

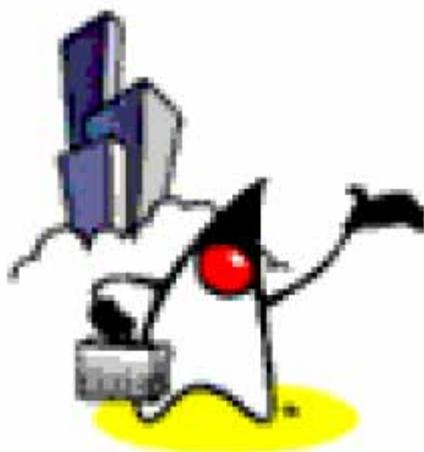


ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΗ & ΑΝΑΠΤΥΞΗ

ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ (jA.E.S.)

ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA



ΜΑΝΑΚΟΣ ΑΝΑΣΤΑΣΙΟΣ

ΜΑΡΤΙΟΣ 2005

Πτυχιακή εργασία, μέρος των απαιτήσεων του τμήματος
Τηλεπληροφορικής & Διοίκησης.

**ΣΧΕΔΙΑΣΗ & ΑΝΑΠΤΥΞΗ
ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ (jA.E.S.)
ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA**

Μανάκος Αναστάσιος
Μάρτιος 2005

Φίλε αναγνώστη/αναγνώστρια,

Όπως είναι αυτονόητο ΔΕΝ ΜΠΟΡΕΙΣ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΣΕΙΣ Ή ΝΑ ΑΝΑΠΑΡΑΓΕΙΣ ΜΕΡΟΣ Ή ΚΑΙ ΟΛΟΚΛΗΡΗ ΑΥΤΗ ΤΗΝ ΕΡΓΑΣΙΑ ΓΙΑ ΤΗΝ ΑΠΟΚΟΜΙΔΗ ΠΡΟΣΩΠΙΚΟΥ ΟΦΕΛΟΥΣ ΧΩΡΙΣ ΤΗΝ ΕΓΓΡΑΦΗ ΑΔΕΙΑ ΜΟΥ. ΣΕ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ ΕΠΙΦΥΛΑΣΣΟΜΑΙ ΓΙΑ ΚΑΘΕ ΝΟΜΙΜΟ ΔΙΚΑΙΩΜΑ ΜΟΥ.

*Ευχαριστώ,
Μανάκος Αναστάσιος*

Για διάφορα σχόλια ή πιθανές ερωτήσεις επικοινωνήστε μαζί μου : a_manakos@yahoo.gr

ΠΕΡΙΛΗΨΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

ΣΧΕΔΙΑΣΗ & ΑΝΑΠΤΥΞΗ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ (jA.E.S.) ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA

ΜΑΡΤΙΟΣ 2005

ΜΑΝΑΚΟΣ ΑΝΑΣΤΑΣΙΟΣ

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου

Επιβλέπων Καθηγητής: Δρ. ΔΗΜΗΤΡΙΟΣ ΛΙΑΡΟΚΑΠΗΣ

Η παρούσα πτυχιακή εργασία περιγράφει την ανάπτυξη μιας εφαρμογής ενός αυτόματου συστήματος εξέτασης βασισμένο σε ερωτήσεις πολλαπλής επιλογής. Η εφαρμογή υλοποιήθηκε με την αντικειμενοστραφή γλώσσα προγραμματισμού Java και χρησιμοποίησε το Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων MySQL.

Το σύστημα είναι κατασκευασμένο σε παραθυρικό περιβάλλον (*java swing*) με αποτέλεσμα να παρέχει στον χρήστη ένα ικανοποιητικό και ευχάριστο περιβάλλον εργασίας. Η διεξαγωγή της ηλεκτρονικής εξέτασης γίνεται με ερωτήσεις πολλαπλής επιλογής όπου ο εξεταζόμενος καλείται να τις απαντήσει μέσα σε ένα προκαθορισμένο χρονικό διάστημα. Οι ερωτήσεις που απαρτίζουν την κάθε εξέταση επιλέγονται “τυχαία” με την χρήση ενός μαθηματικού αλγορίθμου από μια μεγάλη ομάδα ερωτήσεων που θα πρέπει να έχει εισαχθεί στο σύστημα για κάθε εξεταζόμενο μάθημα.

Η βασική οθόνη της εξέτασης αποτελείται από δύο παράπλευρα τμήματα. Το αριστερό τμήμα περιέχει το σύνολο των ερωτήσεων της εξέτασης και το δεξιό τμήμα περιέχει την εκάστοτε επιλεγμένη ερώτηση όπου ο εξεταζόμενος θα επιλέξει την σωστή απάντηση. Το τμήμα με το σύνολο των ερωτήσεων περιέχει ενδείξεις για το ποιες ερωτήσεις έχουν ήδη απαντηθεί ή όχι. Ο εξεταζόμενος έχει την δυνατότητα να απαντά τις ερωτήσεις ακολουθιακά ή να επιλέγει κάποια συγκεκριμένη από το υπάρχον σύνολο. Στο τέλος της εξέτασης ο εξεταζόμενος ενημερώνεται αναλυτικά για τις σωστές ή λανθασμένες απαντήσεις του και για το ποσοστό επιτυχίας του. Επίσης του δίνεται η δυνατότητα αποθήκευσης της εξέτασης σε αρχείο CSV (comma separated values).

Η εισαγωγή των ερωτήσεων στο σύστημα γίνεται από τον χρήστη – καθηγητή είτε μεμονωμένα πληκτρολογώντας ο ίδιος τις διάφορες ερωτήσεις είτε φορτώνοντας αυτές από ένα υπάρχον αρχείο CSV. Επίσης ο καθηγητής μπορεί να διαγράψει ερωτήσεις και να δημιουργεί νέες με βάση κάποιες ήδη υπάρχουσες. Τέλος, μπορεί να εμφανίζει και να εκτυπώνει τα αποτελέσματα των εξετάσεων όλων των χρηστών – σπουδαστών που εξετάστηκαν στα μαθήματά του.

Η εφαρμογή υποστηρίζει και αρχείο διαμόρφωσης όπου μπορούν να καθοριστούν και να τροποποιηθούν διάφορες παράμετροι του συστήματος (π.χ. αλλαγή χρόνου εξέτασης κλπ). Το κείμενο της εργασίας περιέχει και μια εισαγωγή στην χρήση της γλώσσας Java και του Σχεσιακού Συστήματος Βάσης Δεδομένων MySQL και τον τρόπο διασύνδεσής τους.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	1
1. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA	3
1.1 Ιστορική ανασκόπηση της Java	3
1.2 Γιατί λοιπόν Java	4
1.3 Πιθανά μειονεκτήματα της Java	5
1.4 Εφαρμογές Java	5
2. ΕΠΕΞΗΓΗΣΗ ΕΝΝΟΙΩΝ	7
2.1 Εισαγωγή	7
2.2 Αντικειμενοστραφής Προγραμματισμός	7
2.3 Τάξη (Class) & Αντικείμενο (Object)	7
2.4 Υπερφόρτωση (Overloading) – Πολυμορφισμός (Polymorphism)	8
2.5 Κληρονομικότητα (Inheritance)	8
2.6 Διασύνδεση (Interface)	9
3. ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ, SQL κ' JDBC	10
3.1 Εισαγωγή	10
3.2 Λίγα λόγια για τις Σχεσιακές Βάσεις Δεδομένων	10
3.3 Διαχείριση Βάσεων Δεδομένων με την SQL	11
3.4 Το Σύστημα Βάσεων Δεδομένων MySQL	12
3.5 JDBC Driver	12
4. ΠΡΟΣΒΑΣΗ ΣΕ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΜΕ JAVA	14
4.1 Δημιουργία Βάσης Δεδομένων παραδείγματος	14
4.2 Δημιουργία σύνδεσης	16
5. ΠΕΡΙΓΡΑΦΗ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ	19
5.1 Η Ιδέα	19
5.2 Εισαγωγή στο jA.E.S.	20
5.3 Εξέταση μέσω του jA.E.S.	20
6. ΠΑΡΟΥΣΙΑΣΗ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ	21
6.1 Παρουσίαση jA.E.S.	21
6.2 Σπουδαστής βήμα προς βήμα	22
6.3 Καθηγητής βήμα προς βήμα	25
6.4 Υπερφόρτωση του αρχείου αρχικοποίησης της εφαρμογής	27

7. ΛΕΙΤΟΥΡΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ	29
7.1 Εισαγωγή	29
7.2 Χρήστες του jA.E.S.	29
7.2.1 Χρήστης (User)	30
7.2.2 Μαθητής (Student)	30
7.2.3 Καθηγητής (Professor)	31
7.2.4 Διαχειριστής (Administrator)	31
7.3 Μάθημα (Course)	31
7.4 Ερώτηση (Question)	32
7.5 Εξέταση (Examination)	33
7.6 Γεννήτρια Παραγωγής Τυχαίων Αριθμών (Generate Random Number)	34
7.7 Διαχειριστής ΒΔ (Database Manager)	34
7.8 Διαχειριστής Διαδικτύου (Internet Manager)	35
7.9 Διαχειριστής CSV (CSV Manager)	36
7.10 Αυτόματο Σύστημα Εξέτασης (Automatic Examination System)	36
7.11 Config	37
8. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ	40
8.1 Βάση Δεδομένων jA.E.S.	40
8.2 Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Model)	40
9. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΒΛΗΜΑΤΑ – ΒΕΛΤΙΩΣΕΙΣ	42
9.1 Συμπεράσματα από την εκπόνηση της παρούσας εργασίας	42
9.2 Προβλήματα κατά την εκπόνηση της παρούσας εργασίας	42
9.3 Προτάσεις βελτίωσης της παρούσας εργασίας	43
ΗΜΕΡΟΛΟΓΙΟ ΠΟΡΕΙΑΣ ΤΗΣ ΕΡΓΑΣΙΑΣ	44
ΒΙΒΛΙΟΓΡΑΦΙΑ	45
ΠΑΡΑΡΤΗΜΑ Α' – ΕΝΔΕΙΚΤΙΚΟΙ ΚΩΔΙΚΕΣ	47
Αντικείμενο Database Manager	47
Αντικείμενο Question	51
Αντικείμενο Course	56
Αντικείμενο Generate Random Numbers	60
Αντικείμενο Examination	61
Αντικείμενο Examination Main Menu	63
Αντικείμενο Timer Panel	66
Αντικείμενο Question List Panel	68
Αντικείμενο Question Multiple Choice Panel	71
Αντικείμενο Examination Dialog	74

Αφιέρωση

Στην μνήμη του παππού μου, “Μανάκο Αναστάσιο”. Μου λείπει πάρα πολύ,

Στον εαυτό μου...

Ευχαριστίες

Ευχαριστώ πολύ όσους με βοήθησαν στην περάτωση της πτυχιακής μου εργασίας και συγκεκριμένα,

Τον καθηγητή μου, Δρ. Δημήτριο Λιαροκάπη για την πολύτιμη βοήθεια που μου παρείχε, το άψογο πνεύμα συνεργασίας που εκδήλωσε και την εμπιστοσύνη που έδειξε στο άτομό μου.

Επιπλέον, ευχαριστώ την συνάδελφο και καλή μου φίλη Θεοδώρα Αναστασίου για όλους τους γνωστούς λόγους (ηθική υποστήριξη, παρηγοριά στις δύσκολες στιγμές).

Ακόμη, ευχαριστώ τους ανθρώπους που με δίδαξαν να προγραμματίζω. Ο προγραμματισμός είναι κατά μεγάλο μέρος επάγγελμα και δημιουργική εμπειρία, όσο και τεχνική.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, για όλα όσα μου έχουν προσφέρει αυτά τα χρόνια.

Αδύνατον λέει η λογική.

Ρισκοκίνδυνο λέει η πείρα.

Επίπονο λέει η υπερηφάνεια.

Προσπάθησε λέει το όνειρο.

*Αυτήν είναι η πρόσκληση να
κάνουμε το όνειρο πραγματικότητα.*

ΕΙΣΑΓΩΓΗ

Η πτυχιακή εργασία αποτελεί ένα από τα σημαντικότερα βήματα ενός τελειόφοιτου σπουδαστή. Η περάτωσή της, είναι ο καλύτερος τρόπος εφαρμογής των γνώσεων ενός σπουδαστή πάνω σε ένα συγκεκριμένο θέμα, το οποίο φυσικά προέρχεται από το γνωστικό αντικείμενο των σπουδών του.

Η παρούσα πτυχιακή, η οποία έχει τίτλο “Σχεδίαση & Ανάπτυξη Αυτόματου Συστήματος Εξέτασης (jA.E.S.) Στη Γλώσσα Προγραμματισμού Java”, έχει σκοπό να παρουσιάσει ένα νέο τρόπο εξέτασης και αξιολόγησης των σπουδαστών, ο οποίος δεν έρχεται να αντικαταστήσει τον παραδοσιακό γραπτό τρόπο. Αντίθετα, έρχεται να συμπληρώσει την παρούσα εκπαιδευτική αξιολόγηση. Ο νέος αυτός τρόπος επικοινωνίας και ανταλλαγής δεδομένων δε θα μπορούσε παρά να αποτελέσει ένα σημαντικό εργαλείο για την εκπαιδευτική κοινότητα, αφού ο τρόπος λειτουργίας του προσφέρει άπειρα πλεονεκτήματα τόσο για το διδακτικό προσωπικό όσο και για τους σπουδαστές. Οι νέες τεχνολογίες πρέπει να χρησιμοποιηθούν για να εμπλουτίσουν το μάθημα και να το κάνουν πιο ενδιαφέρον.

Στις σελίδες που ακολουθούν, γίνεται μια αναλυτική αναφορά σε όλα τα βήματα που χρειάστηκε να γίνουν ώστε να πραγματοποιηθεί η εφαρμογή jA.E.S. (Automatic Examination System in Java). Με μια σύντομη περιγραφή κάθε κεφαλαίου, η παρούσα πτυχιακή μελετά τα παρακάτω.

Στο **1^ο κεφάλαιο**, γίνεται μια περιγραφή σε θεωρητικό πλαίσιο που αφορά τη βασική εισαγωγή στη γλώσσα προγραμματισμού Java : ιστορική αναδρομή, τι είναι η Java, ποια τα πλεονεκτήματα και μειονεκτήματά της.

Το **2^ο κεφάλαιο**, παρουσιάζει τις βασικές αρχές του αντικειμενοστραφούς προγραμματισμού, όπως αυτές εφαρμόζονται στην Java.

Στο **3^ο κεφάλαιο**, γίνεται μια περιγραφή σε θεωρητικό πλαίσιο που αφορά τον τρόπο λειτουργίας των Σχεσιακών Βάσεων Δεδομένων. Επιπλέον, περιγράφεται το Σύστημα Βάσεων Δεδομένων MySQL και οι τρόποι με τους οποίους μπορεί να προσπελαστεί μια Βάση Δεδομένων με Java.

Στο **4^ο κεφάλαιο**, παρουσιάζεται μέσα από ένα παράδειγμα, ο τρόπος με τον οποίο δημιουργούνται Βάσεις Δεδομένων και Πίνακες στην MySQL μέσω των εντολών της SQL, καθώς επίσης και ο τρόπος με τον οποίο μπορεί να προσπελαστεί μια Βάση Δεδομένων με Java.

Το **5^ο κεφάλαιο**, αρχίζει καθορίζοντας την ιδέα πάνω στην οποία στηρίχτηκε η εφαρμογή jAES και στην συνέχεια ακολουθεί μια σύντομη εισαγωγή σ’ αυτήν. Έπειτα αναλύεται με λεπτομέρεια ο σκοπός της, ώστε ο αναγνώστης να κάνει μια πρώτη γνωριμία, μέσω της περιγραφής της ταυτότητάς της και του στόχου της. Στην συνέχεια, αναλύεται η ανάγκη δημιουργίας μιας εφαρμογής σαν την παρούσα.

Το **6^ο κεφάλαιο**, περιλαμβάνει την παρουσίαση των οθονών που απαρτίζουν την εφαρμογή, ώστε ο αναγνώστης να πάρει μια ιδέα του μοτίβου που ακολουθεί η εφαρμογή.

Στο **7^ο κεφάλαιο**, γίνεται μια αναλυτική περιγραφή στον λειτουργικό σχεδιασμό της εφαρμογής. Προσδιορίζονται τα κύρια αντικείμενα και εντοπίζονται μερικές από τις αλληλεξαρτήσεις τους.

Στο **8^ο κεφάλαιο**, παρουσιάζεται αναλυτικά το Σύστημα Διαχείρισης Βάσεων Δεδομένων πάνω στο οποίο στηρίζεται η όλη λειτουργία της εφαρμογής. Επίσης παρουσιάζεται και το Μοντέλο Οντοτήτων του συγκεκριμένου Συστήματος.

Στο **9^ο κεφάλαιο**, γίνονται κάποιες παρατηρήσεις και συμπεράσματα που αφορούν την παρούσα πτυχιακή, καθώς και προτάσεις αναβάθμισής της.

Στο **Παράρτημα Α**, παρουσιάζονται όλες οι πηγές, έντυπες και ηλεκτρονικές οι οποίες χρησιμοποιήθηκαν για την περάτωση της πτυχιακής εργασίας.

Τέλος στο **Παράρτημα Β**, παρατίθενται ορισμένοι ενδεικτικοί κώδικες των οποίων η παρουσίαση κρίθηκε απαραίτητη για την διαμόρφωση μιας πιο ξεκάθαρης εικόνας για το σύνολο της εργασίας.

ΚΕΦΑΛΑΙΟ 1 : Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA

1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΣΚΟΠΗΣΗ ΤΗΣ JAVA

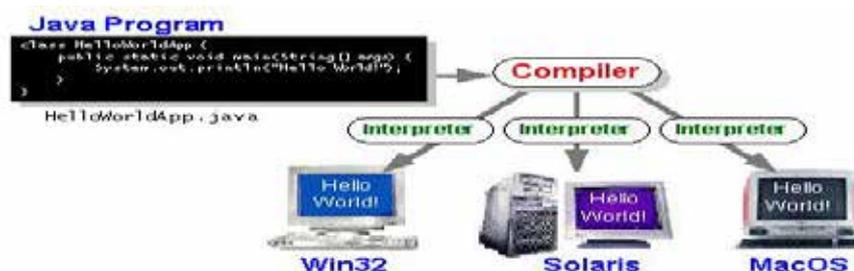
Πριν ξεκινήσει μια λεπτομερής αναφορά για την γλώσσα προγραμματισμού Java, προηγείται μια σύντομη ιστορική αναδρομή στη γλώσσα και στους λόγους δημιουργίας της.

Η λέξη Java πριν από δεκατέσσερα χρόνια θύμιζε ένα νησί στην Ινδονησία. Σήμερα, όποιος έχει δει μια ιστοσελίδα, περιοδικό υπολογιστών ή εφημερίδα ξέρει τι είναι η Java – είναι γλώσσα υπολογιστών που εισήχθη από την Sun Microsystems. Υπάρχουν ήθοιοι, αθλητές ή δολοφόνοι που έχουν ακουστεί πολύ λιγότερο από την Java.

Η γλώσσα Java αναπτύχθηκε από την Sun Microsystems το 1991, σαν μέρος του έργου Green, κατά το οποίο μια ερευνητική ομάδα εργαζόταν για να αναπτύξει λογισμικό για έλεγχο ηλεκτρονικών συσκευών ευρείας κατανάλωσης. Οι ερευνητές ήλπιζαν να αναπτύξουν μια γλώσσα προγραμματισμού που θα λειτουργούσε τις έξυπνες συσκευές στο μέλλον – διαλογικές τηλεοράσεις, διαλογικές τσντιέρες κλπ. Οι ερευνητές της Sun ήθελαν επίσης αυτές οι συσκευές να επικοινωνούν μεταξύ τους, έτσι ώστε η μια συσκευή να δώσει πληροφορίες σε μια άλλη.

Για να προχωρήσουν την έρευνά τους, οι ερευνητές της ομάδας Green ανέπτυξαν μια πρωτότυπη συσκευή με όνομα Star7, ένα μηχανισμό τύπου remote control που μπορούσε να επικοινωνεί με άλλες συσκευές ίδιου τύπου. Η αρχική ιδέα ήταν να αναπτύξουν το λειτουργικό σύστημα Star7 σε C++, την δημοφιλή γλώσσα προγραμματισμού, που αναπτύχθηκε από τον Bjarne Stroustrup, αλλά το μέλος της ομάδας James Gosling ενοχλήθηκε από τον τρόπο που η C++ έκανε την δουλειά, οπότε κλείστηκε στο γραφείο του και έγραψε μια νέα γλώσσα προγραμματισμού που μπορούσε να χειριστεί καλύτερα το Star7. Η γλώσσα ονομάστηκε Oak. Η Sun αργότερα ανακάλυψε ότι το όνομα Oak χρησιμοποιούταν ήδη και αναγκάστηκε να το αλλάξει σε Java.

Η Java είναι μια γλώσσα ανεξάρτητη πλατφόρμας. Δηλαδή ο ίδιος κώδικας μπορεί να δουλέψει σε Microsoft Windows, Unix, Applet Macintosh, Linux και σε κάθε μελλοντικό λειτουργικό σύστημα που υλοποιεί Java Virtual Machine αφού πρώτα μεταφραστεί σε bytecodes. Η ακόλουθη εικόνα δείχνει την ιδέα της εικονικής μηχανής Java.



Εικόνα 1.1 : Εικονική μηχανή της Java

1.2 ΓΙΑΤΙ ΛΟΙΠΟΝ JAVA

Η Java είναι μια πραγματικά αντικειμενοστραφής γλώσσα προγραμματισμού και συνιστάται ανεπιφύλακτα για την απεικόνιση του πραγματικού κόσμου καθώς και για την επίλυση ρεαλιστικών προβλημάτων. Έχει την δυνατότητα να προσαρμόζεται στις τελευταίες σχεδιαστικές μεθοδολογίες και παρέχει μεγάλη επεκτασιμότητα για την διαχείριση μεγάλων έργων.

Οι λόγοι που επιβάλλουν τη χρήση της είναι οι ακόλουθοι :

Ø **Η Java είναι σχεδιασμένη για το Internet.**

Προγράμματα της Java μπορούν να τρέχουν σε προγράμματα πλοήγησης εφοδιασμένα με την κατάλληλη JVM. Αφού το πρόγραμμα κατέβει από το Internet εκτελείται στην μηχανή του πελάτη και όχι στο Server. Αυτά τα προγράμματα της Java είναι γνωστά σαν Java Applets.

Ø **Η Java είναι σχεδιασμένη με στόχο την εξασφάλιση ασφάλειας.**

Ένα Java Applet εκτελείται με απόλυτη ασφάλεια στη μηχανή του χρήστη και δεν μπορεί να κάνει τίποτα που να μην επιτρέπεται από την πολιτική ασφαλείας της JVM, της μηχανής του πελάτη.

Ø **Η Java προσφέρει εκτέλεση σε όλους τους υπολογιστές.**

Με το επίπεδο αφαίρεσης του υλικού της JVM ένα πρόγραμμα Java εκτελείται σε όλους τους υπολογιστές ανεξαρτήτως Λειτουργικού Συστήματος. Αρκεί αυτές να είναι εφοδιασμένες με την κατάλληλη JVM (Java Virtual Machine).

Ø **Η Java είναι μια πλούσια γλώσσα προγραμματισμού.**

Η Java είναι πολύ ισχυρή αντικειμενοστραφής γλώσσα προγραμματισμού για το Internet (και όχι μόνο) και όχι μια απλή script γλώσσα για την σύνθεση συστατικών (π.χ. VBScript με ActiveX controls). Αυτό επιτρέπει να γράφονται προγράμματα τα οποία είναι εύκολο να συντηρηθούν και να εξελιχθούν στο χρόνο και όχι απλά σενάρια (scripts) τα οποία παίζουν το ρόλο της κόλλας για components που έχουν γίνει με άλλες γλώσσες αλλά γενικά δεν μπορούν να αλλάξουν ή να εξελιχθούν.

Ø **Η Java προσφέρει την ευελιξία της δυναμικής σύνδεσης.**

Ένα πρόγραμμα Java εκτελείται από μια εικονική μηχανή και οι τάξεις (τα αρχεία .class που περιέχουν τα bytewcodes) φορτώνονται από την JVM όταν εκτελείται το πρόγραμμα. Αυτό σημαίνει ότι μπορεί κανείς να αντικαταστήσει μια παλιά τάξη με μια νέα ακόμα και την στιγμή που εκτελείται το πρόγραμμα. Αυτό είναι ένα βασικό χαρακτηριστικό των components. Έτσι οι τάξεις Java πάνε πέρα από τον αντικειμενοστραφή προγραμματισμό, στον προγραμματισμό βασισμένο σε συστατικά (component – based programming). Είναι επίσης σημαντικό ότι η Java είναι σχεδιασμένη για αυτό και δεν απαιτείται η προσθήκη κάποιου πολύπλοκου μοντέλου (όπως το COM) της Microsoft, πάνω από μια γλώσσα προγραμματισμού που δεν είναι σχεδιασμένη με αυτόν τον τρόπο (π.χ. Visual C++). Αρκεί κανείς να μελετήσει το specification των Java Beans για να πειστεί για αυτό!!!

Στο παρελθόν της, η Java επέδειξε θεμελιώδη άλματα και πραγματοποίησε το απραγματοποίητο. Με την εξέλιξη και ωρίμανσή της, η Java είναι περισσότερο από προφανές ότι θα γίνει η επικρατέστερη γλώσσα ανάπτυξης εφαρμογών του μέλλοντος.

Μαθαίνοντας λοιπόν Java, δεν βοηθάει απίστευτα μόνο στην ανάπτυξη applets και εφαρμογών, αλλά προετοιμάζει για το μέλλον του προγραμματισμού υπολογιστών και συσκευών.

1.3 ΠΙΘΑΝΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ JAVA

∅ **Τα προγράμματα πλοήγησης ενδέχεται να μην μπορούν να εκτελέσουν τα Applet τους.**

Αυτό διότι τα προγράμματα πλοήγησης ενδέχεται να μην είναι εφοδιασμένα με την τρέχουσα έκδοση της Java.

∅ **Ο κώδικας που είναι μεταγλωττισμένος για μια συγκεκριμένη μηχανή είναι ταχύτερος από τον κώδικα που εκτελείται από έναν μεταφραστή (interpreter).**

Αξίζει να σημειωθεί ότι μια τεχνική που ονομάζεται Just – In Time μεταγλώττιση (JIT compilation) έχει βελτιώσει σημαντικά τους χρόνους εκτέλεσης των προγραμμάτων Java. Αν και ακόμα τα προγράμματα Java δεν έχουν γίνει ταχύτερα από τον μεταγλωττισμένο κώδικα της C++, πολλοί είναι αυτοί που υποστηρίζουν ότι είναι αρκετά γρήγορα για να μην γίνει η ταχύτητα η αιτία για να μην πετύχει η Java.

1.4 ΕΦΑΡΜΟΓΕΣ JAVA

Η Java μπορεί να προκάλεσε ίσως το μεγαλύτερο ενδιαφέρον σε σύγκριση με οποιαδήποτε άλλη εξέλιξη στον κόσμο του Internet αλλά δεν σταματά εκεί. Η Java είναι μια γλώσσα προγραμματισμού για ποικίλες εφαρμογές.

Ίσως είστε πιο εξοικειωμένοι με την Java σε μορφή Java Applet, αλλά η Java μπορεί να δημιουργήσει τοπικές εφαρμογές που να εκτελούνται μέσω της χρήσης ανεξάρτητων JVM.

Τι είναι μια εφαρμογή στην Java

Μια Java εφαρμογή είναι ένα πρόγραμμα σχεδιασμένο για να εκτελεστεί από μια JVM στη μηχανή του χρήστη και όχι από ένα πρόγραμμα πλοήγησης. Η εφαρμογή θα πρέπει να έχει και μια συνάρτηση main (κύρια) που είναι και το σημείο εισόδου στην εφαρμογή (δηλαδή από εκεί ξεκινά το πρόγραμμα). Φυσικά μια συνάρτηση main θα πρέπει να βρίσκεται σε μια τάξη (class). Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού και επομένως δεν μπορεί να έχει καμιά συνάρτηση που δεν είναι τμήμα μιας τάξης στη Java. Θα πρέπει να έχει μια έστω τάξη στην οποία θα περιέχεται η main ακόμα και αν δεν χρειάζεται!

Οι εφαρμογές δεν δεσμεύονται από περιορισμούς όπως τα Java Applets, αντιθέτως μπορούν να αποκτήσουν πλήρη έλεγχο του υπολογιστή του χρήστη. Οι εφαρμογές Java εκτελούνται όπως και οι υπόλοιπες εφαρμογές σε ένα υπολογιστή. Μπορούν να διαβάζουν και να γράφουν σε αρχεία, να εκκινούν άλλες εφαρμογές και γενικώς να κάνουν ότι έχουν προνόμια να κάνουν. Ο μόνος περιορισμός που αντιμετωπίζουν οι εφαρμογές Java είναι ότι δεν μπορούν να ενσωματωθούν σε μια σελίδα στο Internet και να εκτελεστούν σε ένα πλοηγό Internet.

Βήματα δημιουργίας μιας εφαρμογής σε Java

Τα βήματα για να δημιουργηθεί μια εφαρμογή σε Java είναι τα ακόλουθα :

1. Δημιουργία ενός αρχείου κειμένου με τον πηγαίο κώδικα.
2. Αποθήκευση του αρχείου με όνομα ίδιο με την δημόσια τάξη της Java που δημιουργήθηκε στο αρχείο τοποθετώντας ως επίθεμα *.java*.
3. Μεταγλώττιση του αρχείου χρησιμοποιώντας *javac* ώστε να δημιουργηθούν τα αρχεία *.class*.
4. Εκτελέστε *java* με το όνομα της *main* κλάσης σαν παράμετρο στην εντολή γραμμής, για να εκτελεστεί η εφαρμογή.

Η πρώτη εφαρμογή : Το πρόγραμμα “Hello World!”

Σε αυτό το σημείο παρουσιάζεται η πρώτη εφαρμογή στη Java και ο κώδικάς της είναι ο ακόλουθος :

```
public class HelloWorld {  
  
    public static void main (String [] args) {  
  
        System.out.println (“Hello World!”);  
    }  
}
```

Θα πρέπει να αποθηκευτεί το αρχείο με το όνομα **HelloWorld.java** να μεταγλωττιστεί σε *bytecodes* δίνοντας `javac HelloWorld.java` και αν δεν έχει λάθη να εκτελεστεί πληκτρολογώντας `java HelloWorld`.

2.1 ΕΙΣΑΓΩΓΗ

Η Java είναι μια αντικειμενοστραφής (object – oriented) γλώσσα προγραμματισμού. Στο κεφάλαιο αυτό παρουσιάζονται ορισμένες έννοιες που διέπουν τον αντικειμενοστραφή προγραμματισμό.

2.2 ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο Αντικειμενοστραφής Προγραμματισμός, ή αλλιώς OOP (Object Oriented Programming, OOP) είναι ένας τρόπος υλοποίησης προγραμμάτων, ο οποίος έχει ως κεντρική ιδέα, ότι τα προγράμματα θα πρέπει να οργανώνονται κατά τέτοιο τρόπο ώστε να αντιπροσωπεύουν τον τρόπο οργάνωσης των αντικειμένων στον πραγματικό κόσμο.

2.3 ΤΑΞΗ (CLASS) & ΑΝΤΙΚΕΙΜΕΝΟ (OBJECT)

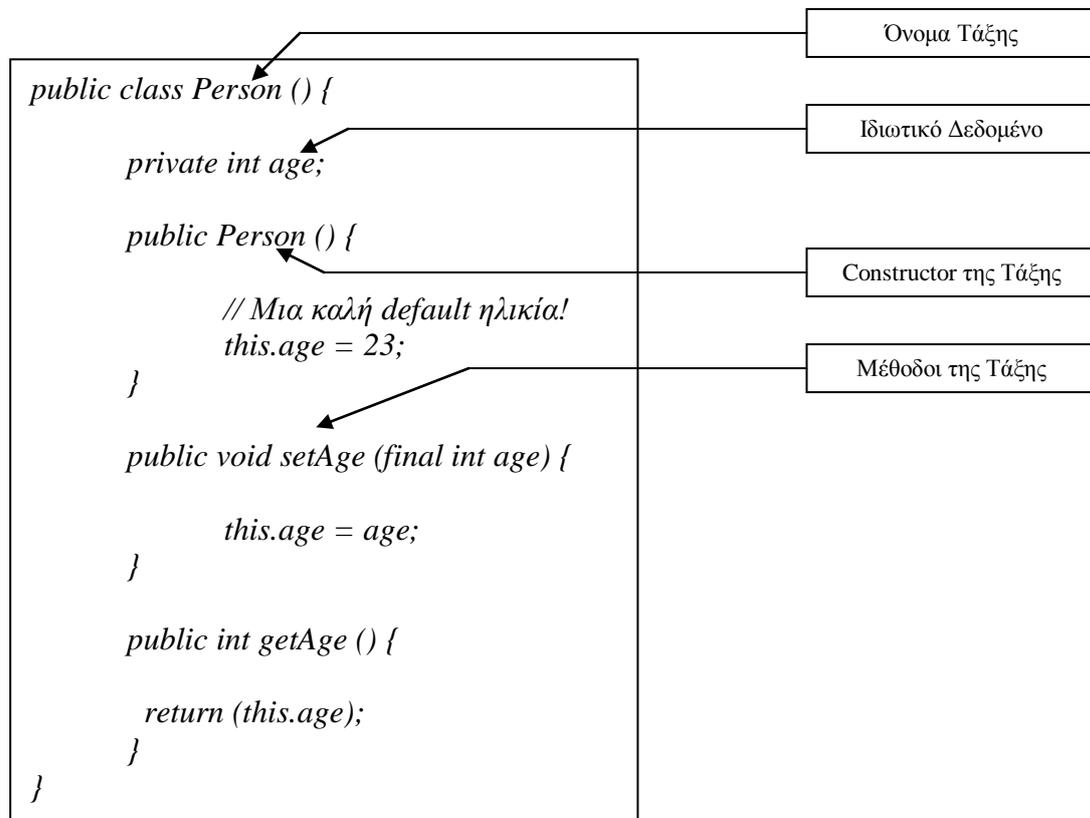
Το αντικείμενο είναι ένα αυτόνομο στοιχείο ενός προγράμματος που αναπαριστά μια σχετική ομάδα χαρακτηριστικών και είναι έτσι σχεδιασμένο ώστε να κάνει κάποιες συγκεκριμένες ενέργειες. Κάθε αντικείμενο έχει ένα συγκεκριμένο ρόλο μέσα σε ένα πρόγραμμα. Γενικότερα, ένα αντικείμενο δεν είναι παρά μια περίπτωση (instance) μιας τάξης.

Συγκεκριμένα, μια τάξη (class) είναι ένα πρότυπο που χρησιμοποιείται για την δημιουργία αντικειμένων με κοινά χαρακτηριστικά.

Μια τάξη είναι μια μονάδα :

- 1. Αφαίρεσης :** Παριστάνει μια οντότητα του χώρου λύσης του προβλήματος.
- 2. Συγκέντρωσης :** Συγκεντρώνει δεδομένα και μεθόδους κάτω από τον ίδιο τύπο.
- 3. Απόκρυψης :** Επιτρέπει τον απόλυτο έλεγχο στο ποιες άλλες τάξεις μπορούν να προσπελάσουν τα δεδομένα της.
- 4. Επέκτασης :** Μπορεί να αποτελέσει την βάση ή την επέκταση ή και τα δυο κάποιας άλλης τάξης.
- 5. Εξέλιξη :** Δεδομένου κάποιων προϋποθέσεων μπορεί να διευκολύνει την εξέλιξη του λογισμικού με συστηματικό τρόπο.

Τα τμήματα της δήλωσης μιας τάξης στην Java



2.4 ΥΠΕΡΦΟΡΤΩΣΗ (OVERLOADING) – ΠΟΛΥΜΟΡΦΙΣΜΟΣ (POLYMORPHISM)

Η υπερφόρτωση συναρτήσεων ονομάζεται και πολυμορφισμός συναρτήσεων. Ο πολυμορφισμός συναρτήσεων, αναφέρεται στην ικανότητα “υπερφόρτωσης” μιας συνάρτησης με περισσότερες της μίας “σημασίες”. Δηλαδή, αλλάζοντας τον αριθμό ή τον τύπο των παραμέτρων, μπορείτε να δώσετε σε δυο ή περισσότερες συναρτήσεις το ίδιο όνομα και θα καλείται η σωστή με αντιστοιχία των χρησιμοποιούμενων παραμέτρων.

2.5 ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ (INHERITANCE)

Η κληρονομικότητα είναι ένας μηχανισμός που επιτρέπει σε μια κλάση να κληρονομεί όλη την συμπεριφορά και τις ιδιότητες μιας άλλης κλάσης. Η κληρονομικότητα είναι μια από τις βασικότερες αρχές στον αντικειμενοστραφή προγραμματισμό και έχει άμεση επίδραση στο πώς σχεδιάζονται και γράφονται οι κλάσεις στη Java.

2.6 ΔΙΑΣΥΝΔΕΣΗ (INTERFACE)

Διασύνδεση (interface) είναι μια δομή παρόμοια με μια τάξη η οποία όμως περιέχει μόνο δηλώσεις μεθόδων και σταθερών. Μπορούν να χρησιμοποιηθούν σε δηλώσεις διαφόρων τάξεων για να εξασφαλιστεί ότι οι τάξεις αυτές θα παρέχουν τουλάχιστον την συμπεριφορά που υποδηλώνεται από τις διασυνδέσεις.

ΚΕΦΑΛΑΙΟ 3 : ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ, SQL και JDBC

3.1 ΕΙΣΑΓΩΓΗ

Οι εφαρμογές Java, παρέχουν δυνατότητες πρόσβασης σε πληροφορίες που βρίσκονται αποθηκευμένες σε Βάσεις Δεδομένων (ΒΔ - DataBases). Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (DBMS – DataBase Management Systems) που χρησιμοποιούνται στην μεγάλη πλειοψηφία των εφαρμογών είναι οι λεγόμενες Σχεσιακές Βάσεις Δεδομένων (Relational DataBases).

3.2 ΛΙΓΑ ΛΟΓΙΑ ΓΙΑ ΤΙΣ ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Στις Σχεσιακές (Relational) βάσεις δεδομένων, τα δεδομένα συνδέονται μεταξύ τους με σχέσεις (relations), οι οποίες προκύπτουν από τα κοινά πεδία που υπάρχουν σε διαφορετικά αρχεία. Τα αρχεία αποκαλούνται πίνακες (tables), οι εγγραφές γραμμές (rows) και τα πεδία στήλες (columns). Η ύπαρξη μιας κοινής τιμής στα πεδία δυο αρχείων καθορίζει και μια σχέση μεταξύ των γραμμών διαφορετικών πινάκων. Οι σχεσιακές βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι λογικά κατανοητές, πολύ ευέλικτες και δεκτικές σε αλλαγές.

Για παράδειγμα θα μπορούσαμε να είχαμε έναν πίνακα σπουδαστών Students με τα εξής στοιχεία (sid, sfname, slname) για τον κωδικό, το όνομα και το επώνυμο του σπουδαστή και έναν άλλο πίνακα για τα μαθήματα Courses με τα εξής στοιχεία (cid, ctitle, cdescription) για τον κωδικό, τον τίτλο και την περιγραφή του μαθήματος. Αν θέλαμε να καταγράψουμε την εγγραφή ενός σπουδαστή σε ένα ή περισσότερα μαθήματα θα μπορούσαμε να δημιουργήσουμε έναν πίνακα Registration με τα ακόλουθα στοιχεία (sid, cid, mark) που θα περιείχε γραμμές με τους κωδικούς του σπουδαστή και του μαθήματος στο οποίο έχει εγγραφεί και με τον βαθμό που πήρε ο σπουδαστής σε αυτό το μάθημα.

Οι πίνακες που ακολουθούν περιέχουν κάποια στοιχεία για πιθανούς σπουδαστές, μαθήματα και εγγραφές.

Πίνακας Students

sid	sfname	slname
1	Αναστάσιος	Μανάκος
2	Θεοδώρα	Αναστασίου

Πίνακας Courses

cid	ctitle	cdescription
1	Java	Internet Programming
2	Php	Script Language

Πίνακας Registration

sid	cid	mark
1	1	10
1	2	7
2	1	5

Από τα πεδία του πίνακα Registration προκύπτει πως ο σπουδαστής με κωδικό 1 “Μανάκος” πήρε στο μάθημα με κωδικό 1 “Java” τον βαθμό 10 και στο μάθημα με κωδικό 2 “Php” τον βαθμό 7. Επίσης, ο σπουδαστής με κωδικό 2 “Αναστασίου” πήρε στο μάθημα με κωδικό 1 “Java” τον βαθμό 5.

Το πεδίο sid για τον πίνακα Students ονομάζεται πρωτεύον κλειδί (primary key) γιατί δεν μπορεί να έχει την ίδια τιμή για δυο διαφορετικές εγγραφές (για δυο διαφορετικές γραμμές του πίνακα Students). Το πρωτεύον κλειδί για τον πίνακα Courses είναι το cid. Τέλος το πρωτεύον κλειδί για τον πίνακα Registration είναι το σύνθετο κλειδί (sid, cid). Τα sid και cid στον πίνακα Registration ονομάζονται ξένα κλειδιά (foreign keys) για τον πίνακα Registration διότι είναι πρωτεύοντα κλειδιά των δυο άλλων πινάκων.

Οι πίνακες Students και Courses συσχετίζονται μεταξύ τους με τις εγγραφές του πίνακα Registration. Είναι φανερό πως με αυτή την υλοποίηση δεν υπάρχει επανάληψη τιμών πέρα από τα ξένα κλειδιά τα οποία επαναλαμβάνονται στον πίνακα Registration. Δηλαδή για παράδειγμα δεν υπάρχει πουθενά δυο φορές το όνομα του σπουδαστή ή ο τίτλος του μαθήματος. Οι πληροφορίες του πίνακα Registration είναι αρκετές για να μας δώσουν τις πληροφορίες της συσχέτισης των δυο άλλων πινάκων.

Με αυτή την υλοποίηση δεν υπάρχει ανάγκη αν αλλάξει για παράδειγμα ο τίτλος ή η περιγραφή ενός μαθήματος να κάνουμε την αλλαγή σε πολλά σημεία. Αρκεί η αλλαγή σε ένα μόνο σημείο. Αυτό που θα χρειαζόταν να αλλάξει σε πολλά σημεία, αν άλλαζε, θα ήταν για παράδειγμα ο κωδικός του σπουδαστή ή του μαθήματος. Γι’ αυτόν ακριβώς τον λόγο είναι λάθος να αποδίδουμε τον ρόλο πρωτεύοντος κλειδιού σε πεδία που περιέχουν πληροφορίες που δεν είναι στον απόλυτο έλεγχό μας. Ο μοναδικός τους ρόλος θα πρέπει να είναι η διάκριση των εγγραφών και τίποτε άλλο. Όπως είναι αυτονόητο τα λίγα αυτά λόγια δεν είναι παρά μόνο μια μικρή εισαγωγή στις σχεσιακές ΒΔ.

3.3 ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΤΗΝ SQL

Για την διαχείριση των σχεσιακών ΒΔ υπάρχει μια γλώσσα πρότυπο, που υποστηρίζεται από όλα τα μεγάλα ΣΔΒΔ και ονομάζεται SQL (Structured Query Language). Αρχικά η SQL ονομαζόταν SEQUEL (από το Structured English QUery Language, Γλώσσα Ερωτήσεων Δομημένων Αγγλικών) και σχεδιάστηκε και υλοποιήθηκε από την IBM. Η γλώσσα SQL μπορεί να θεωρηθεί ως ένας από τους σημαντικότερους λόγους για την επιτυχία των Σχεσιακών Βάσεων Δεδομένων.

Η SQL είναι μια πλήρης γλώσσα βάσεων δεδομένων. Διαθέτει πλήθος εντολών για τον ορισμό δεδομένων, για την δημιουργία ερωτήσεων, για ενημερώσεις, για

διαγραφές δεδομένων κλπ. Επομένως μπορεί να θεωρηθεί ότι είναι ταυτόχρονα Γλώσσα Ορισμού Δεδομένων αλλά και Γλώσσα Χειρισμού Δεδομένων. Επιπλέον παρέχει στον χρήστη την δυνατότητα να προσδιορίζει μόνο ποιο θα είναι το αποτέλεσμα αφήνοντας τις αποφάσεις για το πώς θα εκτελεστεί η ερώτηση στο ΣΔΒΔ. Τέλος περιέχει κανόνες για την εμφύτευση των εντολών της σε ορισμένες γλώσσες προγραμματισμού, όπως η C, η Java κτλ.

Εάν θέλουμε να δούμε την εγγραφή του πίνακα Students για τον σπουδαστή με επώνυμο “Μανάκος”, θα δίνουμε την ακόλουθη εντολή SELECT της SQL :

```
SELECT *  
FROM Students  
WHERE slname = ‘Μανάκος’;
```

που σημαίνει τα εξής : Επέλεξε όλα τα πεδία (*) από τον πίνακα Students (FROM Students) για τα οποία το επώνυμο είναι ‘Μανάκος’ (WHERE slname = ‘Μανάκος’). Το ερώτημα (query) αυτό θα εκτελεστεί από το ΣΔΒΔ το οποίο θα καταστρώσει το καταλληλότερο σχέδιο για την εκτέλεσή του (Query Execution Plan). Αν για παράδειγμα υπάρχει κάποιο ευρετήριο με βάση το οποίο μπορεί να γίνει η αναζήτηση (ευρετήριο με βάση το επώνυμο στην συγκεκριμένη περίπτωση) τότε μπορεί να χρησιμοποιηθεί για να γίνει η αναζήτηση γρηγορότερα. Αν όχι τότε η αναζήτηση θα γίνει ακολουθιακά. Το σημαντικό όμως είναι πως η εντολή SELECT είναι ίδια και στις δυο περιπτώσεις. Γι’ αυτό λέμε πως τα δεδομένα είναι ανεξάρτητα από τα προγράμματα που τα επεξεργάζονται (data independence principle).

3.4 ΤΟ ΣΥΣΤΗΜΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ MySQL

Η MySQL είναι σήμερα ίσως το πιο διαδεδομένο ΣΔΒΔ, που διατίθεται για ελεύθερη χρήση. Οι ενδιαφερόμενοι μπορούν να “κατεβάσουν” την MySQL από το site : www.mysql.com, όπου υπάρχουν εκδόσεις για διάφορες πλατφόρμες. Μαζί με την MySQL διατίθεται και ένα αρκετά αναλυτικό εγχειρίδιο χρήσης, όπως και στις περισσότερες εφαρμογές που βοηθάει στην εγκατάσταση και στη δημιουργία οποιασδήποτε ΒΔ.

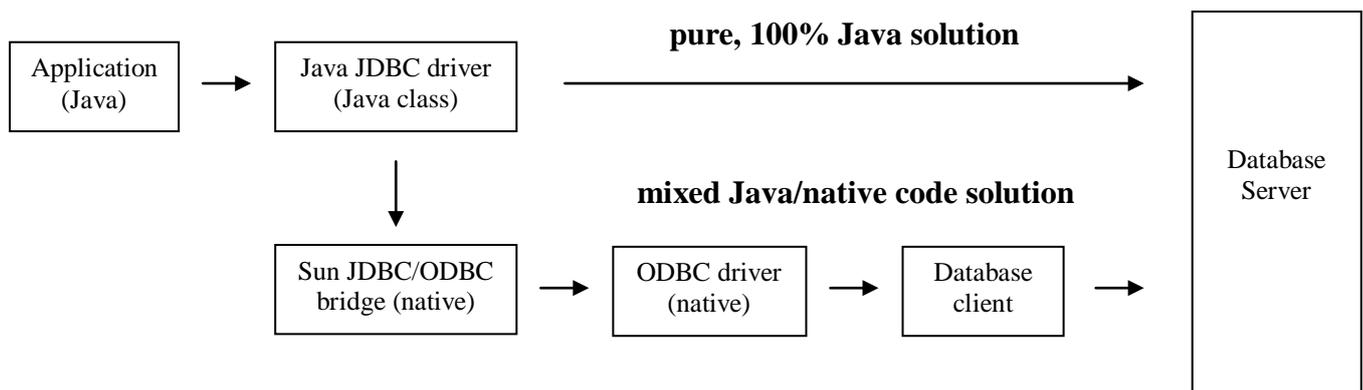
Η MySQL μπορεί να χρησιμοποιηθεί για την αποθήκευση μεγάλου όγκου πληροφοριών και είναι κατάλληλη για χρήση από επιχειρήσεις και οργανισμούς. Δεν είναι δηλαδή απλά ένα “demo” ΣΔΒΔ για εκπαιδευτικούς σκοπούς.

3.5 JDBC DRIVER

Το JDBC (Java DataBase Connectivity) είναι το πλαίσιο που χρησιμοποιείται για τη δημιουργία συνδέσεων με τις βάσεις δεδομένων και περιέχει κλάσεις για τη δημιουργία πινάκων, SQL ερωτήσεων κ.α. Για τη δημιουργία μιας σύνδεσης με μια βάση δεδομένων, εκτός από το JDBC API χρειάζεται επίσης και ένας οδηγός γραμμένος σύμφωνα με το JDBC API τον οποίο συνήθως προσφέρει ο κατασκευαστής της βάσης.

Υπάρχουν τέσσερις τύποι JDBC Drivers οι οποίοι είναι οι εξής :

- ∅ Ο τύπος 1 είναι μια γέφυρα (bridge) για την επικοινωνία του JDBC με το ODBC.
- ∅ Ο τύπος 2 είναι ένας driver γραμμένος σε μια άλλη γλώσσα προγραμματισμού συγκεκριμένη για κάθε ΣΔΒΔ που παρέχει ένα εξωτερικό στρώμα Java για τον προγραμματισμό εφαρμογών Java.
- ∅ Ο τύπος 3 είναι ένας driver γραμμένος σε Java που επικοινωνεί χρησιμοποιώντας ένα γενικό πρωτόκολλο δικτύου, με ένα ενδιάμεσο συστατικό (middleware component) το οποίο επικοινωνεί με το ΣΔΒΔ.
- ∅ Τέλος, ο τύπος 4, είναι ένας driver γραμμένος εξ' ολοκλήρου σε Java που χρησιμοποιεί απευθείας το πρωτόκολλο επικοινωνίας ενός συγκεκριμένου ΣΔΒΔ και έτσι επικοινωνεί απευθείας με το ΣΔΒΔ.



Εικόνα 3.1 : Αρχιτεκτονική Προσπέλασης ΒΔ με Java

ΚΕΦΑΛΑΙΟ 4 : ΠΡΟΣΒΑΣΗ ΣΕ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΜΕ JAVA

4.1 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΠΑΡΑΔΕΙΓΜΑΤΟΣ

Για να κάνουμε κάποια παραδείγματα θα πρέπει να δημιουργήσουμε μια Βάση Δεδομένων. Θα κατασκευάσουμε την Βάση Δεδομένων που συζητήσαμε στο προηγούμενο κεφάλαιο με τους σπουδαστές και τα μαθήματα.

Από τον φλοιό του Λειτουργικού Συστήματος και από τον κατάλογο <mysqlhome>/bin συνδεθείτε ως root ως εξής :

```
C:\mysql\bin>mysql -u root -p

Enter password : *****

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 1 to server version: 4.1.9-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Στο πιο πάνω παράδειγμα φαίνεται η σύνδεση με τον mysql monitor σε ένα σύστημα Windows NT. Φυσικά για να πετύχει η πιο πάνω εντολή θα πρέπει να τρέχει ο MySQL Server.

Στην συνέχεια θα δημιουργήσουμε την Βάση Δεδομένων jAES και θα συνδεθούμε με την Βάση Δεδομένων ως εξής :

```
mysql> CREATE DATABASE jAES;
Query OK, 1 row affected (0.00 sec)

mysql> USE jAES
Connection id: 8
Current database: jAES

mysql>
```

Έπειτα θα δημιουργήσουμε τους τρεις πίνακες της Βάσης Δεδομένων, τον πίνακα Students για τους σπουδαστές, τον πίνακα Courses για τα μαθήματα και τον πίνακα Registration για την εγγραφή των μαθητών σε μαθήματα. Γι' αυτό θα χρησιμοποιήσουμε τρεις φορές την εντολή CREATE TABLE της SQL :

```
mysql> CREATE TABLE Students (  
-> sid integer NOT NULL default '0',  
-> sfname varchar (20) NOT NULL default "",  
-> slname varchar (20) NOT NULL default "",  
-> PRIMARY KEY (sid)  
-> ) TYPE = InnoDB;  
Query OK, 0 rows affected (0.05 sec)  
  
mysql>
```

Όπως είναι φανερό η CREATE TABLE δημιουργεί τον πίνακα Students με τα πεδία sid που είναι ακέραιο (integer) και είναι το πρωτεύον κλειδί του πίνακα (PRIMARY KEY), sfname, slname που είναι πεδία χαρακτήρων μεταβλητού μήκους με ανώτατο όριο τους 20 χαρακτήρες (varchar (20)) και αρχική τιμή το κενό String. Επίσης καθορίζεται πως ο τύπος του πίνακα θα είναι InnoDB (TYPE = InnoDB).

Παρομοίως δίνονται και οι εντολές για την δημιουργία των πινάκων Courses και Registration ως εξής :

```
mysql> CREATE TABLE Courses (  
-> cid integer NOT NULL default '0',  
-> cttitle varchar (100) NOT NULL default "",  
-> cdescription text NOT NULL default "",  
-> PRIMARY KEY (cid),  
-> UNIQUE KEY cttitle (cttitle)  
-> ) TYPE = InnoDB;  
Query OK, 0 rows affected (0.10 sec)  
  
mysql> CREATE TABLE Registration (  
-> sid integer NOT NULL default '0' REFERENCES Students (sid),  
-> cid integer NOT NULL default '0' REFERENCES Courses (cid),  
-> mark real default '0.0',  
-> PRIMARY KEY (sid,cid)  
-> ) TYPE = InnoDB;  
Query OK, 0 rows affected (0.05 sec)  
  
mysql>
```

Τα νέα στοιχεία στις δηλώσεις των δυο αυτών πινάκων είναι τα εξής :

- Ø Το πεδίο cdescription του πίνακα Courses είναι τύπου text και όχι varchar. Αυτό διότι ο τύπος varchar έχει ανώτατο μέγεθος μέχρι 256 χαρακτήρες ενώ ο τύπος text έχει ανώτατο μέγεθος μέχρι 65538 χαρακτήρες. Για την περιγραφή του μαθήματος πιθανώς να απαιτηθούν περισσότεροι από 256 χαρακτήρες και έτσι δηλώνεται το πεδίο ως τύπου text.
- Ø Τα πεδία sid και cid του πίνακα Registration δηλώνουν ότι αναφέρονται στα αντίστοιχα πεδία των πινάκων Students και Courses με τα ίδια ονόματα.

Τέλος με την φράση PRIMARY KEY (sid, cid), δηλώνεται το σύνθετο κλειδί για τον πίνακα Registration που αποτελείται από τα δυο πεδία που είναι τα ξένα κλειδιά του πίνακα.

4.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ

Για να προσπελαστεί μια Βάση Δεδομένων μέσω JDBC από οποιοδήποτε πρόγραμμα Java θα πρέπει πρώτα απ' όλα να δημιουργηθεί μια σύνδεση με την Βάση Δεδομένων. Η δημιουργία μιας σύνδεσης περιλαμβάνει δυο βήματα :

1. Το πρώτο είναι η φόρτωση της τάξης του JDBC Driver στην JVM της εφαρμογής. Αυτό γίνεται με την εντολή : `Class.forName` (“όνομα – τάξης – jdbc – driver”).
Η εφαρμογή αν δεν μπορέσει να εντοπίσει την τάξη του Driver, η μέθοδος `Class.forName` () θα προκαλέσει την εξαίρεση `java.lang.ClassNotFoundException` η οποία θα πρέπει να χειριστεί με ένα `try – catch` block κώδικα.
2. Αφού φορτωθεί ο Driver και καταχωρηθεί στην τάξη `java.sql.DriverManager` ως ένας διαθέσιμος JDBC Driver για την εφαρμογή, στην συνέχεια θα πρέπει να δημιουργηθεί μια σύνδεση με την Βάση Δεδομένων καλώντας την στατική μέθοδο `getConnection` στην τάξη `java.sql.DriverManager`. Η μέθοδος αυτή, στην πιο απλή μορφή της, παίρνει ως παράμετρο ένα URL της Βάσης Δεδομένων, που είναι εξαρτώμενο από το συντακτικό που απαιτεί ο driver και μπορεί να διαφέρει από driver σε driver. Το URL για τον mysql driver ξεκινά με `jdbc:mysql://`. Ακολουθεί το όνομα του server στον οποίο τρέχει ο εξυπηρετητής της MySQL. Τέλος ακολουθεί μια σειρά παραμέτρων. Οι δυο απαραίτητες είναι το όνομα του χρήστη και ο κωδικός. Τέλος η μέθοδος `getConnection` () μπορεί να προκαλέσει την `SQLException` που θα πρέπει να χειριστούμε.

Για παράδειγμα για να αποκτήσουμε μια σύνδεση με την Βάση Δεδομένων jAES που δημιουργήθηκε στην προηγούμενη ενότητα στον MySQL Server που τρέχει στον τοπικό υπολογιστή, θα πρέπει να φτιαχτεί μια κλάση “DatabaseManager” η οποία θα κάνει αυτή την δουλειά.

Στην συνέχεια θα δημιουργηθεί η κλάση DatabaseManager και θα γίνει η σύνδεση με την Βάση Δεδομένων jAES ως εξής :

```
import java.sql.*;

public class DatabaseManager {

    // Δημιουργία μεταβλητών μέλους (member variable)
    private Connection con = null;
    private final String url = "jdbc:mysql://localhost/jAES";
    private final String username = "root";
    private final String password = "tasos_home";
```

```
// Συνάρτηση Δημιουργίας – Constructor
public DatabaseManager () {
}

/*
 * Σύνδεση με την Βάση Δεδομένων.
 * Επιστρέφει true à σε Επιτυχία
 * false à σε Αποτυχία
 */
public Boolean connect () {

    try {
        // Φόρτωση της τάξης του JDBC Driver στην JVM
        Class.forName (“com.mysql.jdbc.Driver”);

        // Δημιουργία σύνδεσης με την Βάση δεδομένων
        this.con = DriverManager.getConnection (this.url,
            this.username, this.password );

        if (!this.con.isClosed ()) {
            System.out.println (“Successfully connected to
                MySQL Server ...”);
        }
    }

    catch (ClassNotFoundException cnfex) {

        // Λάθος στην Class.forName ()
        System.err.println (“Failed to Load JDBC Driver...”);

        cnfex.printStackTrace ();

        return (new Boolean (false));
    }

    catch (SQLException sqllex) {

        // Λάθος στην DriverManager.getConnection ()
        System.err.println (“Unable to connect...”);

        sqllex.printStackTrace ();

        return (new Boolean (false));
    }
}
```

```
/*
 * Αποσύνδεση με την Βάση Δεδομένων.
 * Επιστρέφει true à σε Επιτυχία
 *      false à σε Αποτυχία
 */
public Boolean shutDown () {

    try {

        if (this.con != null) {

            // Αποσύνδεση με την Βάση Δεδομένων
            this.con.close ();

            System.out.println ("Successfully disconnected to
                                MySQL Server...");

        }

    }

    catch (SQLException sqllex) {

        // Λάθος στην con.close ()
        System.err.println ("Unable to disconnect...");

        sqllex.printStackTrace ();

        return (new Boolean (false));

    }

    return (new Boolean (true));

}
```

Θα πρέπει να αποθηκευτεί το αρχείο με το όνομα DatabaseManager.java και να μεταγλωττιστεί σε bytecodes δίνοντας `javac DatabaseManager.java`.

ΚΕΦΑΛΑΙΟ 5 : ΠΕΡΙΓΡΑΦΗ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ (jA.E.S.)

5.1 Η ΙΔΕΑ

Κάθε έργο διαθέτει έναν οραματιστή, ένα πρόσωπο με μια μεγάλη ιδέα. Αυτός είναι ο υπεύθυνος να επιβεβαιώνει ότι η ιδέα ακολουθείται σε όλη την διάρκεια ζωής του έργου. Συχνά κατά τη διάρκεια ανάπτυξης λογισμικού ξεχνιέται ή διογκώνεται η αρχική ιδέα και το τελικό αποτέλεσμα σίγουρα δεν συμφωνεί με την αρχική πρόθεση. Γράφοντας με συνοπτικό τρόπο αυτό που προσπαθεί κάποιος να πραγματοποιήσει και κάνοντας συχνές αναφορές σε αυτό κατά την διάρκεια του έργου, μπορεί να είναι σίγουρος ότι διατηρείται η ακεραιότητα της ιδέας.

Έχει ήδη αναφερθεί ότι θα σχεδιαστεί και θα αναλυθεί μια εφαρμογή ενός “Αυτόματου Συστήματος Εξέτασης”. Ποια είναι όμως τα χαρακτηριστικά που θα την κάνουν ολοκληρωμένη; Παρακάτω βρίσκονται τα κυριότερα χαρακτηριστικά που έρχονται στο μυαλό κάποιου όταν σκέφτεται μια εφαρμογή Αυτόματου Συστήματος Εξέτασης :

- Ø Να καταγράφει ηλεκτρονικές εξετάσεις.
- Ø Να ενημερώνει άμεσα τους χρήστες για το ποσοστό επιτυχίας τους.
- Ø Να διατηρεί τις τρέχουσες πληροφορίες της κάθε εξέτασης για μελλοντική χρήση.
- Ø Να διαχειρίζεται πολλαπλά προφίλ (όπως για καθηγητές, μαθητές κλπ).
- Ø Να υποστηρίζει την ανταλλαγή δεδομένων (εισαγωγή και εξαγωγή) μέσα από αρχεία χωριζόμενων με κόμμα μεταβλητών (comma separated values – CSV).

ΣΗΜΕΙΩΣΗ : Ο όρος αρχείο CSV ή μεταβλητών χωριζόμενων με κόμμα (comma separated values) αναφέρεται σε μια διεθνή μορφή αρχείου κατανοητή στα προγράμματα φύλλων εργασίας. Αν μπορείτε να εισάγετε και να εξάγετε αρχεία CSV, το πρόγραμμά σας θα είναι σε θέση να ανταλλάξει δεδομένα σχεδόν με κάθε πρόγραμμα φύλλου εργασίας ή βάσης δεδομένων (καλό να έχει κανείς μια τέτοια δυνατότητα). Προσοχή: Αν και το όνομα “χωριζόμενα με κόμμα” δείχνει ότι ο διαχωριστικός χαρακτήρας είναι το κόμμα, δεν είναι πάντα έτσι. Αυτό εξαρτάται από την πλατφόρμα.

Ο παραπάνω κατάλογος δεν είναι πλήρης σε καμία περίπτωση, αλλά είναι μια καλή αρχή!

5.2 ΕΙΣΑΓΩΓΗ ΣΤΟ jA.E.S.

Το jA.E.S. είναι ένα ολοκληρωμένο αυτόματο σύστημα παροχής ηλεκτρονικών εξετάσεων. Αυτό σημαίνει ότι μπορεί κάποιος να χρησιμοποιήσει το jA.E.S. ως ένα ολοκληρωμένο εργαλείο παροχής ηλεκτρονικών εξετάσεων.

Στην ουσία δεν αποτελεί υλοποίηση του παραδοσιακού γραπτού τρόπου εξέτασης. Αντιθέτως είναι ένα εργαλείο το οποίο χρησιμοποιεί κάποια βασικά χαρακτηριστικά αυτού και εκμεταλλεύοντας όρους, λειτουργίες και δομές της συμβατικής εξέτασης, επιτυγχάνεται η δημιουργία του.

Το jA.E.S. αναπτύχθηκε σε Java, την δημοφιλή γλώσσα προγραμματισμού που αναπτύχθηκε από τον James Gosling. Επιλέχτηκε η Java και όχι κάποια άλλη γλώσσα προγραμματισμού γιατί η Java είναι μια πραγματικά αντικειμενοστραφής γλώσσα προγραμματισμού. Συνιστάται ανεπιφύλακτα για την απεικόνιση του πραγματικού κόσμου καθώς και για την επίλυση ρεαλιστικών προβλημάτων.

Η λειτουργία του jA.E.S. βασίζεται στην ύπαρξη ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων. Το προϊόν που επιλέγει είναι το MySQL Server, στην εμπορική έκδοση 4.1.9.

5.3 ΕΞΕΤΑΣΗ ΜΕΣΩ ΤΟΥ jA.E.S.

Στη συνέχεια εξετάζεται ο τρόπος λειτουργίας του Αυτόματου Συστήματος Εξέτασης jA.E.S. και ο τρόπος με τον οποίο μπορεί να επιτευχθεί η εξέταση μέσω αυτού.

Ο τρόπος διεξαγωγής της εξέτασης, γίνεται υπό μορφή ερωτήσεων πολλαπλής επιλογής. Οι ερωτήσεις της κάθε εξέτασης επιλέγονται μέσω ενός μαθηματικού αλγορίθμου ο οποίος επιλέγει “τυχαία” ένα υποσύνολο ερωτήσεων από ένα προκαθορισμένο σύνολο ερωτήσεων. Οι ερωτήσεις συμπεριλαμβάνονται στην αντίστοιχη ύλη του εκάστοτε μαθήματος.

Το χρονικό διάστημα εξέτασης που έχει στην κατοχή του ο κάθε σπουδαστής, καθορίζεται από τον υπεύθυνο καθηγητή του συγκεκριμένου μαθήματος και είναι ανάλογο του είδους και της δυσκολίας των ερωτήσεων.

Στο τέλος της εξέτασης, ο σπουδαστής ενημερώνεται αναλυτικά για το ποσοστό επιτυχίας – αποτυχίας του και του δίνεται η δυνατότητα αποθήκευσης της εξέτασης σε αρχείο CSV.

Θα πρέπει να σημειωθεί, πως ο σπουδαστής κατά την διάρκεια της εξέτασης, έχει την δυνατότητα σύνδεσης με δυο από τις δημοφιλέστερες μηχανές αναζήτησης (Google, Yahoo) του Διαδικτύου.

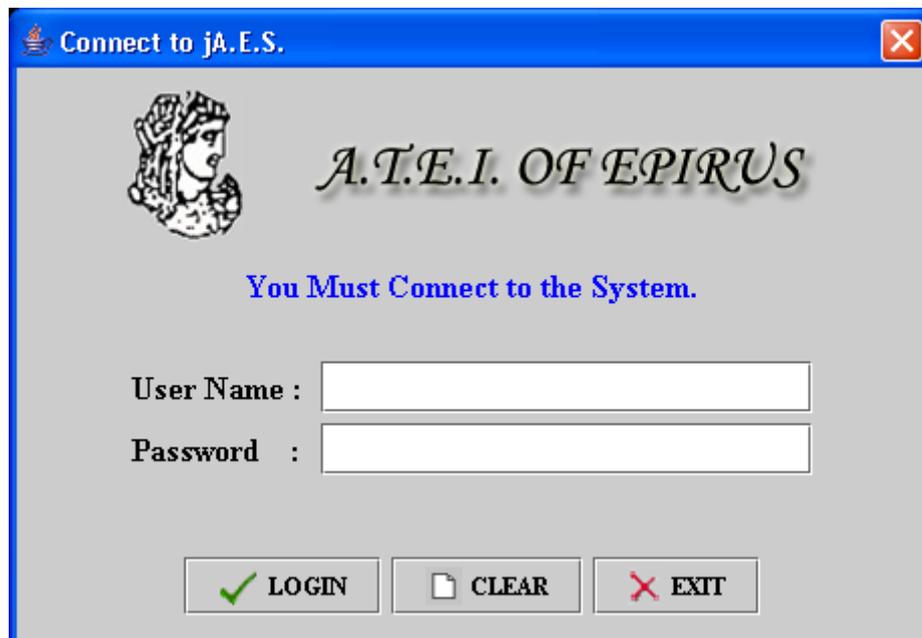
Επιπλέον, η μέθοδος της πολλαπλής επιλογής που υποστηρίζει το jA.E.S. βοηθά σε μεγάλο βαθμό τους σπουδαστές στην κατανόηση της εξεταστέας ύλης, χωρίς την αποστήθιση αυτής.

ΚΕΦΑΛΑΙΟ 6 : ΠΑΡΟΥΣΙΑΣΗ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ (jA.E.S.)

6.1 ΠΑΡΟΥΣΙΑΣΗ jA.E.S.

Το jA.E.S. είναι ένα ολοκληρωμένο σύστημα παροχής ηλεκτρονικών εξετάσεων, το οποίο είναι κατασκευασμένο σε παραθυρικό περιβάλλον εφαρμογής. Όπως κάθε εμπορική εφαρμογή απαρτίζεται από δέντρα, λίστες, τμήματα παραθύρου (τύπου καρτέλας, κυλιόμενα, διαιρούμενα), πίνακες κλπ, έτσι και το jA.E.S. μέσω των κλάσεων Java Foundation Classes (JFC) και το Swing, παρέχει στον χρήστη ένα ικανοποιητικό και ευχάριστο περιβάλλον εργασίας. Στην συνέχεια, παρουσιάζονται οι οθόνες που απαρτίζουν την εφαρμογή.

Κάθε χρήστης, σε οποιαδήποτε ομάδα κι αν ανήκει, πρέπει υποχρεωτικά να συνδεθεί στο σύστημα με το Όνομα Χρήστη και το Συνθηματικό που του αντιστοιχεί.



Εικόνα 6.1 : Κονσόλα Πιστοποίησης Χρήστη

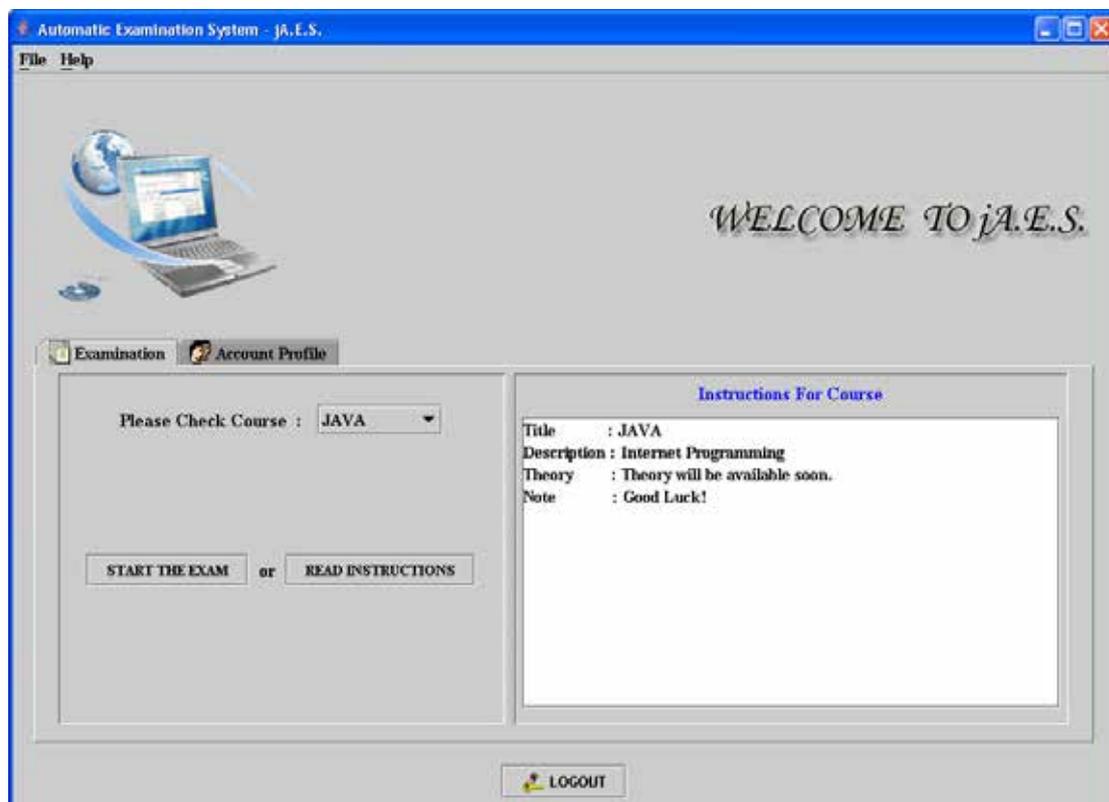
Αφού ολοκληρωθεί η διαδικασία πιστοποίησης του χρήστη από το σύστημα, ο χρήστης ανάλογα με το επίπεδο χρηστών στο οποίο ανήκει μεταφέρεται στην αντίστοιχη κεντρική κονσόλα της εφαρμογής jA.E.S.

6.2 ΣΠΟΥΔΑΣΤΗΣ ΒΗΜΑ ΠΡΟΣ ΒΗΜΑ

Η κεντρική κονσόλα του χρήστη Σπουδαστή αποτελείται από δυο βασικές καρτέλες κάθε μια από τις οποίες έχει συγκεκριμένες λειτουργίες.

Μέσω της πρώτης καρτέλας εμφανίζονται τα μαθήματα που παρακολουθεί ο συγκεκριμένος Σπουδαστής. Με την βοήθεια των κουμπιών *Start Exam* και *Read Instructions* του παρέχεται η δυνατότητα διεξαγωγής της εξέτασης ή εμφάνισης των πληροφοριών (Τίτλος, Περιγραφή, Θεωρία, Σημείωση) για το επιλεγμένο μάθημα αντίστοιχα.

Η εφαρμογή επιτρέπει στον Σπουδαστή την διεξαγωγή όσων εξετάσεων επιθυμεί, χωρίς περιορισμούς. Αυτό έχει σαν αποτέλεσμα την καλύτερη κατανόηση της ύλης πράγμα που συνεπάγεται την καλύτερη προετοιμασία για την τελική εξέταση.



Εικόνα 6.2.1 : Κεντρική Κονσόλα Εφαρμογής στο JA.E.S.

Στην περίπτωση που ο Σπουδαστής ξεκινήσει την εξέταση (πάτημα του κουμπιού *Start Exam*) η εφαρμογή του προβάλλει την κονσόλα της εξέτασης (Εικόνα 6.2.2). Πρόκειται για την σημαντικότερη κονσόλα της εφαρμογής μιας και υποστηρίζει την όλη διαδικασία της εξέτασης.

Ο Σπουδαστής έχει στην διάθεσή του ένα προεπιλεγμένο χρονικό διάστημα μέσα στο οποίο θα κληθεί να απαντήσει ένα προκαθορισμένο σύνολο ερωτήσεων πολλαπλής επιλογής.

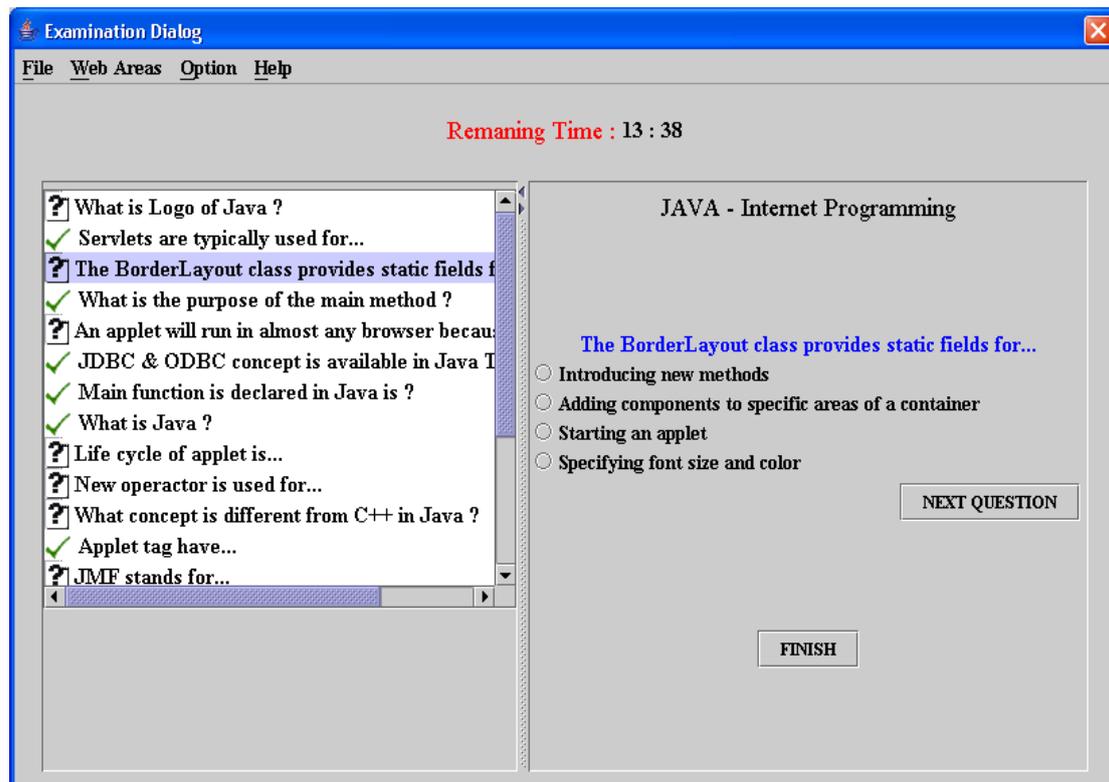
Η μετάβαση από την μια ερώτηση στην αμέσως επόμενη στην σειρά, πραγματοποιείται μέσω του κουμπιού *Next Question*. Σε περίπτωση όμως που ο Σπουδαστής θελήσει να απαντήσει σε μια ερώτηση της επιλογής του, έχει την δυνατότητα να την επιλέξει ο ίδιος κάνοντας κλικ πάνω της.

Η απάντηση των ερωτήσεων πραγματοποιείται, επιλέγοντας μια από τις πιθανές απαντήσεις που αντιστοιχούν σε κάθε ερώτηση. Σε περίπτωση που ο Σπουδαστής αμφιβάλλει για την ορθότητα της απάντησης μιας ερώτησης, έχει την δυνατότητα επιλογής νέας απάντησης χωρίς περιορισμούς. Επιπλέον, η κονσόλα της εξέτασης, περιέχει ενδείξεις για ποιες ερωτήσεις είναι απαντημένες ή όχι αντίστοιχα.

Ο Σπουδαστής έχει την δυνατότητα σύνδεσης με δυο από τις δημοφιλέστερες μηχανές αναζήτησης (Google, Yahoo) του Διαδικτύου μέσω της επιλογής “Web Areas” του μενού επιλογών.

Το πέρας της εξέτασης, σηματοδοτείται με δυο τρόπους :

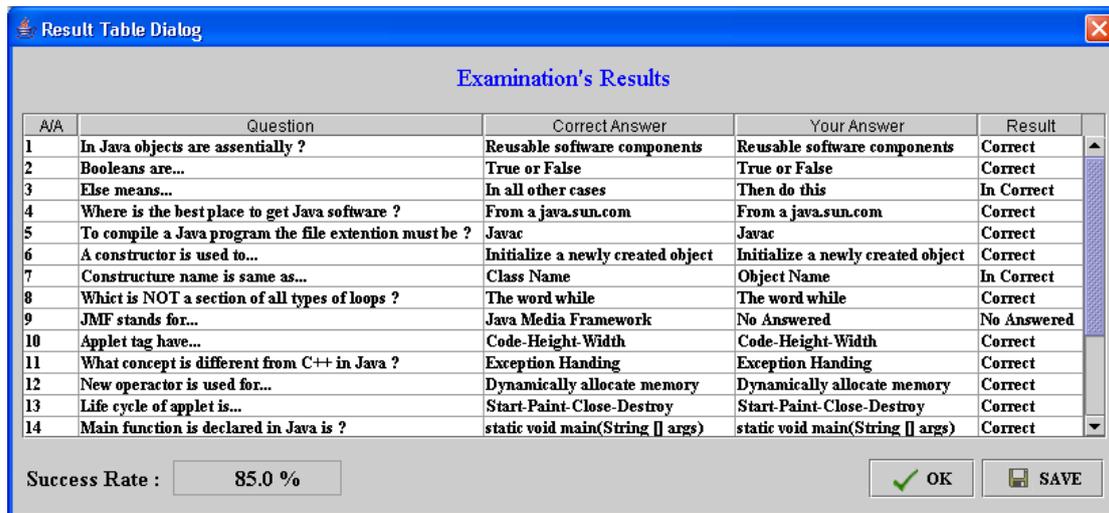
- Επιλέγοντας ο Σπουδαστής το πλήκτρο *Finish* και επιλέγοντας “Yes” στο παράθυρο διαλόγου επιβεβαίωσης που του εμφανίζεται.
- Λήξη του προεπιλεγμένου χρονικού διαστήματος.



Εικόνα 6.2.2 : Κονσόλα Εξέτασης στο JA.E.S.

Μετά την λήξη της εξέτασης ο Σπουδαστής ενημερώνεται άμεσα για το ποσοστό επιτυχίας | αποτυχίας του. Συγκεκριμένα, εμφανίζεται ένας αναλυτικός πίνακας ο οποίος περιέχει το σύνολο των ερωτήσεων της εξέτασης, την σωστή απάντηση, την

απάντηση που έδωσε ο Σπουδαστής και τέλος ένα χαρακτηρισμό για το αν η απάντηση είναι σωστή ή όχι.



A/A	Question	Correct Answer	Your Answer	Result
1	In Java objects are essentially ?	Reusable software components	Reusable software components	Correct
2	Booleans are...	True or False	True or False	Correct
3	Else means...	In all other cases	Then do this	In Correct
4	Where is the best place to get Java software ?	From a java.sun.com	From a java.sun.com	Correct
5	To compile a Java program the file extension must be ?	Javac	Javac	Correct
6	A constructor is used to...	Initialize a newly created object	Initialize a newly created object	Correct
7	Constructure name is same as...	Class Name	Object Name	In Correct
8	Whict is NOT a section of all types of loops ?	The word while	The word while	Correct
9	JMF stands for...	Java Media Framework	No Answered	No Answered
10	Applet tag have...	Code-Height-Width	Code-Height-Width	Correct
11	What concept is different from C++ in Java ?	Exception Handling	Exception Handling	Correct
12	New operator is used for...	Dynamically allocate memory	Dynamically allocate memory	Correct
13	Life cycle of applet is...	Start-Paint-Close-Destroy	Start-Paint-Close-Destroy	Correct
14	Main function is declared in Java is ?	static void main(String [] args)	static void main(String [] args)	Correct

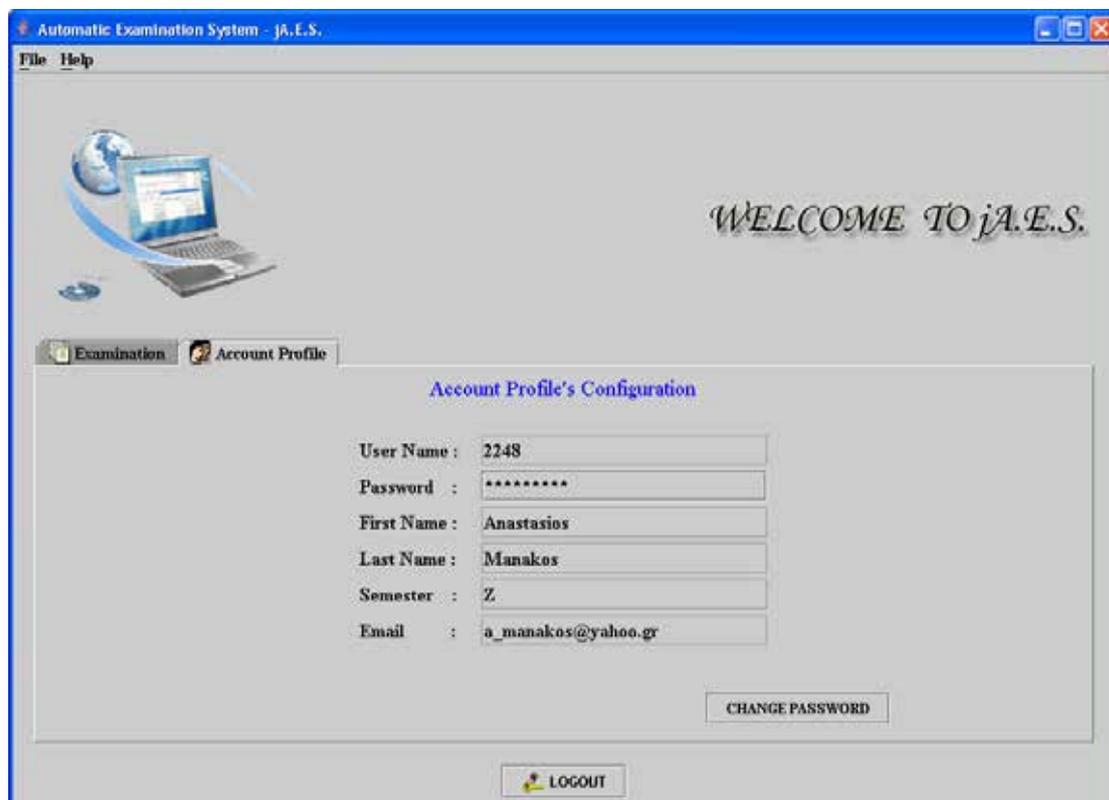
Success Rate : 85.0 %

OK SAVE

Εικόνα 6.2.3 : Κονσόλα Εμφάνισης Αποτελεσμάτων Εξέτασης στο JA.E.S.

Τέλος, ο Σπουδαστής έχει την δυνατότητα εξαγωγής της εξέτασης μέσω του κουμπιού *Save* σε αρχείο CSV.

Μέσω της δεύτερης καρτέλας ο Σπουδαστής έχει την δυνατότητα διαχείρισης του λογαριασμού του, βλέποντας όλα τα απαραίτητα προσωπικά του στοιχεία.



Automatic Examination System - JA.E.S.

File Help

WELCOME TO JA.E.S.

Examination Account Profile

Account Profile's Configuration

User Name : 2248

Password : *****

First Name : Anastasios

Last Name : Manakos

Semester : Z

Email : a_manakos@yahoo.gr

CHANGE PASSWORD

LOGOUT

Εικόνα 6.2.4 : Κονσόλα Εμφάνισης Αποτελεσμάτων Εξέτασης στο JA.E.S.

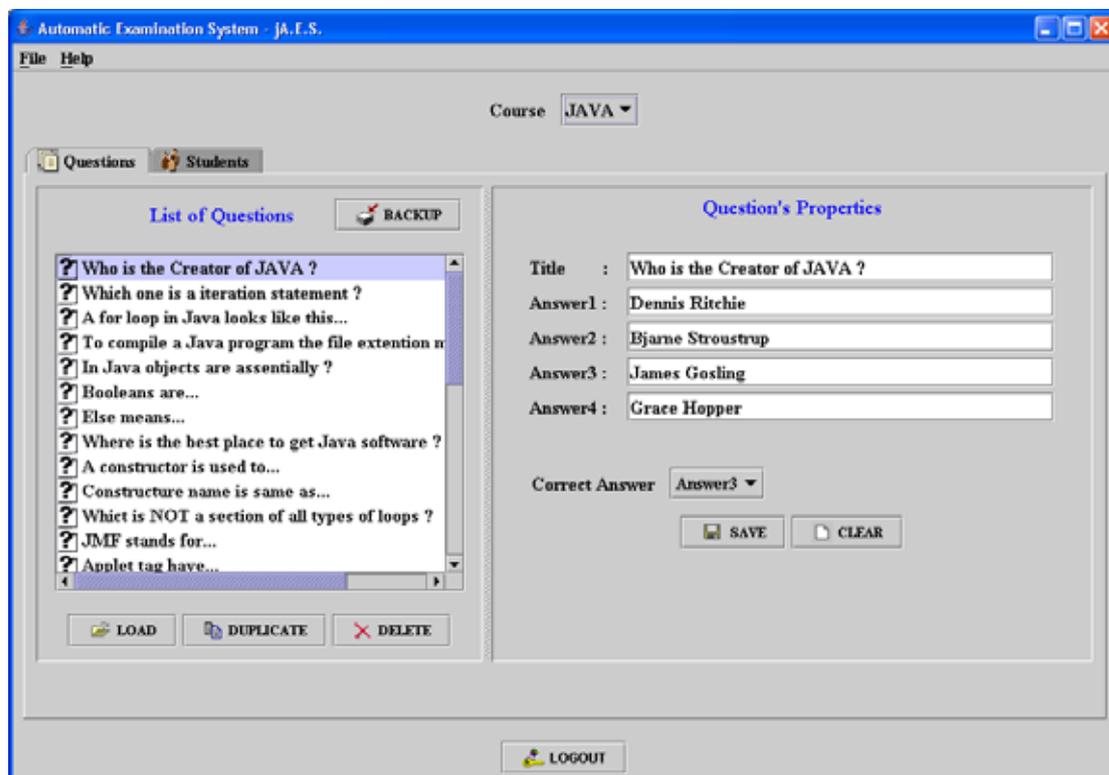
Θα πρέπει να σημειωθεί πως ο Σπουδαστής (όπως και κάθε ενεργός χρήστης του συστήματος jA.E.S.) έχει την δυνατότητα αλλαγής του Συνθηματικού του πατώντας το πλήκτρο *ChangePasswrd*.



Εικόνα 6.2.4 : Κονσόλα Αλλαγής Συνθηματικού στο jA.E.S.

6.3 ΚΑΘΗΓΗΤΗΣ ΒΗΜΑ ΠΡΟΣ ΒΗΜΑ

Η κεντρική κονσόλα του χρήστη Καθηγητή αποτελείται από δυο καρτέλες : μια για την διαχείριση των ερωτήσεων σε κάθε μάθημα που του αντιστοιχεί και μια για την προβολή των εξετάσεων για κάθε σπουδαστή που παρακολουθεί το συγκεκριμένο μάθημα.



Εικόνα 6.3.1 : Κονσόλα Διαχείρισης Ερωτήσεων στο jA.E.S.

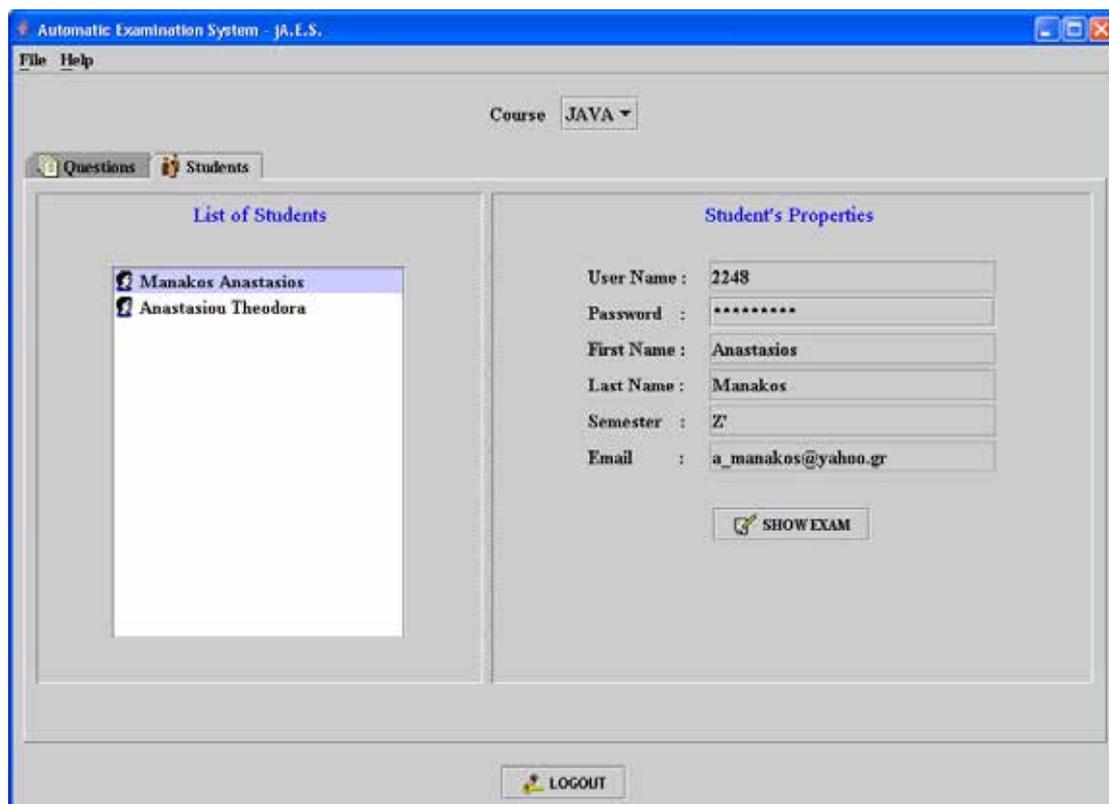
Στην εικόνα 6.3.1, βλέπουμε την κονσόλα διαχείρισης των Ερωτήσεων (Αντικειμένων Question). Στην κονσόλα, φαίνονται όλα τα μαθήματα που αντιστοιχούν στον συγκεκριμένο Καθηγητή, το καθένα από τα οποία περιέχει το δικό του σύνολο ερωτήσεων.

Κάθε Καθηγητής έχει τον πλήρη έλεγχο στο σύνολο των ερωτήσεων για κάθε μάθημα που του αντιστοιχεί. Έχει δυνατότητα εισαγωγής ερωτήσεων είτε μεμονωμένα πατώντας το πλήκτρο *Save*, είτε σαν σύνολο, πατώντας το πλήκτρο *Load*. Σε αυτή την περίπτωση, το σύνολο των ερωτήσεων, φορτώνονται από κάποιο αρχείο CSV.

Επίσης, έχει την δυνατότητα, δημιουργίας μιας ερώτησης με βάση μια ήδη υπάρχουσα πατώντας το πλήκτρο *Duplicate*.

Τέλος, μέσω του πλήκτρου *Backup*, έχει την δυνατότητα εξαγωγής του συνόλου των ερωτήσεων σε αρχείο CSV. Δηλαδή, δημιουργία αντιγράφου ασφαλείας (Backup) του συνόλου των ερωτήσεων.

Όπως έχει ήδη αναφερθεί, μέσω της δεύτερης καρτέλας (Εικόνα 6.3.2), ο Καθηγητής έχει την δυνατότητα προβολής των εξετάσεων για κάθε Σπουδαστή που παρακολουθεί το συγκεκριμένο μάθημα.



Εικόνα 6.3.2 : Κονσόλα Προβολής Μαθητών στο JA.E.S.

Στην κονσόλα φαίνονται όλοι οι Σπουδαστές που παρακολουθούν το συγκεκριμένο μάθημα, καθώς επίσης και όλες οι προσωπικές τους πληροφορίες. Μέσω του πλήκτρου *Show Exam* ο Καθηγητής έχει την δυνατότητα προβολής της εξέτασης για

κάθε Σπουδαστή (Εικόνα 6.3.3). Συγκεκριμένα, η κονσόλα προβολής των εξετάσεων εμφανίζει τα απαραίτητα στοιχεία για τον Σπουδαστή, το ποσοστό επιτυχίας | αποτυχίας του και έναν αναλυτικό πίνακα ο οποίος περιέχει το σύνολο των ερωτήσεων της εξέτασης, την σωστή απάντηση, την απάντηση που έδωσε ο Σπουδαστής και τέλος ένα χαρακτηρισμό για το αν η απάντηση είναι σωστή ή όχι.

Τέλος, μέσω του πλήκτρου *Print*, έχει την δυνατότητα εκτύπωσης της επιλεγμένης εξέτασης.

A/A	Question	Correct Answer	Your Answer	Result
1	In Java objects are essentially ?	Reusable software components	Reusable software components	Correct
2	Booleans are...	True or False	True or False	Correct
3	Else means...	In all other cases	Then do this	In Correct
4	Where is the best place to get Java software ?	From a java.sun.com	From a java.sun.com	Correct
5	To compile a Java program the file extention must be ?	Javac	Javac	Correct
6	A constructor is used to...	Initialize a newly created object	Initialize a newly created object	Correct
7	Constructure name is same as...	Class Name	Object Name	In Correct
8	Whict is NOT a section of all types of loops ?	The word while	The word while	Correct
9	JMF stands for...	Java Media Framework	No Answered	No Answered
10	Applet tag have...	Code-Height-Width	Code-Height-Width	Correct
11	What concept is different from C++ in Java ?	Exception Handling	Exception Handling	Correct
12	New operator is used for...	Dynamically allocate memory	Dynamically allocate memory	Correct
13	Life cycle of applet is...	Start-Paint-Close-Destroy	Start-Paint-Close-Destroy	Correct
14	Main function is declared in Java is ?	static void main(String [] args)	static void main(String [] args)	Correct

Success Rate : 85.0 %

OK PRINT

Εικόνα 6.3.3 : Κονσόλα Προβολής Εξετάσεων στο jA.E.S.

6.4 ΥΠΕΡΦΟΡΤΩΣΗ ΤΟΥ ΑΡΧΕΙΟΥ ΑΡΧΙΚΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Connect to jA.E.S.

UNIVERSITY OF IOANNINA

You Must Connect to the System.

User Name :

Password :

LOGIN CLEAR EXIT

Εικόνα 6.4 : Κονσόλα Πιστοποίησης Χρήστη

Θα πρέπει να σημειωθεί πως δεν πρόκειται για μια νέα έκδοση του jA.E.S. Η εφαρμογή κατά το ξεκίνημά της φορτώνει ένα προεπιλεγμένο αρχείο (jAES.properties), το οποίο περιέχει πληροφορίες για την αρχικοποίησή της. Ο χρήστης έχει την δυνατότητα υπερφόρτωσης του συγκεκριμένου αρχείου με αποτέλεσμα να επηρεάζει άμεσα τον τρόπο λειτουργία της εφαρμογής.

Δείτε περισσότερα όσον αφορά το αρχείο αρχικοποίησης της εφαρμογής στο **7^ο Κεφάλαιο “Λειτουργικός Σχεδιασμός”**.

ΚΕΦΑΛΑΙΟ 7 : ΛΕΙΤΟΥΡΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

7.1 ΕΙΣΑΓΩΓΗ

Στην περιγραφή του συστήματος jA.E.S. αναφέρθηκε πως υλοποιήθηκε το συγκεκριμένο σύστημα και ποια είναι τα βασικά χαρακτηριστικά του. Στην ουσία αυτό που θέλαμε να πετύχουμε ήταν η κατασκευή ενός συστήματος το οποίο να μπορεί να χρησιμοποιηθεί για οποιαδήποτε ανάγκη αυτόματης ηλεκτρονικής εξέτασης. Συνεπώς το σύστημα θα πρέπει να είναι αρκετά δυναμικό και ευέλικτο ώστε να αντιπροσωπεύει κάθε είδους εξέταση. Πως θα μπορούσε όμως να φτιαχτεί ένα αυτόματο σύστημα εξέτασης τόσο αυστηρό ώστε να εξυπηρετεί αυτούς τους σκοπούς;

Τη λύση σε αυτό το πρόβλημα την δίνει ο αντικειμενοστραφής προγραμματισμός. Κάθε συστατικό ενός τέτοιου συστήματος είναι και ένα ξεχωριστό αντικείμενο, ανεξάρτητο και αυτόνομο. Το κάθε συστατικό – αντικείμενο είναι δομημένο από συγκεκριμένες ιδιότητες και λειτουργίες αυστηρώς καθορισμένες ώστε να έχει την ευελιξία να χρησιμοποιείται σε κάθε γενική ηλεκτρονική εξέταση και σε όλες τις ειδικές απαιτήσεις του συστήματος jA.E.S.

7.2 ΧΡΗΣΤΕΣ ΤΟΥ jA.E.S.

Με τον όρο χρήστη εννοούμε κάθε επισκέπτη στο σύστημα. Οι επισκέπτες σε ένα Αυτόματο Σύστημα Εξέτασης μπορούν να κατηγοριοποιηθούν ανάλογα με την ιδιότητά τους, το σκοπό της επίσκεψής τους και των απαιτήσεων που έχουν από το σύστημα αυτό καθ' αυτό. Κύριοι επισκέπτες του συστήματος μας, μιας και μιλάμε για Αυτόματο Σύστημα Εξέτασης είναι οι Σπουδαστές και οι Καθηγητές.

Η ανάγκη όμως να ξεπεράσουμε κάποια λογικά προγραμματιστικά προβλήματα που πηγάζουν από τον απλό διαχωρισμό μας οδηγεί σε μια νέα κατηγοριοποίηση πιο αυστηρή και δομημένη :

- Ø Μαθητής (Student)
- Ø Καθηγητής (Professor)
- Ø Διαχειριστής (Administrator)

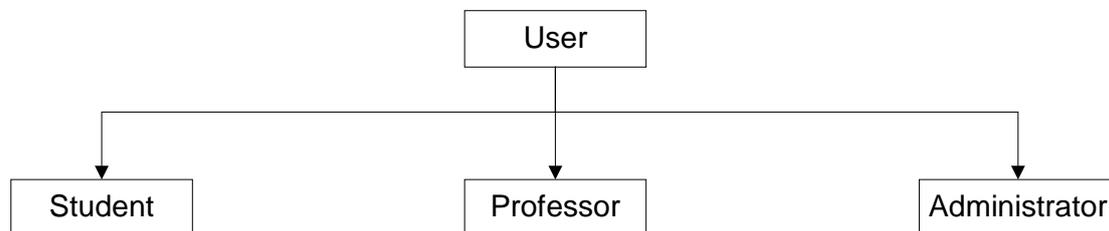
Με αυτόν τον διαχωρισμό μπορούμε να πούμε ότι πιάνουμε όλο το εύρος των πιθανών επισκεπτών.

Όπως είναι φανερό, η ύπαρξη διαφορετικών επισκεπτών στο σύστημα απαιτεί διαφοροποίηση του συστήματος και των λειτουργιών του ανάλογα με τον κάθε χρήστη. Επομένως, θα πρέπει να είναι ικανό να προσαρμόζεται ανάλογα με τις απαιτήσεις ή τις πιθανές απαιτήσεις που μπορεί να έχει ο κάθε επισκέπτης ως προς

την ιδιότητά του. Αντίθετα, ο κάθε επισκέπτης, θα πρέπει να υπακούει σε κανόνες και περιορισμούς που η ιδιότητά του, του προσδίδει στο σύστημα.

Προς το παρόν μιας και η υλοποίηση του συστήματος θα γίνει σε αντικειμενοστραφή γλώσσα προγραμματισμού, μπορούμε να πούμε πως η κάθε οντότητα χρήστη που δημιουργήσαμε αποτελεί και ένα ξεχωριστό αντικείμενο. Το κάθε αντικείμενο περιορίζεται σε συγκεκριμένα στοιχεία που αφορούν την ιδιότητα του ως χρήστη. Οι λειτουργίες του κάθε αντικείμενου αφορούνε μόνο την ανάκτηση και επεξεργασία των στοιχείων του και όχι την χρήση υπηρεσιών του συστήματος.

Η χρήση των αντικειμένων στη δομή που προσπαθούμε να χτίσουμε μας δίνει την δυνατότητα να χρησιμοποιήσουμε την κληρονομικότητα ως ένα σημαντικό κλειδί για το σχεδιασμό του συστήματος. Έτσι μπορούμε να δημιουργήσουμε ένα κληρονομικό δέντρο χρηστών μιας και η κάθε ιδιότητα επισκέπτη εμπεριέχει κοινά στοιχεία με τις άλλες ιδιότητες.



Εικόνα 7.1 : Αναπαράσταση του Δέντρου Χρηστών του Συστήματος

7.2.1 Χρήστης (User)

Το αντικείμενο User είναι το πρώτο στοιχείο του δέντρου κληρονομικότητας το οποίο κληροδοτεί τις ιδιότητές του στα υπόλοιπα αντικείμενα (χρήστες).

Ιδιότητες, θεωρούμε κάθε πληροφορία χρήσιμη και απαραίτητη, η οποία βοηθάει στην ταυτοποίηση και αναγνώριση του χρήστη από το σύστημα. Για παράδειγμα, το συνθηματικό ή το ψευδώνυμο του χρήστη.

7.2.2 Σπουδαστής (Student)

Είναι ίσως το πιο σημαντικό στοιχείο του συστήματός μας μιας και ο όρος “Αυτόματο Σύστημα Εξέτασης” και γενικότερα κάθε εφαρμογής που προσφέρει παρόμοιες υπηρεσίες δημιουργήθηκε από την ανάγκη του σπουδαστή να εξοικειωθεί όσο γίνεται περισσότερο με την χρήση των ηλεκτρονικών υπολογιστών και γενικότερα με τις νέες τεχνολογίες.

Στην συνέχεια θα χρησιμοποιηθεί ο όρος Student ως ένα αντικείμενο το οποίο έχει συγκεκριμένες ιδιότητες και λειτουργίες. Ένα αντικείμενο Student είναι ένας κλώνος User. Δηλαδή, πολλά από τα στοιχεία του, είναι στοιχεία που τα κληρονομεί από την ιδιότητά του, ως User. Πρόσθετα στοιχεία του Student, είναι όλες εκείνες οι

πληροφορίες που τον ταυτοποιούνε ως Σπουδαστή. Αναφορικά, ορισμένες από αυτές είναι το εξάμηνο, το πρόγραμμα μαθημάτων του, η βαθμολογία του κτλ.

7.2.3 Καθηγητής (Professor)

Ο καθηγητής είναι αναπόφευκτα κι αυτός ένα πολύ σημαντικό συστατικό του συστήματος. Είναι υπεύθυνος για την διεξαγωγή ηλεκτρονικών εξετάσεων. Στην συνέχεια θα χρησιμοποιηθεί ο όρος Professor ως ένα αντικείμενο που θα εξυπηρετεί τις ανάγκες ενός καθηγητή.

Ένα αντικείμενο Professor είναι κληρονόμος ενός αντικειμένου User. Δηλαδή, ένας Professor εκτός από την ιδιότητά του ως Καθηγητής περιλαμβάνει και την ιδιότητα του χρήστη που του επιτρέπει να συνδέεται και να επικοινωνεί με το σύστημα.

7.2.4 Διαχειριστής (Administrator)

Απαραίτητη προϋπόθεση στην καλή λειτουργία κάθε συστήματος είναι η ύπαρξη κάποιου ατόμου εξουσιοδοτημένου να επιβλέπει το σύστημα και να ελέγχει τη ροή και την λειτουργία του. Συνήθως, ο ρόλος αυτός ανατίθεται στους Διαχειριστές (Administrators). Τα άτομα αυτά είναι υπεύθυνα για την διόρθωση βλαβών, την επίλυση λαθών και γενικότερα για την ορθή λειτουργία του συστήματος. Επιπλέον, σημαντική είναι και η προσφορά οποιασδήποτε βοήθειας στους υπόλοιπους χρήστες του συστήματος.

Είναι φανερό ότι ένας Διαχειριστής είναι επιφορτισμένος με πλήθος υποχρεώσεων πάνω στο σύστημα που επιβλέπει.

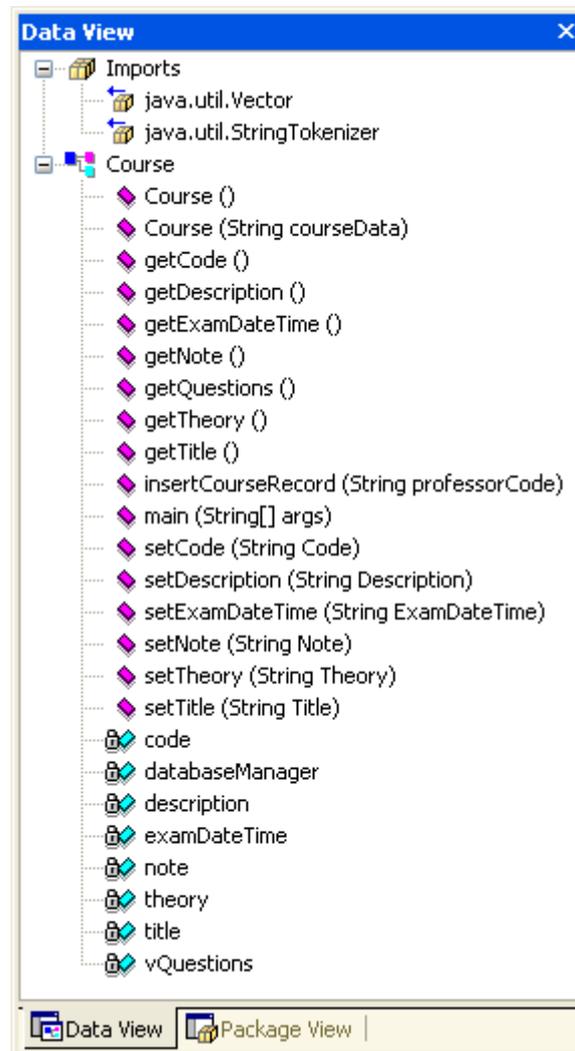
Στο παρόν σύστημα, τον ρόλο του διαχειριστή τον παίζει ο Administrator που είναι ένα αντικείμενο χρήστη κληρονόμος του User. Ο Administrator εμπεριέχει στοιχεία απαραίτητα για να τον αναγνωρίζει το σύστημα ως διαχειριστή καθώς και όλες τις πληροφορίες του χρήστη. Περιέχει επίσης πρόσθετα δικαιώματα που του επιτρέπουν να έχει περισσότερες δυνατότητες από οποιοδήποτε άλλο χρήστη στο σύστημα.

7.3 ΜΑΘΗΜΑ (COURSE)

Στόχος του κάθε μαθήματος είναι η σύλληψη βασικών θεωρητικών και πρακτικών γνώσεων των σπουδαστών. Μέσω αυτού, ο Σπουδαστής αναπτύσσει την μνημονική διαδικασία, τη λογική σκέψη, την αντιληπτική λειτουργία, την νοητική ανάπτυξη, την ιδιοσυγκρασία του, τα κίνητρα μάθησης και γενικότερα τις προσωπικές του εμπειρίες.

Το συγκεκριμένο αντικείμενο, θα αντιπροσωπεύει την τάξη των μαθημάτων. Θα πρέπει δηλαδή να διατηρεί μια συλλογή από αντικείμενα τύπου Question και θα πρέπει να έχει μεθόδους για να επιστρέψει το σύνολο των ερωτήσεων, τον κωδικό, την περιγραφή και τον τίτλο του.

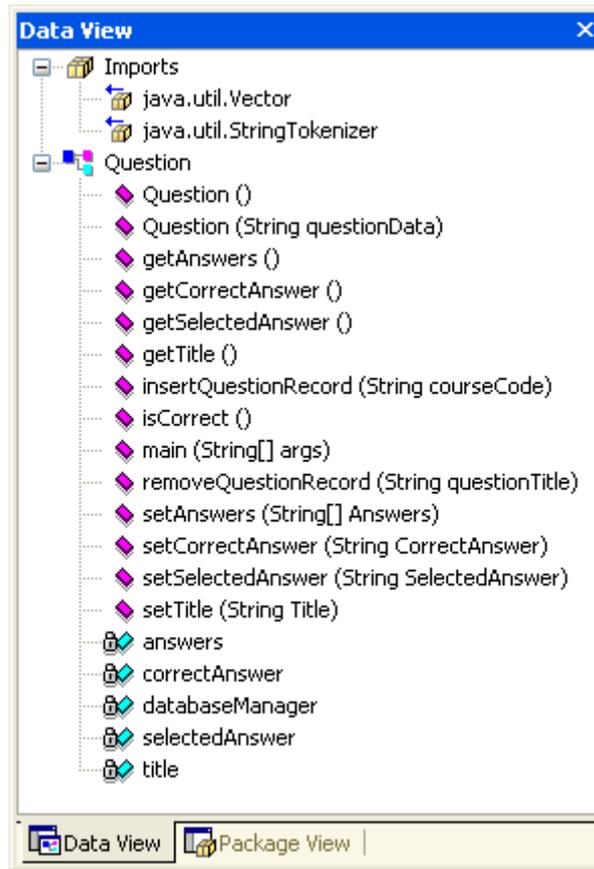
Κάθε Μάθημα – Course, διδάσκεται από συγκεκριμένο Καθηγητή – Professor ο οποίος έχει τον πλήρη έλεγχο του. Δηλαδή, έχει δικαιώματα (ενημέρωσης, εισαγωγής και διαγραφής) στο σύνολο των ερωτήσεων που αντιστοιχούν στο συγκεκριμένο μάθημα.



Εικόνα 7.2 : Διάγραμμα Τάξης Course

7.4 ΕΡΩΤΗΣΗ (QUESTION)

Το αντικείμενο Question, θα αντιπροσωπεύει την τάξη των ερωτήσεων. Δεν αρκεί γι' αυτό ένα String, μιας και μια ερώτηση θα πρέπει να συνοδεύεται από τις πιθανές απαντήσεις της, θα πρέπει να ξέρει ποια από αυτές είναι η σωστή απάντηση και θα πρέπει να έχει μεθόδους για να επιστρέφει τις απαντήσεις της, το κείμενο της ερώτησης, τη σωστή απάντηση και για διευκόλυνση, το αν η απάντηση που δόθηκε είναι σωστή ή όχι.



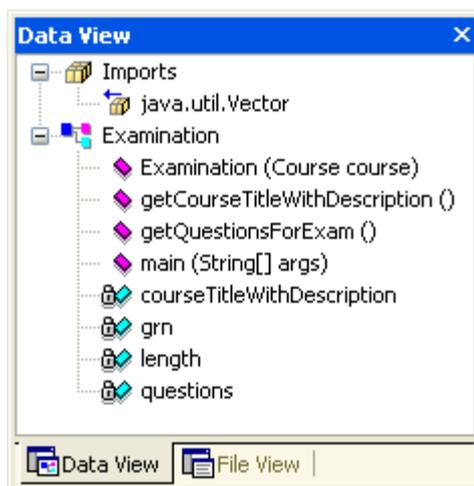
Εικόνα 7.3 : Διάγραμμα Τάξης Question

7.5 ΕΞΕΤΑΣΗ (EXAMINATION)

Η εξέταση είναι μια ευρέως χρησιμοποιούμενη έννοια σε πολλούς τομείς της εκπαιδευτικής δραστηριότητας. Αναφέρεται σε μια διαδικασία αξιολόγησης των γνώσεων που έχει κάποιος.

Το μειονέκτημα αυτής, είναι ότι ελέγχεται μόνο η μνημονική ικανότητα του εξεταζόμενου, χωρίς να εξετάζονται άλλες δεξιότητες. Αυτό συμβαίνει γιατί η εξέταση γίνεται σε μια ορισμένη ύλη και υπό ορισμένα βοηθήματα.

Στην συνέχεια θα χρησιμοποιήσουμε τον όρο Examination, ως ένα αντικείμενο που θα εξυπηρετεί τις ανάγκες μιας εξέτασης. Συγκεκριμένα, το αντικείμενο Examination θα αποτελεί μια συλλογή από αντικείμενα τύπου Question.

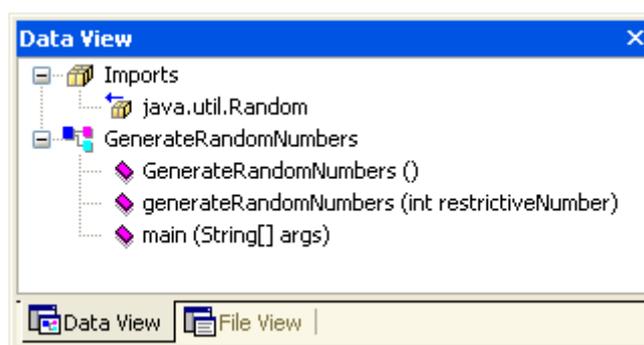


Εικόνα 7.4 : Διάγραμμα Τάξης Examination

7.6 ΓΕΝΝΗΤΡΙΑ ΠΑΡΑΓΩΓΗΣ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ (GENERATE RANDOM NUMBERS)

Όπως έχει ήδη αναφερθεί, το αντικείμενο Examination, θα διατηρεί όλες τις ερωτήσεις μια εξέτασης. Εκείνο που μένει τώρα να ξεκαθαριστεί είναι πως θα καθορίζεται ποιες ερωτήσεις θα αποτελούν την εξέταση.

Στο σύστημά jA.E.S., το Generate Random Numbers, είναι το αντικείμενο που είναι υπεύθυνο για την υλοποίηση αυτού του σεναρίου και συγκεκριμένα, με την βοήθεια ενός αλγορίθμου επιλέγει “τυχαία” από το σύστημα ένα καθορισμένο αριθμό ερωτήσεων.



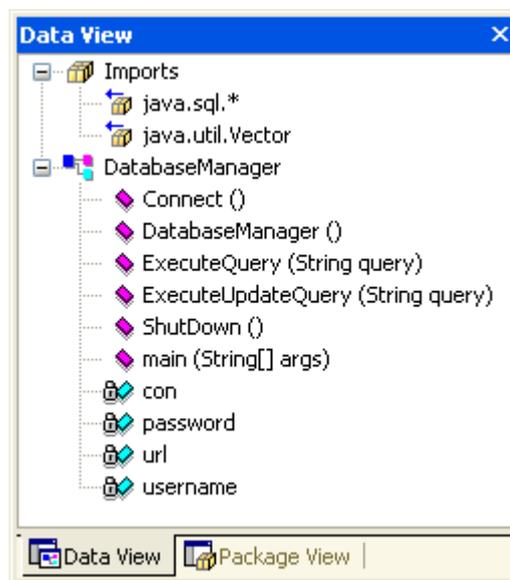
Εικόνα 7.5 : Διάγραμμα Τάξης Generate Random Numbers

7.7 ΔΙΑΧΕΙΡΙΣΤΗΣ ΒΔ (DATABASE MANAGER)

Το Database Manager είναι το αντικείμενο που πρόκειται να ανακτήσει δεδομένα από την Βάση Δεδομένων για λογαριασμό σας! Αν και είναι σημαντική τάξη και θα απαιτήσει αρκετή δουλειά για την υλοποίησή της, είναι πολύ απλή όσον αφορά τον τρόπο χρήσης της.

Το έργο της συγκεκριμένης τάξης αποτελείται από τα εξής βήματα :

1. Register τον Driver, με τον DriverManager χρησιμοποιώντας την μέθοδο [Class.forName \(\)](#).
2. Άνοιγμα ενός Connection με την Βάση Δεδομένων χρησιμοποιώντας τη μέθοδο [DriverManager.getConnection \(\)](#).
3. Δημιουργία ενός Statement χρησιμοποιώντας τη μέθοδο [createStatement \(\)](#).
4. Εκτέλεση του SQL ερωτήματος στην Βάση Δεδομένων χρησιμοποιώντας τη μέθοδο [executeQuery \(String SQL code\)](#) σε περίπτωση SELECT, διαφορετικά τη μέθοδο [executeUpdate \(String SQL code\)](#) σε περίπτωση INSERT, UPDATE κ' DELETE.

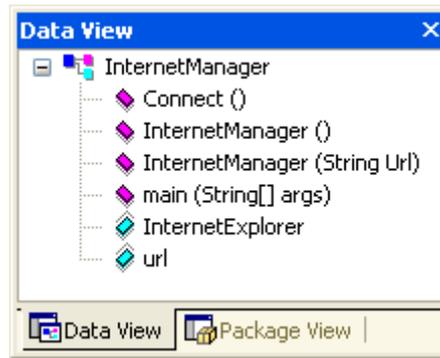


Εικόνα 7.6 : Διάγραμμα Τάξης Database Manager

7.8 ΔΙΑΧΕΙΡΙΣΤΗΣ ΔΙΑΔΙΚΤΥΟΥ (INTERNET MANAGER)

Το αντικείμενο Internet Manager είναι κατά κάποιο τρόπο μοναδική τάξη γι' αυτήν την εφαρμογή. Είναι υπεύθυνη για την φόρτωση HTML σελίδων μέσω του Uniform Resource Locator – URL που έχει καθοριστεί.

Ο Uniform Resource Locator, ή URL, αποτελεί μια αναπαράσταση της μεθόδου πρόσβασης και της θέσης κάποιου πόρου που βρίσκεται στο Internet. Σύμφωνα με την έννοια αυτής της εφαρμογής, θα ανακτώνται ιστοσελίδες από το Internet, οπότε το πρωτόκολλο που θα χρησιμοποιηθεί, όπως και του Διαδικτύου είναι το HTTP. Η σειρά χαρακτήρων που εξαρτάται από το πρωτόκολλο είναι η γνωστή, Web διεύθυνση. Για παράδειγμα, το URL για την ιστοσελίδα του τμήματος Τηλεπληροφορικής & Διοίκησης είναι το ακόλουθο : <http://www.teleinfom.teiep.gr>.

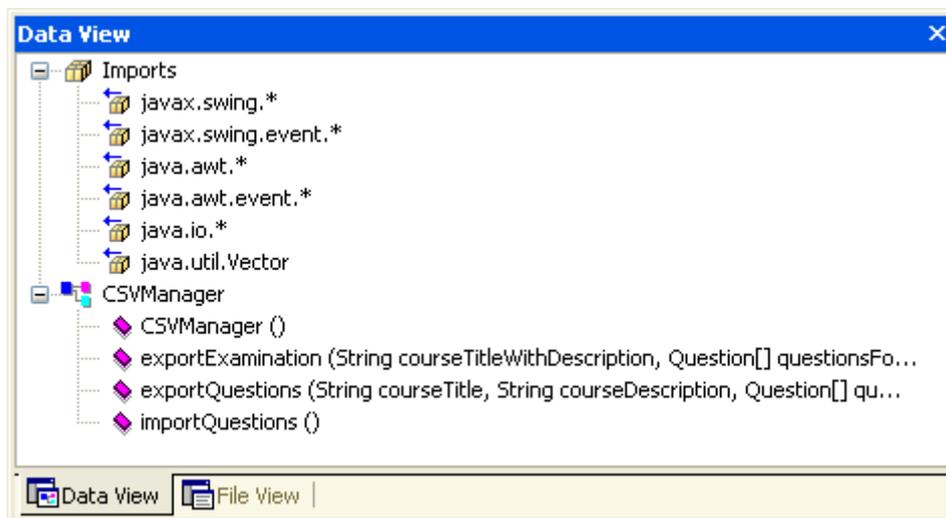


Εικόνα 7.7 : Διάγραμμα Τάξης Internet Manager

7.9 ΔΙΑΧΕΙΡΙΣΤΗΣ CSV (CSV MANAGER)

Το αντικείμενο CSV Manager, θα πρέπει να ξέρει τον τρόπο εισαγωγής και εξαγωγής δεδομένων σε αρχεία χωριζόμενων με κόμμα μεταβλητών (comma separated values – csv).

Ο καθορισμός της επικοινωνίας με το συγκεκριμένο αντικείμενο δεν πρέπει να είναι δύσκολος. Αυτό καθορίζεται σαν βοηθητικό αντικείμενο και αυτό σημαίνει ότι δεν διαθέτει ιδιότητες και δεν συντηρεί πληροφορίες για την κατάσταση στην οποία βρίσκεται.



Εικόνα 7.8 : Διάγραμμα Τάξης CSV Manager

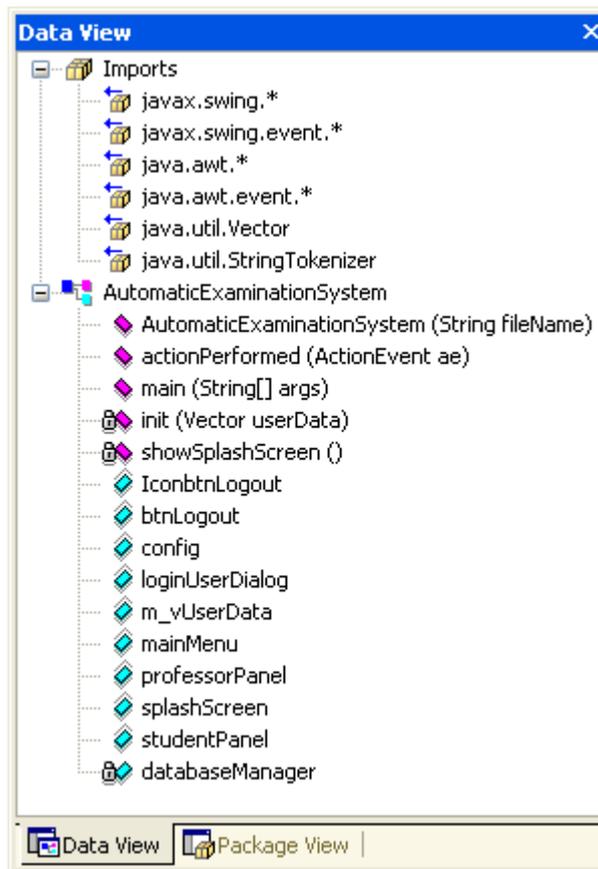
7.10 ΑΥΤΟΜΑΤΟ ΣΥΣΤΗΜΑ ΕΞΕΤΑΣΗΣ (AUTOMATIC EXAMINATION SYSTEM)

Αφού κατασκευαστούν και γίνουν κατανοητές οι τάξεις της εφαρμογής, χρειάζεται κάτι που να τις συνδέσει. Αυτό είναι έργο του αντικειμένου Automatic Examination System. Η συγκεκριμένη τάξη, εκτελεί δυο κύριες λειτουργίες : παρέχει όλα τα

στοιχεία ελέγχου “των οποίων είναι πατέρας” και διαχειρίζεται το περιβάλλον εργασίας.

Ακόμη, ρυθμίζει την λογική της εφαρμογής ζητώντας από τα διάφορα συστατικά να εκτελέσουν ενέργειες βασισμένες στο περιβάλλον εργασίας. Είναι η “κόλλα” που κρατάει τα πάντα συνδεδεμένα.

Λόγω της φύσης της, αυτή η τάξη δεν διαθέτει εξωτερικές μεθόδους που μπορεί να καλούνται – αυτή μόνο καλεί τις άλλες.



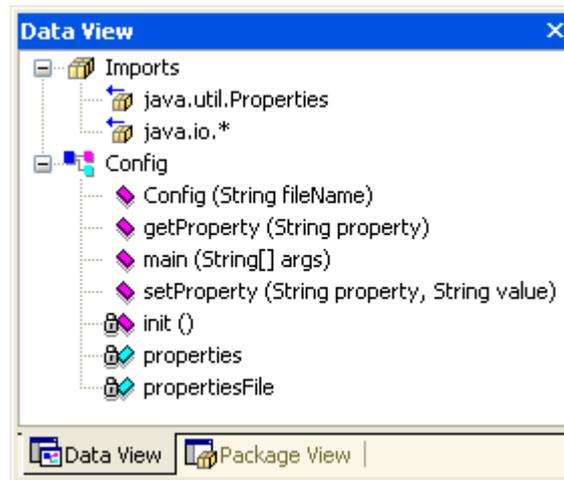
Εικόνα 7.9 : Διάγραμμα Τάξης Automatic Examination System

7.11 CONFIG

Το αντικείμενο Config είναι η σημαντικότερη τάξη της εφαρμογής. Είναι υπεύθυνη για την φόρτωση του αρχείου που περιέχει πληροφορίες οι οποίες χρησιμοποιούνται για την αρχικοποίηση της εφαρμογής.

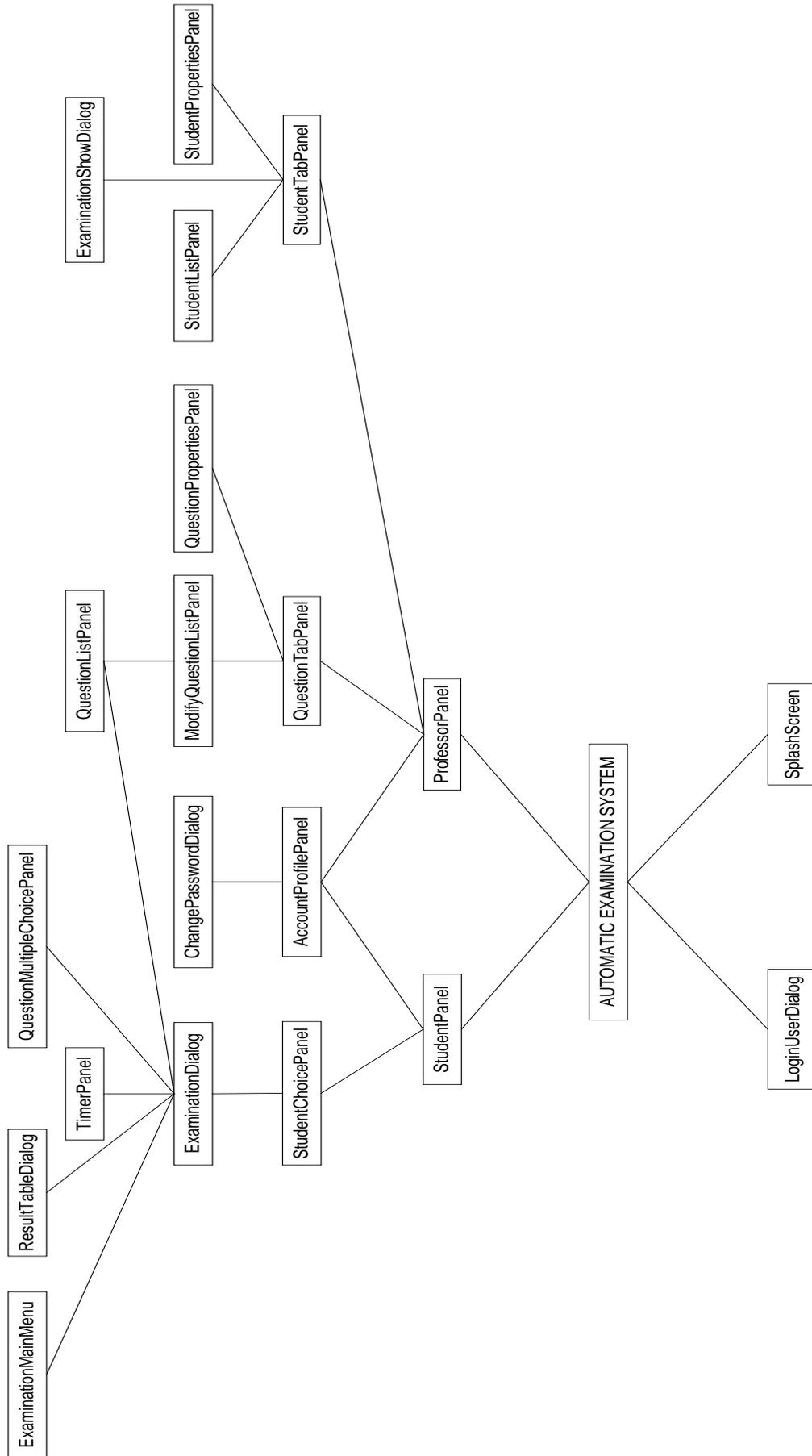
Η εφαρμογή εξ’ ορισμού, χρησιμοποιεί το αρχείο jAES.properties το οποίο μπορεί να υπερφορτωθεί σε περίπτωση επιλογής του χρήστη. Οι πληροφορίες του αρχείου κειμένου jAES.properties πρέπει να αποθηκεύονται ανά γραμμή μέσα σε αυτό και θα πρέπει να έχουν την εξής μορφή : <γνώρισμα> <οριοθέτης> <τιμή γνωρίσματος>.

Για παράδειγμα : IconSplashScreen = Images/splashScreen.gif. Εάν κάποια γραμμή αρχίζει με τον χαρακτήρα '#' τότε θεωρείται σχόλιο και παραλείπεται.



Εικόνα 7.10 : Διάγραμμα Τάξης Config

Στην συνέχεια ακολουθεί το διάγραμμα απεικόνισης των τάξεων που αποτελούν την γραφική διασύνδεση χρήστη – graphical user interface (Εικόνα 7.11).



Εικόνα 7.11 Διάγραμμα Τάξεων στο jA.E.S.

ΚΕΦΑΛΑΙΟ 8 : ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΑΥΤΟΜΑΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΞΕΤΑΣΗΣ

8.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ jA.E.S.

Στην περιγραφή του συστήματος jA.E.S. αναφέρθηκε πως η λειτουργία της εφαρμογής βασίζεται στην ύπαρξη ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων (Σ.Δ.Β.Δ.). Η χρησιμοποίηση ενός Σ.Δ.Β.Δ. ήταν απαραίτητο στοιχείο για την ολοκληρωμένη λειτουργία της εφαρμογής. Μερικοί από τους λόγους που την κάνουν αναντικατάστατη είναι :

- ∅ Η ανάγκη καταχώρησης των στοιχείων των χρηστών του Συστήματος και ιδιαίτερα τα στοιχεία των σπουδαστών και των καθηγητών.
- ∅ Η δυνατότητα διατήρησης των πληροφοριών της κάθε εξέτασης των σπουδαστών.
- ∅ Η ανάγκη καταχώρησης στοιχείων και πληροφοριών των μαθημάτων προς ηλεκτρονική εξέταση.

Γενικότερα, θα ήταν αδύνατη η λειτουργία μιας τέτοιας εφαρμογής χωρίς την καταχώρηση όλου του όγκου της πληροφορίας σε ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων.

Η εφαρμογή jA.E.S. όπως έχει ήδη αναφερθεί χρησιμοποιεί το Σ.Δ.Β.Δ. του MySQL Server το οποίο μπορεί να υποστηρίζει μεγάλες Βάσεις Δεδομένων με πολλούς πίνακες και αντίστοιχες εγγραφές. Ενδεικτικά, αναφέρεται ότι το εν λόγω σύστημα, υποστηρίζει ΒΔ με περισσότερες από 50.000.000 εγγραφές και περισσότερους από 60.000 πίνακες.

Τα “θεμέλια” λοιπόν της εφαρμογής jA.E.S. βασίζονται στην ύπαρξη της Βάσης Δεδομένων η οποία αποτελείται από 8 πίνακες οι οποίοι υπόκεινται στον τύπο μηχανής InnoDB. Αυτός ο τύπος πινάκων ενσωματώθηκε στην MySQL από την έκδοση 3.2.3 και μετά. Θα πρέπει να σημειωθεί πως είναι ο μόνος τύπος που υποστηρίζει τις συνδέσεις μεταξύ των πινάκων.

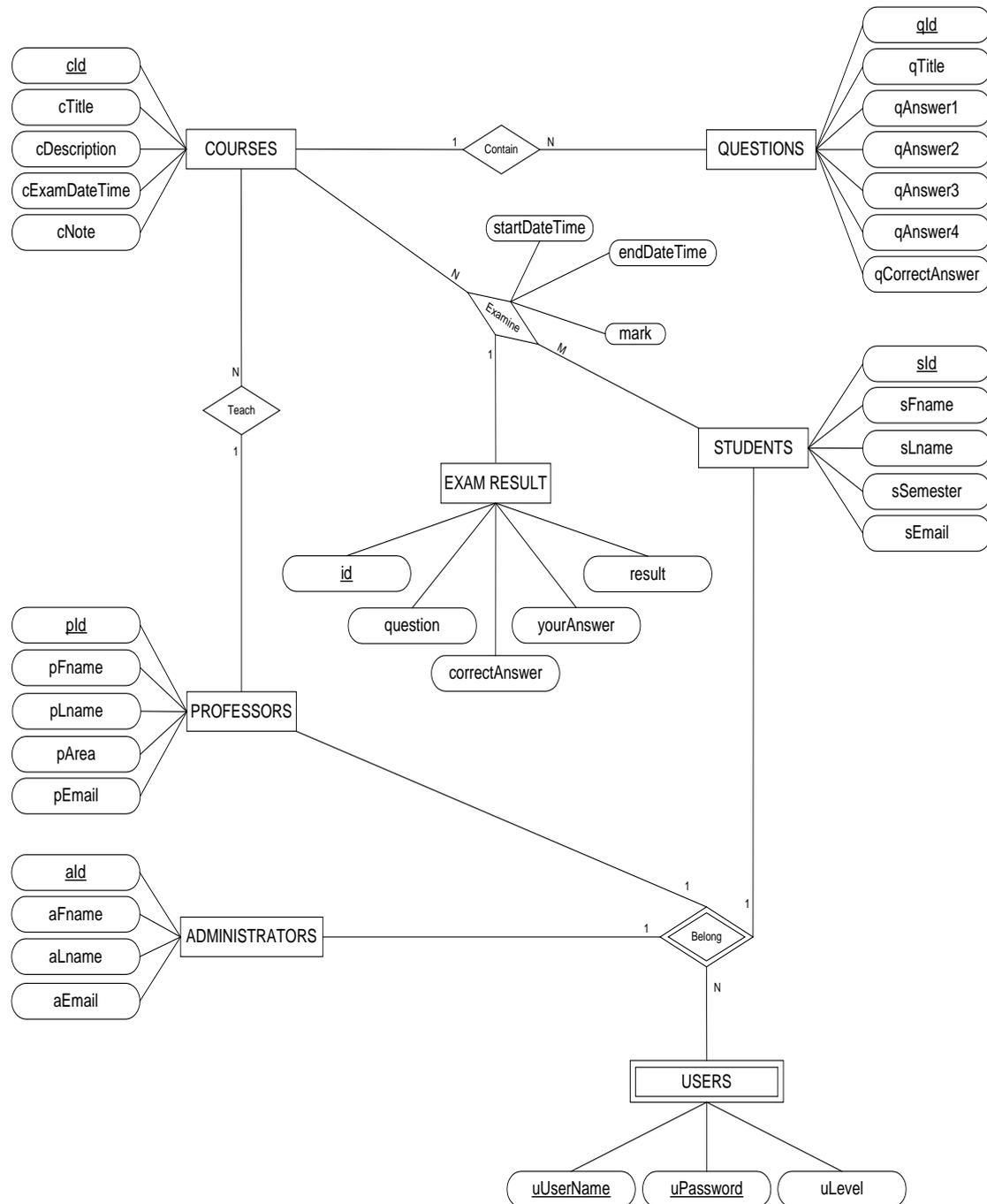
Στην υλοποίηση της ΒΔ του Αυτόματου Συστήματος Εξέτασης δεν θα γίνει ιδιαίτερη αναφορά μιας και η δομή και ο τρόπος λειτουργίας της, δεν επηρεάζει άμεσα το Σύστημα. Αντιθέτως, λειτουργεί ως υποβοήθημα για την ολοκληρωμένη λειτουργία του.

8.2 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ (ENTITY RELATIONSHIP MODEL, E – R MODEL)

Το Μοντέλο Οντοτήτων Συσχετίσεων βασίζεται στην προσέγγιση ότι ο πραγματικός κόσμος αποτελείται από οντότητες, χαρακτηριστικά και συσχετισμούς μεταξύ των

οντοτήτων. Αναπτύχθηκε για να διευκολύνει το σχεδιασμό μιας ΒΔ, επιτρέποντας τον ορισμό ενός σχήματος (schema) που αναπαριστά τη συνολική λογική δομή της ΒΔ.

Χρησιμοποιείται ως εργαλείο (σχεδιαστική γλώσσα) για την σημασιολογική ανάλυση μιας εφαρμογής και τον ορισμό του λογικού σχήματος. Στην συνέχεια, παρουσιάζεται το E – R Model της εφαρμογής jA.E.S.



Εικόνα 8.1 : Μοντέλο Οντοτήτων Συσχετίσεων του jA.E.S.

ΚΕΦΑΛΑΙΟ 9 : ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΒΛΗΜΑΤΑ – ΒΕΛΤΙΩΣΕΙΣ

9.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟ ΤΗΝ ΕΚΠΟΝΗΣΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ

Από την συγκεκριμένη πτυχιακή εργασία, πέραν των γνώσεων και της εμπειρίας που απέκτησα, θεωρώ ότι το σπουδαιότερο συμπέρασμα που αποκόμισα είναι ότι τίποτα δεν πρέπει να θεωρείται δεδομένο, τουλάχιστον όσον αφορά τους υπολογιστές. Συγκεκριμένα, όλες οι λειτουργίες είναι απαραίτητο να ελέγχονται σε όλες τις πιθανές συνθήκες και στις χειρότερες δυνατές περιπτώσεις και μάλιστα από διαφορετικά άτομα, ώστε να φανούν λάθη που πιθανόν κάποιος να μην έχει αντιληφθεί. Είναι τόσο διαφορετική η ανθρώπινη λογική που σε ορισμένες περιπτώσεις χρειάζεται μεγάλη προσπάθεια για να ανιχνεύσει κανείς τις αιτίες που ο υπολογιστής αντιλαμβάνεται κάτι με λανθασμένο τρόπο και να το προβλέψει.

Στόχος της παρούσας πτυχιακής εργασίας ήταν η **υλοποίηση** ενός Αυτόματου Συστήματος Εξέτασης. Ιδιαίτερη έμφαση δόθηκε στην διαδικασία της εξέτασης ώστε να υλοποιηθεί με τον καλύτερο δυνατό τρόπο. Στα πλαίσια της υλοποίησης, οι αποφάσεις για το σύστημα jAES πάρθηκαν ύστερα από ιδιαίτερη μελέτη και συζήτηση με άτομα που ήταν ειδικά πάνω σε αυτό το θέμα.

Η δημιουργία των προτύπων πάνω στα οποία στηρίχτηκε η όλη υλοποίηση, πέρασε και αυτή με τη σειρά της από πολλαπλές αλλαγές ώστε να καταλήξει στην πιο συμβατή μορφή της για όλους τους παράγοντες. Η προσπάθεια που έγινε ήταν όσο το δυνατόν πιο ολοκληρωμένη ώστε το σύστημα jAES να αποκτήσει την δική του ανεξάρτητη και ολοκληρωμένη προσωπικότητα στον χώρο των συστημάτων εξέτασης.

9.2 ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑ ΤΗΝ ΕΚΠΟΝΗΣΗ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ

Κατά την διάρκεια της υλοποίησης εμφανίστηκαν πολλά προβλήματα. Μερικά από τα σημαντικότερα, συγκαταλέγονται παρακάτω :

- Ø Ένα αρκετά δυσνόητο πρόβλημα ήταν η κατανόηση της σύνδεσης των αντικειμένων μεταξύ τους. Χρειάστηκε αρκετός χρόνος για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα και γενικότερα να υλοποιηθεί ολόκληρο το σύστημα.
- Ø Ένα πρόβλημα που ήταν πολύ περίεργο αλλά και δυσεπίλυτο ήταν η δημιουργία ενός “exception” της Java Virtual Machine που παρουσιάστηκε κατά την μεταγλώττιση των αρχείων java. Το συγκεκριμένο πρόβλημα παρουσιάστηκε ξαφνικά και αναπάντεχα. Σε τέτοιες περιπτώσεις, προτιμότερο είναι η επανεκκίνηση του υπολογιστή ή η επανεγκατάσταση του πακέτου λογισμικού λόγω πιθανής αλλοίωσής του.

```
Unexpected Signal : EXCEPTION_FLT_STACK_CHECK (0xc0000092) occurred at PC=0xC1
12
Function=[Unknown.]
Library=(N/A)
```

NOTE: We are unable to locate the function name symbol for the error just occurred. Please refer to release documentation for possible reason and solutions.

Current Java thread:

Dynamic libraries:

```
0x00400000 - 0x0040C000   C:\j2sdk1.4.2_05\bin\javac.exe
0x77F40000 - 0x77FF2000   C:\WINDOWS\System32\ntdll.dll
0x77E40000 - 0x77F3D000   C:\WINDOWS\system32\kernel32.dll
0x77DA0000 - 0x77E3D000   C:\WINDOWS\system32\ADVAPI32.dll
0x77C90000 - 0x77D05000   C:\WINDOWS\system32\RPCRT4.dll
0x77BE0000 - 0x77C33000   C:\WINDOWS\system32\MSVCRT.dll
0x00280000 - 0x002F3000   C:\DOCUME~1\MANAKO~1\LOCALS~1\Temp\iva2.tmp
```

.....

Another exception has been detected while we were handling last error.

Dumping information about last error:

ERROR REPORT FILE = (N/A)

PC = 0x00c1d212

SIGNAL = -1073741678

FUNCTION NAME = (N/A)

OFFSET = 0xFFFFFFFF

LIBRARY NAME = (N/A)

Please check ERROR REPORT FILE for further information, if there is any.

Good bye.

Παρουσίαση του προβλήματος

9.3 ΠΡΟΤΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ

Η παρούσα πτυχιακή εργασία αν και προσφέρει αρκετές δυνατότητες στους χρήστες όπως έχει αναλυθεί ως τώρα, παρουσιάζει και ορισμένα περιθώρια βελτίωσης, τα οποία αναφέρονται παρακάτω :

- Ø Καταρχήν, η εφαρμογή θα έπρεπε να υποστηρίζει την ανταλλαγή των δεδομένων και με Ελληνικούς χαρακτήρες.
- Ø Η εξέταση θα μπορούσε να απαρτίζεται από διάφορους τύπους ερωτήσεων (συμπλήρωσης κενών, True/False) και να μην περιορίζεται μόνο από ερωτήσεις πολλαπλής επιλογής.
- Ø Θα μπορούσε να υπάρχει και ένας ακόμη αλγόριθμος ο οποίος θα καθορίζει “τυχαία” την σειρά εμφάνισης των πιθανών απαντήσεων της κάθε ερώτησης.

ΗΜΕΡΟΛΟΓΙΟ ΠΟΡΕΙΑΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

ΗΜΕΡΟΜΗΝΙΑ	ΔΡΑΣΤΗΡΙΟΤΗΤΑ
20.09.2004	Ανάθεση του θέματος.
30.09.2004	Πρώτη συνάντηση με τον επιβλέποντα καθηγητή για την καταγραφή των κυρίων χαρακτηριστικών της εφαρμογής.
12.10.2004	Σχεδιασμός της εφαρμογής. Χρειάστηκε μια δωδεκάδα από σχέδια μέχρι να καταλήξουμε σε κάποιο που να μας ικανοποιεί.
09.11.2004	Ανάπτυξη των περιπτώσεων χρήσεις και προσδιορισμός των κλάσεων.
25.11.2004	Σχεδιασμός της Βάσης Δεδομένου της εφαρμογής.
30.12.2004	Υλοποίηση της εφαρμογής.
21.01.2005	Παρατηρήσεις του υπεύθυνου καθηγητή σχετικά με τον τρόπο λειτουργίας της εφαρμογής.
01.02.2005	Επέκταση του διαγράμματος E – R Model της Βάσης Δεδομένου της εφαρμογής.
01.03.2005	Συνάντηση με τον επιβλέποντα καθηγητή για τον τελικό έλεγχο της εφαρμογής.
05.03.2005	Αποστολή γενικών παρατηρήσεων σχετικά με το έγγραφο από τον καθηγητή.
06.03.2005	Συνάντηση με τον επιβλέποντα καθηγητή για τον τελικό έλεγχο του εγγράφου.
10.03.2005	Παρουσίαση Πτυχιακής Εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Δικτυακοί Τόποι

Sun Microsystems

Πιθανότατα, ο καλύτερος δικτυακός τόπος για να κοιτάξετε είναι της Sun Microsystems. Αυτοί ανέπτυξαν την Java και έθεσαν την τυποποίηση ώστε να είναι οι καταλληλότεροι στην παροχή ενημερωμένων και διεξοδικών πληροφοριών.

Ο δικτυακός της τόπος είναι : <http://java.sun.com>

Java Developer Connection (JDC)

Το Java Developer Connection, μέρος της Sun Microsystems, είναι το καταλληλότερο μέρος για πληροφορίες για διασκέψεις για την γλώσσα, για επίσημα βιβλία της Java και άλλους πόρους.

Μπορείτε να βρείτε το Java Developer Connection στην διεύθυνση : <http://developer.java.sun.com/developer/jdchome.html>

Κακαρόντζας Γεώργιος, Java: Μια αντικειμενοστραφής Γλώσσα Προγραμματισμού, <http://www.cs.teilar.gr/gkakaran/java/Index.html>

Βιβλία

Οδηγός της Java™ 2

Ένα από τα σημαντικότερα – ίσως το σημαντικότερο κατά την γνώμη μου βιβλίο εκμάθησης της Java.

Αξίζει να σημειωθεί ο συγγραφέας αυτού του βιβλίου, ο Steven Haines, ο οποίος εργάζεται ως μηχανικός λογισμικού για την εταιρεία Wonderware.

Πλήρες εγχειρίδιο της Java 2 Platform

Το “Πλήρες Εγχειρίδιο της Java 2” είναι ένα από τα καλύτερα εισαγωγικά βιβλία στον προγραμματισμό σε Java μέσω πραγματικών παραδειγμάτων.

Ο Rogers Cadenhead, ένας εκ των συγγραφέων αυτού του βιβλίου, συντηρεί την επίσημη περιοχή Web γι’ αυτό το βιβλίο στην παρακάτω διεύθυνση : <http://www.perfect.com/java21pre>

Άλλα βιβλία που μπορείτε να δείτε

Υπάρχουν πολλά βιβλία για παρόμοια θέματα, αλλά και για άλλα θέματα υπολογιστών, που μπορείτε να βρείτε στην περιοχή Web του εκδοτικού Οίκου Μ. Γκιούρδας, στην διεύθυνση : <http://www.mgiurdas.gr>

Περιοδικά

JavaWorld

Ένα από τα καλύτερα ηλεκτρονικά περιοδικά που έχει εκδοθεί για να εξυπηρετεί την κοινωνία των προγραμματιστών της Java. Το JavaWorld διατίθεται δωρεάν στο Web στην διεύθυνση : <http://www.javaworld.com>.

Το JavaWorld εκδίδει συχνά εκπαιδευτικά άρθρα, μαζί με ειδήσεις για ανάπτυξη της Java και άλλα χαρακτηριστικά που ενημερώνονται μηνιαία. Η μορφή που είναι μόνο για Web παρέχει ένα πλεονέκτημα σε σχέση με τα αντίστοιχα τυπωμένα περιοδικά στην περιοχή των άρθρων που περιέχουν πληροφορίες για το πώς να κάνετε διάφορες εργασίες. Όταν σε ένα άρθρο συζητείται ένα συγκεκριμένο θέμα προγραμματισμού, το JavaWorld σας προσφέρει και μια μικροεφαρμογή Java, που κάνει μια επίδειξη του θέματος.

Java Report Online

Το Java Report Online παρέχει ενημερωμένες πληροφορίες προϊόντων, τις τελευταίες τεχνικές Java, όπως επίσης και όλες τις καταχωρήσεις που εμφανίζονται στην τυποποιημένη έκδοση του Java Report.

Μπορεί να βρεθεί στο : <http://www.javareport.com>

ΠΑΡΑΡΤΗΜΑ Α' – ΕΝΔΕΙΚΤΙΚΟΙ ΚΩΔΙΚΕΣ

Το παράρτημα αυτό παρουσιάζει ένα μέρος του κώδικα της εφαρμογής που αναλύθηκε στην παρούσα πτυχιακή εργασία. Το σύνολο του κώδικα παρατίθεται σε ηλεκτρονική μορφή στο αντίστοιχο cd που συνοδεύεται με την εργασία.

ΑΝΤΙΚΕΙΜΕΝΟ Database Manager

```
// Αρχείο : DatabaseManager.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή του πακέτου Structured Query Language (SQL)
import java.sql.*;

// Εισαγωγή της κλάσης Vector του πακέτου util
import java.util.Vector;

// Κλάση DatabaseManager
public class DatabaseManager {

    // Δημιουργία των μεταβλητών μέλους (member variable)
    private Connection con = null;
    private final String url =
"jdbc:mysql://localhost/AutomaticExaminationSystem";
    private final String username = "root";
    private final String password = "";

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public DatabaseManager () {
    }

    /*****
    * Σύνδεση με την Βάση Δεδομένων *
    * Επιστρέφει true ---> σε Επιτυχία *
    * false ---> σε Αποτυχία *
    *****/
    public Boolean Connect () {

        try {

            // Φόρτωση της τάξης του JDBC Driver στην JVM
            Class.forName ("com.mysql.jdbc.Driver");

            // Δημιουργία σύνδεσης με την Βάση Δεδομένων
            this.con = DriverManager.getConnection
(this.url,this.username,this.password);

            if (!this.con.isClosed ()) {
                System.out.println ("Successfully connected
to " + "MySQL server...");
            }
        }
        catch (ClassNotFoundException cnfex) {

            // Λάθος στην Class.forName ()
            // Καταχώρηση του λάθους και επιστροφή false

```

```
        System.err.println ("Failed to load JDBC driver.");

        cnfex.printStackTrace ();

        return (new Boolean (false));
    }
    catch (SQLException sqlex) {

        // Λάθος στην DriverManager.getConnection ()
        // Καταχώρηση του λάθους και επιστροφή false
        System.err.println ("Unable to connect.");

        sqlex.printStackTrace ();

        return (new Boolean (false));
    }
}

return (new Boolean (true));
}

/*****
 * Αποσύνδεση με την Βάση Δεδομένων *
 * Επιστρέφει true ---> σε Επιτυχία *
 * false ---> σε Αποτυχία *
 *****/
public Boolean ShutDown () {

    try {

        if (this.con != null) {

            // Αποσύνδεση με την Βάση Δεδομένων
            this.con.close ();

            System.out.println ("Successfully
disconnected to " + "MySQL server...");
        }
    }
    catch (SQLException sqlex) {

        // Λάθος στην con.close ()
        // Καταχώρηση του λάθους και επιστροφή false
        System.err.println ("Unable to disconnect.");

        sqlex.printStackTrace ();

        return (new Boolean (false));
    }

    return (new Boolean (true));
}

/*****
 * Εκτέλεση ερωτημάτων INSERT - UPDATE - DELETE *
 * προς την Βάση Δεδομένων. *
 * Επιστροφή του πλήθους των γραμμών που ενημερώθηκαν *
 *****/
public int ExecuteUpdateQuery (final String query) {

    int updateCount = 0;
```

```
try {  
    // Δημιουργία εντολής SQL  
    Statement stmt = con.createStatement ();  
  
    // Εκτέλεση του ερωτήματος SQL  
    stmt.executeUpdate (query);  
  
    // Πλήθος των εγγραφών που ενημερώθηκαν  
    updateCount = stmt.getUpdateCount ();  
  
    if (stmt != null)  
        stmt.close ();  
}  
catch (SQLException sqllex) {  
  
    // Λάθος στην stmt.executeUpdate ()  
    // Καταχώρηση του λάθους  
    System.err.println ("Cannot execute query to  
database server.");  
  
    sqllex.printStackTrace ();  
}  
  
return (updateCount);  
}  
  
/*****  
 * Εκτέλεση ερωτημάτων SELECT προς την Βάση Δεδομένων *  
 * Επιστροφή διανύσματος - vector με τα αντικείμενα *  
 *****/  
public Vector ExecuteQuery (final String query) {  
  
    // Δημιουργία του Vector  
    Vector vResult = new Vector ();  
  
    try {  
  
        // Δημιουργία εντολής SQL  
        Statement stmt = con.createStatement ();  
  
        // Εκτέλεση του ερωτήματος SQL  
        stmt.executeQuery (query);  
  
        // Πίνακας αποτελεσμάτων του ερωτήματος SQL  
        ResultSet rs = stmt.getResultSet ();  
  
        ResultSetMetaData rsmd = rs.getMetaData ();  
  
        // Επιστροφή του αριθμού των στηλών του ResultSet  
        int numberOfColumns = rsmd.getColumnCount ();  
  
        String str = null;  
  
        while (rs.next ()) {  
            str = "";  
  
            for (int i=1;i<=numberOfColumns;++i) {
```

```
        Object object = rs.getObject (i);

        if (i<numberOfColumns)
            str = str + object.toString () +
",,";
        else
            str += object.toString ();
    }

    // Προσθήκη της γραμμής στο διάνυσμα - vector
    vResult.addElement (str);
}

if (rs != null)
    rs.close ();

if (stmt != null)
    stmt.close ();
}
catch (SQLException sqlex) {

    // Λάθος στην stmt.executeQuery ()
    // Καταχώρηση του λάθους
    System.err.println ("Cannot execute query to
database server.");

    sqlex.printStackTrace ();
}

return (vResult);
}

// Κύριο σημείο εισόδου της εφαρμογής στην κλάση
public static void main (String [] args) {

    // Δημιουργία αντικειμένου DatabaseManager
    DatabaseManager dm = new DatabaseManager ();

    // Κλήση της μεθόδου Connect ()
    dm.Connect ();

    // Κλήση της μεθόδου ShutDown ()
    dm.ShutDown ();
}
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Question

```
// Αρχείο : Question.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του util
import java.util.Vector;
import java.util.StringTokenizer;

// Κλάση Question
public class Question extends Object {

    // Δημιουργία μεταβλητών μέλους (member variable)
    private String title;
    private String [] answers = new String [4];
    private String correctAnswer;
    private String selectedAnswer;
    private DatabaseManager databaseManager = new DatabaseManager

();

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public Question () {
    }

    // Υπερφορτωμένη Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public Question (final String questionData) {

        /*
        * Διαδικασία διαχωρισμού σειράς λέξεων που χωρίζονται
από κάποιον
        * διαχωριστήρα στα συστατικά της
        */
        String separator = ",";
        StringTokenizer st = new StringTokenizer
(questionData,separator);

        // Ανάκτηση του αριθμού των tokens στο στοιχείο του
διανύσματος
        if (st.countTokens () != 6) {

            // Λάθος, όχι ο σωστός αριθμός tokens
            // Καταχώρηση του λάθους
            System.err.println ("Error ---> Constructor
Question." +
                                "Expected 6
tokens,but received : " + st.countTokens ()
                                );

            return;
        }

        // Εξαγωγή των tokens και ορισμός των ιδιοτήτων
        this.title = st.nextToken ();

        for (int i=0;i<this.answers.length;i++)
            this.answers [i] = st.nextToken ();

        this.correctAnswer = st.nextToken ();
    }
}
```

```
// Μέθοδοι ιδιοτήτων SET/GET
public void setTitle (final String Title) {

    this.title = Title;
}

public String getTitle () {

    return (this.title);
}

public void setAnswers (final String [] Answers) {

    this.answers = Answers;
}

public String [] getAnswers () {

    return (this.answers);
}

public void setCorrectAnswer (final String CorrectAnswer) {

    this.correctAnswer = CorrectAnswer;
}

public String getCorrectAnswer () {

    return (this.correctAnswer);
}

public void setSelectedAnswer (final String SelectedAnswer) {

    this.selectedAnswer = SelectedAnswer;
}

public String getSelectedAnswer () {

    return (this.selectedAnswer);
}

/*****
 * Έλεγχος, αν η απάντηση που δόθηκε *
 * είναι σωστή ή όχι *
 * Επιστρέφει true ---> σε Επιτυχία *
 * false ---> σε Αποτυχία *
 *****/
public boolean isCorrect () {

    return (this.selectedAnswer.equals (this.correctAnswer));
}

/*****
 * Εισαγωγή ενός αντικειμένου Question στην Βάση Δεδομένων *
 * Επιστροφή του πλήθους των γραμμών που ενημερώθηκαν *
 *****/
public int insertQuestionRecord (final String courseCode) {

    int updateCount = 0;
```

```

        /*
        * Κλήση της συνάρτησης Connect () του αντικειμένου
DatabaseManager
        * για την δημιουργία σύνδεσης με την Βάση Δεδομένων
        */
        if (databaseManager.Connect ().booleanValue ()) {

            int counter = 0;

            // Δημιουργία του SQL ερωτήματος

            final String query = "INSERT INTO Questions
(qtitle,qanswer1,qanswer2,qanswer3,qanswer4,qcorrectAnswer,cid) " +
                "VALUES (' +
this.title + "',' + this.answers [counter++] + "',' + this.answers
[counter++] + "',' +
                this.answers
[counter++] + "',' + this.answers [counter++] + "',' +
this.correctAnswer + "',' + courseCode + "')";

            /*
            * Κλήση της συνάρτησης ExecuteUpdateQuery () του
αντικειμένου DatabaseManager
            * για την εκτέλεση του ερωτήματος - query προς την
Βάση Δεδομένων
            */
            updateCount =
this.databaseManager.ExecuteUpdateQuery (query);

            /*
            * Κλήση της συνάρτησης ShutDown () του
αντικειμένου DatabaseManager
            * για τον τερματισμό της σύνδεσης από την Βάση
Δεδομένων
            */
            this.databaseManager.ShutDown ();
        }

        return (updateCount);
    }
}

```

```

/*****
* Ενημέρωση ενός αντικειμένου Question στην Βάση Δεδομένων *
* Επιστροφή του πλήθους των γραμμών που ενημερώθηκαν *
*****/

```

```

public int updateQuestionRecord (final String questionTitle) {

```

```

    int updateCount = 0;

```

```

        /*
        * Κλήση της συνάρτησης Connect () του αντικειμένου
DatabaseManager
        * για την δημιουργία σύνδεσης με την Βάση Δεδομένων
        */
        if (databaseManager.Connect ().booleanValue ()) {

```

```

            int count = 0;

```

```

            // Δημιουργία του SQL ερωτήματος
            final String query = "UPDATE Questions " +

```

```

this.title + "',' +
this.answers [count++] + "',' +
+ this.correctAnswer + "' " +
questionTitle + "';";

        "SET qtitle = '" +
        "qanswer1 = '" +
        "qanswer2 = '" +
        "qanswer3 = '" +
        "qanswer4 = '" +
        "qcorrectanswer = '"
        "WHERE qtitle = '" +

        /*
        * Κλήση της συνάρτησης ExecuteUpdateQuery () του
        αντικειμένου DatabaseManager
        * για την εκτέλεση του ερωτήματος - query προς την
        Βάση Δεδομένων
        */
        updateCount =
this.databaseManager.ExecuteUpdateQuery (query);

        /*
        * Κλήση της συνάρτησης ShutDown () του
        αντικειμένου DatabaseManager
        * για τον τερματισμό της σύνδεσης από την Βάση
        Δεδομένων
        */
        this.databaseManager.ShutDown ();
    }

    return (updateCount);
}

/*****
* Διαγραφή ενός αντικειμένου Question από την Βάση Δεδομένων *
* Επιστροφή του πλήθους των γραμμών που ενημερώθηκαν *
*****/
public int removeQuestionRecord (final String questionTitle) {

    int updateCount = 0;

    /*
    * Κλήση της συνάρτησης Connect () του αντικειμένου
    DatabaseManager
    * για την δημιουργία σύνδεσης με την Βάση Δεδομένων
    */
    if (databaseManager.Connect ().booleanValue ()) {

        // Δημιουργία του SQL ερωτήματος

        final String query = "DELETE FROM Questions " +
            "WHERE
Questions.qtitle = '" + questionTitle + "';";

        /*
        * Κλήση της συνάρτησης ExecuteUpdateQuery () του
        αντικειμένου DatabaseManager

```

```

        * για την εκτέλεση του ερωτήματος - query προς την
        Βάση Δεδομένων
        */
        updateCount =
this.databaseManager.ExecuteUpdateQuery (query);

        /*
        * Κλήση της συνάρτησης ShutDown () του
        αντικειμένου DatabaseManager
        * για τον τερματισμό της σύνδεσης από την Βάση
        Δεδομένων
        */
        this.databaseManager.ShutDown ();
    }

    return (updateCount);
}

// Κύριο σημείο εισόδου της εφαρμογής στην κλάση
public static void main (String [] args) {

    final String questionData = "Who is the Creator of JAVA
?,Dennis Ritchie,Bjarne Stroustrup,James Gosling,Grace
Hopper,Answer3";

    // Δημιουργία αντικειμένου Question
    Question question = new Question (questionData);

    // Κλήση της συνάρτησης insertQuestionRecord ()
    question.insertQuestionRecord ("c25");
}
}
```

ANTIKEIMENO Course

```
// Αρχείο : Course.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του util
import java.util.Vector;
import java.util.StringTokenizer;

// Κλάση Course
public class Course extends Object {

    // Δημιουργία μεταβλητών μέλους (member variable)
    private String code;
    private String title;
    private String description;
    private String examDateTime;
    private String theory;
    private String note;
    private Vector vQuestions = new Vector ();
    private DatabaseManager databaseManager = new DatabaseManager
();

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public Course () {
    }

    // Υπερφορτωμένη Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public Course (final String courseData) {

        /*
        * Διαδικασία διαχωρισμού σειράς λέξεων που χωρίζονται
από κάποιον
        * διαχωριστήρα στα συστατικά της
        */
        String separator = ",";
        StringTokenizer st = new StringTokenizer
(courseData,separator);

        // Ανάκτηση του αριθμού των tokens στο στοιχείο του
διανύσματος
        if (st.countTokens () != 6) {

            // Λάθος, όχι ο σωστός αριθμός tokens
            // Καταχώρηση του λάθους
            System.err.println ("Error ---> Constructor
Course." +
tokens,but received : " + st.countTokens ()
);

            return;
        }

        // Εξαγωγή των tokens και ορισμός των ιδιοτήτων
        this.code = st.nextToken ();
        this.title = st.nextToken ();
        this.description = st.nextToken ();
        this.examDateTime = st.nextToken ();
```

```
        this.theory = st.nextToken ();
        this.note = st.nextToken ();
    }

    // Μέθοδοι ιδιοτήτων SET/GET
    public void setCode (final String Code) {

        this.code = Code;
    }

    public String getCode () {

        return (this.code);
    }

    public void setTitle (final String Title) {

        this.title = Title;
    }

    public String getTitle () {

        return (this.title);
    }

    public void setDescription (final String Description) {

        this.description = Description;
    }

    public String getDescription () {

        return (this.description);
    }

    public void setExamDateTime (final String ExamDateTime) {

        this.examDateTime = ExamDateTime;
    }

    public String getExamDateTime () {

        return (this.examDateTime);
    }

    public void setTheory (final String Theory) {

        this.theory = Theory;
    }

    public String getTheory () {

        return (this.theory);
    }

    public void setNote (final String Note) {

        this.note = Note;
    }

    public String getNote () {
```

```
        return (this.note);
    }

    public Vector getQuestions () {

        /*
         * Κλήση της συνάρτησης Connect () του αντικειμένου
        DatabaseManager
         * για την δημιουργία σύνδεσης με την Βάση Δεδομένων
         */
        if (this.databaseManager.Connect ().booleanValue ()) {

            // Δημιουργία του SQL ερωτήματος
            final String query = "SELECT
        qtitle,qanswer1,qanswer2,qanswer3,qanswer4,qcorrectAnswer " +
            "FROM
        Questions,Courses " +
            "WHERE Questions.cid
        = Courses.cid " +
            "AND Questions.cid =
        '" + this.code + "'" +
            "ORDER BY
        Questions.qid";

            /*
             * Κλήση της συνάρτησης QueryExecute () του
            αντικειμένου DatabaseManager
             * για την εκτέλεση του ερωτήματος - query προς την
            Βάση Δεδομένων
             */
            final Vector vResult =
        this.databaseManager.ExecuteQuery (query);

            /*
             * Κλήση της συνάρτησης ShutDown () του
            αντικειμένου DatabaseManager
             * για τον τερματισμό της σύνδεσης από την Βάση
            Δεδομένων
             */
            this.databaseManager.ShutDown ();

            // Διαδικασία δημιουργίας των αντικειμένων Question
            for (int i=0;i<vResult.size ();i++) {

                // Δημιουργία αντικειμένου Question
                Question question = new Question (((String)
        vResult.elementAt (i)));

                // Προσθήκη του αντικειμένου Question στο
            διάνυσμα
                this.vQuestions.addElement (question);
            }

            return (this.vQuestions);
        }

        /*****
         * Εισαγωγή ενός αντικειμένου Course στην Βάση Δεδομένων *
         * Επιστροφή του πλήθους των γραμμών που ενημερώθηκαν *
        *****/
    }
}
```

```
*****/
public void insertCourseRecord (final String professorCode) {

    /*
     * Κλήση της συνάρτησης Connect () του αντικειμένου
    DatabaseManager
     * για την δημιουργία σύνδεσης με την Βάση Δεδομένων
     */
    if (this.databaseManager.Connect ().booleanValue ()) {

        // Δημιουργία του SQL ερωτήματος

        final String query = "INSERT INTO Courses
        (cid,ctitle,cdescription,cexamdatetime,cnote,pid) " +
        "VALUES (' +
        this.code + "','" + this.title + "','" + this.description + "','" +
        this.examDateTime + "','" + this.note + "','" + professorCode +
        "')";

        /*
         * Κλήση της συνάρτησης ExecuteUpdateQuery () του
        αντικειμένου DatabaseManager
         * για την εκτέλεση του ερωτήματος - query προς την
        Βάση Δεδομένων
         */
        final int updateCount =
        this.databaseManager.ExecuteUpdateQuery (query);

        /*
         * Κλήση της συνάρτησης ShutDown () του
        αντικειμένου DatabaseManager
         * για τον τερματισμό της σύνδεσης από την Βάση
        Δεδομένων
         */
        this.databaseManager.ShutDown ();
    }
}

// Κύριο σημείο εισόδου της εφαρμογής στην κλάση
public static void main (String [] args) {

    final String courseData = "c25,JAVA,Internet
    Programming,2004-11-29 10:00:00,Theory will be available soon.,Good
    Luck!";

    // Δημιουργία αντικειμένου Course
    Course javaCourse = new Course (courseData);

    // Κλήση της συνάρτησης insertCourseRecord ()
    javaCourse.insertCourseRecord ("p125");
}
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Generate Random Numbers

```
// Αρχείο : GenerateRandomNumbers.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των util κλάσεων
import java.util.Random;

// Κλάση GenerateRandomNumbers
public class GenerateRandomNumbers {

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public void GenerateRandomNumbers () {
    }

    /*****
    * Παραγωγή ενός τυχαίου αριθμού μεταξύ του 0 και του αριθμού
    που *
    * καθορίζει ο χρήστης - restrictiveNumber
    *
    * Επιστροφή του παραγόμενου αριθμού
    *
    *****/
    public int generateRandomNumbers (final int restrictiveNumber)
    {

        // Δημιουργία αντικειμένου Random
        Random random = new Random ();

        int number = (int) (random.nextDouble () *
        restrictiveNumber);

        return (number);
    }

    // Κύριο σημείο εισόδου της εφαρμογής στην κλάση
    public static void main (String [] args) {

        // Δημιουργία αντικειμένου GenerateRandomNumbers
        GenerateRandomNumbers grn = new GenerateRandomNumbers ();

        // Περιοριστικός αριθμός
        int restrictiveNumber = 10;

        // Κλήση της συνάρτησης generateRandomNumbers ()
        System.out.println (grn.generateRandomNumbers
        (restrictiveNumber));
    }
}
```

ANTIKEIMENO Examination

```
// Αρχείο : Examination.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή της κλάσης Vector του πακέτου util
import java.util.Vector;

// Κλάση Examination
public class Examination {

    // Δημιουργία των μεταβλητών μέλους (member variable)
    private GenerateRandomNumbers grn = new GenerateRandomNumbers
();
    private final int length = 20;
    final private Question [] questions = new Question [length];
    final private String courseTitleWithDescription;

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public Examination (final Course course) {

        // Διαδικασία απόκτησης τυχαίων αντικειμένων Question για
την διαδικασία της εξέτασης
        final Vector vCourseQuestions = course.getQuestions ();

        for (int i=0;i<this.questions.length;i++) {

            int index = grn.generateRandomNumbers
(vCourseQuestions.size ());

            // Δημιουργία αντικειμένου Question
            this.questions [i] = (Question)
vCourseQuestions.elementAt (index);

            // Αφαίρεση του στοιχείου από το Vector που
βρίσκεται στην θέση index
            vCourseQuestions.removeElementAt (index);
        }

        // Ορισμός της ιδιότητας
        this.courseTitleWithDescription = (course.getTitle () + "
- " + course.getDescription ());
    }

    // Επιστροφή των τυχαίων αντικειμένων Question
    public Question [] getQuestionsForExam () {

        return (this.questions);
    }

    public String getCourseTitleWithDescription () {

        return (this.courseTitleWithDescription);
    }

    // Κύριο σημείο εισόδου της εφαρμογής στην κλάση
    public static void main (String [] args) {
```

```
        final String courseData = "c25,JAVA,Internet  
Programming,2004-11-29 10:00:00,Theory will be available soon.,Good  
Luck!";  
  
        // Δημιουργία αντικειμένου Course  
        final Course course = new Course (courseData);  
  
        // Δημιουργία αντικειμένου Examination  
        Examination exam = new Examination (course);  
  
        final Question [] question = exam.getQuestionsForExam  
();  
  
        for (int i=0;i<question.length;i++)  
            System.out.println (question [i].getTitle ());  
    }  
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Examination Main Menu

```
// ΑΡΧΕΙΟ : ExaminationMainMenu.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του Swing
import javax.swing.*;
import javax.swing.event.*;

// Εισαγωγή των πακέτων του Abstract Window Toolkit (AWT)
import java.awt.*;
import java.awt.event.*;

// Κλάση ExaminationMainMenu
public class ExaminationMainMenu extends JMenuBar {

    // Μενού "File"
    public JMenu menuFile = new JMenu ("File");
        public JMenuItem fileExit = new JMenuItem ("Exit",new
ImageIcon ("Images/btnExit.gif"));

    // Μενού "Web Areas"
    public JMenu menuWebAreas = new JMenu ("Web Areas");
        public JMenuItem WebAreasYahoo = new JMenuItem ("Yahoo");
        public JMenuItem WebAreasGoogle = new JMenuItem
("Google");

    // Μενού "Option"
    public JMenu menuOption = new JMenu ("Option");

    // Μενού "Help"
    public JMenu menuHelp = new JMenu ("Help");
        public JMenuItem HelpAbout = new JMenuItem ("About
Exam",new ImageIcon ("Images/btnHelp.gif"));

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public ExaminationMainMenu () {

        // Μενού "File"

        menuFile.setMnemonic(KeyEvent.VK_F);

        // Αλλαγή γραμματοσειράς
        menuFile.setFont (new Font ("Serif",Font.BOLD,14));

        // Προσθήκη του μενού File
        add (menuFile);

        // Αλλαγή γραμματοσειράς των στοιχείων του μενού File
        fileExit.setFont (new Font ("Serif",Font.BOLD,14));

        // Προσθήκη των στοιχείων του μενού File
        menuFile.add (fileExit);

        // Μενού "Web Areas"

        menuWebAreas.setMnemonic(KeyEvent.VK_W);

        // Αλλαγή γραμματοσειράς
```

```
menuWebAreas.setFont (new Font ("Serif",Font.BOLD,14));

// Προσθήκη του μενού Web Areas
add (menuWebAreas);

// Αλλαγή γραμματοσειράς των στοιχείων του μενού Web
Areas
WebAreasYahoo.setFont (new Font ("Serif",Font.BOLD,14));
WebAreasGoogle.setFont (new Font ("Serif",Font.BOLD,14));

// Προσθήκη των στοιχείων του μενού Web Areas
menuWebAreas.add (WebAreasYahoo);
menuWebAreas.add (WebAreasGoogle);

// Μενού "Option"

menuOption.setMnemonic(KeyEvent.VK_O);

// Αλλαγή γραμματοσειράς
menuOption.setFont (new Font ("Serif",Font.BOLD,14));

// Προσθήκη του μενού Option
add (menuOption);

// Μενού "Help"

menuHelp.setMnemonic(KeyEvent.VK_H);

// Αλλαγή γραμματοσειράς
menuHelp.setFont (new Font ("Serif",Font.BOLD,14));

// Προσθήκη του μενού Help
add (menuHelp);

// Αλλαγή γραμματοσειράς των στοιχείων του μενού Help
HelpAbout.setFont (new Font ("Serif",Font.BOLD,14));

// Προσθήκη των στοιχείων του μενού Help
menuHelp.add (HelpAbout);
}

// Κύριο σημείο εισόδου της εφαρμογής στην κλάση
public static void main (String [] args) {

    // Δημιουργία του αντικειμένου JFrame
    JFrame frame = new JFrame ("Examination Main Menu");

    // Δημιουργία του αντικειμένου ExaminationMainMenu
    ExaminationMainMenu examinationMainMenu = new
ExaminationMainMenu ();

    // Προσθήκη του αντικειμένου ExaminationMainMenu στο
JFrame
frame.setJMenuBar (examinationMainMenu);

// Μετατροπή του μεγέθους του JFrame
frame.setSize (660,450);

// Εμφάνιση του JFrame
```

```
frame.setVisible (true);

/*
 * Τοποθέτηση ενός αισθητήρα για το κλείσιμο της
εφαρμογής
 * σε περίπτωση κλεισίματος του παραθύρου
 */
frame.addWindowListener (
    new WindowAdapter () {
        public void windowClosing (WindowEvent we) {
            System.exit (0);
        }
    }
);

// Προσθήκη του αισθητήρα σε όλα τα στοιχεία των μενού
public void addActionListener (ActionListener al) {

    // Προσθήκη του al σαν αισθητήρα στα στοιχεία του μενού
    "File"
    fileExit.addActionListener (al);

    // Προσθήκη του al σαν αισθητήρα στα στοιχεία του μενού
    "Web Areas"
    WebAreasYahoo.addActionListener (al);
    WebAreasGoogle.addActionListener (al);

    // Προσθήκη του al σαν αισθητήρα στα στοιχεία του μενού
    "Help"
    HelpAbout.addActionListener (al);
}
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Timer Panel

```
// ΑΡΧΕΙΟ : TimerPanel.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του Swing
import javax.swing.*;
import javax.swing.event.*;

// Εισαγωγή των πακέτων του Abstract Window Toolkit (AWT)
import java.awt.*;
import java.awt.event.*;
import java.awt.Font;
import java.awt.Color;

// Κλάση TimerPanel
public class TimerPanel extends JPanel implements Runnable {

    // Δημιουργία των ετικέτων
    JLabel MessageLabel = new JLabel ("Remaning Time
: ",JLabel.RIGHT);
    JLabel TimeLabel = new JLabel ("",JLabel.RIGHT);

    // Δημιουργία μεταβλητών μέλους (member variable)
    protected int examinationTime = 900;
    private int secondTime = 0;
    private int minuteTime = 0;
    private String totalTime = null;

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public TimerPanel () {

        // Δημιουργία αντικειμένου JPanel
        JPanel jp;

        // Αλλαγή γραμματοσειράς στα συστατικά διασύνδεσης χρήστη
        // Ετικετιών
        MessageLabel.setFont (new Font ("Serif",Font.PLAIN,18));
        TimeLabel.setFont (new Font ("Serif",Font.BOLD,16));

        // Αλλαγή χρώματος της ετικέτας QuestionLabel
        MessageLabel.setForeground (Color.red);

        // Προσθήκη κουμπιού πίεσης, ετικέτας και πεδίου κειμένου
στο JPanel
        jp = new JPanel (new FlowLayout (FlowLayout.RIGHT));

        jp.add (MessageLabel);
        jp.add (TimeLabel);
        add (jp);
    }

    // Κύριο σημείο εισόδου της εφαρμογής στην κλάση
    public static void main (String [] args) {

        // Δημιουργία αντικειμένου JFrame
        JFrame frame = new JFrame ("Timer Panel");

        // Δημιουργία αντικειμένου TimerPanel
```

```
TimerPanel timer = new TimerPanel ();

new Thread (timer).start ();

// Προσθήκη του αντικειμένου TimerPanel στο JFrame
frame.getContentPane ().add (timer, BorderLayout.CENTER);

// Αλλαγή μεγέθους του JFrame
frame.setSize (820,80);

// Εμφάνιση του JFrame
frame.setVisible (true);

/* Δημιουργία αισθητήρα παραθύρου για το κλείσιμο της
 * εφαρμογής όταν αποφασιστεί να τελειώσει η χρήση της.
 */
frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
}

// Δήλωση της μεθόδου run Runnable
public void run () {

    try {

        for (int i=this.examinationTime;i>=0;i--) {

            Thread.sleep(1000);

            this.minuteTime = i / 60;

            this.secondTime = i % 60;

            this.totalTime = this.minuteTime + " : " +
this.secondTime;

            TimeLabel.setText (this.totalTime);

        }

    } catch (InterruptedException iex) {

        iex.printStackTrace();

    }

}
```

ΑΝΤΙΚΕΙΜΕΝΟ Question List Panel

```
// ΑΡΧΕΙΟ : QuestionListPanel.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του Swing
import javax.swing.*;
import javax.swing.event.*;

// Εισαγωγή των πακέτων του Abstract Window Toolkit (AWT)
import java.awt.*;
import java.awt.event.*;

// Κλάση QuestionListPanel
public class QuestionListPanel extends JPanel {

    // Δημιουργία αντικειμένου DefaultListModel
    DefaultListModel QuestionsDlm = new DefaultListModel ();

    // Δημιουργία αντικειμένου JList με παράμετρο το αντικείμενο
    DefaultListModel
    JList QuestionsList = new JList (QuestionsDlm);

    // Δημιουργία αντικειμένου CellRenderer
    CellRenderer cellRenderer = new CellRenderer ();

    // Δημιουργία αντικειμένου JScrollPane με παράμετρο το
    αντικείμενο JList
    JScrollPane scrollPane = new JScrollPane (QuestionsList);

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public QuestionListPanel (final Question [] question) {

        // Καθορισμός της κλάσης απεικόνισης κελιών της JList
        QuestionsList.setCellRenderer (cellRenderer);

        // Καθαρισμός των τιμών που περιέχει το αντικείμενο
        DefaultListModel
        QuestionsDlm.clear ();

        // Ανάκτηση του μήκους του πίνακα
        final int length = question.length;

        if (length > 0) {

            // Τοποθέτηση των στοιχείων Question στο
            αντικείμενο DefaultListModel
            for (int i=0;i<length;i++)
                QuestionsDlm.addElement (question [i]);

            // Καθορισμός της πληροφορίας επιλογής : απλή
            επιλογή και επιλογή του πρώτου στοιχείου
            QuestionsList.setSelectionMode
            (ListSelectionMode.SINGLE_SELECTION);
            QuestionsList.setSelectedIndex (0);
        }

        // Τοποθέτηση του επιθυμητού μεγέθους
```

```

        scrollPane.setPreferredSize (new Dimension (350,300));

        // Προσθήκη του Panel κύλισης στο JPanel
        add (scrollPane, BorderLayout.CENTER);
    }

    // Κύριο σημείο εισόδου της εφαρμογής στην κλάση
    public static void main (String [] args) {
    }

    // Κλάση CellRenderer - Λέει στο JList πως να σχεδιάζει κάθε
    κελί
    class CellRenderer extends JLabel implements ListCellRenderer {

        // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
        public CellRenderer () {

            // Δυνατότητα μεταβολής στο χρώμα του φόντου
            setOpaque (true);
        }

        /* Ανασχηματισμός των JLabel κάθε φορά που καλείται αυτή
        * η συνάρτηση. Εδώ είναι το σημείο όπου τοποθετούνται τα
        * χαρακτηριστικά απεικόνισης για κάθε κελί
        */
        public Component getListCellRendererComponent (
            JList list,
            Object value,                // Τιμή που θα
            int index,                    // Δείκτης
            boolean isSelected,          // Ένδειξη
            boolean cellHasFocus        // Ένδειξη ότι έχει
            ) {                            //
            συγκεκριμένο κελί της λίστας

                // Δημιουργία αντικειμένου Question
                Question question = (Question) value;

                /*
                * Τοποθέτηση του σωστού χρώματος φόντου και
                * το στοιχείο είναι επιλεγμένο, και η επιλογή του
                * που θα σχεδιαστεί
                */

                // Έλεγχος για τον αν το στοιχείο έχει επιλεγεί
                if (isSelected) {

                    setBackground (list.getSelectionBackground
                ());

                    setForeground (list.getSelectionForeground
                ());

                    if (question.getSelectedAnswer () == null)
                        setIcon (new ImageIcon
                ("Images/questionSelected.gif"));
                    else

```

```
                setIcon (new ImageIcon
("Images/questionSelectedAnswered.gif"));
            }
            else {
                setBackground (list.getBackground ());
                setForeground (list.getForeground ());

                if (question.getSelectedAnswer () == null)

                    setIcon (new ImageIcon
("Images/questionNormal.gif"));
                else
                    setIcon (new ImageIcon
("Images/questionAnswered.gif"));
            }

            // Καθορισμός του κειμένου του πεδίου στην τιμή που
έχει
            // το τρέχον κελί
            //if (question.getTitle () != null)
                setText (question.getTitle ());

            // Αλλαγή γραμματοσειράς του κειμένου του πεδίου
            setFont (new Font ("Serif",Font.BOLD,16));

        return (this);
    }
}
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Question Multiple Choice Panel

```
// ΑΡΧΕΙΟ : QuestionMultipleChoicePanel.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του Swing
import javax.swing.*;
import javax.swing.event.*;

// Εισαγωγή των πακέτων του Abstract Window Toolkit (AWT)
import java.awt.*;
import java.awt.event.*;
import java.awt.Font.*;
import java.awt.Color;

// Κλάση QuestionMultipleChoicePanel
public class QuestionMultipleChoicePanel extends JPanel {

    // Φόρτιση εικόνας
    // ImageIcon IconbtnAnswer = new ImageIcon
    ("Images/btnAnswer.gif");

    // Δημιουργία των ετικέτων
    JLabel CourseLabel = new JLabel ("",JLabel.CENTER);
    JLabel QuestionLabel = new JLabel ("Question's
Title",JLabel.CENTER);

    // Δημιουργία κουμπίων ραδιοφώνου
    JRadioButton rbAnswer1 = new JRadioButton ("Answer1");
    JRadioButton rbAnswer2 = new JRadioButton ("Answer2");
    JRadioButton rbAnswer3 = new JRadioButton ("Answer3");
    JRadioButton rbAnswer4 = new JRadioButton ("Answer4");
    /*
    * Κρυφό radio button για τον καθαρισμό των radio buttons του
    * bgrpNumbers. Το προσθέτουμε στο bgrpNumbers, αλλά όχι στο
panel
    */
    JRadioButton rbTemp = new JRadioButton ("");

    // Δημιουργία ButtonGroup για την διαχείριση των κουμπίων
ραδιοφώνου
    ButtonGroup bgrpNumbers = new ButtonGroup ();

    // Δημιουργία κουμπιών πίεσης
    JButton btnNext = new JButton ("NEXT QUESTION");
    JButton btnFinish = new JButton ("FINISH");

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public QuestionMultipleChoicePanel (final String
courseTitleWithDescription) {

        // Δημιουργία πλέγματος εμφάνισης 4 X 1
        setLayout (new GridLayout (4,1));

        // Δημιουργία αντικειμένου JPanel
        JPanel jp;

        // Αλλαγή γραμματοσειράς στα συστατικά διασύνδεσης χρήστη
        // Ετικετών
```

```
CourseLabel.setFont (new Font ("Serif",Font.PLAIN,18));
QuestionLabel.setFont (new Font ("Serif",Font.BOLD,16));
// Κουμπιών Ραδιοφώνου
rbAnswer1.setFont (new Font ("Serif",Font.BOLD,14));
rbAnswer2.setFont (new Font ("Serif",Font.BOLD,14));
rbAnswer3.setFont (new Font ("Serif",Font.BOLD,14));
rbAnswer4.setFont (new Font ("Serif",Font.BOLD,14));
// Κουμπιών Πίεσης
btnNext.setFont (new Font ("Serif",Font.BOLD,12));
btnFinish.setFont (new Font ("Serif",Font.BOLD,12));

// Αλλαγή χρώματος της ετικέτας QuestionLabel
QuestionLabel.setForeground (Color.blue);

// Αλλαγή κειμένου ετικέτας
CourseLabel.setText (courseTitleWithDescription);

// Δημιουργία νέου JPanel. Προσθήκη ετικέτας
jp = new JPanel (new FlowLayout ());
jp.add (CourseLabel);
add (jp);

// Προσθήκη των κουμπιών ραδιοφώνου στην ομάδα κουμπιών
bgrpNumbers.add (rbAnswer1);
bgrpNumbers.add (rbAnswer2);
bgrpNumbers.add (rbAnswer3);
bgrpNumbers.add (rbAnswer4);
bgrpNumbers.add (rbTemp);

// Δημιουργία νέου JPanel. Προσθήκη των κουμπιών
ραδιοφώνου στο JPanel
jp = new JPanel (new GridLayout (5,1));
jp.add (QuestionLabel);
jp.add (rbAnswer1);
jp.add (rbAnswer2);
jp.add (rbAnswer3);
jp.add (rbAnswer4);
add (jp);

// Δημιουργία νέου JPanel. Προσθήκη του κουμπιού πίεσης
jp = new JPanel (new FlowLayout (FlowLayout.RIGHT));

jp.add (btnNext);
add (jp);

// Δημιουργία νέου JPanel. Προσθήκη του κουμπιού πίεσης
jp = new JPanel (new FlowLayout ());
jp.add (btnFinish);
add (jp);

// Τοποθέτηση του επιθυμητού μεγέθους
setPreferredSize (new Dimension (400,420));
}

public void questionChanged (final Question question) {

    QuestionLabel.setText (question.getTitle ());

    String [] Answers = question.getAnswers ();

    rbAnswer1.setText (Answers [0]);
```

```
        rbAnswer2.setText (Answers [1]);
        rbAnswer3.setText (Answers [2]);
        rbAnswer4.setText (Answers [3]);
    }

    // Κύριο σημείο εισόδου της εφαρμογής στην κλάση
    public static void main (String [] args) {

        // Δημιουργία αντικειμένου JFrame
        JFrame frame = new JFrame ("Question Multiple Choice
Panel");

        // Δημιουργία αντικειμένου QuestionMultipleChoicePanel
        QuestionMultipleChoicePanel questionMultipleChoicePanel =
new QuestionMultipleChoicePanel ("Course's Title With Description");

        // Προσθήκη του αντικειμένου QuestionMultipleChoicePanel
στο JFrame
        frame.getContentPane ().add
(questionMultipleChoicePanel, BorderLayout.CENTER);

        // Αλλαγή μεγέθους του JFrame
        frame.setSize (420,450);

        // Εμφάνιση του JFrame
        frame.setVisible (true);

        /* Δημιουργία αισθητήρα παραθύρου για το κλείσιμο της
        * εφαρμογής όταν αποφασιστεί να τελειώσει η χρήση της.
        */
        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }

    // Προώθηση του αισθητήρα
    public void addActionListener (ActionListener al) {

        // Προσθήκη του al σαν αισθητήρα για τα αντικείμενα
JRadioButton
        rbAnswer1.addActionListener (al);
        rbAnswer2.addActionListener (al);
        rbAnswer3.addActionListener (al);
        rbAnswer4.addActionListener (al);

        // Προσθήκη του al σαν αισθητήρα για τα αντικείμενα
JButton
        btnNext.addActionListener (al);
        btnFinish.addActionListener (al);
    }
}
```

ΑΝΤΙΚΕΙΜΕΝΟ Examination Dialog

```
// APXEIO : ExaminationDialog.java

// Εισαγωγή των πακέτων που θα χρησιμοποιήσει αυτή η κλάση

// Εισαγωγή των πακέτων του Swing
import javax.swing.*;
import javax.swing.event.*;

// Εισαγωγή των πακέτων του Abstract Window Toolkit (AWT)
import java.awt.*;
import java.awt.event.*;

// Κλάση ExaminationDialog
public class ExaminationDialog extends JDialog implements
ActionListener, ListSelectionListener {

    // Δημιουργία μεταβλητών μέλους (member variable)
    private final Question [] question;
    private String courseTitleWithDescription;
    private String eid;
    private int selectedIndex = 0;

    // Δήλωση των panels
    ExaminationMainMenu examinationMainMenu = new
ExaminationMainMenu ();
    TimerPanel timer = new TimerPanel ();
    QuestionListPanel questionListPanel;
    QuestionMultipleChoicePanel questionMultipleChoicePanel;

    // Δημιουργία pane διαχωρισμού
    JSplitPane splitPane;

    // Συνάρτηση ΔΗΜΙΟΥΡΓΙΑΣ - CONSTRUCTOR
    public ExaminationDialog (final JFrame owner, final Examination
examination, final String eid) {

        // Ορισμός του τίτλου του πλαισίου
        super (owner, "Examination Dialog", true);

        // Δημιουργία αντικειμένου JPanel
        JPanel jp = new JPanel ();

        // Ορισμός των μεταβλητών μέλους
        this.question = examination.getQuestionsForExam ();
        this.courseTitleWithDescription =
examination.getCourseTitleWithDescription ();
        this.eid = eid;

        // Δημιουργία του αντικειμένου QuestionListPanel με τις
ερωτήσεις
        questionListPanel = new QuestionListPanel
(this.question);

        // Καθορισμός της πληροφορίας επιλογής : Επιλογή του
στοιχείου της 1ης γραμμής
        questionListPanel.QuestionsList.setSelectedIndex
(this.selectedIndex);
    }
}
```

```
// Δημιουργία του αντικειμένου
QuestionMultipleChoicePanel με τον τίτλο και την περιγραφή του
μαθήματος
    questionMultipleChoicePanel = new
QuestionMultipleChoicePanel (this.courseTitleWithDescription);

    // Κλήση της συνάρτησης questionChanged του αντικειμένου
QuestionMultipleChoicePanel
    questionMultipleChoicePanel.questionChanged (this.question
[this.selectedIndex]);

    // Δημιουργία του panel διαχωρισμού με τα δυο panels
splitPane = new JSplitPane
(JSplitPane.HORIZONTAL_SPLIT,questionListPanel,questionMultipleChoice
Panel);

    splitPane.setOneTouchExpandable (true);
    splitPane.setDividerLocation (350);

    // Καθορισμός του ελάχιστου μεγέθους για τα δυο συστατικά
στο panel διαχωρισμού
    questionListPanel.setMinimumSize (new Dimension
(350,300));
    questionMultipleChoicePanel.setMinimumSize (new Dimension
(400,420));

    /*
    * Μετατροπή της κλάσης σε αισθητήρα επιλογής λίστας ώστε
να είναι
    * γνωστό το πότε ο χρήστης επιλέγει κάτι
    */
    questionListPanel.QuestionsList.addListSelectionListener
(this);

    // Προσθήκη του αντικειμένου ExaminationMainMenu στο
JFrame
    setJMenuBar (examinationMainMenu);

    // Προσθήκη του ExaminationPanel σαν αισθητήρα του
ExaminationMainMenu
    examinationMainMenu.addActionListener (this);

    // Δημιουργία νέου JPanel. Προσθήκη του TimerPanel
JPanel TimerTempPanel = new JPanel (new FlowLayout ());
    TimerTempPanel.add (timer);
    // Προσθήκη του TimerTempPanel στο panel jp
    jp.add (TimerTempPanel);
    new Thread (timer).start ();

    // Δημιουργία νέου JPanel. Προσθήκη του panel διαχωρισμού
JPanel SplitPanel = new JPanel (new FlowLayout ());
    SplitPanel.add (splitPane);
    // Προσθήκη του SplitPanel στο panel jp
    jp.add (SplitPanel, BorderLayout.CENTER);

    // Προσθήκη του ExaminationPanel σαν αισθητήρα του
QuestionMultipleChoicePanel
    questionMultipleChoicePanel.addActionListener (this);

    // Προσθήκη του panel jp στο παρόν panel
    getContentPane ().add (jp);
```

```
// Αλλαγή μεγέθους του JFrame
setSize (820,560);

// Κεντράρισμα του παραθύρου
Dimension screenSize = Toolkit.getDefaultToolkit
().getScreenSize ();
Dimension windowSize = getSize ();

setBounds ( (screenSize.width - windowSize.width) / 2,
            (screenSize.height - windowSize.height)
/ 2,
            windowSize.width,windowSize.height
);
}
//-----
private void questionChanged (final int index) {

    this.selectedIndex = index;

    // Έλεγχος για το αν ο δείκτης selectedIndex είναι στα
    όρια του πίνακα
    if (this.selectedIndex > 19)
        this.selectedIndex = 0;

    // Δημιουργία αντικειμένου Question
    Question question = this.question [this.selectedIndex];

    // Επιλογή του στοιχείου "Question" που βρίσκεται στην
    γραμμή index
    questionListPanel.QuestionsList.setSelectedIndex
(this.selectedIndex);

    // Επιβεβαίωση ότι το στοιχείο "Question" είναι ορατό
    questionListPanel.QuestionsList.ensureIndexIsVisible
(this.selectedIndex);

    // Αποστολή του στοιχείου "Question" στο panel
    QuestionMultipleChoicePanel
questionMultipleChoicePanel.questionChanged (question);

    if (question.getSelectedAnswer () == null)
        /*
        * Επιλέγουμε το "κρυφό" radio button για να
        δώσουμε την
        * ψευδαίσθηση καθαρισμού του ButtonGroup.
        */
        questionMultipleChoicePanel.rbTemp.setSelected
(true);
    else
        // Κλήση της συνάρτησης getAnswer ()
        this.getAnswer ();
}

private void getAnswer () {

    final String answer = this.question
[this.selectedIndex].getSelectedAnswer ();

    if (answer.equals ("Answer1"))
```

```

        questionMultipleChoicePanel.rbAnswer1.setSelected
(true);
        else {
            if (answer.equals ("Answer2"))

questionMultipleChoicePanel.rbAnswer2.setSelected (true);
            else {
                if (answer.equals ("Answer3"))

questionMultipleChoicePanel.rbAnswer3.setSelected (true);
                else

questionMultipleChoicePanel.rbAnswer4.setSelected (true);
            }
        }
    }

private void endOfExamination () {

    int response = JOptionPane.showConfirmDialog (null,
                                                "ARE YOU SURE YOU WANT TO
END THIS EXAMINATION ?", "End ?",
                                                JOptionPane.YES_NO_OPTION,

JOptionPane.QUESTION_MESSAGE
        );

    // Ο χρήστης τερμάτισε την εφαρμογή
    if (response == 0) {

        // Απόκρυψη του παραθύρου
        setVisible (false);

        // Δημιουργία αντικειμένου ResultTableDialog
        ResultTableDialog rtd = new ResultTableDialog (new
JFrame (),this.courseTitleWithDescription,this.question,this.eid);

        // Εμφάνιση του παραθύρου
        rtd.show ();
    }
}
//-----
// Κύριο σημείο εισόδου της εφαρμογής στην κλάση
public static void main (String [] args) {
}

// Δήλωση της μεθόδου actionPerformed ActionListener
public void actionPerformed (ActionEvent ae) {

    Object source = ae.getSource ();

    // JRadioButton
    if ( (source == questionMultipleChoicePanel.rbAnswer1) ||
        (source == questionMultipleChoicePanel.rbAnswer2)
||
        (source == questionMultipleChoicePanel.rbAnswer3)
||
        (source == questionMultipleChoicePanel.rbAnswer4)
    ) {

```

```
String answer = "";

    if (source ==
questionMultipleChoicePanel.rbAnswer1)
        answer = "Answer1";
    else {
        if (source ==
questionMultipleChoicePanel.rbAnswer2)
            answer = "Answer2";
        else {
            if (source ==
questionMultipleChoicePanel.rbAnswer3)
                answer = "Answer3";
            else
                answer = "Answer4";

        }
    }

    this.question
[this.selectedIndex].setSelectedAnswer (answer);

    // Ανασχηματισμός της λίστας
    questionListPanel.QuestionsList.repaint ();
}

// JButton
if ( (source == questionMultipleChoicePanel.btnNext) ||
    (source == questionMultipleChoicePanel.btnFinish)
    ) {
    if (source ==
questionMultipleChoicePanel.btnNext) {

        // Αύξηση του δείκτη κατά ένα
        this.selectedIndex++;

        // Κλήση της συνάρτησης questionChanged
        questionChanged (this.selectedIndex);

    }
    else
        // Κλήση της συνάρτησης
        this.endOfExamination ();
}

// JMenuItem
if (source == examinationMainMenu.fileExit)
    // Κλήση της συνάρτησης endOfExamination ()
    this.endOfExamination ();
else {

    // Δημιουργία αντικειμένου InternetManager
    InternetManager internetManager;

    if (source == examinationMainMenu.WebAreasYahoo) {
```

```
        internetManager = new InternetManager
("http://www.yahoo.com");

        // Κλήση της συνάρτησης Connect () του
αντικειμένου InternetManager
        internetManager.Connect ();
    }
    else {
        if (source ==
examinationMainMenu.WebAreasGoogle) {

            internetManager = new InternetManager
("http://www.google.com");

            // Κλήση της συνάρτησης Connect () του
αντικειμένου InternetManager
            internetManager.Connect ();
        }
        else {

            if (source ==
examinationMainMenu.HelpAbout)
                System.out.println ("Press
Help");
        }
    }
}

// Δήλωση της μεθόδου valueChanged ListSelectionEvent
public void valueChanged (ListSelectionEvent lse) {

    // Αποφυγή των συμβάντων ρύθμισης
    if (lse.getValueIsAdjusting ())
        return;

    // Ανάκτηση της λίστας από το συμβάν

    JList list = (JList) lse.getSource ();

    int index = list.getSelectedIndex ();

    // Κλήση της συνάρτησης questionChanged ()
    this.questionChanged (index);
}
}
```