

Φώτης Λαζαρίνης
Ουρανία Κουγιουρούκη

Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα

ΘΕΜΑΤΑ ΠΙΣΤΟΠΟΙΗΣΗΣ ΘΕΩΡΗΤΙΚΟΥ
ΤΜΗΜΑΤΟΣ ΕΞΕΤΑΣΕΩΝ Ι.Ε.Κ.



Περιεχόμενα

Πρόλογος	9
Κεφάλαιο 1: Δομή και λειτουργία του υπολογιστή.....	11
Κεφάλαιο 2: Χρήση Λ.Σ. DOS και Windows	19
Κεφάλαιο 3: Δίκτυα Υπολογιστών και Επικοινωνίας.....	27
Κεφάλαιο 4: Unix	37
Κεφάλαιο 5: Δομημένη Σχεδίαση Προγράμματος	45
Κεφάλαιο 6: Ανάπτυξη Εφαρμογών Προγραμματισμού	55
Κεφάλαιο 7: Χρήση Προϊόντων Λογισμικού, Επεξεργασία Κειμένου.....	59
Κεφάλαιο 8: Λογισμικό Διαχείρισης Λογιστικού Φύλλου σε Περιβάλλον Windows	65
Κεφάλαιο 9: Λογισμικό Διαχείρισης Βάσεων Δεδομένων σε Περιβάλλον Windows	73
Κεφάλαιο 10: Ανάπτυξη Εφαρμογών σε Windows	79
Κεφάλαιο 11: Θεωρία Πολυμέσων	89
Κεφάλαιο 12: Τεχνολογία Πολυμέσων (Υλικό).....	105
Κεφάλαιο 13: Επεξεργασία Εικόνας (Animation)	125

Κεφάλαιο 14: Επεξεργασία & Σύνθεση Ήχου	137
Κεφάλαιο 15: Πολυμέσα και Windows	145
Κεφάλαιο 16: Μεθοδολογία & Προγραμματισμός Πολυμέσων	153
Βιβλιογραφία	171

Κεφάλαιο

5

Δομημένη Σχεδίαση Προγράμματος

1. Τι είναι δομημένη ανάλυση και τι δομημένη σχεδίαση;

Δομημένη **ανάλυση** ονομάζεται η διαδικασία αποτύπωσης της δομής των στοιχείων ενός συστήματος, δηλαδή η εύρεση των συστατικών του μερών, των επιμέρους τμημάτων που τα αποτελούν καθώς επίσης και ο τρόπος συσχέτισης των μερών αυτών. Επίσης καθορίζει την αναγνώριση των ιδιαιτεροτήτων του συστήματος, την αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησης του, τις προδιαγραφές απόδοσης του, τον τρόπο λειτουργίας του και την περιγραφή του τρόπου επικοινωνίας μεταξύ των υποσυστημάτων του.

Η δομημένη **σχεδίαση** ανήκει στην κατηγορία τεχνικών σχεδίασης «Διαίρει και Βασίλευε» (Divide & Conquer), και λαμβάνει υπόψη τα αποτελέσματα της δομημένης ανάλυσης. Τα βήματα της είναι:

- Παρέχεται για επίλυση ένα αρχικό πρόβλημα (π.χ. η κατασκευή μιας εφαρμογής).
- Υποδιαιρείται σε μικρότερα και ευκολότερα υποπροβλήματα, που έχουν την ίδια τυποποίηση με το αρχικό πρόβλημα. Η υποδιαίρεση συνεχίζει και για τα υποπροβλήματα που προέκυψαν και τερματίζει όταν καταλήξουμε σε ένα υποπροβλημα που γνωρίζουμε τη λύση του.
- Σχεδιάζεται ανεξάρτητη λύση σε κάθε υποπρόβλημα.
- Συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν ξεκινώντας από τα τελευταία υποπροβλήματα και οδεύοντας προς το αρχικό προκειμένου να σχεδιαστεί μία συνολική λύση στο αρχικό πρόβλημα.

Σημειώνεται ότι σε αυτό το στάδιο, σχεδιάζονται οι λεπτομέρειες της υλοποίησης της λύσης του προβλήματος, επιλέγονται οι κατάλληλες δομές δεδομένων, αναπαρίσταται με κάποια μέθοδο ο αλγόριθμος της λύσης κάθε υποπροβλήματος και καθορίζεται ο τρόπος επικοινωνίας των ανεξάρτητων λύσεων, έτσι ώστε η διαδικασία μετάβασης στην προγραμματιστική υλοποίηση να είναι μια απλή διαδικασία.

2. Ποια είναι τα οφέλη από την εφαρμογή του Δομημένου σχεδιασμού;

Η δομημένη λογική σχεδίαση μας διευκολύνει:

- Στην άμεση εποπτεία της δομής του προβλήματος και της προτεινόμενης λύσης.
- Στην κατανόηση της δομής του προβλήματος και της προτεινόμενης λύσης.
- Στη μείωση του χρόνου και του κόστους σχεδιασμού και ανάπτυξης της λύσης.
- Στη συντήρηση της προτεινόμενης λύσης.
- Στην άμεση επέκταση της προτεινόμενης σχεδίασης.
- Στην παράλληλη μελέτη των υποπροβλημάτων από ομάδες εργασίας.
- Στον εύκολο εντοπισμό ενός σχεδιαστικού λάθους (άρα και ενός λάθους ανάπτυξης).

- Στην άμεση κατασκευή χρονοδιαγράμματος εργασιών για τους εμπλεκόμενους φορείς.
- Στην ευέλικτη περιγραφή των προδιαγραφών λειτουργίας ενός προβλήματος.

3. Τι είναι module;

Η **δομημένη ενότητα** (module) είναι ένας αυτοτελής αλγόριθμος που έχει μια είσοδο, μία έξοδο και εκτελεί μια καθορισμένη επεξεργασία, βασιζόμενη στα στοιχεία εισόδου και στις προδιαγραφές που εξυπηρετεί. Αποτελείται από μια συλλογή δηλώσεων, εντολών και διαδικασιών αποθηκευμένα σαν μια ενιαία μονάδα.

Ο αλγόριθμος υλοποιείται εσωτερικά μόνο με χρήση των βασικών δομών της ακολουθίας, της επιλογής και της επανάληψης, προκειμένου να ικανοποιεί τα κριτήρια του Δομημένου Προγραμματισμού και να γίνεται άμεσα κατανοητή η ροή των εντολών του. Το module παρέχει μια μορφή αφαίρεσης, αφού προκειμένου να το χρησιμοποιήσουμε αρκεί να γνωρίζουμε τη λειτουργία που επιτελεί και τα δεδομένα εισόδου που απαιτεί.

4. Περιγράψτε τα στάδια αντιμετώπισης και επίλυσης ενός προβλήματος.

Το πρώτο στάδιο είναι η **ανάλυση** του προς επίλυση προβλήματος και η **σχεδίαση** του προτεινόμενου σχεδίου λύσης, το οποίο περιλαμβάνει:

- Την αναλυτική καταγραφή των δεδομένων και των ζητούμενων του προβλήματος.
- Την πλήρη αποσαφήνιση των προδιαγραφών του προβλήματος.
- Τον καθορισμό των δομημένων ενοτήτων (modules) που θα συνθέτουν το σχέδιο λύσης.
- Την αναπαράσταση με κάποια μέθοδο του αλγόριθμου κάθε δομημένης ενότητας.
- Τον καθορισμό των δομών δεδομένων που θα χρησιμοποιηθούν.

Το επόμενο στάδιο περιλαμβάνει την **υλοποίηση** του προτεινόμενου σχεδίου λύσης, το οποίο περιλαμβάνει:

- Την επιλογή γλώσσας προγραμματισμού.
- Την κωδικοποίηση των δομημένων ενοτήτων, ξεκινώντας το τελευταίο επίπεδο και συνεχίζοντας ανοδικά.

Το επόμενο στάδιο περιλαμβάνει τον **έλεγχο** του προγράμματος σχεδίου λύσης, το οποίο περιλαμβάνει:

- Την εκτέλεση του προγράμματος με τα δεδομένα ελέγχου που είχε ελεγχθεί και ο αλγόριθμος για να διαπιστωθεί αν παράγει τα επιθυμητά αποτελέσματα.
- Τον εντοπισμό και τη διόρθωση των λογικών λαθών που μπορεί να οφείλονται είτε στην κωδικοποίηση είτε στη λανθασμένη επικοινωνία δομημένων ενοτήτων.

Το επόμενο στάδιο περιλαμβάνει τη **συντήρηση** του προγράμματος σχεδίου λύσης, το οποίο περιλαμβάνει:

- Τον εντοπισμό λαθών που προκύπτουν από τη χρήση του προγράμματος
- Τη βελτιστοποίηση του τρόπου υλοποίησης μερικών ή και όλων των δομημένων ενοτήτων και του τρόπου επικοινωνίας των
- Την επέκταση του προγράμματος ώστε να ικανοποιεί πρόσθετες ανάγκες, που δεν είχαν προβλεφθεί στην αρχική σχεδίαση

5. Τι είναι αλγόριθμος;

Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

6. Τι είναι ψευδοκώδικας;

Ο **ψευδοκώδικας** είναι ένας τρόπος περιγραφής και αναπαράστασης αλγορίθμων με χρήση ειδικών λέξεων μιας γλώσσας προγραμματισμού ή μιας φυσικής γλώσσας, που αν εκτελεστεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

Για παράδειγμα ο ακόλουθος αλγόριθμος υπολογισμού του εμβαδόν κύκλου ($3,14 \cdot R^2$) έχει γραφεί με ψευδοκώδικα.

```
Αλγόριθμος Υπολογισμός  
Εμφάνισε “Δώσε ακτίνα R”  
Διάβασε R  
 $E \leftarrow 3.14 \cdot R \cdot R$   
Εμφάνισε E  
Τέλος Υπολογισμός
```

7. Περιγράψτε τι είναι:

α) **Τμηματικός προγραμματισμός**

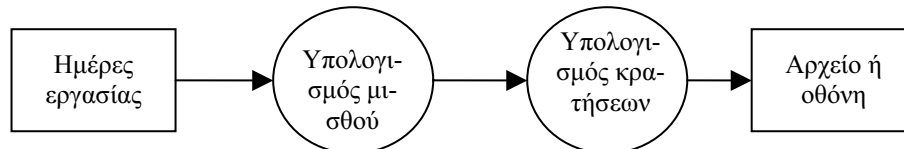
β) **Δομημένος προγραμματισμός**

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

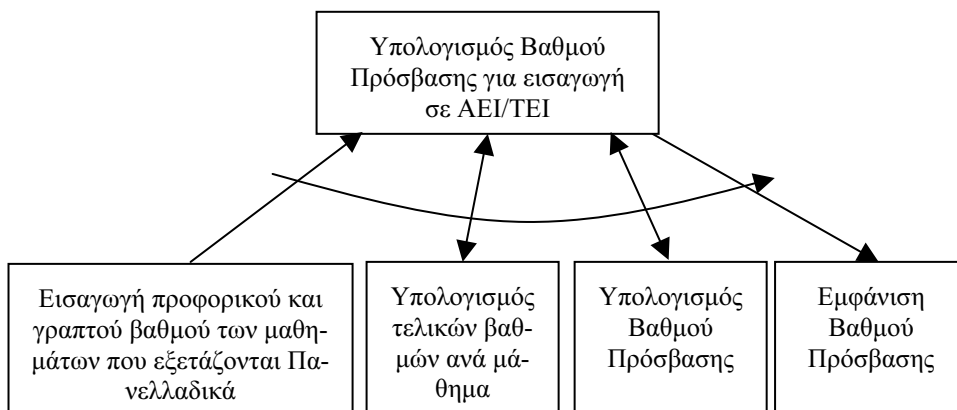
Δομημένος προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων που στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών, τη δομή της ακολουθίας, τη δομή της επιλογής και τη δομή της επανάληψης. Κάθε πρόγραμμα ή ενότητα προγράμματος έχει μία μόνο είσοδο και μία μόνο έξοδο και μπορεί να γραφεί χρησιμοποιώντας συνδυασμούς αυτών των τριών δομών.

Ο δομημένος προγραμματισμός ενθαρρύνει και βοηθά την διάσπαση ενός προγράμματος σε επιμέρους τμήματα, και εμπεριέχει τόσο την ιεραρχική σχεδίαση όσο και τον τμηματικό προγραμματισμό.

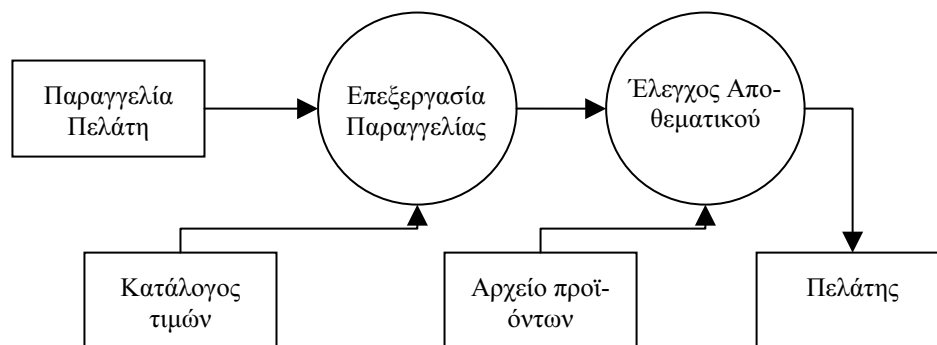
8. Σχεδιάστε ένα διάγραμμα ροής των δεδομένων ενός δομημένου διαγράμματος.



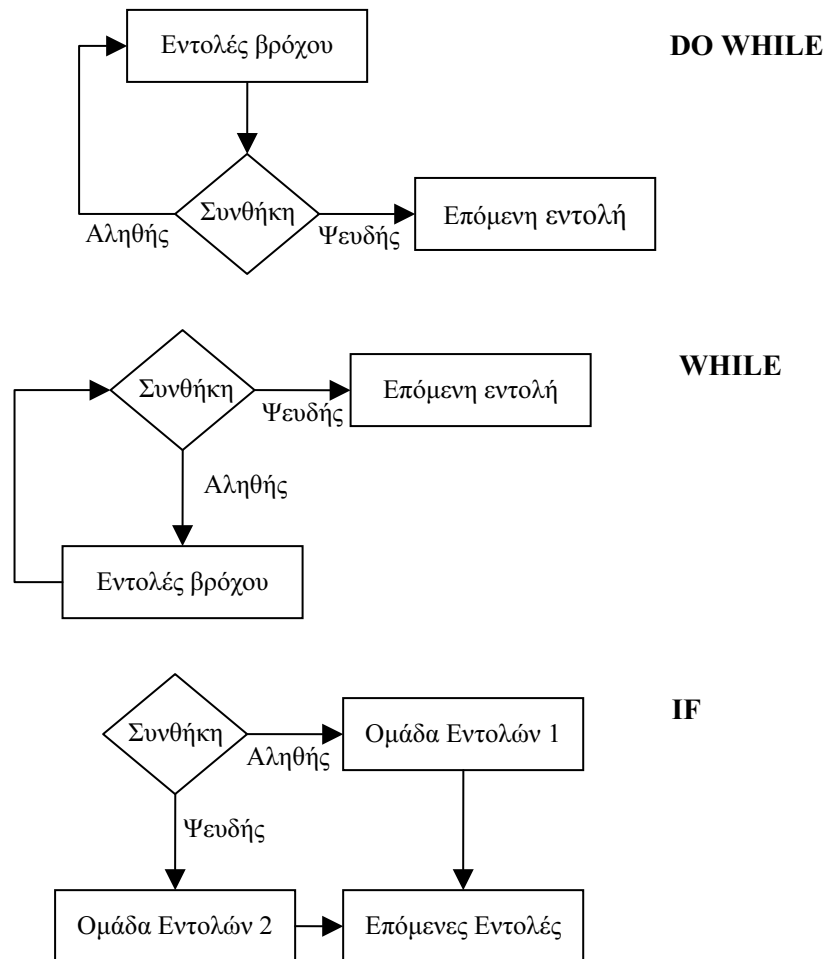
9. Σχεδιάστε ένα δομημένο διάγραμμα.



10. Σχεδιάστε ένα Flow - chart.



11. Παραστήσατε σχηματικά τις δομές DO-WHILE, WHILE και IF.



12. Παραστήσατε σχηματικά τις δομές DO-WHILE & WHILE και περιγράψτε τις διαφορές τους.

Η σχηματική αναπαράσταση δόθηκε στην ερώτηση 11. Οι διαφορές των εντολών είναι:

- Στην εντολή DO-WHILE ο έλεγχος εισόδου στην επανάληψη γίνεται στο τέλος αυτής, ενώ στην εντολή WHILE γίνεται στην αρχή.
- Στην εντολή DO-WHILE η συνθήκη ελέγχου πρέπει να είναι Ψευδής για να συνεχιστεί η επανάληψη, ενώ στην εντολή WHILE πρέπει να είναι Αληθής.
- Στην εντολή DO-WHILE αν δεν ισχύει η συνθήκη εισόδου στην επανάληψη ο βρόχος επανάληψης εκτελείται τουλάχιστον μία φορά, ενώ στην WHILE δεν θα εκτελεστεί καθόλου.

- Η εντολή WHILE είναι η πιο γενική, αφού μέσω αυτής μπορεί να κατασκευαστεί η εντολή DO-WHILE

13. Αναφέρατε τα στάδια της δομημένης σχεδίασης του προγράμματος.

Δες ερώτηση 1.

14. Περιγράψατε την μεθοδολογία σχεδιασμού ενός προγράμματος.

Δες ερώτηση 4.

15. Αναφέρατε κατηγορία πινάκων.

Ανάλογα με την διάστασή τους χαρακτηρίζονται ως μονοδιάστατοι και πολυδιάστατοι. Ανάλογα με το αν διαθέτουν σταθερό ή μεταβαλλόμενο πλήθος στοιχείων χαρακτηρίζονται ως στατικοί ή δυναμικοί πίνακες.

16. Περιγράψτε την σύζευξη δύο σειριακών αρχείων.

Η σύζευξη δύο σειριακών αρχείων αναφέρεται στην τοποθέτηση των εγγραφών δυο ταξινομημένων αρχείων σε ένα καινούριο το οποίο θα είναι επίσης ταξινομημένο. Τα αρχεία εισόδου πρέπει να είναι ταξινομημένα ως προς το ίδιο πεδίο κλειδί και η σειρά αυτή θα διατηρηθεί και στο καινούριο αρχείο. Η διαδικασία έχει ως εξής:

Αλγόριθμος Merge

Άνοιξε αρχείο1 για διάβασμα

Άνοιξε αρχείο2 για διάβασμα

Άνοιξε αρχείο3 για εγγραφή

Όσο όχι Τέλος Αρχείου1 και όχι Τέλος Αρχείου2 επανέλαβε

Διάβασε από το αρχείο1 εγγραφή R1

Διάβασε από το αρχείο2 εγγραφή R2

Αν $R1.κλειδί < R2.κλειδί$ Τότε

Γράψε την εγγραφή R1 στο αρχείο3

Αλλιώς

Γράψε την εγγραφή R2 στο αρχείο3

Τέλος_Αν

Τέλος_επανάληψης

Αν όχι Τέλος Αρχείου1 Τότε

Όσο όχι Τέλος Αρχείου1 επανέλαβε

Διάβασε από το αρχείο1 εγγραφή R

Γράψε την εγγραφή R στο αρχείο3

Τέλος_επανάληψης

Αλλιώς

Όσο όχι Τέλος Αρχείου2 επανέλαβε

Διάβασε από το αρχείο2 εγγραφή R
Γράψε την εγγραφή R στο αρχείο3
Τέλος_επανάληψης
Τέλος_Αν
Κλείσε αρχείο1
Κλείσε αρχείο2
Κλείσε αρχείο3
Τέλος Merge

17. Ποιες είναι οι διαφορές της τεκμηρίωσης ενός προγράμματος από την συντήρησή του.

Η **τεκμηρίωση** ενός προγράμματος δεν συνιστά ανεξάρτητη φάση, αλλά μια παράλληλη διαδικασία που συμπληρώνει όλα τα στάδια ανάπτυξης του προγράμματος, σε αντίθεση με τη συντήρηση που αποτελεί την τελευταία φάση στην ανάπτυξη ενός προγράμματος.

Η **συντήρηση** μιας εφαρμογής είναι κάθε εργασία που γίνεται, ενώ έχει τεθεί σε λειτουργία η εφαρμογή, είτε για να διορθωθεί κάποιο σφάλμα αυτής είτε για να τροποποιηθεί, προκειμένου να εξοπλισθεί με νέες λειτουργίες ή να βελτιστοποιήσει τις ήδη υπάρχουσες. Αντίθετα η τεκμηρίωση είναι η πληροφορία που έχει σχεδιαστεί με σκοπό τη διευκόλυνση της αλληλεπίδρασης μεταξύ του λογισμικού και των ανθρώπων που το διοικούν, το ελέγχουν, το χειρίζονται ή το συντηρούν.

Οι αλλαγές που επιφέρει η διαδικασία της συντήρησης αντανακλούν και στην τεκμηρίωση.

18. Ποιες είναι οι διαφορές των δοκιμών από τους ελέγχους ενός προγράμματος.

Οι **έλεγχοι** ενός προγράμματος αποσκοπούν στην εύρεση και διόρθωση των συντακτικών και λογικών λαθών των ενοτήτων του και του συνολικού προγράμματος. Κύριος σκοπός του ελέγχου είναι η διασφάλιση της σωστής λειτουργίας του προγράμματος αφού έχει τελειώσει η υλοποίησή του. Υπεύθυνη για τον έλεγχο είναι η ομάδα ανάπτυξης του προγράμματος.

Η **δοκιμή** ενός προγράμματος, που έπεται των ελέγχων, είναι η εκτέλεση του με εικονικά δεδομένα εσόδου προκειμένου να εξακριβωθεί η ορθότητα του και να διαπιστωθεί αν παράγει σωστά αποτελέσματα (σύμφωνα με τα αναμενόμενα αποτελέσματα του αλγόριθμου στον οποίο στηρίχτηκε) και αν τηρεί τις προδιαγραφές λειτουργίας του. Υπεύθυνοι για τη δοκιμή ενός προγράμματος είναι η ομάδα ανάπτυξης του προγράμματος και οι τελικοί χρήστες.

19. Αναφέρατε παράγοντες που επηρεάζουν την φιλικότητα, ευελιξία, αξιοπιστία και ταχύτητα του προγράμματος.

Η **φιλικότητα** ενός προγράμματος επηρεάζεται από:

- Την ύπαρξη γραφικού περιβάλλοντος διεπαφής.
- Την καλαισθησία, λειτουργικότητα και ευχρηστία του περιβάλλοντος διεπαφής.
- Την αμεσότητα απόκρισης στις εντολές του χρήστη.
- Την ύπαρξη συστήματος παροχής άμεσης βοήθειας.
- Την ύπαρξη βοηθών (wizards) που αυτοματοποιούν περίπλοκες λειτουργίες.
- Την απόκρυψη προγραμματιστικών λεπτομερειών.

Η **ευελιξία** ενός προγράμματος επηρεάζεται από:

- Τη δυνατότητα προσαρμογής του προγράμματος στις ανάγκες του χρήστη και σε νέες απαιτήσεις.
- Την ύπαρξη εναλλακτικών τρόπων για την εκτέλεση μιας λειτουργίας.
- Τη δυνατότητα εκτέλεσης του προγράμματος σε διαφορετικές πλατφόρμες.
- Τη δυνατότητα αναίρεσης ενεργειών.
- Την έλλειψη προγραμματιστικών περιορισμών.

Η **αξιοπιστία** ενός προγράμματος επηρεάζεται από:

- Την αδιάλειπτη και ομοιόμορφη λειτουργία του, βάσει των προδιαγραφών σχεδίασης του.
- Τη λειτουργικότητα του ανεξάρτητα από το βαθμό επίγνωσης των χρηστών.
- Τη συνεπή αντιμετώπιση εμφανιζόμενων λαθών καθώς και ακραίων περιπτώσεων δεδομένων και ενεργειών.
- Την προσεκτικά σχεδιασμένη υλοποίηση του.

Η **ταχύτητα** ενός προγράμματος επηρεάζεται από:

- Το προγραμματιστικό περιβάλλον υλοποίησης
- Τη χρήση βελτιστοποιητικών μεταφραστών.
- Την υπολογιστική ισχύ και τους υπολογιστικούς πόρους του μηχανήματος στο οποίο θα εκτελεστεί.
- Την επιλογή γρήγορων και αποδοτικών δομών δεδομένων.
- Την ελαχιστοποίηση άσκοπων ενεργειών.
- Τον όγκο των προς επεξεργασία δεδομένων.

Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα

ΘΕΜΑΤΑ ΠΙΣΤΟΠΟΙΗΣΗΣ ΘΕΩΡΗΤΙΚΟΥ
ΤΜΗΜΑΤΟΣ ΕΞΕΤΑΣΕΩΝ Ι.Ε.Κ.

Το βιβλίο περιέχει τα θέματα του θεωρητικού τμήματος πιστοποίησης που έχει προτείνει ο ΟΕΕΚ για την ειδικότητα "Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα" των ΙΕΚ.

Το βιβλίο είναι χωρισμένο σε δυο ενότητες:

- Γενικές γνώσεις
- Ειδικές γνώσεις

Κάθε ενότητα περιέχει τις απαντήσεις στα θέματα πιστοποίησης.

Οι συγγραφείς

Ο **Φώτης Ελ. Λαζαρίνης** είναι πτυχιούχος Πληροφορικής του Πανεπιστημίου Αθηνών και κάτοχος μεταπτυχιακού στα Προηγμένα Πληροφορικά Συστήματα από το Πανεπιστήμιο Γλασκόβης. Έχει εργασθεί ως Μηχανικός Λογισμικού και πλέον εργάζεται ως εκτ. Καθηγητής Εφαρμογών ΤΕΙ και εκπαιδευτικός Μέσης Εκπαίδευσης. Διαθέτει μεγάλη διδακτική πείρα σε προγραμματισμό και χρήση υπολογιστών. Έχει συμμετάσχει με εισηγήσεις και άρθρα σε διεθνή και Ελληνικά συνέδρια και περιοδικά. Είναι συγγραφέας 12 βιβλίων.

Η **Ουρανία Κουγιουρούκη** είναι Φυσικός και επιμορφώτρια ενηλίκων στις Νέες Τεχνολογίες. Έχει εργαστεί επί σειρά ετών ως εισηγήτρια σε σεμινάρια προγραμματισμού και χρήσης υπολογιστών και ως συνεργάτης ΤΕΙ. Εργασίες της έχουν δημοσιευθεί σε διεθνή και Ελληνικά περιοδικά. Έχει συγγράψει 2 βιβλία.

Επισκεφθείτε μας στο Internet
<http://www.klidarithmos.gr>



ΚΛΕΙΔΑΡΙΘΜΟΣ

Σολωμού 57, 10432, ΑΘΗΝΑ, Τηλ. 210-5217635

ISBN 960-209-801-5



9 789602 098011