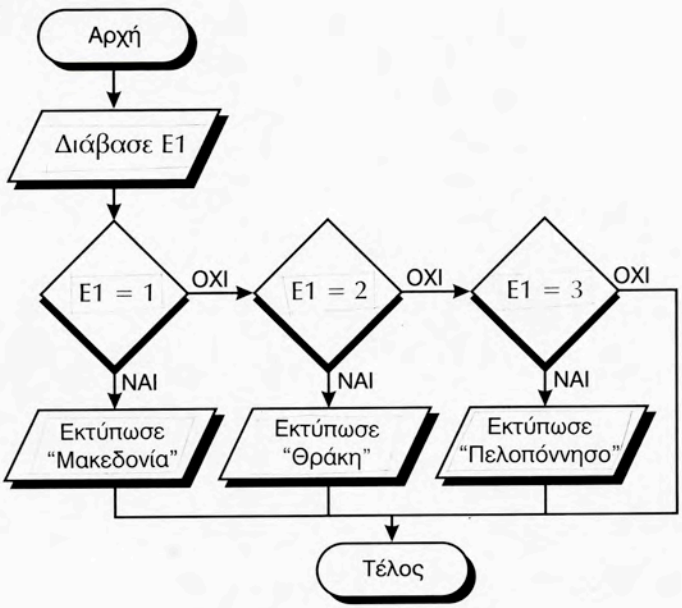
MIN ← 1000000

MAX ← 0

δεν θα υπάρξει πρόβλημα με τον υπολογισμό του μικρότερου και του μεγαλύτερου αριθμού προσπελάσεων.

ΔΤ9.

Διάγραμμα ροής



Αλγόριθμος

```
Αλγόριθμος Φυτώριο
Δεδομένα // Ε //
Αν Ε=1 τότε
    Εκτύπωσε " Μακεδονία"
αλλιώς_αν Ε=2 τότε
    Εκτύπωσε "Θράκη"
αλλιώς_αν Ε=3 τότε
    Εκτύπωσε "Πελοπόννησος"
Τέλος_αν
Τέλος Φυτώριο
```

ΔΤ10.

```
Αλγόριθμος Μουσείο
Δεδομένα // Ε1,Ε2,Ε3,Ε4,Ε5,Ε6,Ε7,Ε8,Ε9,Ε10 //
ΑΤΗR ← Ε1+Ε2+Ε3+Ε4+Ε5+Ε6+Ε7+Ε8+Ε9+Ε10
ΜΟ ← ΑΤΗR/10
Αν Ε1>ΜΟ τότε Εκτύπωσε 101
Αν Ε2>ΜΟ τότε Εκτύπωσε 102
Αν Ε3>ΜΟ τότε Εκτύπωσε 103
...
Αν Ε10>ΜΟ τότε Εκτύπωσε 110
Τέλος Μουσείο
```

Είναι χρήσιμο να τονισθεί στο παράδειγμα αυτό η ανάγκη χρησιμοποίησης 10 διαφορετικών μεταβλητών για τον αριθμό των επισκεπτών κάθε αίθουσας λόγω του ότι χρησιμοποιουνται σε διάφορα σημεία του αλγορίθμου. Να αναφερθεί ότι προβλήματα σαν κι αυτό θα επιλυθούν με επαναληπτική διαδικασία και χρήση δομών δεδομένων (π.χ. πίνακας) σε επόμενο κεφάλαιο.

➤ ***Στο σπίτι***

ΔΣ1.

Να γραφεί αλγόριθμος που θα διαβάζει 200 τιμές, θα υπολογίζει και θα εκτυπώνει το άθροισμα των τιμών (από όσες διαβάσθηκαν) που είναι μεγαλύτερες από το 10.

ΔΣ2.

```
Αλγόριθμος απαξίωση
xronos ← 0.5
arhiki ← 295600
prosfora ← 256000
deval ← 1-Δύναμη(prosfora/arhiki,1/xronos)
Εκτύπωσε deval
Τέλος απαξίωση
```

Για να γενικευτεί ο αλγόριθμος αρκεί οι παραπάνω εντολές να μπουν σε ένα βρόχο επανάληψης και να γίνεται ανάγνωση και όχι ανάθεση τιμών στις μεταβλητές arhiki, prosfora, xronos.

Σχόλιο: Στον προηγούμενο αλγόριθμο γίνεται χρήση της συνάρτησης Δύναμη. Πολλές σύγχρονες γλώσσες υποστηρίζουν τη συνάρτηση αυτή. Σε επόμενο κεφάλαιο θα δοθεί και αλγοριθμική προσέγγιση. Εναλλακτικά μπορεί να χρησιμοποιηθεί η εντολή

deval ← 1-(prosfora/arhiki) ^ (1/xronos)

ΔΣ3.

```
Αλγόριθμος καταναλωτής
poso ← 5000
agores ← 0
euro ← 340,75
flag ← Ψευδής
Αρχή_επανάληψης
    Διάβασε eidos
    timi ← eidos/euro
```

```
    Αν agores+timi<=poso*euro τότε
        agores ← agores+timi
    αλλιώς
        flag ← Αληθής
    Τέλος_Αν
Μέχρις ότου flag=Αληθής
Τέλος καταναλωτής
```

ΔΣ4.

x=13 Το x παίρνει διαδοχικά τις τιμές 40, 20, 10, 5, 16, 8, 4, 2, 1

x=9 Το x παίρνει διαδοχικά τις τιμές 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

x=22 Το x παίρνει διαδοχικά τις τιμές 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

Παρατηρούμε ότι καταλήγουμε στην ίδια ακολουθία ανάθεσης αριθμών από κάποιο σημείο μέχρι την κατάληξη στο x=1.

ΔΣ5.

Χρειάζεται να ακολουθηθεί ο ίδιος αλγόριθμος με τον αλγόριθμο για το Μέσο όρο που δόθηκε στη δραστηριότητα ΔΤ3 (ερώτημα 1).

ΔΣ6.

Για να υπάρξει έλεγχος για την πλέον συμφέρουσα τιμή χρειάζεται να υπολογισθεί η τιμή του κάθε γάλακτος για την ίδια ποσότητα. Επομένως ο αλγόριθμος υπολογίζει την τιμή του κάθε είδους γάλακτος για τα 100ml και μετά υπολογίζει την ελάχιση από αυτές τις τιμές.

```
Αλγόριθμος Τιμή_Γάλα
Δεδομένα // ΓΑΛΑ_Α, ΓΑΛΑ_Β, ΓΑΛΑ_Γ, ΓΑΛΑ_Δ //
ΓΑΛΑ_Α ← 195/3
ΓΑΛΑ_Β ← 205/4
ΓΑΛΑ_Γ ← 400/5
ΓΑΛΑ_Δ ← 450/5.5
MIN ← ΓΑΛΑ_Α
```

```
i ← 1
Αν ΓΑΛΑ_Β<MIN τότε
    MIN ← ΓΑΛΑ_Β
    i ← 2
Τέλος_αν
Αν ΓΑΛΑ_Γ<MIN τότε
    MIN ← ΓΑΛΑ_Γ
    i ← 3
Τέλος_αν
Αν ΓΑΛΑ_Δ<MIN τότε
    MIN ← ΓΑΛΑ_Δ
    i ← 4
Τέλος_αν
Αποτελέσματα // i, MIN //
Τέλος Τιμή_Γάλα
```

ΔΣ7.

Ο αλγόριθμος υλοποίησης είναι παρόμοιος με τον αλγόριθμο που δόθηκε στη Δραστηριότητα ΔΣ2 προηγουμένως. Υπάρχει μόνο η διαφοροποίηση για τον τύπο που πρέπει να χρησιμοποιηθεί και εδώ θα δοθεί ο αντίστοιχος τύπος από την εκφώνηση.

ΔΣ8.

```
Αλγόριθμος Δενδροφύτευση
Δεδομένα // s1, s2, s3 //
s1 ← 0
s2 ← 0
s3 ← 0
Για i από 1 μέχρι 100
    Διάβασε Ε1
    Αν Ε1=1 τότε
       s1 ← s1+1
    αλλιώς_αν  Ε1=2 τότε
       s2 ← s2+1
    αλλιώς_αν  Ε1=3 τότε
       s3 ← s3+1
```

```
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // s1, s2, s3 //
Τέλος Δενδροφύτευση
```

ΔΣ9.

```
Αλγόριθμος οργάνωση_εκδήλωσης
!Σχόλιο price και capacity είναι η τιμή και η χωρητικότητα της
αίθουσας, donation είναι η προσφορά
Διάβασε price1,capacity1
MAX ← capacity
cost ← price1
Διάβασε price2,capacity2
Αν capacity2>MAX τότε
    MAX ← capacity2
    cost ← price2
Τέλος_αν
Διάβασε price3,capacity3
Αν capacity3>MAX τότε
    MAX ← capacity3
    cost ← price3
Τέλος_αν
Διάβασε donate1, donate2, donate3, donate4, donate5
count ← 0
Αν donate1>=cost τότε count ← count+1
Αν donate2>=cost τότε count ← count+1
Αν donate3>=cost τότε count ← count+1
Αν donate4>=cost τότε count ← count+1
Αν donate5>=cost τότε count ← count+1
Αποτελέσματα // count //
Τέλος οργάνωση_εκδήλωσης
```

2.9 Εναλλακτικές προσεγγίσεις

Οπως γίνεται φανερό, το κεφάλαιο αυτό είναι το πλέον βασικό για την εισαγωγή του μαθητή στις αλγοριθμικές έννοιες. Για το λόγο αυτό δίνουμε εδώ μερικά συ-

μπληρωματικά στοιχεία που μπορούν να φανούν χρήσιμα στην εκπαιδευτική διαδικασία.

Α) Εκχώρηση τιμών

Πρόβλημα 1. Δίδονται οι αριθμοί α, β και γ. Να υπολογιστεί η τιμή της παράστασης α(β+γ).

Με αναφορά στο πρόβλημα θα επεξηγηθεί στους μαθητές η έννοια της εντολής **εκχώρησης**. Είναι φανερό ότι για τον υπολογισμό της παράστασης αυτής, πρέπει να προηγηθεί η πρόσθεση και να ακολουθήσει ο πολλαπλασιασμός. Αυτός είναι άλλωστε ο σκοπός της εισαγωγής των παρενθέσεων στην **ένθετη** (infix) μορφή μίας παράστασης, προκειμένου να τροποποιείται η γνωστή ιεραρχία των πράξεων. Ετσι με χρήση μίας βοηθητικής μεταβλητής χ, γράφουμε την επόμενη εντολή εκχώρησης:

χ ← β+γ

Πρέπει να τονιστεί με έμφαση, ότι η εντολή αυτή εμπεριέχει μία δυναμική. Για το λόγο αυτό χρησιμοποιείται και αριστερό βέλος, προκειμένου να δείχνει τη φορά της εκχώρησης. Δεν πρόκειται για εξίσωση, παρόλο που σε άλλα βιβλία μπορεί να χρησιμοποιείται το σύμβολο ίσον "=" ή περιφραστικά η λέξη "θέσε" για τον ίδιο σκοπό. Ας σημειωθεί επίσης ότι οι διάφορες γλώσσες προγραμματισμού χρησιμοποιούν διάφορα σύμβολα για το σκοπό αυτό. Έτσι:

- Όλες οι εκδόσεις της BASIC, χρησιμοποιούν το ίσον και προαιρετικά τη λέξη-κλειδί LET, π.χ. [LET] x=a+b.
- Η C χρησιμοποιεί επίσης το ίσον.
- Η Pascal χρησιμοποιεί το συνδυασμό των χαρακτήρων ":=".Π.χ. x:=a+b.
- Η COBOL χρησιμοποιεί περιφραστικό τρόπο π.χ.

  ADD A TO B GIVING X

  ή το χαρακτήρα ίσον σε συνδυασμό με τη λέξη-κλειδί COMPUTE, π.χ.

  COMPUTE X=A+B

  Η προηγούμενη εντολή ακολουθείται από την

  y ← x*α

και με την ίδια έννοια η μεταβλητή γ προσλαμβάνει το τελικό αποτέλεσμα.

Προφανώς οι δύο εντολές εκχώρησης μπορούν να συνδυαστούν σε μία, την

$$\gamma \leftarrow \alpha*(\beta+\gamma)$$

και γενικά όσο πολύπλοκη και να είναι μία έκφραση, μπορεί να αποτελεί το δεξιό μέλος μίας και μόνο εντολής εκχώρησης.

Κατόπιν τούτων ο πλήρης αλγόριθμος είναι ο εξής:

```
Αλγόριθμος Πρόβλημα_1
Δεδομένα //α, β, γ//
γ ← α*(β+γ)
Αποτελέσματα // γ //
Τέλος Πρόβλημα_1
```

Βασιζόμενοι σε αυτό το πρώτο ολοκληρωμένο παράδειγμα αλγορίθμου σε ψευδογλώσσα, επεξηγούμε στους μαθητές τη χρήση των δύο πρώτων και δύο τελευταίων γραμμών του αλγορίθμου επισημαίνοντας ότι, δεν πρόκειται για εκτελεστέες εντολές, αλλά για τυποποιήσεις που οριοθετούν την αρχή και το τέλος του αλγορίθμου και περιγράφουν σαφώς τα δεδομένα και τα αποτελέσματα. Ας σημειωθεί και εδώ, ότι η γραμμή Δεδομένα μπορεί και να μην υπάρχει.

Σε αυτό το σημείο είναι σκόπιμο να επεξηγηθούν και μερικά στοιχεία της χρησιμοποιούμενης τυπογραφίας. Έτσι οι λέξεις-κλειδιά της ψευδογλώσσας στα βιβλία έχουν αποτυπωθεί με έντονα στοιχεία (**bold**) και διαφορετικό χρώμα. Ωστόσο στον πίνακα και στα τετράδια των μαθητών μπορούν να γράφονται υπογραμμισμένα.

Β) Ακολουθία – Επιλογή

Ο αλγόριθμος πρέπει να εκτελείται πάντοτε.

Πρόβλημα 2. Δίδονται οι αριθμοί α, β, γ και δ. Να βρεθεί η τιμή της παράστασης $\alpha(\beta+\gamma)/\delta$.

Στη συνέχεια παρουσιάζεται το πρόβλημα 2, το οποίο φαίνεται ως απλή παραλλαγή του προηγουμένου. Προτείνεται να τεθεί στην τάξη το ερώτημα "αν ο αλγόριθμος του προβλήματος 2 εκτελείται πάντοτε". Γίνεται συζήτηση για την περίπτωση $\delta=0$ και πως προλαμβάνεται. Παρουσιάζεται η εντολή επιλογής-απόφασης "Αν...τότε".

Καταλήγουμε στον επόμενο αλγόριθμο.

```
Αλγόριθμος Πρόβλημα_2.
Δεδομένα // α, β, γ, δ //
χ ← α*(β+γ)
Αν δ≠0 τότε y ← χ/δ
Αποτελέσματα // y //
Τέλος Πρόβλημα_2
```

Ο αλγόριθμος αυτός διαθέτει δύο εκτελέσιμες εντολές, την εντολή εκχώρησης και την εντολή Αν...τότε. Η σειρά εκτέλεσης των εντολών ενός αλγόριθμου είναι από πάνω προς τα κάτω και αποκαλείται **ακολουθία** (sequence). Έτσι πρώτα εκτελείται η εντολή εκχώρησης και στη συνέχεια η εντολή Αν...τότε.

Στην εντολή Αν...τότε είναι πιθανό όταν ισχύει η συνθήκη, να απαιτείται η εκτέλεση περισσότερων από μία εντολές. Στην περίπτωση αυτή οι διαδοχικές εντολές θα μπορούσε να γραφούν η μία μετά την άλλη στην ίδια γραμμή υιοθετώντας και ένα σύμβολο ως διαχωριστή εντολών. Είναι όμως προτιμότερο να γράφονται από κάτω και σε εσοχή, ενώ το σχήμα επιλογής κλείνει με τη λέξη Τέλος_αν. Π.χ.

```
Αν <συνθήκη> τότε
    <εντολή_1>
    <εντολή_2>
    ..............
    <εντολή_ν>
Τέλος_αν
```

Για λόγους ομοιομορφίας είναι αποδεκτό το ίδιο σχήμα και για την περίπτωση μίας εντολής, δηλ.

```
Αν <συνθήκη> τότε
    <εντολή>
Τέλος_αν
```

Ωστόσο στον αλγόριθμο Πρόβλημα_2 παραμένει ένα ανοικτό ζήτημα. Ποιο είναι το αποτέλεσμα στην περίπτωση δ=0;

Η αντιμετώπιση του ζητήματος αυτού επιδέχεται αρκετές λύσεις.

α) Με τη χρήση της εντολής Αν...τότε...αλλιώς εμφανίζουμε ένα σχετικό μήνυμα. Π.χ.

```
Αλγόριθμος Πρόβλημα_2α.
Δεδομένα // α, β, γ, δ //
χ ← α*(β+γ)
Αν δ≠0 τότε
    y ← χ/δ
αλλιώς
    Γράψε "Αδύνατο"
Τέλος_αν
Αποτελέσματα // y //
Τέλος Πρόβλημα_2α
```

β) Υιοθετούμε μία λογική μεταβλητή με την οποία καθορίζεται στα αποτελέσματα, αν υπάρχει λύση ή όχι. Π.χ.

```
Αλγόριθμος Πρόβλημα_2β.
Δεδομένα // α, β, γ, δ //
flag ← Αληθής
χ ← α*(β+γ)
Αν δ≠0 τότε
     y ← χ/δ
αλλιώς
    flag ← Ψευδής
Τέλος_αν
Αποτελέσματα // y, αν flag=αληθής, αλλιώς αδύνατο //
Τέλος Πρόβλημα_2β
```

γ) Παρόμοια με το (β), αλλά τα αποτελέσματα είναι δύο, το y και το flag. Π.χ.

```
Αλγόριθμος Πρόβλημα_2γ.
Δεδομένα // α, β, γ, δ //
flag ← Αληθής
χ ← α*(β+γ)
Αν δ≠0 τότε
     y ← χ/δ
αλλιώς
    flag ← Ψευδής
Τέλος_αν
Αποτελέσματα // y, flag //
Τέλος Πρόβλημα_2γ
```

Από τις παραπάνω τρεις εναλλακτικές προτάσεις, η πλέον προφανής για το μαθητή είναι η πρώτη. Άλλωστε είναι δυνατό μετά τον υπολογισμό του αποτελέσματος y, να ακολουθεί η εντολή "Γράψε y". Επίσης αντί για τη γραμμή Δεδομένα, να υπάρχει η εντολή "Διάβασε α, β, γ, δ". Ωστόσο χωρίς να απορρίπτουμε την προσέγγιση αυτή, προτιμότερη είναι η τρίτη πρόταση για τους εξής λόγους:

I) Ο αλγόριθμος λύνει το πρόβλημα και δεν ασχολείται με τον τρόπο εισαγωγής δεδομένων και παρουσίασης αποτελεσμάτων.

II) Αν ο αλγόριθμος υλοποιηθεί ως υποπρόγραμμα ή συνάρτηση, τότε είναι προτιμότερο να μην έχει εντολές εισόδου-εξόδου. Οι γραμμές Δεδομένα και Αποτελέσματα παραπέμπουν ακριβώς στο πέρασμα παραμέτρων στη διαδικασία.

Πρόβλημα 3. Δίδονται 3 αριθμοί x, y, z. Να βρεθεί ο μεγαλύτερος.

Με αναφορά αυτό το πρόβλημα γίνεται αντιπαραβολή του τρόπου λειτουργίας του ανθρώπινου εγκέφαλου (παράλληλη) σε σχέση με ένα "άλλο μυαλό" που δουλεύει σειριακά. Καταλήγουμε στη λύση συγκρίνοντας ανά δύο τους αριθμούς κ.λπ. Παρουσίαση του αλγόριθμου αυτού σε ελεύθερο κείμενο. Γενίκευση του προηγουμένου προβλήματος με χιλιάδες αριθμών. Διαπίστωση ότι ο αλγόριθμος "της μηχανής" δουλεύει, ενώ αυτός του μυαλού πρέπει να αλλάξει.

α)

```
Αλγόριθμος Max
Δεδομένα // x, y, z //
M ← x
Αν y>M τότε M ← y
Αν z>M τότε M ← z
Αποτελέσματα // M //
Τέλος Max
```

β)

```
Αλγόριθμος Max
Δεδομένα //x, y, z//
Αν y > x τότε
  Αν z > y τότε
    M ← z
  αλλιώς
    M ← y
  Τέλος_αν
Αλλιώς
  Αν z > x τότε
    M ← z
  αλλιώς
    M ← x
  Τέλος_αν
Τέλος_αν
Αποτέλεσμα //M//
Τέλος Max
```

γ)

```
Αλγόριθμος Max
Δεδομένα //x, y, z//
Αν x >= y και x >= z τότε
  M ← x
αλλιώς_αν y >= x και y >= z
  M ← y
αλλιώς
  M ← z
Τέλος_αν
Αποτέλεσμα //M//
Τέλος Max
```

Στη λύση (β) γίνεται η παρατήρηση ότι ως <εντολή> σε μία Αν...τότε...αλλιώς μπορεί να είναι και μία άλλη Αν...τότε. Αναφερόμαστε τότε σε **εμφωλευμένες** εντολές ελέγχου και συνιστούμε ότι καλύτερα να αποφεύγονται. Εναλλακτικά μπορεί να χρησιμοποιηθεί η δομή Αν...τότε...αλλιώς_αν, όπως έγινε στη λύση (γ).

Προσοχή. Υπάρχει διαφορά στη σύνταξη μεταξύ των εντολών Αν που χρησιμοποιήθηκαν στη (β) και (γ) λύση. Στις εμφωλευμένες εντολές Αν καθε μία κλείνει με Τέλος_αν, ενώ στην εντολή Αν...τότε...αλλιώς_αν υπάρχει μόνο ένα Τέλος_αν.

Γ) Επανάληψη

Ως τώρα ασχοληθήκαμε με μη επαναληπτικούς αλγορίθμους, που δεν παρουσιάζουν ιδιαίτερα προβλήματα κατανόησης από τους μαθητές. Στη συνέχεια θα παρουσιάσουμε την έννοια της επανάληψης, η οποία πρέπει να δοθεί με ιδιαίτερη προσοχή στα αρχικά στάδια.

Πρόβλημα 4. Να γραφεί αλγόριθμος που να εμφανίζει τους αριθμούς από 1 έως 100.

Μετά την εκφώνηση προτρέπουμε τα παιδιά να απαντήσουν βασισμένοι σε ότι προηγήθηκε. Η μόνη προφανής απάντηση είναι ότι θα πρέπει να γραφούν

Σημ. Το παράδειγμα θέλει να δείξει τη διαφορά στη χρήση των διαφόρων μορφών εντολών Αν... τότε. Προφανώς μπορούν να υλοποιηθούν και άλλοι αλγόριθμοι για τη λύση του ίδιου προβλήματος. Η σύνθετη συνθήκη της λύση (γ) αντικαθιστά την εμφωλευμένη Αν. Προτιμότερη λύση είναι η (α), γιατί αν προστεθεί και τέταρτος αριθμός, αρκεί να ενταχθεί μία ακόμη Αν. Επιπλέον η ίδια λογική μπορεί να χρησιμοποιηθεί στο πρόβλημα εύρεσης του μεγαλύτερου αριθμού σε πίνακα.

100 εντολές Γράψε "1", Γράψε "2", κ.λπ. πράγμα που είναι ασύμφορο και πρακτικά αδύνατο για μεγαλύτερους αριθμούς.

1η προσέγγιση. Εντολή Οσο...επανάλαβε

Παρατηρούμε ότι κάθε αριθμός παράγεται από τον προηγούμενό του με απλό τρόπο δηλ. προσθέτοντας κάθε φορά το 1. Μπορεί λοιπόν να χρησιμοποιηθεί μία μεταβλητή, έστω i, η οποία αρχίζει από το 1 και καταλήγει στο 100 αυξανόμενη κατά 1. Συχνά η μεταβλητή αυτή αποκαλείται **μετρητής** (counter). Η εκάστοτε αύξηση του μετρητή μπορεί να γίνει με χρήση της εντολής εκχώρησης

```
i ← i+1
```

Εδώ πρέπει να επισημανθεί στους μαθητές, ότι τίποτα δεν εμποδίζει να χρησιμοποιείται η ίδια μεταβλητή στο αριστερό και το δεξιό μέλος σε μία εντολή εκχώρησης. Ετσι στην προκειμένη περίπτωση η εντολή δρα ως εξής: *"η νέα τιμή της μεταβλητής i είναι η παλιά συν ένα"*.

Η αρχική τιμή του μετρητή ορίζεται εύκολα με την εντολή i ← 1. Το πρόβλημα εστιάζεται στο πότε και πώς θα γίνει αντιληπτό ότι φθάσαμε στην τελική τιμή 100. Μια λύση θα ήταν η χρήση της εντολής Αν i=100 τότε ..., αλλά η χρήση της μας οδηγεί αναπόφευκτα στη χρήση και της εντολής "πήγαινε" (goto), πράγμα που πρέπει να αποφευχθεί. Ωστόσο η εντολή **Οσο** <*συνθήκη*> **επανάλαβε** λύνει το πρόβλημα αυτόματα και έτσι καταλήγουμε στον επόμενο αλγόριθμο.

```
Αλγόριθμος Μέτρημα
i ← 1
Οσο i<= 100 επανάλαβε
    Γράψε i
    i ← i+1
Τέλος_επανάληψης
Τέλος Μέτρημα
```

Το τμήμα του αλγόριθμου που επαναλαμβάνεται, δηλ. από την εντολή Οσο μέχρι το Τέλος_επανάληψης αποκαλείται **βρόχος** (Προσοχή, όχι βρόγχος. Βρόχος=θηλιά, αγγλ. loop, γαλ. boucle, ενώ βρόγχος=πνευμόνι). Προφανώς ο βρόχος εκτελείται όσο η συνθήκη είναι αληθής.

Ο αλγόριθμος αυτός μπορεί να γραφεί με αρκετές παραλλαγές ανάλογα αν η αρχική τιμή του μετρητή είναι μηδέν ή αν η αύξηση του μετρητή προηγείται της ε-

ντολής Γράψε κ.λπ. Ο καθηγητής σε συνεργασία με τους μαθητές μπορεί να αποτυπώσει τις εναλλακτικές λύσεις.

2η προσέγγιση. Εντολή Για...από...μέχρι

Η προηγούμενη προσέγγιση της επαναληπτικότητας είναι η γενικότερη. Ο βρόχος εκτελείται όσο είναι αληθής η συνθήκη που έχει τεθεί. Ωστόσο αν ο αριθμός των φορών επανάληψης του βρόχου είναι γνωστός εκ των προτέρων, είναι προτιμότερο να χρησιμοποιείται η εντολή Για...από...μέχρι. Ετσι ο προηγούμενος αλγόριθμος γίνεται:

```
Αλγόριθμος Μέτρημα
Για i από 1 μέχρι 100
   Γράψε i
Τέλος_επανάληψης
Τέλος Μέτρημα
```

Οπως είναι φανερό στην εντολή αυτή εμπεριέχονται όλα τα στοιχεία που αφορούν το μετρητή, δηλ. αρχική τιμή, τελική τιμή και βήμα μεταβολής. Ο βρόχος τερματίζει αυτόματα όταν εκτελεστεί για την τελική τιμή.

Πρόβλημα 5. Να υπολογιστεί το άθροισμα 1+2+3+...+100.

Μια λύση που ενδεχόμενα μπορεί να προτείνει κάποιος μαθητής, είναι να χρησιμοποιηθεί ο τύπος του αθροίσματος αριθμητικής προόδου. Ωστόσο μπορεί να μην τον θυμόμαστε ή εν γένει να μην υπάρχει πάντα κάποιος τύπος. Εχοντας λύσει προηγούμενα την επαναληπτικότητα του αλγορίθμου, αρκεί στο βρόχο να προστεθούν κάποια νέα στοιχεία (εντολές) που να επιλύουν την επιπλέον απαίτησης της άθροισης.

Προς τούτο χρησιμοποιούμε μία μεταβλητή, έστω S στην οποία σε κάθε επαναληπτικό βήμα αθροίζουμε την τιμή της μεταβλητής i. Η σχετική εντολή εκχώρησης είναι:

$$S \leftarrow S + i$$

Η εντολή αυτή δρα ως εξής: *"η νέα τιμή του S είναι η παλιά συν την τιμή της μεταβλητής i"*. Συχνά η μεταβλητή S αποκαλείται **αθροιστής** λόγω του ρόλου που παίζει. Ενα σημείο που αξίζει προσοχής είναι ότι ο αθροιστής πρέπει πάντα να εκκινεί με κάποια αρχική τιμή (συνήθως μηδέν, αλλά όχι πάντα).

Ετσι ο αλγόριθμος γίνεται:

```
Αλγόριθμος Άθροισμα
S ← 0
Για i από 1 μέχρι 100
    S ← S + i
Τέλος_επανάληψης
Αποτελέσματα // S //
Τέλος Άθροισμα
```

Σχετικά με τον αλγόριθμο αυτό πρέπει να εξηγηθεί λεπτομερώς ο διαφορετικός ρόλος των μεταβλητών i και S. Προτείνεται ο καθηγητής να δημιουργήσει στον πίνακα δύο παραλληλόγραμμα, ένα για το S και ένα για το i, να τοποθετηθεί σε κάθε ένα η αρχική τιμή και να εκτελεστεί ο αλγόριθμος "με το χέρι" γράφοντας σε κάθε επανάληψη τις διαδοχικές τιμές που λαμβάνουν οι μεταβλητές αυτές, δηλ. 1, 2, 3, 4 ... για το i και 0, 1, 3, 6, 10 ... για το S.

Οι παραπάνω αλγόριθμοι γενικεύονται εύκολα για τιμή μέχρι N (αντί 100), αρκεί να τεθεί η γραμμή Δεδομένα // N // και προφανώς η τελική τιμή εκτέλεσης του βρόχου να είναι N.

Πρόβλημα 6. Να βρεθεί το N παραγοντικό.

Το πρόβλημα αυτό μπορεί να δοθεί ως άσκηση αμέσως μετά από το προηγούμενο. Υπενθυμίζεται ότι N!=1.2.3...(N-1).N και προτείνεται να χρησιμοποιηθεί η μεταβλητή P για την υποδοχή του γινομένου. Οι μαθητές κατά πάσα πιθανότητα θα δημιουργήσουν αλγόριθμο με αρχική τιμή του P=0. Συνιστάται να τους προτρέψουμε να εκτελέσουν τον αλγόριθμο στο χέρι, προκειμένου να εντοπίσουν μόνοι τους το λάθος.

```
Αλγόριθμος N_Παραγοντικό
Δεδομένα // N //
P ← 1
Για i από 1 μέχρι N
    P ← P * i
Τέλος_επανάληψης
Αποτελέσματα // P //
Τέλος N_Παραγοντικό
```

Με την ευκαιρία του προβλήματος αυτού καλό είναι να γίνει ένα σχόλιο σχετικά με τη μέγιστη τιμή του N. Είναι γνωστό ότι το N! αυξάνεται πολύ γρήγορα. Και ενώ στο επίπεδο του αλγορίθμου το θέμα αυτό δεν μας απασχολεί, θα μας απασχολήσει σίγουρα στην υλοποίηση.

Πρόβλημα 7. Να βρεθεί το άθροισμα 1+3+5+....+99.

Με το πρόβλημα αυτό παρουσιάζεται στους μαθητές η έννοια του βήματος, η οποία ήταν προφανής στα προβλήματα 4, 5 και 6 και μπορεί να διαφύγει της προσοχής τους. Προτείνουμε στους μαθητές να επιφέρουν τις σχετικές τροποποιήσεις των αλγορίθμων. Η εντολή Για...από...μέχρι τώρα γίνεται:

Για i **από** 1 **μέχρι** 99 **με_βήμα** 2

και αναδεικνύεται η χρησιμότητά της σε ανάλογα προβλήματα.

Ας σημειωθεί ότι οι τιμές από, μέχρι, βήμα μπορούν να είναι οποιεσδήποτε εφ όσον ισχύουν οι επόμενοι περιορισμοί.

- από <= μέχρι με βήμα > 0
- από >= μέχρι με βήμα < 0
- βήμα ≠ 0

Ακόμη μπορούν να λάβουν πραγματικές τιμές, όπως αναφέρεται και στο βιβλίο μαθητή. Ο επόμενος αλγόριθμος εμφανίζει διαδοχικές τιμές μιας συνάρτησης.

```
Αλγόριθμος Συνάρτηση
Για x από -10 μέχρι 10 με_βήμα 0.01
    y ← x^2 - 5*x + 6
    Γράψε x, y
Τέλος_επανάληψης
Τέλος Συνάρτηση
```

Πρόβλημα 8. Να διαβαστούν δύο θετικοί αριθμοί a και b, με a<b.

Στο βιβλίο μαθητή αναφέρεται ότι η εντολή Αρχή_επανάληψης...μέχρις_ότου θα εκτελεστεί οπωσδήποτε μια φορά. Αυτό μπορεί να εκληφθεί από τους μαθητές ως μειονέκτημα. Ωστόσο υπάρχουν πολλές περιπτώσεις που μια σειρά εντολών πρέπει να εκτελεστούν τουλάχιστον μια φορά, όπως στο παράδειγμα αυτό.

```
Αλγόριθμος Εισαγωγή
Αρχή_επανάληψης
    Διάβασε a
    Διάβασε b
Μέχρις_ότου a<b και a>0 και b>0
```

Τέλος Εισαγωγή

Ο αλγόριθμος αυτός είναι ένα παράδειγμα εισαγωγής τιμών με έλεγχο εγκυρότητας (data validation). Η λύση αυτή μειονεκτεί στο γεγονός ότι όλοι οι έλεγχοι εγκυρότητας συνοδεύουν την εντολή μέχρις_ότου. Σε πιο πολύπλοκες περιπτώσεις ελέγχου πρέπει να χρησιμοποιηθεί ξεχωριστή διαδικασία ελέγχου. Σε απλές περιπτώσεις, όπως εδώ, η λύση αυτή είναι η πλέον ενδεδειγμένη.

Μερικές ακόμη παρατηρήσεις

Η χρησιμοποιούμενη ψευδογλώσσα είναι η πλέον ενδεδειγμένη για την παράσταση αλγορίθμων διεθνώς. Η χρήση της συνιστάται για κάθε περίπτωση αλγορίθμου ανεξάρτητα από το χρησιμοποιούμενο βιβλίο ή γλώσσα προγραμματισμού.

Οπως θα παρατήρησε ήδη ο αναγνώστης, στο ρεπερτόριο των εντολών της ψευδογλώσσας δεν υπάρχουν δηλώσεις τύπων δεδομένων των χρησιμοποιούμενων μεταβλητών. Κάτι τέτοιο δεν είναι απαραίτητο στο επίπεδο της αλγοριθμικής ψευδογλώσσας. Για παράδειγμα ένας αλγόριθμος που βρίσκει τον ελάχιστο ν αριθμών, είναι ο ίδιος είτε οι αριθμοί ειναι ακέραιοι, είτε πραγματικοί. Αν από τα δεδομένα του προβλήματος πρέπει να δηλωθεί ο τύπος των μεταβλητών, τότε αυτό γίνεται στην εκφώνηση. Ωστόσο δεν είναι λάθος στις περιπτώσεις αυτές να συμπληρώνεται ο τύπος των μεταβλητών στη γραμμή των δεδομένων.

Μπορούμε επίσης να δεχθούμε και τις συνήθεις μαθηματικές τυπογραφικές συμβάσεις για την αποτύπωση τύπων. Για παράδειγμα είναι αποδεκτό να γραφεί είτε a^2 είτε a^2, όπως επίσης a(i,j) ή a[i,j] ή a_{ij} καθώς και το σύμβολο της τετραγωνικής ρίζας.

Αποδεκτός είναι επίσης και ο περιφραστικός τρόπος για ορισμένες ενέργειες, όπως για παράδειγμα "Θέσε Α=0" ή "Αν Χ είναι άρτιος τότε ...". Προφανώς αυτές οι "ελευθερίες" παύουν στον προγραμματισμό είτε στη ΓΛΩΣΣΑ είτε σε οποιαδήποτε γλώσσα προγραμματισμού.

Το πιο σημαντικό θέμα για τη σωστή γραφή των αλγορίθμων σε ψευδογλώσσα είναι όλες οι δομές ελέγχου (εκτός από την απλή Αν...τότε) να κλείνουν με Τέλος_αν, καθώς και όλες οι δομές επανάληψης να κλείνουν με Τέλος_επανάληψης ή Μέχρις_ότου...Ολες οι εντολές που αποτελούν το σώμα ενός βρόχου, πρέπει να γράφονται σε εσοχή. Αν αυτό δεν συμβαίνει, δεν είναι λάθος για την ορθότητα του

αλγορίθμου, αλλά συμβάλει σημαντικά στην αναγνωσιμότητα και κατανόηση, γι' αυτό ο καθηγητής πρέπει να επιμένει στη σωστή γραφή.

Ενα άλλο σημείο είναι η χρήση των διαγραμμάτων ροής. Τα διαγράμματα αυτά (που αποκαλούνται κακώς ακόμη και σήμερα "λογικά διαγράμματα") έχουν εγκαταληφθεί εδώ και χρόνια, γιατί ενθαρύνουν το μη δομημένα προγραμματισμό (βλ. και J.Martin, Diagramming Techniques for Analysts and Programmers, Prentice Hall, 1985) Τα διαγράμματα ροής εντάχθηκαν στο βιβλίο κύρια για ιστορικούς λόγους και επειδή συμπεριλαμβάνονται στο πρόγραμμα. Καλό είναι η χρήση τους να περιοριστεί για την επεξήγηση των βασικών εννοιών, όπως γίνεται στο κεφ. 2 το βιβλίου.

Κεφάλαιο 3

Δομές Δεδομένων και Αλγόριθμοι

3.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να γίνει κατανοητό στο μαθητή ότι τα δεδομένα αποθηκεύονται στη μνήμη (κύρια ή δευτερεύουσα) του υπολογιστή με κάποια συγκεκριμένη μέθοδο, δηλαδή με τη βοήθεια κάποιας δομής. Επίσης, κάθε δομή δεδομένων συνοδεύεται απαραιτήτως από κάποιες συγκεκριμένες λειτουργίες (δηλαδή πράξεις).

3.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να διακρίνουν τις διάφορες δομές δεδομένων, όπως πίνακας, στοίβα, ουρά κοκ.
- να αντιλαμβάνονται ποιές είναι οι πράξεις που συνοδεύουν κάθε δομή
- να μπορούν να αναπτύξουν αλγοριθμικά απλές αναδρομικές συναρτήσεις
- να εφαρμόσουν μεθόδους ταξινόμησης σε τυχαία σύνολα δεδομένων
- να αντιληφθούν την ιδιαιτερότητα των δυναμικών δομών της λίστας και του δένδρου.

3.3 Οδηγίες – επισημάνσεις

Είναι απαραίτητο να εξηγηθεί ότι δεν υπάρχει μία αδιαφιλονίκητη δομή που να χρησιμοποιείται σε κάθε εφαρμογή, αλλά η δομή επιλέγεται με βάση τις λειτουργίες που πρέπει να εκτελούνται. Θα πρέπει να δοθεί ιδιαίτερη έμφαση στη διάκριση μεταξύ των διαφόρων αλγορίθμων ταξινόμησης, επειδή σε επόμενο κεφάλαιο θα γίνει αναφορά στην ανάλυση της πολυπλοκότητάς τους. Επίσης προσοχή χρειάζεται στην παράγραφο των λιστών και των δένδρων που στηρίζονται στην έννοια της δυναμικότητας και υλοποιούνται με τη βοήθεια δεικτών.

3.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

τρία (3) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- Εμβάθυνση στις έννοιες «δεδομένο», «δομή δεδομένων» και «λειτουργία» επί μίας δομής
- Επεξήγηση της διαλεκτικής σχέσης μεταξύ δομών και αλγορίθμων
- Κατηγοριοποίηση των δομών και έμφαση στη δομή των πινάκων

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες

Περιεχόμενα θεωρητικής παρουσίασης

- Κάλυψη παραγράφων 3.1 μέχρι και 3.3
- Διαφορά μεταξύ δεδομένου και πληροφορίας
- Ορισμός της Πληροφορικής επιστήμης σε συνάρτηση με την έννοια των δεδομένων
- Ορισμός δομής δεδομένων
- Απαρίθμηση λειτουργιών επί των δομών
- Συσχέτιση δομών και αλγορίθμων
- Κατηγοριοποίηση δομών
- Χαρακτηριστικά δομής πίνακα

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα των παραγράφων 3.1 μέχρι 3.3 από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να διδαχθούν δύο από τα παραδείγματα 1, 2, 3, 4 από το τετράδιο του μαθητή. Να δοθεί στους μαθητές προς λύση η δραστηριότητα ΔΤ1 στην τάξη και μία από τις δραστηριότητες ΔΣ1 ή ΔΣ2 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Αποτελεί δεδομένο ότι το ύψος ενός ατόμου είναι 1,90. Πληροφορία είναι ότι το άτομο αυτό είναι ψηλό.
2. Κάθε δομή μπορεί να χρησιμοποιηθεί σε οποιοδήποτε πρόβλημα ή εφαρμογή
3. Δυναμικές είναι οι δομές που αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
4. Ένας πίνακας έχει σταθερό μέγεθος αλλά μεταβαλλόμενο περιεχόμενο
5. Ένας πίνακας μπορεί να αποθηκεύσει και ακεραίους αλλά και πραγματικούς αρθμούς

Απαντήσεις τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1: σωστό	2: λάθος	3: λάθος
4: σωστό	5: λάθος	

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

- Εμβάθυνση στη δομή της στοίβας και των λειτουργιών της
- Εμβάθυνση στη δομή της ουράς και των λειτουργιών της
- Εξέταση της σειριακής αναζήτησης

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες

Περιεχόμενα θεωρητικής παρουσίασης

- Κάλυψη παραγράφων 3.4 μέχρι και 3.6
- Επεξήγηση της δομής της στοίβας και της λογικής LIFO
- Συσχέτιση λογικής ουράς και στοίβας
- Αναφορά στη σειριακή αναζήτηση εισάγοντας ακροθιγώς την έννοια της αποτελεσματικότητας

Περιεχόμενα πρακτικής εφαρμογής

- Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα των αντίστοιχων παραγράφων από το βιβλίο του μαθητή.

- Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθεί στους μαθητές προς λύση η δραστηριότητα ΔΤ6 στην τάξη και η δραστηριότητα ΔΣ4 στο σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Μία ουρά διατηρεί τα δεδομένα ταξινομημένα ως προς τη σειρά άφιξής τους.
2. Η υλοποίηση της ουράς χρησιμοποιεί μία μόνο μεταβλητή-δείκτη για τη διαχείριση των εισαγωγών/διαγραφών, όπως και στην περίπτωση της στοίβας.
3. Όταν ψάχνουμε σε ένα τηλεφωνικό κατάλογο χρησιμοποιούμε τη σειριακή μέθοδο αναζήτησης.

Άσκηση

Έστω ότι σε μία στοίβα εισάγονται τρία στοιχεία με τη σειρά: πρώτα το Α, μετά το Β και τέλος το Γ, τα οποία αργότερα εξάγοναι. Όμως, οι εξαγωγές αυτές μπορούν να συμβούν οποτεδήποτε μεταξύ ή μετά τις εισαγωγές. Για παράδειγμα, μπορεί να συμβεί η αλληλουχία: εισαγωγή Α, εξαγωγή Α, εισαγωγή Β, εξαγωγή Β, εισαγωγή Γ, εξαγωγή Γ. Έτσι στην έξοδο θα λάβουμε με τη σειρά τη διάταξη των στοιχείων Α, Β, Γ. Αν όμως, εκτελεσθούν με τη σειρά οι πράξεις εισαγωγή Α, εξαγωγή Α, εισαγωγή Β, εισαγωγή Γ, εξαγωγή Γ, εξαγωγή Β, τότε στην έξοδο θα λάβουμε με τη σειρά τη διάταξη των στοιχείων Α, Γ, Β. Η ερώτηση είναι αν θεωρήσουμε όλους τους τρόπους αλληλοδιαδοχής πράξεων εισαγωγής/εξαγωγής, πόσες και ποιές είναι οι δυνατές διατάξεις των στοιχείων Α, Β, Γ στην έξοδο;

Απαντήσεις τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1: σωστό	2: λάθος	3: λάθος

Άσκηση

Ο αριθμός των διατάξεων 3 στοιχείων είναι 3!=6. Οι διατάξεις αυτές είναι ΑΒΓ, ΑΓΒ, ΒΑΓ, ΒΓΑ, ΓΑΒ και ΓΒΑ. Από την εκφώνηση είναι γνωστό ότι οι δύο πρώτες διατάξεις είναι δυνατόν να παραχθούν. Υπάρχουν άλλες τρεις διατάξεις που μπορούν να παραχθούν στην έξοδο

- εισαγωγή Α, εισαγωγή Β, εξαγωγή Β, εξαγωγή Α, εισαγωγή Γ, εξαγωγή Γ, οπότε στην έξοδο λαμβάνουμε τη διάταξη ΒΑΓ
- εισαγωγή Α, εισαγωγή Β, εξαγωγή Β, εισαγωγή Γ, εξαγωγή Γ, εξαγωγή Α, οπότε λαμβάνουμε τη διάταξη ΒΓΑ
- εισαγωγή Α, εισαγωγή Β, εισαγωγή Γ, εξαγωγή Γ, εξαγωγή Β, εξαγωγή Α, οπότε λαμβάνουμε τη διάταξη ΓΒΑ

αλλά δεν υπάρχει αλληλουχία εισαγωγών/εξαγωγών που να οδηγεί στη διάταξη ΓΑΒ.

Σχέδιο 3ου μαθήματος

Διδακτικοί στόχοι

- Εμβάθυνση στην έννοια της αναδρομής
- Παρουσίαση των δυναμικών δομών της λίστας και του δένδρου

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες

Περιεχόμενα θεωρητικής παρουσίασης

- Κάλυψη παραγράφων 3.7 μέχρι και 3.9
- Επεξήγηση της ταξινόμησης ως μίας βασικής λειτουργίας, που μπορεί να εφαρμοσθεί σε πίνακες
- Ανάπτυξη των μειονεκτημάτων και των πλεονεκτημάτων της αναδρομής σε σχέση με το χρόνο εκτέλεσης του προγράμματος και το χρόνο ανάπτυξης από τον προαγραμματιστή.

- Σύντομη παρουσίαση των δυναμικών δομών που απαιτούν χρήση μεταβλητών δεικτών που προσφέρονται από τις σύγχρονες γλώσσες προγραμματισμού.

Περιεχόμενα πρακτικής εφαρμογής

- Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα των αντίστοιχων παραγράφων από το βιβλίο του μαθητή

- Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθεί στους μαθητές προς λύση η δραστηριότητα ΔΤ6 στην τάξη και μία από τις δραστηριότητες ΔΣ3 και ΔΣ4 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Η ταξινόμηση είναι χρήσιμη διαδικασία γιατί έτσι εκτελείται γρηγορότερα η αναζήτηση.
2. Η ταξινόμηση ευθείας ανταλλαγής είναι πολύ αποτελεσματική σε πίνακες που είναι ταξινομημένοι κατά την αντίστροφη φορά σε σχέση με την επιθυμητή.
3. Η ταξινόμηση ευθείας ανταλλαγής είναι πολύ αποτελεσματική αν ο πίνακας περιέχει ίσα κλειδιά.
4. Όταν δεν είναι ιδιαίτερα κρίσιμος ο χρόνος απόκρισης μίας εφαρμογής, τότε μπορεί ο αντίστοιχος αλγόριθμος να είναι αναδρομικός.
5. Η αναδρομή στηρίζεται στη δομή της ουράς.
6. Σε ένα πίνακα δεν υπάρχει τρόπος διάκρισης της φυσικής σειράς αποθήκευσης από τη λογική αλληλουχία των στοιχείων.
7. Βασική λειτουργία σε μία δομή λίστας είναι η προσπέλαση σε τυχαίο κόμβο της δομής.

8. Η ρίζα ενός δένδρου δεν είναι παιδί κάποιου άλλου κόμβου.

Απαντήσεις τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1: σωστό	2: λάθος	3: σωστό
4: σωστό	5: λάθος	6: λάθος (χρήση δεικτών)
7: λάθος	8: σωστό	

3.5 Προβληματισμοί και θέματα προς συζήτηση

Θέματα προβληματισμού και συζήτησης μπορούν να προκύψουν από όλες τις δραστηριότητες που προτείνονται στο συγκεκριμένο κεφάλαιο. Συγκεκριμένα μπορούν να προταθούν συζητήσεις σχετικά με:

- Υπάρχει κάποια δομή μεταξύ των δομών που εξετάσθηκαν, η οποία να βρίσκει εύκολα το στοιχείο με τη μικρότερη τιμή; Η απάντηση είναι αρνητική. Για μία τέτοια περίπτωση υπάρχει μία ιδιαίτερη δενδρική δομή, που ονομάζεται σωρός (heap) και έχει αρκετά πολύπλοκους αλγορίθμους εισαγωγής και διαγραφής.
- Οι αλγόριθμοι αναζήτησης που εξετάσθηκαν είναι αρκετά αποτελεσματικοί ή μπορούν να σχεδιασθούν ακόμη καλύτεροι; Η απάντηση σχετίζεται με το περιεχόμενο του επομένου κεφαλαίου.

3.6 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

3.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

1: Δες παράγραφο 3.1

2: Δες παράγραφο 3.1

3: Δες παράγραφο 3.2

4: Δες παράγραφο 3.2

5: Δες παράγραφο 3.2

6: Δες παράγραφο 3.2

7: Δες παράγραφο 3.3

8: Δες παράγραφο 3.4

9: Δες παράγραφο 3.4

10: Δες παράγραφο 3.5

11: Δες παράγραφο 3.5

12: Δες παράγραφο 3.8

13: Ακολουθεί ένας αναδρομικός αλγόριθμος γαι τον υπολογισμό του a^b, όπου το a είναι πραγματικός και το b ακέραιος.

```
Αλγόριθμος Δύναμη
Δεδομένα // a, b //
Αν b = 0 τότε
    power ← 1
αλλιώς
    Αν (b MOD 2) = 0 τότε
        power ← Δύναμη(a*a, b DIV 2)
    αλλιώς
        power ← Δύναμη(a*a,b DIV 2)*a
    Τέλος_Αν
Τέλος_Αν
Αποτελέσματα // power //
Τέλος Δύναμη
```

14: Δες παράγραφο 3.6

15: Δες παράγραφο 3.6

16: Δες παράγραφο 3.7

17: Δες παράγραφο 3.7

3.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

ΔΤ 1

Θεωρούμε τον πίνακα table με ακεραίους που αντιστοιχούν στις ηλικίες των παιδιών. Οι μεταβλητές min_age και max_age αρχικοποιούνται με την τιμή της πρώτης θέσης του πίνακα, ενώ οι μεταβλητές min και max χρησιμοποιούνται για τη εύρεση της ταυτότητας των δύο παιδιών αντίστοιχα.

```
Αλγόριθμος Κατασκήνωση
Δεδομένα // table//
min_age ← table[1]
max_age ← table[1]
min ← 1
max ← 1
Για i από 2 μέχρι 300
    Αν table[i]<min_age τότε
        min_age ← table[i]
        min ← i
    Τέλος_αν
    Αν table[i]>max_age τότε
        max_age ← table[i]
        max ← i
    Τέλος_αν
Τέλος_επανάληψης
Εκτύπωσε min, min_age, max, max_age
Τέλος Κατασκήνωση
```

ΔΤ 2

Η εκφώνηση ουσιαστικά αναφέρεται στην παραλλαγή της ταξινόμησης φυσαλίδας που αντιλαμβάνεται πότε ο πίνακας έχει ουσιαστικά ταξινομηθεί και αποφεύγει τα περιττά περάσματα (που σε συνολικό αριθμό είναι n-1). Αυτό επιτυγχάνεται με τη βοήθεια μίας λογικής μεταβλητής, (μίας σημαίας flag), που πριν κάθε πέρασμα αρχικοποιείται ως ψευδής και αλλάζει ως αληθής αν σε κάποιο πέρα-

σμα γίνει έστω και μία ανταλλαγή. Έτσι, αν σε κάποιο πέρασμα δεν εκτελεσθεί καμία ανταλλαγή, τότε η σημαία παραμένει ψευδής και αυτομάτως τελειώνει ο αλγόριθμος.

```
Αλγόριθμος Φυσσαλίδα2
Δεδομένα // table //
Αρχή_επανάληψης
    flag ← Ψευδής
    Για i από 1 μέχρι n-1
        Αν table[i+1] < table[i] τότε
            αντιμετάθεσε(table[i+1], table[i])
            flag ← Αληθής
        Τέλος_αν
    Τέλος_επανάληψης
Μέχρις_ότου flag=Ψευδής
Αποτελέσματα // table //
Τέλος Φυσσαλίδα2
```

ΔΤ 3

Στη συνέχεια δίνονται οι αλγόριθμοι ώθησης (push) και απώθησης (pop) από στοίβα. Χρησιμοποιείται μία λογική μεταβλητή, η σημαία done, που δηλώνει την επιτυχή εκτέλεση της διαδικασίας. Επίσης, η μεταβλητή top δηλώνει την επάνω θέση της στοίβας που είναι κατειλημμένη από κάποιο στοιχείο. Τα δεδομένα είναι αποθηκευμένα σε ένα μονοδιάστατο πίνακα, που ονομάζεται stack και έχει μέγεθος size. Η μεταβλητή item χρησιμεύει για την αποθήκευση του στοιχείου που εισάγεται ή εξάγεται.

```
Αλγόριθμος Ωθηση
Δεδομένα // top, item //
Αν top<size τότε
    top ← top+1
    stack[top] ← item
    done ← Αληθής
αλλιώς
    done ← Ψευδής
Τέλος_αν
Αποτελέσματα // top, done //
Τέλος Ωθηση
```

```
Αλγόριθμος Απώθηση
Δεδομένα // top //
Αν top<=1 τότε
    item ← stack[top]
    top ← top-1
    done ← Αληθής
αλλιώς
    done ← Ψευδής
Τέλος_αν
Αποτελέσματα // item, top, done //
Τέλος Απώθηση
```

Στον προηγούμενο αλγόριθμο ελέγχεται η συνθήκη «top<=1» ώστε να διαπιστωθεί αν η στοίβα έχει τουλάχιστον ένα στοιχείο. Προφανώς, αν δεν έχει τότε επιστρέφεται το μήνυμα Ψευδής.

ΔΤ 4

Οι αλγόριθμοι εισαγωγής και εξαγωγής από ουρά δίνονται στη συνέχεια. Χρησιμοποιείται μία λογική μεταβλητή, η σημαία done, που δηλώνει την επιτυχή εκτέλεση της διαδικασίας. Επίσης, οι μεταβλητές rear και front δηλώνουν τους δύο δείκτες που δείχνουν αντίστοιχα στην τελευταία θέση και στην πρώτη θέση της ουράς, που είναι ένας πίνακας queue μεγέθους size. Η μεταβλητή item χρησιμεύει για την αποθήκευση του στοιχείου που εισάγεται ή εξάγεται.

```
Αλγόριθμος Εισαγωγή_σε_Ουρά
Δεδομένα // rear, item //
Αν rear < size τότε
    rear ← rear+1
    queue[rear] ← item
    done ← Αληθής
αλλιώς
    done ← Ψευδής
Τέλος_αν
Αποτελέσματα // rear, done //
Τέλος Εισαγωγή_σε_Ουρά

Αλγόριθμος Εξαγωγή_από_Ουρά
Δεδομένα // rear, item //
```

```
Αν rear <= front τότε
    front ← front+1
    item ← queue[front]
    done ← Αληθής
αλλιώς
    done ← Ψευδής
Τέλος_αν
Αποτελέσματα // item, rear, done //
Τέλος Εξαγωγή_από_Ουρά
```

Στον προηγούμενο αλγόριθμο η συνθήκη «rear<=front» ελέγχει αν η ουρά περιέχει στοιχεία. Η ισότητα ισχύει στην περίπτωση που η ουρά περιέχει μόνο ένα στοιχείο.

ΔΤ 5

Η άσκηση αυτή είναι σχετικά σύνθετη επειδή αποτελείται από δύο σκέλη. Σε πρώτη φάση πρέπει να ταξινομηθούν αλφαβητικά τα επίθετα των μαθητών και κατόπιν πρέπει τα επίθετα αυτά να εισαχθούν σε μία ουρά. Η άσκηση επιλύεται με τη βοήθεια του αλγορίθμου ταξινόμησης της παραγράφου 3.7 ή του αλγορίθμου της προηγούμενης ΔΤ2.

ΔΤ 6

Αποτελεί απλή παραλλαγή της μεθόδου ταξινόμησης ευθείας ανταλλαγής, που περιγράφεται στην παράγραφο 3.7. Η διαφορά έγκειται στο ότι στον εξωτερικό βρόχο δεν πρέπει να γίνουν n-1 αλλά 10 επαναλήψεις, ώστε να απομονωθούν στην κορυφή του πίνακα οι 10 μικρότερες ζητούμενες τιμές.

➤ Στο σπίτι

ΔΣ 1

Θα θεωρήσουμε ένα δισδιάστατο πίνακα currency μεγέθους 5x2, όπου κάθε γραμμή του πίνακα αντιστοιχεί σε ένα νόμισμα (όπως τα νομίσματα αναφέρονται στην εκφώνηση), ενώ οι δύο στήλες του πίνακα αντιστοιχούν στην τιμή αγοράς ή πώλησης κάθε νομίσματος. Για λόγους ευκολίας (αποφυγή χρήσης συμβολοσειρών) χρησιμοποιούνται οι ακέραιες μεταβλητές nomisma1 και nomisma2, που

λαμβάνουν τιμές από 1 μέχρι 6. Οι τιμές αυτές αντιστοιχούν στα νομίσματα της εκφώνησης με έκτο νόμισμα τη δραχμή. Επίσης, η πραγματική μεταβλητή poso1 αντιστοιχεί στο ποσό των νομισμάτων nomisma1 που δίνει ο πελάτης, και η μεταβλητή poso2 στο ποσό των νομισμάτων nomisma2 που προκύπτουν από τη μετατροπή.

```
Αλγόριθμος Μετατροπές_νομισμάτων
Δεδομένα // currency //
Διάβασε nomisma1,nomisma2,poso1
Αν nomisma1=6 τότε
   poso2 ← poso1 / currency[nomisma2, 2]
   Δεδομένα //scores//
αλλιώς
   poso2 ← poso1 * currency[nomisma1,1]
Τέλος_αν
Εκτύπωσε poso2
Τέλος Μετατροπές_νομισμάτων
```

ΔΣ 2

Θα θεωρήσουμε ένα δισδιάστατο πίνακα scores μεγέθους 5x5, όπου κάθε γραμμή του πίνακα αντιστοιχεί σε ένα παίκτη, ενώ κάθε στήλη του πίνακα αντιστοιχεί σε έναν αγώνα. Τους συνολικούς πόντους που πέτυχαν οι παίκτες, τους αποθηκεύουμε σε ένα μονοδιάστατο πίνακα 5 θέσεων που ονομάζεται sum. Η μεταβλητή first δηλώνει τον αθλητή που επέτυχε συνολικά τους περισσότερους πόντους (max).

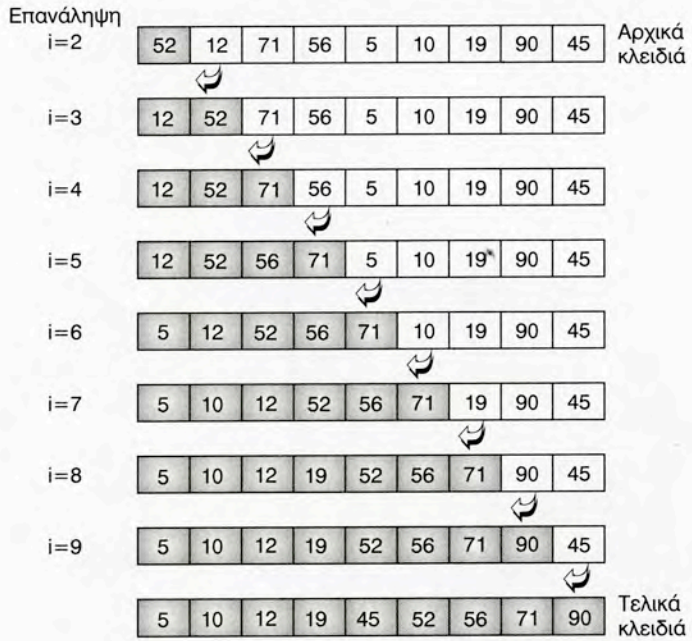
```
Αλγόριθμος Στατιστικά
Δεδομένα // scores //
Για i από 1 μέχρι 5
   sum[i] ← 0
Τέλος_επανάληψης
Για i από 1 μέχρι 5
   Για j από 1 μέχρι 5
      sum[i] ← sum[i]+ scores[i,j]
   Τέλος_επανάληψης
Τέλος_επανάληψης
max ← sum[1]
first ← 1
Για i από 2 μέχρι 5
```

```
    Αν scores[i] > max τότε
        max ← scores[i]
        first ← i
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // first //
Τέλος Στατιστικά
```

ΔΣ 3

Ουσιαστικά πρόκειται για τον αλγόριθμο ταξινόμησης ευθείας εισαγωγής straight insertion sort). Ο αλγόριθμος αυτός είναι ιδανικός για περιπτώσεις δεδομένων που είναι «περίπου» ταξινομημένα και χρησιμοποιείται σε πολλά υβριδικά σχήματα. Σύμφωνα με τον αλγόριθμο αυτό τα στοιχεία διακρίνονται σχηματικά σε μία *ακολουθία προορισμού* (destination sequence) table[1], table[2], ..., table[i-1] και σε μία *ακολουθία πηγής* (source sequence) table[i], ..., table[n]. Αρχικά η ακολουθία προορισμού αποτελείται από ένα στοιχείο, το πρώτο, και σταδιακά μεγαλώνει κατά ένα. Αυτό επιτυγχάνεται θεωρώντας το πρώτο στοιχείο της ακολουθίας πηγής και παρεμβάλοντάς το στην κατάλληλη θέση μεταξύ των στοιχείων της ακολουθίας προορισμού εκτελώντας διαδοχικές συγκρίσεις από τα δεξιά προς τα αριστερά των με τα στοιχεία της ακολουθίας προορισμού. Ο σχετικός αλγόριθμος δίνεται στη συνέχεια.

```
Αλγόριθμος Ευθεία_Εισαγωγή
Δεδομένα // table //
Για i από 2 μέχρι n
    temp ← table[i]
    table[0] ← temp
    j ← i-1
    Οσο temp<table[j] επανάλαβε
        table[j+1] ← table[j]
        j ← j-1
    Τέλος_επανάληψης
    table[j+1] ← temp
Τέλος_επανάληψης
Αποτελέσματα // table //
Τέλος Ευθεία_Εισαγωγή
```



Σχήμα 3.1. *Ταξινόμηση ευθείας εισαγωγής.*

Παράδειγμα. Έστω ότι ο αρχικός πίνακας αποτελείται από τα εννέα κλειδιά 52, 12, 71, 56, 5, 10, 19, 90 και 45. Στο προηγούμενο σχήμα παρουσιάζεται η διαδικασία ταξινόμησης θεωρώντας τον πίνακα αυτό. Κάθε φορά η ακολουθία προορισμού εμφανίζεται με σκίαση, ενώ το πρώτο στοιχείο της ακολουθίας πηγής αναδεικνύεται από το αντίστοιχο βέλος. Το στοιχείο αυτό λαμβάνει την κατάλληλη θέση μέσα στην ακολουθία προορισμού "σπρώχνοντας" μερικά στοιχεία προς τα δεξιά. Η εύρεση της κατάλληλης θέσης γίνεται εύκολα με διαδοχικές συγκρίσεις και μετακινήσεις. Λόγου χάριν, στην πέμπτη σειρά το στοιχείο 10 συγκρίνεται διαδοχικά με τα στοιχεία 71, 56, 52, 12 και 5, οπότε γίνεται αντιληπτό ότι το 10 πρέπει να παρεμβληθεί μεταξύ των 5 και 12. Για να γίνει αυτό τα στοιχεία 12 ως και 71 μετακινούνται μία θέση προς τα δεξιά για να δημιουργηθεί μία κενή θέση για το 10.

Μερικά σχόλια σχετικά με την υλοποίηση (για παράδειγμα σε Pascal) του προηγούμενου αλγορίθμου είναι αναγκαία. Κατ'αρχήν πρέπει να θεωρηθεί ότι ο πίνακας table έχει n+1 θέσεις, με πρώτη θέση, τη θέση table[0], που χρησιμο-

ποιείται για την προσωρινή αποθήκευση του στοιχείου της ακολουθίας πηγής που πρόκειται να εισαχθεί στην ακολουθία προορισμού. Αυτό το τέχνασμα υιοθετείται για να απλουστευθεί η εντολή ελέγχου της εντολής «Όσο ... επανάλαβε». Με απλά λόγια, έτσι δεν γίνεται κάθε φορά ένας επιπλέον έλεγχος για να διαπιστωθεί αν οι συγκρίσεις έφθασαν μέχρι το αριστερό άκρο του πίνακα.

ΔΣ 4

Η απαιτούμενη δομή είναι ένας δισδιάστατος πίνακας, όπου κάθε γραμμή αντιστοιχεί σε ένα CD. Η πρώτη στήλη αντιστοιχεί στον τίτλο του CD, ενώ η δεύτερη αντιστοιχεί στην χρονολογία έκδοσής του. Ο πίνακας αυτός πρέπει να ταξινομηθεί ως προς τη δεύτερη στήλη, και κατόπιν με σειριακή αναζήτηση να εντοπισθούν τα σχετικά CD.

ΔΣ 5

Αποτελεί απλή επέκταση των αλγορίθμων διαχείρισης ουράς που αναφέρονται στην άσκηση ΔΤ5.

3.9 Συμπληρωματικά στοιχεία

Δίνουμε εδώ μερικά επι πλέον στοιχεία θεωρίας σχετικά με την ύλη του κεφαλαίου.

3.9.1 Ταξινόμηση ευθείας επιλογής

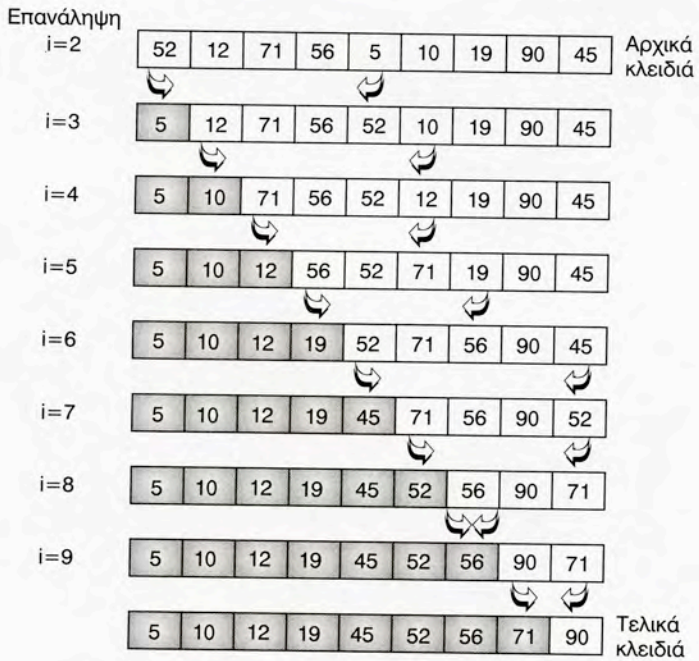
Η μέθοδος αυτή βασίζεται στις ακόλουθες δύο αρχές:

- επιλογή του στοιχείου με το ελάχιστο κλειδί και
- ανταλλαγή αυτού του στοιχείου με το πρώτο στοιχείο του πίνακα.

Αυτές οι λειτουργίες επαναλαμβάνονται για τα υπόλοιπα n-1 στοιχεία, μέχρι στο τέλος να απομείνει μόνο το μεγαλύτερο στοιχείο. Η μέθοδος που περιγράφηκε προηγουμένως ονομάζεται ***ταξινόμηση ευθείας επιλογής*** (straight selection sort) και μπορεί να θεωρηθεί ως αντίθετη της ταξινόμησης ευθείας εισαγωγής. Πιο συγκεκριμένα, η ευθεία εισαγωγή θεωρεί σε κάθε βήμα αφ' ενός το ένα και μο-

ναδικό επόμενο στοιχείο της ακολουθίας πηγής και αφ' ετέρου όλα τα στοιχεία της ακολουθίας προορισμού για να εντοπίσει το κατάλληλο σημείο εισαγωγής. Αντίθετα, η ευθεία επιλογή θεωρεί τα στοιχεία της ακολουθίας πηγής, ώστε να ανιχνεύσει το στοιχείο με το ελάχιστο κλειδί και να το τοποθετήσει ως το επόμενο στοιχείο της ακολουθίας προορισμού.

Παράδειγμα. Η μέθοδος αυτή παρουσιάζεται στο επόμενο σχήμα καθώς εφαρμόζεται στα ίδια γνωστά εννέα κλειδιά. Και πάλι, το ταξινομημένο τμήμα του πίνακα εμφανίζεται με σκίαση, ενώ με τα βέλη εμφανίζονται τα στοιχεία που ανταλλάσσονται αμοιβαία. Λόγου χάριν, στην πρώτη σειρά βρίσκουμε ότι το στοιχεί ο 5 είναι το μικρότερο και αντιμετατίθεται με το πρώτο στοιχείο του πίνακα, το 52. Έτσι προκύπτει η μορφή του πίνακα στη δεύτερη σειρά. τη συνέχεια η διαδικασία προχωρεί με την ίδια λογική μέχρι την τελική ταξινόμηση του πίνακα. Ο αλγόριθμος για την υλοποίηση της μεθόδου ταξινόμησης ευθείας επιλογής αφήνεται στον αναγνώστη ως άσκηση.



Σχήμα 3.2. Ταξινόμηση ευθείας επιλογής.

3.9.2 Συγχώνευση

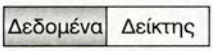
Η συγχώνευση (merging) είναι μία ακόμη βασική λειτουργία των δομών δεδομένων. Ειδικότερα, έχει μελετηθεί το πρόβλημα της συγχώνευσης ενός αριθμού (δύο, τριών ή και περισσοτέρων) ταξινομημένων πινάκων. Στη συνέχεια δίνεται ένας απλός αλγόριθμος συγχώνευσης δύο ταξινομημένων πινάκων σε ένα τρίτο ταξινομημένο πίνακα. Θεωρείται ότι στην είσοδο του αλγορίθμου συγχώνευσης δίνονται δύο ταξινομημένοι πίνακες x και y, μεγέθους n και m στοιχείων αντίστοιχα, ενώ στην έξοδο προκύπτει ένας τρίτος πίνακας z με n+m ταξινομημένα στοιχεία κατά την ίδια φορά. Πιο ειδικά, οι μεταβλητές i, j και k είναι δείκτες για την κίνηση μέσα στους πίνακες x, y και z. Η μέθοδος προχωρά ως εξής. Το μικρότερο στοιχείο από τους πίνακες x και y τοποθετείται στον πίνακα z με ταυτόχρονη αύξηση του αντίστοιχου δείκτη. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου τελειώσουν τα στοιχεία του ενός πίνακα. Ύστερα τα υπόλοιπα στοιχεία του άλλου πίνακα μεταφέρονται στον πίνακα z.

```
Αλγόριθμος Συγχώνευση
Δεδομένα // x(), y(), n, m //
i ← 1
j ← 1
k ← 1
Όσο i <=n και j <=m επανάλαβε
    Αν x[i] < y[j] τότε
        z[k] ← x[i]
        i ← i+1
    αλλιώς
        z[k] ← y[j]
        j ← j+1
    Τέλος_αν
    k ← k+1
Τέλος_επανάληψης
Αν i > n τότε
    Για t από j μέχρι m
        z[k+t-j] ← y[t]
    Τέλος_επανάληψης
αλλιώς
    Για t από i μέχρι n
        z[k+t-i] ← x[t]
Τέλος_αν
```

```
Αποτελέσματα  // z() //
Τέλος Συγχώνευση
```

3.9.3 Λίστες

Στις λίστες το κύριο χαρακτηριστικό είναι ότι οι κόμβοι τους συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης και η σύνδεσή τους γίνεται με δείκτες. Ο *δείκτης* (pointer) είναι ένα πεδίο του κόμβου, το οποίο δεν έχει κάποια συνήθη αριθμητική τιμή (όπως ακέραιος, πραγματικός, χαρακτήρας κ.λπ) αλλά δηλώνει μία διεύθυνση στην κύρια μνήμη. Στο σχήμα 3.3 παρουσιάζεται η δομή του κόμβου μίας λίστας, όπου υποτίθεται ότι το πρώτο πεδίο, που ονομάζεται info, είναι κάποια αλφαριθμητική πληροφορία, ενώ το δεύτερο πεδίο, που ονομάζεται next, είναι ο δείκτης που δείχνει στον επόμενο κόμβο της λίστας.



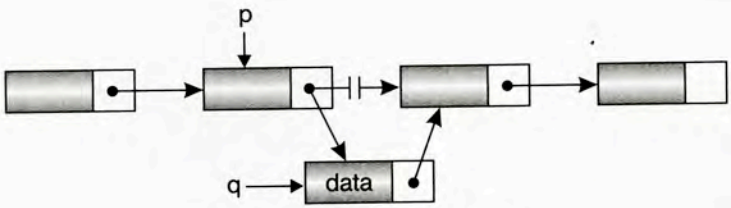
Σχήμα 3.3. *Δομή κόμβου λίστας*

Με τη χρήση δεικτών διευκολύνονται οι λειτουργίες της εισαγωγής και της διαγραφής δεδομένων στις λίστες. Ο επόμενος αλγόριθμος περιγράφει τη λειτουργία της εισαγωγής σε μία λίστα.



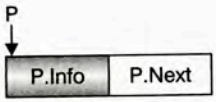
Σχήμα 3.4. *Μία λίστα με τέσσερις κόμβους*

```
Αλγόριθμος Εισαγωγή_σε_λίστα
Δεδομένα // p: δείκτης του κόμβου μετά τον οποίο θα γίνει
         η εισαγωγή data: περιεχόμενο νέου κόμβου
            q: δείκτης της θέσης νέου κόμβου //
q.info ← data
q.next ← p.next
p.next ← q
Τέλος Εισαγωγή_σε_λίστα
```



Σχήμα 3.5. Εισαγωγή σε λίστα

Για την κατανόηση του προηγουμένου αλγορίθμου πρέπει να εξηγηθεί η χρησιμοποιούμενη σύμβαση: αν p είναι ένας δείκτης προς κάποιον κόμβο, τότε p.info είναι το περιεχόμενο του κόμβου και p.next είναι ο δείκτης του ίδιου κόμβου προς τον επόμενο κόμβο. Η εύρεση της θέσης μνήμης όπου θα γίνει η εισαγωγή του νέου κόμβου (στην περίπτωσή μας το q), γίνεται από το σύστημα και δεν απασχολεί τον προγραμματιστή.



Σχήμα 3.6. Δομή κόμβου λίστας

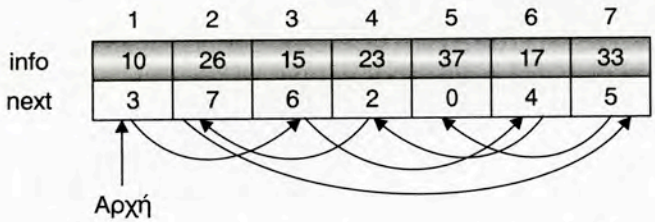
Στη συνέχεια δίνεται ένας απλός αλγόριθμος, που εκτελεί την πράξη της διαγραφής από μία λίστα. Πιο συγκεκριμένα, μας δίνεται ο δείκτης p που δείχνει προς τον κόμβο, του οποίου ο επόμενος θα διαγραφεί. Έτσι αν εφαρμοσθεί ο αλγόριθμος αυτός στη δεύτερη δομή του προηγούμενου σχήματος, θα προκύψει η πρώτη δομή (κατ'αντίστροφη φορά απ'ότι με την εισαγωγή).

```
Αλγόριθμος Διαγραφή_από_λίστα
Δεδομένα // p: δείκτης του κόμβου, του οποίου ο επόμενος
            θα διαγραφεί //
q ← p.next
p.next ← q.next
Τέλος Διαγραφή_από_λίστα
```

Οι λίστες που περιγράφηκαν λέγονται *δυναμικές* (dynamic), γιατί η υλοποίησή τους γίνεται έτσι ώστε να μην απαιτείται εκ των προτέρων καθορισμός του μέγιστου αριθμού κόμβων. Φυσικά, οι δομές αυτές είναι πιο ευέλικτες από τη στατική δομή του πίνακα, επειδή επεκτείνονται και συρρικνώνονται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Οι παλαιότερες γλώσσες προγραμματισμού (ό-

πως οι Algol, Fortran, Cobol, κλπ.) δεν υποστήριζαν αυτές τις δυναμικές δομές, γιατί δεν προσέφεραν τη δυνατότητα δυναμικής παραχώρησης μνήμης κατά την εκτέλεση του προγράμματος. Παρ' όλα αυτά και στις γλώσσες αυτές είναι δυνατό να εφαρμοσθούν τεχνικές επεξεργασίας λιστών κάνοντας όμως χρήση πινάκων με σαφώς από την αρχή προκαθορισμένο μέγεθος.

Για παράδειγμα, στο σχήμα 3.7 παρουσιάζονται δύο πίνακες επτά θέσεων, οι Info και Next (εναλλακτικά μπορεί να χρησιμοποιηθεί ένας δισδιάστατος πίνακας). Στον πίνακα Info είναι αποθηκευμένα τα περιεχόμενα των κόμβων της λίστας, ενώ ο πίνακας Next χρειάζεται για την αποθήκευση των δεικτών. Στην υλοποίηση αυτή οι συγκεκριμένοι 'δείκτες' δεν εξυπηρετούν για να δείχνουν απομακρυσμένες θέσεις του πίνακα, αλλά λαμβάνουν τιμές ακέραιες για να δηλώσουν τη θέση μέσα στον πίνακα Info, όπου βρίσκεται το επόμενο λογικά στοιχείο. Έτσι αν διατρέξουμε τον πίνακα σύμφωνα με τις τιμές των δεικτών αυτών, θα προκύψουν τα δεδομένα σε αύξουσα ταξινομημένη σειρά.



Σχήμα 3.7. *Στατικός πίνακας με χρήση λογικών δεικτών*

3.9.4 Δένδρα

Ολες οι προηγούμενες δομές που εξετάστηκαν, λέγονται γραμμικές, διότι διαθέτουν τη σχέση της συνεχόμενης λογικής γειτονικότητας των στοιχείων τους. Τα δένδρα και οι γράφοι ανοίκουν στις μη γραμμικές δομές.

Η δομή του δένδρου σε γενικές γραμμές δεν είναι τίποτε άλλο παρά η σύνδεση κόμβων με τρόπο ανάλογο ενός πραγματικού δένδρου.

Τα δένδρα είναι ειδικές μορφές γράφων (graphs). Ενας ορισμός των δένδρων είναι ο ακόλουθος:

Δένδρο *είναι ένα σύνολο κόμβων που συνδέονται με ακμές. Υπάρχει ένας μόνο κόμβος που ονομάζεται ρίζα (root) και στον οποίο δεν καταλήγουν, αλλά μόνο ξε-*

κινούν ακμές. Από κάθε κόμβο μπορούν να ξεκινούν καμία, μία ή περισσότερες ακμές. Σε κάθε κόμβο (εκτός της ρίζας) καταλήγει μία μόνο ακμή.

Μερικοί ορισμοί.

Ενας κόμβος α είναι ο **πατέρας** (father) των κόμβων β και γ όταν από τον α αρχίζουν ακμές που καταλήγουν στους β και γ. Αντίστοιχα οι β και γ είναι τα **παιδιά** (children) του α. Η ορολογία αυτή επεκτείνεται προς τα πάνω ή κάτω με παππούδες και εγγονούς και γενικότερα προγόνους και απογόνους. Στο δένδρο του σχήματος 3.12 του βιβλίου μαθητή ο κόμβος Β έχει πατέρα τον κόμβο Α και παιδιά τους κόμβους Δ και Ε.

Βαθμός (degree) ενός κόμβου είναι ο αριθμός των παιδιών του. **Βαθμός ενός δένδρου** είναι ο μέγιστος από τους βαθμούς των κόμβων του. Δένδρα βαθμού n ονομάζονται n-αδικά (n-ary) δένδρα. Τα δένδρα που χρησιμοποιούνται συνήθως είναι δυαδικά (binary), τριαδικά (ternary) ή τετραδικά (quadtree).

Οι κόμβοι που δεν έχουν παιδιά λέγονται **τερματικοί κόμβοι** ή **φύλλα** (terminal nodes ή leaves). Οι υπόλοιποι κόμβοι λέγονται εσωτερικοί ή μη τερματικοί ή κλαδιά (internal ή nonterminal nodes ή branches).

Επίπεδο ενός κόμβου είναι ο αριθμός των προγόνων του (μέχρι τη ρίζα) συν 1. Το μέγιστο επίπεδο των κόμβων ενός δένδρου λέγεται **βάθος** (depth) ή **ύψος** (height) του δένδρου. Το δένδρο του σχήματος 3.12 είναι δυαδικό και έχει ύψος 3.

Υποδένδρο ονομάζεται το δένδρο που σχηματίζεται, αν ως ρίζα ληφθεί ένας οποιοσδήποτε κόμβος.

Αναπαράσταση των δένδρων στη μνήμη του υπολογιστή.

Στη μνήμη ενός υπολογιστή χρησιμοποιείται ανάλογη τεχνική για την αναπαράστασή τους, που χρησιμοποιείται και στις λίστες. Κάθε κόμβος αποτελείται από τα δεδομένα που περιέχει και από ένα σύνολο δεικτών ίσων σε πλήθος με το βαθμό του δένδρου. Οι δείκτες δείχνουν στα παιδιά του κόμβου. Αν δεν υπάρχει κάποιο παιδί, τότε ο αντίστοιχος δείκτης είναι κενός (nil).

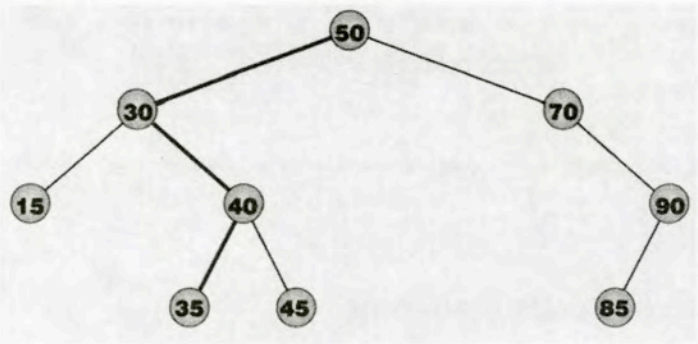
Εναλλακτικά μπορούν να χρησιμοποιηθούν n+1 πίνακες, όπου n ο βαθμός του δένδρου. Στο σχήμα 3.8 φαίνεται η αποθήκευση σε πίνακες του δυαδικού δένδρου του σχήματος 3.12 (βιβλίου μαθητή).

	Left	Info	Right
1	2	A	3
2	4	B	5
3	6	Γ	7
4	0	Δ	0
5	0	E	0
6	0	Z	0
7	0	H	0

Σχήμα 3.8

➤ Δυαδικά δένδρα αναζήτησης

Μια πολύ ενδιαφέρουσα κατηγορία δένδρων είναι τα δυαδικά δένδρα αναζήτησης. Τα δένδρα αναζήτησης είναι διατεταγμένα δένδρα. Διατεταγμένο δένδρο λέγεται το δένδρο στο οποίο έχει σημασία η διάταξη των παιδιών κάθε κόμβου. Ετσι σε ένα δένδρο αναζήτησης η τιμή κάθε κόμβου είναι μεγαλύτερη από την τιμή όλων των κόμβων του αριστερού του υποδένδρου και μεγαλύτερη από την τιμή όλων των κόμβων του δεξιού του υποδένδρου. Ενα δυαδικό δένδρο αναζήτησης φαίνεται στο σχήμα 3.9.



Σχήμα 3.9.

Τα δένδρα αναζήτησης χρησιμοποιούνται όταν είναι σημαντική η γρήγορη προσπέλαση δεδομένων που υπακούουν σε κάποια σειρά κατάταξης.

Για την εύρεση ενός συγκεκριμένου κόμβου σε ένα δυαδικό δένδρο αναζήτησης πρώτα εξετάζεται η τιμή της ρίζας του δένδρου σε σχέση με την τιμή του κόμβου που ζητάμε. Αν ο ζητούμενος κόμβος είναι μικρότερος από τη ρίζα συνεχίζουμε με το αριστερό υποδένδρο, αν είναι μεγαλύτερος με το δεξί (φυσικά αν είναι ίσος τότε η αναζήτηση τελειώνει). Η αναζήτηση σταματά αν βρεθεί ο ζητούμενος κόμβος ή φτάσουμε σε κενό δείκτη. Στο σχήμα 3.9 φαίνεται με παχιά γραμμή η διαδρομή αναζήτησης του κόμβου με τιμή 35 και η μέθοδος αυτή παρουσιάζεται στον αλγόριθμο Search-Node.

```
Αλγόριθμος Search-Node
Δεδομένα // Root : ρίζα δένδρου, Κ : ζητούμενος κόμβος //
Β ← Root
Λ ← ψευδής
Οσο Β ≠ nil και Λ = ψευδής επανάλαβε
    Αν Β.Info = Κ τότε Λ = αληθής
    αλλιώς Αν Κ > Β.Info τότε Β ← Β.Right
                          αλλιώς Β ← Β.Left
          τέλος_αν
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // Αν Λ αληθής τότε επιτυχής αναζήτηση. Αν Λ ψευδής τότε ανεπιτυχής αναζήτηση και Β η θέση στην οποία έπρεπε να βρίσκεται ο ζητούμενος κόμβος //
Τέλος Search-Node
```

Και εδώ εννοείται ότι, αν ένας κόμβος δεικτοδοτείται από το δείκτη Β, τότε Β. Info είναι το πεδίο δεδομένα και Β.Left, Β.Right οι δείκτες προς το αριστερό και δεξί παιδί αντίστοιχα.

Η εισαγωγή κόμβου σε δυαδικό δένδρο αναζήτησης γίνεται με τον εξής τρόπο: Αρχικά ακολουθείται η διαδικασία της αναζήτησης, αναζητώντας τον κόμβο που πρέπει να προστεθεί. Ο κόμβος φυσικά δεν θα υπάρχει και βρεθεί ένας κενός δείκτης. Ο κόμβος προστίθεται με τρόπο ώστε ο κενός δείκτης τώρα να δείχνει σ' αυτόν.

Η διαγραφή κόμβου είναι αρκετά πιο δύσκολη. Αν ο κόμβος που πρέπει να διαγραφεί είναι φύλλο, απλά διαγράφεται. Αν έχει ένα μόνο παιδί, τότε αντικαθίσταται από τον κόμβο παιδί. Αν όμως έχει δύο παιδιά τότε αντικαθίσταται είτε από τον πιο δεξιό κόμβο του αριστερού υποδένδρου του είτε από τον πιο αριστερό κόμβου του δεξιού υποδένδρου.

➤ Διάσχιση δένδρων

Μια άλλη σημαντική λειτουργία στα δένδρα είναι η διάσχιση (traverse). **Διάσχιση** λέγεται η λειτουργία κατά την οποία πρέπει να επισκεφθούμε κάθε κόμβο ακριβώς μια φορά.

Για την επεξήγηση της λειτουργίας αυτής είναι χρήσιμος ο επόμενος αναδρομικός ορισμός του δένδρου.

Δένδρο *είναι ένα σύνολο κόμβων που συνδέονται με ακμές. Υπάρχει ένας ειδικός κόμβος που ονομάζεται ρίζα (root) και στον οποίο δεν καταλήγουν αλλά μόνο ξεκινούν ακμές. Οι υπόλοιποι κόμβοι αποτελούν ξένα μεταξύ τους υποδένδρα (subtrees).*

Υπάρχουν τρεις κύριοι διαφορετικοί τρόποι επίσκεψης των κόμβων. Κάθε τρόπος απαιτεί την εκτέλεση των επόμενων τριών διαδικασιών με κάποια σειρά.

1. Επίσκεψη της ρίζας
2. Επίσκεψη του αριστερού υποδένδρου
3. Επίσκεψη του δεξιού υποδένδρου.

Οι διαδικασίες 2 και 3 απαιτούν την επίσκεψη ενός υποδένδρου που μπορεί να γίνει επαναλαμβάνοντας τις ίδιες διαδικασίες με την ίδια σειρά.

Προδιαταγμένη διάσχιση

Αν η σειρά πραγματοποίησης των διαδικασιών αυτών είναι αυτή που σημειώνεται πιο πάνω (δηλαδή 1, 2, 3), τότε η διάσχιση των κόμβων λέγεται προδιαταγμένη (preorder). Η προδιαταγμένη διάσχιση του δένδρου του σχήματος 3.12 δίδει το εξής αποτέλεσμα:

Α Β Δ Ε Γ Ζ Η

Στην προδιαταγμένη διάσχιση επισκεφτόμαστε τη ρίζα πριν από τα παιδιά της.

Η μέθοδος της προδιαταγμένης διάσχισης περιγράφεται με τον (αναδρομικό) αλγόριθμο Preorder-Traversal.

```
Αλγόριθμος Preorder-Traversal (R)
Δεδομένα // R: δείκτης ρίζας //
```

```
Οσο R ≠ nil επανάλαβε
    'επίσκεψη ρίζας
    Τύπωσε R.Info
    ' επίσκεψη αριστερού υποδένδρου
    A = R.Left
    Preorder-Traversal (A)
    'επίσκεψη δεξιού υποδένδρου
    Δ = R.Right
    Preorder-Traversal (Δ)
Τέλος_επανάληψης
Αποτελέσματα // λίστα κόμβων //
Τέλος Preorder-Traversal
```

Στον αλγόριθμο αυτό ως επεξεργασία του κόμβου εννοείται η εκτύπωση του περιεχομένου του. Η υλοποίηση του αλγορίθμου θα δόσει στην έξοδο τη σειρά επίσκεψης των κόμβων με την προδιαταγμένη διάσχιση που αναφέρθηκε πιο πάνω.

Μεταδιαταγμένη διάσχιση

Αν η σειρά εκτέλεσης των προηγούμενων διαδικασιών γίνεται 2, 3, 1 δηλαδή

- Επίσκεψη του αριστερού υποδένδρου
- Επίσκεψη του δεξιού υποδένδρου
- Επίσκεψη της ρίζας

τότε η διάσχιση αυτή λέγεται μεταδιαταγμένη (postorder) και το αποτέλεσμα για το δένδρο του σχήματος 3.12 είναι:

Δ Ε Β Ζ Η Γ Α

Στη μεταδιαταγμένη διάσχιση επισκεφτόμαστε τη ρίζα μετά τα παιδιά της.

Ενδοδιαταγμένη διάσχιση

Τέλος αν η σειρά εκτέλεσης των διαδικασιών αυτών γίνει 2, 1, 3 δηλαδή

- Επίσκεψη αριστερού υποδένδρου
- Επίσκεψη ρίζας

- Επίσκεψη δεξιού υποδένδρου

τότε ορίζεται η ενδοδιαταγμένη (inorder) διάσχιση, με αποτέλεσμα στο ίδιο γράφημα:

Δ Β Ε Α Ζ Γ Η

Στην ενδοδιαταγμένη διάσχιση επισκεφτόμαστε τη ρίζα μεταξύ των παιδιών της.

Ας σημειωθεί ότι μπορεί να υπάρξουν τρεις ακόμη μέθοδοι διάσχισης, αν η σειρά επίσκεψης των υποδένδρων αντιστραφεί, δηλαδή η επίσκεψη του δεξιού να προηγείται αυτής του αριστερού.

➤ *Πολωνικός συμβολισμός*

Το 1951 ο Πολωνός μαθηματικός Jan Lukasiewicz πρότεινε ένα συμβολισμό για αριθμητικές παραστάσεις χωρίς να κάνει χρήση παρενθέσεων. Ο συμβολισμός εφαρμόστηκε επίσης στην άλγεβρα καθώς και σε άλλα συστήματα τελεστών-τελεστέων και έγινε γνωστός με το όνομα πολωνικός συμβολισμός (polish notation), προς τιμή του Πολωνού μαθηματικού.

Ο πολωνικός συμβολισμός συνίσταται στο να τοποθετούνται οι τελεστές πριν από τους τελεστέους. Για παράδειγμα αντί για τις παραστάσεις:

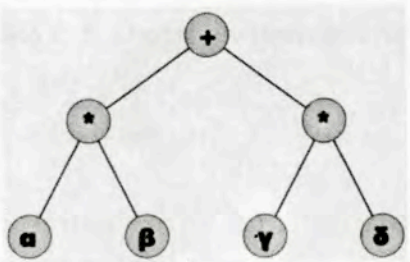
1. α + β γράφουμε + α β
2. (α + β) . γ γράφουμε . + α β γ
3. α + (β . γ) γράφουμε + α . β γ
4. α . β + γ . δ γράφουμε + . α β . γ δ

Στον πολωνικό συμβολισμό η χρήση παρενθέσεων δεν είναι απαραίτητη με την προϋπόθεση ότι κάθε τελεστής έχει συγκεκριμένο αριθμό τελεστέων. Ετσι η έκφραση + α β έχει το ίδιο νόημα με την α + β, αφού γνωρίζουμε ότι ο τελεστής + έχει δύο τελεστέους που ακολουθούν τον τελεστή. Το σημείο μείον (-) δημιουργεί κάποιο πρόβλημα, αφού χρησιμοποιείται και σαν δυαδικός τελεστής (π.χ. α-β) αλλά και σαν μονοδικός (unary) τελεστής (π.χ. -α). Το πρόβλημα λύνεται αρκεί να περιορίσουμε τη χρήση του μείον σαν αποκλειστικά μονοδικού. Αυτό σημαίνει ότι η παράσταση α - β, πρέπει να γραφεί σαν α + (-β) και επόμενα μπορεί να γραφεί στον πολωνικό συμβολισμό + α - β χωρίς πρόβλημα.

Αν αντί να τοποθετούμε τους τελεστές πριν τους τελεστέους τους τοποθετούμε μετά, έχουμε τον **ανάστροφο πολωνικό συμβολισμό** (reverse polish notation). Για παράδειγμα αντί για τις παραστάσεις:

1. $\alpha + \beta$ γράφουμε $\alpha\ \beta +$
2. $(\alpha + \beta) . \gamma$ γράφουμε $\alpha\ \beta + \gamma .$
3. $\alpha + (\beta . \gamma)$ γράφουμε $\alpha\ \beta\ \gamma . +$
4. $\alpha . \beta + \gamma . \delta$ γράφουμε $\alpha\ \beta . \gamma\ \delta . +$

Μπορούμε να πάρουμε την πολωνική και την ανάστροφη πολωνική μορφή μιας παράστασης αν διασχίσουμε το δυαδικό δένδρο της παράστασης με προδιατεταγμένο και αντίστοιχα μεταδιατεταγμένο τρόπο. Π.χ. η παράσταση $\alpha . \beta + \gamma . \delta$ απεικονίζεται με το δυαδικό δένδρο του σχήματος 3.10.



Σχήμα 3.10

Η προδιαταγμένη διάσχιση του δένδρου αυτού δίδει τον πολωνικό συμβολισμό της παράστασης, που αποκαλείται **προθεματική** (prefix) μορφή :

$+ . \alpha\ \beta . \gamma\ \delta$

Η μεταδιαταγμένη διάσχιση του ίδιου δένδρου δίδει τον ανάστροφο πολωνικό συμβολισμό της παράστασης, που αποκαλείται **μεταθεματική** (postfix) μορφή :

$\alpha\ \beta . \gamma\ \delta . +$

Ας σημειωθεί τέλος, ότι η ενδοδιαταγμένη διάσχιση δίδει τη γνωστή **ένθετη** (infix) μορφή της παράστασης, δηλαδή:

$\alpha . \beta + \gamma . \delta$

Ο πολωνικός συμβολισμός χρησιμοποιείται σε μερικές γλώσσες προγραμματισμού αντί του γνωστού μας συμβολισμού με χρήση παρενθέσεων, π.χ. APL και FORTH. Χρήση βρίσκει επίσης και σε ορισμένους τύπους προγραμματιζόμενων αριθμομηχανών (π.χ. Hewlett- Packard). Μεγάλη χρήση βρίσκουν επίσης από τους μεταγλωττιστές και διερμηνευτές για την αποτίμηση εκφράσεων.

Παραστάσεις γραμμένες σε ανάστροφο πολωνικό συμβολισμό μπορούν να υπολογιστούν πολύ εύκολα από τα αριστερά προς τα δεξιά. Αρκεί όταν συναντάται κάποιος τελεστής να γίνεται η αντίστοιχη πράξη με τελεστέους αυτούς που υπολογίστηκαν πιο πριν.

Για να γίνει αυτό απαιτείται η χρήση μιας στοίβας και ο ορισμός μερικών πράξεων, όπως:

ADD Οι δύο τιμές στην κορυφή της στοίβας προστίθενται και το αποτέλεσμα τοποθετείται στην κορυφή της στοίβας.

MUL Οι δύο τιμές στην κορυφή της στοίβας πολλαπλασιάζονται και το αποτέλεσμα τοποθετείται στην κορυφή της στοίβας.

Ο αλγόριθμος αποτίμησης της έκφρασης διατρέχει την έκφραση από τ' αριστερά προς τα δεξιά και για κάθε τελεστέο που συναντά, εκτελεί μια λειτουργία push στη στοίβα. Οταν συναντήσει κάποιον τελεστή, τότε εκτελεί την αντίστοιχη πράξη ADD ή MUL (ή άλλη). Ετσι για να υπολογιστεί η παράσταση α β γ . + (δηλαδή η α+β.γ) πρέπει να γίνουν οι εξής πράξεις:

PUSH α

PUSH β

PUSH γ

MUL

ADD

και η στοίβα έχει στην κορυφή της την τιμή της παράστασης.

3.9.5 Γράφοι

Πολλά πρακτικά προβλήματα και καταστάσεις της καθημερινής μας ζωής μπορούν να περιγραφούν με τη βοήθεια ενός διαγράμματος, που αποτελείται

από ένα σύνολο σημείων και ένα σύνολο γραμμών που ενώνουν ζεύγη σημείων. Για παράδειγμα, θα μπορούσε τα σημεία να συμβολίζουν πρόσωπα και ο γραμμές να δηλώνουν πρόσωπα που είναι φίλοι μεταξύ τους. Η περιγραφή αυτή είναι ένας *γράφος* (graph). Άλλωστε και η δομή του σχήματος 4.1 (βιβλίου μαθητή) είναι ένα παράδειγμα γράφου. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων. Δηλαδή, μπορεί να λεχθεί ότι οι πίνακες, οι λίστες και τα δένδρα είναι υπο-περιπτώσεις των γράφων.

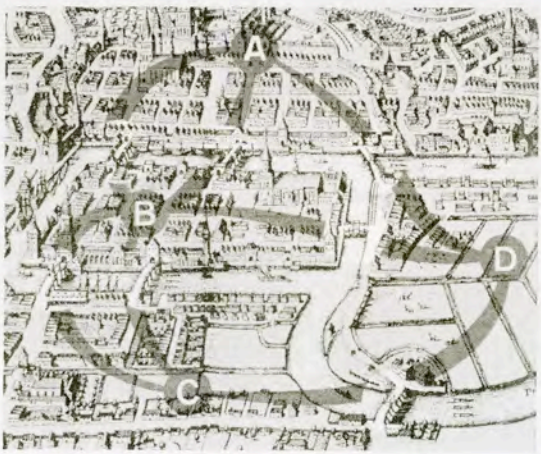
Ο γράφος αποθηκεύεται στη μνήμη του υπολογιστή με τη βοήθεια ενός δισδιάστατου πίνακα a, που ονομάζεται *πίνακας γειτνίασης* (adjacency matrix). Ο πίνακας αυτός είναι τετραγωνικός, δηλαδή nxn, όπου n είναι ο αριθμός των σημείων. Το στοιχείο a_{ij} του πίνακα είναι 0 αν τα σημεία i και j δεν ενώνονται, ενώ αν ενώνονται τότε το στοιχείο a_{ij} λαμβάνει τιμή ίση με το λεγόμενο *βάρος* (weight) της σύνδεσης, όπως είναι η χιλιομετρική απόσταση στο παράδειγμά μας. Ο επόμενος πίνακας είναι ο πίνακας γειτνίασης του προβλήματος του ταχυδρομικού διανομέα. Παρατηρούμε επίσης ότι ο πίνακας αυτός είναι *συμμετρικός* (symmetric), δηλαδή ισχύει $a_{ij} = a_{ji}$.

0	5	13	6
5	0	9	8
13	9	0	10
6	8	11	0

Στη γενική περίπτωση, ο πίνακας γειτνίασης δεν είναι συμμετρικός, δηλαδή δεν ισχύει πάντοτε η σχέση $a_{ij} = a_{ji}$. Για παράδειγμα, έστω ότι οι πλατείες και οι διασταυρώσεις ενός αστικού οδικού δικτύου αντιστοιχούν στα σημεία του γράφου, ενώ οι γραμμές του γράφου δηλώνουν τις αποστάσεις μεταξύ τους. Σε μία τέτοια περίπτωση, οι μονόδρομοι που ενώνουν δύο διαφορετικά σημεία μπορεί να έχουν διαφορετικό μήκος. Αν δεν υπάρχουν βάρη μεταξύ των σημείων, τότε όλα τα στοιχεία του πίνακα που αντιστοιχούν στις συνδέσεις αυτές λαμβάνουν την τιμή 1. Έτσι, ο πίνακας γειτνίασης είναι ένας *δυαδικός* (binary) πίνακας, επειδή αποτελείται από 0 και 1.

Θεωρία Γράφων

Η Θεωρία Γράφων είναι ένα πεδίο με μεγάλο ιστορικό βάθος. Βεβαίως, προϋπήρξε της Πληροφορικής ως ένα πεδίο των Μαθηματικών, αλλά η Πληροφορική έδωσε ώθηση στη Θεωρία Γράφων εμπλουτίζοντάς την με την έννοια του αλγορίθμου. Αναφέρεται ότι η πρώτη ιστορική διατριβή που έθεσε τις βάσεις της Θεωρίας Γράφων ήταν μία μελέτη του Leoard Euler που έλυσε το πρόβλημα των "γεφυρών του Koenigsberg", το 1736. Στο επόμενο σχήμα φαίνεται μια άποψη του Koenigsberg που διασχίζεται από τον ποταμό Pregel, ενώ μέσα στον ποταμό υπάρχουν 2 νησάκια που ενώνονται με τις όχθες και μεταξύ τους με 7 γέφυρες. Οι κάτοικοι του Koenigsberg δεν μπορούσαν να λύσουν το πρόβλημα "ποια είναι η πορεία που κάποιος πρέπει να ακολουθήσει ώστε να ξεκινήσει από ένα σημείο, να περάσει από όλες τις γέφυρες μία μόνο φορά και να επιστρέψει στο ίδιο σημείο". Ο Euler απόδειξε ότι δεν υπάρχει λύση στο πρόβλημα.



Μία άλλη ιστορική μελέτη που σημάδεψε τη Θεωρία Γράφων αποδίδεται στον Kirchoff, που το 1847 ανέπτυξε τη δομή των δένδρων και μελέτησε τις εφαρμογές τους στα ηλεκτρικά δίκτυα, ενώ λίγο αργότερα ο Cayley θεώρησε τη δομή των δένδρων για να απαριθμήσει τα ισομερή των κεκορεσμένων υδρογονανθράκων. Ένα ακόμη ιστορικό πρόβλημα ήταν ο "χρωματισμός χαρτών με 4 χρώματα", δηλαδή η δυνατότητα να χρωματισθούν οι περιοχές ενός πολιτικού χάρτη με 4 χρώματα το πολύ, έτσι ώστε δύο γειτονικές περιοχές να μην έχουν το ίδιο χρώμα. Σημειώνεται ότι το πρόβλημα αυτό τέθηκε το 1850 και επιλύθηκε το 1976.

3.10 Απάντηση δεύτερης λύσης του παραδείγματος της παραγράφου 3.2 βιβλίου μαθητή

Σχετικά με τη δεύτερη προσέγγιση του παραδείγματος του ΟΤΕ (σελίδα 51).

Η αλγοριθμική λύση της δεύτερης προσέγγισης έχει ως εξής. Κατ' αρχήν υποτίθεται ότι οι συνδρομητές είναι ταξινομημένοι αλφαβητικά και μπορούν να αποθηκευτούν σε ένα δισδιάστατο πίνακα (τον κύριο πίνακα) με τόσες γραμμές, όσοι είναι οι συνδρομητές και με δύο στήλες που αντιστοιχούν στο όνομα και το τηλέφωνο του συνδρομητή. Ομως επιπλέον χρειαζόμαστε μια ακόμη βοηθητική δομή, δηλαδή ένα δισδιάστατο πίνακα 24 γραμμών και 2 στηλών, όπου τα στοιχεία της πρώτης στήλης είναι τα γράμματα του αλφαβήτου, ενώ τα στοιχεία της δεύτερης στήλης δίνουν τη θέση στον πρώτο πίνακα, όπου είναι αποθηκευμένος ο πρώτος λεξικογραφικά συνδρομητής του οποίου το όνομα αρχίζει από το αντίστοιχο γράμμα.

Εναλλακτικά ο βοηθητικός πίνακας μπορεί να είναι μονοδιάστατος με 24 στοιχεία, όπου κάθε στοιχείο αντιστοιχεί κατά σειρά σε ένα από τα γράμματα του αλφαβήτου (δηλαδή Α, Β, κ.λπ.). Βέβαια, το περιεχόμενο κάθε θέσης αυτού του μονοδιάστατου πίνακα αντιστοιχεί στο συνδρομητή του οποίου το όνομα είναι το πρώτο λεξικογραφικά στον κύριο πίνακα και αρχίζει βέβαια από το αντίστοιχο γράμμα.

Επομένως για την εύρεση του τηλεφώνου ενός συνδρομητή, η αναζήτηση αρχίζει από τη μικρότερη βοηθητική δομή με βάση το πρώτο γράμμα του ονόματος του συνδρομητή. Ετσι εντοπίζεται εύκολα η θέση στον κύριο πίνακα, όπου είναι αποθηκευμένος ο πρώτος συνδρομητής του οποίου το όνομα αρχίζει από το αντίστοιχο γράμμα. Στη συνέχεια η αναζήτηση συνεχίζεται στον κύριο πίνακα σειριακά μέχρι την εύρεση του ονόματος του σχετικού συνδρομητή και τον εντοπισμό του αριθμού τηλεφώνου του.

Κεφάλαιο 4

Τεχνικές Σχεδίασης Αλγορίθμων

4.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να κατανοήσουν οι μαθητές τις σύγχρονες τεχνικές σχεδίασης αλγορίθμων και να γίνουν ικανοί να αναγνωρίζουν την τεχνική που είναι κατάλληλη για την επίλυση προβλημάτων. Ο διδακτικός στόχος είναι να εξοικειωθούν οι μαθητές με τις διαφορετικές τεχνικές και να μπορούν να εντάξουν προβλήματα σε αλγορίθμους των τεχνικών αυτών, ώστε να βελτιώσουν την αποδοτικότητα επίλυσης ενός προβλήματος.

4.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να κατανοήσουν τις τεχνικές Διαίρει και Βασίλευε, Δυναμικό Προγραμματισμό και Άπληστη προσέγγιση.
- ανάπτυξη,αλγορίθμων και παράθεση παραδειγμάτων για την επίλυση προβλημάτων μέσω των προσεγγίσεων αυτών.
- να αναγνωρίζουν και να διακρίνουν τα κυριότερα είδη των αλγοριθμικών τεχνικών που χρησιμοποιούνται για την επίλυση προβλημάτων
- να εντάσσουν ένα πρόβλημα στην κατάλληλη τεχνική επίλυσης και να μπορούν να τεκμηριώσουν την επιλογή της συγκεκριμένης τεχνικής.
- να χρησιμοποιούν την κατάλληλη τεχνική με στόχο τη βελτίωση της επίδοσης του προτεινόμενου αλγορίθμου.

4.3 Οδηγίες – επισημάνσεις

Ιδιαίτερη έμφαση και προσοχή πρέπει να δοθεί στις δύο προσεγγίσεις των αλγορίθμων: "από επάνω προς τα κάτω" και "από κάτω προς τα επάνω". Οι δύο αυτές προσεγγίσεις αντιστοιχούν στους αλγορίθμους των οικογενειών "Διαίρει και Βασίλευε" και "Δυναμικού Προγραμματισμού". Η πρώτη διάκριση εξυπηρετεί περισσότερο σκοπούς διδακτικής, ενώ η δεύτερη διάκριση είναι περισσότερο τεχνική και έτσι συναντάται στη βιβλιογραφία.

Είναι απαραίτητο να γίνουν αρκετά παραδείγματα για τις τεχνικές αυτές και να δοθεί έμφαση στον τρόπο επιλογής κάθε τεχνικής αναφέροντας τον τρόπο προσαρμογής της σε ένα δεδομένο πρόβλημα. Επίσης, είναι χρήσιμο να γίνει και πάλι συζήτηση για το πρόβλημα της ταξινόμησης και τους αντίστοιχους αλγορίθμους που δόθηκαν προηγουμένως. Για παράδειγμα, θα μπορούσε να θεωρηθεί ότι ο αλγόριθμος της ευθείας εισαγωγής (δες δραστηριότητα ΔΣ3 του προηγούμενου κεφαλαίου) εφαρμόζει την "άπληστη μέθοδο" με την έννοια ότι κάθε φορά εξετάζει τις διαθέσιμες επιλογές και βρίσκει κάθε φορά τη βέλτιστη. Ακόμη, θα μπορούσαν να δοθούν παραδείγματα που αφορούν σε αποτύπωση προβλημάτων σε μορφή γράφων και να τονισθεί η εφαρμογή τους σε ένα εύρος προβλημάτων.

4.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να κατανοήσουν την τεχνική Διαίρει και Βασίλευε
- να γίνει ανάπτυξη αλγορίθμων και παράθεση παραδειγμάτων για την επίλυση προβλημάτων μέσω αυτής της προσέγγισης.
- να χρησιμοποιούν την κατάλληλη τεχνική με στόχο τη βελτίωση της επίδοσης του προτεινόμενου αλγορίθμου.

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Ανάλυση προβλημάτων.** Ανάλυση προβλήματος με πρόταση για συγκεκριμένη μεθοδολογία και ακολουθία βημάτων. Πρόταση έξυπνων και αποδοτικών λύσεων.
- **Μέθοδοι σχεδίασης αλγορίθμων.** Καταγραφή των κυριοτέρων μεθόδων σχεδίασης αλγορίθμων. Εξήγηση της αναγκαιότητας ύπαρξης διαφορετικών τεχνικών προσέγγισης για την επίλυση προβλημάτων.

- **Μέθοδος διαίρει και βασίλευε.** Περιγραφή της μεθόδου και εξήγηση της σπουδαιότητάς της και της κατηγορίας προβλημάτων που αφορά.
- **Δυαδική αναζήτηση.** Καταγραφή του αλγορίθμου της δυαδικής αναζήτησης και σχολιασμός της αποδοτικότητας του.

Περιεχόμενα πρακτικής εφαρμογής

- **Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή.** Θα πρέπει να διδαχθούν οι παράγραφοι 4.1 έως και 4.3 από το βιβλίο του μαθητή. Να απαντηθούν οι ερωτήσεις 1-4 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.
- **Δραστηριότητες από το τετράδιο του μαθητή.** Θα πρέπει να διδαχθούν τα παραδείγματα 1 και 2 από το τετράδιο του μαθητή.
- Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ2 ή ΔΤ3 στην τάξη, και μία από τις δραστηριότητες ΔΣ1, ΔΣ2 ή ΔΣ4 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Η λύση σε ένα πρόβλημα μπορεί να προέλθει από ποικίλες και διαφορετικές προσεγγίσεις, τεχνικές και μεθόδους.
2. Η ανάλυση ενός προβλήματος σε ένα σύγχρονο υπολογιστικό περιβάλλον περιλαμβάνει την εισαγωγή και εξαγωγή δεδομένων.
3. Δεν υπάρχουν πολλά πρακτικά προβλήματα που να μπορούν να περιγραφούν με τη βοήθεια ενός διαγράμματος ή γράφου.
4. Ο γράφος αποθηκεύεται (συνήθως) στη μνήμη του υπολογιστή με τη βοήθεια ενός δισδιάστατου πίνακα.
5. Υπάρχει ένας ενιαίος κανόνας που να αναφέρεται στην επίλυση του συνόλου των προβλημάτων.
6. Στην κατηγορία "Διαίρει και Βασίλευε" εντάσσονται οι τεχνικές που αναλύουν ένα πρόβλημα σε μεγαλύτερα προβλήματα.

7. Ο αλγόριθμος της Δυαδικής αναζήτησης εφαρμόζεται σε έναν μη ταξινομημένο πίνακα.
8. Με την τεχνική "Διαίρει και Βασίλευε" υποδιαιρείται το στιγμιότυπο του προβλήματος σε υπο-στιγμιότυπα του ίδιου προβλήματος.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση

9. Η ανάλυση προβλημάτων περιλαμβάνει:

   α) καταγραφή υπάρχουσας πληροφορίας

   β) καταγραφή αποτελεσμάτων

   γ) αναγνώριση στοιχείων εισόδου του προβλήματος

   δ) πρόταση για την είσοδο και έξοδο των δεδομένων

10. Ένας από τους αλγορίθμους που εντάσσονται στην κατηγορία «Διαίρει και Βασίλευε» είναι:

    α) ταξινόμηση με ευθεία ανταλλαγή

    β) ταξινόμηση με ευθεία επιλογή

    γ) δυαδική αναζήτηση

    δ) υπολογισμός δύναμης

11. Κάθε τεχνική αλγορίθμων πρέπει να έχει:

    α) τη δική της υπολογιστική μηχανή

    β) τη δική της ακολουθία εντολών

    γ) τη δική της είσοδο και έξοδο

    δ) τη δική της γλώσσα προγραμματισμού

Απαντήσεις τεστ αξιολόγησης επίδοσης

1: σωστό	5: λάθος	9: α
2: λάθος	6: λάθος	10: γ
3: λάθος	7: λάθος	11: β
4: σωστό	8: σωστό	

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να κατανοήσουν τις τεχνικές του Δυναμικού Προγραμματισμού και της Άπληστης προσέγγισης
- ανάπτυξη αλγορίθμων και παράθεση παραδειγμάτων για την επίλυση προβλημάτων μέσω των δύο αυτών προσεγγίσεων.
- να αναγνωρίζουν και να διακρίνουν όλα τα είδη των αλγοριθμικών τεχνικών που χρησιμοποιούνται για την επίλυση προβλημάτων
- να εντάσσουν ένα πρόβλημα στην κατάλληλη τεχνική επίλυσης και να μπορούν να τεκμηριώσουν την επιλογή της συγκεκριμένης τεχνικής.
- να χρησιμοποιούν την κατάλληλη τεχνική με στόχο τη βελτίωση της επίδοσης του προτεινόμενου αλγορίθμου.

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Δυναμικός προγραμματισμός.** Εισαγωγή στη λογική και στα βήματα που ακολουθεί η τεχνική του Δυναμικού Προγραμματισμού.
- **Άπληστη Μέθοδος**. Επεξηγείται και δίνεται έμφαση στην αναγκαιότητα της μεθόδου αυτής για την επίλυση προβλημάτων βελτιστοποίησης.
- **Επανάληψη των εννοιών που διδάχθηκαν – Ανακεφαλαίωση.**

Περιεχόμενα πρακτικής εφαρμογής

- **Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή.** Θα πρέπει να διδαχθούν οι παράγραφοι 4.4 έως και 4.5 από το βιβλίο του μαθητή. Να απαντηθούν οι ερωτήσεις 5-8 από το βιβλίο του μαθητή.
- **Δραστηριότητες από το τετράδιο του μαθητή.** Θα πρέπει να διδαχθεί το παράδειγμα 3 από το τετράδιο του μαθητή.
- Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση οι δραστηριότητες ΔΤ1 στην τάξη και μία-δύο από τις δραστηριότητες ΔΣ3, ΔΣ5 και ΔΣ6 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Η φιλοσοφία της τεχνικής του Δυναμικού προγραμματισμού από την αρχή να επιλύονται τα μεγαλύτερα προβλήματα και σταδιακά να επιλύονται τα μικρότερα.
2. Η μέθοδος του Δυναμικού προγραμματισμού χρησιμοποιείται κυρίως για την επίλυση προβλημάτων βελτιστοποίησης, δηλαδή χρησιμοποιείται κυρίως όταν χρειάζεται να βρεθεί το ελάχιστο ή το μέγιστο κάποιου μεγέθους.
3. Η μέθοδος της Άπληστης προσέγγισης προχωρά με βάση σταδιακές επιλογές που αφορούν το βέλτιστο κάθε βήματος, χωρίς μέριμνα για το τελικό βέλτιστο.

Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει

4. Η φιλοσοφία της τεχνικής _____ είναι από την αρχή να επιλύονται τα μικρότερα προβλήματα και σταδιακά να επιλύονται τα μεγαλύτερα ως σύνθεση των απλούστερων.
5. Η μέθοδος της Άπληστης προσέγγισης προχωρά με βάση σταδιακές επιλογές που αφορούν το _____ κάθε βήματος, χωρίς μέριμνα για το τελικό _____.
6. Η _____ μέθοδος, μπορεί να τυποποιηθεί σύμφωνα με την παραδοχή ότι σε κάθε βήμα γίνεται επιλογή της τρέχουσας βέλτιστης επιλογής και υ-

πάρχει επιβεβαίωση ότι αυτή η προσέγγιση εγγυάται τη συνολική βέλτιστη λύση.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1: λάθος	4: Δυναμικού προγραμματισμού
2: σωστό	5: βέλτιστο
3: σωστό	6: άπληστη

4.5 Προβληματισμοί και θέματα προς συζήτηση

- Είναι χρήσιμο να τονισθεί η ιδιαίτερη χρησιμότητα των τεχνικών σχεδίασης των αλγορίθμων και να ζητηθεί από τους μαθητές να αναφερθούν σε παραδείγματα από την καθημερινή τους ζωή, όπου θα μπορούσαν να εντάξουν σε κατάλληλη τεχνική επίλυσης.
- Να συζητηθεί το πρόβλημα της ταξινόμησης με έμφαση στην ύπαρξη διαφορετικών αλγορίθμων και στις διαφορές και ιδιαιτερότητές τους.

4.6 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

4.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

1: Δες παράγραφο 4.1 του βιβλίου (αρχή παραγράφου).

2: Δες παράγραφο 4.1 του βιβλίου (τελευταία παράγραφος)

3: Παρατηρώντας τον τρόπο με τον οποίο οι διάφορες τεχνικές υλοποιούνται παρατηρούμε ότι υπάρχουν δύο προσεγγίσεις: η «από πάνω προς τα κάτω» (ανάλυση σε μικρότερα υποπροβλήματα) και η «από κάτω προς τα επάνω» (σύνθεση από μικρότερα υποπροβλήματα).

4: Δες παράγραφο 4.3 του βιβλίου.

5: Δες παράγραφο 4.4 του βιβλίου.

6: Δες παράγραφο 4.3 του βιβλίου.

7: Πρέπει να θεωρηθεί ένας πίνακας Ν θέσεων αν πρόκειται να υπολογισθεί το Ν!. Σε κάθε θέση του πίνακα τοποθετούμε το παραγοντικό της αντίστοιχης τιμής. Να συζητηθεί ότι αυτή η μέθοδος δεν μπορεί να εφαρμοσθεί για μεγάλο Ν, λόγω του μεγέθους του ακεραίου (τέσσερις χαρακτήρες).

8: Ο υπολογισμός της ακέραιας δύναμης ενός πραγματικού αριθμού, a^b, όπου το a είναι ακέραιος και b πραγματικός παρουσιάσθηκε στο Κεφάλαιο 3. Ακολουθεί ένας νέος απλός (ίσως απλοϊκός) τρόπος με χρήση της φιλοσοφίας "από κάτω προς τα επάνω" (δυναμικός προγραμματισμός), όπου η βασική ιδέα είναι η αποθήκευση των προηγούμενων δυνάμεων του αριθμού (1, 2, ..., b-1) μέχρι την "από κάτω προς τα επάνω" κατάληξη στη δύναμη b.

```
Αλγόριθμος Δύναμη
Δεδομένα // a, b //
power[1] ← 1
Για i από 1 μέχρι b
    power[i] ← power[i]*a
Τέλος_επανάληψης
Αποτελέσματα // power [b] //
Τέλος Δύναμη3
```

Ο αλγόριθμος αυτός μπορεί να βελτιωθεί με χρήση ενδιάμεσων αποτελεσμάτων δυνάμεων χωρίς να αποθηκεύονται όλες οι δυνάμεις. Συγκεκριμένα ο πίνακας power θα αποθηκεύει το αποτέλεσμα της ύψωσης στο τετράγωνο της προηγούμενης θέσης του πίνακα. Για παράδειγμα, αν θα υπολογιζόταν οι δυνάμεις του 2, τότε ο πίνακας θα είχε την τιμή 1 στη θέση 0, την τιμή 2 στη θέση 1, την τιμή 4 στη θέση 2, την τιμή 16 στη θέση 3 κ.ο.κ. όπως φαίνεται κα στο επόμενο σχήμα.

	0	1	2	3	4	...
	1	2	4	16	256	...
Δύναμη του 2	0η	1η	2η	4η	8η	...

Για τον υπολογισμό κάποιας δύναμης απαιτούνται οι κατάλληλες "από κάτω" θέσεις του πίνακα. Για την ανεύρεση του 2^7, αρκούν οι αποθηκευμένες δυνάμεις

μέχρι και την 4η δύναμη του 2, διότι $2^7 = 2^4 * 2^2 * 2^1$. Ο αλγόριθμος που ακολουθεί παρουσιάζει τον υπολογισμό του a^b με τη νέα αυτή ευφυέστερη τεχνική που βασίζεται στο δυναμικό προγραμματισμό.

```
Αλγόριθμος Δύναμη
Δεδομένα // a, b //
power[1] ← a
i ← 1
pow ← 1
Οσο power < b επανάλαβε
    i ← i+1
    pow ← 2*pow
    power[i] ← power[i-1]*power[i-1]
Τέλος_επανάληψης
used ← 0
result ← 1
Οσο used < b επανάλαβε
    Αν used+pow <= n τότε
        result ← result*power[i]
        used ← used+pow
        pow ← pow/2
    Τέλος_αν
    i ← i-1
Τέλος_επανάληψης
Αποτελέσματα \\ result \\
Τέλος Δύναμη
```

4.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

ΔΤ1.

Μπορεί να προταθεί βελτίωση με μετακίνηση της ελάχιστης τιμής στην πρώτη θέση μετά από ένα πρώτο κύκλο εκτέλεσης. Αυτή η τοποθέτηση της ελάχιστης τιμής στην πρώτη θέση του πίνακα προϋποθέτει την ανταλλαγή του στοιχείου που υπήρχε στην πρώτη θέση με τη θέση του πίνακα που είχε την ελάχιστη τιμή.

Επομένως ο δεύτερος κύκλος εκτέλεσης ξεκινά από τη δεύτερη θέση του πίνακα Ο αλγόριθμος Ανταλλαγή(M[1],M[pos]) πραγματοποιεί την ανταλλαγή των θέσεων M[1] και M[pos]:

```
Αλγόριθμος Δύο_Μικρότεροι
Δεδομένα // M //
low1 ← M[1]
pos ← 1
Για i από 2 μέχρι 50
    Αν M[i] < low1 τότε
        low1 ← M[i]
        pos ← i
    Τέλος_αν
Τέλος_επανάληψης
Αντιμετάθεσε(M[1],M[pos])
low2 ← M[2]
Για i από 3 μέχρι 50
    Αν M[i] < low2 τότε low2 ← M[i]
Τέλος_επανάληψης
Αποτελέσματα // low1 , low2 //
Τέλος Δύο_Μικρότεροι
```

ΔΤ2.

Εδώ μπορεί να χρησιμοποιηθεί ο αλγόριθμος της σειριακής αναζήτησης που παρουσιάσθηκε στην Παράγραφο 3.6 του βιβλίου και ο αλγόριθμος της Δυαδικής αναζήτησης που παρουσιάσθηκε στην Παράγραφο 4.3 του βιβλίου και δόθηκε αναλυτικά στο Παράδειγμα 2 του Κεφαλαίου 4 του τετραδίου του μαθητή. Η δυαδική αναζήτηση προϋποθέτει την ταξινόμηση του πίνακα με τα μουσεία της πόλης.

ΔΤ3.

Χρειάζεται να παρακολουθήσετε στην τάξη τον αλγόριθμο που δίνεται και να σχολιάσετε την πορεία του. Ο αλγόριθμος εντάσσεται στην κατηγορία της άπληστης μεθόδου αφού σε κάθε βήμα γίνεται επιλογή της βέλτιστης θέσης με στόχο τη συνολική βέλτιστη λύση.

ΔΤ4.

Η άσκηση αυτή αντιστοιχεί στο γνωστό πρόβλημα του περιοδεύοντος πωλητή. Οπως είναι γνωστό, το πρόβλημα αυτό γενικά είναι δυσχείριστο (εκτός ειδικών περιπτώσεων). Στο σημείο αυτό μπορεί να γίνει μία ευριστική επίλυση με γραφικό τρόπο στον πίνακα υιοθετώντας μία άπληστη τεχνική.

➤ Στο σπίτι

ΔΣ1.

Το πρόβλημα αυτό μπορεί να επιλυθεί με τη μέθοδο Διαίρει και Βασίλευε ξεκινώντας με άνω όριο το 10 και κάτω όριο το 0 αφού είναι γνωστά τα όρια του αριθμού που θα δοθεί.

```
Αλγόριθμος Μαντεύω_Αριθμό
Δεδομένα // number //
low ← 0
high ← 10
found ← 0
Διάβασε number
Οσο low <= high επανάλαβε
    mid ← (low + high)/2
    Αν mid < number τότε
        low ← mid+1
    αλλιώς_αν mid > number τότε
        high ← mid-1
    αλλιώς
        found ← 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // found //
Τέλος Μαντεύω_Αριθμό
```

ΔΣ2.

Χρειάζεται να ακολουθηθεί ο αλγόριθμος της δυαδικής αναζήτησης που παρουσιάσθηκε στην Παράγραφο 4.3 του βιβλίου και δόθηκε αναλυτικά στο Παράδειγμα 2 του Κεφαλαίου 4 του τετραδίου του μαθητή.

ΔΣ3.

Ο αλγόριθμος υπολογίζει τη δύναμη ακεραίου σε ακέραιο x^n

ΔΣ4.

Χρειάζεται να ακολουθηθεί ο αλγόριθμος της δυαδικής αναζήτησης που παρουσιάσθηκε στην Παράγραφο 4.3 του βιβλίου και δόθηκε αναλυτικά στο Παράδειγμα 2 του Κεφαλαίου 4 του τετραδίου του μαθητή.

ΔΣ5.

Χρειάζεται να ταξινομηθούν οι βαθμοί κάθε χώρας με χρήση κάποιου από τους αλγορίθμους ταξινόμησης που είναι ήδη γνωστοί. Εστω ότι ο πίνακας VATH έχει ταξινομηθεί. Θεωρούμε ότι συνολικά στο διαγωνισμό παίρνουν μέρος 20 Ευρωπαϊκές χώρες. Στη συνέχεια πρέπει να υπάρχει το ακόλουθο τμήμα του αλγορίθμου που θα επιλέγει τις χώρες ώστε το άθροισμα της βαθμολογίας όλων των τραγουδιών που θα προχωρήσουν στη δεύτερη φάση να είναι μικρότερο από 1000 βαθμούς:

```
SUM ← VATH[20]
Για i από 19 μέχρι 1 με βήμα -1
    Αν SUM+VATH[i] < 1000 τότε
        SUM ← SUM+VATH[i]
    αλλιώς
        stop ← i-1
        έξοδος από βρόχο
    Τέλος_αν
Τέλος_επανάληψης
```

Η μεταβλητή stop θα δείχνει τη θέση από το τέλος του πίνακα που θα εκφράζει την τελευταία χώρα σε βαθμολογία που θα συμμετάσχει στο διαγωνισμό αφού ο πίνακας είναι ταξινομημένος σε αύξουσα τάξη.

ΔΣ6.

Εξηγήθηκε στην ΔΤ5 παραπάνω.

4.9 Συμπληρωματικά στοιχεία

Δίνουμε εδώ μερικά επιπλέον στοιχεία από την ύλη του κεφαλαίου.

4.9.1 Γρήγορη ταξινόμηση

Αρκετές μέθοδοι ταξινόμησης βασίζονται στη φιλοσοφία διαίρει και βασίλευε. Μία μέθοδος ταξινόμησης αυτού του τύπου είναι ιδιαίτερα αποτελεσματική και για το λόγο αυτό ονομάσθηκε ***γρήγορη ταξινόμηση*** (quicksort). Η γρήγορη ταξινόμηση στηρίζεται στην παρατήρηση ότι είναι προτιμότερο οι ανταλλαγές να γίνονται μεταξύ απομακρυσμένων στοιχείων ενός πίνακα, παρά μεταξύ στοιχείων γειτονικών θέσεων, όπως συμβαίνει στην περίπτωση των μεθόδων ταξινόμησης με ευθεία ανταλλαγή (bubblesort) και με ευθεία εισαγωγή. Ας υποτεθεί ότι ο πίνακας αποτελείται από n στοιχεία που είναι ταξινομημένα κατά φθίνουσα τάξη της τιμής των κλειδιών τους. Τα στοιχεία αυτά είναι δυνατόν να ταξινομηθούν με n/2 ανταλλαγές μόνο, αντιμεταθέτοντας αρχικά το πιο αριστερό με το πιο δεξιό στοιχείο και προχωρώντας σταδιακά και από τα δύο άκρα του πίνακα προς τα μεσαία στοιχεία. Φυσικά αυτό μπορεί να γίνει όταν είναι εκ των προτέρων γνωστό ότι τα στοιχεία είναι ταξινομημένα σε φθίνουσα τάξη. Ωστόσο το παράδειγμα αυτό μας διδάσκει κάτι χρήσιμο.

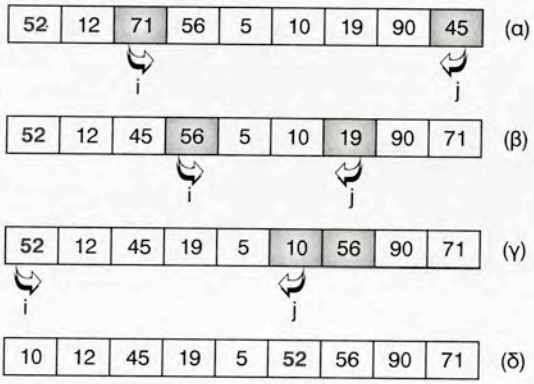
Στη συνέχεια δίνεται ο αλγόριθμος Γρήγορη_Ταξινόμηση. Η βασική ιδέα του αλγορίθμου αυτού είναι η λήψη του πρώτου στοιχείου του πίνακα και η μετακίνησή του στη θέση, όπου τελικά θα αποθηκευθεί ανάλογα με την τιμή του κλειδιού του. Ταυτόχρονα με την αποθήκευση του στοιχείου αυτού στην τελική του θέση γίνεται αναδιάταξη των υπόλοιπων στοιχείων, έτσι ώστε να μην υπάρχει κανένα μικρότερο κλειδί προς τα δεξιά του και κανένα μεγαλύτερο κλειδί προς τα αριστερά του. Έτσι, το πρώτο αυτό στοιχείο παίζει το ρόλο του άξονα και ο πίνακας έ-

χει διαμερισθεί κατά τέτοιο τρόπο, ώστε το αρχικό πρόβλημα έχει να αναχθεί σε δύο απλούστερα προβλήματα, δηλαδή στην ανεξάρτητη ταξινόμηση των δύο υποπινάκων. Η απόσταση μεταξύ του διαμερισμού αυτού και της τελικής ταξινόμησης είναι πολύ μικρή. Μετά το διαμερισμό του πίνακα, ο ίδιος αλγόριθμος εφαρμόζεται στους δύο υποπίνακες, έπειτα στους υποπίνακες των υποπινάκων, κοκ., μέχρις ότου ο κάθε υποπίνακας να αποτελείται από ένα μόνο στοιχείο. Προφανώς, στο σημείο αυτό ο αρχικός πίνακας έχει ταξινομηθεί.

```
Αλγόριθμος Γρήγορη_ταξινόμηση
Δεδομένα  // table, left, right //
Αν right > left τότε
    i ← left
    j ← right+1
    pivot ← table[left]
    Αρχή_επανάληψης
        Αρχή_επανάληψης
            i ← i+1
        Μέχρις ότου table[i] ≥ pivot
        Αρχή_επανάληψης
            j ← j-1
        Μέχρις ότου table[j] ≤ pivot
        Αν i < j τότε Αντιμετάθεσε(table[i], table[j])
    Μέχρις ότου i ≥ j
    Αντιμετάθεσε(table[left], table[j])
    Γρήγορη_ταξινόμηση(table, left, j-1)
    Γρήγορη_ταξινόμηση(table, j+1, right)
Τέλος_Αν
Αποτελέσματα // table //
Τέλος Γρήγορη_ταξινόμηση
```

Παράδειγμα. Έστω ότι δίνεται για ταξινόμηση ο γνωστός από το προηγούμενο κεφάλαιο πίνακας με εννέα στοιχεία (Σχ. 4.1α). Παίρνουμε το πρώτο στοιχείο του πίνακα ως άξονα, δηλαδή ισχύει pivot = 52 και συγκρίνουμε κάθε ένα από τα υπόλοιπα στοιχεία με το στοιχείο αυτό. Πιο συγκεκριμένα, ξεκινώντας από τα αριστερά αναζητούμε το πρώτο στοιχείο που είναι μεγαλύτερο από τον άξονα 52. Το στοιχείο αυτό είναι το 71 (δείκτης i). Κατόπιν, από τα δεξιά αναζητούμε το πρώτο στοιχείο που είναι μικρότερο από το 52, οπότε βρίσκουμε το 45 (δείκτης j). Στη συνέχεια εκτελούμε μία αντιμετάθεση των στοιχείων 71 και 45, οπότε προκύπτει η επόμενη μορφή του πίνακα, όπως φαίνεται στο σχήμα 4.1β. Η διαδικα-

σία συνεχίζεται σαρώνοντας από τα αριστερά για μεγαλύτερα του 52 στοιχεία και από τα δεξιά για μικρότερα του 52 στοιχεία. Έτσι, βρίσκουμε τα στοιχεία 56 (δείκτης i) και 19 (δείκτης j), τα οποία αντιμετατίθονται (Σχ. 4.1γ). Συνεχίζοντας τη σάρωση από τα αριστερά βρίσκουμε το στοιχείο 56 (δείκτης i) και από τα δεξιά το στοιχείο 10 (δείκτης j). Όμως στο σημείο αυτό ισχύει i > j, όποτε δεν εκτελείται αυτή η ανταλλαγή. Αντ'αυτής εκτελείται η αντιμετάθεση του στοιχείου του άξονα με το στοιχείο όπου σταμάτησε ο δείκτης j. Έτσι, ανταλλάσσονται τα στοιχεία 52 και 10, προκύπτει μία νέα μορφή της δομής που παρουσιάζεται στον επόμενο πίνακα (Σχ. 4.1δ).



Σχήμα 4.1 *Κατάσταση πίνακα κατά τις διαδοχικές συγκρίσεις και αντιμεταθέσεις κατά την πρώτη κλήση της διαδικασίας Γρήγορη_ταξινόμηση.*

Το αποτέλεσμα αυτών των συγκρίσεων είναι να υπάρχουν δύο υπο-πίνακες. Ο αριστερός υπο-πίνακας αποτελείται από πέντε στοιχεία, που είναι μικρότερα του 52. Το τελευταίο έχει τοποθετηθεί στην τελική του θέση στον πίνακα, και δεν πρόκειται να "ασχοληθούμε" στο μέλλον μαζί του. Στα δεξιά του 52 θεωρούμε το δεύτερο υπο-πίνακα, που αποτελείται από 3 στοιχεία με τιμές μεγαλύτερες του 52. Η κατάσταση αυτή παρουσιάζεται στο επόμενο σχήμα.



Σχήμα 4.2 *Κατάσταση πίνακα μετά την πρώτη κλήση της διαδικασίας Γρήγορη_ταξινόμηση.*

Στη συνέχεια καλείται αναδρομικά η διαδικασία Γρήγορη_ταξινόμηση για τον αριστερό υποπίνακα και για το δεξιό υποπίνακα.

4.9.2 Οπισθοδρόμηση

Μια πολύ σημαντική τεχνική σχεδίασης αλγορίθμων είναι η οπισθοδρόμηση. Η τεχνική αυτή κάνει χρήση αναδρομής, συχνά σε δομές γράφων και γι' αυτό θεωρείται προωθημένο θέμα.

Η τεχνική της ***οπισθοδρόμησης*** (backtracking) στηρίζεται στην αρχή ότι, αναζητώντας τη λύση μέσα σε ένα σύνολο πιθανών περιπτώσεων, αν ανακαλυφθεί ότι συνεχίζοντας στην ίδια πορεία δεν είναι δυνατόν να βρεθεί η λύση, τότε επανερχόμαστε σε ένα προηγούμενο σημείο και δοκιμάζουμε καινούριο δρόμο σε σχέση με την εξέταση των λύσεων. Η μέθοδος της οπισθοδρόμησης έχει εφαρμογές σε προβλήματα αναζήτησης σε γράφους, όπως για παράδειγμα στο γράφο του σχήματος 4.1 (βιβλίου μαθητή) που απεικονίζει τη σύνδεση πόλεων. Γενικά, η τεχνική αυτή μπορεί να συνοψισθεί στα εξής βήματα:

1) Ανίχνευση κάθε κόμβου του γράφου, με βάση κάποια συγκεκριμένη σειρά μέχρις ότου
   - ή βρεθεί η βέλτιστη λύση (ή γενικώς η λύση),
   - ή διαπιστωθεί ότι αποκλείεται η μέχρι τώρα πορεία, να δώσει τη λύση στη συνέχεια.
2) Στη δεύτερη περίπτωση, η διαδικασία επίλυσης "αναδιπλώνεται", ώστε να προσεγγίσει άλλο κόμβο που να μπορεί να οδηγήσει στη λύση.

Διάσχιση λαβυρίνθου

Ο λαβύρινθος είναι ένα σύνολο διαδρόμων που χωρίζονται από ψηλούς φράκτες, θάμνους ή τοίχους και συναντώνται σε διασταυρώσεις. Πρέπει, λοιπόν, εισερχόμενοι από τη μοναδική είσοδο του λαβυρίνθου να διατρέξουμε όλους τους διαδρόμους και να επιστρέψουμε στη μοναδική είσοδο/έξοδο. Για να μην εγκλωβισθούμε σε ένα λαβυρίνθο για πάντα (θεωρητικά), πρέπει να διατυπώσουμε ένα συστηματικό τρόπο διάσχισής του, δηλαδή έναν αλγόριθμο.

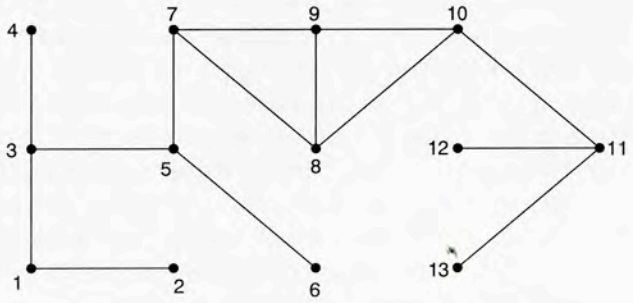
Το πρόβλημα του λαβυρίνθου έρχεται από το βάθος της ελληνικής μυθολογίας. Όπως είναι γνωστό, ο Ανδρόγειος, γυιός του βασιλιά Μίνωα κέρδισε νίκες στα Παναθήναια, αλλά οι αντίπαλοί του τον σκότωσαν. Ο Μίνως κήρυξε τον πόλεμο στην Αθήνα και την κυρίευσε. Οι Αθηναίοι αναγκάσθηκαν κάθε χρόνο να στέλνουν 7 αγόρια και 7 κορίτσια θυσία στο Μινώταυρο που ήταν εγκλωβισμένος στο λαβύρινθο. Ο Θησέας, γυιός του βασιλιά της Αθήνας Αιγαία, δέχθηκε να αντιμεπωπίσει το Μινώταυρο. Με τη βοήθεια της Αριάδνης, κόρης του Μίνωα, κατάφερε να σκοτώσει το Μινώταυρο και να βγει από το λαβύρινθο χρησιμοποιώντας το μίτο.



Σχήμα 4.3. *Ο κήπος του Hampton Court Palace*

Κατά την Αναγέννηση πολλοί βασιλιάδες διακοσμούσαν τους κήπους των ανακτόρων τους με λαβυρίνθους. Ο πιο φημισμένος λαβύρινθος είναι αυτός του Hampton Court Palace, που βρίκεται πολύ κοντά στο Λονδίνο και κατασκευάσθηκε κατά παραγγελία του Γουλιέλμου του 3ου της Οράγγης. Ο λαβύρινθος αυτός παρουσιάζεται στο σχήμα 4.3. Το πρόβλημα της εύρεσης μεθόδου για την διάσχιση ενός λαβυρίνθου απασχόλησε πολλούς μαθηματικούς, αλλά λύθηκε μόνο στο τέλος του 19ου αιώνα.

Δεδομένου ενός λαβυρίνθου, προχωρούμε σε μία αφαιρετική αναπαράστασή του, δηλαδή κατασκευάζουμε ένα γράφο. Οι κορυφές του γράφου αντιστοιχούν στις διασταυρώσεις όπου συναντώνται περισσότεροι από δύο διάδρομοι, ενώ οι συνδέσεις των κορυφών αντιστοιχούν στους διαδρόμους του λαβυρίνθου. Στο σχήμα 4.4 παρουσιάζεται ο γράφος του αντιστοιχεί στο λαβύρινθο του σχήματος 4.3.



Σχήμα 4.4

Έχοντας, λοιπόν, τον αντίστοιχο γράφο πρέπει να διατυπώσουμε ένα αλγόριθμο συστηματικής επίσκεψης των κορυφών του. Ένας αλγόριθμος που επιτυγχάνει αυτό το στόχο λέγεται "*αναζήτηση κατά βάθος*" (depth-first search) και στηρίζεται στη φιλοσοφία της οπισθοδρόμησης. Όπως εξηγήθηκε στην εισαγωγή της παραγράφου, ακολουθούμε μία διαδρομή μέχρι να διαπιστώσουμε, ότι δεν είναι δυνατόν να συνεχίσουμε από εκείνο σημείο, οπότε επιστρέφουμε σε μία προηγούμενη κορυφή, για να συνεχίσουμε από εκεί προς μία νέα κατεύθυνση.

Ο επόμενος αναδρομικός αλγόριθμος υλοποιεί την αναζήτηση κατά βάθος και ονομάζεται AKB, από τα αρχικά της μεθόδου. Ο αλγόριθμος αυτός θεωρεί ότι ο γράφος έχει n κορυφές και είναι αποθηκευμένος στη μνήμη του υπολογιστή με τη μέθοδο του πίνακα γειτνίασης. Επίσης, χρησιμοποιείται ένας βοηθητικός μονοδιάστατος πίνακας mark, που αποτελείται από n στοιχεία τύπου Αληθής/Ψευδής. Ο πίνακας αυτός χρησιμοποιείται για να σημειώνουμε, αν έχουμε επισκεφθεί κάποια κορυφή. Με απλά λόγια, κάθε στοιχείο του πίνακα, που προφανώς αντιστοιχεί σε κάποια κορυφή, αρχικά έχει την τιμή Ψευδής και λαμβάνει την τιμή Αληθής αν επισκεφθούμε την αντίστοιχη κορυφή.

```
Αλγόριθμος AKB(k)
Δεδομένα // a πίνακας γειτνίασης γράφου, n //
mark(k)=Αληθής
Εμφάνισε k
Για j από 1 μέχρι n
    Αν a(k,j)=1 και mark(j) = Ψευδής τότε AKB(j)
Τέλος_επανάληψης
Αποτελέσματα // Λίστα κόμβων γράφου //
Τέλος AKB
```

Σε σχέση με το γράφο του σχήματος, η διαδικασία αρχίζει με την κλήση ΑΚΒ(1), δηλαδή από την κορυφή 1, που μαρκάρεται "ως επισκεφθείσα" θέτοντας mark(1)=Αληθής. Κατόπιν επιλέγεται η κορυφή 2 που είναι γειτονική της κορυφής 1 (επειδή ισχύει a(1,2)=1) και καλείται αναδρομικά η διαδικασία ΑΚΒ(2). Έτσι η κορυφή 2 μαρκάρεται ως επισκεφθείσα (δηλαδή τίθεται mark(2)=Αληθής), και η μέθοδος συνεχίζει από την κορυφή 2. Όμως από την κορυφή αυτή δεν υπάρχει άλλη γειτονική με μεγαλύτερη (του 2) επιγραφή, οπότε ο έλεγχος επιστρέφει (backtrack) στην κορυφή 1. Κατόπιν συνεχίζουμε με την κορυφή 3 καλώντας ΑΚΒ(3), επειδή ισχύει a(1,3)=1 και από εκεί στην κορυφή 4 εκτελώντας ΑΚΒ(4). Όμως από την κορυφή 4 ο έλεγχος επιστρέφει (οπισθοδρομεί) στην κορυφή 3, οπότε εκτελείται η ΑΚΒ(5). Έτσι ο αλγόριθμος συνεχίζει με τη λογική αυτή, οπότε επισκεπτόμαστε τις κορυφές με τη σειρά 1, 2, 3, 4, 5, 6, 5, 7, 8, 9, 10, 11, 12, 13, που ταυτίζεται με τη σειρά απόδοσης της τιμής "Αληθής" στις αντίστοιχες θέσεις του πίνακα mark. Ωστόσο κατά την εκτέλεση του αλγορίθμου συμβαίνει ένας αριθμός οπισθοδρομήσεων. Πιο συγκεκριμένα, οπισθοδρομήσεις συμβαίνουν όταν φθάνουμε σε κορυφές από όπου δεν είναι δυνατόν να συνεχίσουμε (δηλαδή τις κορυφές 2, 4, 6, 12 και 13).

Ο αλγόριθμος Αναζήτησης κατά βάθος σε ένα γράφο είναι ένας γενικός αλγόριθμος και αποτελεί βασικό συστατικό πολλών συνθετότερων αλγορίθμων που σχετίζονται με γράφους. Έτσι, αυτός καθ'αυτός ο αλγόριθμος όπως διατυπώθηκε προηγουμένως δεν δίνει τη λύση στο πρόβλημα της διάσχισης του λαβύρινθου, αλλά αποτελεί τη βάση για τη διατύπωση ενός ειδικού αλγόριθμου. Πρακτικά, αν κάποιος πρέπει να διασχίσει ένα λαβύρινθο, πρέπει να τοποθετεί διάφορα σημάδια όταν φθάνει σε μία διασταύρωση, ώστε να γνωρίζει

- αν έχει ξαναπεράσει τους διαδρόμους που ξεκινούν από τη διασταύρωση αυτή, και μάλιστα κατά την ίδια φορά που πιθανώς ακολουθήσει τώρα, και
- ποιος είναι ο διάδρομος απ' όπου για πρώτη φορά έφθασε στη συγκεκριμένη διασταύρωση, ώστε από εκεί να επιστρέψει προς την έξοδο.

Έτσι, ακολουθώντας αυτήν την τεχνική, κάποιος δεν θα χαθεί μέσα στο λαβύρινθο αλλά με βεβαιότητα θα διασχίσει κάθε διάδρομο του λαβυρίνθου δύο μόνο φορές, από μία κατά τις δύο κατευθύνσεις.

Κεφάλαιο 5

Ανάλυση αλγορίθμων

5.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να κατανοήσουν οι μαθητές τις τεχνικές ανάλυσης των αλγορίθμων και να εξοικειωθούν με την έννοια της πολυπλοκότητας των αλγορίθμων. Ο διδακτικός στόχος είναι να παρουσιασθούν οι λειτουργίες ανάλυσης των αλγορίθμων για συγκεκριμένα είδη προβλημάτων και να γίνουν ικανοί οι μαθητές να αναγνωρίζουν την πολυπλοκότητα των κυριοτέρων ειδών αλγορίθμων.

5.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να κατανοήσουν τις τεχνικές ανάλυσης των αλγορίθμων.
- να αναγνωρίσουν τη σημασία της επίδοσης και της αποδοτικότητας των αλγορίθμων.
- να εξοικειωθούν με διαδικασίες ελέγχου της ορθότητας ενός αλγορίθμου.
- να διακρίνουν και να βρίσκουν το είδος πολυπλοκότητας των αλγορίθμων.
- να εξειδικεύουν την ανάλυση αλγορίθμων για λειτουργίες σε συχνά χρησιμοποιούμενα είδη προβλημάτων.

5.3 Οδηγίες – επισημάνσεις

Ιδιαίτερη έμφαση και προσοχή πρέπει να δοθεί στα παρακάτω θεματικά αντικείμενα:

Είναι απαραίτητο να γίνουν αρκετά παραδείγματα για την ανάλυση και την πολυπλοκότητα των αλγορίθμων. Να επισημανθεί ο ρόλος της χειρότερης "στιγμής" (περίπτωσης) ενός αλγορίθμου για την αποτύπωση της πολυπλοκότητας και της αποδοτικότητάς του.

5.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- να κατανοήσουν τις τεχνικές ανάλυσης των αλγορίθμων.
- να αναγνωρίσουν τη σημασία της επίδοσης και της αποδοτικότητας των αλγορίθμων.
- να εξοικειωθούν με τις διαδικασίες ελέγχου της ορθότητας ενός αλγορίθμου.

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Επίδοση αλγορίθμων:** τρόπος εκτίμησης της επίδοσης (ή αποδοτικότητας) των αλγορίθμων.
- **Χειρότερη περίπτωση ενός αλγορίθμου:** παρουσίαση της χειρότερης περίπτωσης ενός αλγορίθμου σε σχέση με το μέγιστο κόστος εκτέλεσης του αλγορίθμου μετρήσιμο σε υπολογιστικούς πόρους.
- **Μέγεθος εισόδου ενός αλγορίθμου:** καθορισμός μεταβλητών που να εκφράζουν το μέγεθος (size) του αλγορίθμου και να απεικονίζουν τους διαφορετικούς συνδυασμούς τιμών που κρίνουν τη συμπεριφορά ενός αλγορίθμου.
- **Χρόνος εκτέλεσης προγράμματος ενός αλγορίθμου:** υπολογισμός του χρόνου εκτέλεσης και της επίδοσης ενός αλγορίθμου με βάση τον αριθμό των πράξεων που θα εκτελεσθούν.
- **Αποδοτικότητα αλγορίθμων:** Επιλογή του αποδοτικότερου αλγορίθμου με βάση το μικρότερο χρόνο εκτέλεσης για την επίλυση ενός προβλήματος.

- **Ορθότητα αλγορίθμων:** τυποποίηση και απόδειξη για την ορθότητα και εγκυρότητα των προτεινόμενων αλγορίθμων.

Περιεχόμενα πρακτικής εφαρμογής

- **Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή:** Θα πρέπει να διδαχθούν οι παράγραφοι 5.1 μέχρι και 5.2 από το βιβλίο του μαθητή. Να απαντηθούν οι ερωτήσεις 1-8 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.
- **Δραστηριότητες από το τετράδιο του μαθητή:** Θα πρέπει να διδαχθούν τα παραδείγματα 1 και 2 από το αντίστοιχο κεφάλαιο από το τετράδιο του μαθητή. Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ1 ή ΔΤ2 στην τάξη και μία από τις δραστηριότητες ΔΣ1, ΔΣ2 ή ΔΣ3 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Για την κατανόηση της επίδοσης ενός αλγορίθμου χρειάζεται να απαντηθεί το πρωταρχικά ερώτημα για τον τρόπο υπολογισμού του χρόνου εκτέλεσης ενός αλγορίθμου.
2. Μία βασική πράξη είναι η επανάληψη ενός βρόχου τόσες φορές όσες δίνονται από το συγκεκριμένο όριο τιμών μίας μεταβλητής.
3. Το μέγεθος του αλγορίθμου εκφράζεται από κάποια ή κάποιες μεταβλητές και γενικά τα δεδομένα συνιστούν το μέγεθος της εισόδου ενός αλγορίθμου.
4. Οι βρόχοι επανάληψης δεν αποτελούν σημαντικό κριτήριο για το χαρακτηρισμό της επίδοσης ενός αλγορίθμου.
5. Για να έχει έννοια κάθε σύγκριση μεταξύ δύο προγραμμάτων αλγορίθμων θα πρέπει και τα δύο προγράμματα να έχουν συνταχθεί στην ίδια γλώσσα προγραμματισμού και να χρησιμοποιείται η ίδια υπολογιστική πλατφόρμα
6. Δεν είναι απαραίτητο να συνοδεύεται το στάδιο της ανάλυσης και της σχεδίασης του αλγορίθμου από τον αντίστοιχο έλεγχο της ορθότητάς του.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση

7. Για την κατανόηση της επίδοσης ενός αλγορίθμου χρειάζεται να απαντηθεί ένα σύνολο ερωτημάτων. Ένα από τα πρωταρχικά ερωτήματα που προκύπτουν είναι:

   α) πώς υπολογίζεται ο χρόνος για την ανάγνωση ενός αλγορίθμου;

   β) πώς μπορούν να συγκριθούν μεταξύ τους οι διάφοροι τρόποι εξόδου δεδομένων για ένα αλγόριθμο;

   γ πώς μπορεί να γνωρίζει κανείς αν ένας αλγόριθμος είναι βέλτιστος;

   δ) πώς μπορεί να επιλεγεί ο κατάλληλος υπολογιστής για έναν αλγόριθμο ;

8. Μία βασική πράξη μπορεί να είναι:

   α) ανάθεση γλώσσας προγραμματισμού,

   β) σύγκριση μεταξύ δύο μεταβλητών

   γ) επιλογή 10 μεταβλητών

   δ) βρόχος επανάληψης

9. Πρέπει να υπάρχει κάποια ή κάποιες μεταβλητές που να εκφράζουν το μέγεθος του αλγορίθμου. Γενικά, το μέγεθος της εισόδου ενός αλγορίθμου συνίσταται από:

   α) αριθμό εντολών β) δεδομένα

   γ) αριθμό γραμμών δ) αριθμό βρόχων

10. Για να μπορεί να γίνει σύγκριση μεταξύ δύο προγραμμάτων αλγορίθμων θα πρέπει να ικανοποιείται μία από τις παρακάτω προϋποθέσεις:

    α) και τα δύο προγράμματα να έχουν συνταχθεί στην ίδια γλώσσα προγραμματισμού,

    β) να έχει χρησιμοποιηθεί ο ίδιος αριθμός βρόχων σε κάθε πρόγραμμα

    γ) ο μεταφραστής της γλώσσας προγραμματισμού να είναι διαφορετικός για κάθε πρόγραμμα

    δ) να μη χρησιμοποιείται η ίδια υπολογιστική πλατφόρμα,

11. Στο πρόβλημα της ταξινόμησης το μέγεθος της εισόδου του αλγορίθμου είναι
    α) το πλήθος των αντικειμένων που θα ταξινομηθούν
    β) ο αριθμός των βρόχων επανάληψης
    γ) ο αριθμός των βασικών πράξεων
    δ) α αριθμός των συγκρίσεων που θα γίνουν
12. Η απόδειξη της ορθότητας ενός αλγορίθμου θα πρέπει να περιλαμβάνει κάποια από τις παρακάτω συνθήκες:
    α) υπόθεση ότι κάθε τερματισμός της εκτέλεσης του αλγορίθμου οδηγεί σε αποδεκτά αποτελέσματα
    β) απόδειξη ότι θα υπάρξει τερματισμός της εκτέλεσης του αλγορίθμου
    γ) εκτίμηση του αριθμού των βρόχων επανάληψης του αλγορίθμου
    δ) απόδειξη ότι ο αλγόριθμος εξαρτάται από το συγκεκριμένο υπολογιστή.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1:σωστό	7: γ
2:λάθος	8: β
3:σωστό	9: β
4:λάθος	10: α
5:σωστό	11: α
6:λάθος	12: β

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

- να διακρίνουν και να βρίσκουν το είδος πολυπλοκότητας των αλγορίθμων.
- να εξειδικεύουν την ανάλυση αλγορίθμων για λειτουργίες σε συχνά χρησιμοποιούμενα είδη προβλημάτων.

Χώρος υλοποίησης μαθήματος

τάξη

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες.

Περιεχόμενα θεωρητικής παρουσίασης

- **Πολυπλοκότητα αλγορίθμων.** Κατανόηση της σημασίας της πολυπλοκότητας για την αξιολόγηση αλγορίθμων.
- **Ανάλυση αλγορίθμων.** Εξήγηση του σημαντικού ρόλου της ανάλυσης αλγορίθμων για τη μελέτη της επίδοσης των αλγορίθμων.
- **Είδη αλγορίθμων.** Παρουσίαση της πολυπλοκότητας των κυριοτέρων ειδών αλγορίθμων που έχουν παρουσιασθεί σε προηγούμενα κεφάλαια.
- **Επανάληψη των εννοιών που διδάχθηκαν – Ανακεφαλαίωση**

Περιεχόμενα πρακτικής εφαρμογής

- **Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή.** Θα πρέπει να διδαχθούν οι παράγραφοι 5.3 μέχρι και 5.4 από το βιβλίο του μαθητή. Να απαντηθούν οι ερωτήσεις 9-13 του αντίστοιχου κεφαλαίου από το βιβλίο του μαθητή.
- **Δραστηριότητες από το τετράδιο του μαθητή.** Θα πρέπει να διδαχθούν τα παραδείγματα 3 και 4 του αντίστοιχου κεφαλαίου από το τετράδιο του μαθητή. Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ3 ή ΔΤ4, και μία από τις δραστηριότητες ΔΣ4 ή ΔΣ5.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Ένας τρόπος εκτίμησης της επίδοσης ενός αλγορίθμου είναι ο θεωρητικός ή αλλιώς «εκ των προτέρων» όπου χρησιμοποιείται μία μεταβλητή που εκφράζει το μέγεθος του προβλήματος.

2. Αν η πολυπλοκότητα ενός αλγορίθμου είναι f(n), τότε λέγεται ότι είναι τάξης O(g(n)) αν υπάρχουν δύο θετικοί ακέραιοι c και n_0, έτσι ώστε για κάθε $n \geq n_0$ να ισχύει: $|f(n)| \leq c\,|g(n)|$.
3. Η πολυπλοκότητα της Ταξινόμησης ευθείας ανταλλαγής είναι O(n).
4. Πολυωνυμικοί λέγονται οι αλγόριθμοι με πολυπλοκότητα που φράσσεται από κάτω με μία πολυωνυμική έκφραση.
5. Η πολυπλοκότητα της Δυαδικής αναζήτησης είναι O(logn).
6. O(n) είναι η γραμμική πολυπλοκότητα που θεωρείται η καλύτερη επίδοση για έναν αλγόριθμο που πρέπει να εξετάσει ή να δώσει στην έξοδο n στοιχεία.

Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει

7. Ο _____ τρόπος εκτίμησης της επίδοσης ενός αλγορίθμου εισάγει μία μεταβλητή που εκφράζει το μέγεθος του προβλήματος, ώστε η μέτρηση της αποδοτικότητας του αλγόριθμου να ισχύει για οποιοδήποτε σύνολο δεδομένων και ανεξάρτητα από υποκειμενικούς παράγοντες.
8. _____ λέγονται οι αλγόριθμοι με πολυπλοκότητα που φράσσεται από επάνω με μία πολυωνυμική έκφραση.
9. Όταν σε έναν αλγόριθμο κάθε εντολή του προγράμματος του εκτελείται μία φορά ή το πολύ μερικές μόνο φορές ο αλγόριθμος είναι _____ πολυπλοκότητας.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1:σωστό	4:λάθος	7: θεωρητικός
2:λάθος	5:σωστό	8: πολυωνυμικός
3:λάθος	6:σωστό	9:σταθερής

5.5 Προβληματισμοί και θέματα προς συζήτηση

- Να γίνει σαφές ότι για κάθε αλγόριθμο μπορεί να προκύψει μία συνάρτηση που να δίνει επακριβώς τον αριθμό των συγκρίσεων, τον αριθμό των αντιμεταθέσεων κοκ. Ωστόσο, σημαντικό είναι η **ποιοτική** συμπεριφορά του αλγο-

ρίθμου, που δηλώνεται με το συμβολισμό Ο. Είναι καλό ο προγραμματιστής να προσέχει ώστε να επιτύχει την πιο αποτελεσματική κωδικοποίηση ενός αλγορίθμου, αλλά αν ο αλγόριθμος δεν είναι ο καλύτερος δυνατός σε σχέση με το δεδομένο πρόβλημα, τότε η προσπάθεια βελτιστοποίησης του κώδικα είναι μάταιη.

- Σημαντικό, επίσης, ζήτημα είναι η απόδειξη ορθότητας, που αποτελεί μία επαγγελματική πρακτική κατά την κατασκευή μεγάλων και σύνθετων προγραμμάτων. Η ανάπτυξη ενός προγράμματος δεν τελειώνει με την πρώτη απόπειρα κωδικοποίησης αλλά περνά από πολλές φάσεις ελέγχου. Σε όλες τις φάσεις απαραίτητη είναι η ύπαρξη τεκμηρίωσης.

5.6 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

5.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

1. Δες αρχή παραγράφου 5.1
2. Δες παράγραφο 5.1
3. Δες παράγραφο 5.1
4. Δες παράγραφο 5.1
5. Δες παράγραφο 5.1
6. Αποδοτικότητα αλγορίθμου είναι μία έννοια που εκφράζει τη χρήση των υπολογιστικών πόρων (χώρος και χρόνος) από τον αλγόριθμο και αποτελεί το γενικότερο κριτήριο σύγκρισης αλγορίθμων. Έτσι αν ο αλγόριθμος Β δίνει το ίδιο αποτέλεσμα με τον αλγόριθμο Α, αλλά σε λιγότερο χρόνο ή με χρήση λιγότερης μνήμης, τότε είναι αποδοτικότερος του Α.
7. Δες παράγραφο 5.2.
8. Δες παράγραφο 5.2.

9. Δες παράγραφο 5.3.
10. Δες παράγραφο 5.3.
11. Δες παράγραφο 5.3.2
12. Δες παράγραφο 5.3.2 και δραστηριότητες ΔΤ3 κα ΔΤ4 κεφαλαίου 3
13. Δες παράγραφο 5.4.

5.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

ΔΤ1.

1. Το κομμάτι του αλγορίθμου είναι ένας απλός βρόχος. Ο αριθμός των πράξεων είναι n/2, άρα η πολυπλοκότητα είναι O(n).
2. Το κομμάτι είναι ένας διπλός/εμφωλευμένος βρόχος. Ο αριθμός των πράξεων προκύπτει με τον υπολογισμό του διπλού αθροίσματος

$$\sum_{i=1}^{n}\sum_{j=1}^{n}1=\sum_{i=1}^{n}n=n^2$$

Άρα, η πολυπλοκότητα είναι $O(n^2)$.

3. Παρόμοια περίπτωση με την προηγούμενη.

$$\sum_{i=1}^{n/2}\sum_{j=1}^{n}1=\sum_{i=1}^{n/2}n=n^2\ /\ 2$$

Άρα, η πολυπλοκότητα είναι $O(n^2)$

Γενικά, ισχύει ότι κάθε απλός βρόχος έχει γραμμική πολυπλοκότητα, κάθε διπλός βρόχος έχει τετραγωνική πολυπλοκότητα κοκ.

ΔΤ 2:

Απαιτείται ένας μονοδιάστατος πίνακας donation (έστω 30 θέσεων), όπου να είναι αποθηκεύονται τα διάφορα ποσά των δωρεών. Αυτός ο πίνακας πρέπει να σαρωθεί ώστε να αθροισθούν τα επιμέρους ποσά σε ένα συνολικό ποσό sum.

```
Αλγόριθμος Δωρεές_SOS
Δεδομένα // donation //
sum ← 0
Για i από 1 μέχρι 30
    sum ← sum+donation[i]
Τέλος_επανάληψης
Εκτύπωσε sum
Τέλος Δωρεές_SOS
```

Ο αλγόριθμος αυτός είναι ουσιαστικά μία γραμμική διαδικασία. Η πολυπλοκότητα είναι O(n).

ΔΤ 3:

Η παραλλαγή αυτή του αλγορίθμου υπολογισμού αριθμών Fibonacci είναι αρκετά πολύπλοκη στην κατανόηση των λειτουργιών της. Ωστόσο η ανάλυση της πολυπλοκότητάς της είναι εύκολη. Ουσιαστικά, το ειδικός βάρος του αλγορίθμου βρίσκεται στην επαναληπτική δομή "Επανάλαβε όσο n΄)΄0 ...". Μέσα στη δομή αυτή, το n υποδιαιρείται συνεχώς με την εντολή "n DIV 2". Έτσι ο αριθμός των επαναλήψεων του βρόχου αυτού είναι O(logn). Παρόμοια είναι και η εξήγηση στην άσκηση ΔΣ3 (δες στη συνέχεια), σχετικά με την πολυπλοκότητα του αλγορίθμου της δυαδικής αναζήτησης.

ΔΤ 4:

Έστω ότι τα δεδομένα αποθηκεύονται σε μία δομή δισδιάστατου πίνακα, όπου κάθε γραμμή αντιστοιχεί σε μία ταινία, ενώ μία στήλη του πίνακα κρατά τη χρονιά παραγωγής. Το πρόβλημα μπορεί να επιλυθεί απλά σαρώνοντας κάθε γραμμή του πίνακα και ελέγχοντας την τιμή της αντίστοιχης στήλης. Έτσι αν η τιμή είναι 1960, τότε τα στοιχεία της ταινίας/γραμμής θα δίνοναι στο χρήστη, αλλιώς ο έλεγχος θα προχωρεί στην επόμενη γραμμή. Μία τέτοια διαδικασία είναι γραμμική.

➤ ***Στο σπίτι***

ΔΣ 1:

1. Το κομμάτι του αλγορίθμου είναι ένας απλός βρόχος n=100 επαναλήψεων. Άρα, η πολυπλοκότητα είναι γραμμική O(n).
2. Όπως και στην περίπτωση (1) της δραστηριότητας ΔΤ1, το κομμάτι είναι ένας απλός βρόχος, όπου εκτελούνται n-1 επαναλήψεις. Άρα. η πολυπλοκότητα είναι O(n).
3. Παρόμοια περίπτωση με την προηγούμενη. Ο αριθμός των επαναλήψεων του απλού βρόχου είναι (n-1)/3, άρα. η πολυπλοκότητα είναι O(n).

ΔΣ 2:

Σύμφωνα (μεταξύ των άλλων) με τις ασκήσεις ΔΤ1 και ΔΣ1, προκύπτει εύκολα ότι η πολυπλοκότητα του αλγορίθμου είναι γραμμική O(n).

ΔΣ 3:

Στο κυρίως σώμα του αλγορίθμου Δυαδική_Αναζήτηση (παράγραφος 4.3 τετραδίου μαθητή) υπάρχει μία επαναληπτική δομη. Ανάλογα με το αποτέλεσμα της σύγκρισης "ΚΑΤ[mid]΄(΄ΟΝΟΜΑ", θα εκτελεσθεί ένας από τους τρεις κλάδους που έχουν σταθερό κόστος. Επομένως το ερώτημα είναι πόσες φορές θα επαναληφθεί η εντολή "Αν" μέσα στο βρόχο. Αυτό μπορεί να προκύψει παρατηρώντας ότι κάθε φορά το σχετικό διάστημα (high-low) συνεχώς υποδιπλασιάζεται. Ο μέγιστος αριθμός υποδιπλασιασμών είναι log(high-low). Έτσι προκύπτει ότι η πολυπλοκότητα του αλγορίθμου είναι O(logn). Συνεπώς σε ένα πίνακα 10 θέσεων, η δυαδική αναζήτηση τερματίζει με τέσσερις προσπάθειες στη χειρότερη περίπτωση.

ΔΣ 4:

Ο αλγόριθμος που δίνεται στην εκφώνηση είναι ο κλασικός αλγόριθμος ταξινόμησης με ευθεία ανταλλαγή. Όπως είναι γνωστό, η πολυπλοκότητα του αλγορίθμου είναι τετραγωνική $O(n^2)$. Άλλωστε, αυτό αποδεικνύεται και από το γεγονός της ύπαρξης των δύο εμφωλευμένων βρόχων. Ο αλγόριθμος του Παραδείγματος

2 είναι επίσης τετραγωνικός. Αν και τα όρια των επαναλήψεων των βρόχων είναι διαφορετικά κατά μία μονάδα (n-1 αντί n), η διαφορά αυτή είναι ασήμαντη και δεν επηρεάζει την ποιοτική ομοιότητα των δύο αλγορίθμων.

ΔΣ 5:

Έστω ότι συχνά εκτελούνται αναζητήσεις του τύπου «Βρες τις ταινίες του έτους 19??». Στην περίπτωση αυτή αξίζει τον κόπο να έχουμε τον πίνακα ταξινομημένο ως προς τη χρονολογία παραγωγής. Αυτό θα συμβεί χρησιμοποιώντας κάποια από τις μεθόδους ταξινόμησης. Όταν, λοιπόν, ο πίνακας είναι ταξινομημένος ως προς τη χρονολογία παραγωγής, μπορεί να χρησιμοποιηθεί και πάλι η γραμμική μέθοδος αναζήτησης. Όμως στην περίπτωση αυτή, όταν φθάσουμε σε μία ταινία με χρονολογία παραγωγής μεγαλύτερη από τη ζητούμενη, τότε η αναζήτηση σταματά και δεν συνεχίζει μέχρι το τέλος του πίνακα. Η μετατροπή του αλγορίθμου γραμμικής αναζήτησης για να την περίπτωση των ταξινομημένων δεδομένων είναι εύκολη υπόθεση.

5.9 Συμπληρωματικά στοιχεία

Ένας προσεγγιστικός αλγόριθμος: ομαδοποίηση σε κιβώτια.

Το πρόβλημα της ομαδοποίησης σε κιβώτια ορίζεται ως εξής. Δίνονται n αντικείμενα και κιβώτια μέγιστης χωρητικότητας C κιλών. Το κάθε αντικείμενο έχει βάρος l_i κιλών, όπου $l_i < C$ για $1 \leq i \leq n$. Ζητείται η τοποθέτηση των n αντικειμένων αυτών στον ελάχιστο αριθμό κιβωτίων, δεδομένου ότι κάθε αντικείμενο μπορεί να τοποθετηθεί σε ένα και μόνο ένα κιβώτιο.

Έστω ότι n = 6 και c = 100, ενώ τα βάρη των αντικειμένων είναι 50, 60, 30, 70, 50 και 40 αντίστοιχα. Για να βρεθεί ο ελάχιστος αριθμός κιβωτίων που μπορεί να χωρέσει τα αντικείμενα απαιτείται να γίνουν δοκιμές για όλους τους δυνατούς τρόπους. Το σύνολο των δυνατών τρόπων που πρέπει να εξετασθούν είναι n! = 6! = 120, αριθμός που προκύπτει θεωρώντας τις αντίστοιχες διατάξεις των n αντικειμένων. Αν το n είναι μικρός ακέραιος, τότε ίσως είναι εφικτό να γίνουν όλες οι σχετικές δοκιμές ώστε να βρεθεί η καλύτερη λύση, η λεγόμενη *βέλτιστη* (optimal). Ωστόσο, είναι ευνόητο ότι όταν το n είναι ιδιαίτερα μεγάλο, π.χ. 1000, τότε το σύνολο των δυνατών τρόπων που προκύπτει είναι ένας πραγματικά τεράστιος αριθμός (δηλαδή 1000!), οπότε η δοκιμή όλων των σχετικών διατάξεων απαιτεί τε-

ράστιους υπολογιστικούς πόρους. Σε μία τέτοια περίπτωση συχνά ζητείται να βρεθεί όχι η βέλτιστη λύση αλλά μία *υποβέλτιστη* (suboptimal) που εγγυημένα θα προσεγγίζει τη βέλτιστη ικανοποιητικά, η οποία να προκύπτει από έναν αλγόριθμο που δεν είναι ιδιαίτερα χρονοβόρος.

Θεωρώντας και πάλι τις τιμές του συγκεκριμένου παραδείγματος, μπορεί εύκολα να προκύψει ότι ο ελάχιστος αριθμός κιβωτίων είναι 3. Για παράδειγμα, το 1ο με το 5ο, το 2ο με το 6ο και το 3ο με το 4ο αντικείμενο μπορούν να συνδυασθούν σε τρία ζεύγη, όπου το βάρος του κάθε ζεύγους είναι ίσο με 100 κιλά, όσο δηλαδή και η χωρητικότητα των κιβωτίων.

Γενικά, στο πρόβλημα αυτό έχουν δοθεί ως απάντηση πολλοί προσεγγιστικοί αλγόριθμοι. Στη συνέχεια θα εξετασθεί ένας από αυτούς, που στην αγγλική βιβλιογραφία συναντάται με τη μνημονική ονομασία first fit. Η ονομασία αυτή προκύπτει από τη φιλοσοφία, στην οποία στηρίζεται ο αλγόριθμος. Με απλά λόγια, λοιπόν, ο αλγόριθμος θεωρεί ότι τα κιβώτια και τα αντικείμενα είναι αριθμημένα. Έτσι λαμβάνοντας κάθε αντικείμενο με τη σειρά, προσπαθεί να το βάλει στο κιβώτιο με το μικρότερο αύξοντα αριθμό που έχει διαθέσιμο χώρο. Ο επόμενος αλγόριθμος περιγράφει καλύτερα αυτή τη μέθοδο της λύσης.

Αλγόριθμος: Τοποθέτηση αντικειμένου στην πρώτη διαθέσιμη θέση όπου ταιριάζει .	
Είσοδος:	Αριθμός αντικειμένων n. Χωρητικότητα κιβωτίου C. Βάρος κάθε αντικειμένου l_i, όπου $l_i < C$ για $1 \le i \le n$.
Έξοδος:	Πλήθος απαραίτητων κιβωτίων X.
Βήμα 1	Αρίθμησε τα αντικείμενα 1,2,...,n.
Βήμα 2	Αρίθμησε τα κιβώτια 1,2,...,m.
Βήμα 3	Θέσε i=1. Θέσε j=1. Θέσε X=1.
Βήμα 4	Αν το i-οστό αντικείμενο χωρά στο j-οστό κιβώτιο, τότε τοποθέτησέ το εκεί, ανανέωσε τη διαθέσιμη χωρητικότητα του κιβωτίου και πήγαινε στο Βήμα 6, αλλιώς θέσε j=j+1.
Βήμα 5	Αν j>X, τότε θέσε X=j και πήγαινε στο Βήμα 4.
Βήμα 6	Θέσε i=i+1.
Βήμα 7	Αν $i \le n$ πήγαινε στο Βήμα 4 αλλιώς πήγαινε στο Βήμα 8.
Βήμα 8	Τύπωσε τον X.

Αν ο αναγνώστης υλοποιήσει τον προσεγγιστικό υποβέλτιστο αυτόν αλγόριθμο με κάποια γλώσσα προγραμματισμού και δοκιμάσει το πρόγραμμα με τα

προηγούμενα δεδομένα, τότε θα διαπιστώσει ότι απαιτούνται 4 κιβώτια που θα περιέχουν τα αντικείμενα με αρίθμηση (1,3), (2,6), (4) και (5) αντίστοιχα. Έτσι για το συγκεκριμένο παράδειγμα απαιτείται ένα μόνο περισσότερο κιβώτιο σε σχέση με τον ελάχιστο δυνατό αριθμό κιβωτίων. Στο σημείο αυτό αναφέρεται ότι στη γενική περίπτωση ο αριθμός των κιβωτίων που θα προκύψει με τη μέθοδο first-fit είναι στη χειρότερη περίπτωση περίπου 70% μεγαλύτερος από τον ελάχιστο δυνατό αριθμό.

Κεφάλαιο 6

Εισαγωγή στον προγραμματισμό

6.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί ο μαθητές να αναγνωρίζουν τις βασικές τεχνικές προγραμματισμού και να χειρίζονται το προγραμματιστικό περιβάλλον του σχολείου τους ώστε να εκτελούν απλά έτοιμα προγράμματα.

6.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να ορίζουν τι είναι πρόγραμμα.
- Να κατατάσσουν τις γλώσσες προγραμματισμού.
- Να αναγνωρίζουν τα κυριότερα είδη προγραμματισμού.
- Να περιγράφουν τα βασικά χαρακτηριστικά των τεχνικών που χρησιμοποιούνται στον προγραμματισμό.
- Να διατυπώνουν τα πλεονεκτήματα του δομημένου προγραμματισμού.
- Να περιγράφουν τη διαδικασία εκτέλεσης ενός προγράμματος.
- Να αναφέρουν τα βασικά προγράμματα που περιέχει ένα προγραμματιστικό περιβάλλον.
- Να χρησιμοποιούν το προγραμματιστικό περιβάλλον του εργαστηρίου.

6.3 Οδηγίες – επισημάνσεις

Το θεωρητικό μέρος μπορεί να εμπλουτιστεί με παρουσιάσεις του κώδικα προγραμμάτων σε διάφορες γλώσσες προγραμματισμού, ειδικά από αυτές οι οποίες είναι χαρακτηριστικές για κάθε είδος προγραμματισμού.

Ιδιαίτερο βάρος πρέπει να δοθεί στο δομημένο προγραμματισμό και τα πλεονεκτήματα του. Να τονιστεί ότι αποτελεί τον πλέον φυσικό τρόπο μετατροπής του αλγόριθμου σε πρόγραμμα υπολογιστή.

Στο εργαστηριακό μέρος θα έλθουν σε επαφή με το προγραμματιστικό περιβάλλον που θα χρησιμοποιήσουν στα επόμενα μαθήματα. Πρέπει να γνωρίσουν όλα τα προγράμματα του προγραμματιστικού περιβάλλοντος και να εφαρμόσουν όλη τη διαδικασία εκτέλεσης ενός προγράμματος, ώστε να αποκτήσουν τις γνώσεις που θα τους είναι απαραίτητες για να συντάσσουν τα δικά τους προγράμματα στα επόμενα μαθήματα.

6.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων:

Ενα (1) δίωρο μάθημα.

Διδακτικοί στόχοι

Ως διδακτικοί στόχοι του μαθήματος θεωρούνται όλοι οι παραπάνω αναφερόμενοι στόχοι του κεφαλαίου.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Σχέδιο 1ου μαθήματος

Η έννοια του προγράμματος.

- **Ιστορική αναδρομή στις γλώσσες προγραμματισμού.**

  Γλώσσες μηχανής.

  Συμβολικές γλώσσες.

  Γλώσσες υψηλού επιπέδου.

  Πλεονεκτήματα γλωσσών υψηλού επιπέδου.

  Γλώσσες $4^{ης}$ γενιάς.

- **Φυσικές γλώσσες.**

  Διαφορές φυσικών και τεχνητών γλωσσών.

- **Τεχνικές σχεδίασης προγραμμάτων.**

  Ιεραρχική σχεδίαση.

  Διαγραμματικές τεχνικές απεικόνισης της ιεραρχικής σχεδίασης.

  Τμηματικός προγραμματισμός.

  Δομημένος προγραμματισμός.

  Η εντολή Go To και γιατί πρέπει να αποφεύγεται.

  Πλεονεκτήματα του δομημένου προγραμματισμού.

- **Αντικειμενοστραφής προγραμματισμός.**
- **Παράλληλος προγραμματισμός.**
- **Προγραμματιστικά περιβάλλοντα.**

  Διαδικασία μεταγλώττισης προγράμματος.

  Μεταγλωττιστές- διερμηνευτές.

  Συντάκτης.

  Φορτωτής.

Περιεχόμενα πρακτικής εφαρμογής

Δραστηριότητες από το τετράδιο του μαθητή

Πρέπει να δοθεί μία από τις δύο δραστηριότητες στην τάξη ΔΤ1, ΔΤ2. Επίσης να δοθεί η δραστηριότητα ΔΣ1 για το σπίτι.

Στο **εργαστήριο** να δοθεί η δραστηριότητα ΔΕ1 σύμφωνα με τις υποδείξεις που αναφέρθηκαν. Να τους δοθεί έτοιμο ένα πρόγραμμα, το οποίο να τυπώνει ένα μήνυμα, όπως για παράδειγμα 'ΚΑΛΩΣ ΟΡΙΣΑΤΕ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ'. Με τις υποδείξεις σας οι μαθητές αρχικά πρέπει να αλλάξουν το πρόγραμμα, ώστε να γνωρίσουν τη χρήση του συντάκτη. Στη συνέχεια πρέπει να το εκτελέσουν και να πάρουν ένα μήνυμα σφάλματος, ώστε να γνωρίσουν το μεταγλωττιστή και τα μηνύματα λαθών που παράγει και να διορθώσουν ξανά τα λάθη. Τέλος το σωστό πλέον πρόγραμμα πρέπει να το εκτελέσουν και να πάρουν το αποτέλεσμα, το μήνυμα δηλαδή στην οθόνη.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος.

1. Τα προγράμματα σε γλώσσες υψηλού επιπέδου θεωρητικά είναι ανεξάρτητά του υπολογιστή που αναπτύχθηκαν.
2. Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα.
3. Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση.
4. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

5. Χαρακτηριστικό του οπτικού προγραμματισμού είναι

   Α Επιτρέπει τη γραφική δημιουργία του περιβάλλοντος

   Β. Επιτρέπει την ανάπτυξη του προγράμματος σε τμήματα

   Γ. Είναι ταχύτερος στην εκτέλεση των προγραμμάτων

   Δ. Επιτρέπει την διαγραμματική παράσταση της σχεδίασης του προγράμματος

6. Η Basic είναι

   Α. Κατάλληλη για εφαρμογές τεχνητής νοημοσύνης

   Β. Υποστηρίζει την ανάπτυξη παράλληλου προγραμματισμού

   Γ. Μία γλώσσα γενικής χρήσης

   Δ. Κατάλληλη μόνο για εκπαίδευση.

Να συμπληρωθούν τα κενά

7. Ο δομημένος προγραμματισμός στηρίζεται στη χρήση τριών στοιχειωδών δομών: της ....., της ..... και της .....
8. Η μεταγλώττιση ενός προγράμματος γίνεται από τους ..... ή τους .....

Απαντήσεις τεστ αξιολόγησης επίδοσης

1. Σωστό
2. Λάθος
3. Λάθος
4. Σωστό
5. Α
6. Γ
7. Ακολουθίας, Επιλογής, Επανάληψης
8. Μεταγλωττιστές, Διερμηνευτές

6.5 Προβληματισμοί και θέματα προς συζήτηση

Να συζητήσετε για την εξέλιξη των φυσικών γλωσσών σε αντιδιαστολή με αυτή των τεχνητών. Μπορείτε να συζητήσετε την εξέλιξη της ελληνικής γλώσσας, τους λόγους και τους τρόπους που εξελίσσεται με την καθημερινή χρήση σε αντιδιαστολή με τις γλώσσες προγραμματισμού που δεν εξελίσσονται από μόνες τους αλλά βελτιώνονται και εμπλουτίζονται.

Να παρουσιάσετε κώδικα από διάφορες γλώσσες προγραμματισμού, δομημένες και μη, ώστε οι μαθητές να καταλάβουν τη διαφορά στη σαφήνεια, και την απλότητα μεταξύ τους.

6.6 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επίσης προτείνονται οι ακόλουθες διευθύνσεις Διαδικτύου που περιέχουν επιπλέον πληροφορίες:

- http://devcentral.iftech.com/learning/tutorials/

Εκπαιδευτικοί οδηγοί για τις γλώσσες προγραμματισμού Java, C++, Visual Basic.

- http://www.comlab.ox.ac.uk/archive/occam.html

  Πληροφορίες για μία από τις πλέον δημοφιλείς γλώσσες παράλληλου προγραμματισμού, την Occam.

6.7 Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

Όλες οι ερωτήσεις είναι θεωρητικές και ο απαντήσεις βρίσκονται στο κεφάλαιο 6 του βιβλίου.

1. Δες παράγραφο 6.1
2. Δες παράγραφο 6.2.1
3. Δες παράγραφο 6.2.3
4. Δες παράγραφο 6.2.3
5. Δες παράγραφο 6.2.3
6. Δες παράγραφο 6.3
7. Δες παράγραφο 6.3
8. Δες παράγραφο 6.4.1
9. Δες παράγραφο 6.4.3
10. Δες παράγραφο 6.4.3
11. Δες παράγραφο 6.5
12. Δες παράγραφο 6.7
13. Δες παράγραφο 6.7
14. Δες παράγραφο 6.7

6.8 Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

➤ Στην τάξη

ΔΤ1

```
ΑΡΧΗ
 ΟΣΟ συνθήκη 1 ΕΠΑΝΑΛΑΒΕ
    Εντολή2
    ΑΝ συνθήκη 3 ΤΟΤΕ
        Εντολή 4
        Η συνθήκη 1 γίνεται ΨΕΥΔΗΣ
    ΑΛΛΙΩΣ
        Εντολή 5
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΔΤ2

Η δραστηριότητα αυτή αποτελεί αφορμή για συζήτηση του τρόπου που γίνεται ουσιαστικά η ανάλυση ενός προγράμματος σε υποπρογράμματα και συγκεκριμένα πως κάθε υποπρόβλημα του αρχικού σύνθετου προβλήματος αντιμετωπίζεται ως ξεχωριστή ενότητα.

Το πρόγραμμα αναλύεται στα εξής τμήματα:

Εισαγωγή δεδομένων

- Καταχώριση δεδομένων
- Έλεγχος δεδομένων

Επεξεργασία δεδομένων

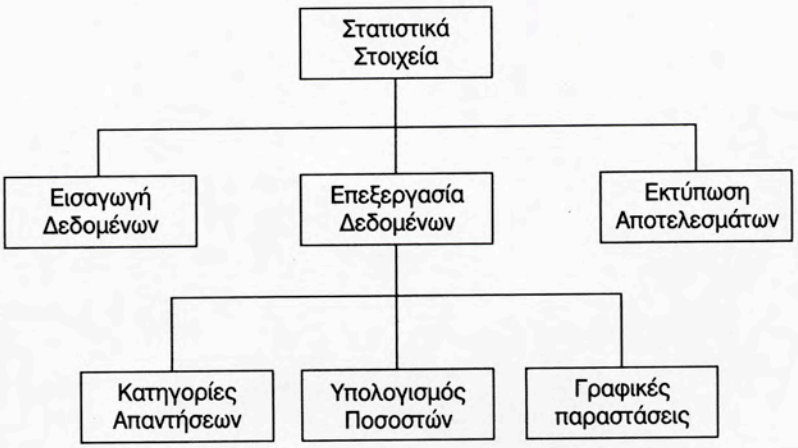
- Υπολογισμός κατηγοριών απαντήσεων
- Υπολογισμός ποσοστών
- Δημιουργία γραφικών παραστάσεων

Εκτύπωση αποτελεσμάτων

- Εκτύπωση πινάκων
- Εκτύπωση γραφικών παραστάσεων

Και αυτά ισχύουν για κάθε κράτος.

Διαγραμματικά μπορεί να αποτυπωθεί ως εξής



➤ ***Στο σπίτι***

ΔΣ1

Η καταγραφή των γλωσσών προτείνεται να γίνει σε έναν πίνακα όπως ο επόμενος

ΓΛΩΣΣΑ	ΤΟΜΕΑΣ ΕΦΑΡΜΟΓΩΝ	ΔΙΑΔΟΣΗ ΣΕ ΠΡΟΣΩΠΙΚΟΥΣ ΥΠΟΛ.	ΕΙΔΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
Pascal	Γενικής χρήσης	Μεγάλη	Δομημένος

ΔΣ2

```
ΑΡΧΗ
ΟΣΟ συνθήκη1 ΤΟΤΕ
    Εντολή1
    ΑΝ συνθηκη2 ΤΟΤΕ
        Εντολή2
        Εντολή3
    ΑΛΛΙΩΣ
        Εντολή4
    ΤΕΛΟΣ_ΑΝ
    Εντολή5
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Εντολή6
ΤΕΛΟΣ
```

Κεφάλαιο 7

Βασικές έννοιες προγραμματισμού

7.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί οι μαθητές να συντάσσουν και να εκτελούν σε δομημένη γλώσσα προγραμματισμού απλά προγράμματα τα οποία να δέχονται διαφόρων τύπων δεδομένα, στη συνέχεια να τα επεξεργάζονται και τελικά να παρέχουν τα αποτελέσματα.

7.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να διακρίνουν τις σταθερές από τις μεταβλητές ενός προγράμματος.
- να αναγνωρίζουν τους διάφορους τύπους μεταβλητών.
- να χρησιμοποιούν σωστά την εντολή εκχώρησης.
- να μετατρέπουν τις αριθμητικές πράξεις σε εντολές προγράμματος.
- να χρησιμοποιούν τις εντολές εισόδου και εξόδου.
- να διατυπώνουν τη δομή ενός προγράμματος.

7.3 Οδηγίες – επισημάνσεις

Γενικές οδηγίες για τον προγραμματισμό

Η έμφαση κατά τη διάρκεια της θεωρητικής παρουσίασης πρέπει να δίνεται στις γενικές έννοιες και αρχές του προγραμματισμού και όχι στον τρόπο που τις υλοποιεί το κάθε συγκεκριμένο προγραμματιστικό περιβάλλον, δηλαδή στις εντολές κάθε γλώσσας. Ο αντικειμενικός σκοπός είναι οι μαθητές να αποκτήσουν δεξιότητες στην ανάπτυξη των προγραμμάτων, και όχι να μάθουν απλά να χρησιμοποιούν τις εντολές και τα εκμεταλλεύονται τα ιδιαίτερα χαρακτηριστικά μίας γλώσσας προγραμματισμού.

Επίσης ιδιαίτερη έμφαση πρέπει να δίνεται στη σχεδίαση του αλγορίθμου. Οι μαθητές πρέπει να ενθαρρύνονται να επιλύουν το πρόβλημα στο χαρτί, *να σχεδιάζουν τα βήματα του αλγόριθμου και μετά να το μετατρέπουν σε κώδικα.* Αυτό πρέπει να το συνηθίσουν από την αρχή, ακόμη και για τα πιο απλά προβλήματα. Μόνο έτσι θα τους γίνει συνείδηση ώστε να σχεδιάζουν τη λύση στα πολύπλοκα προβλήματα.

Οι ιδιαιτερότητες κάθε γλώσσας προγραμματισμού καθώς και οι εντολές της, πρέπει να παρουσιάζονται πριν και κατά την εκτέλεση των εργαστηριακών ασκή-

σεων πάντα σε αναφορά, σύγκριση και αντιδιαστολή με τις γενικές αρχές που έχουν παρουσιαστεί στη θεωρία και αναφέρονται στο βιβλίο.

Οι σημαντικότερες ιδιαιτερότητες αναφέρονται συνοπτικά στα αντίστοιχα κεφάλαια. Συγκεκριμένα στα επόμενα κεφάλαια θα αναφέρονται ορισμένα χαρακτηριστικά των δημοφιλέστερων γλωσσών προσωπικών υπολογιστών, Pascal και Basic, τα οποία καλό είναι να τονίζονται στους μαθητές.

Ειδικές επισημάνσεις για την ύλη του κεφαλαίου 7

Πρέπει να δοθεί ιδιαίτερη έμφαση στην έννοια της μεταβλητής και της εκχώρησης τιμής στις μεταβλητές, καθώς και στη δομή που πρέπει να έχει κάθε πρόγραμμα.

Ένα σημείο που πρέπει να τονιστεί είναι η διαφορά της εκχώρησης τιμής με εντολή του προγράμματος και της εισαγωγής των τιμών από το πληκτρολόγιο κατά τη διάρκεια της εκτέλεσης του προγράμματος.

Οι μαθητές πρέπει να ενθαρρύνονται να σχεδιάζουν πάντα τη λύση στο χαρτί, πριν γράψουν το πρόγραμμα στον υπολογιστή.

Πρέπει να συνηθίσουν από την αρχή, που τα προγράμματα είναι απλά να τηρούν συγκεκριμένους κανόνες στη συγγραφή των προγραμμάτων, ώστε αυτά να είναι κατανοητά και ευκολοδιάβαστα.

Να γράφουν μία εντολή σε κάθε γραμμή, να χρησιμοποιούν εσοχές όπου αυτό χρειάζεται, να γράφουν σχόλια, να χρησιμοποιούν μεταβλητές που το όνομα τους να υποδηλώνει τη χρήση τους.

7.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων:

ένα (1) δίωρο μάθημα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Ως διδακτικοί στόχοι του μαθήματος θεωρούνται όλοι οι παραπάνω αναφερόμενοι στόχοι του κεφαλαίου.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Βασικά στοιχεία της γλώσσας

Αλφάβητο

τύποι δεδομένων

ονοματολογία

σταθερές

μεταβλητές

Έκφραση αριθμητικών πράξεων

Αριθμητικοί τελεστές

ενσωματωμένες συναρτήσεις

Εντολή εκχώρησης

Εντολές εισόδου, εντολές εξόδου

Δομή προγράμματος

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα του κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Κατ' ελάχιστον θα πρέπει να δοθούν στους μαθητές προς λύση η δραστηριότητα 1 ή 2 στην τάξη, η δραστηριότητα 1 στο εργαστήριο και μία δραστηριότητα για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ τόσο σε αυτό το κεφάλαιο όσο και στα επόμενα αναφέρονται στη ΓΛΩΣΣΑ, η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο. Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι, ώστε, να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ.
2. Οι δηλώσεις των σταθερών προηγούνται πάντοτε των δηλώσεων των μεταβλητών.
3. Τα σχόλια τοποθετούνται πάντα στην αρχή του προγράμματος.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

4. Ποια από τις παρακάτω εντολές αυξάνει τη μεταβλητή **Πλήθος** κατά μία μονάδα

   Α) Πλήθος<-Πλήθος+1

   Β) Πλήθος<- +1

   Γ) Πλήθος<- 1

   Δ) Πλήθος+1<-Πλήθος

5. Ποια η τιμή της μεταβλητής Α μετά την εκτέλεση της παρακάτω εντολής:

   A <- (5+4/2*2)*2-(3*2+5-3)^2+9/3-2

   Α) -53

   Β) -37

   Γ) -125

   Δ) –45

Να συμπληρωθούν τα κενά

6.

```
ΠΡΟΓΡΑΜΜΑ Τεστ
……..
    Π=3.14
ΜΕΤΑΒΛΗΤΕΣ
    ……..:Ε, Ακτίνα
ΑΡΧΗ
    ΔΙΑΒΑΣΕ Ακτίνα
Ε <- Π*Ακτίνα^2
ΓΡΑΨΕ 'Εμβαδό :', ……..
……..
```

Απαντήσεις τεστ αξιολόγησης επίδοσης

1. Λάθος
2. Σωστό
3. Λάθος
4. Α
5. Δ
6. ΣΤΑΘΕΡΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ, Ε, ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

7.5 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον οι ακόλουθες διευθύνσεις Διαδικτύου είναι ενδιαφέρουσες και μπορεί να φανούν χρήσιμες:

- http://pascal.miningco.com/

  Πλήρης οδηγός της Pascal. Περιλαμβάνει αρκετά προχωρημένα θέματα.

- http://www.angelfire.com/pa/rjolliff/

  Ένας ακόμη οδηγός προγραμματισμού σε Pascal.

- http://www.xs4all.nl/~dgb/delpascl.html#A_SITES

  Κατάλογος με όλες τις διευθύνσεις που σχετίζονται με Pascal.

7.6 Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

Όλες οι ερωτήσεις είναι θεωρητικές και οι αντίστοιχες απαντήσεις βρίσκονται στο κεφάλαιο 7 του βιβλίου. Συγκεκριμένα:

1. Δες παράγραφο 7.2
2. Δες παράγραφο 7.3
3. Δες παράγραφο 7.4
4. Δες παράγραφο 7.7
5. Δες παράγραφο 7.9
6. Δες παράγραφο 7.9
7. Δες παράγραφο 7.10

7.7 Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

➤ *Στην τάξη*

ΔΤ1.

1. T<-2*π*T_P(M/D)
2. E<-(m*v*u^2)/2
3. F<-T_P(F1^2+F2^2+2*F1*F2*συν(φ))
4. X<-(-β+T_P(β^2-4*α*γ))/(2*α)

ΔΤ2.

1. ΧΑΡΑΚΤΗΡΕΣ
2. ΑΚΕΡΑΙΑ
3. ΠΡΑΓΜΑΤΙΚΗ
4. ΧΑΡΑΚΤΗΡΕΣ
5. ΧΑΡΑΚΤΗΡΕΣ
6. ΧΑΡΑΚΤΗΡΕΣ ή ΛΟΓΙΚΕΣ (η μία τιμή Αληθής και η άλλη Ψευδής)

ΔΤ3.

```
ΠΡΟΓΡΑΜΜΑ Τρίγωνο
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ:Ε, Τ, Α, Β, Γ
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε πλευρές'
ΔΙΑΒΑΣΕ Α, Β, Γ
Τ<-(Α+Β+Γ)/2
Ε<-Τ_Ρ(Τ*(Τ-Α)*(Τ-Β)*(Τ-Γ))
ΓΡΑΨΕ Ε
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Τρίγωνο
```

Για να αποτελεί μία τριάδα αριθμών πλευρές τριγώνου πρέπει το άθροισμα ανά δύο να είναι μεγαλύτερο από την τρίτη.

Αν αυτό δεν εξεταστεί, τότε ένας από τους όρους Τ-Α, Τ-Β, Τ-Γ της ρίζας του υπολογισμού του εμβαδού μπορεί να είναι αρνητική, οπότε η ρίζα δεν υπολογίζεται και το πρόγραμμα σταματάει με λάθος κατά την εκτέλεσή του.

➤ *Στο εργαστήριο*

Βλέπε συνοδευτικό υλικό

➤ *Στο σπίτι*

Βλέπε συνοδευτικό υλικό

Κεφάλαιο 8

Επιλογή και επανάληψη

8.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί οι μαθητές να συντάσσουν και να εκτελούν σε δομημένη γλώσσα προγραμματισμού προγράμματα τα οποία να χρησιμοποιούν και τις τρεις βασικές δομές του δομημένου προγραμματισμού: την δομή της ακολουθίας, της επιλογής και της επανάληψης.

8.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να σχηματίζουν λογικές εκφράσεις, απλές και σύνθετες.
- Να διατυπώνουν τις μορφές των εντολών επιλογής (ελέγχου).
- Να διακρίνουν τις διαφορές των μορφών των εντολών επιλογής.
- Να επιλέγουν την καλύτερη μορφή των εντολών επιλογής.
- Να διατυπώνουν τις εντολές επανάληψης.
- Να διακρίνουν τις διαφορές των διαφορετικών εντολών επανάληψης.
- Να επιλέγουν την καλύτερη δομή επανάληψης και να χρησιμοποιούν την κατάλληλη εντολή.

8.3 Οδηγίες – επισημάνσεις

Η χρήση της σωστής εντολής που θα χρησιμοποιηθεί τόσο για την επιλογή όσο και την επανάληψη είναι ουσιαστικά θέμα σχεδιασμού του κατάλληλου αλγόριθμου. Επισημάνσεις και εναλλακτικές προσεγγίσεις για την επιλογή και την επανάληψη δόθηκαν στο κεφάλαιο 2.

Pascal

Η εντολή FOR της Pascal δεν δίνει τη δυνατότητα ορισμού του βήματος. Σε κάθε επανάληψη η μεταβλητή ελέγχου του βρόχου αυξάνει (ή μειώνεται) κατά 1. Άρα προβλήματα τα οποία απαιτούν αλλαγή του βήματος με τιμή διάφορη του 1, πρέπει να αντιμετωπιστούν είτε με την εντολή for χρησιμοποιώντας όμως κάποιο τέχνασμα (παράδειγμα 3 στο τετράδιο του μαθητή) είτε με κάποια από τις άλλες εντολές επανάληψης. Οι άλλες εντολές επανάληψης while, repeat καθώς και οι εντολές επιλογής if, case είναι παρόμοιες με τις αντίστοιχες εντολές της ΓΛΩΣΣΑΣ.

8.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα.

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- Να σχηματίζουν λογικές εκφράσεις, απλές και σύνθετες.
- Να διατυπώνουν τις μορφές των εντολών επιλογής (ελέγχου).
- Να διακρίνουν τις διαφορές των μορφών των εντολών επιλογής.
- Να επιλέγουν την καλύτερη μορφή των εντολών επιλογής.
- Να συντάσσουν προγράμματα τα οποία χρησιμοποιούν τις βασικές δομές της ακολουθίας και της επιλογής.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Λογικές εκφράσεις

Τελεστές συσχέτισης

Δημιουργία απλών λογικών εκφράσεων

Λογικοί τελεστές: ΟΧΙ, ΚΑΙ, Ή

Δημιουργία σύνθετων λογικών εκφράσεων

Εντολή ΑΝ

Η απλή μορφή της εντολής ΑΝ- ΤΟΤΕ

Η γενική μορφή της εντολής ΑΝ- ΤΟΤΕ- ΑΛΛΙΩΣ

Η εντολή ΑΝ- ΑΛΛΙΩΣ_ΑΝ

Διαφορές των μορφών της εντολής ΑΝ

Εμφωλευμένα ΑΝ

Εντολή ΕΠΙΛΕΞΕ

Σύνταξη της εντολής ΕΠΙΛΕΞΕ

Διαφορές της εντολής ΕΠΙΛΕΞΕ με την εντολή ΑΝ

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα της παραγράφου 8.1 από το βιβλίο του μαθητή και επιπλέον το παράδειγμα 1 από τθ τετράδιο μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθεί στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ1 ή ΔΤ2 στην τάξη. Επίσης να συζητηθεί η δραστηριότητα ΔΤ3 παράλληλα με το παράδειγμα 1. Στο εργαστήριο να δοθεί η δραστηριότητα ΔΕ1 και για το σπίτι η δραστηριότητα ΔΣ2.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Κάθε εντολή ΑΝ περιλαμβάνει υποχρεωτικά το τμήμα ΑΛΛΙΩΣ.
2. Κάθε τμήμα προγράμματος που χρησιμοποιεί την εντολή ΕΠΙΛΕΞΕ μπορεί να γραφεί και με εντολές ΑΝ.
3. Η χρήση εμφωλευμένων ΑΝ είναι καλή προγραμματιστική τακτική.
4. Αν το Α έχει την τιμή 10 και το Β την τιμή 20 τότε η έκφραση (Α >8 ΚΑΙ Β<20) Ή (Α>10 Ή Β=10) είναι αληθής.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

5. Τι θα εκτυπώσουν οι παρακάτω εντολές:

```
Α<-0
Β<-5
Γ<-10
ΑΝ Α>10 ΤΟΤΕ
    ΑΝ Β >20 ΤΟΤΕ
    ΑΝ Γ >10 ΤΟΤΕ
        ΓΡΑΨΕ Γ
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 2*Γ
    ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ Β
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
    ΑΝ Β <10 ΤΟΤΕ
        ΓΡΑΨΕ Α
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```

Α. 0
Β. 10
Γ. 5
Δ. 20

6. Να **συμπληρωθούν τα κενά** ώστε οι επόμενες εντολές να τυπώνουν πάντα τον μεγαλύτερο αριθμό από τους δύο που διαβάστηκαν.

```
ΔΙΑΒΑΣΕ Α,Β
ΑΝ Α<Β …
    ……
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ Α
```

7. Να **συμπληρωθούν τα κενά** ώστε οι επόμενες εντολές να τυπώνουν την τετραγωνική ρίζα.

```
ΔΙΑΒΑΣΕ Α
ΑΝ Α...0 ΤΟΤΕ
        Ρίζα <- Τ_Ρ(Α)
       ΓΡΑΨΕ Ρίζα
.....
           ΓΡΑΨΕ 'Δεν υπάρχει ρίζα'
ΤΕΛΟΣ_ΑΝ
```

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : Λάθος

2 : Σωστό

3 : Λάθος

4 : Λάθος

5: Α

6: ΤΟΤΕ, Α <- Β

7: >=, ΑΛΛΙΩΣ

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

- Να διατυπώνουν τις εντολές επανάληψης.
- Να διακρίνουν τις διαφορές των εντολών επανάληψης.
- Να επιλέγουν την καλύτερη δομή επανάληψης και να χρησιμοποιούν την κατάλληλη εντολή.
- Να συντάσσουν προγράμματα τα οποία χρησιμοποιούν και τις τρεις βασικές δομές της ακολουθίας, της επιλογής και της επανάληψης.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Δομή επανάληψης

Η έννοια του βρόχου

Διαφορετικές δομές επανάληψης

Εντολή ΟΣΟ- ΕΠΑΝΑΛΑΒΕ

Σύνταξη της εντολής

Χρήση της εντολής

Εντολή ΜΕΧΡΙΣ_ΟΤΟΥ

Σύνταξη της εντολής

Χρήση της εντολής

Διαφορές με την εντολή ΟΣΟ

Εντολή ΓΙΑ

Σύνταξη της εντολής

Χρήση της εντολής

Διαφορές με τις άλλες εντολές επανάληψης

Εμφωλευμένες εντολές ΓΙΑ

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν όλα τα παραδείγματα της παραγράφου 8.2 από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθούν στους μαθητές προς λύση η δραστηριότητα ΔΤ4 ή ΔΤ5 στην τάξη. Στο εργαστήριο να δοθεί η δραστηριότητα ΔΕ4, ειδικά αν υπάρχει σύνδεση με Internet, ώστε να παρακινηθούν οι μαθητές να ερευνούν για την απόκτηση πραγματικών δεδομένων για τα προγράμματα τους. Επίσης, αν υπάρχει διαθέσιμος χρόνος η άσκηση ΔΕ3.

Για το σπίτι να δοθεί η ΔΣ5 ή η ΔΣ1.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ αναφέρονται στη ΓΛΩΣΣΑ η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο.

Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι ώστε να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1. Οι εντολές που βρίσκονται σε μία επανάληψη ΓΙΑ εκτελούνται τουλάχιστο μία φορά.
2. Κάθε επανάληψη μπορεί να γραφεί με την εντολή ΟΣΟ- ΕΠΑΝΑΛΑΒΕ.
3. Σε περίπτωση εμφωλευμένων βρόχων, ο εσωτερικός πρέπει να περικλείεται ολόκληρος στον εξωτερικό.
4. Η τιμή του βήματος αναφέρεται υποχρεωτικά σε κάθε εντολή ΓΙΑ.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

5. Πόσες φορές θα εκτελεστεί η εντολή ΓΡΑΨΕ Α

```
Α<-10
ΟΣΟ Α<>0 ΕΠΑΝΑΛΑΒΕ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
        Α<-Α-1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ Α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Α. 10
Β. 0
Γ. 2
Δ. Άπειρες

6. Να **συμπληρωθούν τα κενά** ώστε οι επόμενες εντολές να τυπώνουν το άθροισμα των τετραγώνων των περιττών αριθμών που είναι μικρότεροι από 10.

```
Άθροισμα<-...
ΓΙΑ... ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ ΒΗΜΑ ...
    Άθροισμα<- ... + Ι^2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Άθροισμα
```

7. Να συμπληρωθούν τα κενά ώστε οι επόμενες εντολές να τυπώνουν το άθροισμα των αριθμών από 100 έως 200

```
Κ<-...
Σ<-...
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Σ<-Σ+Κ
    Κ<-Κ+1
ΜΕΧΡΙΣ_ΟΤΟΥ .....
ΓΡΑΨΕ Σ
```

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : Λάθος

2 : Σωστό

3 : Σωστό

4 : Λάθος

5 : Γ

6 : 0, Ι, 2, Άθροισμα

7 : 100, 0, Κ>200

8.5 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον οι ακόλουθες διευθύνσεις Διαδικτύου είναι ενδιαφέρουσες και μπορεί να φανούν χρήσιμες:

- http://pascal.miningco.com/

  Πλήρης οδηγός της Pascal. Περιλαμβάνει αρκετά προχωρημένα θέματα .

- http://www.netalive.org/ckb/

Πολλά παραδείγματα και έτοιμοι αλγόριθμοι σε Pascal και Delphi, που επιλύουν πολλά σύνθετα προγραμματιστικά προβλήματα.

- http://www.angelfire.com/pa/rjolliff/

  Ένας ακόμη οδηγός προγραμματισμού σε Pascal.

- http://www.xs4all.nl/~dgb/delpascl.html#A_SITES

  Κατάλογος με όλες τις διευθύνσεις που σχετίζονται με Pascal.

8.6 Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

Όλες οι ερωτήσεις είναι θεωρητικές και οι απαντήσεις βρίσκονται στο κεφάλαιο 8 του βιβλίου.

1. Δες παράγραφο 8.1
2. Δες παράγραφο 8.1
3. Δες παράγραφο 8.1.1
4. Δες παράγραφο 8.1.1
5. Δες παράγραφο 8.1.1
6. Δες παράγραφο 8.1.2
7. Δες παράγραφο 8.2

8. Δες παράγραφο 8.2.2
9. Δες παράγραφο 8.2.3
10. Δες παράγραφο 8.2.3

8.7 Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

➤ Στην τάξη

ΔΤ1.

Α. ΨΕΥΔΗΣ

Β. ΑΛΗΘΗΣ

Γ. ΑΛΗΘΗΣ

Δ. ΑΛΗΘΗΣ

Ε. ΑΛΗΘΗΣ

ΔΤ2.

Α.

```
ΑΝ ΒΑΘΜΟΣ>ΜΟ ΤΟΤΕ
   ΓΡΑΨΕ 'Πολύ καλά'
ΑΛΛΙΩΣ_ΑΝ ΒΑΘΜΟΣ>=(ΜΟ-2) ΤΟΤΕ
   ΓΡΑΨΕ 'Καλά'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Μέτρια'
ΤΕΛΟΣ_ΑΝ
```

Β.

```
ΑΝ (ΤΜΗΜΑ = 'Γ1' ΚΑΙ ΒΑΘΜΟΣ >15) ΤΟΤΕ
    ΓΡΑΨΕ ΕΠΩΝΥΜΟ
ΤΕΛΟΣ_ΑΝ
```

Γ.

```
ΑΝ Απάντηση <>'ν' 'Η Απάντηση <> 'Ν' 'Η Απάντηση <> 'ο'
& Ή Απάντηση <> 'Ο' ΤΟΤΕ
    ΓΡΑΨΕ ' Λάθος απάντηση...'
ΤΕΛΟΣ_ΑΝ
```

Δ.

```
ΑΝ Χ <= 0 Ή ΗΜ(Χ)=0 ΤΟΤΕ
ΓΡΑΨΕ 'Λάθος δεδομένα...'
ΑΛΛΙΩΣ
    Υ<-(Χ^2+5*Χ)/(Τ_Ρ(Χ)* ΗΜ(Χ))
ΤΕΛΟΣ_ΑΝ
```

ΔΤ3.

Πρέπει να αλλάξει η εντολή ΑΝ που χρησιμοποιείται και να χρησιμοποιηθούν δύο διαφορετικές εντολές ΑΝ. Επίσης μπορούν να χρησιμοποιηθούν και δύο εντολές ΕΠΙΛΕΞΕ.

```
ΑΝ ΝΟ2>700 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ '
       ΓΡΑΨΕ 'Άζωτο πάνω από τα όρια'
ΑΛΛΙΩΣ_ΑΝ ΝΟ2>500 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ'
    ΓΡΑΨΕ 'Άζωτο πάνω από τα όρια'
ΑΛΛΙΩΣ_ΑΝ ΝΟ2>400 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ'
    ΓΡΑΨΕ 'Άζωτο στα όρια ασφαλείας'
ΤΕΛΟΣ_ΑΝ
ΑΝ Ο3 >500 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ'
    ΓΡΑΨΕ 'Όζον πάνω από τα όρια'
ΑΛΛΙΩΣ_ΑΝ Ο3 >300 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ'
    ΓΡΑΨΕ 'Όζον πάνω από τα όρια'
ΑΛΛΙΩΣ_ΑΝ Ο3 >250 ΤΟΤΕ
       ΓΡΑΨΕ 'ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ'
    ΓΡΑΨΕ 'Όζον στα όρια ασφαλείας'
ΤΕΛΟΣ_ΑΝ
```

ΔΤ4.

Υπολογίζει το άθροισμα των κύβων των αριθμών που είναι πολλαπλάσια του 5 και είναι μικρότεροι ή ίσοι με 100. Τυπώνει τους αριθμούς καθώς και την ύψωση τους στον κύβο και τέλος το άθροισμα τους.

```
Ι <- 0
Κ <- 0
ΟΣΟ Ι <=100 ΕΠΑΝΑΛΑΒΕ
    Α <- Ι^3
    Κ <- Κ+Α
    ΓΡΑΨΕ Ι, Α
    Ι <- Ι+5
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Κ
```

```
Ι <- 0
Κ <- 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        Α <- Ι^3
    Κ <- Κ+Α
    ΓΡΑΨΕ Ι, Α
    Ι <- Ι+5
ΜΕΧΡΙΣ_ΟΤΟΥ Ι > 100
ΓΡΑΨΕ Κ
```

Ο καλύτερος τρόπος είναι με χρήση της εντολής ΓΙΑ. Η εντολή ΓΙΑ πρέπει πάντα να χρησιμοποιείται όταν έχουμε προκαθορισμένο αριθμό επαναλήψεων.

ΔΤ5.

Με έντονη γραφή σημειώνονται οι διορθώσεις και οι πρόσθετες εντολές ενώ με διαγραφή οι εντολές που πρέπει να αφαιρεθούν.

```
ΔΙΑΒΑΣΕ Μισθός
Άθροισμα <- 0
ΕΠΑΝΑΛΑΒΕ ΟΣΟ Μισθός <> 0
    Άθροισμα <- 0
ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Άθροισμα <- Άθροισμα+Μισθός
    ΔΙΑΒΑΣΕ Μισθός
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Β.

```
Άθροισμα <- 0
ΔΙΑΒΑΣΕ Μισθός
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Άθροισμα <- 0
ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Άθροισμα <- Άθροισμα+Μισθός
    ΔΙΑΒΑΣΕ Μισθός
ΜΕΧΡΙΣ_ΟΤΟΥ Μισθός = 0
```

Γ.

```
Άθροισμα<-0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
Άθροισμα <- 0
ΔΙΑΒΑΣΕ Μισθός
ΑΝ Μισθός >  Μέγιστος ΤΟΤΕ
    Μέγιστος<- Μισθός
ΤΕΛΟΣ_ΑΝ
ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
    Ελάχιστος <- Μισθός
ΤΕΛΟΣ_ΑΝ
Άθροισμα <- Άθροισμα+Μισθός
ΤΕΛΟΣ_ΓΙΑ
```

➤ *Στο εργαστήριο*

Βλέπε συνοδευτικό υλικό

➤ *Στο σπίτι*

Βλέπε συνοδευτικό υλικό

8.8 Μερικές ακόμη συμβουλές

Σε ένα επαγγελματικό πρόγραμμα η μεγαλύτερη προσπάθεια αναλώνεται στην είσοδο δεδομένων. Τα προγράμματα αυτά έχουν εργονομία, επιτρέπουν τη γρήγορη καταχώριση δεδομένων, εκτελούν πολλαπλούς ελέγχους ορθότητας, επιτρέπουν τη διόρθωση, την κίνηση σε όλο το μήκος και πλάτος της οθόνης κ.α. Προφανώς αυτά τα χαρακτηριστικά ενός καλού προγράμματος απαιτούν μεγάλη πείρα και ενασχόληση με το αντικείμενο και δεν αποτελούν στόχο του μαθήματος. Προοδευτικά μερικά από αυτά τα στοιχεία θα κατακτηθούν από τους μαθητές.

Ελεγχος δεδομένων

Σε πρώτο στάδιο αυτό που πρέπει να μην αγνοηθεί, είναι ο έλεγχος των δεδομένων εισόδου.

Παράδειγμα. Ζητείται να εισαχθεί ένας ακέραιος από 1 μέχρι 10 και στη συνέχεια να ....

Ο μαθητής θα ορίσει έναν τύπο ακεραίου και κατά την εκτέλεση θα εισάγει έναν ακέραιο από 1 έως 10, αφού αυτό λέει η εκφώνηση. Προφανώς αυτό είναι ικανοποιητικό για να ελεγχθεί η συνέχεια. Δεν είναι όμως αρκετό, αν υποτεθεί ότι το πρόγραμμα θα εκτελεστεί από κάποιον άλλο.

Μια πρώτη αντιμετώπιση είναι πριν την εντολή εισόδου να υπάρχει εντολή εξόδου με μήνυμα του τύπου "Δώστε έναν ακέραιο από 1 μέχρι 10". Αλλά και αυτό δεν είναι αρκετό. Αν εισαχθεί ένας ακέραιος, μπορεί να ακολουθήσει έλεγχος αν είναι μεταξύ 1 και 10. Αν όμως εισαχθεί ένας πραγματικός; Στην περίπτωση αυτή θα σημειωθεί σφάλμα. Η λύση είναι η μεταβλητή εισόδου να είναι τύπου πραγματικού και στη συνέχεια να γίνεται μετατροπή σε ακέραια. Γενικεύοντας το πρόβλημα και θεωρώντας ότι ο χρήστης του προγράμματος μπορεί να πληκτρολογήσει οτιδήποτε, θα έπρεπε η μεταβλητή εισόδου να είναι αλφαριθμητική, να γίνουν οι σχετικοί έλεγχοι και μετά η μετατροπή, εφ' όσον είναι όλα καλά.

Γίνεται φανερό ότι και για την εισαγωγή μιας και μόνο τιμής, χρειάζονται αρκετές εντολές ελέγχου, πόσο μάλλον όταν οι εισαγόμενες τιμές είναι περισσότερες.

Σύσταση. Το πρόβλημα αρχικά επιλύεται θεωρώντας σωστή την είσοδο. Γίνεται συζήτηση σχετικά με τους απαιτούμενους ελέγχους και ζητείται από τους μα-

θητές να τους συμπεριλάβουν στο πρόγραμμά τους, είτε στο εργαστήριο, αν υπάρχει χρόνος, είτε στο σπίτι τους.

Είσοδος πολλών δεδομένων

Σε προβλήματα που απαιτείται η είσοδος αρκετών δεδομένων παρουσιάζεται το φαινόμενο ο μαθητής πρώτον να αναλώνει αρκετό χρόνο για τη συγγραφή των σχετικών εντολών και δεύτερο να αναλώνει πολύ χρόνο για την εισαγωγή των δεδομένων σε κάθε επανεκτέλεση του προγράμματος στο στάδιο τη δημιουργίας. Στις περιπτώσεις αυτές είναι προτιμότερο αρχικά να μην υπάρχουν εντολές εισόδου, αλλά οι τιμές να δίνονται στις μεταβλητές με εντολές εκχώρησης. Οταν το πρόγραμμα είναι σωστό, αντικαθιστώνται οι εντολές εκχώρησης με εντολές εισόδου. Αν το προγραμματιστικό περιβάλλον είναι η Basic, προτιμήστε τη χρήση των εντολών READ-DATA.

Το πρόβλημα αυτό γίνεται ιδιαίτερα οξύ στους πίνακες. Για παράδειγμα σε έναν πίνακα 4χ5 απαιτείται η εισαγωγή 20 τιμών σε κάθε εκτέλεση. Στην περίπτωση αυτή συνιστάται η χρήση μεταβλητών m και n για τα όρια του πίνακα. Στις μεταβλητές αυτές αρχικά εκχωρούνται μικρές τιμές κατά τις δοκιμές και αργότερα αντικαθίστανται με τις τιμές της εκφώνησης του προβλήματος.

Σε κάποια περίπτωση μπορεί να δεδομένα να παρέχονται έτοιμα στους μαθητές με τη μορφή αρχείου. Στην περίπτωση αυτή πρέπει να δοθεί στους μαθητές το τμήμα προγράμματος που εκτελεί την είσοδο από το συγκεκριμένο αρχείο. Για την υποβοήθηση του εκπαιδευτικού στο θέμα αυτό, μερικά χρήσιμα στοιχεία για τα αρχεία δεδομένων παρέχονται στο παράρτημα του βιβλίου αυτού.

Κεφάλαιο 9

Πίνακες

9.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί οι μαθητές να χρησιμοποιούν τους πίνακες στα προγράμματα τους.

9.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να επιλέγουν το είδος του πίνακα.
- Να ορίζουν τους πίνακες σε ένα πρόγραμμα.
- Να εισάγουν, να επεξεργάζονται και να τυπώνουν τα στοιχεία ενός πίνακα.
- Να αποφασίζουν αν είναι απαραίτητη η χρήση πίνακα.
- Να χρησιμοποιούν πολυδιάστατους πίνακες.
- Να αναφέρουν τις βασικές επεξεργασίες σε ένα πίνακα.
- Να αναζητούν και να ταξινομούν τα στοιχεία ενός πίνακα.

9.3 Οδηγίες – επισημάνσεις

Οι νέοι προγραμματιστές έχουν την τάση να χρησιμοποιούν πίνακες χωρίς πρώτα να εξετάζουν αν πραγματικά ο πίνακας είναι απαραίτητος στο πρόγραμμά τους.

Πρέπει να δοθεί βάρος στον έλεγχο της αναγκαιότητας χρήσης των πινάκων. Οι πίνακες θεωρούμε ότι είναι στατικές δομές και άρα πρέπει να ορίζονται στην αρχή κάθε προγράμματος. Αν και μερικές γλώσσες προγραμματισμού δίνουν τη δυνατότητα χρήσης δυναμικών πινάκων, καλό είναι οι μαθητές σε αυτό το στάδιο να τους θεωρούν στατικούς.

Οι επεξεργασίες πινάκων έχουν αναφερθεί συνοπτικά στα κεφάλαια 3 και 4.

9.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- Να επιλέγουν το είδος του πίνακα.
- Να ορίζουν τους πίνακες σε ένα πρόγραμμα.
- Να εισάγουν, να επεξεργάζονται και να τυπώνουν τα στοιχεία ενός μονοδιάστατου πίνακα.
- Να αποφασίζουν αν είναι απαραίτητη η χρήση πίνακα.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Μεταβλητή με δείκτη

Ορισμός του πίνακα

Στοιχεία του πίνακα

Μονοδιάστατοι πίνακες

Ορισμός πινάκων σε ένα πρόγραμμα

Ανάγνωση, επεξεργασία και εκτύπωση στοιχείων πίνακα

Πότε πρέπει να χρησιμοποιούνται πίνακες

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα της παραγράφου 9.1 από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθούν στους μαθητές προς λύση μία από τις δραστηριότητες ΔΤ1, ΔΤ2 στην τάξη, η δραστηριότητα ΔΕ1 στο εργαστήριο και η δραστηριότητα ΔΣ1 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ αναφέρονται στη ΓΛΩΣΣΑ η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο.

Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι ώστε να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1) Όλοι οι πίνακες δηλώνονται στο τμήμα δήλωσης μεταβλητών του προγράμματος.

2) Τα στοιχεία ενός πίνακα πρέπει να είναι του ιδίου τύπου.

3) Η χρήση πινάκων αυξάνει την απαιτούμενη μνήμη για την εκτέλεση του προγράμματος.

4) Ο δείκτης ενός μονοδιάστατου πίνακα πρέπει να είναι πάντα Ι.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

5) Η δήλωση ενός ακεραίου πίνακα 5 στοιχείων γίνεται με την εντολή

   Α) ΑΚΕΡΑΙΕΣ:Α[5]

   Β) ΑΚΕΡΑΙΕΣ:Α

   Γ) ΑΚΕΡΑΙΕΣ: Α[1], Α[2], Α[3], Α[4],Α[5]

   Δ) Α[5]:ΠΙΝΑΚΑΣ ΑΚΕΡΑΙΩΝ

6) Ποιες από τις παρακάτω εντολές υπολογίζουν το άθροισμα των στοιχείων ενός πίνακα Α με 10 στοιχεία

Α)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Σ<-Σ+Α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Β)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Σ<-Σ+Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Σ<-Σ+Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Σ<-Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Να συμπληρωθούν τα κενά

7) Οι πίνακες που χρησιμοποιούν ένα μόνο δείκτη για την αναφορά των στοιχείων τους, ονομάζονται ..... πίνακες.

8) Οι επόμενες εντολές δίνουν στα στοιχεία ενός πίνακα την τιμή 0

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 30
    Α[..]<-0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Απαντήσεις τεστ αξιολόγησης επίδοσης

1. Σωστό
2. Σωστό
3. Σωστό
4. Λάθος

5. Α
6. Γ
7. Μονοδιάστατοι
8. Ι

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

- Να χρησιμοποιούν πολυδιάστατους πίνακες.
- Να αναφέρουν τις βασικές επεξεργασίες σε ένα πίνακα.
- Να αναζητούν και να ταξινομούν τα στοιχεία ενός πίνακα.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Ορισμός και χρήση πινάκων δύο διαστάσεων.

Πίνακες με περισσότερες από δύο διαστάσεις.

Τυπικές επεξεργασίες πινάκων.

Εύρεση ελαχίστου και μεγίστου ενός πίνακα.

Υπολογισμός αθροίσματος στοιχείων πίνακα.

Ταξινόμηση και αναζήτηση στοιχείων πίνακα.

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα της παραγράφου 9.3 από το βιβλίο του μαθητή. Αν υπάρχει χρόνος να αναφερθεί και το παράδειγμα 3 από το τετράδιο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθούν στους μαθητές προς λύση οι δραστηριότητες ΔΤ3 και ΔΤ4 στην τάξη, η δραστηριότητα ΔΕ2 ή ΔΕ4 στο εργαστήριο και τουλάχιστον η δραστηριότητα ΔΣ5 για το σπίτι.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ αναφέρονται στη ΓΛΩΣΣΑ η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο.

Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι ώστε να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1. Τα ονόματα και τα εισιτήρια 10 θεάτρων μπορούν να αποθηκευτούν σε ένα δισδιάστατο πίνακα.
2. Οι δισδιάστατοι πίνακες είναι οι πίνακες με τη μεγαλύτερη διάσταση που χειρίζονται οι γλώσσες προγραμματισμού.
3. Η ταξινόμηση των στοιχείων ενός πίνακα πρέπει να γίνεται πάντα πριν από την αναζήτηση.
4. Η δυαδική αναζήτηση είναι πάντα ο καλύτερος τρόπος αναζήτησης.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

5. Ποιες εντολές τυπώνουν τα στοιχεία ενός δισδιάστατου πίνακα 5Χ5

Α)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Β)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ Α[Ι, Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ)

```
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ Α[J]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
        ΓΡΑΨΕ Α[Ι,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Να συμπληρωθούν τα κενά

6. Ένας δισδιάστατος πίνακας με ίσες τις δύο διαστάσεις του λέγεται .....

7. Οι παρακάτω εντολές υπολογίζουν και τυπώνουν το άθροισμα κάθε γραμμής ενός δισδιάστατου πίνακα Α5Χ5

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    Σ<-….
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ …
    Σ<-…+Α[…,…]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ Σ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Απαντήσεις τεστ αξιολόγησης επίδοσης

1. Λάθος
2. Λάθος
3. Λάθος
4. Λάθος
5. Δ
6. ΤΕΤΡΑΓΩΝΙΚΟΣ
7. 0, 5,Σ, I,J

9.5 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον οι ακόλουθες διευθύνσεις Διαδικτύου είναι ενδιαφέρουσες και μπορεί να φανούν χρήσιμες:

- http://pascal.miningco.com/

  Πλήρης οδηγός της Pascal. Περιλαμβάνει αρκετά προχωρημένα θέματα .

- http://www.netalive.org/ckb/

  Πολλά παραδείγματα και έτοιμοι αλγόριθμοι σε Pascal και Delphi, που επιλύουν πολλά σύνθετα προγραμματιστικά προβλήματα.

- http://www.angelfire.com/pa/rjolliff/

  Ένας ακόμη οδηγός προγραμματισμού σε Pascal.

- http://www.xs4all.nl/~dgb/delpascl.html#A_SITES

  Κατάλογος με όλες τις διευθύνσεις που σχετίζονται με Pascal.

9.6 Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

Όλες οι ερωτήσεις είναι θεωρητικές και οι απαντήσεις αναφέρονται στο κεφάλαιο 9 του βιβλίου.

1. Δες παράγραφο 9.1
2. Δες παράγραφο 9.1
3. Δες παράγραφο 9.1
4. Δες παράγραφο 9.1
5. Δες παράγραφο 9.1
6. Δες παράγραφο 9.1
7. Δες παράγραφο 9.1
8. Δες παράγραφο 9.3
9. Δες παράγραφο 9.1
10. Δες παράγραφο 9.2
11. Δες παράγραφο 9.4
12. Δες παράγραφο 9.4
13. Δες παράγραφο 9.4.

9.7 Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

➤ ***Στην τάξη***

ΔΤ1.

Α.

```
ΑΚΕΡΑΙΕΣ:Α[5]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    Α[Ι]<-Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Β.

```
ΑΚΕΡΑΙΕΣ:Α[10]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Α[Ι]<-Ι-1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ.

```
ΧΑΡΑΚΤΗΡΕΣ: Ονόματα[20]
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20
    ΔΙΑΒΑΣΕ Α[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ.

```
ΠΡΑΓΜΑΤΙΚΕΣ:Α[10]
Κ<-500
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    Α[Ι]<-Κ
    Κ<-Κ/2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΔΤ2.

```
Μέγιστο<-ΤΙΜΗ[1]
ΘέσηΜεγ<-1
Ελάχιστο<-ΤΙΜΗ[1]
ΘέσηΕλαχ<-1
ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν
    ΑΝ ΤΙΜΗ [Ι] > Μέγιστο ΤΟΤΕ
        Μέγιστο <-ΤΙΜΗ[Ι]
        ΘέσηΜεγ <-Ι
    ΤΕΛΟΣ_ΑΝ
    ΑΝ ΤΙΜΗ [Ι] < Ελάχιστο ΤΟΤΕ
        Ελάχιστο<-ΤΙΜΗ[Ι]
        ΘέσηΕλαχ<-Ι
```

```
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Μοντέλο[ΘέσηΜεγ], Μοντέλο[ΘέσηΜεγ]
```

ΔΤ3.

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 4
    ΑΝ Ι<>J ΤΟΤΕ
        Α[Ι,J]<-0
    ΑΛΛΙΩΣ
        Α[Ι,J]<-1
    ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΔΤ4.

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
    Βοηθητική<-Α[Ι,3]
    Α[Ι,3] <-Α[Ι,6]
    Α[Ι,6] <- Βοηθητική
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

➤ *Στο εργαστήριο*

Βλέπε συνοδευτικό υλικό

➤ *Στο σπίτι*

Βλέπε συνοδευτικό υλικό

Κεφάλαιο 10

Υποπρογράμματα

10.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί οι μαθητές να χρησιμοποιούν υποπρογράμματα για τη δημιουργία συνθέτων προγραμμάτων.

10.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να αναλύουν ένα σύνθετο πρόγραμμα σε απλά υποπρογράμματα.
- Να διακρίνουν τις συναρτήσεις από τις διαδικασίες.
- Να γράφουν τη δομή των υποπρογραμμάτων.
- Να επιλέγουν τη χρήση διαδικασίας ή συνάρτησης για την υλοποίηση ενός υποπρογράμματος.
- Να χρησιμοποιούν παραμέτρους για την επικοινωνία των υποπρογραμμάτων.
- Να καθορίζουν τις περιοχές εμβέλειας των παραμέτρων.
- Να διατυπώνουν την έννοια της αναδρομής.
- Να συντάσσουν αναδρομικά υποπρογράμματα.
- Να συγκρίνουν αναδρομικές και επαναληπτικές διαδικασίες.

10.3 Οδηγίες – επισημάνσεις

Οι μαθητές πρέπει να ενθαρρύνονται να χρησιμοποιούν υποπρογράμματα.

Ιδιαίτερη έμφαση πρέπει να δοθεί στην σωστή ανάλυση του προγράμματος σε υποπρογράμματα και στην επιλογή του κατάλληλου τύπου συνάρτησης ή διαδικασίας. Η ανάλυση αυτή πρέπει να γίνεται πάντα πριν αρχίσουν να γράφουν το πρόγραμμα. Πρέπει να αποφασίζουν για κάθε υποπρόγραμμα το είδος του, τη λειτουργία του καθώς και τον αριθμό των παραμέτρων που χρειάζεται.

Ο τρόπος με τον οποίο υλοποιούνται τα υποπρογράμματα σε κάθε γλώσσα προγραμματισμού έχει ιδιαιτερότητες.

Pascal

Η Pascal έχει δύο ειδών υποπρογράμματα, τις functions και τις procedures, οι οποίες έχουν τα χαρακτηριστικά των αντιστοίχων της ΓΛΩΣΣΑΣ.

Αυτό που πρέπει να τονιστεί στους μαθητές, με τη χρήση παραδειγμάτων, είναι ότι υπάρχουν δύο είδη παραμέτρων. Αυτές που απλώς περνούν τιμές στη διαδικασία κατά την κλήση της (in parameters ή value parameters) και αυτές που περνούν τιμές κατά την κλήση αλλά επιστρέφουν τιμές μετά την ολοκλήρωση της διαδικασίας (in- out parameters ή variable parameters). Το είδος της παραμέτρου δηλώνεται στην επικεφαλίδα της διαδικασίας στη λίστα των τυπικών παραμέτρων, για παράδειγμα procedure difference (a,b:real;var dif:real). Περισσότερα στοιχεία για τον τρόπο περάσματος παραμέτρων καθώς και παραδείγματα αναφέρονται στο εγχειρίδιο της γλώσσας, που περιλαμβάνεται στο τετράδιο μαθητή και το οποίο οι μαθητές πρέπει να παρακινηθούν να το μελετήσουν.

Αν και η Pascal επιτρέπει τη χρήση καθολικών μεταβλητών καλό είναι οι μαθητές να μάθουν να χρησιμοποιούν μόνο τοπικές μεταβλητές σε κάθε υποπρόγραμμα.

Basic

Στην QuickBASIC υπάρχουν επίσης υποπρογράμματα και συναρτήσεις.

Προσοχή: Η QuickBASIC αποκαλεί ως διαδικασίες τα υποπρογράμματα και τις συναρτήσεις.

Το πέρασμα των παραμέτρων μπορεί να γίνει με αναφορά και με τιμή. Όταν γίνεται πέρασμα με τιμή τότε η μεταβλητή ή η έκφραση στις πραγματικές παραμέτρους περικλείονται σε παρενθέσεις. Ας σημειωθεί ότι η λίστα μεταβλητών στην εντολή κλήσης του υποπρογράμματος αποκαλείται από την QuickBASIC λίστα ορισμάτων.

Η QuickBASIC διαθέτει και ένα άλλο είδος υποπρογράμματος, την υπορουτίνα. Οι υπορουτίνες είναι υποπρογράμματα ενταγμένα στην ίδια ενότητα προγράμματος, χρησιμοποιούν καθολικές μεταβλητές και η κλήση τους γίνεται με την εντολή GOSUB.

Η αναδρομή παρουσιάζει δυσκολίες για τους μαθητές. Ενώ μπορούν να καταλάβουν και να διατυπώσουν τον αναδρομικό ορισμό μίας διαδικασίας ή συνάρτη-

σης δυσκολεύονται να κατανοήσουν την υλοποίησή της από το προγραμματιστικό περιβάλλον. Πρέπει να παρουσιαστούν αναλυτικά όλα τα βήματα του υπολογισμού αναδρομικών προγραμμάτων και να εκτελέσουν και οι ίδιοι εικονικά στο χαρτί αναδρομικά προγράμματα.

10.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων.

δύο (2) δίωρα μάθημα.

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

- Να αναλύουν ένα σύνθετο πρόγραμμα σε απλά υποπρογράμματα.
- Να διακρίνουν τις συναρτήσεις από τις διαδικασίες.
- Να γράφουν τη δομή των υποπρογραμμάτων.
- Να επιλέγουν τη χρήση διαδικασίας ή συνάρτησης για την υλοποίηση ενός υποπρογράμματος.
- Να χρησιμοποιούν παραμέτρους για την επικοινωνία των υποπρογραμμάτων.
- Να καθορίζουν τις περιοχές εμβέλειας των παραμέτρων.

Χώρος υλοποίησης μαθήματος

Τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Τμηματικός προγραμματισμός

Ορισμός

Ανάλυση προβλήματος σε υποπροβλήματα

Διαγραμματική απεικόνιση της ανάλυσης

Υποπρογράμματα

Χαρακτηριστικά των υποπρογραμμάτων

Χρήση των υποπρογραμμάτων

Πλεονεκτήματα του τμηματικού προγραμματισμού

Παράμετροι

Συναρτήσεις- διαδικασίες

Διαφορές συναρτήσεων και διαδικασιών

Σύνταξη συναρτήσεων

Κλήση συναρτήσεων

Διαδικασίες

Σύνταξη διαδικασιών

Κλήση διαδικασιών

Τυπικές παράμετροι

Πραγματικές παράμετροι

Εμβέλεια παραμέτρων

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα των παραγράφων 10.1– 10.6 από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθούν στους μαθητές προς λύση η δραστηριότητα ΔΤ1 ή ΔΤ2 στην τάξη, η δραστηριότητα ΔΕ1 στο εργαστήριο και η δραστηριότητα ΔΣ1 για το σπίτι. Εάν υπάρξει χρόνος να δοθεί στο εργαστήριο και η δραστηριότητα ΔΕ3.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ αναφέρονται στη ΓΛΩΣΣΑ η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο.

Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι ώστε να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1. Η κλήση των διαδικασιών γίνεται με απλή αναφορά του ονόματος τους.
2. Κάθε υποπρόγραμμα πρέπει να έχει μόνο μία είσοδο και μία έξοδο.
3. Οι συναρτήσεις μπορούν να υπολογίζουν και να επιστρέφουν μόνο μία τιμή.

Επιλέξτε μεταξύ των προτεινόμενων μία σωστή απάντηση.

4. Ποια η επικεφαλίδα της συνάρτησης **Εμβαδό** που υπολογίζει το εμβαδό ενός τριγώνου (Ε=1/2*β*υ).

   Α) ΣΥΝΑΡΤΗΣΗ Εμβαδό(β,υ)

   Β) ΣΥΝΑΡΤΗΣΗ Εμβαδό

   Γ) ΣΥΝΑΡΤΗΣΗ Εμβαδό(β,υ):ΠΡΑΓΜΑΤΙΚΗ

   Δ) ΠΡΑΓΜΑΤΙΚΗ ΣΥΝΑΡΤΗΣΗ Εμβαδό

5. Τι θα τυπώσουν οι επόμενες εντολές

```
...
Α<-10
Β<-5
ΚΑΛΕΣΕ διαδ(Α,Β)
ΓΡΑΨΕ Α,Β
...
```

```
ΔΙΑΔΙΚΑΣΙΑ διαδ(Γ,Δ)
…..
Α<-0
Β<-0
ΓΡΑΨΕ Α,Β
…..
```

Α) 10 5

0 0

Β) 10 5

10 5

Γ) 0 0

0 0

Δ) 0 0

10 5

6. Οι μεταβλητές που ισχύουν μόνο στο υποπρόγραμμα που δηλώθηκαν λέγονται .....

7. Η λίστα των ..... παραμέτρων εμφανίζεται στη δήλωση των υποπρογραμμάτων ενώ η λίστα των ..... παραμέτρων στην κλήση τους.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1) Λάθος
2) Σωστό
3) Σωστό
4) Γ
5) Δ
6) Τοπικές
7) Τυπικών, Πραγματικών

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

- Να διατυπώνουν την έννοια της αναδρομής.
- Να συντάσσουν αναδρομικά υποπρογράμματα.
- Να συγκρίνουν αναδρομικές και επαναληπτικές διαδικασίες.

Χώρος υλοποίησης μαθήματος

τάξη και εργαστήριο.

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

Πίνακας, διαφάνειες, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή.

Περιεχόμενα θεωρητικής παρουσίασης

Αναδρομή

Η έννοια της αναδρομής.

Ορισμός αναδρομικών υποπρογραμμάτων.

Σύγκριση αναδρομικών και επαναληπτικών διαδικασιών.

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν τα παραδείγματα της παραγράφου 10.7 από το βιβλίο του μαθητή. Επίσης το παράδειγμα 2 από το τετράδιο μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθούν στους μαθητές προς λύση η δραστηριότητα ΔΤ3 στην τάξη και η δραστηριότητα ΔΕ4 στο εργαστήριο.

Τεστ αξιολόγησης επίδοσης

Οι ερωτήσεις του τεστ αναφέρονται στη ΓΛΩΣΣΑ η οποία παρουσιάζεται στη θεωρία και περιλαμβάνεται στο βιβλίο.

Οι ερωτήσεις όμως μπορούν να μετατραπούν εύκολα έτσι ώστε να αναφέρονται στην πραγματική γλώσσα προγραμματισμού η οποία χρησιμοποιείται στο εργαστήριο.

Συμπληρώστε με σωστό ή λάθος

1. Η αναδρομικές διαδικασίες είναι προτιμότερες από τις αντίστοιχες επαναληπτικές.
2. Αναδρομή ονομάζεται η δυνατότητα ενός υποπρογράμματος να καλεί τον εαυτό του.
3. Ο υπολογισμός του 4! με αναδρομική διαδικασία απαιτεί τον υπολογισμό του 3!, 2!, 1! και 0!

Να συμπληρωθούν τα κενά

4. Ο ορισμός κάθε αναδρομικού υποπρογράμματος έχει δύο τμήματα: την ..... και την .....

Απαντήσεις τεστ αξιολόγησης επίδοσης

1. Λάθος
2. Σωστό
3. Σωστό
4. Αναδρομική σχέση, τιμή βάσης

10.5 Προτεινόμενες πηγές πληροφόρησης

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον οι ακόλουθες διευθύνσεις Διαδικτύου είναι ενδιαφέρουσες και μπορεί να φανούν χρήσιμες:

- http://pascal.miningco.com/

  Πλήρης οδηγός της Pascal. Περιλαμβάνει αρκετά προχωρημένα θέματα .

- http://www.netalive.org/ckb/

  Πολλά παραδείγματα και έτοιμοι αλγόριθμοι σε Pascal και Delphi, που επιλύουν πολλά σύνθετα προγραμματιστικά προβλήματα.

- http://www.angelfire.com/pa/rjolliff/

  Ένας ακόμη οδηγός προγραμματισμού σε Pascal.

- http://www.xs4all.nl/~dgb/delpascl.html#A_SITES

  Κατάλογος με όλες τις διευθύνσεις που σχετίζονται με Pascal.

10.6 Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

Όλες οι ερωτήσεις είναι θεωρητικές και οι απαντήσεις περιέχονται στο βιβλίο του μαθητή στο κεφάλαιο 10. Συγκεκριμένα:

1. Δες παράγραφο 10.1
2. Δες παράγραφο 10.2
3. Δες παράγραφο 10.4
4. Δες παράγραφο 10.5
5. Δες παράγραφο 10.5.1
6. Δες παράγραφο 10.5.2
7. Δες παράγραφο 10.5.3
8. Δες παράγραφο 10.5.3
9. Δες παράγραφο 10.6
10. Δες παράγραφο 10.7
11. Δες παράγραφο 10.7

10.7 Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

➤ ***Στην τάξη***

ΔΤ1.

Α) Διαδικασία

Β) Συνάρτηση

Γ) Συνάρτηση

Δ) Διαδικασία

Ε) Συνάρτηση

Ζ) Διαδικασία

Η) Συνάρτηση

ΔΤ2.

Α)

```
ΔΙΑΔΙΚΑΣΙΑ Τετράγωνο (Χ2)
ΠΡΑΓΜΑΤΙΚΕΣ Χ,Χ2
ΑΡΧΗ
    ΔΙΑΒΑΣΕ Χ
    Χ2<-Χ^2
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Τετράγωνο
```

Β)

```
ΔΙΑΔΙΚΑΣΙΑ Μικρότερο (Α, Β, Μικρ)
ΠΡΑΓΜΑΤΙΚΕΣ Α, Β, Μικρ
ΑΡΧΗ
    ΑΝ Α>Β ΤΟΤΕ
        Μικρ ← Α
    ΑΛΛΙΩΣ
```

```
        Μικρ ← Β
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Μικρότερο
```

Γ)

```
ΔΙΑΔΙΚΑΣΙΑ Υπολογισμός_ΦΠΑ(Τιμή,ΦΠΑ)
ΠΡΑΓΜΑΤΙΚΕΣ:Τιμή, ΦΠΑ
ΑΡΧΗ
    ΦΠΑ <-Τιμή*0.18
    ΓΡΑΨΕ ΦΠΑ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Υπολογισμός_ΦΠΑ
```

Δ)

```
ΣΥΝΑΡΤΗΣΗ Άρτιος(Α):ΛΟΓΙΚΗ
ΑΚΕΡΑΙΕΣ:Α
ΑΡΧΗ
    ΑΝ Α mod 2 =0 ΤΟΤΕ
        Άρτιος <-ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ
        Άρτιος<-ΨΕΥΔΗΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ Άρτιος
```

ΔΤ3.

Αναδρομή: Παραγοντικό(Ν)<-Ν*Παραγοντικό(Ν-1)

Παραγοντικό(0)<-1

Παραγοντικό(4)	4*Παραγοντικό(3)
Παραγοντικό(3)	3*Παραγοντικό(2)
Παραγοντικό(2)	2*Παραγοντικό(1)
Παραγοντικό(1)	1*Παραγοντικό(0)
Παραγοντικό(1)	1

Παραγοντικό(2) 2*1

Παραγοντικό(3) 3*2

Παραγοντικό(4) 6*4

Επανάληψη:

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    Παραγοντικό<- Παραγοντικό*Ν
```

Ι	Παραγοντικό
1	1
2	2
3	2*3
4	6*4

➤ *Στο εργαστήριο*

Βλέπε συνοδευτικό υλικό

➤ *Στο σπίτι*

Βλέπε συνοδευτικό υλικό

Κεφάλαιο 11

Σύγχρονα προγραμματιστικά περιβάλλοντα

11.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου, είναι οι μαθητές να προσεγγίσουν σύγχρονα προγραμματιστικά περιβάλλοντα και να κατανοήσουν τις βασικές αρχές σύγχρονων τεχνικών προγραμματισμού όπως ο αντικειμενοστραφής και ο οδηγούμενος από τα γεγονότα προγραμματισμός.

11.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να μπορούν να ορίζουν τις έννοιες αντικείμενο, ιδιότητα, γεγονός και μέθοδος,
- να διατυπώνουν τη μεθοδολογία σχεδιασμού μιας εφαρμογής σε ένα σύγχρονο προγραμματιστικό περιβάλλον,
- να μπορούν να αναλύουν και να σχεδιάζουν μια εφαρμογή μέσα από αντικείμενα και γεγονότα,
- να αναγνωρίζουν τα γεγονότα που συμβαίνουν σε ένα πρόγραμμα και να καθοδηγούν τη ροή εκτέλεσής του,
- να κατασκευάζουν απλά προγράμματα σε σύγχρονα προγραμματιστικά περιβάλλοντα συνδυάζοντάς τα με τις γνώσεις τους στον τμηματικό και στον δομημένο προγραμματισμό.

11.3 Οδηγίες – επισημάνσεις

Επισημαίνουμε ότι η προσοχή του διδάσκοντα θα πρέπει να επικεντρωθεί και στα παρακάτω σημεία:

1. Να γίνει σαφές στους μαθητές ότι ο αντικειμενοστραφής και ο οδηγούμενος από τα γεγονότα προγραμματισμός είναι τεχνικές προγραμματισμού και όχι περιβάλλοντα προγραμματισμού.
2. Οι μαθητές θα πρέπει να συνδυάσουν τις καθημερινές έννοιες και προϋπάρχουσες γνώσεις για να αντιληφτούν το νέο τρόπο προγραμματισμού. Πρέπει να γίνει ξεκάθαρο στους μαθητές, μέσα από παραδείγματα, ότι η νέα προσέγγιση στο προγραμματισμό δεν είναι κάτι καινούργιο αλλά η προσομοίωση του φυσικού μας κόσμου στο χώρο των υπολογιστών.
3. Να δοθεί ιδιαίτερη έμφαση στο τρόπο συνδυασμού όλων των τεχνικών προγραμματισμού και αναλυθεί σε ποια σημεία του προγράμματος εφαρμόζεται ο τμηματικός και σε ποια ο δομημένος προγραμματισμός.

4. Αναφέρατε στους μαθητές διαφορές μεταξύ κλασικών εφαρμογών που ακολουθούν τις αρχές του δομημένου προγραμματισμού και σύγχρονων εφαρμογών που συνδυάζουν όλες τις τεχνικές προγραμματισμού που εφαρμόζονται στο κεφάλαιο.

5. Να επισημανθεί ότι η ροή εκτέλεσης μιας εφαρμογής σε ένα σύγχρονο περιβάλλον εργασίας κατευθύνεται από το χρήστη. Ο προγραμματιστής πρέπει να σχεδιάζει την εφαρμογή αντιμετωπίζοντας το πρόβλημα από την πλευρά του χρήστη. Με τον τρόπο αυτό θα αντιληφθεί ποια αντικείμενα πρέπει να χρησιμοποιήσει στο μέσο επικοινωνίας της εφαρμογής.

6. Οι μαθητές πρέπει να είναι σε θέση να αναλύουν το περιβάλλον εργασίας της εφαρμογής. Όταν αντιληφθούν τα αντικείμενα που περιέχει και το ρόλο τους μέσα σε αυτή, θα μπορούν πολύ εύκολα να προσδιορίσουν τα γεγονότα που πιθανώς να συμβούν κατά το χρόνο εκτέλεσης.

7. Να αναλύσετε στους μαθητές τον τρόπο πρόκλησης γεγονότων μέσα από κώδικα (trigger events) και να τους επιστήσετε την προσοχή για την αποφυγή γεγονότων που προκαλούν την επανάληψη του ίδιου γεγονότος (cascading events).

8. Πρέπει να καταστεί σαφές στους μαθητές ότι δεν συμπεριλαμβάνουμε εντολές κώδικα σε όλες τις διαδικασίες των γεγονότων που πρόκειται να συμβούν κατά το χρόνο εκτέλεσης, αλλά μόνο σε αυτές για τις οποίες θέλουμε να αντιδρά η εφαρμογή.

9. Να επεξηγηθούν στους μαθητές οι γενικές διαδικασίες και ο ρόλος τους μέσα σε μια εφαρμογή. Να αναφερθούν πλεονεκτήματά τους, όπως ο περιορισμός των εντολών κώδικα, η εύκολη αντίληψη των εντολών προγράμματος, η απλούστευση της εφαρμογής, η δυνατότητα επαναχρησιμοποίησής τους σε άλλο πρόγραμμα.

10. Να περιγράψετε και αν είναι δυνατό να επιδείξετε στους μαθητές σύγχρονα περιβάλλοντα εργασίας και εφαρμογές που συνδυάζουν τα πλεονεκτήματα ενός γραφικού περιβάλλοντος εργασίας.

11. Οι μαθητές πρέπει να αντιληφθούν ότι σε ένα σύγχρονο περιβάλλον δεν χρειάζεται να γράφουμε πρόγραμμα για ένα πρόβλημα που επιλύεται ορθά από μια υπάρχουσα εφαρμογή. Με τη σύνδεση διαφορετικών αυτόνομων προγραμμάτων δημιουργούμε δυναμικά περιβάλλοντα εργασίας.

12. Να δοθεί ιδιαίτερη προσοχή στην διδασκαλία των χαρακτηριστικών των αντικειμενοστραφών περιβαλλόντων ανάπτυξης εφαρμογών γιατί περιέχουν έννοιες δυσνόητες και πρωτόγνωρες για το μεγαλύτερο μέρος των μαθητών.

11.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

τρία (3) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να ορίζουν τις έννοιες αντικείμενο, ιδιότητα, γεγονός και μέθοδος,
- να αναλύουν και να σχεδιάζουν μια εφαρμογή μέσα από αντικείμενα και γεγονότα,
- να κατασκευάζουν απλά προγράμματα σε σύγχρονα προγραμματιστικά περιβάλλοντα συνδυάζοντάς τα με τις γνώσεις τους στον τμηματικό και στον δομημένο προγραμματισμό.

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή

Περιεχόμενα θεωρητικής παρουσίασης

- Αντικειμενοστραφής προγραμματισμός
- Χαρακτηριστικά αντικειμενοστραφών περιβαλλόντων ανάπτυξης εφαρμογών
- Οδηγούμενος από γεγονότα προγραμματισμός

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Πρέπει να αναφερθούν αντικείμενα από το φυσικό μας περιβάλλον και να περιγραφούν χαρακτηριστικά τους και ενέργειες που πραγματοποιούνται με αυτά. Να συσχετιστούν με αντικείμενα από το κόσμο των υπολογιστών και να αναλυθούν οι έννοιες ιδιότητα και μέθοδος. (παράγραφοι 11.1 και 11.2)

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη, με την προτεινόμενη σειρά :

- η δραστηριότητα ΔΤ1 και ΔΤ2.

Να υλοποιηθεί στο εργαστήριο:

- το παράδειγμα 1 του τετραδίου.
- η δραστηριότητα ΔΕ1.

Να δοθεί στους μαθητές για αντιμετώπιση στο σπίτι :

- η δραστηριότητα ΔΣ1.

Για την πραγματοποίηση των παραπάνω δραστηριοτήτων πρέπει ήδη να έχουν αναφερθεί αντίστοιχα παραδείγματα στην τάξη.

Για τις δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές να δίνονται τυχόν διευκρινήσεις που απαιτούνται από την πλευρά των μαθητών.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους ανατίθενται στο σπίτι αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Ο αντικειμενοστραφής προγραμματισμός είναι ένα σύγχρονο περιβάλλον εργασίας.
2. Οι μέθοδοι καθορίζουν τη συμπεριφορά ενός αντικειμένου.
3. Ένα γεγονός προκαλείται μόνο με μια ενέργεια του χρήστη.
4. Η κλάση προσδιορίζει τις αρχικές ιδιότητες και τη συμπεριφορά ενός αντικειμένου
5. Σε μια εφαρμογή είναι δυνατό να συνδυάσουμε το τμηματικό με το δομημένο προγραμματισμό.
6. Στον οδηγούμενο από τα γεγονότα προγραμματισμό μπορούμε να συμπεριλαμβάνουμε μόνο διαδικασίες γεγονότων.

Συμπληρώστε τα κενά με τη λέξη που λείπει.

7. Η αντικειμενοστραφή σχεδίαση στηρίζεται στα .....
8. Η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά τα χαρακτηριστικά και τη συμπεριφορά του καλείται .....
9. Οι μέθοδοι ενεργούν σε ένα αντικείμενο κατά το χρόνο ..... της εφαρμογής.
10. Στον οδηγούμενο από τα γεγονότα προγραμματισμό μπορούμε να συμπεριλάβουμε διαδικασίες γεγονότων και .....διαδικασίες.
11. Με τις τεχνικές του αντικειμενοστραφή αντικειμενοστραφή και του οδηγούμενου από τα γεγονότα προγραμματισμού επιτυγχάνεται ο ..... προγραμματισμός.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : λάθος

2 : σωστό

3 : λάθος

4 : σωστό

5 : σωστό

6 : λάθος

7 : δεδομένα

8 : ενθυλάκωση

9 : εκτέλεσης

10 : γενικές

11 : τμηματικός

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να διατυπώνουν τη μεθοδολογία σχεδιασμού μιας εφαρμογής σε ένα σύγχρονο προγραμματιστικό περιβάλλον,
- να αναλύουν και να σχεδιάζουν μια εφαρμογή μέσα από αντικείμενα και γεγονότα,
- να αναγνωρίζουν τα γεγονότα που συμβαίνουν στο περιβάλλον εργασίας ενός προγράμματος και να καθοδηγούν τη ροή εκτέλεσής του,
- να κατασκευάζουν απλά προγράμματα σε σύγχρονα προγραμματιστικά περιβάλλοντα συνδυάζοντάς τα με τις γνώσεις τους στον τμηματικό και στον δομημένο προγραμματισμό.

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή

Περιεχόμενα θεωρητικής παρουσίασης

- Υλοποίηση εφαρμογών σε σύγχρονο προγραμματιστικό περιβάλλον
- Δημιουργία διεπαφής χρήστη
- Διαδικασίες κώδικα
- Ροή εκτέλεσης εφαρμογών

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να παρουσιαστεί και να αναλυθεί, το παράδειγμα που αναφέρεται στην υλοποίηση εφαρμογών (παράγραφος 11.3.1).

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη, με την προτεινόμενη σειρά :

- η δραστηριότητα ΔΤ3 και ΔΤ4.

Να υλοποιηθεί στο εργαστήριο:

- η δραστηριότητα ΔΕ2 ή ένα από τα παραδείγματα 2 και 3 του τετραδίου εργασιών.

Για την υλοποίηση της δραστηριότητας ΔΕ2, αν δεν έχει επιδειχτεί το παράδειγμα της βολής στο κεφάλαιο 7, πρέπει να επισημανθούν τα βασικά σημεία που πρέπει να προσέξει ο μαθητής.

Να δοθούν στους μαθητές για αντιμετώπιση στο σπίτι :

- η δραστηριότητα ΔΣ2 ή ΔΣ3.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Σε ένα σύγχρονο προγραμματιστικό περιβάλλον δίνεται ιδιαίτερη έμφαση στο τρόπο επικοινωνίας της εφαρμογής με το χρήστη.
2. Ο σχεδιασμός μιας εφαρμογής σε ένα αντικειμενοστραφές περιβάλλον καθορίζεται από τα αντικείμενα.
3. Ο χρήστης παρακολουθεί την εκτέλεση μιας σύγχρονης εφαρμογής και αποκρίνεται σε αυτή όταν του ζητηθεί.
4. Η αρχική συμπεριφορά των αντικειμένων προσδιορίζεται κατά το χρόνο σχεδιασμού μέσω των ιδιοτήτων που υποστηρίζουν.
5. Η υλοποίηση μιας εφαρμογής σε ένα σύγχρονο προγραμματιστικό περιβάλλον ξεκινά με τη δημιουργία του κώδικα

Συμπληρώστε τα κενά με τη λέξη που λείπει.

6. Τα εργαλεία σχεδιασμού της διεπαφής του χρήστη με την εφαρμογή είναι τα .....
7. Σε ένα σύγχρονο περιβάλλον εργασίας κυρίαρχος είναι ο .....
8. Για την ολοκλήρωση της ανάπτυξης μιας εφαρμογής σε σύγχρονο προγραμματιστικό περιβάλλον πρέπει να γίνει η συγγραφή και η ..... του κώδικα.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : σωστό

2 : σωστό

3 : λάθος

4 : σωστό

5 : λάθος

6 : αντικείμενα

7 : χρήστης

8 : εκσφαλμάτωση

Σχέδιο 3ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να προσδιορίζουν τα στοιχεία που αποτελούν ένα γραφικό προγραμματιστικό περιβάλλον,
- να περιγράφουν τον τρόπο επικοινωνίας μιας εφαρμογής με το χρήστη και το περιβάλλον της,
- να σχεδιάζουν τον τρόπο με τον οποίο θα επικοινωνούν διαφορετικές εφαρμογές.

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή

Περιεχόμενα θεωρητικής παρουσίασης

- Στοιχεία γραφικού προγραμματιστικού περιβάλλοντος
- Μενού επιλογών
- Πλαίσια διαλόγου
- Επικοινωνία εφαρμογών

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Να δοθούν εναλλακτικοί τρόποι δημιουργίας του τρόπου διασύνδεσης με το χρήστη στο παράδειγμα που έχουμε υλοποιήσει στην παράγραφο 11.3.1. Προσπαθήστε να δώσετε ιδιαίτερη έμφαση στις ερωτήσεις 11, 12 και 13 του κεφαλαίου.

Δραστηριότητες από το τετράδιο του μαθητή

Να υλοποιηθεί στο εργαστήριο :

- η δραστηριότητα ΔΕ3.
- ή εναλλακτικά η δραστηριότητα ΔΣ2, που έχει ήδη σχεδιάσει ο μαθητής στο σπίτι.
- ή εναλλακτικά το παράδειγμα 4 του τετραδίου.
- ή εναλλακτικά τα παραδείγματα 5 και 6.

Να δοθεί στους μαθητές για αντιμετώπιση στο σπίτι :

- η δραστηριότητα ΔΣ4.

Για την αντιμετώπιση της δραστηριότητας ΔΣ4, είναι απαραίτητο να αναφερθεί σους μαθητές ο τρόπος που μπορούν να μορφοποιήσουν τη γραμματοσειρά ενός πλαισίου κειμένου στο προγραμματιστικό περιβάλλον του σχολείου. Για παράδειγμα στη Visual Basic οι μαθητές πρέπει να γνωρίζουν το χειρισμό του ειδικού αντικειμένου Font ή να χρησιμοποιούν τις ιδιότητες FontBold, FontItalic, FontSize, ForeColor και όποια άλλη κρίνετε απαραίτητη για την ολοκλήρωση της άσκησης.

Για τις δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές να δίνονται τυχόν διευκρινήσεις που απαιτούνται από την πλευρά των μαθητών.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους ανατίθενται στο σπίτι αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Ένα σύγχρονο προγραμματιστικό περιβάλλον παρέχει προς το προγραμματιστή εργαλεία για οπτική ανάπτυξη εφαρμογών.
2. Τα πλαίσια διαλόγου χρησιμοποιούνται για την εμφάνιση πληροφοριών προς το χρήστη και δεν επιτρέπουν την εισαγωγή δεδομένων.
3. Σε ένα σύγχρονο περιβάλλον εργασίας μπορούμε να επιτύχουμε την αμφίδρομη επικοινωνία διαφορετικών εφαρμογών.
4. Όταν επιτύχουμε τη διασύνδεση ενός αντικειμένου μπορούμε στη συνέχεια να το τροποποιήσουμε από την εφαρμογή μας.
5. Σε ένα σύγχρονο περιβάλλον χρησιμοποιούμε αποκλειστικά τα προκαθορισμένα πλαίσια διαλόγου που προσφέρει.

Συμπληρώστε τα κενά με τη λέξη που λείπει.

6. Τα αντικείμενα μενού υποστηρίζουν το γεγονός .....
7. Με την ..... τα δεδομένα φυλάσσονται μαζί στην εφαρμογή μας
8. Η συνεργασία ανεξάρτητων εφαρμογών αποτελεί ένα υψηλό σύστημα .....

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : σωστό

2 : λάθος

3 : σωστό

4 : λάθος

5 : λάθος

6 : κλικ

7 : ενσωμάτωση

8 : πολυδιεργασίας ή συνεπιτέλεσης

11.5 Προβληματισμοί και θέματα προς συζήτηση

Θέματα προβληματισμού και συζήτησης μπορούν να προκύψουν από όλες τις δραστηριότητες που προτείνονται στο συγκεκριμένο κεφάλαιο. Συγκεκριμένα μπορούν να προταθούν συζητήσεις σχετικά με :

- Την εξέλιξη των περιβαλλόντων προγραμματισμού.
- Τον τρόπο συνδυασμού των τεχνικών προγραμματισμού και τα αποτελέσματά του.
- Τις δυνατότητες που προσφέρουν οι σύγχρονες τεχνικές προγραμματισμού.
- Να αναπτυχθούν τα χαρακτηριστικά γραφικών περιβαλλόντων εργασίας.
- Να γίνει μια εκτίμηση των μελλοντικών εξελίξεων που θα προκύψουν για τα περιβάλλοντα προγραμματισμού και τα περιβάλλοντα εργασίας.

11.6 Προτεινόμενες πηγές πληροφόρησης

Βιβλιογραφία

Όλη η προτεινόμενη βιβλιογραφία και οι διευθύνσεις στο διαδίκτυο, όπως καταγράφονται στο βιβλίο του μαθητή.

Επιπλέον βιβλιογραφία

1. Timothy A. Budd : An Introduction to Object-Oriented Programming, Addison-Wesley, 1991.
2. Peter Coad, Edward Yourdon, Edward Yourdan : Object Oriented Design, Yourdon Press Computing Series, 1991.
3. Steve Teixeira : Delphi 4 Developer's Guide (Developer's Guide Series), Sams, Pacheco, 1998.

11.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

Σε ορισμένες από τις ερωτήσεις οι απαντήσεις είναι ελευθέρου τύπου και ταυτόχρονα σε μεγάλο βαθμό υποκειμενικές. Κατά συνέπεια δεν προτείνονται συγκεκριμένες απαντήσεις. Οι υπόλοιπες ερωτήσεις προέρχονται από το θεωρητικό τμήμα του κεφαλαίου και οι απαντήσεις μπορούν πολύ εύκολα να αναζητηθούν σε αυτό :

Ερώτηση	Απάντηση
2	11.3
3	11.2.2
5	11.1.1
7	11.1.4
8	11.2
10	11.1.2
12	11.5.1
13	11.6

11.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

Οι προτεινόμενες δραστηριότητες οι απαντήσεις είναι ελευθέρου τύπου και ταυτόχρονα σε μεγάλο βαθμό υποκειμενικές. Κατά συνέπεια δεν προτείνονται συγκεκριμένες απαντήσεις, παρά μόνο αναφέρονται πληροφοριακά στοιχεία όπου κρίνεται απαραίτητο.

ΔΤ3.

Στο συνοδευτικό υλικό του βιβλίου υπάρχουν ολοκληρωμένα και τα τρία παραδείγματα. Μπορείτε να συμβουλευτείτε τη δική μας προσέγγιση στη δημιουργία του τρόπου επικοινωνίας χρήστη-εφαρμογής.

➤ *Στο εργαστήριο*

Οι προτεινόμενες δραστηριότητες αναφέρονται στην υλοποίηση εφαρμογών με τη χρήση νέων τεχνικών και κατά συνέπεια η ανάπτυξή τους είναι σε μεγάλο βαθμό υποκειμενική. Στο συνοδευτικό υλικό του βιβλίο υπάρχουν λυμένα τα παραδείγματα μέσα από τη δική μας προσέγγιση και είναι δυνατό να τα συμβουλευτείτε ή να τα χρησιμοποιήσετε αυτούσια.

➤ *Στο σπίτι*

Οι προτεινόμενες δραστηριότητες αναφέρονται στην υλοποίηση εφαρμογών με τη χρήση νέων τεχνικών και κατά συνέπεια η ανάπτυξή τους είναι σε μεγάλο βαθμό υποκειμενική. Στο συνοδευτικό υλικό του βιβλίο υπάρχουν λυμένα τα παραδείγματα μέσα από τη δική μας προσέγγιση και είναι δυνατό να τα συμβουλευτείτε ή να τα χρησιμοποιήσετε αυτούσια.

Κεφάλαιο 12

Σχεδίαση διεπαφής χρήστη

12.1. Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου αυτού είναι οι μαθητές να αναπτύξουν δεξιότητες κυρίως σχεδιασμού και (κατά δεύτερο λόγο) υλοποίησης απλού, φιλικού, εργονομικού και εύχρηστου περιβάλλοντος διεπαφής.

12.2. Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να γνωρίζουν και να μπορούν να εφαρμόσουν βασικούς κανόνες της εργονομίας λογισμικού
- να αναπτύξουν δεξιότητες σχεδιασμού περιβάλλοντος διεπαφής
- να μπορούν να υλοποιούν ένα απλό, φιλικό και εύχρηστο περιβάλλον διεπαφής

12.3. Οδηγίες – επισημάνσεις

Επισημαίνουμε ότι η προσοχή του διδάσκοντα θα πρέπει να επικεντρωθεί και στα παρακάτω σημεία:

- Να τονιστεί τους μαθητές ότι η κατάχρηση πολυμεσικών δεδομένων σε μια εφαρμογή, συχνά επιφέρει αποτελέσματα διαφορετικά από τα επιδιωκόμενα. Πολλές φορές η παρουσία ιδιαίτερα πολλών πολυμεσικών δεδομένων στην εφαρμογή λειτουργεί εις βάρος της αποτελεσματικότητάς της. Ο χρήστης μπορεί να μαγνητίζεται από την παρουσία τους, η προσοχή του να παρασύρεται από την ελκυστική μορφή της και να χάνει το περιεχόμενο της εφαρμογής.
- Η παροχή ανάδρασης από την μεριά του υπολογιστή προς τον χρήστη είναι πολύ σημαντικός παράγοντας για τη σωστή λειτουργία της επικοινωνίας του χρήστη με την εφαρμογή. Ο χρήστης δεν θα πρέπει να έχει την αίσθηση ότι ο υπολογιστή τον αγνοεί. Σε μια τέτοια περίπτωση γρήγορα απογοητεύεται και σταματά τη χρήση της εφαρμογής. Η εφαρμογή θα πρέπει πάντοτε, όποτε ο χρήστης ζητά μια διεργασία, να του αποκρίνεται. Κι αν δεν μπορεί να εμφανίσει άμεσα τα αποτελέσματα της διεργασίας που ζήτησε ο χρήστης, θα πρέπει να του εμφανίζει σχετικό ενημερωτικό μήνυμα.
- Θα πρέπει να περάσει στους μαθητές η αίσθηση ότι θέμα σχεδίασης διεπαφής χρήστη δεν τίθεται μόνο σε γραφικό περιβάλλον, αλλά σε κάθε υπολογιστικό περιβάλλον. Επιπλέον, η σχεδίαση διεπαφής χρήστη αφορά τόσο το λογισμικό όσο και το υλικό των υπολογιστικών συστημάτων.

12.4. Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

ένα (1) δίωρο μάθημα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του 1ου μαθήματος είναι οι μαθητές να μπορούν:

- να γνωρίσουν και να μπορούν να εφαρμόζουν βασικούς κανόνες της εργονομίας λογισμικού,
- να αναπτύξουν δεξιότητες σχεδιασμού περιβάλλοντος διεπαφής
- να μπορούν (με τη βοήθεια του διδάσκοντα αν απαιτείται) να υλοποιούν ένα απλό, φιλικό και εύχρηστο περιβάλλον διεπαφής

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενες μέθοδοι και μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής

Περιεχόμενα θεωρητικής παρουσίασης (παράγραφοι από το βιβλίο του μαθητή)

- Διεπαφή χρήστη
- Τύποι διεπαφής χρήστη (χαρακτηριστικά γραφικών διεπαφών χρήστη)
- Γενική σχεδίαση διεπαφής χρήστη (φάσεις ανάπτυξης διεπαφής χρήστη)
- Οπτική σχεδίαση διεπαφής χρήστη (το χρώμα, μηνύματα λάθους)

- Ηχητική σχεδίαση διεπαφής χρήστη

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να παρουσιαστεί από το βιβλίο του μαθητή, το παράδειγμα της παραγράφου 12.3, έτσι ώστε να γίνει αντιληπτό στους μαθητές ότι η διεπαφή χρήστη αφορά και μη γραφικά περιβάλλοντα .

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη:

- η δραστηριότητα ΔΤ1 και αν υπάρχει διαθέσιμος χρόνος και η δραστηριότητα ΔΤ2

Να αντιμετωπιστούν στο εργαστήριο:

- η δραστηριότητα ΔΕ1

Για τη δραστηριότητα ΔΕ1 να γίνει σαφές στους μαθητές ότι πέραν των δεδομένων που τους παρέχονται, θα πρέπει να λάβουν υπόψη τους και άλλα δεδομένα που είναι απαραίτητο να συνυπολογιστούν.

Να δοθούν στους μαθητές για αντιμετώπιση στο σπίτι :

- η δραστηριότητα ΔΣ1 ή η δραστηριότητα ΔΣ2.
- Η δραστηριότητα ΔΣ3 θεωρείται προαπαιτούμενη της δραστηριότητας ΔΕ1. Εφόσον το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον διδαχθεί ανά μία ώρα σε δύο διαφορετικές ημέρες την εβδομάδα και όχι σε δύο συνεχόμενες ώρες κατά την ίδια ημέρα, μπορεί να δοθεί η δραστηριότητα ΔΣ3 για επίλυση στο σπίτι.

Για τις δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές να δίνονται διευκρινήσεις που τυχόν απαιτούνται.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους ανατίθενται στο σπίτι αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους.

Τεστ αξιολόγησης επίδοσης

Α. Συμπληρώστε με σωστό ή λάθος

1. Τα μηνύματα λάθους δεν πρέπει να έχουν επικριτικό τόνο προς το χρήστη.
2. Από τα βασικά χαρακτηριστικά των γραφικών διεπαφών χρήστη είναι τα προγράμματα γραφικών.
3. Ο τόνος της φωνής του αφηγητή παίζει σημαντικό ρόλο στην επίδραση της εφαρμογής στο χρήστη.
4. Ο πιο φιλικός τύπος διεπαφής χρήστη είναι ο GUI.

Β. Συμπληρώστε τα κενά με τη λέξη που λείπει

1. Τα μηνύματα λάθους πρέπει να είναι.....
2. Χρόνος ..... λέγεται ο χρόνος που χρειάζεται το σύστημα για να αρχίσει να παρέχει τα αποτελέσματα της ενέργειας που ζήτησε ο χρήστης.
3. Ως τεχνική κωδικοποίησης στη σχεδίαση διεπαφής χρήστη αναγνωρίζεται το .....

Απαντήσεις τεστ αξιολόγησης επίδοσης

Α1 : σωστό	Α2 : λάθος	Α3 : σωστό	Α4 : σωστό
Β1: εξειδικευμένα	Β2: απόκρισης	Β3: χρώμα	

12.5. Προβληματισμοί και θέματα προς συζήτηση

Σαν θέμα συζήτησης που μπορεί να τεθεί από τη μεριά του διδάσκοντα εφόσον υπάρχει διαθέσιμος χρόνος προτείνεται η αναφορά εκ μέρους του στην υλική διάσταση της διεπαφής χρήστη.

12.6. Προτεινόμενες πηγές πληροφόρησης

Βιβλιογραφία

Όλη η προτεινόμενη βιβλιογραφία του κεφαλαίου, όπως καταγράφεται στο βιβλίο του μαθητή.

Επιπλέον διευθύνσεις στο Διαδίκτυο

- http://mjmi.engin.umich.edu/Conf/NO95/TALKS/TE14.1.html
- http://explorer.scrtec.org/aux/about/wInterfaceInfo.shtml
- http://www1.viaweb.com/softpro/misc-user-interface-design.html
- http://www1.pitt.edu/~laudato/UI/
- http://www.iol.ie/~nmurphy/hire.htm
- http://www.acm.org/sigchi/chi97/proceedings/tutorial/dmh.htm

12.7. Απαντήσεις ερωτήσεων κεφαλαίου βιβλίου μαθητή

1. Οι διαφορετικές αποδόσεις που έχουν κατά καιρούς χρησιμοποιηθεί ή και χρησιμοποιούνται ακόμα για τον όρο user interface, υποδηλώνουν άλλος λιγότερο και άλλος περισσότερο αυτό που πραγματικά αντιπροσωπεύει ο όρος. Σκοπός του θέματος συζήτησης δεν είναι να επαναπροσδιοριστεί η απόδοση του όρου, αλλά να συνειδητοποιήσουν οι μαθητές την ποικιλία των αποδόσεων που μπορούν να χρησιμοποιούνται για κάποιον ξενόγλωσσο όρο (και όχι μόνο της πληροφορικής). Ακόμα, και ανάλογα και με τις προτιμήσεις τους που θα προκύψουν από τη συζήτηση, να διαπιστώσουν ότι το προφανέστερο για αυτούς, δεν είναι προφανές για όλους.
2. Η συζήτηση μπορεί να στηριχθεί στην άποψη *"μια εικόνα αξίζει όσο χίλιες λέξεις"*. Η δυναμική της εικόνας είναι αναμφισβήτητη. Η συζήτηση μπορεί να διευρυνθεί ιδιαίτερα. Χαρακτηριστικά μπορούν να αναφερθούν σαν πλεονε-

κτήματα : άμεση αντίληψη, κωδικοποίηση αναπαραστάσεων, περιεκτικότητα γραφικού δεδομένου σε αντιπαράθεση με το κείμενο.

3. Δες παράγραφο 12.2.1.
4. Δες παράγραφο 12.3.1.
5. Στη συζήτηση να γίνει προσπάθεια να αναδειχθούν μέσα από προσωπικές εμπειρίες οι χρωματικές κωδικοποιήσεις των μαθητών σχετικά με καταστάσεις. Να γίνει προσπάθεια αιτιολόγησης αυτών των κωδικοποιήσεων. Να καταγραφούν για να καταφανεί ποιες είναι οι συνηθέστερες και κοινές μεταξύ των μαθητών.
6. Δες παράγραφο 12.4.2
7. Η απάντηση σε μεγάλο βαθμό είναι υποκειμενική. Όμως σαν γενικά αποδεχτές σκέψεις και προτάσεις των μαθητών παραθέτονται : ηχητικό μηνύματα λανθασμένης καταχώρησης δεδομένων, ηχητικό μήνυμα αποπεράτωσης διαδικασίας επεξεργασίας.

12.8. Απαντήσεις δραστηριοτήτων κεφαλαίου τετραδίου μαθητή

Σημείωση:

Σε πολλές από τις προτεινόμενες δραστηριότητες, οι απαντήσεις είναι σχετικά υποκειμενικές. Κατά συνέπεια, σε αυτές τις περιπτώσεις δεν προτείνονται παρά μόνο ενδεικτικές απαντήσεις.

Απαντήσεις δραστηριοτήτων

➤ ***Στην τάξη***

ΔΤ1.

Η δραστηριότητα στοχεύει στο να παρακινήσει τους μαθητές να θέσουν σε λειτουργία την φαντασία τους για να εντοπίσουν την πιο κατάλληλη μεταφορά.

Επιτυχημένη μεταφορά για το χρήστη είναι αυτή του αρχαιολόγου και για το δείκτη του ποντικιού αυτή της σκαπάνης.

ΔΤ2.

Η δραστηριότητα στοχεύει ταυτόχρονα στο να εξασκηθούν οι μαθητές στο γνωστικό αντικείμενο του συγκεκριμένου χώρου και στο να καλλιεργήσουν δεξιότητες μεθοδολογικού χαρακτήρα.

Οι απαντήσεις είναι σχετικά υποκειμενικές. Ενδεικτικά αναφέρονται σαν πιθανές οι εξής:

- εκπαιδευτικές εφαρμογές διδασκαλίας γλώσσας : αφήγηση (για επίδειξη προφοράς),
- εφαρμογές απευθυνόμενες σε άτομα με ειδικές ανάγκες (προβλήματα όρασης): όλες οι μορφές ηχητικών δεδομένων (αφήγηση, μουσική επένδυση, ηχητικά σήματα).

➤ Στο εργαστήριο

ΔΕ1.

Η προτεινόμενη δραστηριότητα έχει την μορφή μικρού έργου (project). Επιχειρείται δηλαδή να συμπεριλάβει σε αυτήν όλες τις βασικές παραμέτρους που λαμβάνονται υπόψη κατά τη σχεδίαση της διεπαφής χρήστη. Σαν μόνο κατάλληλο περιβάλλον υλοποίησης της εφαρμογής κρίνεται αντικειμενοστραφές, οδηγούμενο από γεγονότα, γραφικό προγραμματιστικό περιβάλλον (π.χ. Visual Basic).

Σε αυτήν την περίπτωση θα πρέπει να υποδειχθεί στους μαθητές η ύπαρξη των φακέλων έτοιμων γραφικών που περιλαμβάνει το προγραμματιστικό περιβάλλον, ώστε να αξιοποιηθούν κατάλληλα τα γραφικά.

Ιδιαίτερη προσοχή θα πρέπει να δοθεί :

- στη σταθερή θέση των πλήκτρων αλληλεπίδρασης στις διαφορετικές οθόνες της εφαρμογής,
- στη σωστή ομαδοποίηση των πλήκτρων,

- στο αν έγινε πρόβλεψη εκ μέρους των μαθητών για πλήκτρο επιστροφής στην αρχική οθόνη,
- στο αν έγινε πρόβλεψη για ύπαρξη πλήκτρου εξόδου από όλες τις οθόνες της εφαρμογής,
- στη χρήση των χρωμάτων παρασκηνίου και προσκήνιου,
- στη χρήση των γραμματοσειρών,
- στις οδηγίες χρήσης που θα διατυπώσουν στο σχετικό πλαίσιο κειμένου.

➤ *Στο σπίτι*

ΔΣ1.

Η δραστηριότητα στοχεύει στο να αναπτύξουν οι μαθητές ικανότητες δημιουργίας και καταγραφής μηνυμάτων λαθών.

Οι απαντήσεις είναι σχετικά υποκειμενικές. Ενδεικτικά αναφέρονται σαν πιθανές απαντήσεις.

α) "Πρέπει να πληκτρολογήσετε κάποιον αριθμό"

β) "Μπορείτε να ανοίξετε μόνο *τάδε* τύπου αρχεία"

γ) "Δεν έχετε συνδεθεί στο Διαδίκτυο".

ΔΣ2.

Η δραστηριότητα αυτή έχει σκοπό να αναδείξει πτυχές της ψυχολογίας χρωμάτων μέσω των αναπαραστάσεων των μαθητών. Παράλληλα συνηγορεί υπέρ της συνεργατικής δράσης με την προτροπή σύγκρισης των αποτελεσμάτων κάθε ενός μαθητή με τα αποτελέσματα των συμμαθητών του και έτσι καλλιεργεί δεξιότητες μεθοδολογικού χαρακτήρα .

ΔΣ3.

Η δραστηριότητα στοχεύει στο να συλλογιστούν, να αναλύσουν και τελικά να προετοιμάσουν οι μαθητές την υλοποίηση της δραστηριότητας ΔΕ1.

Κεφάλαιο 13

Εκσφαλμάτωση προγράμματος

13.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου, είναι οι μαθητές να μπορούν να προσδιορίζουν τα λάθη ενός προγράμματος και να είναι σε θέση να τα χειριστούν προγραμματιστικά.

13.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να αναγνωρίζουν ένα λάθος,
- να ορίζουν τις κατηγορίες λάθους,
- να αναφέρουν εργαλεία εκσφαλμάτωσης και τον τρόπο χειρισμού τους,
- να εφαρμόζουν τις τεχνικές χειρισμού λαθών κατά το χρόνο εκτέλεσης.

13.3 Οδηγίες – επισημάνσεις

Επισημαίνουμε ότι η προσοχή του διδάσκοντα θα πρέπει να επικεντρωθεί και στα παρακάτω σημεία:

1. Να γίνει σαφές στους μαθητές ότι η διαδικασία της δοκιμής και της εκσφαλμάτωσης ενός προγράμματος, είναι απαραίτητη πριν το πρόγραμμα παραδοθεί για λειτουργία σε πραγματικό περιβάλλον εργασίας.

2. Προσπαθήστε να περιγράψετε τα εργαλεία εκσφαλμάτωσης που προσφέρει το προγραμματιστικό περιβάλλον του εργαστηρίου. Ο μαθητής πρέπει να τα μάθει και να τα χρησιμοποιήσει για να μπορέσει να κατανοήσει τις τεχνικές εκσφαλμάτωσης.

3. Ο μαθητής, πρέπει να αντιληφθεί ότι, δεν είναι δυνατό κατά τη φάση ανάπτυξης ενός προγράμματος, να προβλέψουμε όλα τα λάθη που πιθανόν να παρουσιαστούν στο περιβάλλον εργασίας του προγράμματος. Να μάθουν οι μαθητές να συμπεριλαμβάνουν στις διαδικασίες κώδικα και εντολές χειρισμού λαθών που μπορεί να προκύψουν από απρόβλεπτες καταστάσεις.

13.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

ένα (1) δίωρο μάθημα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να αναγνωρίζουν ένα λάθος,
- να ορίζουν τις κατηγορίες λάθους,
- να αναφέρουν εργαλεία εκσφαλμάτωσης και τον τρόπο χειρισμού τους,
- να εφαρμόζουν τις τεχνικές χειρισμού λαθών κατά το χρόνο εκτέλεσης.

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής, περιβάλλον ανάπτυξης εφαρμογών σε υπολογιστή

Περιεχόμενα θεωρητικής παρουσίασης

- Κατηγορίες λαθών
- Εκσφαλμάτωση
- Εργαλεία εκσφαλμάτωσης
- Χειρισμός λαθών κατά το χρόνο εκτέλεσης

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Πρέπει να αναφερθούν παραδείγματα λαθών και κυρίως σημαντικά λάθη που έχουν παρουσιαστεί κατά καιρούς στο λογισμικό και το υλικό των υπολογιστών (παράγραφοι 13.1 και 13.2) ώστε να γίνει κατανοητή η σπουδαιότητα της εκσφαλμάτωσης.

Δραστηριότητες από το τετράδιο του μαθητή

Να αντιμετωπιστούν στην τάξη, με προτεινόμενη σειρά :

- η δραστηριότητα ΔΤ1.
- η δραστηριότητα ΔΤ2.
- προαιρετικά η δραστηριότητα ΔΤ3.

Η δραστηριότητα ΔΤ1, προϋποθέτει τον προσδιορισμό συγκεκριμένου προγράμματος για να είναι σε θέση οι μαθητές να καταγράψουν τα πιθανά λάθη που θα προκύψουν.

Η δραστηριότητα ΔΤ2, πρέπει να πραγματοποιηθεί αφού ο μαθητής έρθει πρώτα σε επαφή με εργαλεία εκσφαλμάτωσης στο σχολικό εργαστήριο.

Για τη δραστηριότητα ΔΤ3, αν δεν υπάρχει χρόνος να αντιμετωπιστεί στην τάξη, μπορεί να αναφερθεί ως τρόπος προσέγγισης η παράγραφος 13.4. του βιβλίου μαθητή.

Να υλοποιηθεί στο εργαστήριο:

- η δραστηριότητα ΔΕ1.
- προαιρετικά η δραστηριότητα ΔΕ2.

Η δραστηριότητα ΔΕ1, προϋποθέτει ότι έχει ήδη υλοποιηθεί το πρόγραμμα του τριώνυμου του κεφαλαίου 11. Αν ο μαθητής δεν έχει τη δυνατότητα να κατασκευάσει το συγκεκριμένο παράδειγμα, επιλέξτε κάποιο παρόμοιο για την επίδειξη των εργαλείων εκσφαλμάτωσης.

Πριν την υλοποίηση της δραστηριότητα ΔΕ2, πρέπει να περιγράψετε στους μαθητές τους κωδικούς λάθους που επιστρέφει το προγραμματιστικό εργαλείο του εργαστηρίου για κάθε λάθος. Οι κωδικοί λάθους που επιστρέφει η Visual Basic

για τα λάθη που είναι δυνατό να παρουσιαστούν κατά την εκτέλεση της συγκεκριμένης εργασίας, μπορούν εύκολα να αναφερθούν από το αντίστοιχο λυμένο παράδειγμα που έχουμε συμπεριλάβει στο συνοδευτικό υλικό του βιβλίου. Η υλοποίηση της συγκεκριμένης δραστηριότητας είναι χρονοβόρα και μπορεί απλά να επιδειχτεί ο τρόπος αντιμετώπισής της από το συνοδευτικό υλικό του βιβλίου.

Να δοθούν στους μαθητές για αντιμετώπιση στο σπίτι :

- οι δραστηριότητες ΔΣ1 και ΔΣ2.

Η δραστηριότητα ΔΣ1, αν και έχει κοινά στοιχεία με την δραστηριότητα ΔΕ2, διαφοροποιείται γιατί αναφέρεται στη αντιγραφή αρχείων από και προς οποιοδήποτε αποθηκευτικό μέσο.

Για την αντιμετώπιση των ασκήσεων στο τετράδιο, δεν είναι απαραίτητο να αναφερθούν στους μαθητές, οι κωδικοί που επιστρέφει το προγραμματιστικό περιβάλλον για τα πιθανά λάθη, αλλά είναι δυνατό να χρησιμοποιήσουν σταθερές. Για παράδειγμα όταν χρειάζεται να ελέγξουν αν ο οδηγός δισκέτας είναι έτοιμος, μπορούν να χρησιμοποιήσουν μια σταθερά της μορφής DiskNotReady.

Για τις δραστηριότητες για το σπίτι που αναθέτονται στους μαθητές να δίνονται τυχόν διευκρινήσεις που απαιτούνται από την πλευρά των μαθητών.

Οι απαντήσεις των μαθητών στις δραστηριότητες που τους ανατίθενται στο σπίτι αποτελούν θέμα συζήτησης σε επίπεδο τάξης μετά την υλοποίησή τους.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Τα λάθη που προκύπτουν κατά το χρόνο εκτέλεσης είναι δυνατό να προκαλέσουν τη διακοπή του προγράμματος.
2. Η ανίχνευση των λογικών λαθών ενός προγράμματος γίνεται από το μεταγλωτιστή.
3. Με τα εργαλεία εκσφαλμάτωσης παρατηρούμε τι συμβαίνει στο παρασκήνιο της εφαρμογής κατά το χρόνο εκτέλεσης.
4. Σε ένα πρόγραμμα μπορούμε να συμπεριλάβουμε γενικές διαδικασίες χειρισμού λάθους.

5. Τα λάθη κατά το χρόνο εκτέλεσης ανιχνεύονται με τα εργαλεία εκσφαλμάτωσης.
6. Τα πλαίσια διαλόγου μπορούν να χρησιμοποιηθούν ως εργαλεία εκσφαλμάτωσης.

Συμπληρώστε τα κενά με τη λέξη που λείπει.

7. Η λάθος συγγραφή μιας εντολής κώδικα είναι λάθος που εμφανίζεται κατά το χρόνο ..... του προγράμματος*
8. Κατά την διαδικασία της εκσφαλμάτωσης μπορούμε να διακόψουμε την εκτέλεση του προγράμματος χρησιμοποιώντας ....
9. Τα λογικά λάθη εντοπίζονται με τη διαδικασία ..... του προγράμματος.

Απαντήσεις τεστ αξιολόγησης επίδοσης

1 : σωστό

2 : λάθος

3 : σωστό

4 : σωστό

5 : λάθος

6 : σωστό

7 : σχεδιασμού

8 : σημεία διακοπής

9 : ελέγχου

13.5 Προβληματισμοί και θέματα προς συζήτηση

Θέματα προβληματισμού και συζήτησης μπορούν να προκύψουν από όλες τις δραστηριότητες που προτείνονται στο συγκεκριμένο κεφάλαιο. Συγκεκριμένα μπορούν να προταθούν συζητήσεις σχετικά με :

- Την ποιότητα του λογισμικού και του υλικού

- Τις διαδικασίες ελέγχου που εφαρμόζουν μεγάλες εταιρίες για τον εντοπισμό λαθών (beta-versions λογισμικού, τεστ υλικού, κ.ά.)
- Να αναφέρουν οι μαθητές, περιπτώσεις λαθών που αντιμετωπίζουν στα προγράμματά που χρησιμοποιούν καθημερινά ή περιπτώσεις λαθών που εντόπισαν κατά την ανάπτυξη προγραμμάτων.

13.6 Προτεινόμενες πηγές πληροφόρησης

Βιβλιογραφία

Όλη η προτεινόμενη βιβλιογραφία και οι διευθύνσεις στο διαδίκτυο, όπως καταγράφονται στο βιβλίο του μαθητή.

Επιπλέον βιβλιογραφία

1. Lyn dupre : Bugs in Writing-A Guide to Debugging Your Prose, Addison-Wesley Pub Co, 1998
2. Glenford J. Myers : Art of Software Testing, John Wiley & Sons, 1979

13.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή

Σε ορισμένες από τις ερωτήσεις οι απαντήσεις είναι ελευθέρου τύπου και ταυτόχρονα σε μεγάλο βαθμό υποκειμενικές. Κατά συνέπεια δεν προτείνονται συγκεκριμένες απαντήσεις. Οι υπόλοιπες ερωτήσεις προέρχονται από το θεωρητικό τμήμα του κεφαλαίου και οι απαντήσεις μπορούν πολύ εύκολα να αναζητηθούν σε αυτό :

Ερώτηση	Απάντηση
3	13.4
4	13.3

13.8. Απαντήσεις δραστηριοτήτων τετραδίου μαθητή

➤ Στην τάξη

Σημείωση:

Στις προτεινόμενες δραστηριότητες, οι απαντήσεις είναι ελευθέρου τύπου και ταυτόχρονα σε μεγάλο βαθμό υποκειμενικές. Κατά συνέπεια δεν προτείνονται συγκεκριμένες απαντήσεις, παρά μόνο αναφέρονται πληροφοριακά στοιχεία όπου κρίνεται απαραίτητο.

ΔΤ3.

Πριν από την εκτέλεση της δραστηριότητας, ο μαθητής πρέπει να έρθει σε επαφή με εργαλεία εκσφαλμάτωσης.

ΔΤ3.

Ορισμένες τεχνικές χειρισμού λαθών κατά το χρόνο εκτέλεσης αναφέρονται στη παράγραφο 13.4 του κεφαλαίου.

➤ Στο εργαστήριο

Οι προτεινόμενες δραστηριότητες, αναφέρονται στην υλοποίηση διαδικασιών για το χειρισμό λαθών που παρουσιάζονται κατά το χρόνο εκτέλεσης ενός προγράμματος και κατά συνέπεια η ανάπτυξή τους είναι σε μεγάλο βαθμό υποκειμενική και εξαρτάται από το προγραμματιστικό εργαλείο του εργαστηρίου. Στο συνοδευτικό υλικό του βιβλίο υπάρχουν λυμένα τα παραδείγματα μέσα από τη δική μας προσέγγιση και είναι δυνατό να τα συμβουλευτείτε ή να τα χρησιμοποιήσετε αυτούσια.

➤ Στο σπίτι

Οι προτεινόμενες δραστηριότητες, αναφέρονται στην υλοποίηση διαδικασιών για το χειρισμό λαθών που παρουσιάζονται κατά το χρόνο εκτέλεσης ενός προγράμματος και κατά συνέπεια η ανάπτυξή τους είναι σε μεγάλο βαθμό υποκειμενική και εξαρτάται από το προγραμματιστικό εργαλείο του εργαστηρίου. Στο συνοδευτικό υλικό του βιβλίο υπάρχουν λυμένα τα παραδείγματα μέσα από τη δική μας προσέγγιση και είναι δυνατό να τα συμβουλευτείτε ή να τα χρησιμοποιήσετε αυτούσια.

Η δραστηριότητα ΔΣ2 είναι εξαρτώμενη από τη διαδικασία χειρισμού λαθών που θα κατασκευάσει ο μαθητής κατά την υλοποίηση της δραστηριότητας ΔΣ1.

Κεφάλαιο 14

Αξιολόγηση - Τεκμηρίωση

14.1 Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί ο μαθητές να αξιολογούν τα προγράμματα και να συντάσσουν την τεκμηρίωση αυτών, και τέλος να διακρίνουν τα στάδια στον κύκλο ζωής των προγραμμάτων.

14.2 Ειδικοί διδακτικοί σκοποί

Μετά την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- να αξιολογούν τα προγράμματα βάσει των κριτηρίων,
- να αντιλαμβάνονται το συγκερασμό των κριτηρίων σε μία λύση και να αποφασίζουν, ανάλογα με το πρόβλημα, ποιο κριτήριο θα έχει τον κύριο ρόλο στη λύση που επιλέγουν,
- να αναγνωρίσουν τα θέματα για τα οποία πρέπει να υπάρχει τεκμηρίωση. Καθώς και τη σημασία της κατά τη φάση της συντήρησης,
- να είναι σε θέση να συντάξουν φάκελο προγράμματος με τα απαραίτητα στοιχεία, όπως αλγόριθμο, πρόγραμμα, δεδομένα ελέγχου, αποτελέσματα, τεκμηρίωση για τον προγραμματιστή και το χρήστη,
- να διακρίνουν τις φάσεις ζωής ενός προγράμματος.

14.3 Οδηγίες – επισημάνσεις

Πρέπει να τονιστεί ότι δεν υπάρχουν καλές και κακές λύσεις, αλλά λύσεις που ικανοποιούν τα κριτήρια αξιολόγησης. Κάθε πρόγραμμα μπορεί να έχει πολλές λύσεις τις οποίες μπορούμε να περιορίσουμε με τη βοήθεια των κριτηρίων. Θα πρέπει να δοθεί έμφαση στα συμπεράσματα που προκύπτουν από τον κύκλο ζωής προγράμματος, ώστε να γίνει σαφές ότι η απασχόληση με ένα πρόγραμμα δεν σταματά παρά μόνο όταν το πρόγραμμα αποσυρθεί. Αυτό ακριβώς αναδεικνύει την σημασία της τεκμηρίωσης. Ενθαρρύνετε τους μαθητές να δοκιμάσουν τα προγράμματα που έχουν λύσει μέχρι τώρα δοκιμάζοντας την αξιοπιστία τους και κάνοντας τις σχετικές βελτιώσεις. Παροτρύνετε τους μαθητές να διατυπώνουν εναλλακτικές λύσεις για το ίδιο πρόβλημα.

14.4 Προγραμματισμός μαθημάτων κεφαλαίου

Προτεινόμενος αριθμός μαθημάτων

δύο (2) δίωρα μαθήματα

Σχέδιο 1ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να ορίζουν τα κριτήρια αξιολόγησης

Χώρος υλοποίησης μαθήματος

τάξη και εργαστήριο.

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής

Περιεχόμενα θεωρητικής παρουσίασης

- **Κάλυψη παραγράφων 14.1.** Ορισμός της αξιολόγησης και των κριτηρίων της.

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθούν οπωσδήποτε τα παραδείγματα 2 και 3 του κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Θα πρέπει να δοθεί στους μαθητές μία δραστηριότητα από τις ΔΤ1, ΔΤ3 και ΔΤ4 σύμφωνα με την εκτίμησή σας.

Από τις δραστηριότητες στο εργαστήριο μπορείτε να δώσετε την ΔΕ1 και την ΔΕ2 μαζί, για να τονίσετε την μικρή αλλαγή, άρα την ευελιξία του προγράμματος.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Οι πολλοί έλεγχοι επί των δεδομένων σε ένα πρόγραμμα αυξάνουν την ταχύτητα του προγράμματος.
2. Ένα πρόγραμμα είναι αξιόπιστο όταν απορρίπτει τα δεδομένα που είναι εκτός προδιαγραφών.
3. Ευέλικτο λέμε το πρόγραμμα που έχει πολλές λύσεις.
4. Ταχύ λέγεται το πρόγραμμα που επεξεργάζεται γρήγορα τις πληροφορίες.
5. Απλό πρόγραμμα λέμε το πρόγραμμα που χρησιμοποιεί τις απλούστερες δομές για την λύση του προβλήματος.
6. Κάθε πρόβλημα έχει μία και μοναδική λύση.

Άσκηση

Δίνονται οι μετρήσεις από τους σταθμούς ρύπανσης μιας πόλης από το πληκτρολόγιο. Ζητείται να υπολογίζεται η μέση τιμή. Οι σταθμοί μέτρησης είναι 10 αλλά ενδέχεται να μην έχουν στείλει όλοι τις μετρήσεις τους, ενώ ενδέχεται σύντομα να αυξηθούν σε 20.

Απαντήσεις τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1: λάθος	2: σωστό	3: λάθος
4: λάθος	5: σωστό	6: λάθος

Άσκηση

Υπάρχουν δύο τρόποι με τους οποίους μπορεί να αντιμετωπιστεί το παραπάνω πρόβλημα:

Ο αριθμός των μετρήσεων να εισάγεται στο πρόγραμμα κάθε φορά που εκτελείται.

Αυτός ο τρόπος είναι ο καλύτερος στην περίπτωση που το πλήθος των μετρήσεων αλλάζει συχνά για παράδειγμα ξέρουμε ότι κάποιοι σταθμοί δεν λειτουργούν συνεχώς.

Αν οι αλλαγές αυτές είναι σπάνιες, για παράδειγμα όταν δημιουργείται ένας νέος σταθμός μέτρησης, τότε η καλύτερη λύση είναι η χρησιμοποίηση μίας σταθεράς, όπως έχει αναφερθεί και στο κεφάλαιο 7.

Σε αυτή την περίπτωση χρειάζεται βέβαια να αλλαχτεί το πρόγραμμα σε πιθανή αλλαγή των σταθμών μέτρησης αλλά αρκεί μία αλλαγή στην εντολή N=6, που εξασφαλίζει ταχύτητα και προστασία από τα πιθανά λάθη.

Σχέδιο 2ου μαθήματος

Διδακτικοί στόχοι

Διδακτικοί στόχοι του μαθήματος είναι οι μαθητές να μπορούν:

- να συντάσσουν φάκελο τεκμηρίωσης προγράμματος.
- να διακρίνουν της φάσεις ζωής προγράμματος.

Χώρος υλοποίησης μαθήματος

τάξη, εργαστήριο

Προτεινόμενα υλικά και εποπτικά μέσα διδασκαλίας

πίνακας, διαφάνειες, υπολογιστής

Περιεχόμενα θεωρητικής παρουσίασης

- **Κάλυψη παραγράφων 14.2, 14.3.** Λόγοι, κατηγορίες και αναγκαιότητα της τεκμηρίωσης. Η σημασία του VTOC στην τεκμηρίωση μεγάλων προγραμμάτων.

Περιεχόμενα πρακτικής εφαρμογής

Εφαρμογές, παραδείγματα από το βιβλίο του μαθητή

Θα πρέπει να διδαχθεί το 7ο παράδειγμα του κεφαλαίου από το βιβλίο του μαθητή.

Δραστηριότητες από το τετράδιο του μαθητή

Η δραστηριότητα ΔΕ5 προσφέρεται για ομαδική δουλειά, φάκελο τεκμηρίωσης και δοκιμαστικό αρχείο.

Ενθαρρύνετε όσους θέλουν να δώσουν εναλλακτική λύση στις ΔΤ2, ΔΣ1 και ΔΣ2.

Να δοθεί η ΔΣ4 στην οποία θα χρησιμοποιηθούν σαν ενότητες τα παραδείγματα 2 και 3 του βιβλίου.

Στην δραστηριότητα ΔΕ4 ο όρος «αξιόπιστο ψηφίο ελέγχου» σημαίνει ότι ένας αναριθμητισμός θα πρέπει να παράγει διαφορετικό ψηφίο ελέγχου. Τονίστε τη σημασία της τεκμηρίωσης για άλλη μία φορά.

Τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1. Οι έλεγχοι είναι το τελευταίο στάδιο στην ζωή ενός προγράμματος.
2. Με το VTOC αποτυπώνουμε την λογική του προγράμματος.
3. Η τεκμηρίωση χρήστη δίνει πληροφορίες για την χρήση του προγράμματος.
4. Χωρίς τεκμηρίωση η ανάπτυξη του προγράμματος γίνεται ταχύτερα, γι' αυτό πρέπει να την αποφεύγουμε.
5. Το κόστος υλικού είναι μικρότερο του κόστους λογισμικού σε μία μηχανογραφική λύση.

Άσκηση

Δίνονται οι πωλήσεις μίας εταιρείας, για κάθε πώληση δίνεται ο κωδικός πωλητή, ο κωδικός προϊόντος, η πωληθείσα ποσότητα, και η τιμή μονάδας.

Οι πωλήσεις δίνονται με αύξουσα σειρά του κώδικα πωλητή και του κώδικα προϊόντος. Κάθε πωλητής έχει πολλές πωλήσεις του ίδιου αλλά και διαφορετικών προϊόντων.

Μετά την επεξεργασία των στοιχείων εκτυπώνεται η παρακάτω κατάσταση. Να σχεδιάσεις τον οπτικό πίνακα αποτελεσμάτων (VTOC).

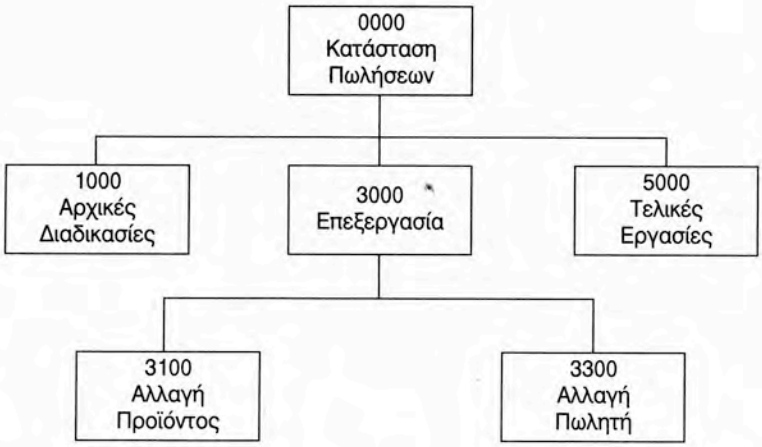
ΚΑΤΑΣΤΑΣΗ ΠΩΛΗΣΕΩΝ ΠΩΛΗΤΩΝ		
Κωδικός Πωλητή	**Κωδικός Προϊόντος**	**Αξία**
XXXXXX	XXXXXX	XXXXXXX
	XXXXXX	XXXXXXX .....
Σύνολο Αξίας Πωλητή		XXXXXXXXXX
XXXXXX	XXXXXX	XXXXXXX
	XXXXXX	XXXXXXX .....
Σύνολο Αξίας Πωλητή		XXXXXXXXX
.....	.....	.....
Γενικό Σύνολο Αξίας		XXXXXXXX

Απαντήσεις τεστ αξιολόγησης επίδοσης

Συμπληρώστε με σωστό ή λάθος

1: λάθος	2: λάθος	3: σωστό
4: λάθος	5: σωστό	

Άσκηση



14.5 Προβληματισμοί και θέματα για συζήτηση

Θέματα προβληματισμού και συζήτησης μπορούν να προκύψουν από όλες τις δραστηριότητες που προτείνονται στο συγκεκριμένο κεφάλαιο. Συγκεκριμένα μπορούν να προταθούν συζητήσεις σχετικά με :

- Τη σημασία των κριτηρίων αξιολόγησης για την παραγωγή ποιοτικών προγραμμάτων. Να τονιστεί ότι η σημασία της ποιότητας είναι αμφίπλευρη. Δημιουργεί ευχαριστημένους χρήστες και σίγουρους για την δουλεία τους προγραμματιστές.
- Τη σημασία της καλής τεκμηρίωσης, της προσεκτικής επιλογής της λύσης σε σχέση με το κόστος λογισμικού, διαχωρίζοντας κοινές και εξειδικευμένες εφαρμογές.

14.6 Προτεινόμενες πηγές πληροφόρησης

Διευθύνσεις στο διαδίκτυο

- www.gold.net.au/~~other/techical/delphi-pascal/pascal-ArabicToRoman.html

- http/hackensackhigh.org/roman.html

14.7 Απαντήσεις ερωτήσεων βιβλίου μαθητή.

Όλες οι ερωτήσεις είναι θεωρητικές και ο απαντήσεις βρίσκονται στο κεφάλαιο 14ο του βιβλίου.

14.8 Απαντήσεις δραστηριοτήτων τετραδίου μαθητή.

➤ Στην τάξη

ΔΤ1.

Να δίνεται για ημέρα αναφοράς η σημερινή ημέρα από το πληκτρολόγιο. Πρέπει να ελέγχεται αν δόθηκε σωστή ημέρα. Ό έλεγχος είναι:

```
.....
!!Έλεγχος σωστής Ημέρας
Αρχή
D<-0
ΟΣΟ d=0 ΕΠΑΝΑΛΑΒΕ
    ΔΙΑΒΑΣΕ Ημέρα
    ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 7
        ΑΝ Ημέρα=day(ι) ΤΟΤΕ
                D<-ι
                I<-8
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
.....
```

Όπου ο πίνακας day περιέχει τα λεκτικά των ημερών. Η μεταβλητή d θα διαφυλάξει την τιμή του ι για τον τελικό προσδιορισμό της ημέρας. Όταν ο αριθμός είναι αρνητικός αυτό θα σημαίνει ότι ζητείται ημέρα προς τα "πίσω". Δηλαδή όταν η διαφορά είναι –10, το ερώτημα θα είναι: Πριν από 10 ημέρες ποια ημέρα ήταν; Τότε θα υπολογίζεται πάλι το υπόλοιπο της ακέραιης διαίρεσης και θα βρί-

σκεται η ημέρα που απέχει 3 μέρες από την Τετάρτη πηγαίνοντας προς τα πίσω. Αυτή είναι η Κυριακή.

Επειδή κάθε εβδομάδα έχει 7 ημέρες, γίνεται η ακέραιη διαίρεση. Το υπόλοιπο αυτής της διαίρεσης, σε συνδυασμό με την ημέρα βάσης, θα μας δίνει την νέα ημέρα.

ΔΤ2.

Απορρίπτονται οι περιπτώσεις 3, 5, 6 και 7.

Εναλλακτική λύση δίσεκτου έτους.

```
.....
Δ<-Ψευδής
ΑΝ  Ε mod 4 = 0 ΤΟΤΕ
        Δ<-Αληθής
        Αν  Ε mod 100 = 0 ΤΟΤΕ
                Δ<-Ψευδής
                Αν  Ε mod 400 = 0 Τότε
                    Δ<-Αληθής
                ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
.....
```

ΔΤ3.

Κάντε την παρατήρηση ότι είναι λογικό να υποθέσει κανείς ότι για την σύγκριση τεσσάρων θα γίνονται με τη σειρά οι εξής συγκρίσεις: α>β, β>γ, γ>δ. δηλαδή κάθε φορά που θα ζητείται η επέκταση για ένα ακόμα αριθμό, θα προστίθεται το αντίστοιχο **ΑΝ** μέσα στην επαναληπτική διαδικασία του παραδείγματος 3 του βιβλίου.

```
ΑΝ α>β ΤΟΤΕ
    temp <- α
    α<-β
    β<-temp
ΤΕΛΟΣ_ΑΝ
```

ΔΤ4.

Να εισαχθούν οι λέξεις σε ένα πίνακα και μετά να χρησιμοποιηθεί οποιοσδήποτε αλγόριθμος ταξινόμησης από το 3^{ο} κεφάλαιο. Αφού βρεθεί η λύση, ζητήστε από τους μαθητές να τροποποιήσουν το πρόγραμμα ώστε ο χρήστης να επιλέγει και να δίνει ο ίδιος το πλήθος των στηλών. Να εξετάσουν και να ορίσουν το μέγιστο αριθμό των στηλών που μπορεί να τυπώνει το πρόγραμμα τους.

➤ Στο σπίτι

ΔΣ1.

Πρέπει να δίνεται άλλη μία ημερομηνία (ηη/μμ/εεεε) μαζί με την ημέρα (πχ. Δευτέρα). Με την εισαγωγή δεύτερης ημερομηνίας βρίσκεται η απόσταση σε μέρες των δύο ημερομηνιών. Το υπόλοιπο της ακέραιης διαίρεση με το 7 θα συνδυαστεί με την ημέρα βάσης. Μπορούν να χρησιμοποιήσουν την λύση της ΔΤ1 σαν υποπρόγραμμα. Σε μία πλήρη λύση θα πρέπει να ληφθούν υπ΄ όψη οι παρατηρήσεις του 4^{ου} παραδείγματος του βιβλίου μαθητή.

ΔΣ2.

Τονίστε ότι ο τόνος (') τοποθετείται στο τελευταίο γράμμα του αριθμού, για παράδειγμα ο αριθμός 21 παριστάνεται σαν **κα'** και όχι σαν **κ'α'**.

ΔΣ3.

Στον πίνακα οι συνδυασμοί μεταξύ των συνθηκών υλοποιούν τα αποτελέσματα της έρευνας, για παράδειγμα η περιπτώσεις 3, 4, 5, 6 σημαίνουν ότι ο πελάτης θέλει οπωσδήποτε να ταξιδέψει.

Συνθήκες	Περιπτώσεις							
	1η	2η	3η	4η	5η	6η	7η	8η
Ζητά εισιτήριο Α' θέσης	Α	Α	Α	Α	Α	Α	Ψ	Ψ
Ζητά εισιτήριο Τουριστικής θέσης	Ψ	Ψ	Α	Α	Α	Α	Α	Α
Υπάρχει εισιτήριο Α' θέσης	Α	Ψ	Α	Α	Ψ	Ψ	-	-
Υπάρχει εισιτήριο Τουριστικής θέσης	-	-	Ψ	Α	Α	Ψ	Α	Ψ
Ενέργειες								
Έκδοση εισιτήριο Α' θέσης	Ν		Ν					
Έκδοση εισιτήριο Τουριστικής θέσης				Ν	Ν		Ν	
Λίστα αναμονής		Ν				Ν		Ν

Στην 4η περίπτωση η ενέργεια είναι η συμφέρουσα στον πελάτη. Αν η πολιτική της εταιρείας είναι διαφορετική, θα είναι διαφορετική και η ενέργεια. Οι περιπτώσεις 4 και 5 μπορούν να συμπτυχθούν.

ΔΣ4.

Χρησιμοποιήστε τα 2ο και 3ο παραδείγματα του βιβλίου σαν υποπρογράμματα. Στο φάκελο να υπάρχει ένα στοιχειώδες VTOC.

➤ *Στο εργαστήριο*

ΔΕ1.

Στηριχθείτε στο 1ο παράδειγμα του τετραδίου.

ΔΕ2.

Στηριχθείτε στο 1ο παράδειγμα του τετραδίου.

ΔΕ3.

Στο πρόβλημα του πλήθους των ημερών από την αρχή του χρόνου προτείνονται δύο διαφορετικοί πίνακες με διαφορετικό υπολογισμό.

Η μία λύση χρησιμοποιεί ένα πίνακα που έχει αθροιστικά τις μέρες από την αρχή του χρόνου.

Δείκτης μηνός	Μέρες
1	0
2	31
3	59
4	90
5	120
6	151
7	181
8	212
9	243
10	273
11	304
12	334

Αν η ημερομηνία είναι 20/5/εεεε τότε έχουν περάσει Μέρες (5)+20=120+20=140 ημέρες από την αρχή του χρόνου.

Η άλλη λύση χρησιμοποιεί ένα πίνακα που έχει αθροιστικά τις μέρες από την αρχή του χρόνου που είναι πάνω ή κάτω από το όριο των 30 ημερών.

Δείκτης μηνός	Μέρες
1	0
2	1
3	-1
4	0
5	0
6	1
7	1
8	2
9	3
10	3
11	4
12	4

Αν η ημερομηνία είναι 20/5/εεεε τότε έχουν περάσει (5-1)*30+Μέρες(5)+20=4*30+0+20=120+20=140 ημέρες από την αρχή του χρόνου.

Αν αντί για την αρχή του χρόνου θεωρηθεί σαν η ημερομηνία 1/1/1, μπορεί να βρεθεί πόσες μέρες μεσολάβησαν από τότε. Προσέξτε ότι πρέπει να προστεθεί και ο έλεγχος για τα δίσεκτα χρόνια.

ΔΕ4.

Να γίνει συζήτηση ποιος από τους τρόπους ψηφίου ελέγχου είναι ασφαλέστερος.

ΔΕ5.

Το μέγιστο μήκος του ρωμαϊκού αριθμού είναι 12 ψηφία (μικρότερος του 4000). Αν οι μαθητές έχουν φθάσει σε καλό επίπεδο κατανόησης των θεμάτων που έχουν παρουσιαστεί μέχρι τώρα, μια ιδέα είναι να χωρίσετε την τάξη σε ομάδες, για παράδειγμα των τριών. Κάθε ομάδα θα λύσει το πρόβλημα αναθέτοντας στα μέλη της τον σχεδιασμό και την υλοποίηση των ενοτήτων του προγράμματος. Στην συνέχεια τα προγράμματα να ανταλλαγούν μεταξύ των ομάδων για την φάση των ελέγχων. Να συζητηθούν τα πλεονεκτήματα και οι δυσκολίες της ομαδικής εργασίας. Να τονιστούν τα πλεονεκτήματα από την κατάτμηση του προγράμματος σε ενότητες.

Παράρτημα

Αρχεία δεδομένων

Οπως είναι γνωστό, ένα πρόγραμμα προκειμένου να χρησιμοποιήσει κάποια δεδομένα πρέπει αυτά να βρίσκονται στην κεντρική μνήμη. Ομως η τελευταία έχει περιορισμένο μέγεθος και επί πλέον δεν προσφέρεται για μόνιμη αποθήκευση στοιχείων. Για τους λόγους αυτούς χρησιμοποιούνται οι βοηθητικές μνήμες, που αν και πιο αργές, δεν έχουν τα προηγούμενα μειονεκτήματα. Οι πάσης φύσης πληροφορίες και δεδομένα αποθηκεύονται στις βοηθητικές μνήμες με τη μορφή αρχείων.

Η έννοια του αρχείου δεν είναι καινούργια. Στη γενική μορφή ένα αρχείο είναι μια συλλογή πληροφοριών. Πριν την εμφάνιση των υπολογιστών τα διάφορα αρχεία τηρούνταν χειρογραφικά συνήθως σε χαρτί. Κάθε αρχείο έχει μικρή χρησιμότητα, αν οι πληροφορίες που περιέχει δεν βρίσκονται εύκολα και γρήγορα. Ετσι σταδιακά οι διάφορες επιχειρήσεις και οργανισμοί αντικαθιστούν τα αρχεία σε καρτέλες με σύνολα αρχείων σε υπολογιστές.

Η ονομασία αρχείο παρέμεινε στα σύγχρονα υπολογιστικά συστήματα και σημαίνει ένα σύνολο δεδομένων οποιασδήποτε μορφής. Ετσι υπάρχουν **αρχεία προγραμμάτων** (program files), **κειμένου** (text files), αρχεία που περιέχουν προγράμματα σε γλώσσα μηχανής και ονομάζονται **δυαδικά** (binary files) ή **εκτελέσιμα** (executable) και αρχεία που έχουν μια αυστηρά καθορισμένη από το χρήστη μορφή και περιέχουν **δεδομένα** (data files). Από εδώ και στο εξής θα αναφερόμαστε κυρίως σε αυτό το είδος αρχείων. Τέλος ένα ιδιαίτερο είδος αρχείων είναι τα αρ-

χεία που περιέχουν πληροφορίες για άλλα αρχεία και τις διευθύνσεις τους και ονομάζονται **κατάλογοι** ή **ευρετήρια** (directories).

1. Χαρακτηριστικά των αρχείων - Ορισμοί

Χαρακτήρας (character) είναι η βασική μονάδα πληροφορίας. Αντιστοιχεί σε ένα byte και αποτελεί το ελάχιστο μέγεθος της απ' ευθείας προσπελάσιμης πληροφορίας.

Πεδίο (field) είναι ένα σύνολο χαρακτήρων που περιγράφει ή αντιπροσωπεύει ένα δεδομένο. Το ονοματεπώνυμο, η διεύθυνση και το ΑΦΜ ενός πελάτη είναι παραδείγματα πεδίων. Κάθε πεδίο χαρακτηρίζεται από το **μήκος** του (το πλήθος των χαρακτήρων) και τον **τύπο** του (αν περιέχει δηλ. αριθμητικούς ή αλφαβητικούς ή άλλους χαρακτήρες). **Υποπεδίο** είναι ένα μέρος ενός πεδίου, το οποίο μπορεί να χρησιμοποιείται εναλλακτικά. Για παράδειγμα το πεδίο ημερομηνία μπορεί να χωρισθεί σε τρία υποπεδία ημέρα, μήνας, έτος.

Λογική εγγραφή (logical record) ή απλά **εγγραφή** είναι ένα σύνολο αλληλοσυσχετιζομένων πληροφοριών που περιγράφουν μια οντότητα. Ολα τα διαθέσιμα πεδία π.χ. για την οντότητα πελάτης συγκροτούν την εγγραφή (record) πελάτη. Η διάταξη των πεδίων μέσα στην εγγραφή αποκαλείται **γραμμογράφηση** (layout).

Εφιστούμε την προσοχή του αναγνώστη στο γεγονός ότι η λέξη εγγραφή είναι συνώνυμη με τη λογική εγγραφή.

Αρχείο (file). Μετά από αυτά ένα αρχείο ορίζεται ως ένα σύνολο ομοειδών εγγραφών. Ετσι αποκαλούμε αρχείο πελατών, το σύνολο των εγγραφών πελατών που διαθέτουμε.

Σημαντική τέλος είναι η έννοια του κλειδιού. **Κλειδί** (key) είναι ένα πεδίο της εγγραφής που προσδιορίζει τη λογική εγγραφή και αποτελεί τη βάση οργάνωσης του αρχείου. Για παράδειγμα σε ένα αρχείο πελατών κλειδί είναι ο κωδικός του πελάτη. Ας σημειωθεί ότι σε ένα αρχείο μπορεί να ορίζονται και περισσότερα κλειδιά (πχ. κωδικός, ονοματεπώνυμο, ΑΦΜ) ή και πιο σύνθετα, που αποτελούνται από συνδυασμό περισσότερων πεδίων.

Η σχέση μεταξύ κλειδιού και εγγραφής στις πιο πολλές περιπτώσεις είναι αμφιμονοσήμαντη, δηλαδή ένα κλειδί αντιστοιχεί σε ακριβώς μία εγγραφή, όπως για

παράδειγμα συμβαίνει με τον κωδικό του πελάτη. Σε άλλες περιπτώσεις όμως ένα κλειδί αντιστοιχεί σε περισσότερες από μια εγγραφές (**διπλά κλειδιά** - doublicate keys). Π.χ. αν σε ένα αρχείο τιμολογίων ορισθεί ως κλειδί η ημερομηνία, τότε όλες οι εγγραφές που έγιναν την ίδια ημέρα, θα έχουν το ίδιο κλειδί. Ακόμη μια εγγραφή μπορεί να καθορίζεται από περισσότερα κλειδιά, που το ένα λέγεται **πρωτεύον** (primary key) και το ή τα άλλα **δευτερεύοντα** (secondary ή alternate keys). Για παράδειγμα σε ένα αρχείο πελατών ο κωδικός πελάτη είναι το πρωτεύον κλειδί, ενώ το επώνυμο του πελάτη θα είναι δευτερεύον.

2. Είδη αρχείων

Ανάλογα με το περιεχόμενό τους τα αρχεία που χρησιμοποιούνται για αποθήκευση και επεξεργασία δεδομένων κατατάσσονται στις εξής κατηγορίες:

1. **Κύρια αρχεία** (master files). Είναι τα βασικά αρχεία μιας εφαρμογής. Τα πεδία τους περιέχουν είτε περιγραφικές πληροφορίες (π.χ. κωδικός, όνομα, χρώμα, βάρος προϊόντος) είτε επεξεργασμένες πληροφορίες (διαθέσιμη ποσότητα, τιμή προϊόντος). Παραδείγματα κύριων αρχείων είναι: αρχείο πελατών, αρχείο προϊόντων, αρχείο υπαλλήλων κ.λπ.

2. **Αρχεία κινήσεων** ή **μεταβολών** ή **δοσοληψιών** (transaction files). Στο κύριο αρχείο μιας εφαρμογής χρειάζεται να γίνονται προσθήκες, διαγραφές ή τροποποιήσεις. Οι μεταβολές που αφορούν ένα κύριο αρχείο καταγράφονται σε ένα αρχείο μεταβολών. Κατά διαστήματα το κύριο αρχείο ενημερώνεται αφού ληφθούν υπόψη τα περιεχόμενα του αρχείου μεταβολών.

3. **Αρχεία δεικτών** (index files). Τα αρχεία αυτά χρησιμοποιούνται σαν ένας κατάλογος ή πίνακας περιεχομένων και παρέχουν μέσω των αντίστοιχων κλειδιών-δεικτών τις θέσεις των διαφόρων εγγραφών σε άλλα αρχεία.

4. **Βοηθητικά αρχεία** (auxiliary files). Περιέχουν ειδικές πληροφορίες που χρησιμοποιούνται σε συνδυασμό με άλλα αρχεία π.χ. αρχεία πινάκων, παραμέτρων, σημειώσεων κ.λπ.

5. **Προσωρινά** (work) και **ενδιάμεσα** (intemediary). Τα αρχεία αυτά χρησιμοποιούνται στην επεξεργασία των πληροφοριών των υπόλοιπων αρχείων για την αποθήκευση προσωρινών αποτελεσμάτων.

6. **Ιστορικά αρχεία** (historical files). Περιέχουν στοιχεία προηγούμενων περιόδων.

7. **Εφεδρικά αρχεία** (backup files). Πρόκειται για αρχεία αντίγραφα άλλων αρχείων, που συνήθως αποθηκεύονται σε λιγότερο εύκολα προσπελάσιμη μορφή. Χρησιμοποιούνται στην περίπτωση που θα συμβεί κάτι στο κανονικό αρχείο. Κατά μια έννοια αποτελούν τα πλέον ανεκτίμητα αρχεία.

3. Επεξεργασία αρχείων

Οι βασικότερες λειτουργίες που μπορούν να εκτελεστούν επί των αρχείων είναι οι εξής:

- **Ανάκτηση** (retrieval) πληροφοριών από αρχείο. Με τον όρο αυτό αναφερόμαστε στη λήψη του περιεχομένου μιας εγγραφής χωρίς να γίνει καμμιά μεταβολή επί του αρχείου. Για να πραγματοποιηθεί η λειτουργία αυτή, πρώτα γίνεται ο εντοπισμός της εγγραφής με βάση συνήθως κάποιο κλειδί της και στη συνέχεια ακολουθεί η ανάγνωση της εγγραφής.

Συχνά η λειτουργία της ανάκτησης αποκαλείται **προβολή** στα προγράμματα εφαρμογών, γιατί αφού βρεθεί η εγγραφή, το περιεχόμενο της προβάλλεται στην οθόνη.

Σχετική είναι και η λειτουργία της **ερώτησης** (query). Ο όρος αυτός αναφέρεται στην αναζήτηση και ανάκτηση εγγραφών του αρχείου, οι οποίες πληρούν συγκεκριμένες συνθήκες που θέτει ο χρήστης. Τέτοιες ερωτήσεις σε ένα αρχείο υπαλλήλων μπορεί να είναι: "Δώσε τους υπαλλήλους των τμημάτων 1 έως 6, που έχουν πανεπιστημιακή ή τεχνολογική εκπαίδευση, προυπηρεσία μεγαλύτερη των 10 ετών και μισθό μικρότερο των 200.000 δρχ.".

- **Ενημέρωση** (updating). Ενημέρωση ενός αρχείου ονομάζεται η με οποιονδήποτε τρόπο μεταβολή των χαρακτηριστικών ενός αρχείου (πλήθος ή/και περιεχόμενο των εγγραφών του). Ο όρος ενημέρωση καλύπτει τρία διαφορετικά είδη επέμβασης σε ένα αρχείο: προσθήκη ή δημιουργία (addition) νέων εγγραφών, διαγραφή ή ακύρωση (deletion) εγγραφών και μεταβολή ή τροποποίηση (modification) του περιεχομένου των εγγραφών.

Η προσθήκη αναφέρεται στη δημιουργία μιας νέας εγγραφής μέσα σε ένα αρ-

χείο. Με κατάλληλη μέθοδο η νέα εγγραφή τοποθετείται στη σωστή θέση επί του αρχείου με βάση συνήθως την τιμή του κλειδιού.

Στη **διαγραφή** μιας εγγραφής ο χώρος που καταλαμβάνεται απελευθερώνεται και μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας νέας (φυσική διαγραφή) ή τοποθετείται κατάλληλα ένας ενδείκτης διαγραφής (λογική διαγραφή).

Η **τροποποίηση** τέλος, αναφέρεται στη μεταβολή του περιεχομένου ενός ή περισσότερων πεδίων μιας εγγραφής χωρίς να αλλάξει η θέση της εγγραφής μέσα στο αρχείο.

Πρέπει να σημειωθεί ότι η οποιαδήποτε μεταβολή μπορεί να γίνει σε όλα τα πεδία μιας εγγραφής εκτός του κλειδιού, γιατί τότε αλλάζει η σχέση μεταξύ των εγγραφών (στις περισσότερες περιπτώσεις). Μια αλλαγή στο κλειδί μιας εγγραφής συνήθως οδηγεί στην ακύρωση της εγγραφής και στην ταυτόχρονη δημιουργία νέας με τις ίδιες πληροφορίες.

- **Ταξινόμηση** (sorting). Ταξινόμηση ενός αρχείου είναι η λειτουργία κατά την οποία οι εγγραφές του αρχείου διατάσσονται κατά αύξουσα ή φθίνουσα σειρά με βάση την τιμή ενός πεδίου ή συνδυασμού από αυτά.

- **Σύζευξη** (merging). Σύζευξη αρχείων είναι η λειτουργία κατά την οποία δύο ή περισσότερα αρχεία ενώνονται και δημιουργούν ένα επίσης ταξινομημένο.

Οι δύο τελευταίες λειτουργίες επειδή γίνονται πολύ τακτικά και απαιτούν συνήθως πολύ χρόνο, πραγματοποιούνται από έτοιμες ρουτίνες sort-merge που παρέχουν τα διάφορα υπολογιστικά συστήματα.

Υπάρχουν ακόμη οι λειτουργίες της **συνένωσης** ή **προσάρτησης** (append) κατά την οποία δύο αρχεία δημιουργούν ένα νέο, καθώς και η αντίστροφη λειτουργία, από ένα αρχείο να δημιουργηθούν περισσότερα. Επίσης η **αντιγραφή** (copy) ενός αρχείου στο ίδιο ή άλλο μέσο, η **δημιουργία αντιγράφων εφεδρείας** (back-up), η δημιουργία προσωρινών ή ενδιάμεσων αρχείων κ.α.

Η ενημέρωση ενός αρχείου αποτελεί μια από τις πιο συνηθισμένες ενέργειες. Σε όλες τις εφαρμογές υπάρχουν προγράμματα ενημέρωσης των βασικών ή και άλλων αρχείων. Οι λειτουργίες της ενημέρωσης παρουσιάζονται χωριστά από τα προγράμματα σε ένα μενού και ο χρήστης επιλέγει αυτή που επιθυμεί. Μπορούν όμως να υλοποιηθούν και από ένα μόνο πρόγραμμα. Σε κάθε περίπτωση δίνεται το επιθυμητό κλειδί και ακολουθεί η αναζήτηση της εγγραφής με το κλειδί αυτό. Η

συνέχεια εξαρτάται από το αποτέλεσμα της αναζήτησης και την ζητούμενη λειτουργία.

Παρουσιάζουμε στη συνέχεια ένα αλγοριθμικό πλαίσιο για την ενημέρωση ενός αρχείου.

```
Αλγόριθμος Update-File
Άνοιγμα αρχείου
Αρχή
 Εισαγωγή κλειδιού
 Αν κλειδί=0 τότε κλείσιμο αρχείου : Εξοδος
 Αναζήτηση εγγραφής
 Αν η εγγραφή υπάρχει τότε
    Προβολή μάσκας οθόνης
    Προβολή πεδίων εγγραφής
    Προβολή μηνύματος "Ακύρωση-Μεταβολή-Τίποτα"
    Εισαγωγή απάντησης χρήστη
    Επιλογή
        - Περίπτωση "Ακύρωση"
        Ακύρωση εγγραφής
        Πήγαινε στην αρχή
        - Περίπτωση "Μεταβολή"
        Εισαγωγή νέων δεδομένων
        Επανεγγραφή εγγραφής
        Πήγαινε στην αρχή
        - Περίπτωση αλλιώς (τίποτα)
        Πήγαινε στην αρχή
    Τέλος επιλογών
 αλλιώς
    ' Δημιουργία εγγραφής
    Προβολή μάσκας οθόνης
    Εισαγωγή δεδομένων
    Εγγραφή εγγραφής
    Πήγαινε στην αρχή
Τέλος-αν
Τέλος
Τέλος Update-File
```

Ο αλγόριθμος αυτός μπορεί να υλοποιηθεί σε κάθε γλώσσα προγραμματι-

σμού, αλλά προφανώς θα πρέπει να ληφθούν υπόψη οι ιδιομορφίες κάθε μιας. Γίνεται φανερό ότι κάθε "εντολή" του αλγόριθμου παραπέμπει σε μια ρουτίνα, που μπορεί να έχει μεγάλη έκταση ανάλογα με τον αριθμό των πεδίων, το είδος των ελέγχων κλπ. Οπωσδήποτε όμως η λογική του διατηρείται και μπορεί να ακολουθείται σαν πρότυπο στις εφαρμογές. Ας σημειωθεί ακόμη ότι στην περίπτωση που η εγγραφή υπάρχει, πρώτα προβάλεται στην οθόνη και μετά μπορεί να ακολουθήσει η μεταβολή ή η διαγραφή της. Αν ο χρήστης δεν επιθυμεί τίποτα από τα δυο, τότε με την πίεση π.χ. του Enter επανέρχεται στην αρχή (περίπτωση αλλιώς). Ετσι συμπεριλαμβάνεται και η λειτουργία της προβολής, που γίνεται πολύ απλά, αφού έχει προηγηθεί η ανάκτησή της.

Ας σημειωθεί τέλος ότι καθόλου προφανής δεν είναι η εντολή αναζήτηση εγγραφής του αλγόριθμου. Η λειτουργία αυτή που αποσκοπεί στην ανάκτηση της εγγραφής, γίνεται φανερό ότι είναι η πιο κρίσιμη στην όλη υπόθεση της διαχείρισης των αρχείων. Η αναζήτηση της εγγραφής εξαρτάται από την οργάνωση του αρχείου και τη χρησιμοποιούμενη μέθοδο προσπέλασης.

4. Κατηγορίες λογικών εγγραφών

Σύμφωνα με μια από τις κατατάξεις οι λογικές εγγραφές σε σχέση με το μήκος τους χωρίζονται στις εξής κατηγορίες:

- **Λογικές εγγραφές σταθερού μήκους**. Είναι η πιο συνηθισμένη κατηγορία για αρχεία δεδομένων όπως π.χ. πελατών, ειδών αποθήκης, υπαλλήλων κλπ. Τα πεδία των εγγραφών αυτών είναι επίσης καθορισμένου μήκους.

Είναι δυνατόν όμως στο ίδιο αρχείο να υπάρχουν εγγραφές διαφορετικού τύπου, δηλαδή αποτελούμενες από διαφορετικά πεδία. Στην περίπτωση αυτή επικεφαλής της εγγραφής τίθεται ένα πεδίο που καθορίζει τον τύπο της εγγραφής. Το πεδίο αυτό είναι σταθερού και γνωστού μήκους (π.χ. ένα byte). Μετά την ανάγνωση της εγγραφής το πρόγραμμα γνωρίζει τα πεδία ανάλογα με τον τύπο της εγγραφής.

- **Λογικές εγγραφές μεταβλητού μήκους.** Μια εγγραφή θεωρείται μεταβλητού μήκους αν ανήκει σε μια από τις επόμενες περιπτώσεις.

α) Αν περιέχει ένα ή περισσότερα πεδία μεταβλητού μήκους.

Για το χειρισμό τέτοιου είδους αρχείων χρησιμοποιείται η τεχνική της **στίξης**,

κατά την οποία ειδικοί χαρακτήρες χρησιμοποιούνται ως διαχωριστές των μεταβλητού μήκους πεδίων και εγγραφών. Οι συνηθέστερα χρησιμοποιούμενοι χαρακτήρες διαχωρισμού είναι το κενό και το κόμμα για τα πεδία και ο συνδυασμός των χαρακτήρων αλλαγής γραμμής και επιστροφής (line feed-carriage return) για τις εγγραφές. Ακόμη ένας επιπλέον χαρακτήρας τέλους αρχείου (EOF-End Of File) προσδιορίζει το φυσικό τέλος του αρχείου.

Ενα παράδειγμα τέτοιου αρχείου είναι τα αρχεία κειμένου, όπου κάθε γραμμή κειμένου είναι μια εγγραφή που αποτελείται από ένα πεδίο προφανώς διαφορετικού μήκους.

β) Αν περιέχει μεταβλητό αριθμό πεδίων σταθερού μήκους.

Σε αυτά τα αρχεία χρησιμοποιείται το λεγόμενο **πεδίο μήκους**, στο οποίο περιέχεται ο αριθμός των πεδίων σταθερού μήκους που ακολουθούν. Το πεδίο μήκους είναι σταθερού και γνωστού μήκους (π.χ. δύο ψηφία) και τίθεται στην αρχή της εγγραφής. Διαβάζοντας αυτό γνωρίζουμε πόσα ακόμα πεδία πρέπει να αναγνωσθούν για να συμπληρωθεί η εγγραφή.

Δεν είναι στην ουσία διαφορετική η περίπτωση του μικτού σχήματος, κατά το οποίο μπορεί να υπάρχει ένα σταθερό τμήμα, ακολουθούμενο από ένα μεταβλητό τμήμα που αποτελείται από μεταβλητό αριθμό εγγραφών σταθερού μήκους. Στην περίπτωση αυτή το πεδίο μήκους είναι ένα πεδίο του σταθερού τμήματος. Εδώ τα υπόλοιπα πεδία αποκαλούνται **ρυμουλκούμενα** (trailers).

Ας σημειωθεί ότι το πεδίο μήκους μπορεί να καθορίζει το μήκος ενός πεδίου ή και όλης της εγγραφής. Ετσι μπορούν να δημιουργηθούν εγγραφές, οι οποίες περιέχουν μεταβλητό αριθμό πεδίων, που το καθένα είναι μεταβλητού μήκους. Ο συνολικός αριθμός των πεδίων καθορίζεται από ένα πεδίο μήκους στην αρχή της εγγραφής, αλλά το μήκος κάθε πεδίου καθορίζεται από ένα άλλο πεδίο μήκους που μπαίνει στην αρχή κάθε πεδίου.

- Λογικές εγγραφές απροσδιορίστου μήκους.

Πρόκειται για εγγραφές των οποίων το μήκος καθώς και η γραμμογράφησή τους καθορίζονται κατά τη στιγμή της επεξεργασίας τους. Στην ουσία πρόκειται για σειρές χαρακτήρων χωρίς διαχωριστές πεδίων ή εγγραφών.

Μερικοί κατασκευαστές τα αποκαλούν και **δυαδικά** (binary) και μπορούν να θεωρηθούν ως εκφυλισμένη περίπτωση κατά την οποία η εγγραφή είναι μήκους 1 byte.

5. Φυσική αποθήκευση των αρχείων

Οι εγγραφές των αρχείων καταλαμβάνουν διάφορες περιοχές του μαγνητικού μέσου που αποθηκεύονται. Στις ταινίες κάθε εγγραφή τίθεται μετά την άλλη στη σειρά. Στους δίσκους οι εγγραφές καταλαμβάνουν θέσεις ενός τομέα, όταν αυτός συμπληρωθεί, παίρνουν θέσεις από τον επόμενο, όταν γεμίσει όο το ίχνος, πηγαίνουμε στο επόμενο κ.ο.κ.

Πρέπει να σημειωθεί ότι ειδικά στους δίσκους η κατάσταση δεν είναι πάντα όπως περιγράφηκε πριν. Και για την ακρίβεια είναι έτσι μόνο στην αρχή. Οταν αργότερα υπάρχει μια δραστηριότητα επί του δίσκου, δηλαδή διαγράφονται κάποια αρχεία, ενώ δημιουργούνται νέα, ταυτόχρονα μπορεί να προστίθονται καινούργιες εγγραφές σε κάποια από αυτά, τότε η εικόνα αυτή αλλάζει. Ενα αρχείο μπορεί να μη καταλαμβάνει συνεχόμενες περιοχές του δίσκου αλλά διάσπαρτες. Το φαινόμενο αυτό ονομάζεται **κατάτμηση** (fragmentation). Ευτυχώς όμως το λειτουργικό σύστημα φροντίζει, ώστε εμείς να νομίζουμε πάντα το αρχείο μας σαν μια συνεχή παράθεση εγγραφών (σαν έναν πίνακα δηλαδή).

Εδώ ίσως είναι η κατάλληλη θέση για να ορίσουμε τη λεγόμενη **φυσική εγγραφή**. Ανάλογα με τις ανάγκες μας εμείς μπορούμε να καθορίζουμε λογικές εγγραφές διαφόρων μεγεθών. Οταν ζητάμε να διαβάσουμε ή να γράψουμε μια εγγραφή, τότε το σύστημα εκτελεί μια ενέργεια εισόδου-εξόδου και πραγματοποιεί τη μεταφορά των δεδομένων μεταξύ κύριας μνήμης και δίσκου. Ομως για τεχνικούς λόγους που έχουν να κάνουν με τη βελτιστοποίηση της απόδοσης, κατά την ενέργεια αυτή δεν μεταφέρεται η λογική εγγραφή, αλλά μια συγκεκριμένη ποσότητα πληροφοριών που καθορίζεται από το σύστημα. Η ποσότητα αυτή αποτελεί τη φυσική εγγραφή. Μια φυσική εγγραφή μπορεί να είναι για παράδειγμα το περιεχόμενο ενός (ή περισσοτέρων) τομέων του δίσκου.

Τα περιεχόμενα της φυσικής εγγραφής, μετά τη μεταφορά τους από το δίσκο, αποθηκεύονται σε μια περιοχή μνήμης που αφιερώνεται για το σκοπό αυτό και που αποκαλείται **ενδιάμεση μνήμη** (buffer). Προφανώς το σύστημα φροντίζει ώστε μέσα σε αυτή τη φυσική εγγραφή να βρίσκεται και η λογική εγγραφή που μας ενδιαφέρει. Ετσι το πρόγραμμά μας την παραλαμβάνει από την ενδιάμεση μνήμη για τα περαιτέρω.

Το γεγονός αυτό έχει μια σημαντική επίπτωση που πρέπει να γνωρίζει ο χρήστης. Οταν ολοκληρωθεί η επεξεργασία μιας (λογικής) εγγραφής, δεν σημαίνει κατ' ανάγκη ότι το ενημερωμένο περιεχόμενό της έχει μεταφερθεί στο δίσκο. Αυτό

θα γίνει όταν ολοκληρωθεί η επεξεργασία της φυσικής εγγραφής. *Γι' αυτό δεν πρέπει να αποσύρεται μια δισκέτα ή να κλείνει ο υπολογιστής, αν το πρόγραμμα εφαρμογής δεν έχει τερματιστεί κανονικά.* Φυσικά ο ίδιος κίνδυνος απώλειας κάποιων δεδομένων υπάρχει και όταν σημειωθεί κάποια πτώση τάσης.

Σχετική είναι και η έννοια της **ομαδοποίησης** (blocking) εγγραφών που χρησιμοποιείται για την καλύτερη αποτελεσματικότητα των μαγνητικών μέσων. Η ομαδοποίηση συνίσταται στη δημιουργία φυσικών εγγραφών ή **ομάδων** (blocks) που αποτελούνται από περισσότερες λογικές εγγραφές. Το πλήθος των λογικών εγγραφών ανά ομάδα καθορίζεται είτε από το σύστημα είτε συνηθέστερα από το χρήστη και ονομάζεται **συντελεστής ομαδοποίησης** (blocking factor).

Αν το αρχείο θεωρηθεί ότι βρίσκεται σε ταινία, τότε μεταξύ των ομάδων υπάρχει και ένα κενό που λέγεται **διάκενο μεταξύ εγγραφών** (inter record gap-IRG). Το μέγεθος του διακένου εξαρτάται από τα τεχνικά χαρακτηριστικά της μονάδας μαγνητικής ταινίας. Τυπικό μέγεθος είναι 3/4 της ίντσας.

6. Οργάνωση και προσπέλαση αρχείων

Ο τρόπος με τον οποίο οργανώνονται και αποθηκεύονται τα δεδομένα στα φυσικά μέσα αποθήκευσης (δισκέτες, ταινίες, δίσκοι) είναι καθοριστικός για τη μορφή της επεξεργασίας που θα ακολουθήσει, με στόχο πάντα τη γρήγορη λήψη απαντήσεων.

Οι εγγραφές των αρχείων οργανώνονται μεταξύ τους με σκοπό να διευκολύνουν τον εντοπισμό και την ανάκτησή τους και γενικά τη συντήρηση του αρχείου. Για την επιλογή της μεθόδου οργάνωσης λαμβάνονται υπόψη μερικά από τα βασικά χαρακτηριστικά του αρχείου. Τα κυριότερα από αυτά είναι:

- είδος επεξεργασίας (διαλογική - interactive, κατά δοσοληψία - transaction processing, κατά δέσμες - batch)
- μέγεθος και προβλεπόμενος ρυθμός αύξησης του αρχείου
- συχνότητα των εργασιών ενημέρωσης (εισαγωγή, διαγραφή, τροποποίηση)
- ποσοστό χρησιμοποιούμενων εγγραφών από τα προγράμματα
- κατηγορίες εγγραφών που χρησιμοποιούνται τακτικότερα.

Οι συνηθέστερες οργανώσεις αρχείων -που αποκαλούνται και κλασικές ή παραδοσιακές σε αντιδιαστολή με την οργάνωση των Βάσεων Δεδομένων- είναι οι εξής τρεις: σειριακή, σειριακή με δείκτες και απευθείας οργάνωση.

7. Μέθοδοι Προσπέλασης

Η θεώρηση που έχουν οι χρήστες για τη μορφή των δεδομένων κατά την επεξεργασία τους δεν συμφωνεί πάντοτε με το φυσικό τρόπο οργάνωσης και αποθήκευσης αυτών στη βοηθητική μνήμη.

Οι τεχνικές και ειδικές ρουτίνες που διαχειρίζονται την αποθήκευση και την ανάκτηση των δεδομένων, ονομάζονται **μέθοδοι προσπέλασης** (access methods) και αναλαμβάνουν όλες τις λειτουργίες εισόδου/εξόδου και τη μεταφορά μεταξύ κύριας μνήμης και περιφερειακών συσκευών.

Ο όρος προσπέλαση στην γενικότερη του έννοια δηλώνει τον εντοπισμό μιας εγγραφής με στόχο την ανάκτηση, διαγραφή ή τροποποίησή της. Ο χρόνος προσπέλασης αναφέρεται στο απαιτούμενο χρονικό διάστημα.

Είδη προσπέλασης υπάρχουν δύο, η **σειριακή** ή **διαδοχική** (sequential) και η **άμεση** (random). Στη σειριακή προσπέλαση ο χρόνος προσπέλασης εξαρτάται από τη θέση της εγγραφής, ενώ στην άμεση προσπέλαση είναι ανεξάρτητος από τη θέση της εγγραφής.

Οι δυνατότητες προσπέλασης στις διάφορες βοηθητικές μνήμες διαφέρουν. Έτσι στις μαγνητικές ταινίες μπορούμε να έχουμε μόνο σειριακή προσπέλαση, αφού για να βρεθεί μια εγγραφή, πρέπει να γίνει η ανάγνωση όλων όσων προηγούνται.

Αντίθετα στους δίσκους έχουμε δυνατότητα για σειριακή και άμεση προσπέλαση. Οι εγγραφές ενός αρχείου σε ένα δίσκο καταλαμβάνουν διαδοχικούς τομείς, ίχνη και κυλίνδρους. Μπορούμε να διαβάσουμε με τη σειρά όλους τους τομείς ενός ίχνους, στη συνέχεια όλους τους τομείς του επόμενου ίχνους κοκ μέχρι εξάντλησης του κυλίνδρου και να συνεχίσουμε στον επόμενο κύλινδρο κλπ. Επιτυγχάνουμε έτσι τη σειριακή προσπέλαση.

Είναι δυνατό όμως να γνωρίζουμε τη θέση κάθε εγγραφής επί του δίσκου (φυσικές διευθύνσεις), δηλαδή σε ποιο κύλινδρο, ίχνος και τομέα βρίσκεται. Έτσι μπορούμε να διαβάσουμε αμέσως μια εγγραφή, χωρίς να κάνουμε το ίδιο για όσες

προηγούνται. Προς τούτο αρκεί να διατάξουμε τις κεφαλές του δίσκου να κινηθούν στο δεδομένο κύλινδο, να επιλέξουμε το επιθυμητό ίχνος και να διαβάσουμε το συγκεκριμένο τομέα που περιέχει τη ζητούμενη πληροφορία. Αυτό σημαίνει ότι ο χρόνος προσπέλασης στο δίσκο είναι άθροισμα των επόμενων δύο χρόνων:

α) Τ1 ο χρόνος εντοπισμού κυλίνδρου (seek time), που απαιτείται για τη μετακίνηση των μαγνητικών κεφαλών στο συγκεκριμένο κύλινδρο και

β) Τ2 ο χρόνος αναμονής μέχρι ο υπόψη τομέας περάσει κάτω από την κεφαλή.

Ετσι ο συνολικός χρόνος είναι Τ1+Τ2. Από αυτούς ο Τ2 είναι κατά μέσον όρο ίσος με το χρόνο μισής περιστροφής του δίσκου, ενώ ο Τ1 δεν μπορεί να καθοριστεί, γιατί δεν είναι γνωστή η θέση εκκίνησης των μαγνητικών κεφαλών. Ετσι ενώ φαινομενικά ο Τ1 εξαρτάται από τη θέση της εγγραφής, κάτι τέτοιο δεν συμβαίνει γιατί η προηγούμενη θέση των κεφαλών ήταν τυχαία, με την έννοια ότι δεν είναι γνωστή. Για το λόγο αυτό η άμεση προσπέλαση αποκαλείται και τυχαία μεταφράζοντας τον όρο random. Ομως σε καμία περίπτωση δεν υπονοείται ότι η επιλογή της θέσης γίνεται "στην τύχη". Πιστεύουμε ότι ο όρος άμεση προσπέλαση είναι προτιμότερος.

Στην επόμενη παράγραφο παρουσιάζονται η σειριακή οργάνωση και παραδείγματα σχετικών προγραμμάτων.

8. Σειριακή οργάνωση (sequential organization).

Σε αυτή τη μέθοδο οργάνωσης οι εγγραφές είναι οργανωμένες με τέτοιον τρόπο, ώστε για να διαβαστεί η ν-οστή εγγραφή, πρέπει να διαβαστούν οι προηγούμενες ν-1 εγγραφές. Δηλαδή οι φυσικές θέσεις στις οποίες αποθηκεύονται οι λογικές εγγραφές ακολουθούν κάποια αλληλουχία ή σειρά σε σχέση με τις αντίστοιχες φυσικές θέσεις των προηγούμενων ή επόμενων λογικών εγγραφών. Η σειρά αυτή δημιουργείται με βάση κάποιο χαρακτηριστικό που όταν είναι ένα πεδίο της εγγραφής, ονομάζεται κλειδί (key) και είναι ο κώδικας ταξινόμησης. Η προσπέλαση, ανάγνωση και ενημέρωση γίνονται με τη σειρά αυτή.

Κατά συνέπεια η προσπέλαση μιας εγγραφής συνήθως δεν είναι γρήγορη, γιατί απαιτούνται κατά μέσον όρο ν/2 άχρηστες προσπελάσεις πριν εντοπισθεί η ζητούμενη, ενώ η προσθήκη και η διαγραφή εγγραφών γίνεται μόνο με τη δημιουργία νέου αρχείου.

Είναι όμως δυνατό να υπάρχουν και μη ταξινομημένα σειριακά αρχεία. Στα αρχεία αυτά οι εγγραφές καταχωρούνται με σειρά άφιξης. Πρόκειται συνήθως για αρχεία μεταβολών ή προσωρινά αρχεία, τα οποία τις πιο πολλές φορές ταξινομούνται προκειμένου να είναι χρήσιμα (για την ενημέρωση π.χ. άλλων αρχείων). Μια ειδική περίπτωση τέτοιων αρχείων αποτελούν τα **αρχεία κειμένου** (text files).

Η σειριακή μέθοδος οργάνωσης συνιστάται σε περιπτώσεις όπου οι απαιτήσεις επιβάλλουν τη "σάρωση" και ανάκτηση για επεξεργασία του μεγαλύτερου πλήθους των εγγραφών ενός αρχείου όπως π.χ. διάβασμα ολοκλήρου του αρχείου υπαλλήλων για τον υπολογισμό της μισθοδοσίας τους. Η σειριακή οργάνωση είναι η μόνη που επιτρέπεται για αρχεία σε μαγνητικές ταινίες (και γι' αυτό ήταν πολύ δημοφιλής μέθοδος την εποχή των διάτρητων δελτίων και της επεξεργασίας με ταινίες), αλλά χρήσιμη και σε μαγνητικούς δίσκους και δισκέτες. Στους δίσκους οι εγγραφές αποθηκεύονται η μια μετά την άλλη σε διαδοχικά αυξανόμενες φυσικές διευθύνσεις.

Γενικά στις συσκευές άμεσης προσπέλασης η σειριακή οργάνωση χρησιμοποιείται ευρύτατα για τη δημιουργία αρχείων πινάκων, ευρετηρίων, λεξικών ή ταξινομημένων κινήσεων παρά για τα κύρια αρχεία των εφαρμογών.

8.1 Επεξεργασία σειριακών αρχείων

Στα σειριακά αρχεία η άμεση ανάκτηση πληροφοριών είναι πρακτικά αδύνατη. Μπορεί να είναι αποδοτική μόνο σε πολύ μικρού μεγέθους αρχεία. Συχνά στις εφαρμογές τέτοια μικρά αρχεία πινάκων ή παραμέτρων "ανεβαίνουν" με την έναρξη της διαδικασίας στην κύρια μνήμη και η αναζήτηση γίνεται από εκεί με σημαντικά μεγαλύτερη ταχύτητα.

Στην περίπτωση που σε μια εφαρμογή υπάρχει κύριο σειριακό αρχείο, η ενημέρωσή του γίνεται με την επόμενη διαδικασία.

Ολες οι μεταβολές που πρόκειται να συμβούν στο βασικό αρχείο, συγκεντρώνονται σε ένα σειριακό αρχείο μεταβολών, το οποίο ταξινομείται με τον ίδιο τρόπο που είναι ταξινομημένο το βασικό αρχείο. Ακολουθεί ένας έλεγχος ορθότητας (validation) των μεταβολών αυτών. Κάθε εγγραφή του αρχείου μεταβολών, εκτός από τα υπόλοιπα στοιχεία, διαθέτει και έναν κωδικό λειτουργίας ΚΛ, ο οποίος αν είναι:

- 1, σημαίνει ότι η εγγραφή αυτή είναι καινούργια και πρέπει να ενταχθεί στο κύριο αρχείο (δημιουργία).
- 2, σημαίνει ότι η εγγραφή αυτή δεν είναι καινούργια, αλλά θα μεταβάλλει ένα ή περισσότερα πεδία της αντίστοιχης εγγραφής του βασικού αρχείου (τροποποίηση)
- 3, σημαίνει ότι η εγγραφή αυτή θα διαγράψει την αντίστοιχη στο βασικό αρχείο.

Η συνέχεια είναι η ακόλουθη. Τα δύο αρχεία διαβάζονται παράλληλα, ελέγχεται η **ταύτιση** των κλειδιών (matching) και παράγεται το ενημερωμένο βασικό αρχείο ως εξής:

Κάθε εγγραφή του βασικού αρχείου για την οποία δεν υπάρχει αντίστοιχη εγγραφή στο αρχείο μεταβολών, μεταφέρεται αυτούσια στο αρχείο εξόδου.

Κάθε εγγραφή που υπάρχει στο αρχείο μεταβολών, ενώ δεν υπάρχει στο βασικό αρχείο (ΚΛ=1), μεταφέρεται στη σωστή θέση στο αρχείο εξόδου.

Οι εγγραφές του αρχείου μεταβολών με ΚΛ=2 μεταφέρονται στο αρχείο εξόδου αντικαθιστώντας τις αντίστοιχες του βασικού αρχείου, ενώ δεν αντιγράφονται οι ακυρωμένες εγγραφές (ΚΛ=3).

Ανάλογη είναι η διαδικασία όταν το αρχείο μεταβολών περιέχει στοιχεία δοσοληψιών, που πρέπει να μεταβάλουν μερικά πεδία του βασικού αρχείου. Για παράδειγμα σε ένα αρχείο πελατών ποσά που προέρχονται από πωλήσεις ή εισπράξεις πρέπει να ενημερώσουν τη χρέωση και πίστωση των πελατών.

Αλλες συνήθεις λειτουργίες σε σειριακά αρχεία είναι η έκδοση καταστάσεων, η συγχώνευση (merging) δύο ή περισσότερων αρχείων για τη δημιουργία ενός νέου κλπ.

8.2 Υλοποίηση των σειριακών αρχείων στις γλώσσες προγραμματισμού

Turbo Pascal

Ο **τύπος αρχείων κειμένου (text files)** είναι ένας τύπος αρχείων που υποστηρίζεται από την Turbo Pascal. Ένα αρχείο κειμένου αποτελείται από σειρές χαρα-

κτήρων που χωρίζονται σε γραμμές. Κάθε γραμμή τερματίζεται με το σύμβολο τέλος-γραμμής (end of line, **eoln**) που στην πραγματικότητα είναι οι χαρακτήρες **CR** και **LF**. Το τέλος του αρχείου σημειώνεται με το σύμβολο τέλος-αρχείου (end-of-file, **eof**).

Υπάρχουν δύο προκαθορισμένες μεταβλητές αρχείου οι **Input** και **Output** οι οποίες αναφέρονται στην είσοδο από το πληκτρολόγιο και την έξοδο στην οθόνη. Εκτός από αυτά τα δύο προκαθορισμένα αρχεία μπορούν σε κάθε πρόγραμμα να χρησιμοποιηθούν και άλλα αρχεία κειμένου.

Οι βασικές αρχές οι οποίες καθορίζουν τη χρήση των αρχείων κειμένου στη Turbo Pascal, εκτός των αρχείων Input, Output, είναι οι παρακάτω:

- **Επικεφαλίδα προγράμματος:** Μία καλή πρακτική είναι όλα τα αρχεία, τα οποία πρόκειται να χρησιμοποιηθούν σε ένα πρόγραμμα να αναφέρονται στη λίστα αρχείων στην επικεφαλίδα του προγράμματος. Στη λίστα αυτή αναφέρονται και τα προκαθορισμένα αρχεία Input, Output.

  Program Όνομα_προγράμματος (λίστα Μεταβλητών_Αρχείου)

- **Δηλώσεις**: Όλα τα αρχεία που χρησιμοποιούνται, οι μεταβλητές αρχείου, πρέπει να ορίζονται στο τμήμα δηλώσεων του προγράμματος. Για τα αρχεία τύπου κειμένου χρησιμοποιείται ο τύπος μεταβλητής text.

  var

  Μεταβλητή_Αρχείου:text

- **Συσχέτιση μεταβλητών με τα πραγματικά αρχεία:** Κάθε μεταβλητή αρχείου προτού χρησιμοποιηθεί πρέπει να συσχετισθεί με ένα φυσικό αρχείο που βρίσκεται στο δίσκο. Η συσχέτιση γίνεται με τη χρήση της προκαθορισμένης συνάρτησης της Turbo Pascal **assign**.

  Assign (Μεταβλητή_Αρχείου, όνομα-αρχείου)

- **Άνοιγμα αρχείων για είσοδο δεδομένων:** Για να διαβαστούν δεδομένα από οποιοδήποτε αρχείο, αυτό πρέπει να ανοιχθεί για είσοδο με την προκαθορισμένη συνάρτηση reset. Η εντολή **reset** τοποθετεί το δείκτη του αρχείου στην αρχή του αρχείου.

  Reset (Μεταβλητή_Αρχείου)

- **Άνοιγμα αρχείων για έξοδο δεδομένων:** Όλα τα αρχεία τα οποία χρησιμοποιούνται για έξοδο δεδομένων πρέπει να ανοίξουν με τις διαδικασίες **rewrite** ή **append**. Και οι δύο ανοίγουν ένα αρχείο για γράψιμο αλλά η rewrite αδειάζει το αρχείο και τοποθετεί το δείκτη στην αρχή ενώ η append τοποθετεί τον δείκτη του αρχείου στο τέλος επιτρέποντας έτσι την πρόσθεση νέων εγγραφών.

  Rewrite (Μεταβλητή_Αρχείου)

  Append(Μεταβλητή_Αρχείου)

- **Είσοδος:** Τα δεδομένα διαβάζονται από το αρχείο με χρήση των **read, readln**.

  Read(Μεταβλητή_Αρχείου, Λίστα μεταβλητών)

- **Έξοδος:** Τα δεδομένα καταχωρούνται στο αρχείο με τη χρήση των **write, writeln**.

  Write(Μεταβλητή_Αρχείου, Λίστα Μεταβλητών)

- **Κλείσιμο αρχείων:** όλα τα αρχεία τα οποία χρησιμοποιούνται σε ένα πρόγραμμα πρέπει να κλείνουν με την εντολή **close**.

  Close(Μεταβλητή_Αρχείου)

Για τον έλεγχο του τέλους γραμμής και του τέλους του αρχείου χρησιμοποιούνται δύο προκαθορισμένες συναρτήσεις οι **eoln** και **eof**.

Η συνάρτηση **eoln** ελέγχει την ύπαρξη του χαρακτήρα τέλους-γραμμής και τότε επιστρέφει την τιμή True. Αντίστοιχα η συνάρτηση **eof** ελέγχει το τέλος του αρχείου.

Eoln (Μεταβλητή_Αρχείου)

Eof (Μεταβλητή_Αρχείου)

QuickBASIC

Τα σειριακά αρχεία στη γλώσσα αυτή έχουν μεταβλητού μήκους εγγραφές με διαχωριστές μεταξύ των πεδίων. Διαχωριστές εγγραφών είναι το ζεύγος των χαρακτήρων CR και LF, ενώ το τέλος αρχείου προσδιορίζεται από το χαρακτήρα EOF (Ascii 26).

Με την εντολή OPEN ανοίγει ένα σειριακό αρχείο ως αρχείο εισόδου ή εξόδου ή για προσάρτηση. Η σύνταξη της εντολής είναι:

```
OPEN ονομα_αρχείου [FOR τυπος] AS [#]αριθμος_αρχειου
```

όπου τυπος μπορεί να είναι INPUT, OUTPUT ή APPEND αντίστοιχα.

Εντολές εγγραφής είναι οι PRINT και WRITE με σύνταξη:

```
PRINT #αριθμος_αρχειου, λιστα_μεταβλητών
```

Η ένταξη των διαχωριστικών χαρακτήρων γίνεται με ευθύνη του προγραμματιστή. Η εντολή WRITE έχει παρόμοια σύνταξη, αλλά θέτει αυτόματα κόμματα ως διαχωριστικές.

Η εντολή ανάγνωσης είναι η INPUT με σύνταξη:

```
INPUT #αριθμος_αρχείου, λιστα_μεταβλητών
```

Η συνάρτηση EOF(αριθμος_αρχείου) εξεταζόμενη πριν την INPUT δίδει τιμή αληθής, αν έχει επέλθει το τέλος αρχείου.

8.3 Παραδείγματα

1. Το επόμενο πρόγραμμα δέχεται από το πληκτρολόγιο τιμές πεδίων ενός απλοϊκού αρχείου αποθήκης και τις εγγράφει στο αρχείο.

Turbo Pascal

```
program files1(input,output);
uses crt;
var
   cod:integer;
   per:string[20];
   tm:integer;
   q:real;
   stock:Text;
begin
clrscr;
assign(stock,'stock.dat');
rewrite(stock);
write('Κωδικός:');
```

```
readln(cod);
while cod>0 do
  begin
    write('Περιγραφή :');
    readln(per);
    write('Τιμή Μονάδας :');
    readln(tm);
    write('Ποσότητα =');
    readln(q);
    writeln(stock,cod:3,per:20,tm:4,q:6);
    clrscr;
    write('Κωδικός :');
    readln(cod);
  end;
close(stock);
end.
```

Quick Basic

```
' Δημιουργία σειριακού αρχείου
OPEN "STOCK.DAT" FOR OUTPUT AS #1
DO
    INPUT "Κωδικός =",cod$
    IF cod$="" THEN EXIT DO
    INPUT "Περιγραφή =",per$
    INPUT "Τιμή μονάδος =",tm
    INPUT "Ποσότητα  =",q!
    PRINT#1,cod$;",";per$;",";tm;",";q!
LOOP
CLOSE
END
```

2. Το επόμενο πρόγραμμα πραγματοποιεί μια απλή εκτύπωση (dump) του προηγούμενου αρχείου.

Turbo Pascal

```
program files2(input,output);
uses crt;
var
```

```
   xx:char;
   grami:string[80];
   stock:Text;
begin
clrscr;
assign(stock,'stock.dat');
reset(stock);
while not eof(stock) do
  begin
    readln(stock,grami);
    writeln(grami);
  end;
close(stock);
write('Πίεσε οποιοδήποτε πλήκτρο για συνέχεια...');
xx:=readkey;
end.
```

Quick Basic

```
' Εκτύπωση σειριακού αρχείου
OPEN "STOCK.DAT" FOR INPUT AS #1
CLS
WHILE NOT EOF(1)
    LINE INPUT#1,z$
    PRINT z$
WEND
CLOSE
END
```

Ας σημειωθεί ότι το ίδιο πρόγραμμα εκτελείται για κάθε σειριακό αρχείο μια και δεν είναι απαραίτητο να είναι γνωστή η γραμμογράφησή του (οι εντολές readln και LINE INPUT διαβάζουν γραμμές δηλ. ολόκληρη την εγγραφή).

3. Το επόμενο πρόγραμμα πραγματοποιεί μια μορφοποιημένη εκτύπωση του αρχείου με τίτλους και αριθμό σελίδας στην οθόνη ή τον εκτυπωτή.

Turbo Pascal

```
program files3(input,output,lst,stock);
uses crt;
const
  max_crt=22;
  max_prn=63;

var
   xx:char;
   cod,lines,choice,max_lines,pages:integer;
   per:string[20];
   tm:integer;
   q:real;
   stock,lst:Text;

procedure PRINT_TITLES(var lst:text;var lines,pages:integer);
begin
   writeln(lst,'ΚΩΔΙΚΟΣ     ΠΕΡΙΓΡΑΦΗ     ΤΙΜΗ     ΠΟΣΟΤΗΤΑ');
   lines:=0;
   pages:=pages+1
end;

begin
  clrscr;
  assign(stock,'stock.dat');
  reset(stock);
  clrscr;
  pages:=0;
  writeln ('ΕΚΤΥΠΩΣΗ ΑΡΧΕΙΟΥ');
  writeln;
  writeln ('   1.ΟΘΟΝΗ');
  writeln ('   2.ΕΚΤΥΠΩΤΗΣ');
  repeat
   gotoxy(5,5); write('Επιλογή:');readln(choice);
  until (choice=1) or (choice=2);
  if choice=1 then
  begin
     assign (lst,'con');
```

```
    max_lines:=max_crt
  end
  else
  begin
      assign(lst,'LPT1');
      max_lines:=max_prn;
   end;
  rewrite(lst);
  print_titles(lst,lines,pages);
  while not eof(stock) do
  begin
    readln(stock,cod,per,tm,q);
    if linesmax_lines then
    begin
      writeln (lst,'ΣΕΛΙΔΑ: ',pages:2);
      if choice=1 then
      begin
        write('Πίεσε ένα πλήκτρο για συνέχεια..');
        xx:=readkey;
        clrscr; {Νέα οθόνη}
      end
      else
          writeln(lst,chr(12)); {Αλλαγή σελίδας}
      print_titles(lst,lines,pages);
    end;
    writeln(lst,cod:5,' ',per:20,' ',tm:4,' ',q:10:2);
    lines:=lines+1;
  end;
   writeln (lst,'ΣΕΛΙΔΑ: ',pages:2);
   write('Τέλος Δεδομένων. Πίεσε ένα πλήκτρο ..');
   xx:=readkey;
 close(stock); close(lst);
end.
```

Quick Basic

```
' Εκτύπωση σειριακού αρχείου
k$ = "": CLS
page = 1: lines = 100
PRINT "ΕΚΤΥΠΩΣΗ ΑΡΧΕΙΟΥ"
PRINT
PRINT "1. ΟΘΟΝΗ"
PRINT "2. ΕΚΤΥΠΩΤΗΣ"
LOCATE 7, 1: PRINT "Επιλογή : ."
DO WHILE VAL(k$) 1 OR VAL(k$)  2
    k$ = INKEY$: LOCATE 7, 11, 1: PRINT k$;
LOOP
k = VAL(k$)
IF k = 1 THEN
    OPEN "O", #3, "SCRN:": maxl = 22
ELSE
    OPEN "O", #3, "LPT1:": maxl = 66
END IF
OPEN "I", #1, "STOCK.DAT"
' Printing
WHILE NOT EOF(1)
    GOSUB ReadNextRecord
    GOSUB PrintOneLine
WEND
CLOSE : END
ReadNextRecord:
    INPUT #1, cod$, per$, tm, q!
    RETURN
PrintOneLine:
    IF lines  maxl THEN GOSUB PrintTitle
    PRINT #3, cod$; " "; per$; TAB(30);
    PRINT #3, USING "#####"; tm;
    PRINT #3, USING "#######.##"; q!
    lines = lines + 1
    RETURN
PrintTitle:
    IF k = 1 THEN
       GOSUB Press: CLS
```

```
    ELSE
       PRINT #3, CHR$(12);
    END IF
    PRINT #3, "ΚΩΔ. Π Ε Ρ Ι Γ Ρ Α Φ Η      ΤΙΜΗ ΜΟΝ. ΠΟΣΟΤΗΤΑ";
    PRINT #3, TAB(70); "ΣΕΛ:"; page
    PRINT #3, STRING$(79, "-")
    page = page + 1: lines = 3
    RETURN
Press:
    LOCATE 24, 1
    PRINT "Πίεσε οποιοδήποτε πλήκτρο για συνέχεια ...";
    w$ = INPUT$(1): RETURN
```

Στο πρόγραμμα αυτό η εντολή OPEN χρησιμοποιείται με τη συνεπτυγμένη εναλλακτική μορφή της. Ως αρχείο #3 ανοίγει η οθόνη ή ο εκτυπωτής ανάλογα με την επιλογή του χρήστη. Αν το πρόγραμμα χρησιμοποιηθεί μόνο για εμφάνιση στην οθόνη, τότε αυτό το βήμα δεν είναι απαραίτητο και οι εντολές PRINT #3 θα αντικατασταθούν με απλές PRINT. Αντίστοιχα αν πρόκειται για εκτύπωση, τότε θα χρησιμοποιηθούν εντολές LPRINT.

Το πρόγραμμα χρησιμοποιεί ανεξάρτητες ρουτίνες για την ανάγνωση μιας εγγραφής και για την εκτύπωση κάθε γραμμής και των τίτλων. Η μεταβλητή lines έχει ως αρχική τιμή 100, ώστε να προκληθεί αμέσως εκτύπωση των τίτλων. Ο αλγόριθμος του προγράμματος είναι υποδειγματικός για τις περισσότερες περιπτώσεις εκτυπώσεων και συνιστάται η υιοθέτησή του από τον αναγνώστη.

4. "Φόρτωμα" τιμών σε πίνακα

Turbo Pascal

```
program files4(input, output, myfile);
uses crt;

Var
  Pinakas:array[1..50,1..50] of integer;
  i,j,M,N:integer;
  myfile:text;
begin
   clrscr;
   assign (myfile,'ARXEIO.DAT');
   reset (myfile);
```

```
   {ΜΕΤΑΦΟΡΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΑΡΧΕΙΟΥ ΣΤΟΝ ΠΙΝΑΚΑ}
   readln (myfile, M, N);
   for i:=1 to M do
       begin
       for j:=1 to N do
         read(myfile,pinakas[i,j]);
   readln (myfile);
   end;

   {ΕΚΤΥΠΩΣΗ ΤΟΥ ΠΙΝΑΚΑ}
     for i:=1 to M do
     begin
     for j:=1 to N do
      write (pinakas[i,j],' ' );
      writeln
     end;
   readln;
 end.
```

Quick Basic

```
DIM a(20, 20)
OPEN "I", #1, "data.dat"
INPUT #1, m, n
FOR i = 1 TO m
    FOR j = 1 TO n
        INPUT #1, a(i, j)
    NEXT j
NEXT i
CLOSE : END
```

Με το πρόγραμμα αυτό τα δεδομένα του αρχείου μεταφέρονται στον πίνακα a.

Το πρόγραμμα αυτό (ή άλλο ανάλογο) μπορεί να βοηθήσει την εκπαιδευτική διαδικασία στο εργαστήριο. Ο καθηγητής μπορεί να πρετοιμάσει δεδομένα σε ένα αρχείο (με πρόγραμμα εισαγωγής ή και με έναν editor), τα οποία να παραδόσει στους μαθητές. Οι τελευταίοι εντάσσουν στο πρόγραμμά τους το προηγούμενο

πρόγραμμα και μπορούν να συνεχίσουν με την επεξεργασία του πίνακα. Ετσι δεν δαπανάται χρόνος για την εισαγωγή των δεδομένων σε κάθε επανεκτέλεση του προγράμματος των μαθητών κατά τη σύνταξη και διόρθωσή του.

5. Αποθήκευση πίνακα σε αρχείο

Turbo Pascal

```
program files5(input, output, myfile);
uses crt;

const {ΟΡΙΑ ΤΟΥ ΠΙΝΑΚΑ}
   N=4;
   M=5;

Var
  Pinakas:array[1..M,1..N] of integer;
  i,j,MM,NN:integer;
  myfile:text;

begin
   assign (myfile,'ARXEIO.DAT');
   rewrite (myfile);

   {ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΣΤΟΝ ΠΙΝΑΚΑ}
   for i:=1 to M do
     for j:=1 to N do
      begin
        write ('PINAKAS[', I:2,',',J:2,']  ');
readln(pinakas[i,j]);
     end;

   {ΜΕΤΑΦΟΡΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΣΤΟ ΑΡΧΕΙΟ}
   write(myfile,M:4,N:4);
   for i:=1 to M do
   begin
   writeln(myfile);
       for j:=1 to N do
           write(myfile,pinakas[i,j]:5);
   end;
```

```
   close (myfile);

   writeln (' Τα στοιχεία του πίνακα μεταφέρθηκαν');
   readln
 end.
```

Quick Basic

```
' Save table
OPEN "O",#1,"ARRAY.DAT"
PRINT #1,m,n
FOR i=1 TO m
    FOR j=1 TO n
              PRINT #1,a(i,j);
    NEXT j
    PRINT #1
NEXT i
CLOSE:RETURN
```

Η ρουτίνα αυτή αποθηκεύει στην πρώτη γραμμή του αρχείου τις τιμές των διαστάσεων του πίνακα και σε κάθε επόμενη τα στοιχεία κάθε γραμμής του πίνακα.

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 ‹‹ ΔΙΑΤΙΘΕΤΑΙ 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 τμήματος αυτού του βιβλίου, που καλύπτεται απο δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Δια Βίου Μάθησης και Θρησκευμάτων/ ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.

ΕΚΔΟΣΗ: 2012 - ΑΝΤΙΤΥΠΑ: 1.000 - ΑΡ. ΣΥΜΒΑΣΗΣ:

ΕΚΤΥΠΩΣΗ: Ν. ΚΑΡΕΝΤΖΟΣ - ΒΙΒΛΙΟΔΕΣΙΑ: Α. ΠΑΠΑΔΑΚΗΣ & ΣΙΑ Ε.Ε.

11010011 11110101 11100011 11110111 11100001
11110001 11100111 11110100 11011110 11110001
11101001 11100001 00101110 00100000 11000010
11110001 11011110 11101010 11100101 11110010
00100000 11110100 11101111 00100000 11101010
11110001 11110101 11101100 11011101 11101101
11101111 00100000 11101100 11011110 11101101
11110101 11101100 11100001 00101110 00100000
11000011 11101001 11100001 00100000 11100010
11101111 11011110 11101000 11100101 11101001
11100001 00100000 11110011 11110100 11100001
00100000 11101100 11100001 11101000 11011110
11101100 11100001 11110100 11011100 00100000
11110011 11101111 11110101 00100000 11110000
11011110 11100011 11100001 11101001 11101101
11100101 00100000 11110011 11110100 11101111
00100000 01110111 01110111 01110111 00101110
01110011 01110100 01110101 01100100 01111001
00110100 01100101 01111000 01100001 01101101
01110011 00101110 01100111 01110010

Κωδικός Βιβλίου: 0-22-0205-01-2012
ISBN 978-960-06-2448-9



ΙΤΥΕ
"ΔΙΟΦΑΝΤΟΣ"
ινστιτουτο
τεχνολογιας
υπολογιστων & εκδοσεων



(02)02202050120129