

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Παράρτημα Χανίων

Τμήμα Τηλεπικοινωνιών & Δικτύων Η/Υ



Θέμα:

**Αυτοματοποιημένος
χωροχρονοπρογραμματισμός των
παραδόσεων και των εξετάσεων στα
Ιδρύματα Τεχνολογικής Εκπαίδευσης**

Πτυχιακή εργασία

Σπουδαστή : Παντελάκη Παναγιώτη

Επιβλέπων : Δρ. Μηχ. Πετράκης Νικόλαος

**Χανιά
Ιούνιος 2003**

Εν' συντομία

Το πρόβλημα της δημιουργίας του ωρολογίου προγράμματος παραδόσεων (ΩΠΠ) και του ωρολογίου προγράμματος εξετάσεων (ΩΠΕ) των συμβατικών Τμημάτων του Παραρτήματος Χανίων παίρνει ολοένα και μεγαλύτερες διαστάσεις. Βασικές αιτίες είναι η ίδρυση νέων Τμημάτων, η προσθήκη καινούργιων μαθημάτων, η αύξηση του αριθμού των εισακτέων φοιτητών καθώς και η αναπόφευκτη διεύρυνση του εκπαιδευτικού προσωπικού. Όμως, όλα τα παραπάνω είναι θεμιτά και επικουρούν στην ανάπτυξη του Ιδρύματος. Ο καθοριστικότερος παράγοντας που τείνει να κάνει το πρόβλημα άλυτο είναι ο περιορισμένος αριθμός των χώρων διδασκαλίας που έχει μείνει σχεδόν στάσιμος τα τελευταία χρόνια. Αν συνυπολογιστούν και οι χωρητικότητές των, είτε σε ώρα παράδοσης, είτε σε ώρα εξέτασης τότε κρίνονται απολύτως ανεπαρκείς.

Η παρούσα διπλωματική εργασία άδραξε την πρόκληση για την εύρεση αυτοματοποιημένης λύσης του παραπάνω προβλήματος το οποίο μελέτησε προσεκτικά, ανέλυσε σε βάθος, συνέθεσε τις αλγοριθμικές λύσεις των επιμέρους προβλημάτων, τις υλοποίησε και αφήνει σε εσάς ... να κρίνετε το αποτέλεσμα!

1. Εισαγωγή

Παρατηρείται ότι η ταχύτητα, με την οποία εξελίσσεται η κοινωνία μας, καθώς και η έλλειψη του ελεύθερου χρόνου είναι φαινόμενα που χαρακτηρίζουν τη σημερινή εποχή. Επίσης, με την πάροδο του χρόνου αυξάνουν οι απαιτήσεις καθώς και η ανάγκη για την επίλυση προβλημάτων γρήγορα και αξιόπιστα. Αντιμετωπίζοντας αυτές τις προκλήσεις, αναπτύχθηκε η έρευνα και η τεχνολογία. Ένα από τα προβλήματα τα οποία υπάρχουν και προκάλεσε το ενδιαφέρον μας είναι η δημιουργία του Ωρολογίου Προγράμματος Παραδόσεων (ΩΠΠ) και του Ωρολογίου Προγράμματος Εξετάσεων (ΩΠΕ) στα Τεχνολογικά Εκπαιδευτικά Ιδρύματα.

Το Τμήμα Ηλεκτρονικής, αρχικά ήταν το μόνο Τμήμα του Παραρτήματος Χανίων και είχε μικρό αριθμό διδασκόντων και φοιτητών. Με την πάροδο του χρόνου όμως οι ανάγκες πολλαπλασιάστηκαν. Προστέθηκαν νέα Τμήματα, αλλά και ο αριθμός των εισακτέων φοιτητών ανά Τμήμα αυξήθηκε. Δημιουργήθηκε το Τμήμα Τηλεπικοινωνιών και Δικτύων Ηλεκτρονικών Υπολογιστών, και στη συνέχεια το Τμήμα Φυσικών Πόρων και Περιβάλλοντος. Έτσι, αυξήθηκαν οι ανάγκες σε χώρους διδασκαλίας, αλλά, δυστυχώς, ο αριθμός των παρέμεινε σταθερός. Η δημιουργία του ΩΠΠ και του ΩΠΕ πλέον, χειροκίνητα, ήταν μια δύσκολη και επίπονη διαδικασία.

Κρίθηκε λοιπόν αναγκαία, η δημιουργία μιας εφαρμογής η οποία, θα είχε φιλικό περιβάλλον προς το χρήστη, θα μας έδινε αποτελέσματα γρήγορα και αξιόπιστα, και θα είχε την δυνατότητα για μελλοντική επεκτασιμότητα. Θέλοντας να ικανοποιήσουμε αυτές τις απαιτήσεις, εκμεταλλευτήκαμε τις δυνατότητες των σύγχρονων λειτουργικών συστημάτων, των γλωσσών προγραμματισμού, και των συστημάτων διαχείρισης δεδομένων. Το αποτέλεσμα ήταν μια εφαρμογή, η οποία υλοποιήθηκε σε Visual C++ 6.0. Ονομάστηκε «χρονοπρογραμματισμός», και τα δεδομένα αυτής καταχωρούνται σε Βάση Δεδομένων (ΒΔ), η οποία είναι σχεδιασμένη σε Access 2000. Η επικοινωνία μεταξύ της εφαρμογής και της ΒΔ γίνεται μέσω του προτύπου ODBC.

Η παρούσα εργασία αποτελείται από επτά κεφάλαια. Μετά το πρώτο που είναι μια σύντομη εισαγωγή του προβλήματος, στο Κεφάλαιο 2 περιέχονται βασικές έννοιες των ΒΔ, περιγράφονται οι δυνατότητες της Access 2000 και αναλύεται η ΒΔ Timetable.mdb στην οποία καταχωρούνται τα δεδομένα. Επίσης, στο ίδιο κεφάλαιο γίνεται αναφορά για το πρότυπο ODBC, το οποίο είναι υπεύθυνο για την επικοινωνία της εφαρμογής μας με τη ΒΔ, καθώς και για το E-R Model της ΒΔ Timetable.mdb.

Στο Κεφάλαιο 3, αναφέρονται οι δυνατότητες της Visual C++ 6.0, και επισημαίνονται με ιδιαίτερη έμφαση αυτές που συνέβαλλαν στην υλοποίηση της εφαρμογής «χρονοπρογραμματισμός». Γίνεται αναφορά, για τις εξαιρέσεις, για τη χρησιμότητα τους μέσα σε μια εφαρμογή, για τα στοιχεία ελέγχου Active X, όπως και για τη λειτουργικότητα που παρέχουν. Συνεχίζοντας, αναφέρεται για την δυναμική κατανομή της μνήμης, και πως επιτυγχάνεται με την βοήθεια προσαρμοσμένων τάξεων της MFC.

Στο Κεφάλαιο 4 περιγράφονται οι περιορισμοί που θέτονται, και πρέπει να τηρηθούν, για την υλοποίηση του ΩΠΠ. Επίσης, φαίνεται το λογικό διάγραμμα του αλγορίθμου του ΩΠΠ και περιγράφεται αναλυτικά η λειτουργία του.

Στο Κεφάλαιο 5 αναλύουμε τη λογική του αλγορίθμου του ΩΠΕ, καθώς και τους περιορισμούς του, και παρουσιάζουμε τα λογικά διαγράμματα λειτουργίας του. Αναφέρουμε τα σημεία στα οποία μπορεί να επέμβει ο χρήστης, μέσω των ρυθμίσεων του προγράμματος, και να

μεταβάλλει τα αποτελέσματα του αλγορίθμου προς την επιθυμητή κατεύθυνση.

Στο Κεφάλαιο 6 δίνονται οδηγίες για τη χρήση της εφαρμογής. Περιγράφονται, οι τρόποι εισαγωγής, διαγραφής και μεταβολής των δεδομένων που υπάρχουν στην ΒΔ Timetable.mdb, καθώς και οι δυνατότητες της εκτύπωσης βασικών δεδομένων. Αναφέρεται στις ρυθμίσεις, που μπορούν να γίνουν από το χρήστη στους αλγορίθμους του ΩΠΠ και ΩΠΕ, και τον τρόπο με τον οποίο επηρεάζει κάθε αλλαγή.

Στο Κεφάλαιο 7 παρουσιάζονται τα συμπεράσματα που προέκυψαν στο χρονικό διάστημα της υλοποίησης της εφαρμογής μας και κάποιες σκέψεις εμπλουτισμού της εφαρμογής με περαιτέρω δυνατότητες. Γίνεται αναφορά στη βιβλιογραφία που χρησιμοποιήθηκε για την ολοκλήρωση της εφαρμογής, καθώς και κρίσεις-σχόλια για αυτήν.

Στο παράρτημα Α δίδονται αποσπάσματα κώδικα, σε Visual C++ 6.0, που χρησιμοποιήθηκε σε βασικά σημεία υλοποίησης των αλγορίθμων του ΩΠΠ και ΩΠΕ.

Τέλος στο παράρτημα Β περιέχονται δείγματα εκτυπώσεων και γίνεται μια ανάλυση για τον τρόπο με τον οποίο λειτουργεί η διαδικασία της εκτύπωσης.

Θα ήθελα να ευχαριστήσω θερμά τους πολυαγαπημένους μου γονείς Εμμανουήλ και Άννα καθώς και την αδερφή μου Ειρήνη, στους οποίους αφιερώνω αυτή την πτυχιακή εργασία, για την συμπαράσταση που μου έδειξαν ώστε να ολοκληρώσω με επιτυχία την σχολή μου.

Επίσης, ένα μεγάλο ευχαριστώ στον καθηγητή και εισηγητή μου Δρ. Μηχ. Πετράκη Νικόλαο για την καθοδήγηση και συμπαράσταση που μου έδειξε, ώστε να υλοποιηθεί επιτυχώς η συγκεκριμένη πτυχιακή εργασία.

1. Θεμελιώδεις έννοιες και ορολογία της θεωρίας των Βάσεων Δεδομένων

a. Η έννοια της Βάσης Δεδομένων

Βάση Δεδομένων (ΒΔ) (DataBase) (DB) είναι μια συλλογή από εγγραφές και αρχεία, τα οποία είναι οργανωμένα έτσι ώστε να εξυπηρετούν ένα συγκεκριμένο σκοπό. Στόχο να εξασφαλίζει την σωστή εισαγωγή δεδομένων σε αυτήν. Επίσης μας δίνει τη δυνατότητα να μοιραστούμε τις πληροφορίες μας με πολλά άτομα, αλλά κυρίως δίνει το δικαίωμα δυο ή παραπάνω άτομα να προσπελάζουν τα ίδια δεδομένα ταυτόχρονα. Αυτό επιτυγχάνεται με ένα σύστημα διαχείρισης βάσεων δεδομένων (*Database Management System, DBMS*), το οποίο είναι υπεύθυνο για τη διανομή και την αξιοπιστία των πληροφοριών μας. Σχεδόν όλα τα σύγχρονα συστήματα διαχείρισης βάσεων δεδομένων χειρίζονται και αποθηκεύουν τις πληροφορίες χρησιμοποιώντας το σχεσιακό (relational) μοντέλο διαχείρισης βάσεων δεδομένων.

Ο όρος *σχεσιακό* προκύπτει από το γεγονός ότι κάθε εγγραφή της ΒΔ περιέχει πληροφορίες συσχετισμένες με ένα θέμα. Επίσης, τα δεδομένα που αφορούν δυο κατηγορίες πληροφοριών (όπως οι διδάσκοντες και τα μαθήματα που έχουν επιλέξει για να διδάξουν) μπορούν να αντιμετωπίζονται ως μια οντότητα που βασίζεται σε συσχετισμένες τιμές δεδομένων. Για παράδειγμα θα ήταν περιττή η αποθήκευση του ονόματος και των λοιπών στοιχείων ενός διδάσκοντα, καθώς και το όνομα και το εξάμηνο του μαθήματος κάθε φορά που του καταχωρούμε ένα μάθημα που έχει επιλέξει για να διδάξει.

Έτσι σε ένα σχεσιακό σύστημα, τα στοιχεία των επιλογών μαθημάτων περιέχουν και δυο πεδία δεδομένων στα οποία αποθηκεύονται κάποια στοιχεία όπως ο κωδικός του διδάσκοντα που έχει επιλέξει το μάθημα και τον κωδικό του συγκεκριμένου μαθήματος που έχει επιλέξει. Ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (*Relational Database Management System RDBMS*) διαχειρίζεται όλα τα δεδομένα σε πίνακες. Στους πίνακες αποθηκεύονται πληροφορίες για κάποιο θέμα (όπως διδάσκοντες – μαθήματα). Πιο συγκεκριμένα, οι στήλες κάθε πίνακα περιέχουν τα διαφορετικά είδη των πληροφοριών για αυτό το θέμα (για παράδειγμα, αν έχει διδακτορικό ο διδάσκοντας ή σε ποιο εξάμηνο ανήκει το μάθημα). Ακόμα και όταν ανακτάμε πληροφορίες από έναν ή περισσότερους πίνακες (με τη βοήθεια ερωτημάτων) το αποτέλεσμα είναι ένας άλλος πίνακας.

Κάθε σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) παρέχει πλήρη έλεγχο σε ότι αφορά τον τρόπο με τον οποίο ορίζουμε τα δεδομένα, δουλεύουμε με αυτά και τον τρόπο με τον οποίο τα μοιραζόμαστε με τους άλλους. Το σύστημα μας εφοδιάζει με δυνατότητες που κάνουν εύκολη την καταγραφή και των χειρισμό μεγάλων ποσοτήτων δεδομένων σε πολλούς πίνακες.

Οι ΒΔ για την ανάκτηση – εισαγωγή - ενημέρωση – διαγραφή δεδομένων χρησιμοποιούν μια γλώσσα βάσεων δεδομένων που ονομάζεται SQL (Structured Query Language, Δομημένη Γλώσσα Ερωτημάτων). Η SQL αναπτύχθηκε στα πλαίσια ενός ερευνητικού προγράμματος της IBM το 1970, και υιοθετήθηκε ως επίσημο πρότυπο για σχεσιακές ΒΔ από οργανισμούς σαν το Αμερικάνικο Ίδρυμα Εθνικών Προτύπων (American National Standards Institute ANSI) και ο Διεθνής Οργανισμός Προτύπων (International Standards Organization ISO). Αν και υπάρχει τυποποίηση για την SQL, η πραγματικότητα είναι ότι οι περισσότερες εταιρείες λογισμικού έχουν χρησιμοποιήσει παραλλαγές ή επεκτάσεις της γλώσσας για να μπορεί να συνεργάζεται με τα ιδιαίτερα χαρακτηριστικά των προϊόντων τους. Επίσης κάποια προϊόντα εξελίχθηκαν πριν εγκαθιδρυθούν τα πρότυπα, και έτσι οι εταιρείες που παράγουν αυτά τα προϊόντα εφηύραν μια δική τους σύνταξη, η οποία διαφέρει από το επίσημο πρότυπο. Μια εντολή SQL που πρόκειται να εκτελεστεί από την εφαρμογή Microsoft SQL Server μπορεί να χρειαστεί κάποια τροποποίηση για να εκτελεστεί από άλλες ΒΔ που υποστηρίζουν την SQL όπως η Oracle.

Για να λύσει αυτό το πρόβλημα ένας μεγάλος όμιλος σημαντικών εταιρειών παραγωγής υπολογιστών και λογισμικού – στον οποίο συμμετείχαν περισσότερες από 30 εταιρείες μεταξύ

των οποίων και η Microsoft Corporation – σχημάτισε πριν από αρκετά χρόνια την ομάδα SQL Access Group. Σκοπός τους ήταν να ορίσουν μια υλοποίηση SQL που θα ήταν κοινά αποδεκτή, και θα μπορούσαν να τη χρησιμοποιούν όλοι για να επικοινωνούν μεταξύ τους. Έτσι, αυτές οι εταιρείες ανέπτυξαν από κοινού τη διασύνδεση κοινής γλώσσας (Common Language Interface, CLI) για τις πιο συνηθισμένες μορφές της SQL, και αφιερώθηκαν στην υποστήριξη των προϊόντων τους που θα επέτρεπε σε οποιαδήποτε εφαρμογή που χρησιμοποιούσε τη CLI να μπορεί να επεξεργάζεται τις ΒΔ τους. Αρκετές από αυτές τις εταιρείες παρουσίασαν από κοινού αυτή τη δυνατότητα στις αρχές του 1992. Στο μεταξύ, η Microsoft επισημοποίησε τη CLI για τους σταθμούς εργασίας, και ανακοίνωσε ότι όλα τα προϊόντα της θα χρησιμοποιούσαν αυτή τη διασύνδεση για να έχουν πρόσβαση σε οποιαδήποτε ΒΔ SQL. Η Microsoft ονόμασε αυτή την επισημοποιημένη διασύνδεση πρότυπο Ανοιχτής Συνδεσιμότητας Βάσεων Δεδομένων, ΑΣΒΔ (Open Database Connectivity ODBC) [5]. Παρακάτω θα περιγραφεί αναλυτικότερα αυτό το πρότυπο (ODBC) όπου χρησιμοποιήσαμε στην εφαρμογή μας.

Η MS Access 2000 είναι το DBMS το οποίο διαχειρίζεται τη ΒΔ. Μας δίνει τη δυνατότητα να δημιουργήσουμε Πίνακες αποθήκευσης δεδομένων, Ερωτήματα επεξεργασίας δεδομένων, Φόρμες ενημέρωσης δεδομένων, Εκθέσεις δηλαδή εκτύπωση δεδομένων, Σελίδες πρόσβασης στη ΒΔ από το Διαδίκτυο, Μακροεντολές που αυτοματοποιούν κοινές εργασίες και τέλος Λειτουργικές Μονάδες όπου είναι συλλογή δηλώσεων και διαδικασιών της Visual Basic.

Έτσι λοιπόν ενώ είχαμε τη δυνατότητα με Φόρμες να ενημερώνουμε τους πίνακες των δεδομένων μας, με Λειτουργικές Μονάδες να υλοποιήσουμε τους αλγορίθμους ΩΠΠ και ΩΠΕ καθώς και με Εκθέσεις να τυπώνουμε τα δεδομένα μας δεν χρησιμοποιήσαμε την Access 2000 για την υλοποίηση της εφαρμογής μας. Χρησιμοποιήσαμε μόνο τους Πίνακες, για αποθήκευση των δεδομένων μας, λόγω της απαιτητικότητας που υπήρχε σε ταχύτητα και πολυπλοκότητα. Επίσης η εφαρμογή που δημιουργήσαμε είναι ένα .exe αρχείο, το οποίο με τις ανάλογες βιβλιοθήκες DLL μπορεί να τρέχει σε οποιοδήποτε μηχάνημα με λειτουργικό σύστημα Windows. Η ΒΔ δεν είναι απαραίτητο να είναι σε Access 2000, και ούτε είναι απαραίτητο να υπάρχει η Access 2000 άμα αποθηκεύουμε τη ΒΔ σε άλλο σύστημα RDBMS. Το πρότυπο ODBC, που αναλύεται παρακάτω, μας δίνει τη δυνατότητα η ΒΔ μας να χειρίζεται από ένα άλλο σύστημα RDBMS όπως για παράδειγμα έναν SQL Server ή Oracle.

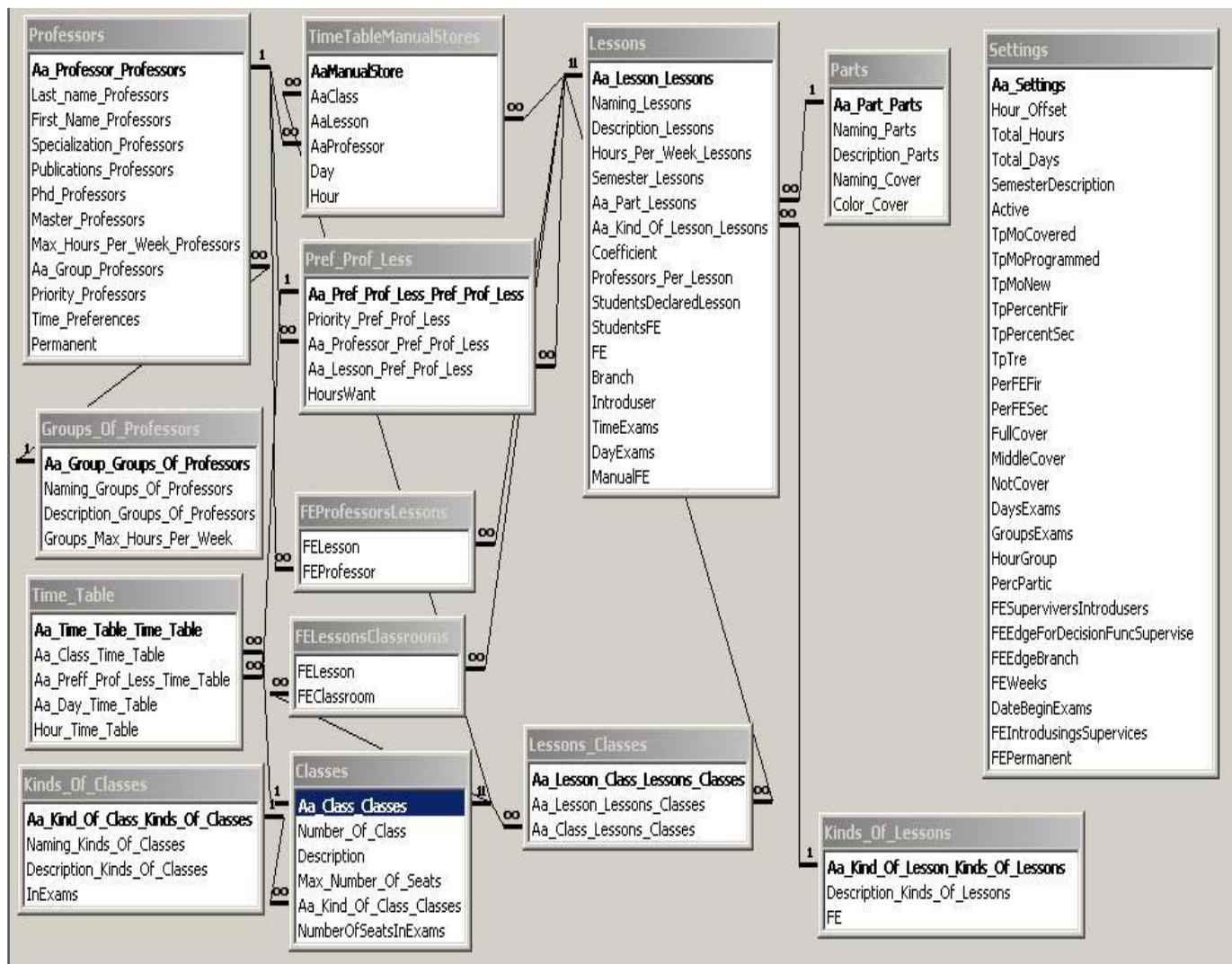
Αα	Πίνακας	Επεξήγηση
1.	Professors	Δεδομένα καθηγητών ανεξαρτήτως σχολής
2.	Lessons	Μαθήματα όλων των σχολών
3.	TimeTableManualStores	Χειροκίνητες καταχωρήσεις ΩΠΠ
4.	Settings	Ρυθμίσεις εφαρμογής
5.	Pref_Prof_Less	Προτιμήσεις μαθημάτων κάθε διδάσκοντα
6.	Groups_Of_Professors	Κατηγορίες διδασκόντων
7.	Time_Table	ΩΠΠ
8.	Kinds_Of_Classes	Κατηγορίες χώρων διδασκαλίας
9.	FEProfessorsLessons	Για ΩΠΕ επιτηρητές για κάθε μάθημα
10.	FELessonsClassrooms	Για ΩΠΕ χώρος εξέτασης μαθημάτων
11.	Classes	Χώροι διδασκαλίας Ιδρύματος
12.	Lessons_Classes	Το κάθε μάθημα που δύναται να διδαχθεί
13.	Kinds_Of_Lessons	Είδη μαθημάτων

Πίνακας 2.1. Χρησιμότητα κάθε πίνακα της ΒΔ *Timetable.mdb*

Το E-R Model (Entity Relationship Model) δείχνει γραφικά τις συσχετίσεις των πινάκων

μιας ΒΔ. Το E-R Model της ΒΔ μας δηλαδή της Timetable.mdb φαίνεται στο σχήμα 2.1. Παρατηρούμε ότι αποτελείται από τους πίνακες Professors, Lessons, TimeTableManualStores, Parts, Settings, Pref_Prof_Less, Groups_Of_Professors, Time_Table, Kinds_Of_Classes, FEProfessorsLessons, FELessonsClassrooms, Classes, Lessons_Classes, Kinds_Of_Lessons όπου στον πίνακα 2.1. επεξηγείται η χρησιμότητα του κάθε πίνακα της ΒΔ Timetable.mdb περιληπτικά.

Θα παρατηρήσετε επίσης ότι οι πίνακες μεταξύ τους ενώνονται με γραμμές οι οποίες στην μια άκρη τους έχουν το σύμβολο 1 και στην άλλη άκρη το σύμβολο ∞. Αυτό σημαίνει ότι έχουμε σχέση 1 προς πολλά σε αυτή την περίπτωση. Για παράδειγμα ο πίνακας Groups_Of_Professors με τον πίνακα Professors έχουν σχέση 1 προς πολλά που σημαίνει ότι κάθε καθηγητής ανήκει σε μια και μόνο κατηγορία. Σε μια κατηγορία όμως μπορούν να ανήκουν πολλοί καθηγητές. Επειδή θα ήταν άσκοπο να αναλύσουμε όλα τα πεδία του κάθε πίνακα, παρακάτω επεξηγούνται τα πιο σημαντικά πεδία, καθώς και η χρησιμότητα τους.



Σχήμα 2.1. E-R Model της ΒΔ Timetable

Έτσι λοιπόν στον πίνακα Professors στο πεδίο Time_Preferences αποθηκεύονται οι προτιμήσεις, ως προς το ωράριο, του κάθε διδάσκοντα. Είναι τύπου Αλφαριθμητικού, και το μέγεθος του είναι 168 χαρακτήρες. Το 168 προέρχεται από τον πολλαπλασιασμό 7 ημέρες εβδομάδας × 24 ώρες ανά ημέρα = 168. Ο κάθε χαρακτήρας αυτού του αλφαριθμητικού περιέχει και μια προτίμηση του διδάσκοντα. Για παράδειγμα το γράμμα ‘k’ συμβολίζει την ουδέτερη προτίμηση του σε μια επιλογή. Η καταχώρηση – εύρεση προτιμήσεων γίνεται από αριστερά προς τα δεξιά. Π.χ. αν είχαμε την ακολουθία χαρακτήρων kakzkzkakzkkaa κ.ο.κ και οι ημέρες

διδασκαλίας ήταν 5 και οι ώρες διδασκαλίας ανά ημέρα ήταν 3, για να αποκωδικοποιούσαμε αυτή την πληροφορία θα λέγαμε ότι ο διδάσκοντας την πρώτη ημέρα και πρώτη ώρα έχει προτίμηση ουδέτερη (πρώτος χαρακτήρας k), την δεύτερη ημέρα και πρώτη ώρα έντονη επιθυμία (δεύτερος χαρακτήρας a), την πρώτη ημέρα και δεύτερη ώρα αρνητική προτίμηση (έκτος χαρακτήρας z). Επίσης το πεδίο Max_Hours_Per_Week_Professors στον πίνακα Professors δηλώνει τις επιθυμητές ώρες διδασκαλίας που καταχωρούνται χειροκίνητα από το χρήστη.

Ο πίνακας Pref_Prof_Less, δείχνει τα μαθήματα που επιθυμούν να διδάξουν οι διδάσκοντες. Εδώ υπάρχει το πεδίο Hours_Want το οποίο υποδηλώνει για κάθε προτίμηση του κάθε καθηγητή τις ώρες που επιθυμεί να διδάξει. Η άλλη χρησιμότητα αυτού του πίνακα είναι ότι μέσω των προτιμήσεων των μαθημάτων κάθε καθηγητή, μπορούμε να βρούμε πολύ εύκολα σε πιο (ή σε πια) τμήμα ανήκει κάθε καθηγητής, και πόσες ώρες διδάσκει στο καθένα. Στον πίνακα Classes το πεδίο Max_Number_Of_Seats αποθηκεύεται το σύνολο των θέσεων κάθε χώρου διδασκαλίας. Το πεδίο NumberOfSeatsInExams δηλώνει την χωρητικότητα κάθε χώρου διδασκαλίας σε περίπτωση που συμμετέχει στις εξετάσεις. Για το αν ένας χώρος διδασκαλίας είναι και χώρος εξέτασης (την περίοδο των εξετάσεων) φαίνεται από το είδος του κάθε χώρου διδασκαλίας του πίνακα KindsOfClasses στο πεδίο InExams στο οποίο είναι τύπου Boolean. Αν είναι TRUE αυτή η τιμή τότε όλοι οι χώροι διδασκαλίας που ανήκουν σε αυτή την κατηγορία είναι χώροι εξέτασης.

Στον πίνακα Lessons το πεδίο StudentsDeclaredLesson δηλώνει τον αριθμό των φοιτητών που έχει δηλώσει το συγκεκριμένο μάθημα. Το πεδίο StudentsFE δηλώνει τους αναμενόμενους φοιτητές για τις εξετάσεις. Ο αλγόριθμος του ΩΠΕ λαμβάνει υπόψη του για κάθε μάθημα την τιμή στο πεδίο StudentsFE. Τα πεδία Introducer, TimeExams, DayExams είναι τα στοιχεία καταχώρησης προγραμματισμού κατά τη διάρκεια των εξετάσεων δηλαδή ποιος είναι ο εισηγητής, ποια μέρα θα γίνει η εξέταση και ποια ώρα. Το πεδίο FE είναι τύπου BOOLEAN και δηλώνει αν το μάθημα συμμετέχει ή όχι στις εξετάσεις. Επίσης το πεδίο ManualFE δηλώνει αν καταχωρήθηκε χειροκίνητα ή όχι το συγκεκριμένο μάθημα για εξέταση. Το πεδίο Branch δηλώνει για το κάθε μάθημα σε πιο κλάδο ανήκει.

b. Πρότυπο ODBC

Το ODBC (Open Database Connectivity) όπως προαναφέραμε είναι πρότυπο Ανοιχτής Συνδεσιμότητας Βάσεων Δεδομένων, (ΑΣΒΔ) [4]. Αυτό σημαίνει ότι σχεδιάστηκε για να προσφέρει πρόσβαση σε δεδομένα που βρίσκονται σε δίκτυα οργανισμών – επιχειρήσεων. Επίσης υποστηρίζει την πρόσβαση σε σχεσιακές ΒΔ.

Με τη χρήση του ODBC μας δίνεται η δυνατότητα να δημιουργήσουμε εφαρμογή και μέσω αυτής να επεξεργαζόμαστε δεδομένα που βρίσκονται σε ένα διακομιστή δικτύου SQL ή στη βάση δεδομένων SQL ενός μεγάλου συστήματος. Οπότε έχουμε ευελιξία στην επεξεργασία δεδομένων, δεν έχουμε πρόβλημα συμβατότητας, και το πιο σημαντικό μπορούμε να προσπελάσουμε δεδομένα διαφορετικών συστημάτων χωρίς να χρειαστεί να αλλάξουμε κάποια φόρμα ή αναφορά.

Για την εφαρμογή μας χρησιμοποιούμε την ΒΔ Timetable.mdb. Επίσης χρησιμοποιούμε το πρότυπο ODBC για να επικοινωνήσει η εφαρμογή μας με τη ΒΔ. Για να επικοινωνήσει όμως η εφαρμογή μας με τη ΒΔ μέσω του προτύπου ODBC πρέπει να “δημοσιοποιήσουμε” την ΒΔ, ώστε να μετατρέψει το ΛΣ τη ΒΔ από τη μορφή που είναι με βάση τους drivers της MSAccess2000 που έχει στην απαιτούμενη μορφή που χρειάζεται για αυτό το σκοπό. Τα βήματα που είναι απαραίτητα για αυτή την υλοποίηση φαίνονται στο Σχήμα 2.2. και είναι τα ακόλουθα:

BHMA 1ο Ανοίγουμε τον πίνακα ελέγχου από το Start->Settings->Control

Panel.

BHMA 2ο Επιλέγουμε Administrative Tools.

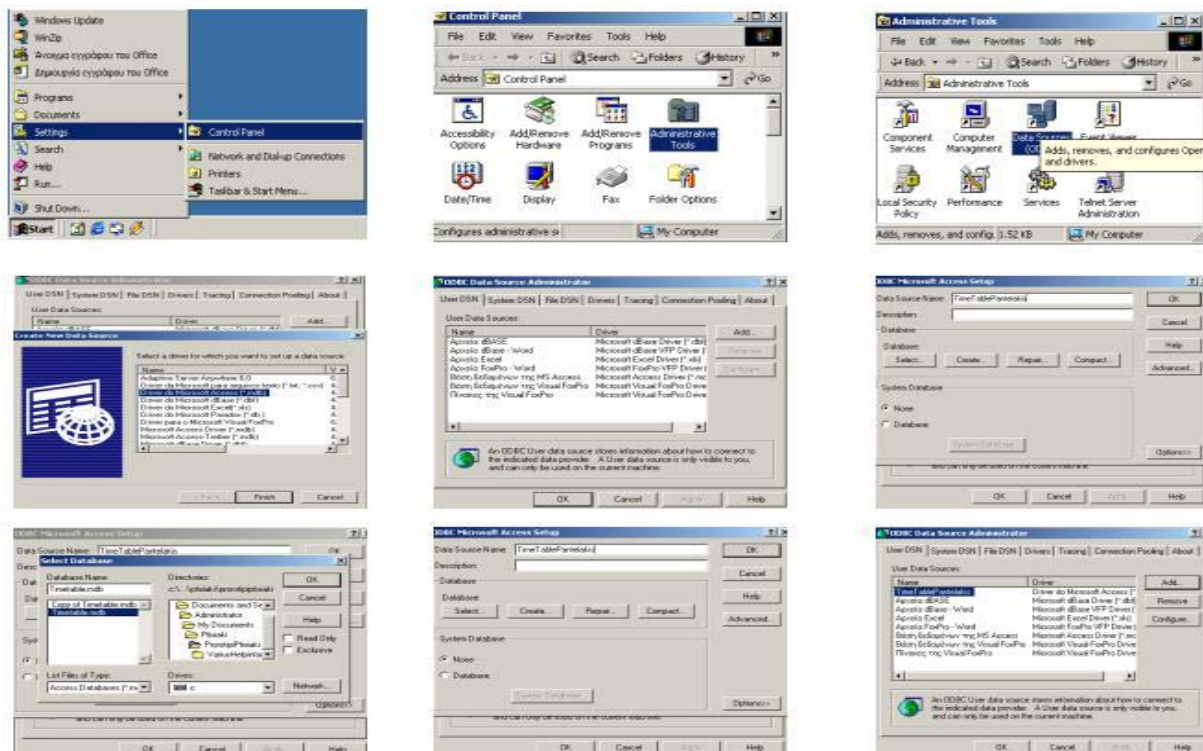
BHMA 3ο Στη συνέχεια επιλέγουμε DataSources(ODBC).

BHMA 4ο Στη συνέχεια από την καρτέλα που εμφανίζεται επιλέγουμε User DSN και πατάμε το πλήκτρο δεξιά ADD.

BHMA 5ο Σε αυτό το βήμα επιλέγουμε τους drivers τους οποίους υποστηρίζει η ΒΔ από την οποία θα πάρουμε τα στοιχεία. Στην περίπτωση μας επιλέγουμε Driver do Microsoft Access (*.mdb) και πατάμε το κουμπί Finish.

BHMA 6ο Στη συνέχεια στο πλαίσιο διαλόγου που εμφανίζεται στο πάνω μέρος εκεί που λέει Data Source Name είναι το όνομα με το οποίο θα δημοσιοποιηθεί η ΒΔ μας. Στην περίπτωση μας γράφουμε TimeTablePantelakis. Μετά την πληκτρολόγηση πατάμε κάτω το πλήκτρο Select για να επιλέξουμε την προς δημοσιοποίηση ΒΔ.

BHMA 7ο Στη συνέχεια με τη βοήθεια του Explorer που εμφανίζεται εντοπίζουμε την προς δημοσιοποίηση ΒΔ που στην εφαρμογή μας είναι η Timetable, όπου εδώ αξίζει να παρατηρήσουμε την επιλογή Network δηλαδή αναζήτηση ΒΔ μέσω δικτύου, την επιλέγουμε με το ποντίκι και πατάμε Ok. Στη συνέχεια στο πλαίσιο που εμφανίζεται πάλι πατάμε το πλήκτρο Ok. Εδώ όπως βλέπετε υπάρχει πλέον το όνομα δημοσίευσης και το οποίο πλέον μπορούμε να το δούμε από οποιαδήποτε εφαρμογή υποστηρίζει το πρότυπο ODBC. Τέλος στο πλαίσιο που εμφανίζεται πατάμε πάλι Ok και τελειώνει η διαδικασία.



Σχήμα 2.2. Διαδικασία δημοσίευσης μιας Βάσης Δεδομένων στο ODBC

2. Αντικειμενοστραφής Προγραμματισμός – Visual C++ 6.0

a. Εισαγωγή

Ο αντικειμενοστραφής προγραμματισμός όπως γνωρίζουμε είναι η εξέλιξη στον τομέα του προγραμματισμού. Η Visual C++ 6.0 είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, με μεγάλες δυνατότητες. Δίνει στον προγραμματιστή πολλά εργαλεία έτοιμα προς χρήση, που παλαιότερα θα ήταν αδιανόητο ακόμα και η δημιουργία τέτοιων στοιχείων χειρισμού. Αυτό είναι εντονότερο στον τομέα των γραφικών που όπως θα διαπιστώσει ο χρήστης εφαρμογής είναι ένα πολύ φιλικό περιβάλλον. Σε αυτό το κεφάλαιο θα επισημάνουμε κρίσιμα σημεία που βοήθησαν στην υλοποίηση της εφαρμογής του χρονοπρογραμματισμού. Έτσι λοιπόν θα αναφερθούμε για τις κατηγορίες της MFC και συγκεκριμένα για την κατηγορία CList, για τις εξαιρέσεις και το ρόλο που έχουν σε μια εφαρμογή, καθώς και για τα Active X Control που χρησιμοποιήθηκαν στην εφαρμογή μας..

b. Κατηγορίες της MFC

Σε οποιοδήποτε αντικειμενοστραφές πρόγραμμα που δημιουργούμε τα αντικείμενα πρέπει να ομαδοποιούνται και να αποθηκεύονται σε συλλογές διαφορετικών τύπων και διαστάσεων. Κάθε αντικείμενο σε μια διάταξη έχει μια θέση ή δείκτη με αρχή στο μηδέν, που χρησιμοποιείται για τον εντοπισμό και την αναφορά στο αντικείμενο. Οι λίστες είναι χρήσιμες όταν θέλουμε να θεωρήσουμε τα δεδομένα μας σε σειριακή διάταξη. Μπορούμε να διασχίσουμε τη λίστα προς τα εμπρός ή αντιστρόφως από ένα στοιχείο στο επόμενο.

Υπάρχουν γενικές κατηγορίες αντιστοιχίας όπως είναι οι : CmapWordToOb απορρέουσα από την CObject, CmapWordToPtr δείκτης προς τη μνήμη void* κ.α. Η Visual C++ 6.0 μας δίνει τη δυνατότητα να δημιουργήσουμε προσαρμοσμένες κατηγορίες συλλογής δεδομένων, όπου έχουμε πολλά θετικά. Ένα από αυτά είναι εάν επιχειρήσουμε να καταχωρήσουμε λάθος αντικείμενο σε μια προσαρμοσμένη διάταξη, λίστα, ή αντιστοιχία, τότε ο μεταγλωττιστής θα εισάγει ένα μήνυμα λάθους για να μας ειδοποιήσει. Το άλλο πλεονέκτημα είναι ότι δεν χρειάζεται να αλλάξουμε τον τύπο στους γενικούς δείκτες CObject*(δηλαδή από ένα CobArray), ξανά στο συγκεκριμένο αντικείμενο μας για να το χρησιμοποιήσουμε.

Για να χρησιμοποιήσουμε τα πρότυπα στο πρόγραμμα μας θα πρέπει να συμπεριλάβουμε την ακόλουθη γραμμή επικεφαλίδας

```
#include "afxtempl.h"
```

Κατόπιν μπορούμε να ορίσουμε την δική μας κατηγορία ασφαλούς τύπου Clist, την οποία χρησιμοποιήσαμε στην εφαρμογή μας, π.χ.

```
Clist<CVT_Professor*,CVT_Professor*>m_ListProfessors;
```

Τα σύμβολα<και>είναι αγκύλες γωνίας και όχι τελεστές μικρότερο ή μεγαλύτερο. Η πρώτη παράμετρος CVT_Professor* καθορίζει τους τύπους των αντικειμένων των δεικτών που επιστρέφει η διάταξη, όταν χρησιμοποιούμε την GetAt() ή άλλες συναρτήσεις πρόσβασης. Η δεύτερη CVT_Professor*>καθορίζει τι τύπου είναι τα δεδομένα εισόδου. Τέλος το m_ListProfessors είναι το όνομα της διάταξης λίστας.

c. Εξαιρέσεις

Μια εξαίρεση είναι ένα αντικείμενο που έχει πληροφορίες για μια διαδικασία που δεν λειτούργησε σωστά. Το ευνοϊκό σημείο των εξαιρέσεων είναι ότι μπορούμε να δημιουργήσουμε μια εξαίρεση όταν κάτι δεν λειτουργήσει σωστά σε μια συνάρτηση χαμηλού επιπέδου και να την ορίσουμε να καλέσει αυτομάτως μια συνάρτηση που μπορεί να ασχοληθεί με όλες αυτές τις εξαιρέσεις. Είναι σχεδόν αδύνατο να δημιουργηθεί μια εφαρμογή χωρίς να υπάρξουν λάθη, είτε από τον προγραμματιστή, είτε από το ίδιο το σύστημα. Επίσης είναι πολύ δύσκολο να προβλέψεις όλες τις περιπτώσεις των λαθών κάθε μηχανήματος. Έτσι αν προσθέσουμε τις δυο ακόλουθες γραμμές σε οποιονδήποτε κώδικα, θα ανιχνεύσουμε λάθη τα οποία μπορεί να δημιουργηθούν και θα είναι δυσάρεστα για το χρήστη.

Στο παράδειγμα αυτό έχουμε μια μεταβλητή Professor τύπου CVT_Professor . Η μεταβλητή αυτή παίρνει την τιμή του πρώτου δεδομένου της λίστας m_ListProfessors. Αν η λίστα είναι άδεια, τότε η εντολή m_ListProfessors.GetHead() μας επιστρέφει NULL, οπότε στην επόμενη γραμμή κώδικα αν Professor είναι NULL, και πάμε να βάλουμε δεδομένα, το σύστημα θα ανιχνεύσει το

```
try
{
    CVT_Professor *Professor;
    Professor=m_ListProfessors.GetHead();
    Professor.SetAvailable(3,4,TRUE);
}
catch(...)
{
    MesssageBox("Πρόβλημα στη συνάρτηση...");
}
```

λάθος και θα πάει στο catch Block και θα μας ενημερώσει. Η εντολή catch(...) ενημερώνεται από το σύστημα για όλων των ειδών τις εξαιρέσεις. Αν θέλουμε να ανιχνεύσουμε συγκεκριμένες εξαιρέσεις π.χ. για άνοιγμα αρχείων τότε χρησιμοποιούμε την κάτωθι σύνταξη

```
catch(CfileException *e)
{
    if(e->m_cause==CfileException::fileNotFound)
        AfxMessageBox("Ξεχάσαμε να ανοίξουμε το αρχείο");
    e->Delete();
}
```

Η τελευταία εντολή λέει στο σύστημα ότι αντιμετωπίσαμε την εξαίρεση επιτυχώς και δεν υπάρχει λόγος να καταρρεύσει. Για περισσότερες πληροφορίες σχετικά με τις εξαιρέσεις βλέπε [1] σελ. 294,[2].

d. Active X Control

Τα Active X Control είναι έτοιμα στοιχεία ελέγχου τα οποία μπορούμε να χρησιμοποιήσουμε και έτσι να απλοποιήσουμε την εφαρμογή μας. Με αυτό τον τρόπο δεν σπαταλάμε χρόνο άδικα για δημιουργία στοιχείων ελέγχου γραφικών ή όχι που θα διευκόλυναν τον χρήστη ή ακόμα και εμάς. Στην εφαρμογή μας κάναμε πολύ εκτενή χρήση του MSFlexGrid Active X Control. Αυτό το στοιχείο ελέγχου είναι ένας πίνακας δεδομένων διαστάσεων $M \times N$, του οποίου μπορούμε να αλλάξουμε τη διάσταση, τον αριθμό των στηλών και των γραμμών που περιέχει, καθώς και να επέμβουμε σε κάθε κελί ξεχωριστά και να του αλλάξουμε το χρώμα, τη γραμματοσειρά, κ.ο.κ. Αυτό το στοιχείο το χρειαστήκαμε για τη γραφική απεικόνιση των προτιμήσεων ως προς το ωράριο για τους καθηγητές καθώς και για απεικόνιση δεδομένων σε

πίνακες. Επίσης υπάρχουν και άλλα πολλά στοιχεία ελέγχου για ΒΔ, Πολυμέσα κ.α. Η Visual C++ 6.0 δίνει στο χρήστη τη δυνατότητα να δημιουργήσει τα δικά του στοιχεία Active X, αλλά και να επέμβει στα ήδη υπάρχοντα και να τα προσαρμόσει στις προτιμήσεις του. Αυτό γίνεται επειδή κάθε φορά που εισάγεται σε μια εφαρμογή ένα στοιχείο ελέγχου, αυτόματα εισάγονται και τα αρχεία που υποστηρίζουν τις διάφορες λειτουργίες του.

3. Σχεδίαση Αλγορίθμου Ωρολογίου Προγράμματος Παραδόσεων

a. Περιγραφή

Για την εύρεση του ωρολογίου προγράμματος παραδόσεων των μαθημάτων μιας Σχολής υπάρχουν κάποιοι περιορισμοί ώστε να λειτουργεί σωστά το σύστημα διδασκαλίας. Οι συνθήκες που πρέπει να ικανοποιούνται όταν προγραμματίζεται ένα μάθημα είναι οι ακόλουθες:

- Ο διδάσκων που το έχει επιλέξει να μπορεί να το διδάξει και να μην έχει δηλώσει αρνητική προτίμηση εκείνη τη χρονική στιγμή.
- Το σύνολο των διδασκόντων μια συγκεκριμένη χρονική στιγμή να μην ξεπερνάει το μέγιστο επιτρεπόμενο αριθμό διδασκόντων που χρειάζονται για το μάθημα.
- Ο χώρος διδασκαλίας στον οποίο προγραμματίστηκε να διδαχθεί το μάθημα, να έχει δηλωθεί ότι μπορεί να γίνει εκεί π.χ. το μάθημα Τεχνολογία Τυπωμένων Κυκλωμάτων και Κατασκευές διδάσκεται στο Εργαστήριο 6 που είναι το Εργαστήριο Τυπωμένων Κυκλωμάτων, και δεν μπορεί να διδαχθεί σε άλλο χώρο π.χ. Εργαστήριο 2 που είναι το Εργαστήριο μ/Υ.
- Την ημέρα και ώρα που προγραμματίστηκε ένα μάθημα για διδασκαλία δεν πρέπει να προγραμματιστεί μάθημα ίδιου τμήματος και εξαμήνου παρά μόνο όταν έχουν τον ίδιο αριθμό τμημάτων, και ο αριθμός των τμημάτων είναι μεγαλύτερος ή ίσος του αριθμού των μαθημάτων.
- Κανένα μάθημα να μην έχει υπερβαίνει τον προβλεπόμενο αριθμό ωρών..
- Ο διδάσκων που έχει προγραμματιστεί για να διδάξει ένα μάθημα να μην έχει προγραμματιστεί ταυτόχρονα για άλλο μάθημα.
- Επίσης πρέπει να επισημανθεί ότι επιδιώκουμε το κάθε μάθημα να είναι συνεχόμενο δίωρο ή τρίωρο και όχι σπαστό ανά μια ώρα.

Λαμβάνοντας υπόψη τους παραπάνω περιορισμούς καταλήξαμε στον παρακάτω αλγόριθμο ο οποίος μας δίνει μια ικανοποιητική λύση στο πρόβλημα μας.

b. Περιγραφή του αλγορίθμου του προγράμματος παραδόσεων

Δημιουργεί μια ταξινομημένη λίστα με τους διδάσκοντες, η οποία είναι σε αύξουσα διάταξη ως προς την προτεραιότητα που έχει δοθεί στον κάθε Διδάσκοντα. Η προτεραιότητα που έχει δοθεί προέρχεται στην περίπτωση μας από τη βαθμίδα που έχει ο κάθε Διδάσκοντας στη Σχολή, και η τιμή που μπορεί να πάρει είναι από το 1 έως το 255. Για παράδειγμα μπορούμε να θέσουμε προτεραιότητα 1 στον Πρόεδρο του Τμήματος και προτεραιότητα 15 σε κάθε Εργαστηριακό Συνεργάτη. Οι προτεραιότητες είναι μεταβλητές και μπορούν να αλλάξουν από τον χρήστη του προγράμματος. Επιλέγοντας κάθε διδάσκοντα από τη λίστα όπου ελέγχουμε αν έχει υπόλοιπο ωρών για διδασκαλία από τις “επιθυμητές” του ώρες στο σύνολο των ωρών που επιθυμεί να διδάξει. Επίσης όμως πρέπει να έχει υπόλοιπο ωρών και για κάθε μάθημα που επιθυμεί ξεχωριστά με τη σειρά που αναφέραμε. Στη συνέχεια επιλέγουμε από την λίστα προτίμησης μαθημάτων, του κάθε διδάσκοντος, που είναι ταξινομημένη σε αύξουσα διάταξη ανάλογα με την σειρά προτίμησης που έχει δώσει σε κάθε μάθημα, ένα μάθημα κάθε φορά και κάνουμε τις παρακάτω ενέργειες:

Ελέγχουμε αν το μάθημα έχει υπολειπόμενες ώρες για διδασκαλία και αν έχει συνεχίζει αλλιώς παίρνει το επόμενο μάθημα από τις επιλογές του διδάσκοντος ή τον επόμενο διδάσκοντα, αν έχουν τελειώσει τα μαθήματα του τρέχοντος εκπαιδευτικού.

Με το μάθημα επιλογής του διδάσκοντα που έχουμε δημιουργούμε 3 πίνακες σύγκρισης διαστάσεων (Total_Hours_Day × Total_Days_Week)(ΩΠΕΣ ANA ΗΜΕΡΑ × ΜΕΠΕΣ ANA

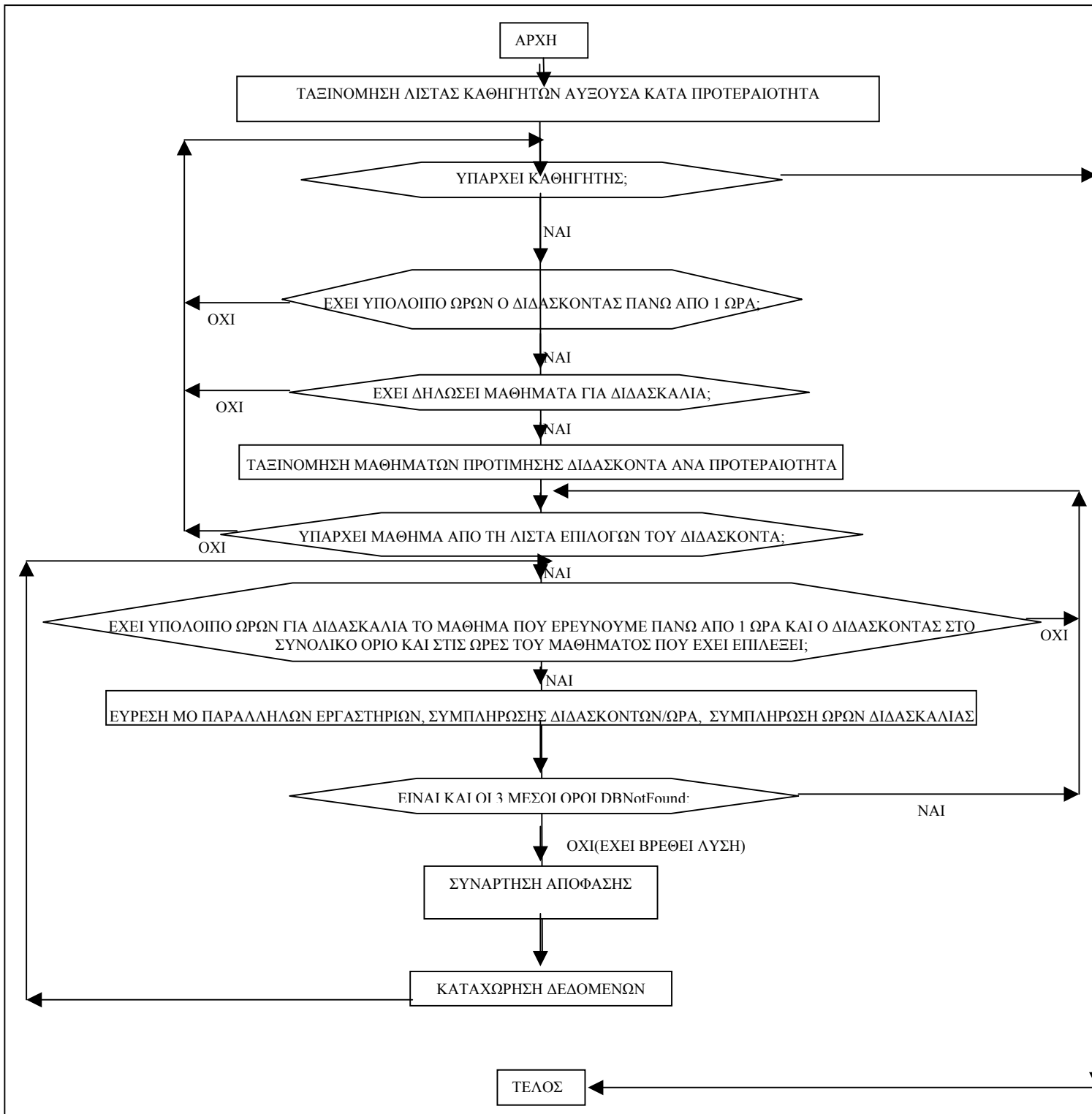
ΕΒΔΟΜΑΔΑ). Ο πρώτος πίνακας ονομάζεται **TotalMatchingTableSameLesson** όπου εδώ όταν “βλέπουμε” διαθέσιμη ώρα σημαίνει ότι έχει προγραμματιστεί ήδη το μάθημα που ερευνούμε αλλά δεν έχει συμπληρώσει τον κατάλληλο αριθμό των διδασκόντων που πρέπει να το διδάξουν. Ο δεύτερος πίνακας ονομάζεται **TotalMatchingTableSamePartSemester** όπου εδώ όταν “βλέπουμε” διαθέσιμη ώρα σημαίνει ότι έχει προγραμματιστεί μάθημα ίδιου τμήματος, ίδιου εξαμήνου, και με τον ίδιο αριθμό τμημάτων αλλά δεν έχει συμπληρώσει τον αριθμό των αλληλοεπικαλυπτόμενων μαθημάτων. Ο τρίτος πίνακας ονομάζεται **TotalMatchingTableNew** όπου εδώ όταν “βλέπουμε” διαθέσιμη ώρα σημαίνει ότι δεν έχει προγραμματιστεί καθόλου το τρέχον μάθημα που ερευνούμε, δεν έχει προγραμματιστεί καθόλου μάθημα ίδιου εξαμήνου και τμήματος. Οι παραπάνω πίνακες σύγκρισης όπου είναι διαθέσιμοι σημαίνει ότι ο διδάσκοντας μπορεί να διδάξει (όπου εδώ φαίνεται και το ποσοστό επιθυμίας που έχει δηλώσει ο διδάσκοντας) και ότι υπάρχουν χώροι που έχουν δηλωθεί για χώροι διεξαγωγής των μαθημάτων και είναι διαθέσιμοι (τουλάχιστον ένας). Ο τελευταίος περιορισμός ισχύει μόνο για τους δυο τελευταίους πίνακες σύγκρισης, αφού ο πρώτος που αναφέραμε έχει προγραμματιστεί ήδη το μάθημα και θα προγραμματιστεί ένας επιπλέον διδάσκοντας στην ήδη προγραμματισμένη ώρα.

Συνεχίζοντας με τα στοιχεία που έχουμε ο αλγόριθμος καλεί 3 συναρτήσεις.

Η πρώτη συνάρτηση που καλεί είναι η **RunAlgorithmProgrammed(&TempDayLesson,&TempHourLesson)** η οποία συνάρτηση δέχεται δυο ορίσματα που είναι οι διευθύνσεις σε μεταβλητές τύπου BYTE και οι οποίες θα αλλάξουν μέσα στη συνάρτηση αν βρεθεί συνεχόμενο δώρο στον πίνακα **TotalMatchingTableSameLesson** και επιστρέφει τον μέσο όρο προτίμησης εκείνου του ωραρίου που βρήκε. Η δεύτερη συνάρτηση που καλεί είναι η **RunAlgorithmCovered(&TempDayCovered, &TempHourCovered, &Temp_Class_Covered)** η οποία συνάρτηση δέχεται τρία ορίσματα που είναι οι διευθύνσεις σε μεταβλητές τύπου BYTE και οι οποίες θα αλλάξουν μέσα στη συνάρτηση αν βρεθεί συνεχόμενο δώρο στον πίνακα **TotalMatchingTableSamePartSemester** και το τρίτο όρισμα είναι διεύθυνση αντικειμένου της τάξης **CVT_Classroom** που ενημερώνεται από τη συνάρτηση για το ποια αίθουσα βρήκε συνεχόμενης διαθεσιμότητας για να γίνει το μάθημα. Όπως και η προηγούμενη συνάρτηση, επιστρέφει τον μέσο όρο προτίμησης εκείνου του ωραρίου που βρήκε. Η τρίτη και τελευταία συνάρτηση που καλεί ο αλγόριθμος είναι η **RunAlgorithmNew(&TempDayNew, &TempHourNew, &Temp_Class_New,&ThreeHoursLesson)** η οποία συνάρτηση δέχεται τέσσερα ορίσματα που είναι οι διευθύνσεις σε μεταβλητές τύπου BYTE και οι οποίες θα αλλάξουν μέσα στη συνάρτηση αν βρεθεί συνεχόμενο δώρο ή τρίωρο στον πίνακα **TotalMatchingTableNew**, το τρίτο όρισμα είναι διεύθυνση αντικειμένου της τάξης **CVT_Classroom** που ενημερώνεται από τη συνάρτηση για το ποια αίθουσα βρήκε συνεχόμενης διαθεσιμότητας για να γίνει το μάθημα, και το τέταρτο όρισμα είναι μια διεύθυνση σε μεταβλητή BOOLEAN η οποία αλλάζει μέσα στη συνάρτηση και αν είναι TRUE σημαίνει ότι βρήκε συνεχόμενο τρίωρο. Στη συνέχεια παίρνοντας του 3 μέσους όρους από τις συναρτήσεις που αναφέραμε ξεκινάει ο αλγόριθμος μια διαδικασία σύγκρισης ώστε να επιλέξει την καλύτερη λύση σύμφωνα με τις ρυθμίσεις που έχει κάνει ο χρήστης από το μενού **Ρυθμίσεις-> Ρυθμίσεις Αλγορίθμου Προγράμματος** για να αποφασίσει ποια λύση να επιλέξει. Όποιος μέσος όρος είναι η καλύτερη λύση, αυτές οι τιμές καταχωρούνται. Αν καμία από τις τρεις συναρτήσεις δεν έχει βρει καμία λύση τότε επιστρέφει την τιμή **DBDontFound** που είναι ίση με το 9 και έχει οριστεί σαν **#define** στο αρχείο **GlobalsTimeTable.h** η οποία είναι η μοναδική τιμή όταν δεν βρει λύση ο αλγόριθμος. Σε περίπτωση που δεν έχει βρει καμία λύση ή τελειώσουν οι ώρες του μαθήματος ή τελειώσουν οι ώρες του Διδάσκοντος τότε ανάλογα παίρνει το επόμενο μάθημα του διδάσκοντος ή παίρνει τον επόμενο διδάσκοντα. Παρακάτω φαίνεται το λογικό διάγραμμα του αλγορίθμου ωρολογίου προγράμματος, όπου φαίνεται αναλυτικά η λογική που ακολουθείται για την επίλυση του προβλήματος.

c. Λογικό διάγραμμα αλγορίθμου ωρολογίου προγράμματος παραδόσεων

Παρατηρώντας το λογικό διάγραμμα στο σχήμα 4.1. που ακολουθεί στηρίζεται σε εντελώς γραμμική λογική, που έχει ως αποτέλεσμα να έχουμε γραμμική λύση. Αυτό σημαίνει ότι μπορούμε να περιμένουμε ένα αποτέλεσμα, και κάθε μας ενέργεια να αναλύεται πολύ εύκολα.



Σχήμα 4.1. Αλγόριθμος ΩΠΠ.

Με διάφορες ρυθμίσεις που μπορούμε να κάνουμε από τις ρυθμίσεις αλγορίθμου ωρολογίου προγράμματος παραδόσεων, καθώς και την προτεραιότητα του κάθε διδάσκοντα, τις προτιμήσεις

ωραρίων του, την προτεραιότητα στις προτιμήσεις στα μαθήματα του, καθώς και στα μαθήματα για τους χώρους διδασκαλίας που μπορούν να διεξαχθούν, έχουμε μια προβλέψιμη λύση. Βέβαια αυτό δεν μας δίνει και την βέλτιστη λύση, που όπως καταλαβαίνετε βέλτιστη λύση είναι πολύ δύσκολο να βρεθεί, καθώς υπάρχουν πολλές αντιφατικές προτιμήσεις, περιορισμοί ακόμα και συνθήκες. Έτσι λοιπόν αν δεν είχαμε γραμμική λύση αλλά πληθώρα από λύσεις που πήγαζαν από τυχαίες τιμές όπως (στην ημέρα έναρξης της εβδομάδας, στην ώρα έναρξης διδασκαλίας, στην προτεραιότητα που παίρνει κάθε φορά τον καθηγητή, στις τυχαίες επιλογές των προτιμήσεων του ωραρίου του καθηγητή, στην τυχαία επιλογή κάθε φορά για χώρο διδασκαλίας κ.α.) πιθανόν να είχαμε μια καλύτερη λύση, αλλά αυτό θα σήμαινε ότι θα χρειαζόμασταν πολύ μεγάλη υπολογιστική ισχύ, και μεγάλη χρονική διάρκεια υπολογισμού.

4. Σχεδίαση αλγορίθμου ωρολογίου προγράμματος εξετάσεων

a. Περιγραφή

Για την εύρεση του προγράμματος εξετάσεων μιας Σχολής υπάρχουν και εδώ, όπως και στο ΩΠΠ, κάποιοι περιορισμοί ώστε να διεξαχθούν σωστά και χωρίς προβλήματα οι εξετάσεις. Οι περιορισμοί πρέπει να τηρούνται όταν προγραμματίζεται ένα μάθημα για εξέταση, αναλύονται παρακάτω:

- Ο διδάσκων που έχει προγραμματιστεί για εισηγητής ή επιτηρητής του μαθήματος να μην έχει δηλώσει αρνητική προτίμηση εκείνη τη χρονική στιγμή.
- Το σύνολο των εκπαιδευτικών που έχουν οριστεί για να επιβλέψουν την εξέταση του μαθήματος να είναι ο ακριβής αριθμός που απαιτείται για την εξέταση.
- Η χωρητικότητα των χώρων, στους οποίους προγραμματίστηκε να εξεταστεί το μάθημα, πρέπει να είναι επαρκή σε σχέση με τον αναμενόμενο αριθμό φοιτητών.
- Την ημέρα που προγραμματίστηκε ένα μάθημα για εξέταση δεν πρέπει να προγραμματιστεί μάθημα ίδιου τμήματος, εξαμήνου και κατεύθυνσης. Μαθήματα ίδιου τμήματος και εξαμήνου αλλά διαφορετικής κατεύθυνσης μπορούν να προγραμματιστούν την ίδια ημέρα και ίδια ώρα.
- Ο διδάσκων που έχει προγραμματιστεί για εισηγητή ή επιτήρηση κάποιου μαθήματος να μην έχει προγραμματιστεί ταυτόχρονα για άλλο μάθημα επιτηρητής ή εισηγητής.
- Πρέπει να εκμεταλλευτούμε όσο το δυνατόν καλύτερα τον αριθμό των αναμενόμενων φοιτητών και τις αίθουσες εξεταστικής. Δηλαδή, αν ένα μάθημα έχει φοιτητές για εξετάσεις 25 άτομα να μην προγραμματιστεί για εξέταση μια αίθουσα χωρητικότητας 60 ατόμων, αλλά σε μια αίθουσα π.χ. 25 ατόμων, αν φυσικά είναι διαθέσιμη. Ταυτόχρονα όμως πρέπει να μειώσουμε το σύνολο των αιθουσών που θα προγραμματιστούν για εξέταση, ώστε να μειωθεί η απασχόληση του εκπαιδευτικού προσωπικού που θα προγραμματιστεί για επιτήρηση.

Λαμβάνοντας υπόψη τους παραπάνω περιορισμούς καταλήξαμε στον παρακάτω αλγόριθμο ο οποίος μας δίνει μια ικανοποιητική λύση στο πρόβλημα.

b. Λογική αλγορίθμου ωρολογίου προγράμματος εξετάσεων

Η διαδικασία χωρίζεται σε δυο μέρη. Το πρώτο μέρος είναι η εύρεση των εισηγήσεων δηλαδή, το κάθε μάθημα πότε θα εξεταστεί, σε ποιες αίθουσες και ποιοι θα είναι οι εισηγητές. Το δεύτερο μέρος αποτελείται από την εύρεσης των επιτηρητών που θα επιβλέπουν το κάθε μάθημα. Αξίζει να σημειωθεί ότι οι προτιμήσεις του ωραρίου των διδασκόντων που συμμετέχουν στην εξεταστική ανάγονται από τις προτιμήσεις που έχουν στο ωρολόγιο πρόγραμμα διδασκαλίας. Η καινούργια προτίμηση που είναι για τις εξετάσεις, πηγάζει από την μικρότερη προτίμηση που έχει ο καθηγητής σε ένα δίωρο ή τρίωρο. Για παράδειγμα το Π2 της εξεταστικής δεσμεύει από τις 10:30-12:30. Αυτό σημαίνει ότι πρέπει να δούμε τις προτιμήσεις του διδάσκοντα στις 10:00-11:00, 11:00-12:00, 12:00-13:00. Αν είχε έντονο πράσινο στο 10:00-11:00 και πορτοκαλί στις άλλες δυο ώρες θα παίρναμε σαν καλύτερη προτίμηση το πορτοκαλί. Επίσης αν ένα από τις 3 ώρες ήταν κόκκινο θα παίρναμε σαν επιλογή το κόκκινο δηλαδή το μη διαθέσιμο. Αντίθετα π.χ. στο Π1 δεσμεύονται μόνο 2 ώρες επειδή είναι 08:00-10:00.

Στα λογικά διαγράμματα που ακολουθούν την ανάλυση του κάθε αλγορίθμου, φαίνονται οι λογικές πράξεις που γίνονται για την επίλυση του προβλήματος του προγράμματος εξετάσεων. Σκοπός τους είναι η κατανόηση των απλών πράξεων που χρειάζονται ώστε να

επιλυθεί το πρόβλημά μας. Όπως θα παρατηρήσετε, στηρίζονται σε εντελώς γραμμική λογική, χωρίς τυχαίες τιμές, που έχει ως αποτέλεσμα να έχουμε γραμμική λύση. Αυτό σημαίνει ότι μπορούμε να περιμένουμε ένα αποτέλεσμα, και κάθε μας ενέργεια να αναλύεται πολύ εύκολα. Έτσι με διάφορες ρυθμίσεις που μπορούμε να κάνουμε από τις ρυθμίσεις αλγορίθμου προγράμματος εξετάσεων, καθώς και την προτεραιότητα του κάθε διδάσκοντα, τις προτιμήσεις ωραρίων του, καθώς και την χωρητικότητα χώρων εξέτασης, έχουμε μια προβλέψιμη λύση. Βέβαια αυτό δεν μας δίνει και την βέλτιστη λύση, που όπως καταλαβαίνετε βέλτιστη λύση είναι πολύ δύσκολο να βρεθεί, καθώς υπάρχουν πολλοί παράγοντες οι οποίοι πολλές φορές είναι αντιφατικοί μεταξύ τους. Έτσι λοιπόν αν δεν είχαμε γραμμική λύση αλλά πληθώρα από λύσεις που πήγαζαν από τυχαίες τιμές όπως (στην ημέρα έναρξης εξεταστικής, στην ώρα έναρξης εξεταστικής, στην προτεραιότητα που επιλέγει κάθε φορά τον καθηγητή, στην ανακατανομή κάθε φορά μαθημάτων στη διάρκεια της εξεταστικής κ.α.) πιθανόν να είχαμε μια καλύτερη λύση, αλλά αυτό θα σήμαινε ότι θα χρειαζόμασταν πολύ μεγάλη υπολογιστική ισχύ, και μεγάλη χρονική διάρκεια υπολογισμού.

i.Πρώτο μέρος αλγορίθμου προγράμματος εξετάσεων (εύρεση εισηγήσεων)

Το πρώτο μέρος αποτελείται από δυο αλγορίθμους. Ο πρώτος αλγόριθμος σκοπό έχει να προγραμματίσει τα μαθήματα σε χώρους διδασκαλίας που έχουμε λιγότερες κενές θέσεις CLSF(Combination Less Seats Free). Ο δεύτερος αλγόριθμος σκοπό έχει να μειώσει τις αίθουσες που βρήκε ο CLSF και ονομάζεται CCLSF(Compress Combination Less Seats Free). Οι αλγόριθμοι CLSF και CCLSF αναλύονται ως εξής:

1. Αλγόριθμος CLSF (Combination Less Seats Free)

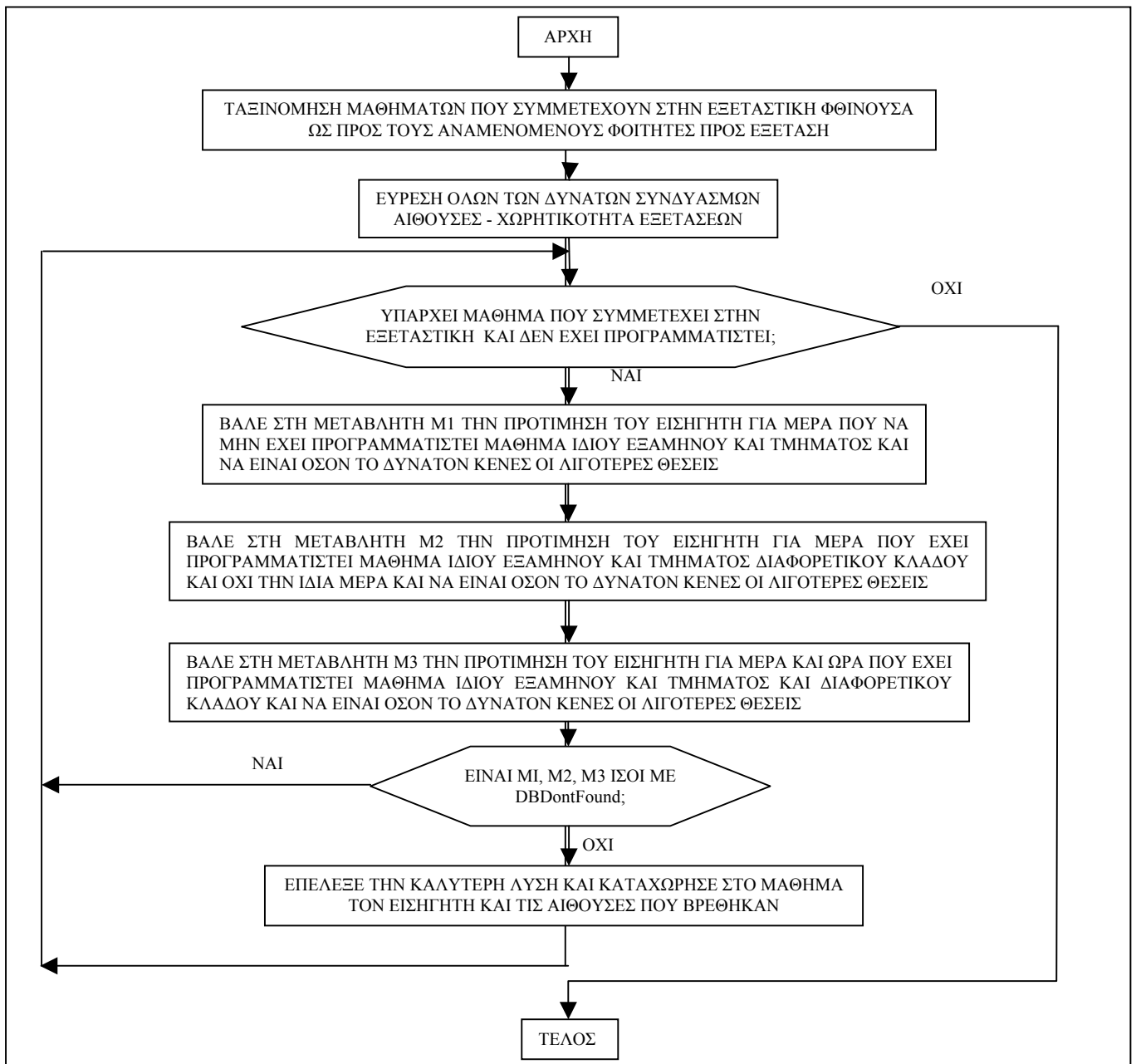
Στο ξεκίνημα του αλγορίθμου ταξινομούμε τα μαθήματα σε φθίνουσα διάταξη ως προς τους αναμενόμενους φοιτητές προς εξέταση. Αυτό γίνεται επειδή θέλουμε να έχουμε περισσότερες επιλογές για τα μαθήματα με μεγάλο αριθμό αναμενόμενων φοιτητών όσον αφορά την κάλυψη των χώρων εξέτασης. Σε αντίθετη περίπτωση αν η διάταξη ήταν αύξουσα ή τυχαία, θα υπήρχε διασπορά των κενών χώρων εξέτασης, με αποτέλεσμα μαθήματα, με πολλούς φοιτητές προς εξέταση, να μην προγραμματίζονται.

Το επόμενο βήμα είναι να βρούμε όλους τους δυνατούς συνδυασμούς (αίθουσες – χωρητικότητες χώρων εξέτασης), και να ταξινομήσουμε αυτή τη λίστα σε αύξουσα διάταξη ως προς το σύνολο των θέσεων που βρέθηκαν. Αυτό γίνεται επειδή όποτε έχουμε ένα συνδυασμό αιθουσών διαθέσιμο, συγκρίνοντας αυτή τη λίστα με το διαθέσιμο αριθμό αιθουσών και με τους αναμενόμενους φοιτητές υπό εξέταση να βρούμε εύκολα και γρήγορα ποιος συνδυασμός μας εξυπηρετεί. Σε αυτή την πράξη (όπου είναι και ο περιορισμός για τον αλγόριθμο της εξεταστικής) ο μέγιστος αριθμός χώρων διδασκαλίας προς εξέταση είναι 31, επειδή απεικονίζουμε με μεταβλητή τύπου **int** τον κάθε συνδυασμό.

Η επιλογή του συνδυασμού των χώρων εξέτασης αφού μας ικανοποιεί η χωρητικότητα στο σημείο της λίστας συνδυασμών γίνεται στην αρχή με μια λογική πράξη **OR**, μεταξύ των διαθέσιμων αιθουσών μας και των αιθουσών της λίστας συνδυασμών, και στη συνέχεια με μια πράξη **XOR** του αποτελέσματος που προέκυψε με τις διαθέσιμες αίθουσες μας. Αν δεν μας ικανοποιεί το αποτέλεσμα συνεχίζουμε μέχρι να βρούμε μια λύση ή μέχρι να τελειώσει η λίστα. Αυτό γίνεται σε κάθε μέρα και ώρα που είναι διαθέσιμος ο εισηγητής. Αν οι αναμενόμενοι φοιτητές προς εξέταση είναι περισσότεροι από το συνολικό διαθέσιμο αριθμό θέσεων των χώρων διδασκαλίας που έχουμε, τότε ο αλγόριθμος δεν μπαίνει στη διαδικασία αναζήτησης, αλλά θα προγραμματίσει αυτό το μάθημα για εξέταση όταν όλοι οι χώροι εξέτασης είναι διαθέσιμοι κάποια ημέρα, ώρα και φυσικά ο εισηγητής.

Η διαδικασία αυτή ακολουθείται σε τρεις επιμέρους αλγορίθμους (αλγόριθμος εύρεσης λύσης που να μην έχει προγραμματιστεί καθόλου μάθημα ίδιου εξαμήνου

και τμήματος, αλγόριθμος εύρεσης λύσης που να έχει προγραμματιστεί μάθημα ίδιου εξαμήνου και τμήματος, αλγόριθμος εύρεσης λύσης που να έχει προγραμματιστεί μάθημα ίδια μέρα και ώρα ίδιου εξαμήνου και τμήματος). Ο κάθε αλγόριθμος μας επιστρέφει μια προτίμηση ωραρίου του καθηγητή και τον συνδυασμό των αιθουσών που βρήκε για αυτή τη λύση, καθώς τη μέρα και ώρα. Μετά υπάρχει η συνάρτηση απόφασης που θα επιλέξει ποια λύση μας ικανοποιεί. Θα κοιτάξει πρώτα για τη λύση με τα λιγότερα κενά καθίσματα και αν υπάρχουν πάνω από μια λύσεις με λιγότερα κενά καθίσματα επιλέγει αυτήν που έχει καλύτερη προτίμηση ο εισηγητής. Στο σχήμα 5.1. που ακολουθεί φαίνεται το λογικό διάγραμμα του CLSF.



Σχήμα 5.1. Πρώτο μέρος αλγορίθμου ΩΠΕ (Αλγόριθμος CLSF)

2. Αλγόριθμος CCLSF(Compress Combination Less Seats Free)

Η λειτουργία αυτού του αλγορίθμου δεν επηρεάζει την κατανομή που έκανε ο CLSF, αλλά επιδιώκει να μειώσει τις αίθουσες που έχουν προγραμματιστεί για εξέταση

τη συγκεκριμένη ημέρα και ώρα. Έτσι ταξινομεί τα μαθήματα της συγκεκριμένης ημέρας και ώρας σε φθίνουσα διάταξη, αφαιρεί τα μαθήματα από τις προγραμματισμένες αίθουσες που βρέθηκαν με την προηγούμενη λύση και προσπαθεί για κάθε μάθημα να δεσμεύει τις μεγαλύτερες διαθέσιμες αίθουσες. Η τελευταία αίθουσα που θα χρησιμοποιήσει για αυτό το μάθημα, θα προσπαθήσει να είναι το υπόλοιπο του αριθμού των αναμενόμενων φοιτητών που δεν χωράει στις αίθουσες που έχουμε δεσμεύσει, και ταυτόχρονα θα προσπαθήσει να υπάρχουν οι λιγότερες κενές θέσεις. Έτσι έχουμε μια ανακατανομή στις αίθουσες που βρήκε ο CLSF, και έχει ως αποτέλεσμα την μείωση του απαιτούμενου προσωπικού ως προς τις επιτηρήσεις. Για παράδειγμα έστω ότι είχαμε π.χ. την Τρίτη στο Π1 τα μαθήματα Νο1, Νο2, με αντίστοιχες απαιτήσεις για φοιτητές εξετάσεων 75,23 και οι αίθουσες μας ήταν Α1, Α2, Α3, Α4, Α5 με αντίστοιχες χωρητικότητες εξετάσεων 60,40,30,17,6 και ο CLSF είχε κατανείμει στο μάθημα Νο1 τις αίθουσες Α2,Α3,Α5 και στο μάθημα Νο2 την αίθουσα Α1 δηλαδή θα χρειαζόμασταν 4 αίθουσες δηλαδή 8 επιτηρητές. Ο CCLSF με την λογική που αναφέραμε θα έπαιρνε πρώτο το μάθημα Νο1 που είναι και το μεγαλύτερο και θα του δώσει για αρχή την Α1, και στη συνέχεια βλέπουμε ότι έχει υπόλοιπο 15 άτομα. Αυτό σημαίνει ότι αποδέχεται σαν αίθουσα λύσης την Α1. Έτσι η επόμενη μεγαλύτερη διαθέσιμη αίθουσα είναι η Α2, που όμως βλέπουμε ότι έχουμε πλεόνασμα σε θέσεις 25. Η Α2 προσωρινά δεν είναι λύση και βλέπει την αμέσως επόμενη μεγαλύτερη διαθέσιμη αίθουσα που είναι η Α3 που και από αυτή έχει πλεόνασμα 15 θέσεις. Με την ίδια λογική βλέπει και την Α4 που έχει πλεόνασμα 2 θέσεις. Η επόμενη αίθουσα όπως βλέπουμε είναι η Α5 που όμως έχει έλλειμμα 9 θέσεων άρα γυρνάει πίσω κατά μια αίθουσα και αποδέχεται την Α4 σαν δεύτερη αίθουσα. Με την ίδια λογική για το μάθημα Νο2 επιλέγει την αίθουσα Α3. Έτσι όπως παρατηρούμε ενώ στην αρχή χρειαζόμασταν 4 αίθουσες, τώρα χρειαζόμαστε 3 αίθουσες, δηλαδή 2 επιτηρητές λιγότερους. Στο σχήμα 5.2. φαίνεται το λογικό διάγραμμα του CCLSF.

ii. Δεύτερο μέρος αλγορίθμου ωρολογίου προγράμματος εξετάσεων

Για το δεύτερο τμήμα, που είναι και η εύρεση των επιτηρητών των μαθημάτων που προγραμματίσαμε, δημιουργήσαμε δύο αλγορίθμους που μπορούν να αλληλοσυμπληρώνονται. Οι αλγόριθμοι αυτοί αναλύονται ως εξής:

1. 1ος Αλγόριθμος δεύτερου μέρους προγράμματος εξετάσεων

Στον 1ο αλγόριθμο δημιουργούμε μια λίστα καθηγητών ταξινομημένη σε αύξουσα διάταξη ως προς την προτεραιότητα των καθηγητών, όπως περιγράψαμε σε προηγούμενες αναφορές, και παίρνοντας κάθε καθηγητή υπολογίζουμε πόσες επιτηρήσεις του αντιστοιχούν ανά τμήμα που διδάσκει, από τις εισηγήσεις που ήδη έχουν βρεθεί. Σε αυτό βοηθάει ο συντελεστής λ που μας δίνει την αναλογία (Ανάγκες σε ώρες εξετάσεων)/(Διαθέσιμες ώρες), για το κάθε Τμήμα, και πηγάζει από την κάτωθι σχέση

$$\lambda = (\Sigma \text{AE} * \Sigma \text{EA} + \text{AM} * \text{ΛΕΕ}) * \Delta \text{E} / (\text{EE} * \Sigma \Omega \Delta \Delta \text{E})$$

Όπου:

ΣΑΕ:Σύνολο Αιθουσών που προγραμματίστηκαν για την Εξεταστική από το πρώτο τμήμα του αλγορίθμου για το Τμήμα που μελετάμε.

ΣΕΑ:Σύνολο Επιτηρητών που χρειάζονται ανά Αίθουσα.

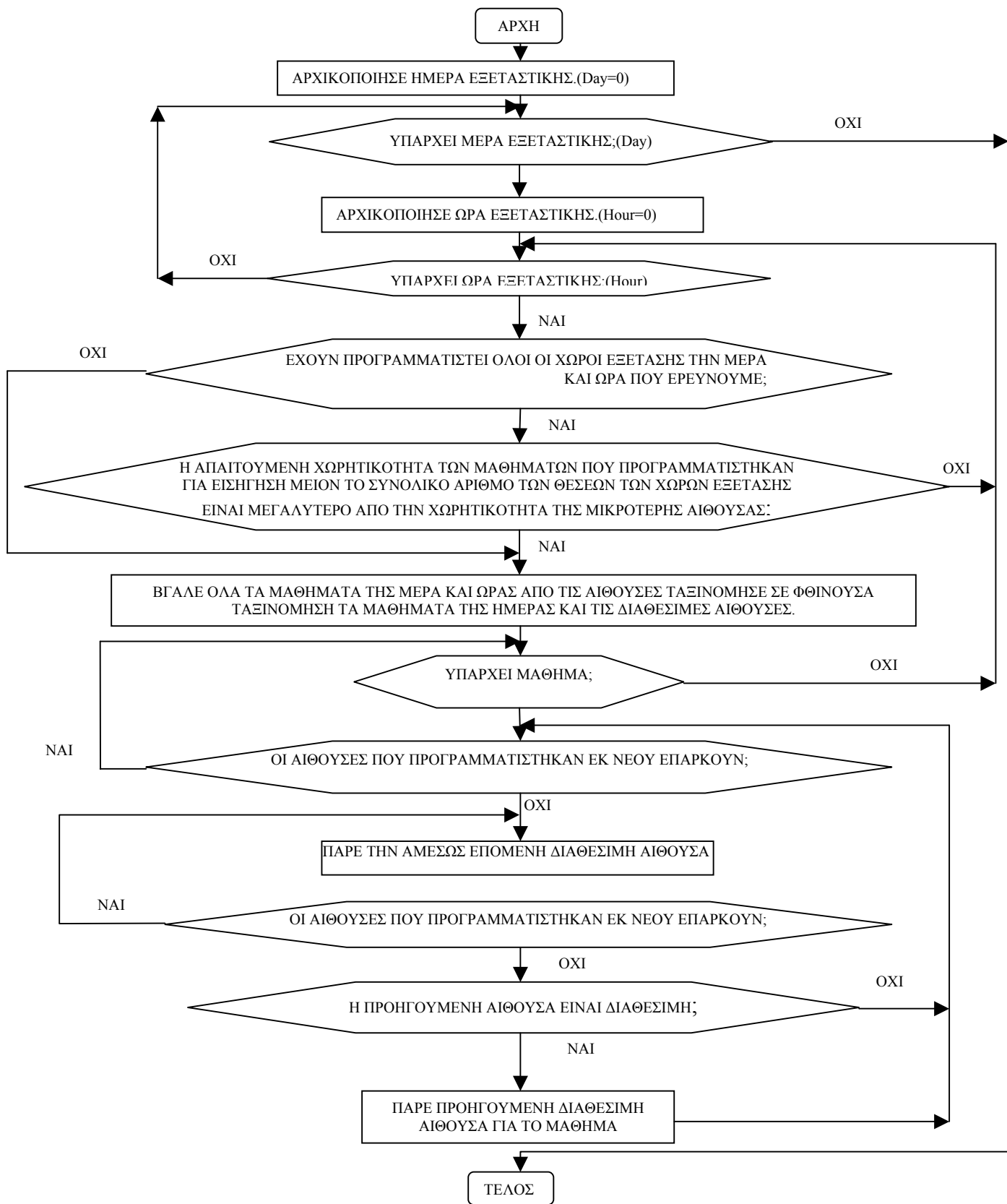
ΑΜ:Αριθμός Μαθημάτων που θα εξεταστούν στο Τμήμα

ΛΕΕ:Λόγος Εισηγήση/Επιτήρηση

ΔΕ:Διάρκεια εξέτασης

ΕΕ:Εβδομάδες Εξεταστικής

ΣΩΔΔΕ:Σύνολο Ωρών Διδασκαλίας Διδασκόντων ανά Εβδομάδα για κάθε τμήμα



Σχήμα 5.2. Πρώτο μέρος αλγορίθμου ΩΠΕ (Αλγόριθμος CCLSF)

Οπότε, ο υπολογισμός για τις επιτηρήσεις που αντιστοιχούν σε κάθε διδάσκοντα (ΕΔ) για κάθε

τιμήμα που διδάσκει υπολογίζεται από τον τύπο:

$$E\Delta=(\lambda*\Omega\Delta E*E\bar{E}/\Delta E)-A\bar{E}\Delta*\Sigma E\bar{E}$$

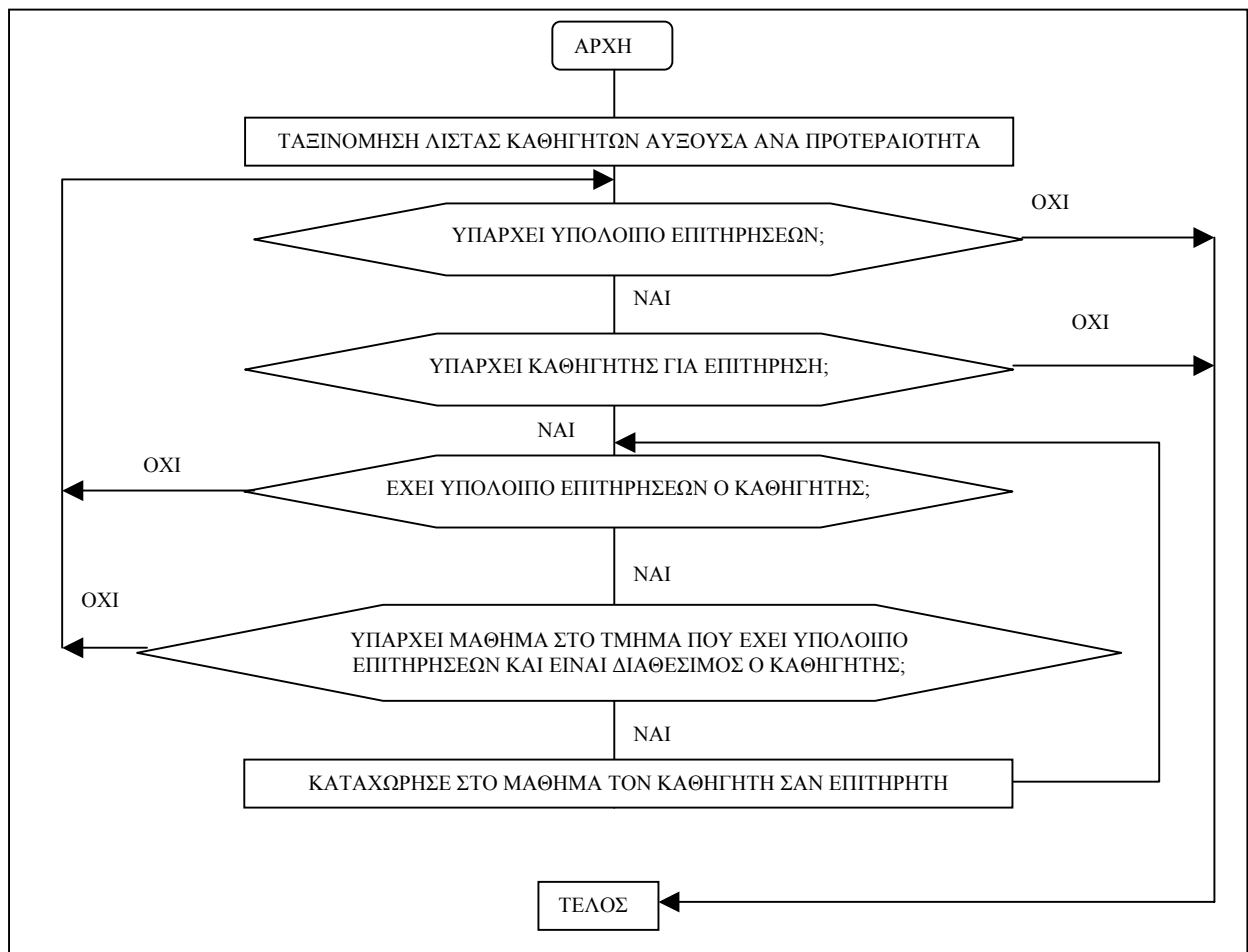
Όπου: ΩΔΕ:Ώρες Διδάσκοντα Ανά Εβδομάδα ανά τμήμα που διδάσκει

ΑΕΔ:Αριθμός Εισηγήσεων Διδάσκοντα

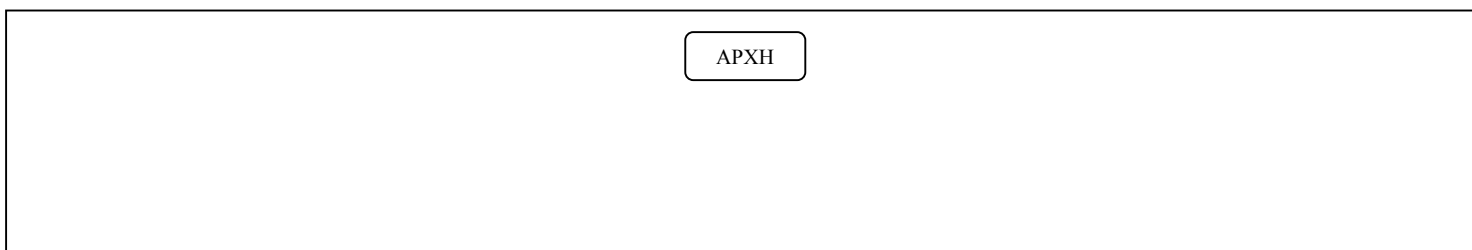
Επειδή πολλές φορές θέλουμε να μειώσουμε το ποσοστό συμμετοχής του εκπαιδευτικού προσωπικού στις εξετάσεις, μια λύση είναι ο εισηγητής να έχει διπλό ρόλο δηλαδή, να συμπεριλαμβάνετε και σαν επιτηρητής στον υπολογισμό μας. Έτσι αν έχουμε ΑΜ αριθμό μαθημάτων που θα εξεταστούν στο Τμήμα, με αυτή τη ρύθμιση έχουμε ΑΜ λιγότερους επιτηρητές, οπότε ο συντελεστής λ γίνεται ως εξής:

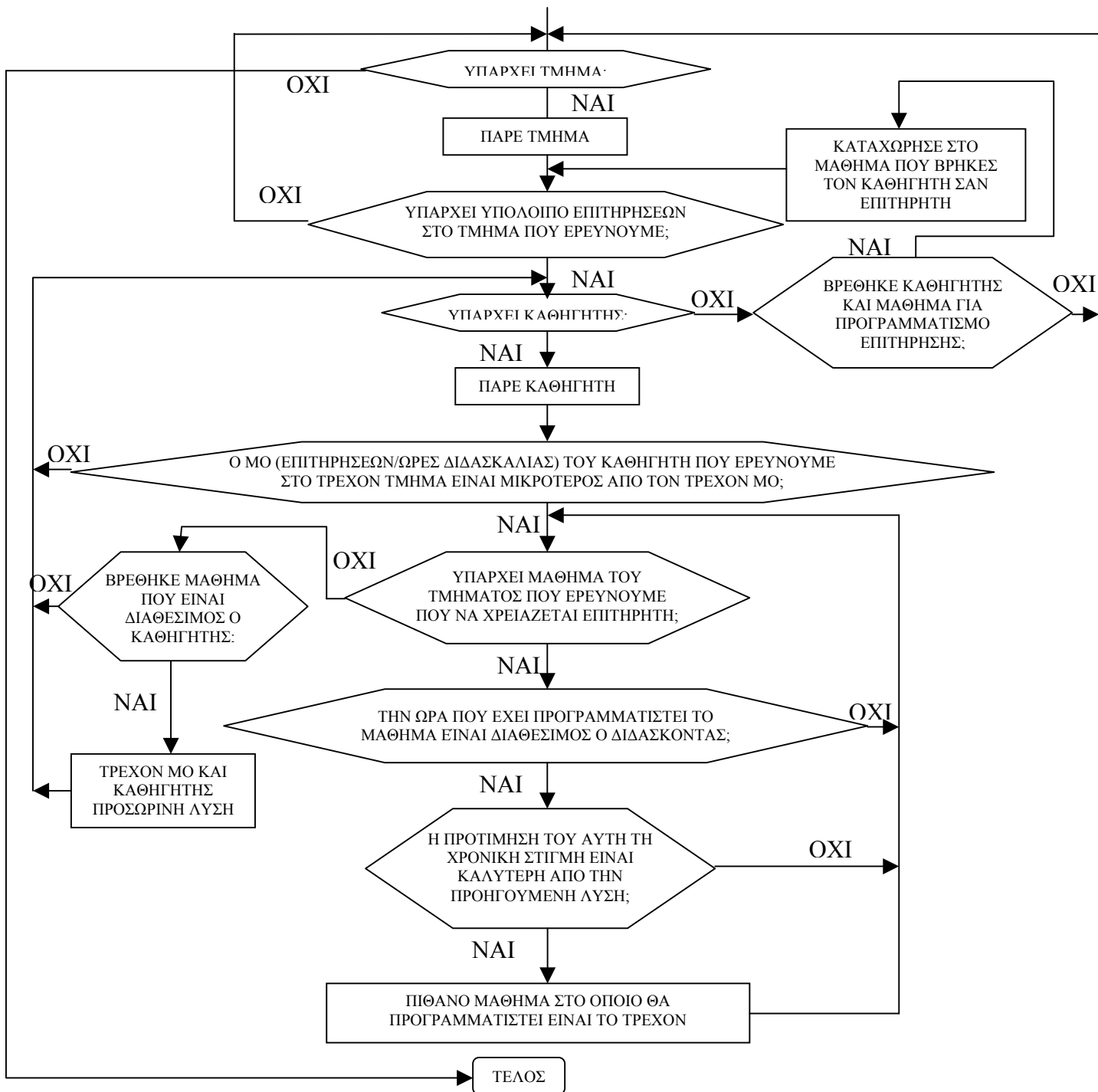
$$\lambda=(\Sigma A\bar{E}*\Sigma E\bar{A} + A\bar{M}*(\Sigma E\bar{E}-1))*\Delta E/(E\bar{E}*\Sigma \Omega\Delta\Delta E)$$

Το τμήμα (ή τα τμήματα) που ανήκει κάθε διδάσκοντας φαίνεται από τις επιλογές του στα μαθήματα και από τις επιθυμητές ώρες που έχει για το καθένα δηλωθέν μάθημα. Έτσι ο αλγόριθμος ελέγχει κάθε χρονική στιγμή αν είναι διαθέσιμος ο καθηγητής, τι προτεραιότητα έχει, αν υπάρχει μάθημα που δεν έχει συμπληρώσει επιτηρητές και αν η καινούργια επιλογή του καθηγητή είναι καλύτερη από την προηγούμενη επιλέγει την καινούργια. Αν βρει επιλογή DBToo_Much τότε τον προγραμματίζει στο μάθημα που βρήκε αυτή την υψηλή προτίμηση, χωρίς να ψάξει παρακάτω. Στο σχήμα 5.3 που ακολουθεί φαίνεται το λογικό διάγραμμα του 1ου αλγορίθμου του 2ου μέρους του ΩΠΕ.



Σχήμα 5.3. Δεύτερο μέρος ΩΠΕ (1ος Αλγόριθμος)





Σχήμα 5.4. Δεύτερο μέρος ΩΠΕ (2ος Αλγόριθμος)

2. 2ος Αλγόριθμος δεύτερου μέρους προγράμματος εξετάσεων

Στον δεύτερο αλγόριθμο για να καταχωρήσουμε μια επιτήρηση ενός τμήματος υπολογίζουμε για κάθε καθηγητή ανεξαρτήτως προτεραιότητας το κάτωθι κλάσμα:

$$ΜΟ = (ΣΕΙΣΔ * ΛΕΕ + ΣΕΠΔ + 1) / ΩΔΔ$$

Όπου:

ΣΕΠΔ= Σύνολο Επιτηρήσεων που έχει προγραμματιστεί στο τμήμα ο Διδάσκοντας μέχρι στιγμής

ΣΕΙΣΔ= Σύνολο Εισηγήσεων που έχει προγραμματιστεί στο τμήμα ο Διδάσκοντας μέχρι στιγμής

ΛΕΕ:Λόγος Εισήγηση/Επιτήρηση





ΩΔΔ= Ώρες Διδασκαλίας Καθηγητή στο τμήμα που ερευνούμε την επιτήρηση

Το +1 μπαίνει επειδή θέλουμε να δούμε αν με την πρόσθετη επιτήρηση που θα προστεθεί, υπάρχει καθηγητής ο οποίος θα είναι με μικρότερο ΜΟ. Αυτό χρειάζεται επειδή ένας καθηγητής με πολλές ώρες διδασκαλίας επηρεάζει σε πολύ μικρότερο βαθμό το κλάσμα από ένα καθηγητή με λιγότερες ώρες. Έτσι φαίνεται ξεκάθαρα ότι όποιος έχει το μικρότερο αριθμό του κλάσματος, που αναφέραμε, θα χρεωθεί την επιτήρηση. Προσοχή όμως, υπάρχει περίπτωση να μην χρεωθεί επιτήρηση ο καθηγητής που έχει το μικρότερο ΜΟ επειδή δεν υπάρχει μάθημα που να χρειάζεται επιτηρητές την χρονική στιγμή που επιθυμεί, αλλά να υπάρχει ανάγκη από επιτηρητές σε μάθημα που έχει δηλώσει απόλυτη αρνητική προτίμηση, οπότε λογικά ο αλγόριθμος δεν τον προγραμματίζει. Αυτό έχει σαν αποτέλεσμα την επιτήρηση αυτή να την χρεωθούν άλλοι καθηγητές. Στο σχήμα 5.4. φαίνεται το λογικό διάγραμμα του 2ου αλγορίθμου του δεύτερου μέρους του ΩΠΕ.

Ο δεύτερος αλγόριθμος μας δίνει καλύτερη λύση από τον πρώτο, επειδή κάθε φορά που είναι να καταχωρήσει μια επιτήρηση υπολογίζει για όλους τους καθηγητές το κλάσμα ΜΟ που προαναφέραμε, και επιλέγει την καλύτερη λύση. Βέβαια αυτό απαιτεί μεγάλη υπολογιστική ισχύ. Για παράδειγμα σε Celeron 300MHz ο δεύτερος αλγόριθμος με τα δεδομένα που έχουμε καταχωρήσει για το τμήμα Ηλεκτρονικής και Φυσικών Πόρων χρειάζεται περίπου 13 με 14 λεπτά, ενώ σε Plato 2,6GHZ χρειάζεται περίπου 3 λεπτά. Ο αντίστοιχος χρόνος για τον 1ο αλγόριθμο είναι μερικά δευτερόλεπτα και στους 2 υπολογιστές που προαναφέραμε. Επίσης ο 1ος αλγόριθμος μπορεί να τελειώσει και να μην έχει καλύψει όλα τα μαθήματα από άποψη επιτηρήσεων. Ο δεύτερος αλγόριθμος έχει πολύ μικρότερη πιθανότητα να σταματήσει χωρίς να προγραμματίσει επιτηρήσεις για μαθήματα, και αν δεν έχει προγραμματίσει επιτηρήσεις σημαίνει ότι δεν μπορεί να συνεχίσει πλέον με τον γραμμικό τρόπο που λειτουργεί. Έτσι όπως καταλαβαίνετε ο δεύτερος αλγόριθμος είναι εν' μέρει συμπλήρωμα του 1ου αλγορίθμου. Και οι δυο αλγόριθμοι μπορούν να τρέξουν ανεξάρτητα μεταξύ τους μετά την εύρεση εισηγήσεων, ή και συμπληρωματικά ο ένας για τον άλλο. Βέβαια δεν έχει σημασία να τρέξει ο 1ος αλγόριθμος μετά τον 2ο. Η προτεινόμενη λύση στην εύρεση επιτηρήσεων μετά την εύρεση των εισηγήσεων είναι ο 2ος αλγόριθμος μόνος του ή πρώτα ο 1ος αλγόριθμος και στη συνέχεια, αν δεν συμπληρωθούν οι επιτηρήσεις, να τρέξει και ο 2ος αλγόριθμος.

5. Χρήση της εφαρμογής

a. Ενημέρωση στοιχείων

Η εφαρμογή δίνει τη δυνατότητα στο χρήστη να εισάγει, να διορθώσει και να διαγράψει δεδομένα από τους πιο σημαντικούς πίνακες της Βάσης Δεδομένων. Η διαδικασία που χρειάζεται για την ενημέρωση των δεδομένων μιας εγγραφής είναι κοινή σε όλες τις φόρμες. Έτσι λοιπόν για να διορθώσουμε ένα πεδίο αρκεί να πατήσουμε διπλό κλικ σε αυτό. Στη συνέχεια αν τα δεδομένα του είναι τύπου λίστας, το πεδίο ενημερώνεται άμεσα με το κλείσιμο της λίστας. Σε περίπτωση που δεν είναι πλαίσιο λίστας, πατώντας το ποντίκι αλλού στην φόρμα ή το πλήκτρο Tab ενημερώνεται το πεδίο. Αυτές οι διορθώσεις γίνονται δυναμικά στην μνήμη. Για να αποθηκευτούν οι αλλαγές που έχουν γίνει, ο χρήστης αρκεί να πατήσει το πλήκτρο με τη δισκέτα  στην γραμμή εργαλείων. Σε αντίθετη περίπτωση αν θέλουμε να αναιρέσουμε όλες τις αλλαγές που κάναμε αρκεί να πατήσουμε το κουμπί με τη δισκέτα και την κόκκινη διαγώνιο . Στην περίπτωση που θέλουμε να εισάγουμε μια καινούργια εγγραφή, αρκεί να πατήσουμε το μπλε βελάκι στη γραμμή εργαλείων με το πράσινο + στο δεξί του άκρο . Με αυτή την ενέργεια στο τέλος των εγγραφών δημιουργείται μια νέα εγγραφή με στοιχεία ΝΕΑ ΕΓΓΡΑΦΗ. Αυτή είναι η νέα σας εγγραφή και μπορείτε να την επεξεργαστείτε όπως προαναφέραμε. Αν θέλουμε να διαγράψουμε μια εγγραφή, αρκεί να έχουμε επιλέξει την αντίστοιχη εγγραφή, και να πατήσουμε το πλήκτρο Del ή το μπλε βελάκι με το κόκκινο X στο δεξί του άκρο . Αυτό θα σημάνει όμως τη δυναμική διαγραφή της εγγραφής. Με το που γίνει αυτή η ενέργεια στο βασικό πίνακα κάθε φόρμας ενημέρωσης στοιχείων υπάρχει ένα πεδίο που ονομάζεται Διαγραμμένο, και παίρνει την τιμή ΝΑΙ. Για να αναιρεθεί αυτή η ενέργεια ή για να ολοκληρωθεί η διαγραφή από τη ΒΔ τηρούμε τις διαδικασίες που αναφέραμε. Αν θέλουμε να ταξινομήσουμε ως προς μια στήλη ενός πίνακα δεδομένων αρκεί να πατήσουμε στην επικεφαλίδα της στήλης ένα κλικ Παρακάτω στις παραγράφους 6.1.1., 6.1.2., 6.1.3, 6.1.4. εξηγείται η λειτουργία της κάθε επιλογής του χρήστη.

i.Ενημέρωση στοιχείων Τμημάτων

Στην ενημέρωση Τμημάτων έχουμε τη δυνατότητα να εισάγουμε – διαγράψουμε Τμήμα, καθώς και να ενημερώσουμε δεδομένα με τις διαδικασίες που έχουμε προαναφέρει. Για να εισέλθετε σε αυτή τη φόρμα επιλέγετε Ενημέρωση->Τμήματα. Στον πίνακα 6.1., που ακολουθεί, φαίνονται τα πεδία του πίνακα τμήματα σχολής, καθώς και ποια πεδία μπορούμε να ενημερώσουμε

Στήλη	Πληροφορίες	Προσβάσιμο
1.	Αύξων αριθμός	ΟΧΙ
2.	Συντομογραφία Τμήματος	ΝΑΙ
3.	Αναλυτική περιγραφή Τμήματος	ΝΑΙ
4.	Χαρακτήρας κάλυψης Τμήματος	ΝΑΙ
5.	Χρώμα απεικόνισης Τμήματος	ΝΑΙ
6.	Διαγραμμένο	ΝΑΙ
7.	Κωδικός	ΟΧΙ

Πίνακας 6.1. Πεδία πίνακα τμημάτων σχολής

Στον παραπάνω πίνακα η στήλη με τον τίτλο Στήλη συμβολίζει την αντίστοιχη στήλη

ενημέρωσης βασικών στοιχείων διδασκόντων. Η μέτρηση ξεκινάει από αριστερά προς τα δεξιά. Η στήλη με τον τίτλο Δεδομένα δείχνει τι δεδομένα παρουσιάζονται στην κάθε στήλη. Στην στήλη με τον τίτλο Προσβάσιμο δηλώνει αν έχουμε τη δυνατότητα να διορθώσουμε τα αντίστοιχα δεδομένα. Έτσι λοιπόν από τον πίνακα 6.1. βλέπουμε ότι η στήλη Νο1 δεν μπορεί να ενημερωθεί από το χρήστη αφού είναι δυναμικά φτιαγμένη και μας δείχνει κάθε φορά τον αύξοντα αριθμό της εγγραφής που βρισκόμαστε. Η στήλη 15 δεν μπορεί να ενημερωθεί επειδή περιέχει το κλειδί του καθηγητή στη Βάση Δεδομένων, το οποίο είναι ξεχωριστό για κάθε καθηγητή και δίνεται κάθε φορά από την εφαρμογή.

Τα πεδία τα οποία χρήζουν επισήμανσης είναι αυτά των στηλών Νο4 και Νο5. Ο χαρακτήρας κάλυψης στη στήλη Νο4 δηλώνει ένα χαρακτήρα ο οποίος θα φαίνεται όταν παρουσιάζεται η κάλυψη των αιθουσών είτε στην εξεταστική είτε στο πρόγραμμα παραδόσεων. Επίσης επειδή μια ματιά ίσον χίλιες λέξεις, πολλές φορές η απεικόνιση του τμήματος με χρώμα όπως είναι η κάλυψη, στον προγραμματισμό καθηγητών για επιτηρήσεις το πεδίο της στήλης Νο5 μας δίνει τη δυνατότητα να αλλάξουμε αυτή την επιλογή αλλάζοντας τον κωδικό του χρώματος.

ii.Ενημέρωση στοιχείων χώρων διδασκαλίας

Η φόρμα ενημέρωση χώρων διδασκαλίας περιλαμβάνει ένα πίνακα με όλα τα στοιχεία που απαρτίζουν ένα χώρο διδασκαλίας. Όλα τα στοιχεία μπορούν να αλλάξουν εκτός των στοιχείων των στηλών Νο1 και Νο8. Σε αυτό τον πίνακα φαίνονται όλοι οι χώροι διδασκαλίας του ιδρύματος δηλαδή όλων των τμημάτων που συστεγάζονται. Οι πληροφορίες που παρέχει ο πίνακας φαίνονται παρακάτω στον πίνακα 6.2.:

Στήλη	Πληροφορίες	Προσβάσιμο
1.	Αύξων αριθμός	OXI
2.	Συντομογραφία χώρου διδασκαλίας	NAI
3.	Αναλυτική περιγραφή χώρου διδασκαλίας	NAI
4.	Χωρητικότητα που έχει για διδασκαλία	NAI
5.	Χωρητικότητα που έχει σαν χώρος εξέτασης	NAI
6.	Τι κατηγορία είναι	NAI
7.	Διαγραμμένο	NAI
8.	Κωδικός	OXI

Πίνακας 6.2. Πίνακας χώρων διδασκαλίας σχολής

Αξίζει να εξηγήσουμε τη σημασία του πεδίου Κατηγορία χώρου διδασκαλίας. Επιλέγοντας μια κατηγορία για ένα χώρο διδασκαλίας, από μια λίστα με δεδομένα τα οποία αλλάζουν μόνο από το χειριστή του συστήματος, έχουμε αμέσως ενημερώσει την εφαρμογή μας για το αν ο χώρος διδασκαλίας αυτός συμμετέχει στην εξεταστική.

iii.Ενημέρωση στοιχείων μαθημάτων

Αν επιλέξουμε από το μενού Ενημέρωση->Μαθήματα βλέπουμε τρεις πίνακες. Ο πρώτος πίνακας έχει όλα τα δεδομένα για κάθε μάθημα, τα οποία θα αναλυθούν παρακάτω, ο δεύτερος πίνακας έχει όλους τους χώρους διδασκαλίας που υπάρχουν στον χώρο που διεξάγεται η παράδοση των μαθημάτων και των εξετάσεων, και ένας τρίτος πίνακας που επεξηγεί σε ποιους χώρους διδασκαλίας μπορεί να διδαχθεί ένα μάθημα. Η λειτουργία του κάθε πίνακα, καθώς και η χρήση του αναλύονται ως εξής:

1. Πίνακας μαθημάτων σχολής

Σε αυτόν τον πίνακα φαίνονται όλα τα μαθήματα που υπάρχουν στις σχολές με ταξινόμηση αύξουσα ως προς το τμήμα, αύξουσα ως προς το εξάμηνο και αύξουσα ως προς την συντομογραφία του μαθήματος. Για την ταξινόμηση κάθε στήλης αυτού του πίνακα ισχύουν

αυτά που αναφέραμε στην εισαγωγή του βου Κεφαλαίου. Για να πετύχουμε όμως την αρχική ταξινόμηση αρκεί να πατήσουμε κλικ στην επικεφαλίδα της πρώτης στήλης. Ο πίνακας μαθημάτων σχολής αποτελείται από δεκαπέντε στήλες των οποίων τα δεδομένα φαίνονται στον παρακάτω πίνακα 6.3.

Στήλη	Πληροφορίες	Προσβάσιμο
1.	Αύξων αριθμός μαθήματος	OXI
2.	Τμήμα μαθήματος	NAI
3.	Εξάμηνο μαθήματος	NAI
4.	Συντομογραφία μαθήματος	NAI
5.	Είδος μαθήματος π.χ. αν είναι θεωρία ή εργαστήριο	NAI
6.	Ώρες που διδάσκεται ανά εβδομάδα	NAI
7.	Αριθμό τμημάτων που έχει το μάθημα	NAI
8.	Αριθμό διδασκόντων	NAI
9.	Πλήρη ονομασία μαθήματος	NAI
10.	Διαγραμμένο	NAI
11.	Αριθμός φοιτητών που το δήλωσαν	NAI
12.	Αναμενόμενοι φοιτητές για εξέταση	NAI
13.	Κλάδο που ανήκει το μάθημα	NAI
14.	Εξετάζεται θεωρητικά	NAI
15.	Κωδικός μαθήματος	OXI

Πίνακας 6.3. Πίνακας μαθημάτων σχολής

Οι στήλες Νο1, Νο 15 δεν μπορούν να αλλάξουν για τους λόγους που αναφέραμε σε προηγούμενες παραγράφους. Για να διορθώσουμε τα υπόλοιπα στοιχεία αυτού του πίνακα ακολουθούμε τις γνωστές διαδικασίες. Η στήλη Νο2 σε διαδικασία διόρθωσης είναι τύπου λίστας όπου εμφανίζει όλα τα τμήματα που συστεγάζονται. Για τη διαδικασία επεξεργασίας των στοιχείων της λίστας των τμημάτων γίνεται αναφορά παρακάτω στην παράγραφο ενημέρωση τμημάτων. Για τα δεδομένα της στήλης Νο5, που δείχνει το είδος του κάθε μαθήματος π.χ. αν είναι θεωρία ή εργαστήριο, είναι και αυτά σε λίστα, μόνο που για να αλλάξουν αυτά τα δεδομένα πρέπει να ενημερωθεί ο διαχειριστής του προγράμματος. Η στήλη Νο6 μας δείχνει τις ώρες ανά εβδομάδα που χρειάζεται για να διεξαχθεί το μάθημα. Η στήλη Νο7 μας δείχνει πόσα τμήματα αυτού του μαθήματος θα δοθούν. Η στήλη Νο8 δείχνει πόσοι διδάσκοντες είναι αναγκαίοι για να διεξαχθεί το μάθημα. Η στήλη Νο11 μας δείχνει πόσοι φοιτητές έχουν δηλώσει το μάθημα για να το διδαχθούν. Επειδή όμως συνήθως οι φοιτητές που συμμετέχουν στην εξεταστική είναι λιγότεροι από αυτούς που το έχουν δηλώσει δημιουργήσαμε την στήλη Νο 12 που δείχνει τους αναμενόμενους φοιτητές στην εξεταστική. Η στήλη αυτή μπορεί να αλλάξει από εδώ με τον τρόπο που έχουμε πει, δηλαδή, διπλό κλικ κτλ, αλλά και από τις ρυθμίσεις προγράμματος εξετάσεων αλλάζοντας το ποσοστό των φοιτητών προς εξέταση. Στη στήλη 13 φαίνεται ο κλάδος που ανήκει κάθε μάθημα. Επειδή παρατηρήθηκε το φαινόμενο σε κάποια εξάμηνα να υπάρχουν παραπάνω μαθήματα από τις μέρες της εξεταστικής, υπήρχε πρόβλημα στη διεξαγωγή των εξετάσεων και έτσι κρίθηκε χρήσιμη η δυνατότητα κατάταξης κάποιων μαθημάτων σε κλάδους. Οι κλάδοι έχουν νόημα μόνο για μαθήματα ίδιου τμήματος και εξαμήνου. Η κατηγοριοποίηση που έχει γίνει είναι η εξής:

- Κατ' αρχήν μαθήματα ίδιου κλάδου δεν μπορούν να προγραμματιστούν για εξέταση την ίδια μέρα.
- Μαθήματα διαφορετικού κλάδου μπορούν να προγραμματιστούν για εξέταση την ίδια ημέρα.

- Την ίδια ημέρα και ώρα μπορούν να προγραμματιστούν μαθήματα για εξέταση μόνο όταν είναι όλα τα υποψήφια μαθήματα πάνω από ένα όριο κλάδου και ανήκουν σε διαφορετικούς κλάδους. Το όριο του κλάδου που αναφέραμε μπορεί να το ρυθμίσει ο χρήστης από τις ρυθμίσεις προγράμματος εξετάσεων. Έτσι όπως καταλαβαίνετε μαθήματα πάνω από το όριο που αναφέραμε είναι μαθήματα κατεύθυνσης, ενώ κάτω από το όριο είναι βασικά μαθήματα, που επειδή όμως δεν χωράνε στην εξεταστική τα δηλώνουμε σε διαφορετικούς κλάδους ώστε να μπουν την ίδια μέρα.

Τέλος στη στήλη Νο14 μπορούμε να δηλώσουμε αν το μάθημα συμμετέχει ή όχι στην εξεταστική. Αυτή η επιλογή αξίζει να προσεχθεί επειδή μπορεί να μην προγραμματίζεται κάποιο μάθημα επειδή δεν έχει δηλωθεί ότι εξετάζεται. Το ίδιο επίσης ισχύει και για τη στήλη φοιτητές εξετάσεων.

2. Πίνακας χώροι διδασκαλίας

Σε αυτό τον πίνακα φαίνονται όλοι οι χώροι διδασκαλίας του ιδρύματος δηλαδή όλων των τμημάτων που συστεγάζονται. Σε αυτό τον πίνακα δεν μπορούμε να επέμβουμε και να αλλάξουμε δεδομένα αλλά μας χρειάζεται για να δηλώσουμε προτεινόμενους χώρους διεξαγωγής του μαθήματος για τον αλγόριθμο του προγράμματος παραδόσεων. Η ταξινόμηση και σε αυτόν τον πίνακα είναι σε αύξουσα διάταξη και μπορεί να αλλάξει με τη γνωστή διαδικασία Ο πίνακας αυτός έχει τα στοιχεία του πίνακα 6.2.

Τα δεδομένα αυτά είναι χρήσιμα ώστε με ένα γρήγορο κούταγμα ο χειριστής της εφαρμογής να επιλέξει τον κατάλληλο ή κατάλληλους χώρους για να διεξαχθεί το μάθημα για διδασκαλία. Για να δηλώσουμε μια αίθουσα από αυτό τον πίνακα σαν προτεινόμενη αίθουσα για διδασκαλία αρκεί να πατήσουμε το βελάκι που δείχνει προς τους προτεινόμενους χώρους διδασκαλίας, αφού βέβαια έχουμε επιλέξει μια αίθουσα, που είναι δεξιά από αυτόν τον πίνακα.

3. Πίνακας προτεινόμενοι χώροι διδασκαλίας

Τέλος στον πίνακα προτεινόμενοι χώροι διδασκαλίας είναι όλοι οι χώροι διδασκαλίας που προτείνονται για να διεξαχθεί η διδασκαλία του μαθήματος. Τα πεδία που έχει αυτός ο πίνακας είναι ακριβώς τα ίδια με του πίνακα της παραγράφου 6.2. οπότε δεν υπάρχει λόγος για επανάληψη. Για να διαγραφτεί ένας χώρος διδασκαλίας από τους προτεινόμενους χώρους διδασκαλίας αρκεί να τον επιλέξουμε και να πατήσουμε το πλήκτρο με το βελάκι που δείχνει στον πίνακα με τους χώρους διδασκαλίας της σχολής. Αξίζει να σημειώσουμε ότι ο πίνακας αυτός, δεν μπορεί να ταξινομηθεί αλλά είναι ταξινομημένος με τη σειρά που έχουν καταχωρηθεί οι προτεινόμενες αίθουσες. Αυτό βοηθάει εν' μέρει στην επιλογή των χώρων διδασκαλίας που επιθυμούμε κατά κάποιο τρόπο.

iv.Ενημέρωση στοιχείων καθηγητών

Σε αυτό το σημείο ο χρήστης έχει τη δυνατότητα να επεξεργαστεί τα στοιχεία των καθηγητών των τμημάτων. Για να επιλέξετε αυτή τη λειτουργία επιλέγετε από το μενού Ενημέρωση->Καθηγητές. Όπως θα παρατηρήσετε, η φόρμα που ανοίγει αποτελείται από 4 πίνακες. Ο πρώτος πίνακας αναφέρεται στους διδάσκοντες της σχολής, ο δεύτερος στις προτιμήσεις στα μαθήματα που έχει, ο τρίτος στο σύνολο των μαθημάτων της σχολής και ο τέταρτος στις προτιμήσεις που έχει στο ωράριο ο καθηγητής. Οι τέσσερις αυτοί πίνακες αναλύονται ως εξής:

1. Πίνακας διδάσκοντες σχολής

Στον πίνακα αυτόν μπορούμε να δούμε στοιχεία όλων των καθηγητών της σχολής. Αποτελείται από 15 στήλες των οποίων το περιεχόμενο επεξηγείτε στον πίνακα 6.4. που ακολουθεί:

Στήλη	Πληροφορίες	Προσβάσιμο
-------	-------------	------------

1.	Αύξων αριθμό τρέχουσας εγγραφής	OXI
2.	Επώνυμο καθηγητή	NAI
3.	Όνομα καθηγητή	NAI
4.	Προτεραιότητα καθηγητή σε σχέση με τους άλλους καθηγητές	NAI
5.	Βαθμίδα καθηγητή	NAI
6.	Ώρες από το νόμο. Πηγάζει από τη βαθμίδα που ανήκει.	OXI
7.	Ώρες επιθυμητές. Σύνολο επιθυμητών του καθηγητή για διδασκαλία	NAI
8.	Ώρες πραγματικές. Σύνολο ωρών που έχει προγραμματιστεί για διδασκαλία στο πρόγραμμα παραδόσεων	OXI
9.	Ειδικότητα του καθηγητή	NAI
10.	Αν έχει διδακτορικό	NAI
11.	Αν έχει Master	NAI
12.	Αριθμός δημοσιεύσεων	NAI
13.	Αν έχει διαγραφτεί η εγγραφή. Αν NAI, έχει διαγραφτεί μόνο από τη μνήμη και όχι από τη Βάση Δεδομένων.	NAI
14.	Αν είναι μόνιμος	NAI
15.	Κωδικός εγγραφής στη Βάση Δεδομένων	OXI

Πίνακας 6.4. Πίνακας καθηγητών σχολής

Η στήλη 6 δεν μπορεί να προσπελαθεί για διόρθωση, επειδή περιέχει τις ώρες από τη βαθμίδα που ανήκει ο καθηγητής. Έτσι αξίζει να παρατηρήσετε κάθε φορά που αλλάζετε βαθμίδα σε ένα καθηγητή, αυτόματα αλλάζει και αυτό το πεδίο τιμή. Οι τιμές για κάθε βαθμίδα έχουν δηλωθεί στη Βάση δεδομένων και αλλάζουν από τον διαχειριστή του προγράμματος όταν του ζητηθεί. Η στήλη 8 δεν μπορεί να ενημερωθεί επειδή μας δείχνει το σύνολο των ωρών που έχει προγραμματιστεί ο καθηγητής για διδασκαλία στο πρόγραμμα παραδόσεων.

Για τη στήλη Νο5 που είναι η βαθμίδα του καθηγητή, ανοίγει ένα πλαίσιο λίστας με όλες τις βαθμίδες που έχουν καταχωρηθεί στη Βάση Δεδομένων, και επιλέγεται από τη λίστα.

Για τις στήλες Νο2, Νο3, Νο9, γράφετε ότι χαρακτήρα θέλετε για τα δεδομένα σας.

Για τις στήλες Νο4, Νο7, Νο12 γράφετε μόνο αριθμούς και μάλιστα ακέραιους θετικούς μέχρι 255.

Για τις στήλες Νο10, Νο11 παρατηρήστε ότι με το διπλό κλικ αλλάζουν τιμή κυκλικά, δηλαδή αν είναι NAI θα γίνει OXI στο αντίστοιχο κελί κ.ο.κ.

2. Πίνακας προτιμήσεις μαθημάτων


Ο πίνακας αυτός αποτελεί τις προτιμήσεις μαθημάτων που έχει κάθε καθηγητής. Έχει οκτώ στήλες και ταξινομημένος σε φθίνουσα διάταξη ανάλογα με το ποιο μάθημα επιθυμεί πιο πολύ. Τα δεδομένα του πίνακα με τις επιλογές των μαθημάτων των καθηγητών φαίνονται στον πίνακα 6.5. που ακολουθεί.

Στήλη	Πληροφορίες	Προσβάσιμο
1.	Προτεραιότητα μαθήματος	OXI
2.	Συντομογραφία μαθήματος	OXI
3.	Είδος μαθήματος π.χ. αν είναι θεωρία ή εργαστήριο	OXI
4.	Ώρες ανά εβδομάδα που διδάσκεται	OXI
5.	Αριθμό τμημάτων που έχει το μάθημα	OXI
6.	Ώρες που επιθυμεί ο καθηγητής να διδάξει από αυτό το μάθημα	NAI
7.	Πραγματικές ώρες που έχει προγραμματιστεί ο καθηγητής για να διδάξει αυτό το μάθημα	OXI

8.	Κωδικός μαθήματος	ΟΧΙ
----	-------------------	-----

Πίνακας 6.5. Πίνακας προτιμήσεων μαθημάτων καθηγητών

Όπως παρατηρούμε στον παραπάνω πίνακα σε κανένα πεδίο δεν μπορούμε να επέμβουμε αλλά μόνο στην στήλη 6, όπου μπορούμε να δηλώσουμε πόσες ώρες επιθυμεί να διδάξει ο καθηγητής από αυτό το μάθημα. Σε αυτό το πεδίο μπορούμε να επέμβουμε με τη γνωστή διαδικασία, δηλαδή πατώντας διπλό κλικ, και πληκτρολογώντας την τιμή των επιθυμητών ωρών του καθηγητή. Ο αριθμός αυτός δεν πρέπει να ξεπερνάει το γινόμενο (Ωρες Μαθήματος X Αριθμός Τμημάτων) του μαθήματος για ευνόητους λόγους. Η μόνη εξαίρεση σε αυτό τον κανόνα είναι κάποια εικονικά μαθήματα που ονομάζονται ΕΠΙΤΗΡΗΣΗ, και έχουν δημιουργηθεί για να δοθούν ώρες επιτήρησης σε ΕΤΠ για την κάλυψη των αναγκών. Το πεδίο αυτό έχει διπλό ρόλο:

Από αυτό το πεδίο φαίνεται πόσες ώρες επιθυμεί να προγραμματιστεί ο καθηγητής σε κάθε μάθημα άρα και σε κάθε τμήμα που ανήκουν τα μαθήματα. Βάση αυτών των ωρών ο καθηγητής θα προγραμματιστεί με αντίστοιχες ώρες επιτηρήσεων και εισηγήσεων στο πρόγραμμα εξετάσεων για το κάθε τμήμα. Μια άλλη χρησιμότητα του συγκεκριμένου πεδίου στον αλγόριθμο του προγράμματος παραδόσεων, όπου ελέγχουμε να μην υπερβεί ο καθηγητής το όριο των ωρών σε κάθε μάθημα της προτίμησης του, αλλά και το σύνολο των ωρών που έχει προγραμματιστεί να μην υπερβαίνει το σύνολο των επιθυμητών ωρών διδασκαλίας του πίνακα της παραγράφου 6.6.6.1. Να σημειωθεί ότι κάτω ακριβώς από αυτόν τον πίνακα υπάρχει ένα στοιχείο ελέγχου το οποίο είναι απενεργοποιημένο και μας δείχνει το σύνολο των επιθυμητών ωρών που πηγάζει από το άθροισμα των επιμέρους προτιμήσεων. Ο πίνακας αυτός δεν μπορεί να υποστεί ταξινόμηση με τη διαδικασία που αναφέραμε στην προηγούμενη παράγραφο, επειδή είναι ήδη ταξινομημένος ως προς την προτίμηση που έχει σε κάθε μάθημα ο καθηγητής. Για να αφαιρέσουμε ένα μάθημα από τις προτιμήσεις του καθηγητή αρκεί να επιλέξουμε αυτό το μάθημα και να πατήσουμε το κουμπί με το εικονίδιο . Στη συνέχεια βέβαια πρέπει να πατήσουμε αποθήκευση δεδομένων για να αποθηκευτούν οι αλλαγές. Η διαδικασία της καταχώρησης μαθήματος εξηγείται στην επόμενη παράγραφο.

3. Πίνακας μαθημάτων σχολής

Ο πίνακας αυτός αποτελεί το σύνολο των μαθημάτων της σχολής. Τα δεδομένα αυτού του πίνακα φαίνονται στον πίνακα 6.6. Έχει οκτώ στήλες οι οποίες αναλύονται στη συνέχεια.

Στήλη	Πληροφορίες	Προσβάσιμο
1.	Αύξων αριθμός μαθήματος	ΟΧΙ
2.	Τμήμα μαθήματος	ΟΧΙ
3.	Εξάμηνο μαθήματος	ΟΧΙ
4.	Συντομογραφία μαθήματος	ΟΧΙ
5.	Είδος μαθήματος π.χ. αν είναι θεωρία ή εργαστήριο	ΟΧΙ
6.	Ωρες ανά εβδομάδα που διδάσκεται	ΟΧΙ
7.	Αριθμό τμημάτων που έχει το μάθημα	ΟΧΙ
8.	Κωδικός μαθήματος	ΟΧΙ

Πίνακας 6.6. Πίνακας μαθημάτων σχολής

Σε αυτόν τον πίνακα δεν μπορούμε να επέμβουμε και να κάνουμε διορθώσεις όπως στους προηγούμενους πίνακες. Είναι βασικός όμως ώστε να μπορεί ο καθηγητής να επιλέγει τα μαθήματα προτίμησης του από εδώ. Έτσι αν έχουμε επιλέξει ένα μάθημα, για να το καταχωρήσουμε στις προτιμήσεις του καθηγητή αρκεί να πατήσουμε το βελάκι που δείχνει προς τα πάνω που είναι ακριβώς πάνω από τον πίνακα τον οποίο αναλύουμε. Η ταξινόμηση αυτού του πίνακα είναι ίδια με τον πίνακα μαθημάτων σχολής της παραγράφου 6.1.3. και ισχύουν οι ίδιες διαδικασίες ταξινόμησης.

4. Πίνακας προτιμήσεων ωραρίου

Σε αυτό το σημείο της φόρμας ο καθηγητής μπορεί να επιλέξει με γραφικό τρόπο τις προτιμήσεις που έχει ως προς το ωράριο. Ο πίνακας επιλογών ωραρίου αποτελείται από τόσες στήλες όσες είναι και οι μέρες που παραδίδονται τα μαθήματα συν μια στήλη, και τόσες γραμμές όσες είναι οι ώρες διδασκαλίας συν μια. Κάτω από τον κεντρικό πίνακα προτιμήσεων ωραρίου υπάρχει μια παλέτα από χρώματα όπου κάθε χρώμα αντιστοιχεί σε μια προτίμηση του καθηγητή. Το έντονο πράσινο είναι η υψίστη προτεραιότητα που βάζει ο καθηγητής στην θετική προτίμησή του. Η επόμενη βαθμίδα, το ανοικτό πράσινο είναι πάλι θετική προτίμηση του καθηγητή αλλά σε μικρότερη προτεραιότητα από την προηγούμενη προτίμηση. Το επόμενο χρώμα που είναι ακόμα πιο ανοικτό πράσινο από τα δυο προηγούμενα χρώματα, έρχεται και αυτό σε μικρότερη θετική προτίμηση. Το λευκό συμβολίζει ουδετερότητα στην προτίμηση του καθηγητή, δηλαδή δεν τον ενδιαφέρει τη χρονική στιγμή που θα προγραμματιστεί για να διδάξει ή για να επιτηρήσει ή να εισηγηθεί μάθημα. Το πορτοκαλί χρώμα δηλώνει μια μικρή αρνητική προτίμηση στα σημεία που θα βρεθεί. Εκεί ο καθηγητής θα προγραμματιστεί αν δεν υπάρχει άλλη λύση από τις προηγούμενες. Τέλος, το κόκκινο χρώμα δηλώνει την απόλυτα αρνητική προτίμηση του καθηγητή, οπότε ο καθηγητής δεν προγραμματίζεται για κανένα λόγο σε αυτή την προτίμηση. Να σημειωθεί ότι αυτές οι προτιμήσεις είναι για το πρόγραμμα παραδόσεων. Επειδή αυτές ισχύουν και κατά τη διάρκεια των εξετάσεων, κατά ένα μεγάλο ποσοστό, γίνεται αναγωγή αυτών των προτιμήσεων σε ώρες εξετάσεων όπως αναφέραμε στον αλγόριθμο του προγράμματος εξετάσεων. Η επιλογή ωραρίου γίνεται ως εξής:

- Επιλέγουμε με το ποντίκι πατώντας κλικ στην παλέτα με τα χρώματα στο χρώμα το οποίο θέλουμε για την προτίμηση του καθηγητή.
- Στη συνέχεια στον βασικό πίνακα με τα ωράρια επιλέγουμε την περιοχή που θέλουμε να έχει αυτή την προτίμηση που επιλέξαμε.
- Για να αποθηκευτούν ή να αναιρεθούν αυτές οι αλλαγές ακολουθούμε τις διαδικασίες των προηγούμενων παραγράφων.

b. Ρυθμίσεις

Ο χρήστης έχει τη δυνατότητα να κάνει κάποιες ρυθμίσεις ώστε να μπορέσει να επέμβει στους αλγόριθμους του ωρολογίου προγράμματος και του προγράμματος εξετάσεων, καθώς και στην εμφάνιση της εκτύπωσης κάποιων παραμέτρων.

i. Βασικές ρυθμίσεις

Με αυτή την επιλογή ο χρήστης έχει τη δυνατότητα να ρυθμίσει το εξάμηνο για το οποίο αναφέρεται το ΩΠΠ ή το ΩΠΕ. Η επιλογή αυτή αποθηκεύεται στη Βάση Δεδομένων, και είναι διαθέσιμη πάντα η τελευταία αλλαγή.

ii. Ρυθμίσεις προγράμματος παραδόσεων

Σε αυτό το σημείο ο χρήστης έχει τη δυνατότητα να ρυθμίσει κάποιες παραμέτρους για το πρόγραμμα των παραδόσεων. Κατ' αρχήν βλέπουμε ότι υπάρχουν οι επιλογές Προτεραιότητα μέσου όρου εργαστηρίων, Προτεραιότητα μέσου όρου συμπλήρωσης διδασκόντων και Προτεραιότητα μέσου όρου συμπλήρωσης ωρών διδασκαλίας, απενεργοποιημένες. Η κάθε μια επιλογή απεικονίζει την προτεραιότητα που θέλουμε να δώσουμε στον αλγόριθμο του ωρολογίου προγράμματος σε περίπτωση ισότητας των παραπάνω μέσων όρων. Η προτεραιότητα του κάθε μέσου όρου αλλάζει, αν παρατηρήσετε, κάθε φορά που πατάτε το κουμπί [**Αναδιάταξη μέσων όρων**]. Στη συνέχεια βλέπουμε δυο πεδία στα οποία μπορούμε να επέμβουμε και να τα διορθώσουμε. Αυτά ονομάζονται «**ποσοστό μείωσης 1ου μέσου όρου**» και «**ποσοστό μείωσης 2ου μέσου όρου**». Έχοντας αυτά στην τιμή 0, η προτεραιότητα που προαναφέραμε είναι απόλυτη δηλαδή αν δύο μέσοι όροι ήταν ίδιοι ο αλγόριθμος θα επέλεγε

την επιλογή με την καλύτερη προτεραιότητα. Αν είχαμε βάλει όμως και τιμές στα πεδία που προαναφέραμε τότε αν είχαμε δηλώσει σε ένα μέσο όρο προτεραιότητα 1 και ήταν μικρότερος από τον μέσο όρο No2 τότε αν ο πρώτος μειωνόταν με το ποσοστό που θα είχαμε επιλέξει και θα προέκυπτε μικρότερος ή ίσος από τους άλλους δυο μέσους όρους, ο αλγόριθμος θα επέλεγε τον πρώτο μέσο όρο. Το ίδιο ισχύει και για το ποσοστό μείωσης του δεύτερου μέσου όρου. Έτσι λοιπόν μετά τις αλλαγές μας αρκεί να πατήσουμε το πλήκτρο [**Καταχώρηση**] και να αποθηκευτούν οι αλλαγές που κάναμε ή [**ΑΚΥΡΩΣΗ**] για να ακυρωθούν. Να σημειωθεί ότι κάθε φορά που πατάμε το κουμπί [**Καταχώρηση**] οι αλλαγές καταχωρούνται αυτόματα στη Βάση Δεδομένων και αντικαθιστούν τις προηγούμενες τιμές.

iii.Ρυθμίσεις προγράμματος εξετάσεων

Στη συνέχεια επιλέγοντας ρυθμίσεις αλγορίθμου προγράμματος εξετάσεων εμφανίζεται ένας διάλογος όπου η αριστερή του στήλη έχει την ίδια λογική με τις ρυθμίσεις του αλγορίθμου του προγράμματος παραδόσεων. Η δεξιά του στήλη έχει κάποια πεδία τα οποία έχουν καθοριστικό ρόλο στην εξεταστική. Έτσι λοιπόν έχουμε το πεδίο «**Ποσοστό συμμετοχής δηλωθέντων φοιτητών**» όπου εδώ μπορούμε να ρυθμίσουμε το σύνολο των φοιτητών που συμμετέχουν στην εξεταστική. Τα αποτελέσματα αυτής της επιλογής φαίνονται στην ενημέρωση μαθημάτων, όπως έχουμε αναλύσει διεξοδικά στην παράγραφο 6.1.2. Αν δεν αλλάξει τιμή αυτό το πεδίο, δεν επηρεάζονται οι τρέχουσες τιμές που έχουν εισαχθεί είτε χειροκίνητα είτε με εφαρμογή ποσοστού παλαιότερα. Το πεδίο «**Ωρες εξέτασης**» δηλώνει τη χρονική διάρκεια της εξέτασης. Αυτό συμβάλλει στην μετατροπή των προτιμήσεων στο ωράριο στο πρόγραμμα παραδόσεων του διδάσκοντα σε αυτό των εξετάσεων. Το πεδίο Ομάδες εξετάσεων μας λέει πόσες ομάδες εξεταστικών θα γίνουν κατά τη διάρκεια κάθε ημέρας της εξεταστικής. Μπορούμε να δηλώσουμε κάθε μέρα εξεταστικής με ένα αριθμό δηλαδή π.χ. 0=Δευτέρα, 1=Τρίτη κ.ο.κ. Να σημειωθεί ότι δεν μπορούμε να δηλώσουμε μέρα εξεταστικής διαφορετική από αυτές που έχουμε δηλώσει για το ωράριο λειτουργίας της σχολής, που είπαμε πως ρυθμίζεται μόνο από το διαχειριστή της εφαρμογής. Κάτω από το πεδίο δήλωσης των ημερών της εξεταστικής υπάρχει ένα απενεργοποιημένο πεδίο που κάθε φορά που ενημερώνεται το πεδίο ημερών εξεταστικής μας δηλώνει τη διάρκεια σε μέρες της εξεταστικής. Αυτό το πεδίο αλλάζει μόνο όποτε αλλάζουμε τις ημέρες των εξετάσεων. Το επόμενο πεδίο είναι [**Εβδομάδες εξετάσεων**] που χρειάζεται στον υπολογισμό των επιτηρήσεων που αναλογούν στον καθένα.. Το επόμενο πεδίο ονομάζεται «**Ποσοστό ευνόησης διδασκόντων**» όπου εδώ κρίνουμε κατά πόσο θα ευνοήσουμε ή όχι τους διδάσκοντες στον πρώτο αλγόριθμο εύρεσης επιτηρήσεων που αναφέραμε στην παράγραφο 5.2.2.1.ως προς τις επιτηρήσεις που θα προγραμματιστούν. Το επόμενο πεδίο είναι πεδίο επιλογής που ενημερώνουμε τους αλγορίθμους εύρεσης επιτηρήσεων για τον αριθμό των επιτηρητών που πρέπει να ορισθούν ανά αίθουσα. Το επόμενο πεδίο είναι η «**ημερομηνία έναρξης της εξεταστικής περιόδου**». Εδώ ο χρήστης πρέπει να προσέξει ότι μπορεί να δηλώσει ημερομηνία την οποία αντιστοιχεί ημέρα διαφορετική από αυτήν στην οποία έχουμε δηλώσει στο πεδίο Μέρες εξεταστικής. Η εφαρμογή αφήνει το χρήστη ελεύθερο να δηλώσει, χωρίς όμως να αποθηκεύσει, αυτές τις προτιμήσεις. Όμως όταν ο χρήστης πατήσει το πλήκτρο [**Καταχώρηση**] και υπάρχει ασυμφωνία, η εφαρμογή ενημερώνει το χρήστη για αυτή τη διαφορά και δεν τον αφήνει να αποθηκεύσει τις αλλαγές που έχει κάνει στις ρυθμίσεις. Το πεδίο [**Αναλογία Εισήγησης/ Επιτήρησης**] δηλώνει την αντιστοιχία που υπάρχει μεταξύ εισήγησης και επιτήρησης. Την λειτουργία αυτής της επιλογής την αναφέραμε στην παράγραφο 5.2.2.1. Το πεδίο, το οποίο είναι τύπου Check Box, με την ένδειξη «**συμμετέχουν μόνο οι μόνιμοι στις επιτηρήσεις**» **ρυθμίζει τον αλγόριθμο του προγράμματος εξετάσεων αν συμμετέχουν μόνο οι μόνιμοι καθηγητές στον προγραμματισμό των επιτηρήσεων**. Με την επιλογή [**Καταχώρηση**] η εφαρμογή διαγράφει το παλιό πρόγραμμα από τη μνήμη αλλά και από τη Βάση Δεδομένων ώστε να μην υπάρξουν προβλήματα σε περίπτωση ανάκτησης παλιού προγράμματος με νέες ρυθμίσεις. Από τις

χειροκίνητες ρυθμίσεις, διαγράφονται μόνο αυτές όπου έχουν καταχωρηθεί εκτός ορίων σύμφωνα με τις νέες ρυθμίσεις.

c. Χρονοπρογραμματισμός

Στη γραμμή του μενού υπάρχει η επιλογή χρονοπρογραμματισμός όπου εδώ είναι οι αλγόριθμοι που αναλύσαμε στα κεφάλαια 4, 5. Με την επιλογή αυτή βλέπουμε δυο επιλογές. Η πρώτη είναι πρόγραμμα παραδόσεων και η άλλη είναι πρόγραμμα εξετάσεων. Η λειτουργία και ο σκοπός της κάθε επιλογής αναλύονται στις παραγράφους 6.2.1., 6.2.2.

i. Πρόγραμμα παραδόσεων

1. Εμφάνιση προγράμματος

Με αυτή την επιλογή εμφανίζεται η φόρμα με τα δεδομένα του ΩΠΠ. Έτσι λοιπόν ο χρήστης έχει στη διάθεση του πίνακες με τις εξής ενδείξεις: “Παρατηρήσεις και Προβλήματα που Βρέθηκαν Μετά την Υλοποίηση του Προγράμματος”, “Πρόγραμμα ανά Τμήμα και Εξάμηνο”, “Ατομική καρτέλα καθηγητή με μαθήματα που προγραμματίστηκε”, “Προγραμματισμός χώρων διδασκαλίας”, “Κάλυψη αιθουσών” και τέλος “Χειροκίνητες καταχωρήσεις”. Τα δεδομένα αυτών των πινάκων αναλύονται ως εξής:

a. Παρατηρήσεις και προβλήματα που βρέθηκαν μετά την υλοποίηση του προγράμματος

Σε αυτό τον πίνακα ο χρήστης έχει στη διάθεση του κάποια δεδομένα τα οποία θα του φανούν χρήσιμα για την υλοποίηση του προγράμματος παραδόσεων. Τα δεδομένα του πίνακα αποτελεσμάτων επεξηγούνται στον παρακάτω πίνακα

Τίτλος	Επεξήγηση
Σύνολο ωρών μαθημάτων Σχολής	Ώρες όλων των μαθημάτων των τμημάτων που συστεγάζονται στη Σχολή
Σύνολο ωρών μαθημάτων που προγραμματίστηκαν	Πόσες ώρες από το σύνολο των ωρών της Σχολής έχουν προγραμματιστεί για διδασκαλία
Σύνολο ωρών καθηγητών σχολής	Άθροισμα των επιθυμητών ωρών όλων των καθηγητών. Όχι το άθροισμα των επιμέρους αλλά του συνόλου των επιθυμητών.
Σύνολο ωρών καθηγητών που προγραμματίστηκαν	Άθροισμα των προγραμματισμένων ωρών του κάθε διδάσκοντα
Έναρξη – Τέλος προβλημάτων που δεν έχουν συμπληρώσει ώρες διδασκαλίας	Εδώ φαίνονται τα μαθήματα που δεν έχουν ολοκληρώσει τις ώρες διδασκαλίας τους
Έναρξη – Τέλος καθηγητών που έχουν προγραμματιστεί ταυτόχρονα σε διαφορετική αίθουσα	
Έναρξη – Τέλος Μαθημάτων που έχουν δοθεί παραπάνω φορές από το συντελεστή που επιτρέπεται από μαθήματα άλλου εξαμήνου.	Μαθήματα ίδιου τμήματος και εξαμήνου μπορούν να προγραμματιστούν για συνδιδασκαλία όταν έχουν τον ίδιο αριθμό τμημάτων. Όταν ξεπεραστεί αυτός ο αριθμός τότε έχουμε πρόβλημα και εμφανίζεται ενημερωτική ένδειξη.
Έναρξη - Τέλος Παραπάνω ωρών προγραμματισμού καθηγητή.	Αν ξεπεραστεί το σύνολο των επιθυμητών ωρών του καθηγητή στον προγραμματισμό του για διδασκαλία τότε έχουμε αυτή την ένδειξη.
Έναρξη – Τέλος καθηγητών που έχουν	Όταν προγραμματιστεί ο καθηγητής σε μέρα

προγραμματιστεί σε αρνητικές προτιμήσεις τους.	και ώρα που έχει δηλώσει απόλυτα αρνητική προτίμηση (δηλαδή κόκκινο χρώμα).
Έναρξη – Τέλος Καθηγητών που έχουν προγραμματιστεί για μάθημα που δεν έχουν δηλώσει	
Έναρξη – Τέλος Μαθημάτων που έχουν προγραμματιστεί παραπάνω από το επιτρεπτό όριο	Μαθήματα τα οποία έχουν υπερβεί το γινόμενο Ώρες ανά εβδομάδα X Αριθμό Τμημάτων
Έναρξη – Τέλος Μαθημάτων που έχουν προγραμματιστεί αλλά δεν έχουν συμπληρώσει καθηγητές	Μαθήματα τα οποία έχουν παραπάνω από 1 καθηγητές για διδασκαλία και δεν έχουν προγραμματιστεί όλοι σε κάποιο διάστημα.
Έναρξη – Τέλος Μαθημάτων που έχουν προγραμματιστεί και έχουν παραπάνω καθηγητές από τον κανονικό αριθμό	Μαθήματα τα οποία έχουν προγραμματιστεί για διδασκαλία, και έχουν προγραμματιστεί παραπάνω καθηγητές.
Έναρξη – Τέλος Παραπάνω καθηγητές για εισηγήσεις	Όταν ένα μάθημα έχει περισσότερους από ένα εισηγητές

b. Πρόγραμμα ανά τμήμα και εξάμηνο

Ο πίνακας αυτός μας δείχνει το πρόγραμμα παραδόσεων των τμημάτων. Η αριστερή στήλη έχει τις ώρες διδασκαλίας, και η γραμμή αμέσως μετά από κάθε τμήμα έχει τα μαθήματα που προγραμματίστηκαν ανά εξάμηνο και την μέρα σε κάθε εξάμηνο. Τα περιεχόμενα του κάθε κελιού είναι χώρος διδασκαλίας – μάθημα – διδάσκων ή διδάσκοντες.

c. Ατομική καρτέλα καθηγητή με μαθήματα που προγραμματίστηκε

Εδώ φαίνεται ο προγραμματισμός του κάθε καθηγητή καθώς και τα ωράρια προτίμησης του. Για τον κάθε καθηγητή έχουμε ένα πίνακα με στήλες όσες οι μέρες λειτουργίας του τμήματος και ώρες όσες το ωράριο λειτουργίας του τμήματος.

d. Προγραμματισμός χώρων διδασκαλίας

Εδώ έχουμε για τον κάθε χώρο διδασκαλίας για κάθε χρονική στιγμή ποιο μάθημα έχει προγραμματιστεί για διδασκαλία και από ποιους καθηγητές.

e. Κάλυψη αιθουσών

Εδώ έχουμε την κάλυψη αιθουσών, δηλαδή με μια γρήγορη ματιά μπορούμε να δούμε για κάθε χρονική στιγμή, σε κάθε χώρο διδασκαλίας, ποιο τμήμα έχει προγραμματιστεί για διδασκαλία. Τα περιεχόμενα αυτού του πίνακα είναι το γράμμα κάλυψης και το χρώμα κάλυψης του τμήματος που έχουμε αναφέρει στην παράγραφο 6.1.4.

f. Χειροκίνητες καταχωρήσεις

Επειδή κανένα πρόγραμμα δεν μπορεί να έχει επιτυχία 100% έχουμε δώσει το δικαίωμα στο χρήστη να μπορεί να προγραμματίσει χειροκίνητα κάποια μαθήματα και καθηγητές. Έτσι λοιπόν σε αυτόν τον πίνακα φαίνονται όλες οι χειροκίνητες καταχωρήσεις. Για να γίνει μια χειροκίνητη καταχώρηση πρέπει να επιλέξουμε όλα τα δεδομένα που βρίσκονται κάτω από αυτό τον πίνακα και στη συνέχεια να πατήσουμε το κουμπί καταχώρηση. Έτσι λοιπόν όταν ξεκινάμε για προγραμματισμό του προγράμματος παραδόσεων μπορούμε να λάβουμε υπόψη μας αυτές τις καταχωρήσεις. Η διαδικασία αυτή φαίνεται στις παραγράφους που ακολουθούν.

2. Ξεκίνα από την αρχή

Ενεργοποιώντας αυτή την επιλογή θα λάβουμε μια ερώτηση για το αν θέλουμε να κρατήσουμε τις καταχωρήσεις της μνήμης ή να διαγραφούν αυτές οι καταχωρήσεις από τη μνήμη. Αν επιλέξουμε να διαγραφούν οι καταχωρήσεις τότε ο αλγόριθμος ξεκινάει όπως περιγράψαμε στο λογικό διάγραμμα του προγράμματος παραδόσεων. Αν επιλέξουμε να μην διαγραφούν, θα τρέξει ο ίδιος αλγόριθμος, αλλά με τη διαφορά ότι δεν θα πειραχτούν οι ήδη υπάρχουσες καταχωρήσεις. Αυτό ισχύει είτε αυτές είναι χειροκίνητες είτε από πρόγραμμα που έχει ήδη υπολογιστεί με κάποια δεδομένα.

3. Συνέχεια

Με αυτό τον τρόπο ο αλγόριθμος του ΩΠΠ χρησιμοποιεί τις καταχωρήσεις της μνήμης χωρίς να τις μεταβάλλει, είτε αυτές είναι χειροκίνητες είτε παλιό πρόγραμμα.

4. Προβλήματα αλγορίθμου

Επιλέγοντας προβλήματα αλγορίθμου έχουμε την εμφάνιση με τα αποτελέσματα του πίνακα “Παρατηρήσεις και προβλήματα που βρέθηκαν μετά την υλοποίηση του προγράμματος”.

5. Επανεμφάνιση προγράμματος

Τέλος με αυτή την επιλογή έχουμε επανεμφάνιση όλων των δεδομένων του προγράμματος παραδόσεων.

ii. Πρόγραμμα εξετάσεων

Με αυτή την επιλογή εμφανίζεται η φόρμα με τα δεδομένα του προγράμματος εξετάσεων. Έτσι λοιπόν παρατηρούμε ότι έχουμε στη διάθεση μας πίνακες με: “Παρατηρήσεις και προβλήματα του προγράμματος εξετάσεων”, “Πρόγραμμα εξετάσεων ανά Τμήμα και Εξάμηνο”, “Ατομική καρτέλα καθηγητή με μαθήματα που προγραμματίστηκε”, “Κάλυψη αιθουσών” και τέλος “Πίνακας επιτηρήσεων”. Τα δεδομένα αυτών των πινάκων αναλύονται ως εξής:

1. Παρατηρήσεις και προβλήματα του προγράμματος εξετάσεων

Σε αυτό τον πίνακα έχουμε τις ενδείξεις που ενημερώνεται ο χρήστης, όπως και στο πρόγραμμα παραδόσεων, για τα προβλήματα και τα δεδομένα που είναι απαραίτητα για την υλοποίηση του προγράμματος εξετάσεων. Τα δεδομένα αυτού του πίνακα φαίνονται στον πίνακα που ακολουθεί

Τίτλος	Επεξήγηση
Ποσοστό αναλογία ώρας διδασκαλίας/ ώρα εξέτασης	Αντιστοιχία ώρας επιτήρησης σε πόσες ώρες διδασκαλίας αντιστοιχεί. Είναι γενικό για όλη τη σχολή και για κάθε τμήμα ξεχωριστά.
Κατώφλι απόφασης αλγορίθμου επιτηρήσεων	Σημείο κάτω από το οποίο δεν προγραμματίζει για επιτηρήσεις καθηγητές. Είναι γενικό για όλη τη σχολή και για κάθε τμήμα ξεχωριστά
Σύνολο αιθουσών που χρειάζονται για την εξεταστική	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Σύνολο μαθημάτων που πρέπει να εξεταστούν	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Σύνολο ωρών ανά εβδομάδα	Σύνολο ωρών επιθυμητών για διδασκαλία των καθηγητών. Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά

Σύνολο καθηγητών που προγραμματίστηκαν για επιτήρηση	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Σύνολο καθηγητών που προγραμματίστηκαν για εισήγηση	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Σύνολο μαθημάτων που προγραμματίστηκαν στην εξεταστική	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Σύνολο διδασκόντων που συμμετέχουν στην εξεταστική	Είναι γενικό για όλα τα τμήματα και για κάθε τμήμα ξεχωριστά
Έναρξη – Τέλος Μαθημάτων που δεν προγραμματίστηκαν	Μαθήματα που συμμετέχουν στην εξεταστική και δεν προγραμματίστηκαν για εξέταση
Έναρξη – Τέλος Μαθημάτων που προγραμματίστηκαν και υπάρχει ανεπαρκής χωρητικότητα	Εδώ εμφανίζονται προβλήματα με μαθήματα των οποίων οι θέσεις, των αιθουσών που έχουν προγραμματιστεί για να διεξαχθεί η εξέτασης τους, δεν επαρκούν.
Έναρξη – Τέλος Μαθημάτων που προγραμματίστηκαν την ίδια χρονική στιγμή σε ίδια αίθουσα	
Έναρξη – Τέλος Μαθημάτων που προγραμματίστηκαν παραπάνω καθηγητές από το επιτρεπόμενο όριο	Εδώ φαίνονται τα μαθήματα τα οποία έχουν παραπάνω επιτηρητές από το αναμενόμενο
Έναρξη – Τέλος Μαθημάτων που δεν προγραμματίστηκαν όλοι οι καθηγητές για τις εξετάσεις	Εδώ φαίνονται τα μαθήματα τα οποία έχουν λιγότερους επιτηρητές από το αναμενόμενο

2. Πρόγραμμα εξετάσεων ανά Τμήμα και Εξάμηνο

Σε αυτόν τον πίνακα φαίνεται το πρόγραμμα των εξετάσεων του κάθε τμήματος. Σε κάθε κελί δεδομένων φαίνεται το μάθημα, ο εισηγητής και σε ποιες αίθουσες έχει προγραμματιστεί να γίνει η εξέταση.

3. Κάλυψη αιθουσών

Εδώ φαίνεται η κάλυψη των αιθουσών, όχι όλης της σχολής αλλά αυτών που συμμετέχουν στην εξεταστική. Όπως και στο πρόγραμμα παραδόσεων έτσι και εδώ έχουμε σε κάθε κελί χρώμα και γράμμα κάλυψης τμήματος.

4. Πίνακας επιτηρήσεων

Εδώ πέρα φαίνεται αναλυτικά ο κάθε καθηγητής πότε και που έχει προγραμματιστεί. Έτσι λοιπόν έχουμε μια στήλη που λέει σύνολο επιθυμητών ωρών ανά εβδομάδα, σύνολο εισηγήσεων που έχει και σύνολο επιτηρήσεων που έχει. Σε κάθε κελί δεδομένων αν έχει προγραμματιστεί για εισήγηση υπάρχει η λέξη Εισ. , Το τμήμα του μαθήματος, την ώρα της εξέτασης και οι αίθουσες που θα εξεταστεί το μάθημα. Κάτι αντίστοιχο είναι και για την επιτήρηση μόνο που η λέξη είναι Επιτ., όχι όλες οι αίθουσες εξέτασης του μαθήματος αλλά αυτή που θα τον προγραμματίσει ο αλγόριθμος. Αξίζει να επισημάνουμε ότι όσα κελιά έχουν χρώμα κάλυψης τμήματος σημαίνει ότι έχει προγραμματιστεί την αντίστοιχη μέρα σε μάθημα ή μαθήματα ίδιου τμήματος για εισήγηση ή επιτήρηση (δεν έχει σημασία), ενώ όσα κελιά είναι λευκά σημαίνει ότι έχει προγραμματιστεί την αντίστοιχη ημέρα σε πάνω από 2 διαφορετικά τμήματα για εισήγηση ή επιτήρηση. Αυτό μας διευκολύνει με μια γρήγορη ματιά να δούμε αντίστοιχα επιτηρήσεις και εισηγήσεις που έχει ο καθηγητής σε κάθε τμήμα.

d. Εκτυπώσεις

Στο σημείο αυτό έχει ολοκληρωθεί η εφαρμογή και το μόνο που απομένει είναι η

εκτύπωση των αποτελεσμάτων. Έτσι μπορούμε να εκτυπώσουμε το πρόγραμμα παραδόσεων για όλα τα τμήματα, το πρόγραμμα παραδόσεων όλων των χώρων διδασκαλίας, το πρόγραμμα παραδόσεων όλων των διδασκόντων της σχολής, την κάλυψη των χώρων διδασκαλίας κατά τη διάρκεια των παραδόσεων, το πρόγραμμα εξετάσεων όλων των τμημάτων, το πρόγραμμα επιτηρήσεων όλων των τμημάτων, και τέλος την κάλυψη αιθουσών κατά τη διάρκεια εξεταστικής. Οι εκτυπώσεις αυτές μπορούν να γίνουν και επιλεκτικά από το χρήστη.

e. Εργασίες

Σε αυτή την επιλογή ο χρήστης έχει τη δυνατότητα να αποθηκεύσει το ωρολόγιο πρόγραμμα και το πρόγραμμα εξετάσεων στην Βάση δεδομένων, καθώς και τη διαγραφή του από τη μνήμη.

6. Συμπεράσματα

Η υλοποίηση της συγκεκριμένης εφαρμογής ξεκίνησε τον Απρίλιο του 2002. Στη αρχή στόχος ήταν η εύρεση και η υλοποίηση μόνο του αλγορίθμου του ΩΠΠ. Στη συνέχεια όμως προέκυψε η ανάγκη, για μια ολοκληρωμένη εφαρμογή που θα καταστρώνει και το ΩΠΕ. Οι βασικές γνώσεις που υπήρχαν για τον προγραμματισμό σε Turbo C,C++ από αντίστοιχα μαθήματα που διδάχθηκαν στη Σχολή, καθώς και σχεδίαση – διαχείριση ΒΔ βοήθησαν αρκετά στην υλοποίηση της εφαρμογής μας. Το χρονικό διάστημα για την δημιουργία των πρώτων πειραματικών εφαρμογών στο περιβάλλον της Visual C++ 6.0 ήταν περίπου 2 μήνες. Τα προβλήματα που έπρεπε να αντιμετωπιστούν ήταν η κατανόηση των δυνατοτήτων της Visual C++ και κυρίως των γραφικών με τη χρήση Active X στοιχείων. Σε αυτό βοήθησαν πολύ τα βιβλία βλέπε βιβλιογραφία [1], το οποίο θεωρώ ιδανικό και βασικό για κάποιον που θέλει να ασχοληθεί σοβαρά με τον προγραμματισμό, και [2] όπου έχει απλές και κατανοητές εφαρμογές για τη χρήση βασικών Active X αντικειμένων, καθώς και άλλων αντικειμένων της MFC (Microsoft Foundation Classes) που χρησιμοποιήθηκαν στην εφαρμογή μας, και είχαν σαν αποτέλεσμα την συμπίκνωση – απλούστευση του κώδικα.

Πρέπει να σημειωθεί ότι η εφαρμογή θα μπορούσε να εμπλουτιστεί με πολλά άλλα χαρακτηριστικά. Υπάρχουν πολλές δυνατότητες επέκτασης, όπως πρόσβαση από το διαδίκτυο για καταχώρηση ατομικών δεδομένων του κάθε χρήστη καθώς και ενημέρωση του για το πρόγραμμα των εξετάσεων ή το ωρολόγιο, την δημιουργία αλγορίθμου αυτόματης επιλογής διδακτικού προσωπικού χωρίς την ταλαιπωρία των χειροκίνητων επιλογών.

Αυτό που πρέπει να προσέξει ο χρήστης της εφαρμογής είναι η προσεκτική καταχώρηση τόσο των δεδομένων, όσο και των ρυθμίσεων του κάθε αλγορίθμου.

Το σύστημα θεωρείται αρκετά ανοικτό και εύκολο στην επεκτασιμότητα. Αυτό ισχύει για την εφαρμογή και για τη ΒΔ. Η Βάση Δεδομένων που χρησιμοποιείται, η οποία είναι σε Access 2000 μπορεί πολύ εύκολα να μετατραπεί σε επόμενη έκδοση της Access, ή ακόμα και σε έναν SQL Server ή άλλο σύστημα Διαχείρισης Βάσης Δεδομένων. Μπορούν να τοποθετηθούν πρόσθετα πεδία, πίνακες και λειτουργίες στη Βάση Δεδομένων χωρίς ιδιαίτερη δυσκολία. Για την εξέλιξη της εφαρμογής υπάρχουν κάποιες ιδέες οι οποίες προήλθαν με την πάροδο του χρόνου. Έτσι λοιπόν εκτός τον γραμμικό τρόπο επίλυσης που επιλέξαμε σε όλους μας τους αλγορίθμους, μπορούμε να έχουμε πολλές λύσεις με γεννήτρια τυχαίων αριθμών που θα επηρεάζει κάθε φορά διαφορετικά βασικά στοιχεία του αλγορίθμου. Μερικά από αυτά είναι η τυχαία αρχικοποίηση ημέρας και ώρας έναρξης στην αναζήτηση αποτελεσμάτων, τυχαία επιλογή διδασκόντων, επέμβαση με αλλαγές σε ήδη υπάρχον πρόγραμμα ΩΠΠ και ΩΠΕ. Επίσης παρακάτω αναφέρονται κάποια συμπεράσματα τα οποία προέκυψαν κατά την υλοποίηση και τα οποία πιστεύουμε να βοηθήσουν όποιον ενδιαφερθεί να ασχοληθεί με τον προγραμματισμό:

- Όταν έχουμε αποτελέσματα τα οποία πηγάζουν από κάποια αποθηκευμένα δεδομένα, καλό είναι να τα καταχωρούμε μόνο μια φορά. Αυτό μας διευκολύνει στο να επεκτείνουμε ποιο εύκολα την εφαρμογή μας, χωρίς να πρέπει να θυμόμαστε πόσα πεδία πρέπει να ενημερώσουμε αφού κάποια δεδομένα μπορούν να προέλθουν μετά από λίγες πράξεις.

- Οι ονομασίες των αντικειμένων καλό είναι να είναι ενδεικτικές για το που ανήκουν και που αναφέρονται. Για παράδειγμα στην εφαρμογή χρονοπρογραμματισμός όλες οι φόρμες εμφάνισης δεδομένων ορίζονται ως αντικείμενα CForm. Επίσης, το όνομα των εξωτερικών μεταβλητών ή συναρτήσεων που αναφέρονται στο πρόγραμμα των εξετάσεων, αρχίζει με FE(For Exams).

- Η χρήση define συμβολικών ονομάτων είναι ότι το πιο σημαντικό όπως ξέρουμε στον προγραμματισμό. Στη συγκεκριμένη εφαρμογή χρειάστηκαν 70 συμβολικά ονόματα.

- Επίσης σημαντική ήταν η χρησιμότητα των δυναμικών λιστών, που όπως ανάφερα πολύ σύντομα έχουν μεγάλη χρησιμότητα επειδή ρυθμίζουμε ανά πάσα στιγμή τον όγκο της μνήμης που καταλαμβάνει η εφαρμογή μας και είναι ανοιχτή σε αρκετά μεγάλο όγκο δεδομένων. Είναι πολύ καλύτερο σε σχέση με την στατική δέσμευση της μνήμης όπως όταν δημιουργούμε πίνακες δεδομένων που όπως ξέρουμε δεν μεταβάλλονται δυναμικά.

- Τέλος η σημασία των εξαιρέσεων είναι ένα πολύ σημαντικό κομμάτι στον προγραμματισμό. Δεν νοείται σημαντική εφαρμογή χωρίς τη χρήση συναρτήσεων ανίχνευσης εξαιρέσεων. Στην εφαρμογή χρησιμοποιείται σε μεγάλο ποσοστό η ανίχνευση των γενικών εξαιρέσεων και όχι συγκεκριμένων. Υπάρχει για ενημέρωση του χρήστη το όνομα της συνάρτησης στο οποίο υπάρχει το πρόβλημα, ώστε να ανιχνευθεί- επιλυθεί γρήγορα. Έτσι δεν έχουμε μηνύματα του τύπου η εφαρμογή θα τερματίσει ή Unhandled Exception κ.οκ.

Βιβλιογραφία

- [1] Windows 2000 Programming with Visual C++ του Ben Ezzell ISBN 0-7821-2642-1
- [2] Εγχειρίδιο της Visual C++ 6 του Davis Chapman εκδόσεις Γκιούρδας ISBN 960-512-153-0
- [3] Learn Microsoft SQL Server 2000 ISBN 1-55622-763-9 του Jose A. Ramalho
- [4] Οδηγός της Microsoft για την Access 97 ISBN 960-332-074-9
- [5] “C: How to program” (second edition), H. M. Deitel, P. J. Deitel, , Prentice-Hall,1999.

Επίσης εξαιρετικής χρησιμότητας ήταν οι σημειώσεις από τα μαθήματα Γλώσσα Προγραμματισμού C, Τεχνολογία Λογισμικού, Αντικειμενοστρωφικός Προγραμματισμός, Αρχές Βάσεων Δεδομένων, Μικροϋπολογιστές, Νευρωνικά Δίκτυα, Λειτουργικά Συστήματα, Θεωρία Πληροφορίας, Συστήματα Διασύνδεσης με Χρήση Η/Υ.

Ακρωνύμια

ΒΔ	Βάση Δεδομένων
CLSF	Combination Less Seats Free
CCLSF	Compress Combination Less Seats Free
ΩΠΠ	Ωρολόγιο Πρόγραμμα Παραδόσεων
ΩΠΕ	Ωρολόγιο Πρόγραμμα Εξετάσεων
ODBC	Open Database Connectivity
ΑΣΒΔ	Ανοιχτή Συνδεσιμότητα Βάσεων Δεδομένων
RDBMS	Relational Database Management System RDBMS
E-R-M	Entity Relationship Model
MFC	Microsoft Foundation Class
SQL	Structured Query Language
DLL	Dynamic Link Library

Παράρτημα Α.

Αποσπάσματα κώδικα της εφαρμογής σε C++.

```
void CForm_ProgrammFE::FindAllCombinationsWithLessSheetsClassroomsSheets()//Συνάρτηση Εύρεσης όλων των δυνατών συνδυασμών
//αιθουσών – θέσεων
εξετάσεων
{
    POSITION Pos;//Δείκτης λίστας
    CVT_Classroom *Cl;//Χώρος διδασκαλίας
    Struc_CombinationStrClassroomsSheets *Combination;
    BYTE Temp;
    CString StrCombination;//Αλφαριθμητικό σύγκρισης
    double NumOfLoops;//Αριθμός συνδυασμών
    char ch[32];
    int TotalSheets,TempSheets,TempNum;
    CList<CVT_Classroom *,CVT_Classroom*>GoodListCl,TempList;

try
{
    while(m_ListCombinClassroomsSheets.GetCount(>0)//διαδικασία αρχικοποίησης λίστας συνδυασμών
    {
        m_ListCombinClassroomsSheets.RemoveHead();
    }

    //Εναρξη εύρεση καλύτερων λύσεων
    StrCombination.Empty();
    NumOfLoops=pow(2,FEGetTotalClassrooms()-1);//(2^Σύνολο αιθουσών)-1
    TotalSheets=0;
    while(NumOfLoops>0)
    {
        while(GoodListCl.GetCount(>0)
        {
            GoodListCl.RemoveHead();
        }

        while(TempList.GetCount(>0)
        {
            TempList.RemoveHead();
        }

        StrCombination=(CString)(_ui64toa(NumOfLoops,ch,2));//Μετατροπή αριθμού σε CString
        TempSheets=0;
        TempNum=0;
        while(TempNum<StrCombination.GetLength())//Όσο η προσωρινή μεταβλητή είναι μικρότερη του μήκους του
αλφαριθμητικού
        {
            if(StrCombination.GetAt(TempNum)=='1')
            {
                Cl=FindClassFromPosition(FEGetTotalClassrooms()-
StrCombination.GetLength()+TempNum,ConditionExams);
                GoodListCl.AddHead(Cl);
            }
            TempNum++;
        }
        Combination=new Struc_CombinationStrClassroomsSheets;//Δημιουργία δομής Struc_CombinationStrClassroomsSheets
        Combination->ClassroomsLong=NumOfLoops;//Συνδυασμός αιθουσών
        Combination->Sheets=GetTotalSheetsFromList(&GoodListCl,ConditionExams);//Σύνολο αιθουσών συνδυασμού
        m_ListCombinClassroomsSheets.AddHead(Combination);//Καταχώρηση δομής στη λίστα συνδυασμών
        NumOfLoops--;
    }
    SortListCombineClassroomsSheetsOrderSheetsAsc();//Ταξινόμηση λίστας συνδυασμών αύξουσα ανά χωρητικότητα συνδυασμών
}
catch(...)
{
    MessageBox("Πρόβλημα στη συνάρτηση void
CForm_ProgrammFE::FindAllCombinationsWithLessSheetsClassroomsSheets()");
}

return;
}
```

```

void CForm_ProgrammFE::OnFefinishsupervices() //2ος Αλγόριθμος εύρεσης επιτηρήσεων ΩΠΕ
{
    float Higher,Temp;
    BYTE Priority,Day,Hour;
    CVT_Lessons *Less,*TempLess;
    CVT_Professor *Prof,*ProfHigher;
    CVT_Part *Part;
    POSITION Pos,PosProf;
    BOOL Found;
try
{
    if(GetTotalProfessorsProgrammedForIntroducing(NULL)>0)//Αν υπάρχουν εισηγητές
    {
        BeginWaitCursor();
        Higher=100;
        Pos=m_ListParts.GetHeadPosition();
        while(Pos!=NULL)
        {
            Part=m_ListParts.GetNext(Pos);
            ProfessorsPart=Part;
            Less=NULL;
            FindEfficientAndEdgeForSupervices(Part);
            Found=TRUE;
            while(TotalProfessorsForSupervices>0 && Found==TRUE)
            {
                Found=FALSE;
                PosProf=m_ListProfessorsOrderByCategory.GetHeadPosition();
                while(PosProf!=NULL)
                {
                    Prof=m_ListProfessorsOrderByCategory.GetNext(PosProf);
                    if( ( FEPermanent==TRUE && Prof->GetPermanent()==TRUE) ||
FEPermanent==FALSE) &&
                    FEGetProfessorHoursWantFromLessons(Prof,Part)>0)
                    {
                        Temp=(FEGetProfessorSupervices(Prof,Part)+1+FEGetProfessorIntroducings(Prof,Part)*
FEIntroducingsSupervices)/FEGetProfessorHoursWantFromLessonsOrByTheLaw(Prof,Part);
                        //κλάσμα υπολογισμού επιτηρήσεων που έχει προγραμματιστεί ο καθηγητής
                        //ως προς τις ώρες διδασκαλίας στο τμήμα που ερευνούμε
                        TempLess=FoundLessonForProgrammSupervisor(Prof,Part);//Εύρεση μαθήματος που
να χρειάζεται
                        //επιτήρηση και να είναι του τμήματος του
                        //καθηγητή.NULL αν δεν βρεθεί
                        if(Temp<Higher && TempLess!=NULL)
                        {
                            Less=TempLess;
                            ProfHigher=Prof;
                            Higher=Temp;
                            Found=TRUE;
                        }
                    }
                }
            }
            if(Less!=NULL)
            {
                TempLess=FoundLessonForProgrammSupervisor(Prof,Part);
                Less->SetProfessorProgrammedFE(ProfHigher);
                TotalProfessorsForSupervices--;
                Less=NULL;
                Higher=100;
            }
        }
        ShowResults();
        EndWaitCursor();
    }
    else
    {
        MessageBox("Δεν έχουν βρεθεί εισηγητές. Παρακαλώ τρέξτε πρώτα τον αλγόριθμο εύρεσης
εισηγήσεων", "", MB_ICONINFORMATION);
    }
}
catch(...)
{
}
}

```

```

        MessageBox("Πρόβλημα στη συνάρτηση void CForm_ProgrammFE::OnFefinishservices() ");
    }

return;
}

void CForm_TimeProgramming::RunAlgorithm(BYTE Try)//Αλγόριθμος ΩΠΠ
{
    BYTE Prefer_Lesson;
    POSITION PosProf;
    try
    {
        BeginWaitCursor();
        PosProf=m_ListProfessorsOrderByCategory.GetHeadPosition();
        while (PosProf!=NULL)
        //ΟΣΟ
        ΥΠΑΡΧΟΥΝ ΚΑΘΗΓΗΤΕΣ
        {
            ProfessorTimeProgrammingCurrent=m_ListProfessorsOrderByCategory.GetNext(PosProf);
            Prefer_Lesson=0;//ΤΡΕΧΟΥΣΑ ΠΡΟΤΙΜΗΣΗ ΜΑΘΗΜΑΤΟΣ
            //ΑΠΟ ΠΡΟΤΙΜΗΣΕΙΣ ΜΑΘΗΜΑΤΩΝ
            ΚΑΘΗΓΗΤΗ
            while ( (FTTGetProfessorLeftHours(ProfessorTimeProgrammingCurrent)>0) &&
                (Prefer_Lesson<ProfessorTimeProgrammingCurrent->GetTotalLessons()) )
            //ΟΣΟ ΥΠΑΡΧΕΙ ΥΠΟΛΟΙΠΟ ΩΡΩΝ ΤΟΥ ΚΑΘΗΓΗΤΗ ΚΑΙ ΟΣΟ ΥΠΑΡΧΟΥΝ ΜΑΘΗΜΑΤΑ
            ΕΠΙΛΟΓΗΣ
            //ΤΟΥ ΚΑΘΗΓΗΤΗ ΓΙΑ ΔΙΔΑΣΚΑΛΙΑ
            {
                LessonTimeProgrammingCurrent=
                (CVT_Lessons *) (ProfessorTimeProgrammingCurrent->GetLesson(Prefer_Lesson)); //ΜΑΘΗΜΑ ΠΡΟΤΙΜΗΣΗΣ
                //ΠΑΡΕ ΤΟ ΜΑΘΗΜΑ ΤΗΣ ΑΥΞΟΥΣΑΣ
                //ΠΡΟΤΙΜΗΣΗΣ ΤΟΥ
                if(LessonTimeProgrammingCurrent->GetLessSemester(>0)
                {
                    TotalMatchingTableSameLesson->Reset();
                    TotalMatchingTableSamePartSemester->Reset();
                    TotalMatchingTableNew->Reset();

                    CopyDataProfessorToMatchings();
                    //ΕΝΗΜΕΡΩΣΗ ΤΩΝ ΕΙΚΟΝΙΚΩΝ ΠΙΝΑΚΩΝ ΜΕ
                    //ΠΡΟΤΙΜΗΣΕΙΣ ΩΡΑΡΙΟΥ ΚΑΘΗΓΗΤΗ ΚΑΙ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ
                    CopyDataVTTotalMatchingSameLesson();
                    //ΕΝΗΜΕΡΩΣΗ ΤΟΥ TotalMatchingTableSameLesson
                    CopyDataVTTotalMatchingSamePartSemester();
                    //ΕΝΗΜΕΡΩΣΗ ΤΟΥ TotalMatchingTableSamePartSemester
                    CopyDataVTTotalMatchingNew();
                    //ΕΝΗΜΕΡΩΣΗ ΤΟΥ TotalMatchingTableNew
                    CopyDataMatchingsToMatchings();
                    //ΕΝΗΜΕΡΩΣΗ ΕΙΚΟΝΙΚΩΝ ΠΙΝΑΚΩΝ ΓΙΑ ΔΙΑΘΕΣΙΜΕΣ ΑΙΘΟΥΣΕΣ
                    //ΠΟΥ ΜΠΟΡΕΙ ΝΑ ΓΙΝΕΙ ΤΟ ΜΑΘΗΜΑ
                    //ΤΡΕΞΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΠΟΥ
                    ΕΡΕΥΝΟΥΜΕ
                    if(Try==FirstTry)
                    {
                        RunAlgorithmMaster();//ΠΡΩΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΕ
                        ΣΥΝΕΧΟΜΕΝΑ ΔΙΩΡΑ
                        //ΑΛΛΗΛΟΕΠΙΚΑΛΥΨΕΙΣ ΚΤΛ
                    }
                    else if(Try==SecondTry)
                    {
                        RunAlgorithmHoursProfessorsLessonsLeft();//ΔΕΥΤΕΡΟΣ
                        ΑΛΓΟΡΙΘΜΟΣ
                        //ΑΝΑΖΗΤΗΣΗ ΓΙΑ ΜΙΑΣ ΩΡΑΣ
                    }
                }
            }
        }
    }
}

```

```
                Prefer_Lesson++;
                //ΠΑΡΕ ΕΠΟΜΕΝΟ ΜΑΘΗΜΑ ΠΡΟΤΙΜΗΣΗΣ ΚΑΘΗΓΗΤΗ
            }
        }
    catch(...)
    {
        MessageBox("Λάθος στη συνάρτηση void CForm_TimeProgramming::RunAlgorithm(BYTE
Try)");
        //        pE->ReportError();
        //        //throw;
    }
    EndWaitCursor();
    return;
}
```

Παράρτημα Β.

Δείγματα εκτυπώσεων της εφαρμογής

Η διαδικασία της εκτύπωσης με τη βοήθεια της MFC είναι πλέον απλή για τον προγραμματιστή. Μέσω των συναρτήσεων **BOOL OnPreparePrinting(CPrintInfo* pInfo), void OnPrint(CDC* pDC, CPrintInfo* pInfo), void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo), void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo)** κ.α. μας δίνεται η δυνατότητα, να επικοινωνήσουμε με τη συσκευή και να την ενημερώσουμε, μέσω της τάξης **CDC***, π.χ. για το σύνολο των σελίδων που πρόκειται να σταλούν, το μέγεθος των δεδομένων κ.α.. Επίσης μπορούμε να πάρουμε πληροφορίες για την μονάδα στην οποία πρόκειται να σταλούν τα δεδομένα, μέσω της τάξης **CPrintInfo**, όπως να μετρήσουμε την περιοχή εκτύπωσης, ύψος – μήκος κάθε φορά που προετοιμάζεται η εφαρμογή να εκκινήσει εκτύπωση ή να προβεί σε προεπισκόπηση εκτύπωσης, την κατάσταση την οποία βρίσκεται η συσκευή μας κ.α.. Τα δεδομένα αυτά μπορούμε να τα έχουμε σε ότι μονάδα μέτρησης επιθυμούμε pixel, cm κ.ο.κ.

Με αυτό τον τρόπο, και με λίγες πράξεις, μπορούμε να υπολογίσουμε το μέγεθος της γραμματοσειράς που θα χρησιμοποιήσουμε σε κάθε εκτύπωση, τις διαστάσεις πινάκων δεδομένων κ.α. (Μετά από υπολογισμούς καταλήξαμε ότι μια ικανοποιητική αναλογία ύψος/μήκος γραμματοσειράς είναι το 2). Έτσι λοιπόν, έχουμε τη δυνατότητα να εκτυπώσουμε το ΩΠΠ των Τμημάτων, τον προγραμματισμό των χώρων διδασκαλίας στο ΩΠΠ, τον προγραμματισμό των διδασκόντων στο ΩΠΠ, την κάλυψη των χώρων διδασκαλίας στο πρόγραμμα παραδόσεων, το ΩΠΕ των Τμημάτων, την κατάσταση εισηγήσεων - επιτηρήσεων των Τμημάτων κατά τη διάρκεια του ΩΠΕ, το διάγραμμα κάλυψης αιθουσών κατά τη διάρκεια της εξεταστικής. Δείγματα εκτυπώσεων, με τη σειρά που αναφέραμε, φαίνονται στις επόμενες σελίδες. Αξίζει να παρατηρήσετε τη σμίκρυνση την οποία γίνεται στη γραμματοσειρά, όταν προγραμματίζονται δυο μαθήματα ίδιου εξαμήνου και τμήματος για συνδιδασκαλία την ίδια ημέρα και ώρα, όπου το κελί δεδομένων διαιρείται σε δυο ίσα μέρη, και τα μαθήματα στοιχίζονται το ένα αριστερά και το άλλο δεξιά. Όταν ο αριθμός των μαθημάτων είναι παραπάνω από δύο τότε το κελί δεδομένων διαιρείται σε τέσσερα μέρη.

ΠΕΡΙΕΧΟΜΕΝΑ ΒΙΒΛΙΟΥ

7. **Εισαγωγή**
8. **Θεμελιώδεις έννοιες και ορολογία της θεωρίας των Βάσεων Δεδομένων**
 - a. Η έννοια της Βάσης Δεδομένων
 - b. Πρότυπο ODBC
9. **Αντικειμενοστραφής Προγραμματισμός – Visual C++ 6.0**
 - a. Εισαγωγή
 - b. Κατηγορίες της MFC
 - c. Εξαιρέσεις
 - d. Active X Control
10. **Σχεδίαση Αλγορίθμου Ωρολογίου Προγράμματος**
 - a. Περιγραφή
 - b. Λογική αλγορίθμου ωρολογίου προγράμματος
 - c. Λογικό διάγραμμα αλγορίθμου ωρολογίου προγράμματος
11. **Σχεδίαση Αλγορίθμου Προγράμματος εξετάσεων**
 - a. Περιγραφή
 - b. Λογική αλγορίθμου προγράμματος εξετάσεων
 - i.Πρώτο μέρος αλγορίθμου προγράμματος εξετάσεων (εύρεση εισηγήσεων)
 1. Αλγόριθμος CLSF (Combination Less Seats Free)
 2. Αλγόριθμος CCLSF(Compress Combination Less Seats Free)
 - ii.Δεύτερο μέρος αλγορίθμου προγράμματος εξετάσεων(εύρεση επιτηρήσεων)
 1. 1ος Αλγόριθμος δεύτερου μέρους προγράμματος εξετάσεων
 2. 2ος Αλγόριθμος δεύτερου μέρους προγράμματος εξετάσεων
 - c. Λογικό διάγραμμα αλγορίθμου προγράμματος εξετάσεων
12. **Χρήση της εφαρμογής**
 - a. Ενημέρωση στοιχείων
 - i.Ενημέρωση στοιχείων Τμημάτων
 - ii.Ενημέρωση στοιχείων χώρων διδασκαλίας
 - iii.Ενημέρωση στοιχείων μαθημάτων
 - iv.Ενημέρωση στοιχείων καθηγητών
 - b. Ρυθμίσεις
 - i.Βασικές ρυθμίσεις
 - ii.Πρόγραμμα παραδόσεων
 - iii.Πρόγραμμα εξετάσεων

- c. Χρονοπρογραμματισμός
 - i. Πρόγραμμα παραδόσεων
 - ii. Πρόγραμμα εξετάσεων
- d. Εκτυπώσεις
- e. Εργασίες

13. Συμπεράσματα

Παραρτήματα

- A. Δείγματα κώδικα Visual C++ 6.0
- B. Δείγματα εκτυπώσεων

Βιβλιογραφία

Ακρωνύμια