

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

## ΕΠΛ231: Δομές Δεδομένων και Αλγόριθμοι

**Διδάσκοντες:** Γιώργος Πάλλης **Γραφείο:** ΘΕΕ-01 B119

**E-mail:** [pallis@ucy.ac.cy](mailto:pallis@ucy.ac.cy)

**Ιστοσελίδα Μαθήματος:** <http://www.cs.ucy.ac.cy/courses/EPL231>

### ΠΕΡΙΓΡΑΦΗ

Το μάθημα μελετά μεθόδους οργάνωσης των πληροφοριών, αλγόριθμους μετασχηματισμού τους και την ανάλυση της πολυπλοκότητας των αλγορίθμων. Για την υλοποίηση των προγραμμάτων θα χρησιμοποιηθεί η γλώσσα προγραμματισμού JAVA.

### ΔΙΑΛΕΞΕΙΣ ΚΑΙ ΕΡΓΑΣΤΗΡΙΟ

Η διδασκαλία του μαθήματος αποτελείται από διαλέξεις και φροντιστήρια (συμπληρωματικές διαλέξεις, επίλυση ασκήσεων και θεωρητική εμπέδωση της ύλης). Η παρακολούθηση των διαλέξεων από τους φοιτητές είναι υποχρεωτική. Οι φοιτητές παρακαλούνται όπως προσέρχονται στην αίθουσα των διαλέξεων έγκαιρα. Στα εργαστήρια του μαθήματος οι φοιτητές θα έχουν την δυνατότητα να υποβοηθούνται στην υλοποίηση των αρχών που διδάσκονται στις διαλέξεις. Φροντιστήρια και Εργαστήρια θα γίνονται κάθε εβδομάδα στις προγραμματισμένους ώρες και αίθουσες, εκτός και αν γίνει σχετική ανακοίνωση.

Το υλικό των διαλέξεων και του φροντιστηρίου θα ανεβαίνει στις σελίδες του μαθήματος. Το υλικό για τα εργαστήρια θα ανεβαίνει στο **blackboard**. Ανακοινώσεις και απορίες για το μάθημα θα γίνονται επίσης μέσω της πλατφόρμας **blackboard**.

### ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ

Τα μαθήματα **ΕΠΛ 111** (Διακριτές Δομές στην Πληροφορική και Υπολογισμό) και **ΕΠΛ 133** (Αντικειμενοστραφή Προγραμματισμό) είναι προαπαιτούμενα για την παρακολούθηση του ΕΠΛ 231.

### ΣΤΟΧΟΙ

Με το πέρας του μαθήματος οι φοιτητές αναμένονται να είναι σε θέση:

- Να αναλύουν και να συγκρίνουν την αποδοτικότητα αλγορίθμων βάσει των τάξεων  $O$ ,  $\Omega$  και  $\Theta$ .
- Να σχεδιάζουν και να υλοποιούν αλγόριθμους οι οποίοι ελαχιστοποιούν τον χρόνο εκτέλεσής τους όπως και τον χώρο που χρησιμοποιούν.
- Να χρησιμοποιούν, να υλοποιούν και να επεκτείνουν τις δομές δεδομένων που θα μελετηθούν στο μάθημα.
- Να εφαρμόζουν τους αλγόριθμους που θα μελετηθούν στο μάθημα σε τυχαία δεδομένα.
- Να ορίζουν τους αφηρημένους τύπους δεδομένων που απαιτούνται για την οργάνωση των δεδομένων εισόδου προβλημάτων.
- Να επιλέγουν ή και να δημιουργούν κατάλληλες δομές δεδομένων και κατάλληλους αλγόριθμους για υλοποίηση αφηρημένων τύπων δεδομένων.
- Να επιλύουν και να υλοποιούν αποδοτικές λύσεις σε σύνθετα υπολογιστικά προβλήματα χρησιμοποιώντας τη γλώσσα JAVA.

## ΠΕΡΙΕΧΟΜΕΝΟ

- Πολυπλοκότητα αλγόριθμων και ανάλυση μέσης και χειρίστης περίπτωσης
- Τύποι δεδομένων και αφηρημένοι τύποι δεδομένων
- Τύποι λίστας, στοίβας και ουράς
- Αλγόριθμοι ταξινόμησης και ανάλυση της αποδοτικότητάς τους
- Μη γραμμικές δομές δεδομένων. Δένδρα. Δένδρα διερεύνησης
- Ισοζυγισμένα δένδρα
- Ουρές προτεραιότητας
- Γράφοι και αλγόριθμοι επεξεργασίας τους
- Bit-Διανύσματα. Τεχνικές κατακερματισμού (hashing)

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Συνίσταται η χρήση των πιο κάτω βιβλίων καθώς και η χρήση των διαφανειών του διδάσκοντα.

- M. A. Weiss, Data Structures and Algorithm Analysis in JAVA. 3rd edition, Addison Wesley, 2011
- R. Sedgewick and K. Wayne Algorithms in JAVA, 4th edition by, Addison-Wesley, 2011
- Για περισσότερες πληροφορίες μπορείτε να συμβουλευτείτε και το ακόλουθα βιβλίο: M. Goodrich, R. Tamassia. Data Structures and Algorithms in JAVA, 5th edition, Wiley, 2011

Παρακαλείστε όπως προμηθεύσετε τις σημειώσεις των διαλέξεων από την ιστοσελίδα του μαθήματος πριν από την έναρξη *κάθε διάλεξης*.

## ΑΞΙΟΛΟΓΗΣΗ

Το μάθημα αξιολογείται με 7.5 μονάδες ECTS. Η επίδοση των φοιτητών θα αξιολογείται συνεχώς με βάση τη συμμετοχή στο μάθημα και κατ' οίκον εργασία. Η κατ' οίκον εργασία θα περιλαμβάνει υποχρεωτικές θεωρητικές και προγραμματιστικές ασκήσεις. Επίσης θα υπάρξει μία ενδιάμεση εξέταση, 3 διαγνωστικά τεστ (quizzes) και μια τελική εξέταση. Η αναλογία ως προς τον τελικό βαθμό είναι η εξής:

- **Τελική εξέταση:** 40%
- **Διαγνωστικά τεστ:** 40% - Τα διαγνωστικά τεστ θα δίνονται στα πλαίσια του φροντιστηριακού μαθήματος. Οι ημερομηνίες είναι στο πρόγραμμα διδασκαλίας του μαθήματος. Κάθε διαγνωστικό τεστ θα μετρά 10% της βαθμολογίας. Αν ένας φοιτητής δε μπορέσει να παρευρεθεί δικαιολογημένα σε ένα διαγνωστικό τεστ (π.χ., λόγω ασθένειας με πιστοποιητικό γιατρού, παρουσία στην Εθνική Φρουρά ως έφεδρος), η βαρύτητα της βαθμολογίας του θα επιμερισθεί στα άλλα τρία διαγνωστικά τεστ.
- **Προγραμματιστικές Ασκήσεις:** 15% (Οι προγραμματιστικές ασκήσεις θα εξετάζονται με παρουσίαση της άσκησης από το φοιτητή). Αν ο φοιτητής κάνει 3 αδικαιολόγητες απουσίες τότε δε θα προσμετρείται η βαθμολογία που έλαβε στις προγραμματιστικές ασκήσεις.
- **Θεωρητικές Ασκήσεις:** 5% με την υποβολή όλων των ασκήσεων
- **Quiz της ημέρας:** 5% bonus (2 απροειδοποίητα διαγνωστικά τεστ (quiz) πάνω στην ύλη του μαθήματος που διδάχθηκε τη συγκεκριμένη μέρα. Η βαθμολογία τους θα προσμετρείται αν ο φοιτητής/ρια επιτυγχάνει πάνω από 60%).

Για την εξασφάλιση προβιβάσιμου βαθμού στο μάθημα (δηλ. τουλάχιστον πέντε), απαιτείται **ο σταθμισμένος μέσος όρος της τελικής εξέτασης και των διαγνωστικών τεστ να είναι τουλάχιστον 4,5.**

## **ΟΔΗΓΙΕΣ ΕΚΠΟΝΗΣΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΑΣΚΗΣΕΩΝ**

Η παράδοση των προγραμματιστικών ασκήσεων θα γίνεται σε ηλεκτρονική μορφή στον υπεύθυνο βοηθό μέσω του blackboard.

Οι προγραμματιστικές ασκήσεις θα κρίνονται με βάση τα ακόλουθα τρία κριτήρια:

- **Ορθότητα:** ο πρόγραμμα σας πρέπει τρέχει ορθά για οποιαδήποτε είσοδο.
- **Κατανοητά Σχόλια:** Γράφετε κατανοητά σχόλια που να εξηγούν την λειτουργία της κάθε κλάσης/πεδίου/μεθόδου.
- **Αναφορά Προγράμματος:** Θα πρέπει να παραδώσετε μία αναφορά η οποία να περιγράφει συνοπτικά τα διάφορα συστατικά (κλάσεις, μεθόδους, διαπροσωπείες, κτλ.) του προγράμματος σας. Επίσης, η αναφορά θα πρέπει να καταγράφει τυχόν υποθέσεων τις οποίες έχετε κάνει.
- **Αποδοτικότητα: το πρόγραμμα σας:** θα πρέπει να χρησιμοποιήσει αποδοτικές δομές δεδομένων και αλγόριθμους όπου αρμόζει. Κατά την υλοποίηση του προγράμματος σας, ακολουθείστε τους παρακάτω κανόνες:
  - **IDE:** Προτρέπεται η χρήση ενός Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης (π.χ., eclipse IDE) για τη συγγραφή των προγραμματιστικών σας εργασιών.
  - **Σχεδίαση Προγράμματος:** Προτρέπεται όπως το πρόγραμμα σας σχεδιάζεται από πάνω προς τα κάτω, αποφασίζοντας δηλαδή πρώτα τις κλάσεις και διασυνδέσεις τους, στη συνέχεια τις μεταβλητές και μεθόδους και τέλος υλοποιώντας το σώμα των συναρτήσεων. Θα πρέπει να υπάρχει η κατάλληλη άρθρωση, στοίχιση του κώδικα, σαφήνεια και κατάλληλη ονοματολογία σε οντότητες του προγράμματος.
  - **Σχόλια:** Τα αντικείμενα σας θα πρέπει να είναι σχολιασμένα έτσι ώστε να μπορεί να προκύπτει κάποιος οδηγός λειτουργίας μέσω javadoc. Ο ίδιος ο οδηγός δε χρειάζεται να είναι μέρος του παραδοτέου.
  - **Έλεγχος Μονάδων:** Κάθε αντικείμενο πρέπει να συνοδεύεται από τους σχετικούς οδηγούς ελέγχου (drivers), οι οποίοι θα επικυρώνουν την ορθότητα των συναρτήσεων και λειτουργιών του εν λόγω αντικειμένου (unit testing). Οι έλεγχοι θα πρέπει να διερευνούν τις ακραίες περιπτώσεις λειτουργίας των συναρτήσεων και του αντικειμένου ευρύτερα. Προτρέπεται επίσης η χρήση εργαλείων ανάλυσης και βελτιστοποίησης του προγράμματος (π.χ., χρήση profilers).
  - **GPL:** Σε μια προσπάθεια εξοικείωσης με τεχνικές ανάπτυξης ανοικτού λογισμικού, συνίσταται η χρήση του GPL προοιμίου (preamble) σε κάθε αρχείο πηγαίου κώδικα.

**Σε περίπτωση αξιοποίησης εργαλείων Τεχνητής Νοημοσύνης (TN) κατά την εκπόνηση εργασιών, ο/η φοιτητής/τήτρια θα πρέπει να αναφέρει ρητά και επακριβώς το εργαλείο TN που χρησιμοποίησε και με ποιον τρόπο το αξιοποίησε. Η αυτούσια αντιγραφή από εργαλεία TN απαγορεύεται.**

## ΟΔΗΓΙΕΣ ΕΚΠΟΝΗΣΗΣ ΘΕΩΡΗΤΙΚΩΝ ΕΡΓΑΣΙΩΝ

- Για κάθε άσκηση, διαβάστε την εκφώνηση της προσεκτικά και σιγουρευτείτε ότι κατανοείτε αυτό που σας ζητείται να κάνετε.
- Διατυπώνετε τις σκέψεις σας με τρόπο κατά το δυνατόν απλό, κατανοητό και συγχρόνως μεστό, αποφεύγοντας τις διφορούμενες απαντήσεις.
- Μη γράφετε περισσότερα από αυτά που ζητούνται σε κάθε θέμα.
- Εξετάζετε πάντοτε στο τέλος πόσο λογικό είναι το αποτέλεσμα στο οποίο καταλήξατε.
- Υπάρχουν δύο είδη προβλημάτων: προβλήματα απόδειξης και αλγοριθμικά προβλήματα.

**(α) Προβλήματα απόδειξης:** Για αυτό το είδος προβλημάτων θα πρέπει να δίνονται καθαρά όλα τα βήματα της απόδειξης με σαφείς επεξηγήσεις σχετικά με το πώς έχουν προκύψει. Βήματα της απόδειξης τα οποία παραλείπονται δυνατό να θεωρηθούν σημαντικά και να οδηγήσουν σε μείωση της βαθμολογίας ανεξάρτητα από την τελική κατάληξη της απόδειξης.

**(β) Αλγοριθμικά προβλήματα:** Μια πλήρης λύση θα αναφέρει τα ακόλουθα:

- i. Μια σύντομη περιγραφή του αλγορίθμου με λόγια.
- ii. Τις εισόδους και τις εξόδους του αλγορίθμου, τους τύπους των δεδομένων, τις δομές που χρησιμοποιήσατε και οτιδήποτε άλλο θα πρέπει να ξέρει κάποιος που διαβάζει τη λύση σας.
- iii. Οι αλγόριθμοι θα πρέπει να περιγράφονται από μια σειρά από καθαρά βήματα. Στους αλγόριθμους θα μπορούν να χρησιμοποιηθούν αλγόριθμοι που θα συζητηθούν στην τάξη χωρίς περιγραφή. Αν όμως χρησιμοποιηθεί κομμάτι κάποιου αλγορίθμου ή/και κομμάτι κάποιας δομής δεδομένων αυτά θα πρέπει να εξηγηθούν.
- iv. Οι αλγόριθμοί σας θα πρέπει να είναι ορθοί για όλα τα στιγμιότυπα του προβλήματος και να λαμβάνουν υπόψη τους ακόμα και τις οριακές περιπτώσεις.
- v. Για το σύνολο των μονάδων ενός προβλήματος, ο αλγόριθμος θα πρέπει να είναι και ορθός και βέλτιστος. Ένας ορθός αλγόριθμος θα παίρνει μονάδες, έστω και αν αυτός δεν είναι βέλτιστος.
- vi. Για κάθε αλγόριθμο που παρουσιάζετε θα πρέπει να παραθέτετε και τον χρόνο εκτέλεσής του.
- vii. Για την επίλυση των ασκήσεων απαγορεύεται η χρήση μεταβλητών που ορίζονται ως static ή global. Η χρήση στατικών ή γενικών μεταβλητών θα οδηγήσει αυτόματα σε μηδενισμό της άσκησης.
- viii. Στις αναδρομικές συναρτήσεις δεν πρέπει να περνούν παράμετροι δείκτες σε μεταβλητές οι οποίες επιστρέφουν την τελική απάντηση. Κάθε κλήση της συνάρτησης πρέπει να λύνει ένα κομμάτι του προβλήματος και να επιστρέφει την απάντησή της στην κλήση που την έχει καλέσει για περαιτέρω επεξεργασία.

Σε περίπτωση αξιοποίησης εργαλείων Τεχνητής Νοημοσύνης (TN) κατά την εκπόνηση εργασιών, ο/η φοιτητής/τήτρια θα πρέπει να αναφέρει **ρητά και επακριβώς το εργαλείο TN που χρησιμοποίησε και με ποιον τρόπο το αξιοποίησε. Η αυτούσια αντιγραφή από εργαλεία TN απαγορεύεται.**