

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ



ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: << ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΛΕΞΙΚΟΥ ΤΗΣ ΝΕΟΕΛΛΗΝΙΚΗΣ ΓΛΩΣΣΑΣ >>

E-Dictionary

Εισηγητής: Ιωάννης Παχουλάκης

Σπουδαστής: Μιχάλης Καπαρουνάκης

ΑΜ: 1474

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

Summary

The topic of this project is the design and implementation of a digital dictionary and seeks the highest and best use of the possibilities offered by technology to search and display information quickly and easily. The dictionary includes apart from the vocabulary, the Main Names section and the Greek-English Acronyms section, designed to provide concise additional information. The dictionary also gives many possibilities such as adding entries, editing, deleting and storing data to the database scalability.

The dictionary was created in a BlueJ development environment using Java programming language. The basic form of the dictionary was designed with the help of JFormDesigner avoiding the low-level programming and creating a user-friendly and simple interface. As a result, the dictionary user will be able to find the word he is looking up and read its information in detail.

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

Σύνοψη

Το θεματικό αντικείμενο της παρούσας πτυχιακής είναι η σχεδίαση και η υλοποίηση ενός ψηφιακού λεξικού και έχει ως στόχο τη μέγιστη και βέλτιστη εκμετάλλευση των δυνατοτήτων που παρέχει η τεχνολογία για την αναζήτηση και εμφάνιση των πληροφοριών εύκολα και γρήγορα. Στο λεξικό περιλαμβάνονται εκτός από το λεξικό της γλώσσας, το τμήμα των κύριων ονομάτων και το τμήμα με ελληνικό και αγγλικό αρκτικόλεξο που σαν σκοπό έχουν να παράσχουν επιπλέον επιγραμματικές πληροφορίες. Το λεξικό επιπλέον δίνει πολλές δυνατότητες όπως προσθήκη λημμάτων, επεξεργασία, διαγραφή και αποθήκευση δεδομένων έτσι ώστε να έχει επεκτασιμότητα η βάση δεδομένων.

Το λεξικό δημιουργήθηκε σε περιβάλλον ανάπτυξης εφαρμογών blue-j με χρήση της γλώσσας Java. Η βασική φόρμα του λεξικού σχεδιάστηκε με τη βοήθεια του JFormDesigner αποφεύγοντας έτσι τον χαμηλού επιπέδου προγραμματισμό και δημιουργώντας μια πολύ εύχρηστη και απλή διεπαφή. Σαν αποτέλεσμα ο χρήστης που θα χρησιμοποιήσει το λεξικό, θα μπορέσει με ευκολία να βρει τη λέξη που αναζητεί και να διαβάσει λεπτομερώς τα στοιχεία της.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Abstract.....	3
Σύνοψη	4
Πίνακας περιεχομένων	5
Πίνακας εικόνων.....	7
Πίνακας πινάκων	7
1. Εισαγωγή.....	8
2. Τεχνολογίες υλοποίησης.....	9
2.1 Blue-j.....	9
2.2 JFormDesigner	9
2.3 Γλώσσα προγραμματισμού java.....	9
2.3.1 Χαρακτηριστικά της java	9
2.3.2 Το περιβάλλον της γλώσσας	10
2.3.3 Ο συλλέκτης σκουπιδιών (garbage collector)	11
2.3.4 Κλάσεις και αντικείμενα	11
2.3.5 Αντικειμενοστραφής προγραμματισμός.....	12
2.3.6 Γεγονότα.....	13
2.3.7 Windows στη java	14
2.4 Βάση δεδομένων	16
3. Ηλεκτρονικό λεξικό e-Dictionary.....	18
3.1 Περιγραφή της βάσης δεδομένων	18
3.2 Ανάλυση απαιτήσεων ηλεκτρονικού λεξικού.....	18
3.3 Σχεδιάζοντας τη διεπαφή.....	19
3.3.1 Οι 8 χρυσοί κανόνες του Shneiderman για τη σχεδίαση διεπιφανειών.....	19
3.3.2 Σχεδίαση της βασικής φόρμας	10
3.3.3 Τα βασικά συστατικά της διεπιφάνειας.....	20
3.4 Προγραμματίζοντας με java	21
3.4.1 Πως φορτώνονται τα δεδομένα.....	21
3.4.2 Προσθήκη δεδομένων	22
3.4.3 Επικοινωνία με το χρήστη.....	23
3.5 Περιγραφή του λεξικού από την πλευρά του χρήστη	24
3.5.1 Δυνατότητες λεξικού.....	26
3.5.2 Επιπλέον υλικό.....	28
3.5.3 System tray.....	29
3.5.4 Χειρισμός αρκτικόλεξου	30
3.5.5 Οδηγίες εκτέλεσης	32

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

4. Αποτελέσματα.....	33
4.1 Συμπεράσματα	33
4.2 Επεκτάσεις	33
5. Βιβλιογραφία - Πηγές.....	34
Παράρτημα	35

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1. μεταγλωττιστής παραδοσιακής γλώσσας.....	10
Εικόνα 2. μεταγλωττιστής Java	11
Εικόνα 3. διάγραμμα κλάσεων-υποκλάσεων	13
Εικόνα 4. ιεραρχική δομή των κυριότερων κλάσεων για δημιουργία windows	14
Εικόνα 5. ιεραρχική δομή μερικών υποκλάσεων της component	15
Εικόνα 6. συστατικά JFrame window	16
Εικόνα 7. περιβάλλον εργασίας JFormDesigner.....	20
Εικόνα 8. βασικά συστατικά διεπαφής.....	21
Εικόνα 9. προσθήκη δεδομένων στο λεξικό	23
Εικόνα 10. πλαίσιο διαλόγου με χρήστη.....	24
Εικόνα 11. πραγματική μορφή του λεξικού κατά την εκκίνηση	25
Εικόνα 12. εμφάνιση σημασιολογικών στοιχείων ενός λήμματος	26
Εικόνα 13. αποθήκευση δεδομένων στη βάση	27
Εικόνα 14. διαγραφή δεδομένων	27
Εικόνα 15. παράθυρο γραμματικής.....	28
Εικόνα 16. εικονίδιο λεξικού στο παρασκήνιο.....	29
Εικόνα 17. πόροι κατά την εκκίνηση	30
Εικόνα 18. κατανάλωση πόρων στο παρασκήνιο.....	30
Εικόνα 19. πλαίσιο ελέγχου για αγγλικό αρκτικόλεξο.....	31
Εικόνα 20. πλαίσιο ελέγχου για ελληνικό αρκτικόλεξο	31

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

πίνακας 1. γεγονότα αλληλεπίδρασης.....	13-14
---	-------

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

1. Εισαγωγή

Το αντικείμενο της πτυχιακής εργασίας είναι ο σχεδιασμός μιας εύχρηστης και ελκυστικής εφαρμογής, η οποία θα παρέχει σε κάθε χρήστη πολύτιμες πληροφορίες (σημασιολογικές, συντακτικές κτλ.) του λεξιλογίου της νεοελληνικής γλώσσας. Επίσης θα περιλαμβάνει σε ξεχωριστή κατηγορία κύρια ονόματα καθώς και ένα αρκτικόλεξο.

Σκοπός αυτής της πτυχιακής ήταν η σχεδίαση και δημιουργία ενός ψηφιακού λεξικού για το ευρύτερο κοινό. Είναι ένα χρήσιμο βοήθημα για όλους όσους χρησιμοποιούν Η/Υ, από τον μαθητή του σχολείου μέχρι τον επιστήμονα. Η πληροφόρηση του χρήστη γίνεται μ' έναν τρόπο απλό και κατανοητό, για το ακριβές σημασιολογικό περιεχόμενο μιας λέξης, για την προέλευση της λέξης, για τη σωστή χρήση της κλπ.

Η εφαρμογή δημιουργήθηκε σε περιβάλλον ανάπτυξης εφαρμογών blue - j με χρήση της γλώσσας java. Σαν επιπλέον βοήθημα χρησιμοποιήθηκε το JFormDesigner, ένα πρόγραμμα στο οποίο μπορούμε να δημιουργήσουμε εύκολα ένα γραφικό περιβάλλον επικοινωνίας (GUI). Για την κατασκευή της επίσης, χρησιμοποιήθηκε βάση δεδομένων της Microsoft Access η οποία επιτρέπει την διαχείριση των διαφόρων κατηγοριών και των λημμάτων του λεξικού.

Μπορούμε να αναφερθούμε γενικά στην λειτουργία της εφαρμογής λέγοντας ότι είναι μία stand-alone εφαρμογή, της οποίας μπορεί να κάνει χρήση οποιοσδήποτε. Ο χρήστης πέρα από την πρόσβαση που έχει στο κυρίως λεξικό το οποίο περιλαμβάνει πάνω από 20.000 λήμματα, έχει στη διάθεσή του δυο επιπλέον κατηγορίες: τα κύρια ονόματα στην οποία περιλαμβάνονται κυρίως ονόματα προσώπων, υπαρκτών ή μυθικών, λαών και φυλών κτλ. και τα αρκτικόλεξα ή κοινώς ακρωνύμια (ελληνικό και αγγλικό). Επιπλέον για να έχει επεκτασιμότητα η βάση δεδομένων δίνεται η δυνατότητα προσθήκης πληροφοριών σε αυτήν, επεξεργασίας και αποθήκευσης.

Θα δούμε πιο λεπτομερώς σε επόμενο κεφάλαιο πως και ποιες λειτουργίες ακριβώς μπορούμε να εκτελέσουμε με το ψηφιακό λεξικό. Πρώτα θα γίνει μια παρουσίαση της γλώσσας java και στη συνέχεια μια αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την σχεδίαση και υλοποίηση του λεξικού.

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

2. Τεχνολογίες υλοποίησης

2.1 Blue-j

Το blue-j είναι ένα περιβάλλον ανάπτυξης java προγραμμάτων που σχεδιάστηκε ειδικά για να υποστηρίξει την εκπαίδευση νέων προγραμματιστών στην Java. Σχεδιάστηκε και υλοποιήθηκε απ' την ομάδα Blue-j του πανεπιστημίου *Monash* της Μελβούρνης στην Αυστραλία. Περισσότερες πληροφορίες στη διεύθυνση <http://bluej.monash.edu>.

2.2 JFormDesigner

Το πρόγραμμα αυτό δίνει τη δυνατότητα να σχεδιάζουμε πολύ εύκολα ένα γραφικό περιβάλλον επικοινωνίας (GUI).

Αυτό που μας ενδιέφερε πολύ ήταν η βασική φόρμα του λεξικού να είχε μια ομοιομορφία όσο αναφορά στην αρχιτεκτονική που θα παρουσιάζονται τα δεδομένα. Για το λόγο αυτό η ακρίβεια στη σχεδίαση της φόρμας θα ήταν ένα πολύ δύσκολο κομμάτι και χρονοβόρο αν ο κώδικας γραφόταν στο χέρι, δεδομένου και της πολυπλοκότητας που κρύβουν οι διαχειριστές διάταξης της java. Το JFormDesigner δίνει τη λύση πάνω σε αυτό το πρόβλημα πολύ εύκολα, σχεδιάζοντας υψηλής ποιότητας GUI έχοντας ως γνώση τις βασικές αρχές των διαχειριστών διάταξης της java.

2.3 Γλώσσα προγραμματισμού Java

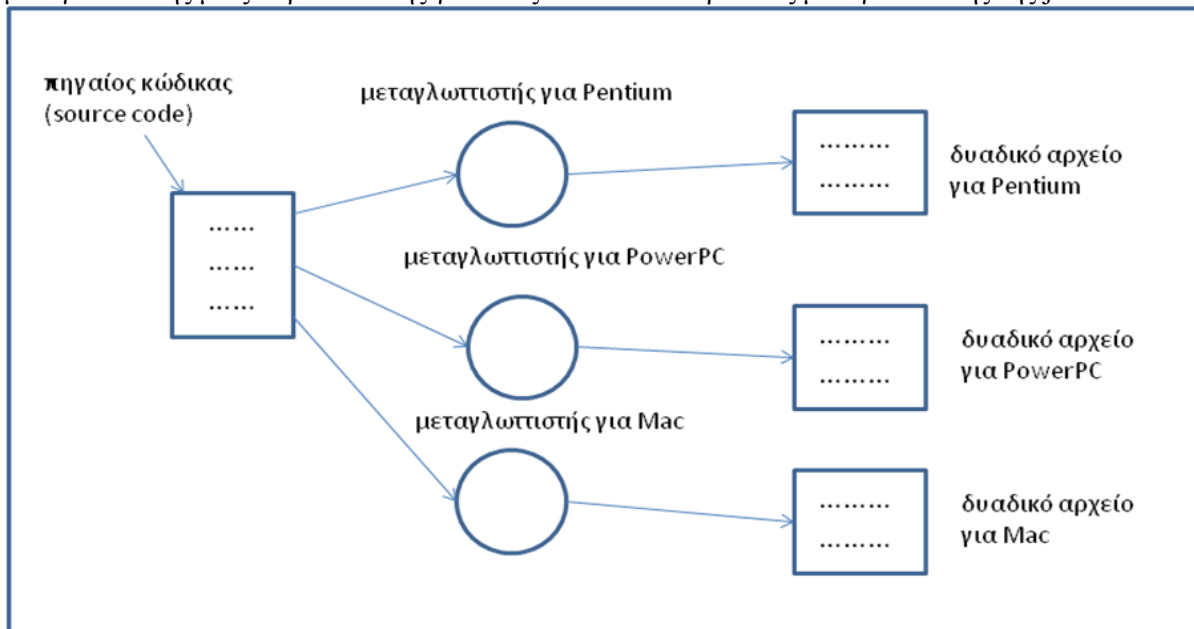
2.3.1 Χαρακτηριστικά της Java

Η ραγδαία εξάπλωση του Internet και του World-Wide Web δημιούργησαν την ανάγκη νέων τρόπων ανάπτυξης και διανομής του λογισμικού. Οι απαιτήσεις αυτές οδήγησαν στην δημιουργία της γλώσσας προγραμματισμού Java, από την εταιρία Sun Microsystems M. Η Java σχεδιάστηκε με σκοπό την ανάπτυξη εφαρμογών που θα τρέχουν σε ετερογενή δικτυακά περιβάλλοντα. Η Java έχει τα ακόλουθα χαρακτηριστικά:

- Αντικειμενοστραφής (ομοιότητες εντολών με τη C++).
- Δημιουργία ανεξάρτητων εφαρμογών και applets (applet = προγράμματα που περιλαμβάνονται σε HTML σελίδες και εκτελούνται από τον Web Browser).
- Είναι Interpreted γλώσσα. Αυτό σημαίνει ότι ο java compiler δεν παράγει εκτελέσιμο κώδικα αλλά μια μορφή ψευδοκώδικα (byte-code) το οποίο από μόνο του δεν τρέχει σε καμία μηχανή. Προκειμένου λοιπόν να εκτελεστεί απαιτείται η χρήση ενός interpreter (=διερμηνέα) για να μετατρέψει το byte-code σε πραγματικό εκτελέσιμο κώδικα. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα στα java byte-codes να μπορούν να τρέξουν σε οποιοδήποτε μηχάνημα, κάτω από οποιοδήποτε λειτουργικό, αρκεί να έχει εγκατασταθεί ένας java interpreter. Επίσης ένα άλλο χαρακτηριστικό του java byte-code είναι το μικρό του μέγεθος. Αυτό το κάνει ιδανικό για μετάδοση μέσω του δικτύου.
- Κατανεμημένη (distributed). Δηλαδή ένα πρόγραμμα σε Java είναι δυνατό να το φέρουμε από το δίκτυο και να το τρέξουμε. Επίσης είναι δυνατό διαφορετικά κομμάτια του προγράμματος να έρθουν από διαφορετικά sites.
- Ασφαλής (secure). Στο δίκτυο όμως ελλοχεύουν πολλοί κίνδυνοι για τον χρήστη-παραλήπτη μιας δικτυακής εφαρμογής, γι' αυτό η Java έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο applet γραμμένο για τέτοιο σκοπό.
- Είναι multithreaded. Η Java υποστηρίζει εγγενώς την χρήση πολλών threads. Προκειμένου να το επιτύχει αυτό σε συστήματα με έναν επεξεργαστή, το Java runtime system (interpreter) υλοποιεί ένα δικό χρονοδρομολογητή (scheduler), ενώ σε συστήματα που υποστηρίζουν πολυεπεξεργασία η δημιουργία των threads ανατίθεται στο λειτουργικό σύστημα. Φυσικά όλα αυτά είναι αόρατα τόσο στον προγραμματιστή όσο και στον χρήστη.
- Υποστηρίζει multimedia εφαρμογές. Με αυτό εννοούμε ότι η Java παρέχει ευκολίες στη δημιουργία multimedia εφαρμογών. Αυτό επιτυγχάνεται τόσο με την ευελιξία της σαν γλώσσα όσο και με τις πλούσιες και συνεχώς εμπλουτιζόμενες βιβλιοθήκες της.

2.3.2 Το περιβάλλον της γλώσσας

Όπως αναφέραμε στα χαρακτηριστικά, η java μπορεί να δημιουργήσει προγράμματα, τα οποία τρέχουν σε μια ποικιλία υπολογιστών και λειτουργικών συστημάτων. Αυτό το καταλαβαίνουμε καλύτερα αν παρατηρήσουμε τα παρακάτω σχήματα, όπου στο πρώτο φαίνεται πως λειτουργεί ο μεταγλωττιστής μιας παραδοσιακής γλώσσας και στο δεύτερο ένας μεταγλωττιστής της java.

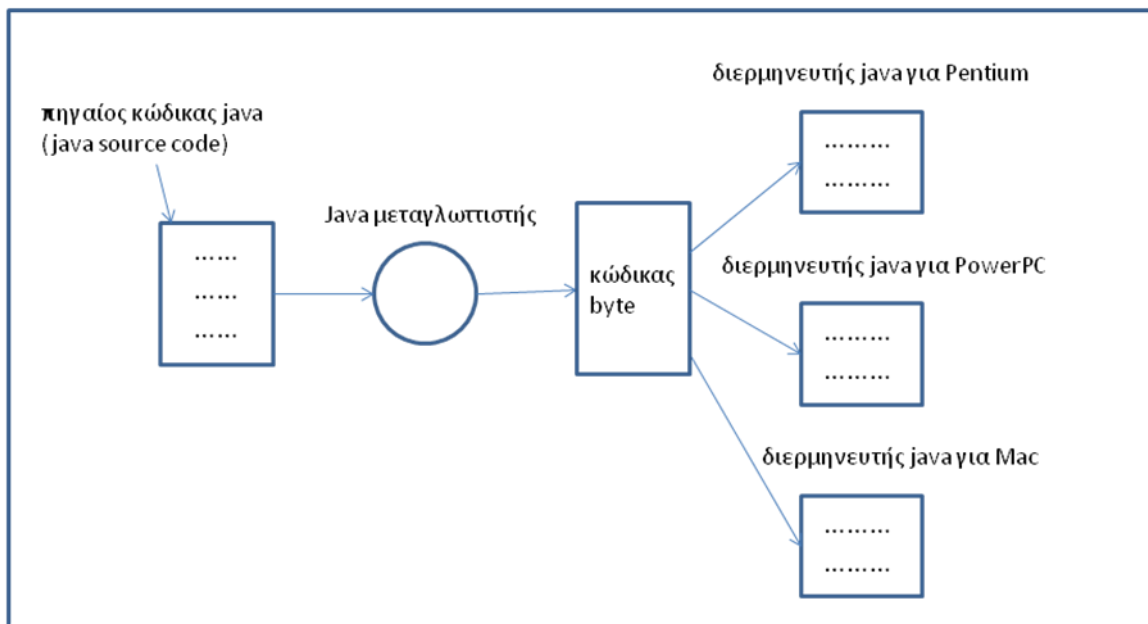


Εικόνα 2. μεταγλωττιστής παραδοσιακής γλώσσας

Σε μια παραδοσιακή γλώσσα (Pascal, C, κτλ.), ο μεταγλωττιστής μετατρέπει το πρόγραμμα το οποίο έχουμε γράψει (source code) σε δυαδικό αρχείο (binary file), δηλαδή σε κώδικα γλώσσας μηχανής κατάλληλο για το συγκεκριμένο επεξεργαστή.

Τα πράγματα είναι διαφορετικά στην περίπτωση της java. Το περιβάλλον ανάπτυξης περιλαμβάνει δυο μέρη: α)το μεταγλωττιστή, ο οποίος αντί για δυαδικό αρχείο δημιουργεί ένα είδος κώδικα που είναι ανεξάρτητος από πλατφόρμα και λέγεται κώδικας byte, και β)το διερμηνευτή κώδικα byte (προσαρμοσμένο σε κάθε πλατφόρμα), ο οποίος ονομάζεται και *Εικονική Μηχανή Java* (Java Virtual Machine, JVM) ή *διερμηνευτής java χρόνου εκτέλεσης* (Java Runtime Interpreter). Ο διερμηνευτής κώδικα byte διαβάζει τον κώδικα byte και εκτελεί τις κατάλληλες εντολές γλώσσας μηχανής, που αντιστοιχούν στο συγκεκριμένο επεξεργαστή.

Τα πιο συνηθισμένα προγράμματα πλοήγησης στο internet εμπεριέχουν το διερμηνευτή κώδικα byte (Java Virtual Machine). Όταν λοιπόν φορτώσουν μια σελίδα από το internet, η οποία περιέχει οδηγίες για την εκτέλεση ενός προγράμματος java, αυτόματα κατεβάζουν τον κώδικα της java από την περιοχή της σελίδας και τρέχουν την JVM πάνω στον κώδικα αυτό. Έτσι επιτυγχάνεται η παρουσίαση κινούμενων εικόνων, μουσικής και πολυμέσων, φορμών εισαγωγής δεδομένων κτλ, που είναι γραμμένα σε java περιέχονται στη σελίδα.



Εικόνα 2. μεταγλωττιστής java

2.3.3 Ο συλλέκτης σκουπιδιών (Garbage collector)

Ακόμα μία ιδέα που βρίσκεται πίσω από τη JAVA είναι η ύπαρξη του συλλέκτη σκουπιδιών (garbage collector). Συλλογή σκουπιδιών είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Υπεύθυνη για αυτό είναι και πάλι η εικονική μηχανή η οποία μόλις «καταλάβει» ότι ο σωρός (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στο σωρό σε αντίθεση με τη C++ όπου αποθηκεύονται κυρίως στη στοίβα - stack) κοντεύει να γεμίσει ενεργοποιεί το συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής δε χρειάζεται να ανησυχεί για το πότε και αν θα ελευθερώσει ένα συγκεκριμένο τμήμα της μνήμης, ούτε και για δείκτες (pointers) που αναφέρονται σε άδειο χώρο μνήμης. Αυτό είναι ιδιαίτερα σημαντικό αν σκεφτούμε ότι ένα μεγάλο ποσοστό κατάρρευσης των προγραμμάτων οφείλονται σε λανθασμένο χειρισμό της μνήμης.

2.3.4 Κλάσεις και αντικείμενα

Θα εξετάσουμε τώρα την “καρδιά” της java, δηλαδή τις κλάσεις (classes) και τα αντικείμενα (objects) που δημιουργούνται από αυτές. Οι έννοιες “κλάση” και “αντικείμενο” είναι βασικές στον αντικειμενοστραφή προγραμματισμό (object oriented programming).

Η κλάση είναι μια γενική φόρμα για την κατασκευή αντικειμένων που έχουν παρόμοια χαρακτηριστικά. Θα μπορούσαμε να πούμε δηλαδή ότι είναι μια “συνταγή” δημιουργίας αντικειμένων.

Αφού ορίσουμε μια κλάση, στη συνέχεια μπορούμε να χρησιμοποιήσουμε τον ορισμό της (τη “συνταγή”) για να δημιουργήσουμε αντικείμενα. Κάθε αντικείμενο λέγεται και στιγμιότυπο (instance) της κλάσης.

Παίρνοντας το πρώτο παράδειγμα από τη ζωή μας, θα μπορούσαμε να θεωρήσουμε ως κλάση την έννοια “δέντρο” και ως αντικείμενα της κλάσης τη “λεμονιά”, τη “μηλιά”, το “έλατο” κτλ. Όλα τα αντικείμενα έχουν τα γενικά χαρακτηριστικά της κλάσης “δέντρο” (ρίζα, κορμό, κλαδιά, φύλλα), αλλά το καθένα έχει και κάποιες ιδιαίτερες ιδιότητες, όπως το είδος του καρπού, το σχήμα, το αν είναι φυλλοβόλο ή όχι, κτλ. Στην εφαρμογή, με την οποία θα ασχοληθούμε στα επόμενα κεφάλαια, χρειάστηκε να δημιουργήσουμε κουμπιά (buttons). Στην java υπάρχει η κλάση **Button**, η οποία

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

καθορίζει τα γενικά χαρακτηριστικά που πρέπει να έχει ένα κουμπί, δηλαδή το μέγεθός του, την ετικέτα του, το πώς θα ανταποκρίνεται (απλό ή διπλό πάτημα) κτλ. Με βάση την κλάση **Button** μπορούμε να δημιουργήσουμε αντικείμενα (συγκεκριμένα κουμπιά), τα οποία να έχουν όλα τα χαρακτηριστικά και τις ιδιότητες που περιέχονται στη **Button**, αλλά επιπλέον το καθένα από αυτά να μπορεί να έχει και τα δικά του ιδιαίτερα χαρακτηριστικά και ιδιότητες.

Στην java υπάρχουν έτοιμες κλάσεις, αλλά μπορεί και ο ίδιος ο χρήστης να δημιουργήσει τις δικές του. Δυο είναι τα στοιχεία που συμπεριλαμβάνονται στον ορισμό μιας κλάσης: Οι *μεταβλητές* (variables), οι οποίες διαφοροποιούν το ένα αντικείμενο της κλάσης από το άλλο και οι *μέθοδοι* (methods), οι οποίες καθορίζουν τι μπορούμε να κάνουμε στα (ή με τα) αντικείμενα μιας κλάσης. Η μέθοδος είναι μια ομάδα από δηλώσεις και εντολές, η οποία εκτελεί μια εξειδικευμένη λειτουργία του αντικειμένου.

2.3.5 Αντικειμενοστραφής προγραμματισμός

Η java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού. Σ' αυτό το είδος προγραμματισμού (Object oriented programming, OOP) μπορούμε να θεωρήσουμε τα στοιχεία του προγράμματος ως αντικείμενα. Σύμφωνα μ' αυτή τη θεώρηση, μια κλάση παρουσιάζει μια συγκεκριμένη εικόνα της "προς τα έξω". Αυτή η εικόνα εμφανίζει τις λειτουργίες (μεθόδους) που επιτελεί η κλάση καθώς και τα δεδομένα που χρησιμοποιεί. Οτιδήποτε άλλο συμβαίνει μέσα στην κλάση, όπως π.χ. οι λεπτομερείς ενέργειες που γίνονται μέσα στις μεθόδους, παραμένει κρυφό. Η όλη ιδέα μοιάζει με την ιδέα του αυτοκινήτου: Για να οδηγήσει κάποιος ένα αυτοκίνητο, δε χρειάζεται να ξέρει τη δομή και τη λειτουργία του κινητήρα ή το τι συμβαίνει μέσα στο κιβώτιο ταχυτήτων. Κάνει μερικές λειτουργικές ενέργειες, τις οποίες του παρέχουν τα εξαρτήματα του αυτοκινήτου (τιμόνι, γκάτζι, φρένο, ταχύτητες κτλ.) και δε χρειάζεται να ξέρει πώς λειτουργούν τα εξαρτήματα αυτά. Τη βασική ιδέα του αντικειμενοστραφούς προγραμματισμού υλοποιούν τρεις αρχές: η *ενθυλάκωση* (encapsulation), η *κληρονομικότητα* (inheritance) και ο *πολυμορφισμός* (polymorphism).

-Η *ενθυλάκωση*, παρουσιάζει την εξωτερική όψη ενός αντικειμένου και αποκρύπτει την πολύπλοκη λογική που υπάρχει μέσα του. Αυτό επιτρέπει στον προγραμματιστή να τροποποιεί ή να βελτιώνει τους αλγόριθμους οι οποίοι υπάρχουν σε μια κλάση, χωρίς να ανησυχεί μήπως προκαλέσει απρόβλεπτα αποτελέσματα. Με άλλα λόγια, τη στιγμή που σε μια κλάση τα δεδομένα και οι χρησιμοποιούμενες μέθοδοι δεν αλλάζουν, κάθε αλλαγή στον τρόπο υλοποίησής τους παραμένει διαφανής για το χρήστη. Τα δεδομένα και οι μέθοδοι μιας κλάσης μπορεί να δηλωθούν ως **public**, ως **private**, ως **protected**, ή χωρίς κανέναν από τους προηγούμενους τρεις *προσδιοριστές προσπέλασης*.

Οι μεταβλητές, οι μέθοδοι και οι κλάσεις που δηλώνονται ως **public**, είναι δυνατόν να προσπελαστούν από οποιαδήποτε άλλη κλάση, όπου και αν βρίσκεται αυτή.

Όσες μεταβλητές και μέθοδοι χαρακτηρίζονται ως **private**, μπορούν να προσπελαστούν και να χρησιμοποιηθούν μόνο από το ίδιο το αντικείμενο. Δε μπορούν να προσπελαστούν από πουθενά έξω από αυτό.

Τέλος, οι μεταβλητές και οι μέθοδοι που έχουν δηλωθεί ως **protected** σε μια κλάση, μπορούν να προσπελαστούν από όλες τις κλάσεις που βρίσκονται στο ίδιο πακέτο ή από τις *υποκλάσεις* (subclasses) αυτής της κλάσης για τις οποίες θα μιλήσουμε στη συνέχεια.

Εάν δε χρησιμοποιηθεί κανένας από τους παραπάνω προσδιοριστές για μια μεταβλητή, μέθοδο, ή κλάση, τότε αυτή μπορεί να χρησιμοποιηθεί μόνο από τις κλάσεις που βρίσκονται στο ίδιο πακέτο.

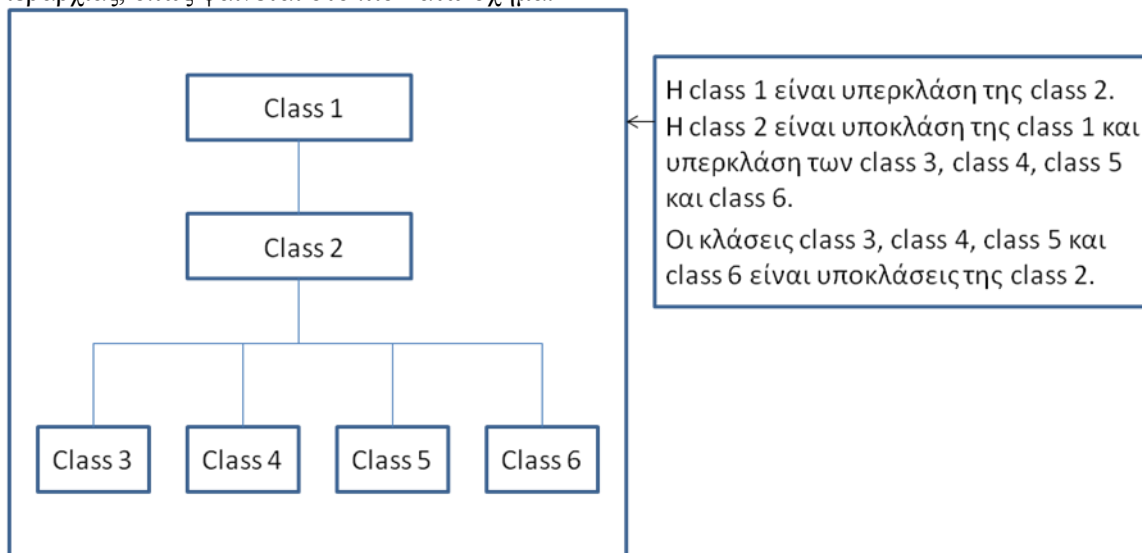
-Ένα άλλο χαρακτηριστικό του αντικειμενοστραφούς προγραμματισμού είναι ο **πολυμορφισμός**. Πολυμορφισμό έχουμε στην περίπτωση που η ίδια μέθοδος ή ο ίδιος τελεστής χρησιμοποιείται με διαφορετικούς τύπους δεδομένων. Για να πάρουμε μια ιδέα για την έννοια του πολυμορφισμού, ας εξετάσουμε ένα ανάλογο παράδειγμα από την καθημερινή ζωή. Έστω ότι βρισκόμαστε σε ένα σχολείο. Κάθε μαθητής γνωρίζει τη σημασία που έχει το κουδούνι. Όταν αυτό κτυπά, κάποιοι μαθητές

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

φεύγουν από το σχολείο γιατί σχόλασαν, άλλοι πηγαίνουν στην αίθουσά τους για το επόμενο μάθημα, και άλλοι αφήνουν την αίθουσα των μαθημάτων και πηγαίνουν στο σχολικό εργαστήριο γιατί πρόκειται να παρακολουθήσουν ένα πείραμα. Συμπέρασμα: στο χτύπημα του κουδουνιού όλοι οι μαθητές ανταποκρίνονται, αλλά ο τρόπος με τον οποίο ανταποκρίνεται ο καθένας είναι διαφορετικός.

-Η **κληρονομικότητα** είναι μια από τις βασικές ιδέες του αντικειμενοστραφούς προγραμματισμού. Με την κληρονομικότητα, επιτυγχάνεται η επαναχρησιμοποίηση κώδικα ο οποίος έχει γραφεί πριν και είναι ένα από τα σημαντικά πλεονεκτήματα των γλωσσών OOP σε σχέση με τις συμβατικές γλώσσες. Σύμφωνα με την ιδέα αυτή, ένα αντικείμενο μπορεί να κληρονομή τα δεδομένα και τις μεθόδους ενός άλλου αντικειμένου. Μπορεί επίσης να τους επιφέρει τροποποιήσεις.

Στην java, όλες οι κλάσεις (οι διαθέσιμες καθώς και αυτές που γράφουμε), ακολουθούν μια ιεραρχία. Κάθε κλάση έχει μια υπερκλάση (superclass), η οποία βρίσκεται ιεραρχικά πάνω από αυτή. Μπορεί επίσης να έχει μία ή περισσότερες υποκλάσεις (subclasses), οι οποίες βρίσκονται σε κατώτερο επίπεδο ιεραρχίας, όπως φαίνεται στο πιο κάτω σχήμα:



Εικόνα 3. διάγραμμα κλάσεων-υποκλάσεων

Οι υποκλάσεις κληρονομούν αυτόματα όλες τις μεταβλητές και τις μεθόδους από τις υπερκλάσεις τους. Στην κορυφή όλων των κλάσεων της java βρίσκεται η κλάση **Object**. Όλες οι κλάσεις κληρονομούν από αυτή την υπερκλάση. Έτσι, όταν ορίζουμε μια κλάση για την οποία δεν υπάρχει ορισμός υπερκλάσης, η java θεωρεί αυτόματα ότι η οριζόμενη κλάση είναι υποκλάση της **Object**.

2.3.6 Γεγονότα

Όταν συμβαίνει ένα γεγονός (event), όταν για παράδειγμα ο χρήστης αλληλεπιδρά με κάποιο στοιχείο μιας γραφικής εφαρμογής κινώντας το ποντίκι, πατώντας ένα κουμπί ή επιλέγοντας ένα στοιχείο μιας λίστας, δημιουργείται ένα αντικείμενο γεγονότος (event object). Το αντικείμενο γεγονότος περιέχει πληροφορίες για το ίδιο το γεγονός καθώς επίσης και για την πηγή από την οποία προέρχεται, ψάχνει δε να βρει μια ειδική μέθοδο η οποία θα χειριστεί το γεγονός αυτό. Υπάρχουν διάφορες κατηγορίες γεγονότων, μερικές από τις οποίες έχουν χρησιμοποιηθεί στην εφαρμογή με την οποία θα ασχοληθούμε στη συνέχεια, και κάποιες από αυτές είναι:

ActionEvent	Δημιουργείται από κάποια ενέργεια του χρήστη, όπως πάτημα σε ένα κουμπί, διπλοπάτημα σε κάποιο από τα περιεχόμενα μιας λίστας ή ενός μενού, ή enter σε ένα στοιχείο κειμένου.
-------------	---

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

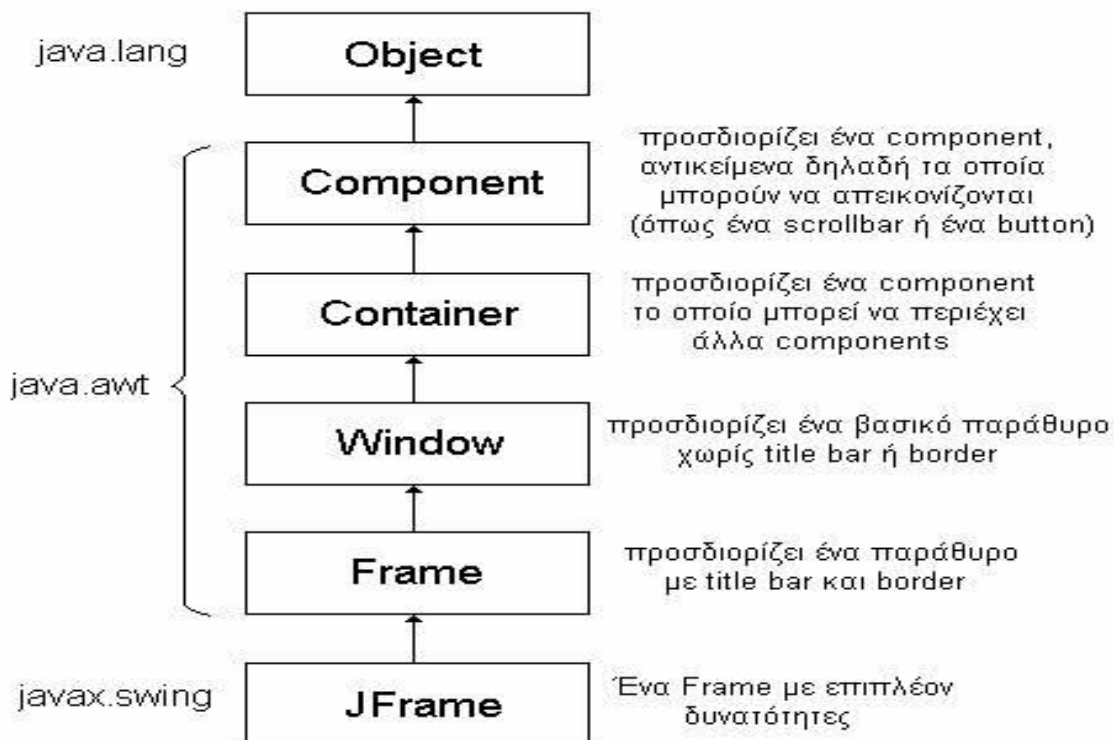
MouseEvent	Δημιουργείται όταν συμβαίνει μετακίνηση, μεγιστοποίηση, ελαχιστοποίηση ή κλείσιμο παραθύρου.
KeyEvent	Δημιουργείται όταν πιέζεται ή απελευθερώνεται ένα πλήκτρο του πληκτρολογίου.

πίνακας 1. γεγονότα αλληλεπίδρασης

2.3.7 Windows στη java

Τα βασικά συστατικά στοιχεία που χρησιμοποιούμε για τη δημιουργία των windows στη java, βρίσκονται συγκεντρωμένα σε δύο πακέτα το **java.awt** και το **javax.swing**.

Το πακέτο java.awt ξεκίνησε με την java 1.1, οι κλάσεις του οποίου ήταν αυτές που αρχικά χρησιμοποιούνταν για την δομή ενός GUI (Graphical User Interface). Με την java 2, στη συνέχεια, πολλές από αυτές υπερκαλύφθηκαν από τις κλάσεις του πακέτου javax.swing, οι περισσότερες από τις οποίες αφορούν συστατικά GUI, τα οποία καλούνται και **Swing components**. Τα Swing components παρέχουν μεγαλύτερες δυνατότητες σε σχέση με τις αντίστοιχες κλάσεις του java.awt, χωρίς όμως να παύουν να σχετίζονται και να βασίζονται σε αυτές. Η βασική διαφορά των δύο αυτών πακέτων είναι ότι το πρωταρχικό java.awt εξαρτάται περισσότερο από τον native κώδικα ενός γραφικού περιβάλλοντος, γι' αυτό και έχει σημαντικά περιορισμένες δυνατότητες σε σχέση με την ανάπτυξη ενός GUI. Αντίθετα, τα Swing components επειδή είναι εξολοκλήρου φτιαγμένα σε java 2 δεν περιορίζονται από τα χαρακτηριστικά της πλατφόρμας στην οποία τρέχουν, ενώ παρέχουν και δυνατότητες διαφοροποίησης στην γραφιστική τους απεικόνιση (pluggable look-and-feel). Η ιεραρχική δομή των κυριότερων κλάσεων για τη δημιουργία windows, μαζί με τα αντίστοιχα πακέτα όπου αυτές ανήκουν, έχει ως εξής:

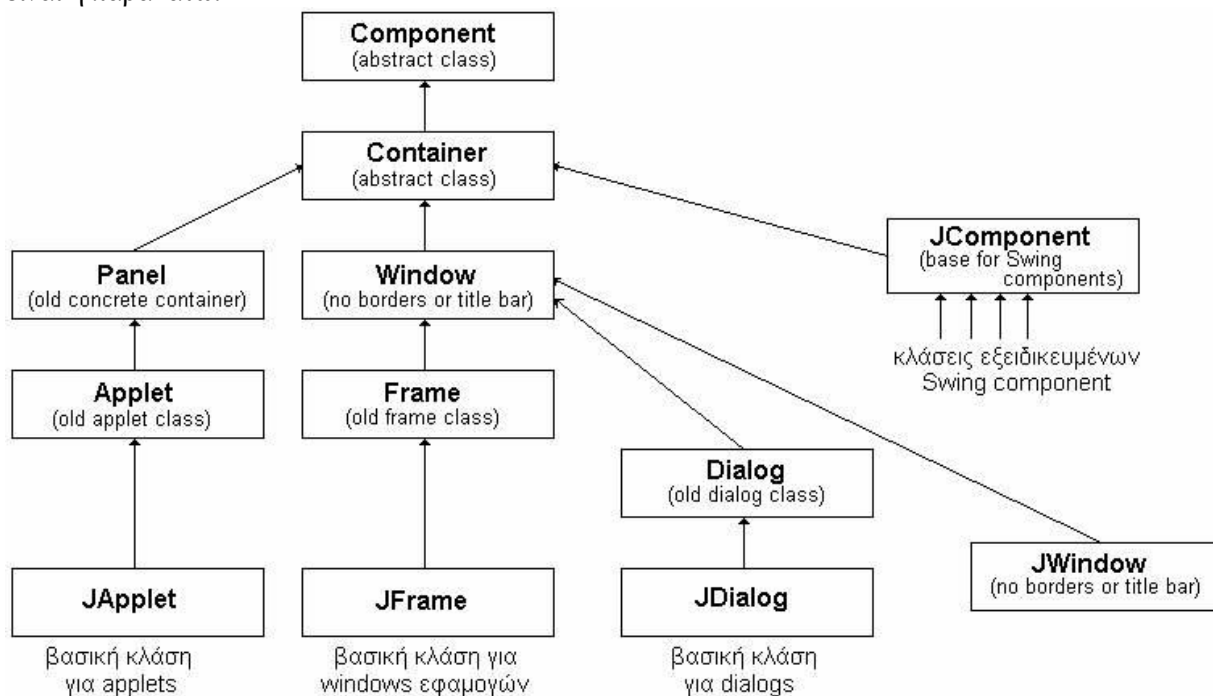


Εικόνα 4. ιεραρχική δομή των κυριότερων κλάσεων για δημιουργία windows

Η κλάση Component είναι η βασική κλάση της δομής και η βάση των Swing components. Μία επίσης βασική κλάση είναι η Container η οποία επιτρέπει σε ένα αντικείμενο Component να περιέχει άλλα components (π.χ. ένα menubar να περιέχει menus κτλ.). Με βάση την παραπάνω δομή καταλήγουμε τελικά στην κλάση JFrame η οποία μπορεί να δημιουργήσει ένα window με επιπλέον δυνατότητες. Η κλάση αυτή αποτελεί ένα τυπικό γραφιστικό περιβάλλον για μία εφαρμογή.

-Components και Containers

Είδαμε πως ένα **component** αφορά ένα οποιοδήποτε γραφιστικό συστατικό το οποίο μπορεί να απεικονιστεί, να εμφανιστεί δηλαδή στη οθόνη του Η/Υ. Επίσης σύμφωνα και με την παραπάνω ιεραρχική δομή, ένα component μπορεί να είναι ένα αντικείμενο οποιασδήποτε κλάσης προέρχεται από την κλάση Component (όπως για παράδειγμα τα αντικείμενα της κλάσης JFrame). Μια πιο αναλυτική -αλλά και πάλι όχι ολοκληρωμένη- ιεραρχική δομή μερικών υποκλάσεων της Component είναι η παρακάτω:



Εικόνα 5. ιεραρχική δομή μερικών υποκλάσεων της component

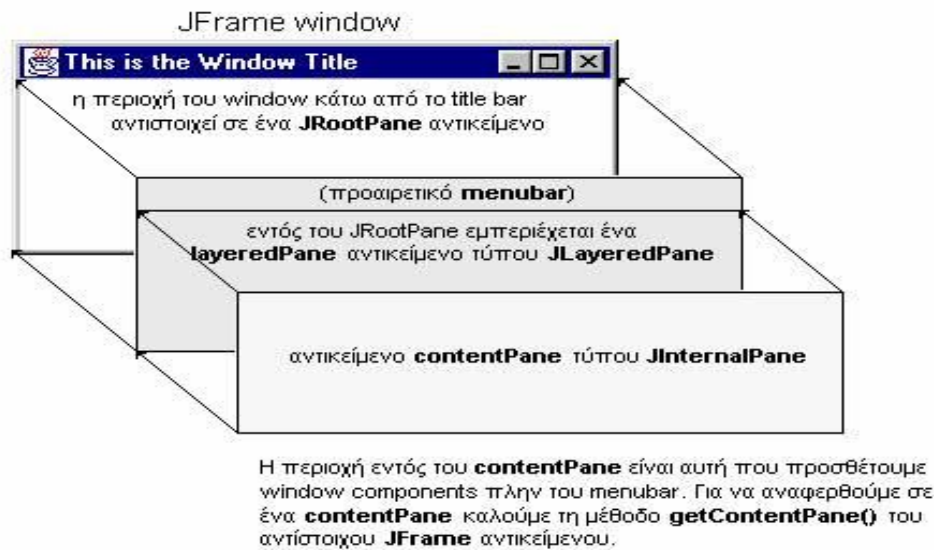
Οι παραπάνω υποκλάσεις της Component χρησιμοποιούνται αρκετά συχνά από τις εφαρμογές μας. Επιπρόσθετα, προσδιορίζονται και με έναν διαφορετικό τρόπο: όλες οι κλάσεις οι οποίες προέρχονται από την κλάση Container έχουν το χαρακτηριστικό ότι μπορούν να ενσωματώσουν άλλα αντικείμενα, οι κλάσεις των οποίων είναι υποκλάσεις της Component. Για τον λόγο αυτόν οι κλάσεις αυτές ονομάζονται και **containers**. Επιπλέον επειδή η κλάση Container είναι άμεση υποκλάση της Component, **κάθε container αντικείμενο είναι επίσης και component**. Μπορεί λοιπόν ένα αντικείμενο container να ενσωματώνει άλλα container αντικείμενα. Από την ιδιότητα αυτή εξαιρούνται -για ευνόητους λόγους- τα αντικείμενα της κλάσης Window και των υποκλάσεών της. Τα αντικείμενα αυτά, δεν μπορούν να ενσωματώνονται σε άλλα container αντικείμενα.

Οι κλάσεις JApplet, JFrame, JDialog, JComponent και JWindow ανήκουν όλες στο πακέτο javax.swing. Η κλάση Applet ανήκει στο πακέτο java.applet και όλες οι υπόλοιπες στο πακέτο java.awt. Η βασική διαφορά ανάμεσα σε ένα JFrame και ένα Window αντικείμενο, είναι πως ένα JFrame αντικείμενο αποτελεί το βασικό παράθυρο για μία εφαρμογή (σε αντίθεση με ένα Window). Για τον λόγο αυτόν τα JFrame αντικείμενα είναι αυτά που σε μία εφαρμογή περιέχουν όλα τα υπόλοιπα. Έτσι σε ένα τέτοιο αντικείμενο μπορούμε για παράδειγμα να ενσωματώσουμε:

- αντικείμενα της κλάσης Window τα οποία θα χρειαστεί η εφαρμογή μας
- αντικείμενα της κλάσης JDialog τα οποία αφορούν dialog window, προκειμένου να εισάγουμε δεδομένα που θα χρησιμοποιήσει η εφαρμογή μας
- αντικείμενα της κλάσης JComponent (και των υποκλάσεων αυτής) τα οποία αφορούν Swing components, στοιχεία δηλαδή του GUI όπως menus, buttons, checkboxes κλπ.

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

Προκειμένου σε ένα αντικείμενο JFrame να ενσωματώσουμε οποιοδήποτε από τα παραπάνω αντικείμενα, κάνουμε χρήση των window pane. Έτσι ένα JFrame window αποτελείται από τα εξής pane:



Εικόνα 6. συστατικά JFrame window

JRootPane: η περιοχή του JFrame παράθυρου, που βρίσκεται αμέσως κάτω από το title bar

layeredPane: αντικείμενο τύπου **JLayeredPane** το οποίο εμπεριέχεται μέσα σε ένα JRootPane και αφορά στη διαχείριση του menubar (εάν υπάρχει), καθώς επίσης και στη διαχείριση διαφορετικών ομάδων component που ομαδοποιούνται σε διαφορετικά layer

contentPane: αντικείμενο τύπου **JContentPane** το οποίο αντιστοιχεί στην περιοχή του layeredPane κάτω από το menubar και αφορά στη διαχείριση όλων των υπόλοιπων GUI component που περιλαμβάνει ένα window, καθώς και στην εμφάνιση κειμένου ή ζωγραφικής εντός των ορίων του

glassPane: αποτελεί αντικείμενο το οποίο αντιστοιχεί στην περιοχή του JRootPane και αφορά components τα οποία χρειάζεται να απεικονίζονται πάντοτε πάνω από όλα τα υπόλοιπα components του window (π.χ. pop-up menus, γραφικά τα οποία τρέχουν μπροστά από ένα στατικό background κ.ο.κ.)

2.4 Βάση δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά οργανωμένα (formatted) σχετιζόμενα δεδομένα. Στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων (DBMS).

Εφόσον η εφαρμογή θα χρησιμοποιεί τεράστιο πλήθος συσχετιζόμενων δεδομένων ήταν λογικό από την αρχή η χρήση μιας βάσης δεδομένων. Στην java συμπεριλαμβάνεται μια διασύνδεση JDBC η οποία καθορίζει τον τρόπο επικοινωνίας μιας εφαρμογής με σχεσιακές βάσεις δεδομένων. Η γλώσσα με αυτόν τον τρόπο κατορθώνει να υποστηρίξει τις σχεσιακές βάσεις δεδομένων που δημιουργούνται από διαφορετικά DBMS, όπως κατορθώνει να υποστηρίξει και τα διαφορετικά λειτουργικά συστήματα. Το JDBC δίνει στον προγραμματιστή τις εξής δυνατότητες:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

- Να συνδέσει μια εφαρμογή με μια υπάρχουσα βάση δεδομένων.
- Να δημιουργήσει εντολές SQL και να τις περάσει στη βάση δεδομένων προκειμένου να εκτελεστούν διάφορες εργασίες.
- Να πάρει τα αποτελέσματα των επεξεργασιών των εντολών SQL από τη βάση δεδομένων.
- Να επεξεργαστεί περαιτέρω τα αποτελέσματα που έλαβε.

Το πρόγραμμα για την κατασκευή της Βάσης Δεδομένων που θα χρησιμοποιήσουμε θα είναι η Microsoft Office Access 2007.

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

3. Ηλεκτρονικό λεξικό e-Dictionary

3.1 Περιγραφή της βάσης δεδομένων

Για την δημιουργία της εφαρμογής είναι απαραίτητη η δημιουργία βάσης δεδομένων στην οποία θα καταχωρούνται όλα τα δεδομένα που θα διαχειρίζεται ο χρήστης μέσω της εφαρμογής. Τα βασικά δεδομένα που περιλαμβάνονται στην βάση είναι τα εξής: Ένας πίνακας με τα στοιχεία των λημμάτων του λεξικού τα οποία μεταξύ άλλων είναι η ετυμολογία, η ερμηνεία, τα συνώνυμα, τα αντίθετα και τα επιρρήματα. Επίσης περιλαμβάνει ένα πίνακα με τα κύρια ονόματα με επιπλέον πληροφορία την ερμηνεία. Παρομοίως έχουμε δυο επιπλέον πίνακες με το ελληνικό και αγγλικό αρκτικόλεξο αντίστοιχα που περιλαμβάνουν στοιχεία την ερμηνεία.

Για τη σωστή λειτουργία της εφαρμογής η βάση δεδομένων θέλουμε να εξασφαλίζει:

- για κάθε πίνακα να μην επιτρέπει τις διπλότυπες εγγραφές
- για κάθε πίνακα να μην επιτρέπει την εισαγωγή μηδενικών εγγραφών
- την σωστή ταξινόμηση των δεδομένων σε αλφαβητική σειρά
- την συμπίκνωση των δεδομένων της βάσης για εξοικονόμηση μνήμης

3.2 Ανάλυση απαιτήσεων ηλεκτρονικού λεξικού

Η ανάλυση απαιτήσεων, για οποιοδήποτε έργο πληροφορικής, προσφέρει στο σύνολο των εμπλεκομένων (προγραμματιστών, αναλυτών, χρηστών, πελατών) σημαντική ευκαιρία για καλύτερη οργάνωση, προγραμματισμό και διαχείριση έργου. Τους φέρνει σε επαφή με την πραγματική φύση του έργου, φωτίζει αθέατες πλευρές και αδυναμίες, εντοπίζει τις ελλείψεις, προφυλάσσει και προειδοποιεί. Εν ολίγοις, η εξέλιξη του έργου είναι σε μεγάλο βαθμό προβλέψιμη και οι δυσάρεστες εκπλήξεις περιορίζονται αισθητά.

Θέλουμε λοιπόν να δημιουργήσουμε ένα ηλεκτρονικό λεξικό το οποίο θα παρέχει στο χρήστη πολύτιμες πληροφορίες (σημασιολογικές, συντακτικές κτλ.) του λεξιλογίου της νεοελληνικής γλώσσας. Το λεξικό θα είναι φτιαγμένο για όλο το κοινό, για κάθε Έλληνα χρήστη, ανεξαρτήτως ηλικίας και μόρφωσης, επαγγέλματος και γραμματικών γνώσεων, από το μαθητή του σχολείου μέχρι τον επιστήμονα. Ωστόσο μπορούμε να υποθέσουμε πως το κοινό που θα προσεγγίσει κυρίως το λεξικό θα είναι μαθητές αλλά και ακαδημαϊκοί/καθηγητές.

Η εφαρμογή θα σχεδιαστεί από την αρχή, δεν θα χρησιμοποιηθεί δηλαδή κάποια έτοιμη πλατφόρμα. Αυτό που μας ενδιαφέρει είναι η διαχείριση του περιεχομένου να γίνεται εύκολα και απλά, να μην απαιτείται συντήρηση ούτε και σύνδεση με κάποιο δίκτυο και να συνεργάζεται με μια βάση δεδομένων της Access που διαθέτουμε. Θέλουμε να μπορούμε να κάνουμε προσθήκη, διαγραφή και επεξεργασία των λημμάτων του λεξικού, οι όποιες ενέργειες κάνουμε θα αποθηκεύονται μόνιμα στη βάση δεδομένων. Σε ξεχωριστές καρτέλες επίσης να υπάρχουν οι κατηγορίες ΚΥΡΙΑ ΟΝΟΜΑΤΑ και ΑΡΚΤΙΚΟΛΕΞΑ στις οποίες θα δίνεται και εδώ η δυνατότητα για προσθήκη, διαγραφή και επεξεργασία. Σε όλες τις κατηγορίες θα υπάρχει δυνατότητα αναζήτησης ενός λήμματος. Θέλουμε επίσης η εφαρμογή να μπορεί να εμφανίζεται στη γραμμή εργασιών του υπολογιστή μας όταν πατάμε το κουμπί της ελαχιστοποίησης στο παράθυρό της.

Το λεξικό θα δουλεύει σε περιβάλλον των Windows. Δεν θα χρειάζεται κάποια εγκατάσταση παρά μόνο στην περίπτωση που δεν είναι εγκατεστημένη η Java και η Access στον υπολογιστή του χρήστη. Σε αυτήν την περίπτωση πρέπει να δίνονται πλήρεις οδηγίες για τα συμπληρωματικά αρχεία που θα χρησιμοποιηθούν προκειμένου να λειτουργήσει η εφαρμογή. Το λεξικό δεν θα πραγματοποιεί κάποιο έλεγχο στον υπολογιστή για να τσεκάρει ότι είναι εγκατεστημένη η java και η access. Ο χρόνος ανταπόκρισης του λεξικού κατά την εκκίνηση της λειτουργίας του να είναι maximum 10 δευτερόλεπτα.

3.3 Σχεδιάζοντας την διεπαφή

Αρκετοί από τους υπέρμαχους της χρηστοκεντρικής σχεδίασης έχουν παρουσιάσει σύνολα ‘χρυσών κανόνων’ ή ευρετικές μεθόδους. Αν και αναπόφευκτα αυτοί οι κανόνες σχεδίασης είναι ευρείας εφαρμογής, οι οποίοι μπορεί να μην είναι πάντα εφαρμόσιμοι σε κάθε περίπτωση, παρέχουν στους σχεδιαστές μια χρήσιμη λίστα ή σύνοψη βασικών υποδείξεων για την σχεδίαση. Είναι σαφές ότι οποιοσδήποτε σχεδιαστής ακολουθήσει αυτούς τους απλούς κανόνες, θα παράγει ένα καλύτερο σύστημα από έναν σχεδιαστή που θα τους αγνοήσει.

3.3.1 Οι 8 χρυσοί κανόνες του Shneiderman για την σχεδίαση διεπιφανειών

Οι 8 χρυσοί κανόνες του Shneiderman μας παρέχουν μια βολική και περιεκτική σύνοψη των βασικών αρχών της σχεδίασης των διεπιφανειών:

- 1) Ομοιομορφία και συνέπεια στην διεπιφάνεια και αποφυγή απροσδόκητης συμπεριφοράς
- 2) Σύντομοι χειρισμοί για την διευκόλυνση των έμπειρων χρηστών
- 3) Συνεχής ανάδραση της κατάστασης του συστήματος
- 4) Οι υπο-διάλογοι πρέπει να ολοκληρώνονται σε λίγα βήματα
- 5) Πρόβλεψη για σφάλματα των χρηστών
- 6) Δυνατότητα αναίρεσης μιας ή περισσότερων ενεργειών
- 7) Ο έλεγχος της αλληλεπίδρασης θα πρέπει να είναι από την πλευρά του χρήστη και όχι του συστήματος
- 8) Το φορτίο βραχύχρονης μνήμης του χρήστη θα πρέπει να ελαχιστοποιηθεί

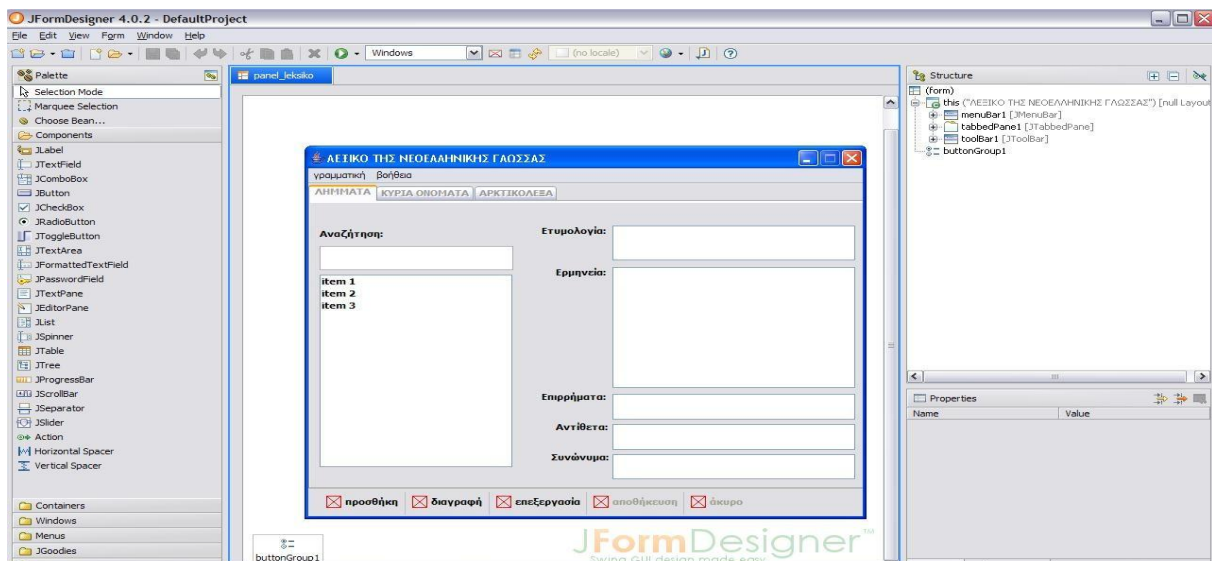
Το πώς εφαρμόζονται αυτοί οι κανόνες στην πράξη θα το δούμε στην συνέχεια αναλύοντας την διαδικασία σχεδίασης.

3.3.2 Σχεδίαση της βασικής φόρμας

Αυτό που μας ενδιέφερε πολύ ήταν η βασική φόρμα του λεξικού να είχε μια **ομοιομορφία** όσο αναφορά στην αρχιτεκτονική που θα παρουσιάζονται τα δεδομένα. Για το λόγο αυτό η ακρίβεια στη σχεδίαση της φόρμας θα ήταν ένα πολύ δύσκολο κομμάτι και χρονοβόρο αν ο κώδικας γραφόταν στο χέρι, δεδομένου και της πολυπλοκότητας που κρύβουν οι διαχειριστές διάταξης της java. Το JFormDesigner δίνει τη λύση πάνω σε αυτό το πρόβλημα πολύ εύκολα, σχεδιάζοντας υψηλής ποιότητας GUI αποφεύγοντας έτσι τον χαμηλού επιπέδου προγραμματισμό.

Στην παρακάτω εικόνα βλέπουμε το περιβάλλον του JFormDesigner και την βασική φόρμα που έχουμε κατασκευάσει για το λεξικό:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας



Εικόνα 7. περιβάλλον εργασίας JFormDesigner

Αριστερά στην εικόνα βρίσκεται η παλέτα με όλα τα συστατικά του Swing και η οποία περιλαμβάνει κουμπιά, πλαίσια ελέγχου, ετικέτες, περιγράμματα, τμήματα παραθύρων με δυνατότητες κύλισης, μενού, γραμμές εργαλείων και πολλά άλλα. Στα δεξιά αναγράφονται λεπτομερειακά οι ιδιότητες των αντικειμένων που έχουμε τοποθετήσει στην περιοχή της σχεδίασης (κέντρο) της εφαρμογής μας. Παρατηρήστε με πόση ακρίβεια είναι σχεδιασμένα τα πλαίσια κειμένου, την ίση απόσταση μεταξύ τους, το μήκος και το πλάτος τους (εκτός από το πλαίσιο κειμένου της ερμηνείας). Πάνω σε αυτήν την αρχιτεκτονική βασίστηκαν και οι άλλες δυο καρτέλες ενώ η μπάρα εργαλείων στην κάτω μεριά παρέμεινε η ίδια σε όλη την εφαρμογή. Έτσι δεν επιτυγχάνουμε μόνο ομοιομορφία στην αρχιτεκτονική του λεξικού, αλλά και μια κομψή με ακρίβεια σχεδιασμένη διάταξη των αντικειμένων που χρησιμοποιήσαμε.

Έχοντας λοιπόν έτοιμο το σχέδιο, το JFormDesigner αναλαμβάνει να μετατρέψει ότι έχουμε δημιουργήσει σε κώδικα java. Αυτόν τον κώδικα θα τον επεξεργαστούμε περαιτέρω με τη βοήθεια του Blue-j έτσι ώστε να καλύψουμε προγραμματιστικά τις επιπλέον τεχνικές απαιτήσεις που έχει η εφαρμογή.

3.3.3 Τα βασικά συστατικά της διεπιφάνειας

Είδαμε σε προηγούμενο κεφάλαιο ότι το swing αποτελείται από πολλά πακέτα τα οποία περιλαμβάνουν εκατοντάδες κλάσεις και διασυνδέσεις. Η κλάση JComponent είναι η υπερκλάση όλων των ελαφρών συστατικών του Swing. Υποκλάσεις της JComponent που θα χρησιμοποιηθούν στην εφαρμογή είναι:

AbstractButton – Είναι η ανώτερη κλάση κουμπιών. Υποκλάσεις της είναι οι κλάσεις **JButton** (κλασικά και βελτιωμένα κουμπιά), **JMenuItem** (μενού και συστατικά μενού), **JToggleButton** (πλαίσια ελέγχου και ραδιοπλήκτρα).

JLabel – Ετικέτα που μπορεί να περιέχει κείμενο, εικόνα ή και τα δύο.

JList – Μια λίστα η οποία μπορεί να πάρει διάφορες μορφές.

JMenuBar – Μια γραμμή μενού που μπορεί να προστεθεί σε οποιοδήποτε συστατικό υποδοχέα.

JScrollBar – Η ράβδος κύλισης του Swing.

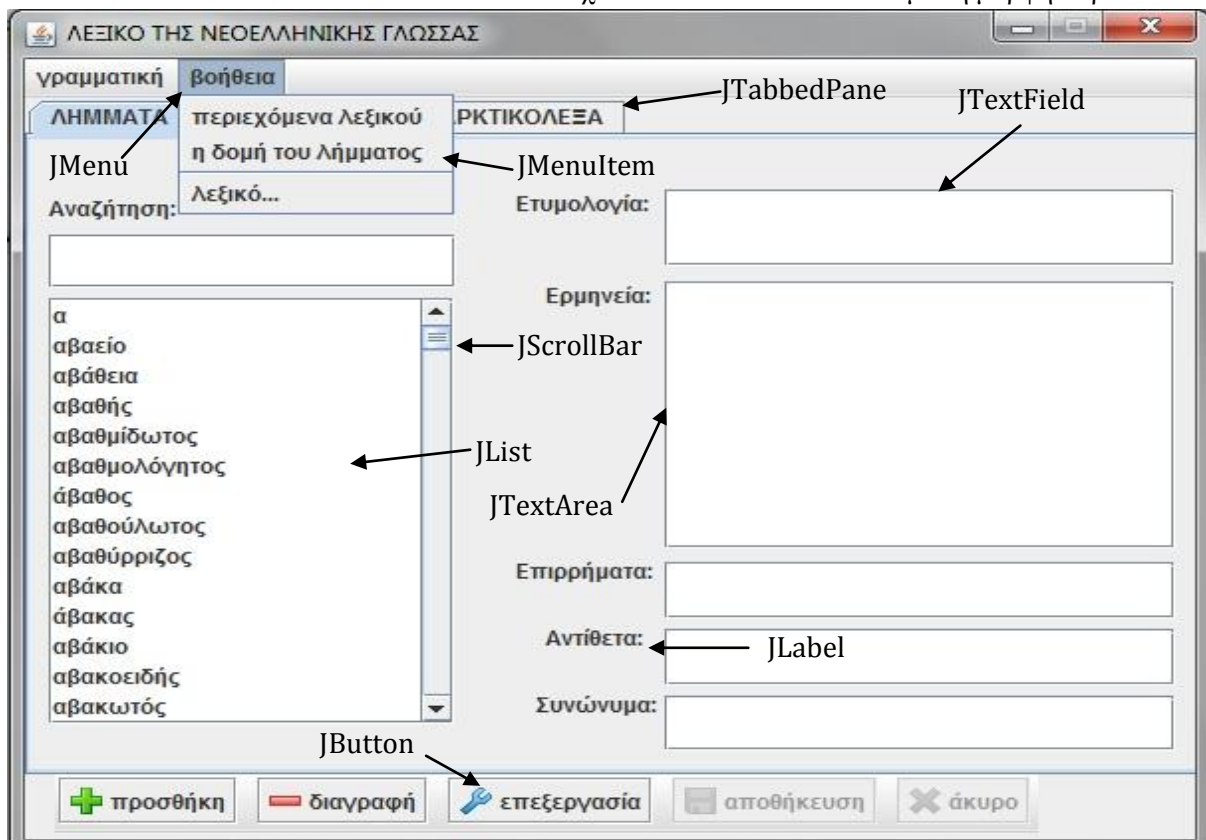
JScrollPane – Παράθυρο που περιέχει ράβδο κύλισης.

Τα συστατικά κειμένου κληρονομούν κοινές μεθόδους από την κλάση **JTextComponent**:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

JTextField και **JTextArea** – Μπορούμε να τοποθετήσουμε κείμενο στο πεδίο ή να πάρουμε κείμενο από αυτό με τις μεθόδους `setText` και `getText`.

Ειδικά πλαίσια: **JTabbed** – Τοποθετεί κάθε στοιχείο πίσω από τα υπόλοιπα με τη μορφή καρτελών.



Εικόνα 8. βασικά συστατικά διεπαφής

3.4 Προγραμματίζοντας με java

Σε αυτήν την ενότητα θα εμβαθύνουμε σε κάποια βασικά σημεία του κώδικα που δημιουργήσαμε έτσι ώστε να κατανοήσουμε πως ακριβώς λειτουργεί η εφαρμογή από την πλευρά του προγραμματιστή.

3.4.1 Πως φορτώνονται τα δεδομένα

Ένα ιδιαίτερο σημείο που πρέπει να σταθούμε, είναι να δούμε τον τρόπο με τον οποίο φορτώνεται η βάση δεδομένων του λεξικού. Υπάρχουν δύο τρόποι για να γίνει μια σύνδεση με τη βάση δεδομένων: 1) Απευθείας σύνδεση με τη βάση. Ο πιο απλός τρόπος αλλά ίσως να μην δουλέψει σε όλα τα συστήματα. 2) Καταχώρηση της βάσης δεδομένων με το ODBC. Λίγο πιο περίπλοκο αλλά θα δουλέψει σε όλα τα συστήματα. Ακολουθήθηκε η πρώτη προσέγγιση για ένα και μόνο λόγο: δεν απαιτεί εμπλοκή του χρήστη στην διαχείριση αρχείων προέλευσης δεδομένων ODBC. Ο χρήστης σε καμία περίπτωση δεν πρέπει να του ζητηθεί να κάνει εργασίες προκειμένου να κάνει το λεξικό να δουλέψει στο σύστημά του. Επιπλέον η πρώτη προσέγγιση είναι πιο γρήγορη και εύκολη λύση.

```
.....  
.....  
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
String url = "jdbc:odbc:Driver={Microsoft Access Driver " + "(*.mdb,  
*.accdb)};DBQ=source\\dokimi.accdb";  
con = DriverManager.getConnection(url,"cap","cap");
```

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

.....
.....
Στην πρώτη γραμμή φορτώνουμε τους κατάλληλους οδηγούς με τη μέθοδο `forName` της κλάσης `Class`. Στη δεύτερη γραμμή δημιουργούμε ένα αλφαριθμητικό το οποίο περιέχει την κατάλληλη URL που προσδιορίζει μονοσήμαντα την πηγή των δεδομένων. Σημειώστε `dokimi` είναι το όνομα της βάσης δεδομένων. Στην τέταρτη γραμμή δημιουργούμε ένα αντικείμενο `Connection` (`con`). Το αντικείμενο αυτό πραγματοποιεί τη σύνδεση με την πηγή δεδομένων και κατασκευάζει ένα περιβάλλον μέσα στο οποίο μπορούμε να δημιουργούμε και να εκτελούμε εντολές. Τα "`cap`", "`cap`" δεν είναι τίποτα άλλο από το `username` και το `password` της βάσης δεδομένων.

Αφού λοιπόν φορτώσαμε τη βάση το επόμενο βήμα είναι να μεταφέρουμε τα δεδομένα στις λίστες που τις αντιστοιχούν. Έτσι λοιπόν για την λίστα με τα κύρια ονόματα τα δεδομένα μεταφέρονται με την εκτέλεση του παρακάτω κώδικα:

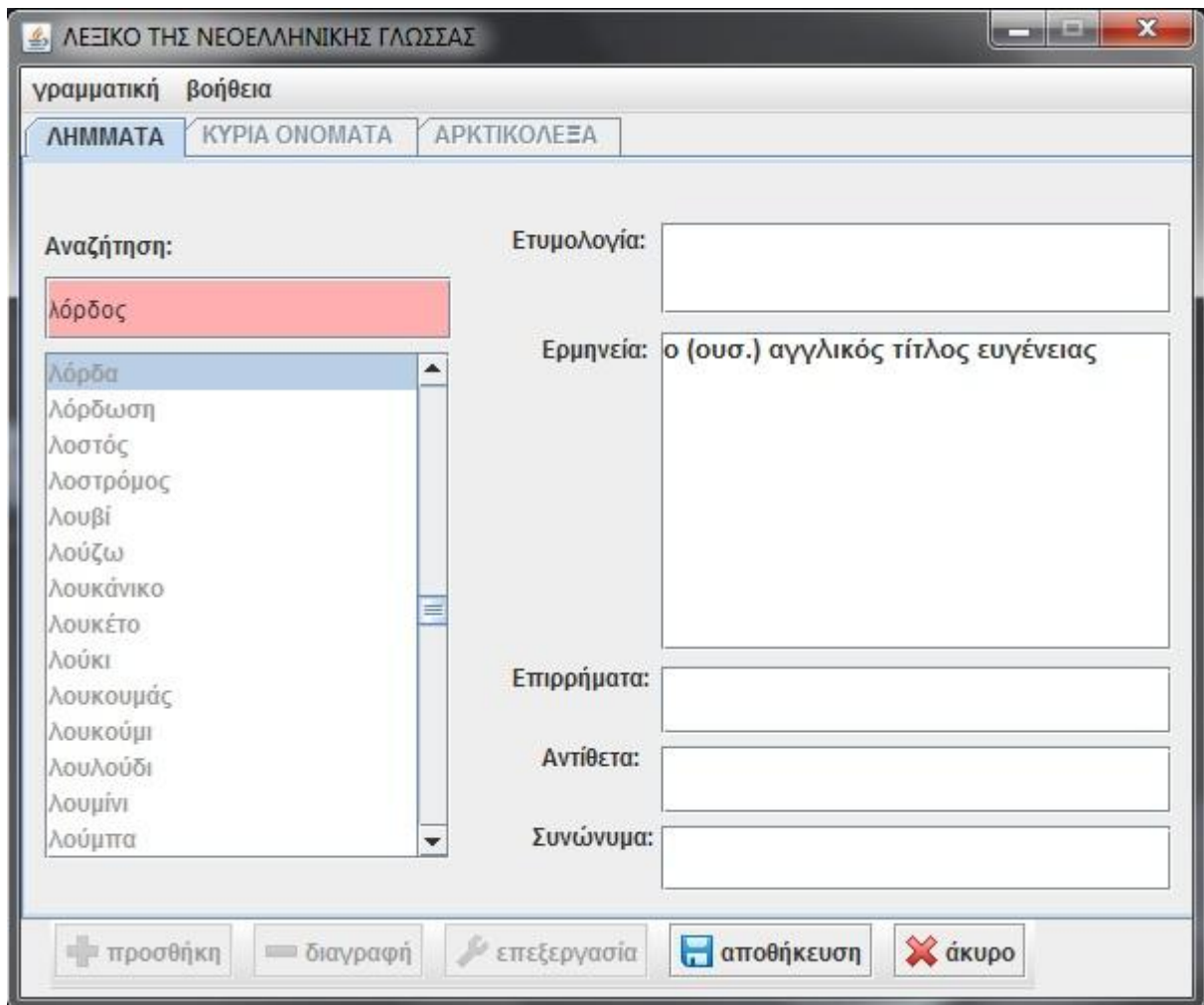
```
.....  
rs1 = stmt1.executeQuery("SELECT * FROM kyria_onomata");  
    while(rs1.next()){  
        listModel_kyriaonomata.addElement(rs1.getString(1));  
    }  
.....
```

3.4.2 Προσθήκη δεδομένων

Ας δούμε τώρα αυτή τη λειτουργία αναλυτικά που δίνει τη δυνατότητα στο χρήστη να κάνει προσθήκη λημμάτων στη βάση:

```
.....  
button_add.addActionListener(new ActionListener(){  
    public void actionPerformed(ActionEvent e){  
        if(selidaTabbed==0){ //ελέγχει ποια κατηγορία είναι επιλεγμένη  
            list_limmata.setEnabled(false); //απενεργοποιεί τη λίστα  
            Buttons1Change(); //μέθοδος που απενεργοποιεί καθορισμένα κουμπιά  
            DisableTabs1_2(); //μέθοδος που απενεργοποιεί τις υπόλοιπες καρτέλες  
            textField_search.requestFocusInWindow(); //μαρκάρει το υποχρεωτικό πεδίο με τον κέρσορα  
            κειμένου  
            textField_search.setBackground(Color.PINK); //αλλάζει το χρώμα του υποχρεωτικού πεδίου  
            σε ρόζ  
            EditTextsTrue(); //ενεργοποιεί όλα τα πεδία κειμένου  
            ClearTexts(); //καθαρίζει όλα τα πεδία κειμένου  
        }  
    }  
.....
```

Μπορούμε να δούμε ότι το πάτημα σ' ένα κουμπί δημιουργεί ένα γεγονός `ActionEvent`. Ένας ακροατής που θέλει να παρακολουθεί αυτό το κουμπί, πρέπει να υλοποιεί τη διασύνδεση `ActionListener` και να συνδέεται μαζί του με τη μέθοδο `addActionListener()`. Η διασύνδεση `ActionListener` παρέχει μόνο μια μέθοδο, τη μέθοδο `ActionPerformed(ActionEvent e)`. Μόλις πατήσουμε το κουμπί προσθήκη καλείται η μέθοδος `ActionPerformed()` του ακροατή με όρισμα το συγκεκριμένο γεγονός και αναλαμβάνει να αποκριθεί κάνοντας τις ενέργειες που του έχουμε ορίσει. Στην παρακάτω εικόνα βλέπουμε τη συμπεριφορά της εφαρμογής πατώντας το κουμπί προσθήκη:



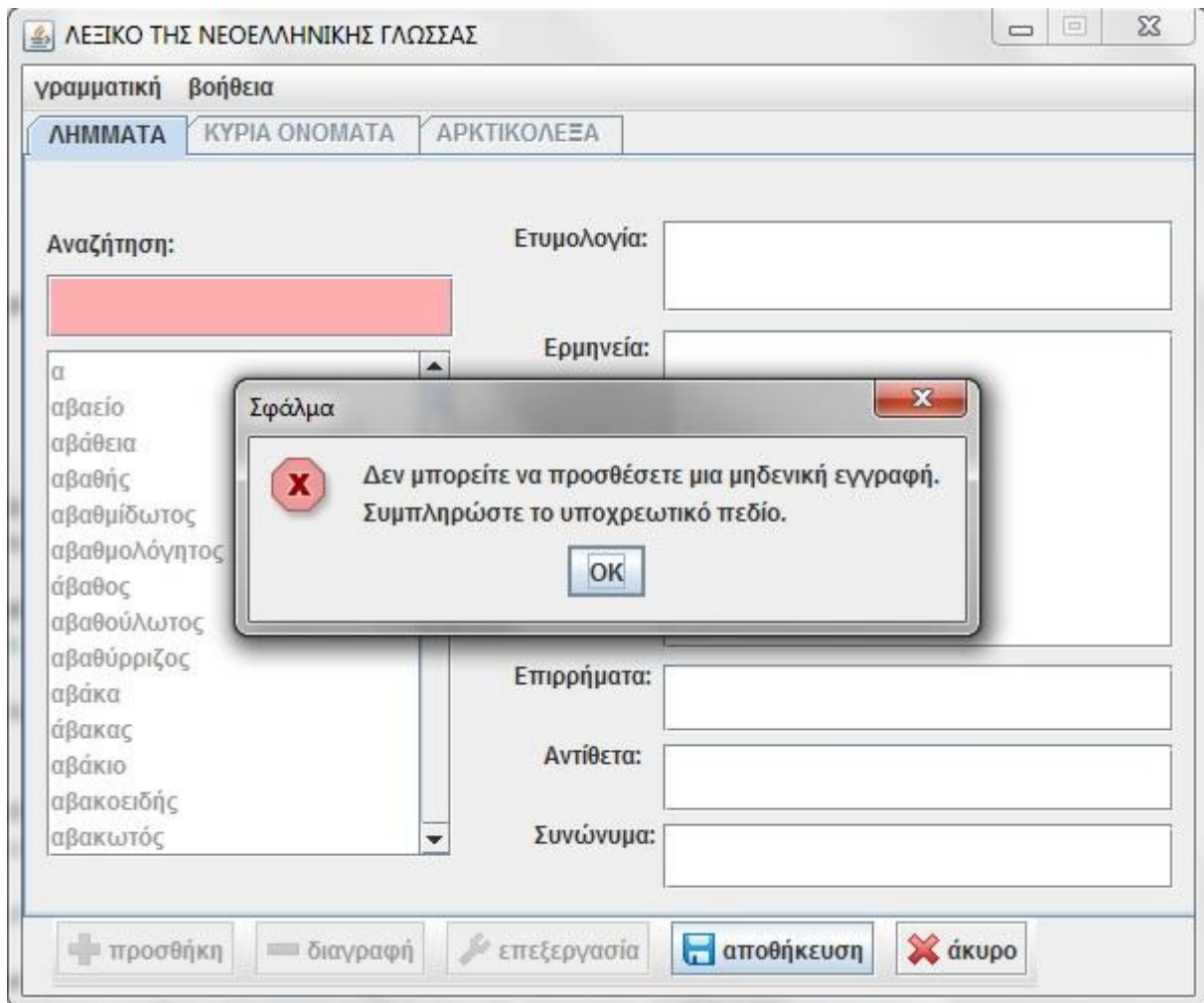
Εικόνα 9. προσθήκη δεδομένων στο λεξικό

Στη φάση αυτή η εφαρμογή περιμένει αφού συμπληρώσει ο χρήστης τις πληροφορίες που θέλει, να πατήσει το κουμπί αποθήκευση ή άκυρο.

3.4.3 Επικοινωνία με το χρήστη

Η κλάση JOptionPane διαθέτει δεκάδες μεθόδους που επιτρέπουν τη δημιουργία διαφόρων ειδών πλαισίων διαλόγου με το χρήστη για διάφορα θέματα όπως εμφάνιση μηνύματος, επιβεβαίωση, δημιουργία πλαισίου διαλόγου κτλ. Ας υποθέσουμε ότι κατά την διαδικασία προσθήκης δεδομένων πατήσουμε αποθήκευση χωρίς να έχουμε γράψει τότε θα εμφανιστεί το ακόλουθο μήνυμα:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας



Εικόνα 10. πλαίσιο διαλόγου με χρήστη

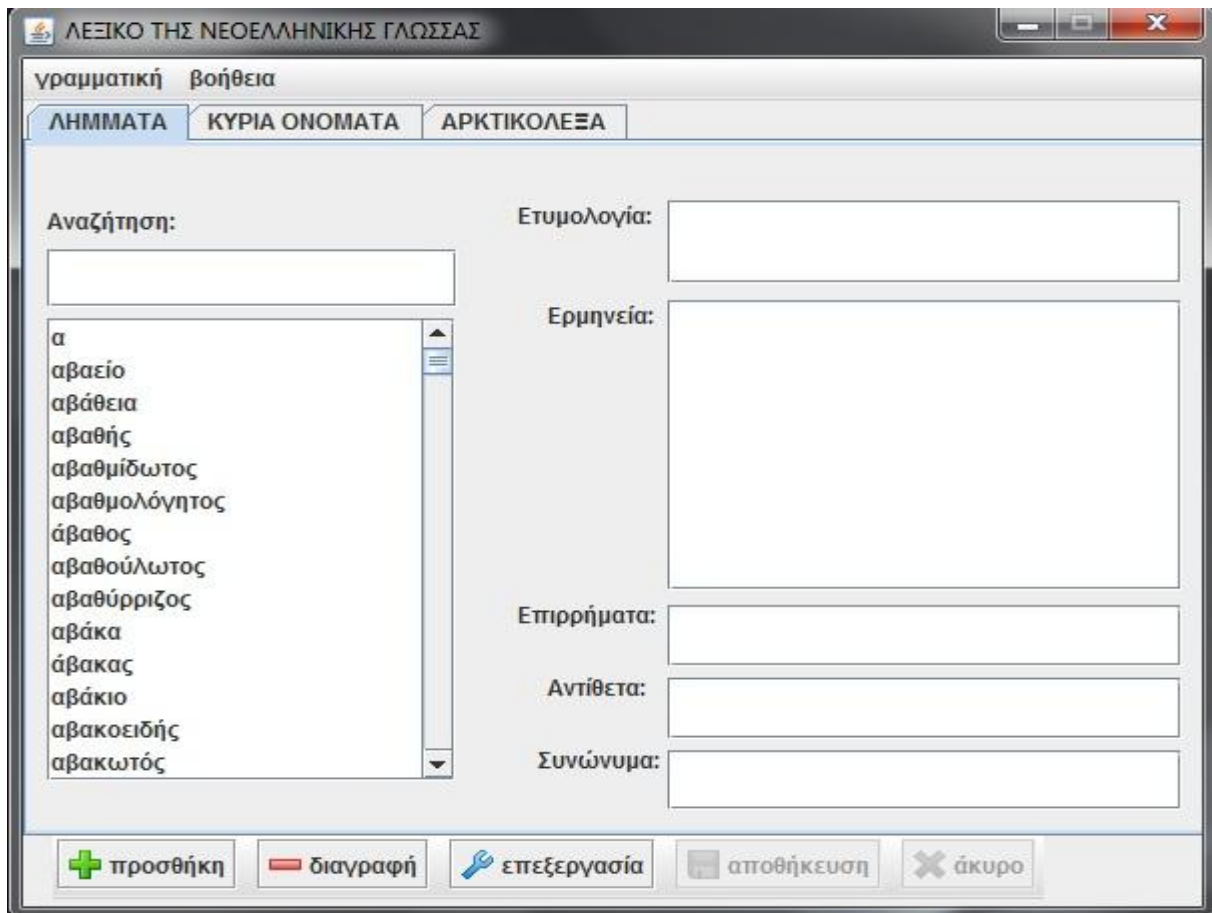
Ο κώδικας που εμφανίζει το πλαίσιο διαλόγου είναι:

```
.....  
public void ErrorMessage(){JOptionPane.showMessageDialog(this,"Δεν μπορείτε να προσθέσετε μια  
μηδενική εγγραφή. \n"+  
"Συμπληρώστε το υποχρεωτικό πεδίο.", "Σφάλμα", JOptionPane.ERROR_MESSAGE);}  
.....
```

3.5 Περιγραφή του λεξικού από την πλευρά του χρήστη

Η πραγματική μορφή της εφαρμογής κατά την εκκίνηση:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

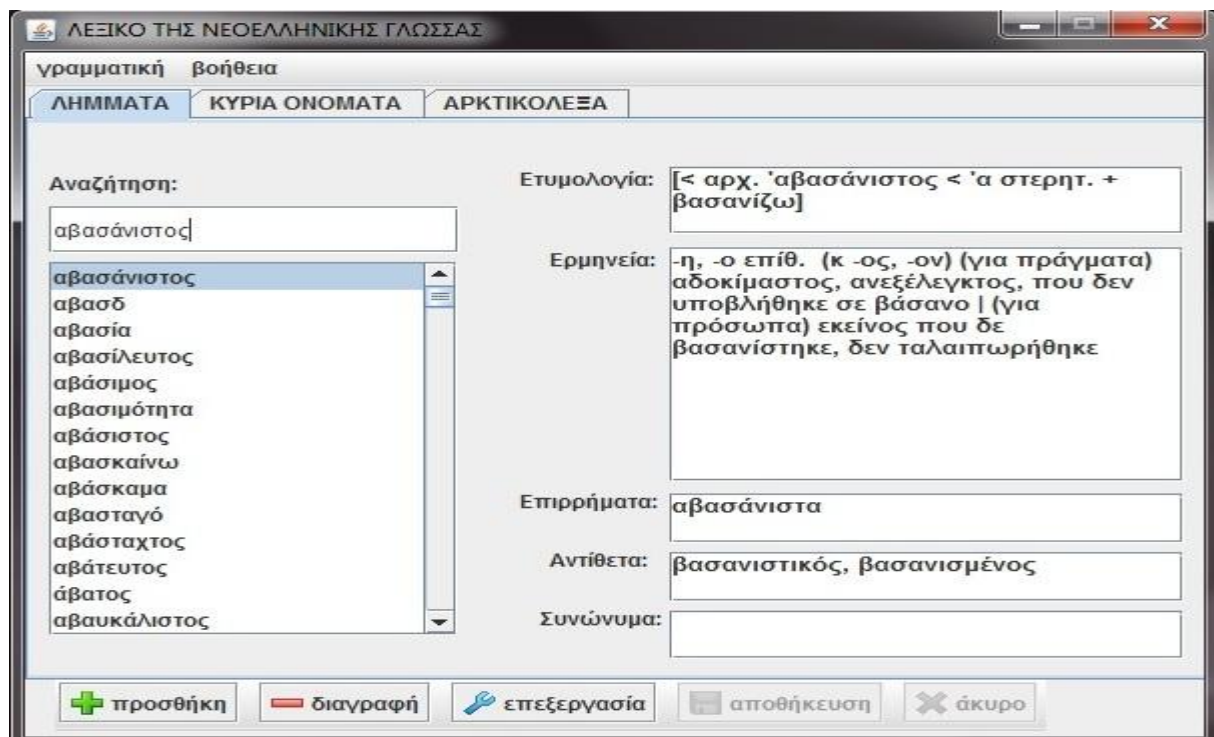


Εικόνα 11. πραγματική μορφή του λεξικού κατά την εκκίνηση

Το πρώτο πράγμα που θα πρέπει να δούμε είναι ο τρόπος με τον οποίο χωρίζονται οι τρεις κατηγορίες του λεξικού. Οι κατηγορίες ΛΗΜΜΑΤΑ, ΚΥΡΙΑ ΟΝΟΜΑΤΑ και ΑΡΚΤΙΚΟΛΕΞΑ είναι οργανωμένες σε ξεχωριστές καρτέλες (tabbed panes στη java) και η πλοήγηση ανάμεσα σε αυτές γίνεται πολύ γρήγορα.

Στα αριστερά βρίσκεται η λίστα στην οποία φορτώνονται τα δεδομένα της βάσης και πιο συγκεκριμένα το βασικό λεξιλόγιο της κατηγορίας ΛΗΜΜΑΤΑ. Αντίστοιχα φορτώνονται και τα δεδομένα των πινάκων των άλλων δυο κατηγοριών. Αξίζει να σημειώσουμε πως στην κατηγορία ΑΡΚΤΙΚΟΛΕΞΑ ο χρήστης έχει τη δυνατότητα να διαλέξει και να εμφανίσει μέσω ενός πλαισίου ελέγχου, το αγγλικό ή ελληνικό αρκτικόλεξο.

Στα δεξιά βλέπουμε τον τρόπο με τον οποίο παρουσιάζονται άμεσα οι διάφορες σημασιολογικές έννοιες ενός λήμματος. Δείτε στην παρακάτω εικόνα μια τέτοια παρουσίαση.



Εικόνα 12. εμφάνιση σημασιολογικών στοιχείων ενός λήμματος

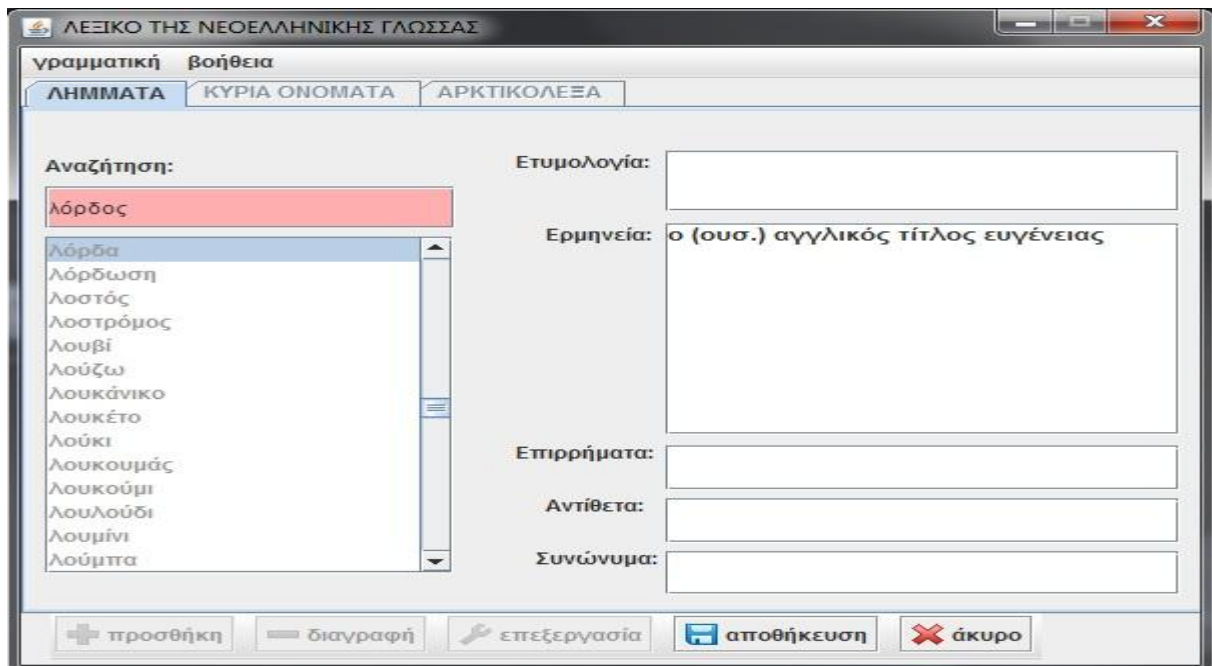
Πάνω ακριβώς από την λίστα βρίσκεται ένα πλαίσιο κειμένου στο οποίο ο χρήστης προκειμένου να ψάχνει ανάμεσα στα χιλιάδες λήμματα της λίστας μια λέξη, μπορεί να την πληκτρολογήσει απευθείας εκεί και η λίστα αυτόματα να τον παραπέμψει στο αντίστοιχο λήμμα.

3.5.1 Δυνατότητες λεξικού

Στην κάτω μεριά βρίσκεται η μπάρα εργαλείων μέσω των οποίων μπορούμε να εκτελέσουμε κάποιες λειτουργίες της εφαρμογής (ισχύουν και για τις τρεις κατηγορίες):

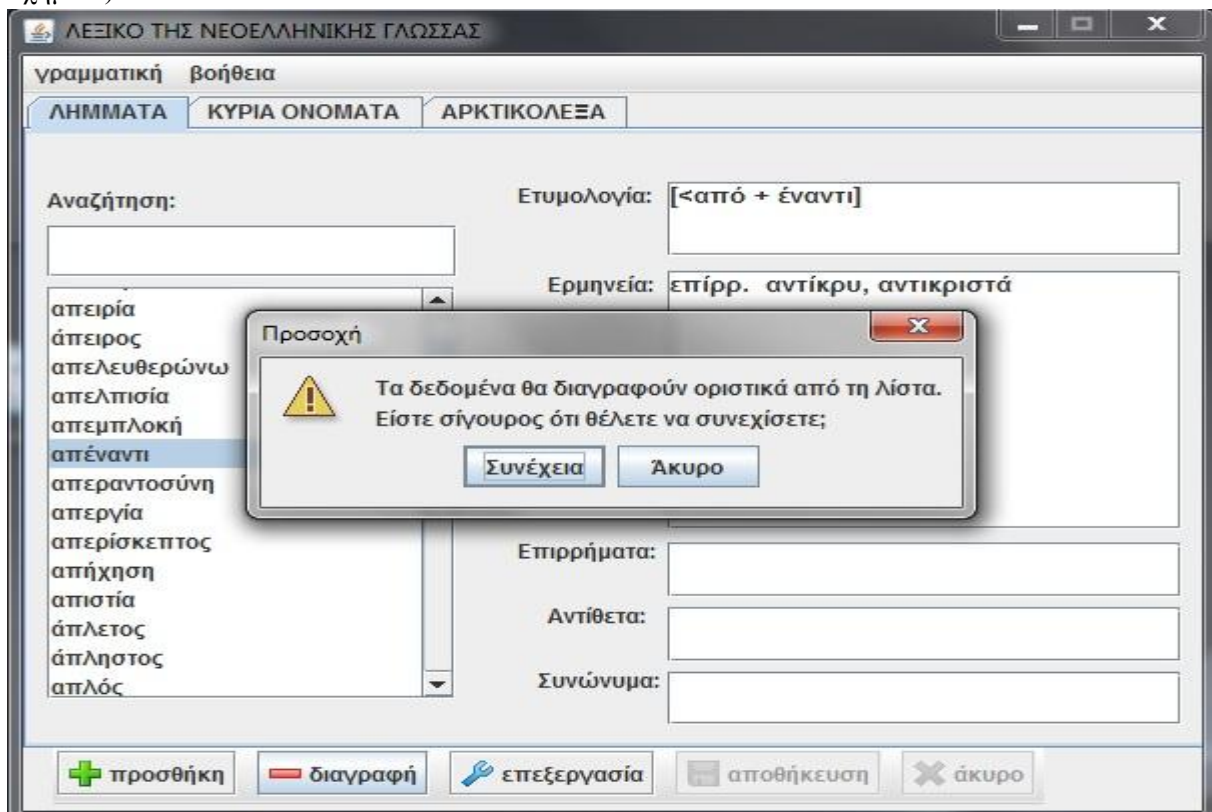
- **Προσθήκη:** βασική λειτουργία αφού δίνει τη δυνατότητα στο χρήστη να κάνει προσθήκη λημμάτων στη βάση, καθιστώντας έτσι την εφαρμογή επεκτάσιμη στα δεδομένα. Εκτελώντας τη λειτουργία αυτή, η ίδια διεπαφή από παρουσίαση δεδομένων μετατρέπεται σε μια φόρμα συμπλήρωσης και υποβολής στοιχείων λήμματος. Το μαρκαρισμένο (υποχρεωτικό) πλαίσιο είναι το σημείο όπου πρέπει να γράψουμε το λήμμα που θέλουμε να προσθέσουμε. Καθώς συμπληρώνετε αυτό το πλαίσιο η λίστα σας παραπέμπει σε άλλες λέξεις παρόμοιες έτσι ώστε να τσεκάρετε κατευθείαν ότι η λέξη που θα προσθέσετε δεν υπάρχει στο λεξικό (δείτε το σχήμα 3 για να καταλάβετε). Ύστερα αν θέλουμε μπορούμε να συμπληρώσουμε και τα πλαίσια του λήμματος με τις σημασιολογικές πληροφορίες που διαθέτουμε. Πατώντας μόλις τελειώσουμε το κουμπί "αποθήκευση" όλα τα στοιχεία που γράψαμε αποθηκεύονται μόνιμα στη βάση δεδομένων.:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας



Εικόνα 13. αποθήκευση δεδομένων στη βάση

- **Διαγραφή:** η λειτουργία αυτή δίνει τη δυνατότητα στο χρήστη να κάνει οριστική διαγραφή κάποιου επιλεγμένου λήμματος. Επιλέξτε από τη λίστα το λήμμα που θέλετε να διαγράψετε και πατήστε το κουμπί "διαγραφή". Πριν τη διαγραφή εμφανίζεται ένα προειδοποιητικό μήνυμα (δείτε σχήμα 4).



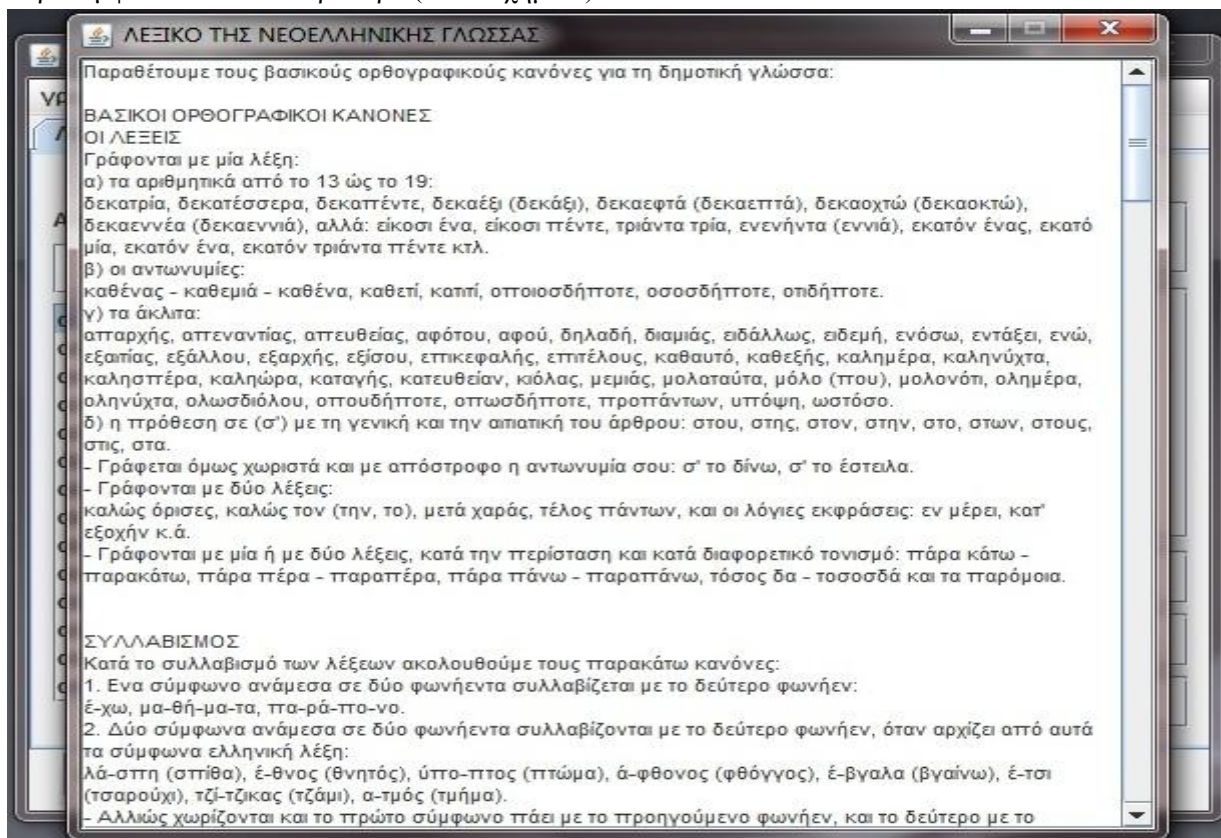
Εικόνα 14. διαγραφή δεδομένων

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

- **Επεξεργασία:** με αυτή τη λειτουργία μπορούμε να τροποποιήσουμε ή ακόμα και να συμπληρώσουμε κάποια υπάρχοντα δεδομένα στη βάση μας. Εφόσον έχετε επιλέξει το λήμμα στο οποίο θα του αλλάξετε ή θα προσθέσετε πληροφορία, πατήστε το κουμπί "επεξεργασία" και πλέον θα μπορείτε στα πλαίσια κειμένου της λέξης να γράψετε τα στοιχεία που θέλετε.
- **Αποθήκευση:** γίνεται μόνιμη αποθήκευση των εργασιών που πραγματοποιεί ο χρήστης με τις λειτουργίες "προσθήκη" και "επεξεργασία".
- **Άκυρο:** ακριβώς όπως αποθηκεύει ο χρήστης τις εργασίες του, έτσι έχει και την δυνατότητα να τις ακυρώσει κατά την διάρκεια της εκτέλεσής τους.

3.5.2 Επιπλέον υλικό

Παρότι πλέον τώρα με το internet ο χρήστης μπορεί ανά πάσα στιγμή να βρεθεί με ένα κλικ μπροστά σε ένα πλήθος πληροφοριών, θα ήταν παράλειψη σε μια τέτοια εφαρμογή να μην παρατίθενται έστω και επιγραμματικά οι βασικοί ορθογραφικοί κανόνες της ελληνικής γλώσσας. Ο λόγος μας δεν μπορεί να είναι μεμονωμένες, αποκομμένες, σκόρπιες λέξεις. Οι λέξεις δηλώνουν πληροφορίες (<<σημασίες>>) και εξ ορισμού υποχρεούνται να συντάσσονται, να συνδέονται ώστε να προκύπτει νόημα. Για να δείτε το υλικό μεταβείτε στο μενού "γραμματική" και η εφαρμογή θα σας παραπέμνει σε ένα νέο παράθυρο (δείτε σχήμα 5).

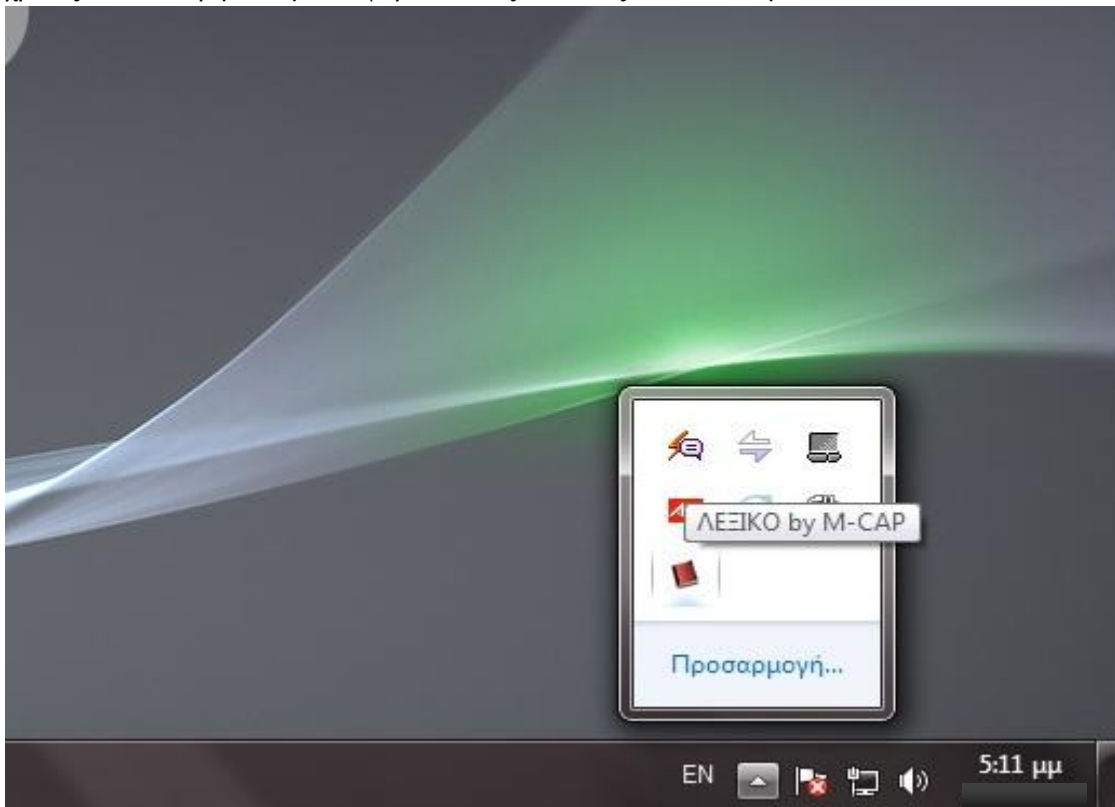


Εικόνα 15. παράθυρο γραμματικής

Θα ήταν πολύ καλό για την εφαρμογή στο μέλλον να εμπλουτιστεί με περισσότερο υλικό έτσι ώστε να μετατραπεί σε ένα πολύ καλό εκπαιδευτικό βοήθημα. Στο μενού "βοήθεια" μπορείτε να δείτε (και καλό θα ήταν να διαβάσετε) τις περειαίρω εξηγήσεις για την μορφή και τη δομή που έχουν τα στοιχεία των λημμάτων στα πλαίσια.

3.5.3 System tray

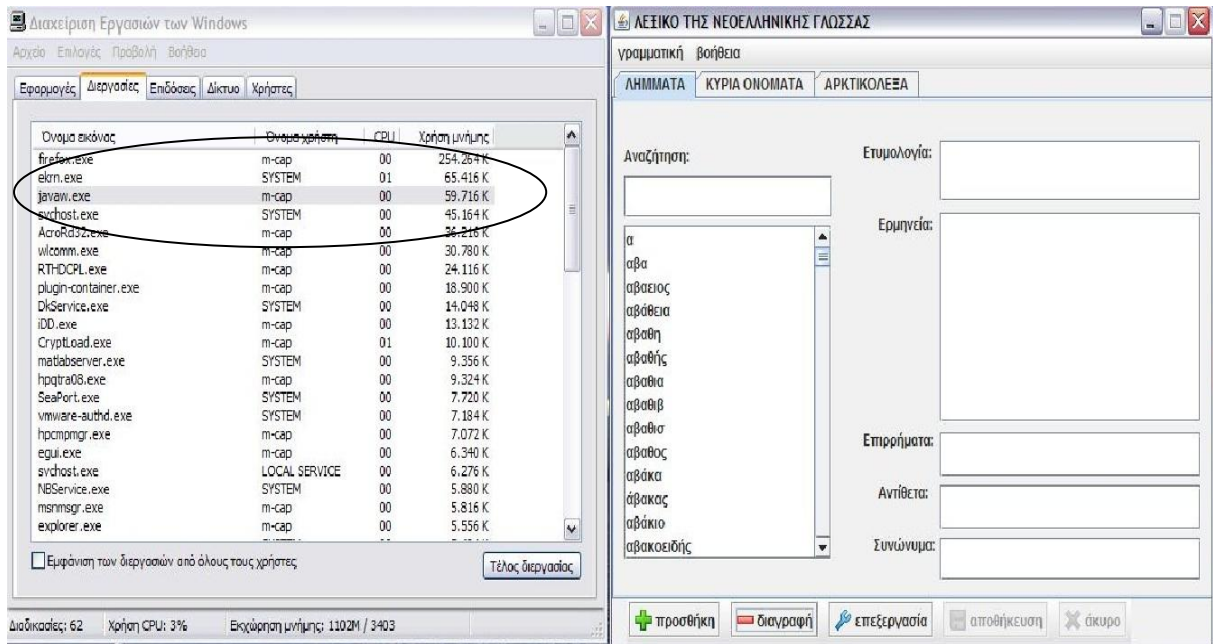
Επειδή τυχαίνει κάποια προγράμματα να τα χρησιμοποιούμε ανά τακτά χρονικά διαστήματα, θα ήταν δύσκολο να ανοίγουμε το ηλεκτρονικό λεξικό κάθε φορά που θα θέλουμε να το χρησιμοποιούμε, να φορτώνει ολόκληρη η βάση δεδομένων για να πάρουμε μια πληροφορία και μετά να το κλείνουμε. Έτσι λοιπόν μια υπηρεσία που δίνει η εφαρμογή είναι ότι μπορεί να ελαχιστοποιείται στο system tray του υπολογιστή μας κάνοντας έτσι ταχύτερη την πρόσβαση σε αυτό χωρίς να χρειάζεται να περιμένουμε να φορτωθούν ξανά και ξανά τα δεδομένα.



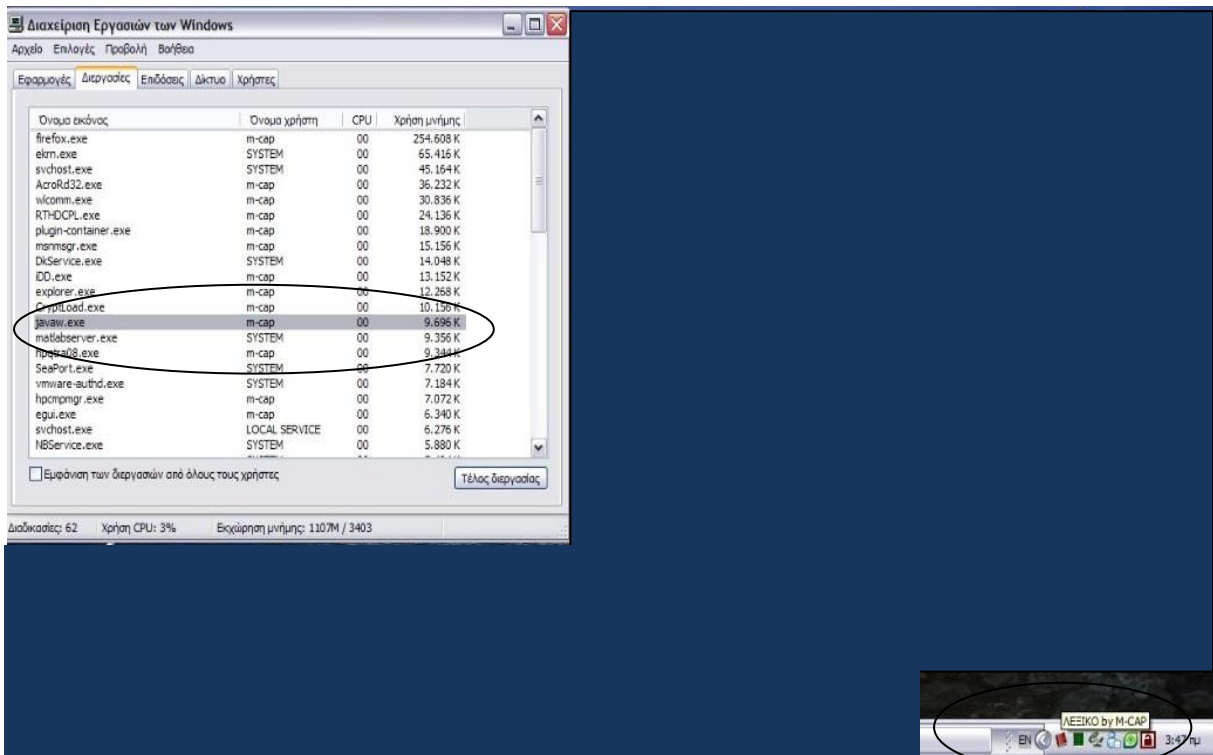
Εικόνα 16. εικονίδιο λεξικού στο παρασκήνιο

Μια σημαντική παρατήρηση που πρέπει να κάνουμε σχετικά με αυτό, είναι πως όταν τρέχει το λεξικό στο παρασκήνιο καταναλώνει τους ελάχιστους δυνατούς πόρους μνήμης έτσι ώστε να μην επιβαρύνετε η επίδοση του συστήματος σε άλλες διεργασίες. Κατά τη εκκίνηση και όταν το λεξικό χρησιμοποιείται φαίνεται να καταναλώνει πόρους της τάξεως των 60 MB. Όταν μπαίνει στο παρασκήνιο

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας



Εικόνα 17. πόροι κατά την εκκίνηση

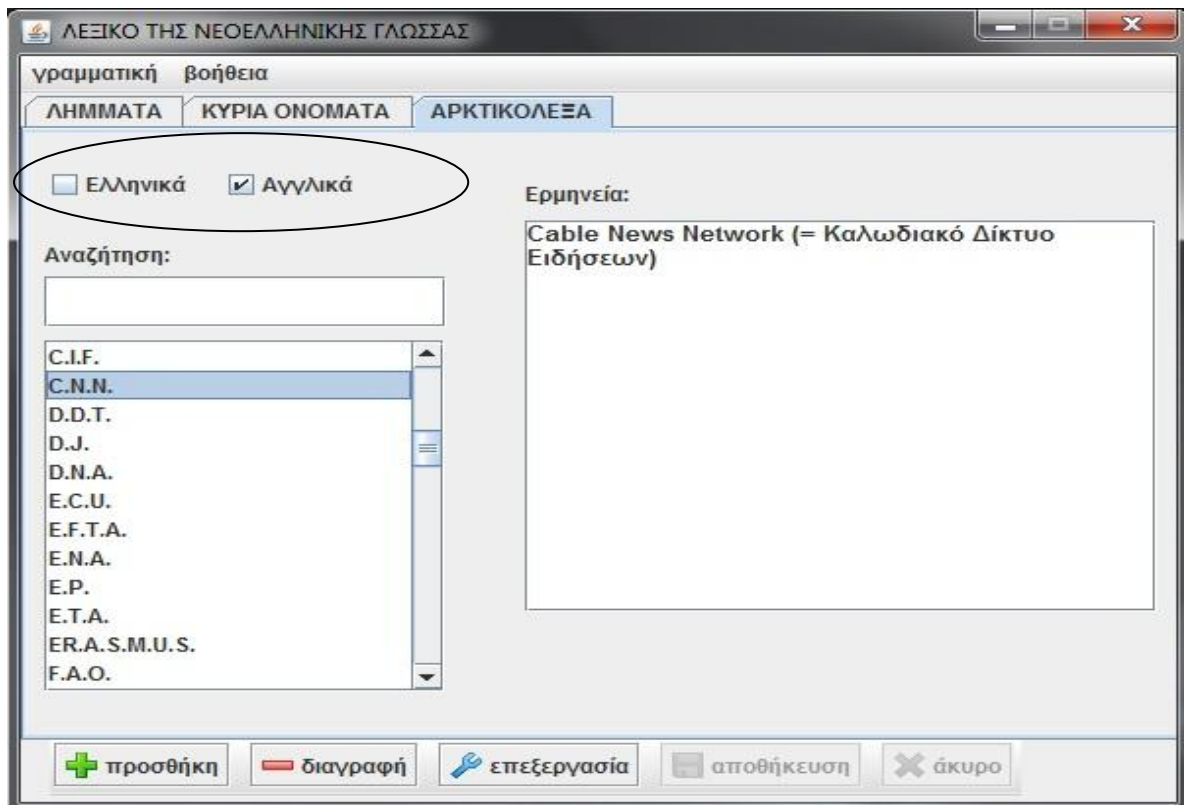


Εικόνα 18. κατανάλωση πόρων στο παρασκήνιο

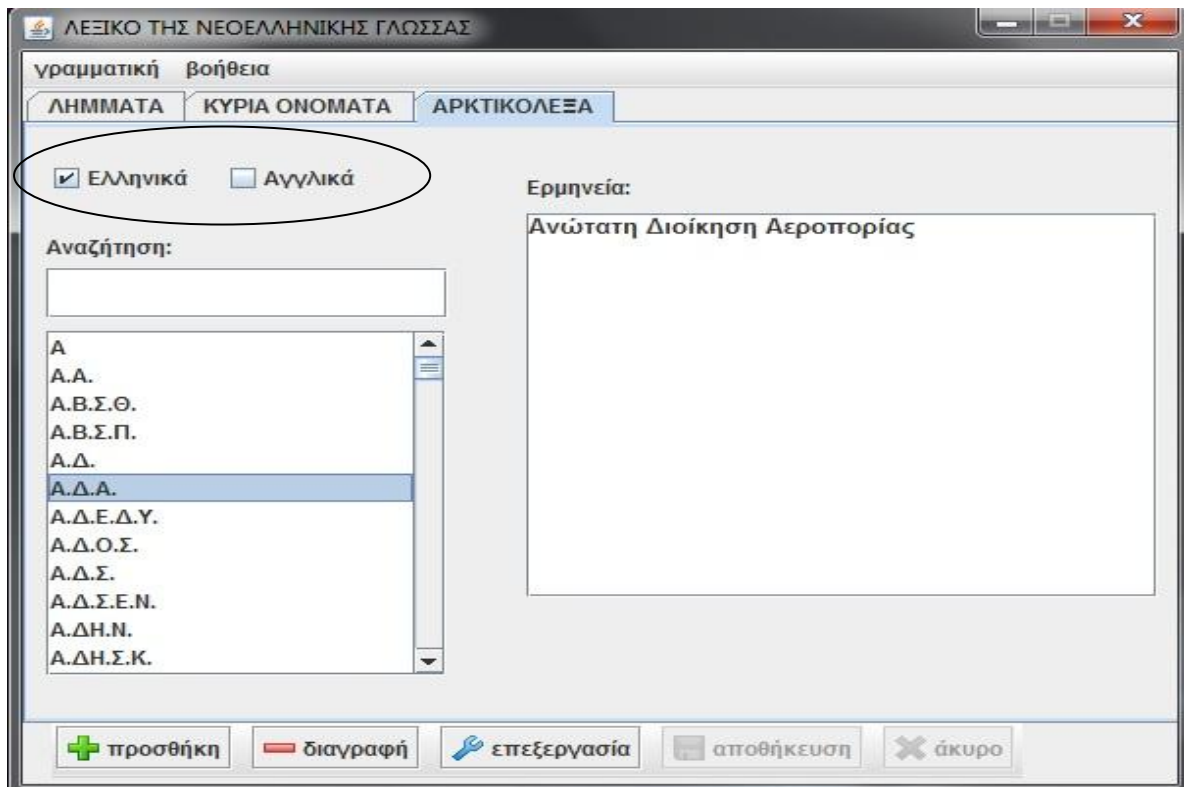
3.5.4 Χειρισμός αρκτικόλεξου

Όταν βρισκόμαστε στην καρτέλα Αρκτικόλεξα, ο χρήστης έχει τη δυνατότητα να επιλέξει μέσα από δύο πλαίσια ελέγχου αν θα εμφανίζεται το ελληνικό ή το αγγλικό αρκτικόλεξο. Μόνο μια επιλογή μπορεί να είναι διαθέσιμη κάθε φορά:

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας



Εικόνα 19. πλαίσιο ελέγχου για αγγλικό αρκτικόλεξο



Εικόνα 20. πλαίσιο ελέγχου για ελληνικό αρκτικόλεξο

Πτυχιακή εργασία: ηλεκτρονικό λεξικό της νεοελληνικής γλώσσας

3.5.5 Οδηγίες εκτέλεσης

Η εφαρμογή δεν χρειάζεται κάποιο είδος εγκατάστασης, εκτελείται απ' ευθείας πατώντας διπλό κλικ στο εικονίδιο **DICTIONARY** που βρίσκεται στον φάκελο. Ωστόσο υπάρχουν δυο παράμετροι που πρέπει να προσέξουμε πριν ξεκινήσουμε να κάνουμε οτιδήποτε.

- Ο υπολογιστής μας θα πρέπει να μπορεί να εκτελεί προγράμματα java. Αυτό θα συμβεί αρκεί να είναι εφοδιασμένος με την κατάλληλη JVM (java virtual machine). Εάν δεν είναι εγκατεστημένη, τότε ανατρέξτε στο φάκελο e-dictionary\setups\ και τρέξτε το αρχείο με το όνομα **jdk-6u5-windows-i586-p.exe** έτσι ώστε ο υπολογιστής να τρέχει java προγράμματα. Η έκδοση αυτή ισχύει για 32-bit συστήματα που τρέχουν σε οποιαδήποτε windows.
- ο υπολογιστής μας θα πρέπει να έχει εγκατεστημένη την access 2007. Σε διαφορετική περίπτωση εγκαθιστούμε το AccessDatabaseEngine.exe, ένα αρχείο που βρίσκεται στο φάκελο e-dictionary\setups. Με το AccessDatabaseEngine θα εγκατασταθούν όλα εκείνα τα συστατικά που χρειάζεται ο υπολογιστής σας για να χειριστεί τη βάση δεδομένων με το λεξικό. Το AccessDatabaseEngine_x64 τρέξτε το αν έχετε 64-bit σύστημα.

Αφού τσεκάρετε αυτά τα δύο σημεία είστε έτοιμοι για χρησιμοποιήσετε το λεξικό.

4. Αποτελέσματα

Τα ηλεκτρονικά λεξικά παρέχουν όλες τις πολύτιμες πληροφορίες (λεξιλογικές, σημασιολογικές και σχετικές με τη χρήση της γλώσσας) κυρίως για τον εκπαιδευτικό και τον μαθητή, τις οποίες παρέχουν και τα παραδοσιακά λεξικά, ενώ ταυτόχρονα τις καθιστούν περισσότερο εύχρηστες και ελκυστικές. Η χρήση τους είναι μια ενδιαφέρουσα πρόταση η οποία δεν είναι και τόσο διαδεδομένη εδώ στην Ελλάδα, κυρίως στο χώρο της εκπαιδευτικής τεχνολογίας.

4.1 Συμπεράσματα

Η πτυχιακή αυτή πέρα από το λεξικό που δημιουργήσαμε, μπορεί να εξυπηρετήσει και σκοπούς εκπαιδευτικούς για την εκμάθηση της γλώσσας java. Ο κώδικας περιέχει όλα εκείνα τα στοιχεία που εξετάζονται σε τέτοια μαθήματα και η χρήση του για την κατασκευή βήμα-βήμα αυτής της εφαρμογής από ένα αρχάριο προγραμματιστή θα του έδινε πολύ καλή εξοικείωση και γνώση της γλώσσας.

4.2 Επεκτάσεις

Πιθανές επεκτάσεις της εφαρμογής που θα μπορούσαν να γίνουν:

1. Σύστημα ενημέρωσης (update) της βάσης δεδομένων μέσω διαδικτύου.
2. Εμπλουτισμός της γραμματικής με πιο οργανωμένη δομή και πληροφορία
3. Ενσωμάτωση Online ερμηνευτή, έτσι ώστε όταν πχ. ο δείκτης του ποντικιού σταματήσει σε μια λέξη να εμφανίζεται σε παραθυράκι η ερμηνεία της.

5. Βιβλιογραφία - Πηγές

[1] Εισαγωγή στη java 2: Γιώργος Λιακέας

[2] Διεπαφή χρήστη – υπολογιστή, μια σύγχρονη προσέγγιση: Δημοσθένης Ακουμιανάκης

[3] Τεχνολογία λογισμικού: Εμμ. Α. Γιακουμάκης

[4] Επικοινωνία ανθρώπου – υπολογιστή: Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, 3η έκδοση

<http://www.scribd.com/doc/528991/-Java>

<http://el.wikipedia.org/wiki/Java>

www.java2s.com/Tutorial/Java/CatalogJava.htm

Τα αρχεία AccessDatabaseEngine και AccessDatabaseEngine_x64 διανέμονται δωρεάν από τη Microsoft στο site: www.microsoft.com/downloads

Το αρχείο jdk-6u5-windows-i586-p διανέμεται δωρεάν στο site: www.java.com/en/download/index.jsp

Παράρτημα

Παρουσίαση πτυχιακής σε Power Point Slides

**ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ
ΛΕΞΙΚΟΥ ΤΗΣ ΝΕΟΕΛΛΗΝΙΚΗΣ
ΓΛΩΣΣΑΣ**

e-Dictionary

Καπαρουνάκης Μιχάλης Α.Μ 1474

1/11

ΕΙΣΑΓΩΓΗ

- σκοπός αυτής της πτυχιακής είναι η σχεδίαση και δημιουργία ενός ψηφιακού λεξικού
- χρήσιμο βοήθημα για όλους όσους χρησιμοποιούν Η/Υ, από τον μαθητή του σχολείου μέχρι τον επιστήμονα
- αναπτύχθηκε σε περιβάλλον ανάπτυξης εφαρμογών blue - j με χρήση της γλώσσας java
- σαν επιπλέον βοήθημα χρησιμοποιήθηκε το JFormDesigner

2/11

Java

• Η java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε για να δώσει λύσεις σε μία σειρά πρακτικών προβλημάτων που εμφανίζονται σε όλες τις ήδη υπάρχουσες γλώσσες.

• Χαρακτηριστικά γλώσσας java:

- Απλή
- Αντικειμενοστραφής
- Έχουσα δικτυακή γνώση
- Μεταφράσιμη
- Εύρωστη
- Ασφαλής
- Ανεξάρτητη αρχιτεκτονικής
- Μεταφέρσιμη
- Υψηλής απόδοσης
- Πολυνηματική
- Δυναμική γλώσσα

3/11

Java

- **Η έννοια της τάξης (class)**
Μία τάξη (class) είναι ένα καλούπι από το οποίο δημιουργούνται αντικείμενα.

- **Η έννοια του αντικειμένου (object)**
Ένα αντικείμενο δεν είναι παρά μία περίπτωση (instance) μιας τάξης.

- **Κληρονομικότητα (Inheritance)**
Μία τάξη Java μπορεί να επεκταθεί (extend) από μία άλλη τάξη. Αυτή η άλλη τάξη ονομάζεται υποτάξη της αρχικής που είναι η υπερτάξη της.

- **Πολυμορφισμός (polymorphism)**
Ο πολυμορφισμός μας δίνει το δικαίωμα να διατηρούμε δομές δεδομένων με αντικείμενα μιας υπερτάξης τα οποία μπορεί να είναι στη πραγματικότητα αντικείμενα των υποτάξεων και να διατρέχουμε αυτή τη δομή καλώντας τη κοινή μέθοδο.

4/11

Τεχνολογίες υλοποίησης

Blue-j: Το blue-j είναι ένα περιβάλλον ανάπτυξης java προγραμμάτων που σχεδιάστηκε ειδικά για να υποστηρίξει την εκπαίδευση νέων προγραμματιστών στην Java.

JFormDesigner: δίνει τη δυνατότητα να σχεδιάζουμε πολύ εύκολα ένα γραφικό περιβάλλον επικοινωνίας (GUI).

Γιατί JFormDesigner;

- Απλός editor
- Δεν απαιτεί τεράστιο χώρο στο σκληρό δίσκο
- Πολύς 'ελαφρύς' όταν δουλεύουμε σε αυτόν

5/11

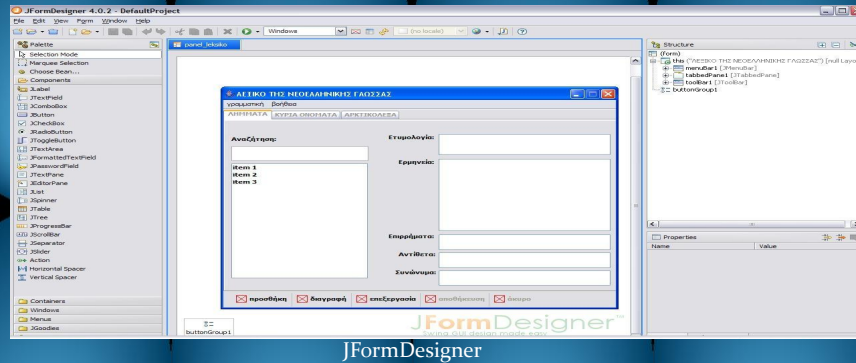
Απαιτήσεις λογισμικού

- η διαχείριση του περιεχομένου να γίνεται εύκολα και απλά
- να μην απαιτείται συντήρηση ούτε και σύνδεση με δίκτυο
- να συνεργάζεται με βάση δεδομένων που διαθέτουμε
- δυνατότητες όπως προσθήκη, διαγραφή και επεξεργασία των λημμάτων του λεξικού
- δυνατότητα αναζήτησης ενός λήμματος
- να εμφανίζεται στη γραμμή εργασιών του Η/Υ
- δεν θα χρειάζεται κάποια εγκατάσταση

6/11

Βασική σχεδίαση

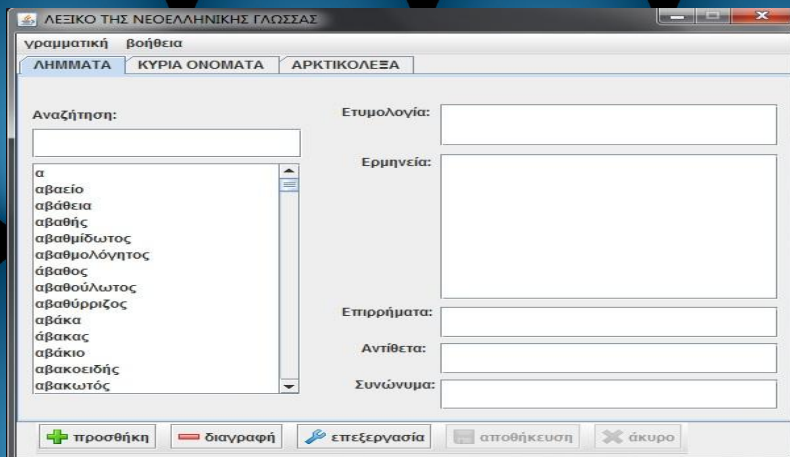
- η βασική φόρμα του λεξικού έχει μια ομοιομορφία όσο αναφορά στην διάταξη



7/11

λεξικό

Βασική μορφή του λεξικού:



8/11

λεξικό

Οι κατηγορίες ΛΗΜΜΑΤΑ, ΚΥΡΙΑ ΟΝΟΜΑΤΑ και ΑΡΚΤΙΚΟΛΕΞΑ είναι οργανωμένες σε ξεχωριστές καρτέλες

Δυνατότητες λεξικού

- Προσθήκη, διαγραφή, επεξεργασία, αποθήκευση
- **System tray**: η εφαρμογή μπορεί να ελαχιστοποιείται στο system tray του υπολογιστή μας κάνοντας έτσι ταχύτερη την πρόσβαση σε αυτό
- πόροι στο παρασκήνιο της τάξεως των 10 MB



9/11

λεξικό

-Τα ηλεκτρονικά λεξικά παρέχουν όλες τις πολύτιμες πληροφορίες κυρίως για τον εκπαιδευτικό και τον μαθητή

-Η χρήση τους είναι μια ενδιαφέρουσα πρόταση η οποία δεν είναι και τόσο διαδεδομένη εδώ στην Ελλάδα, κυρίως στο χώρο της εκπαιδευτικής τεχνολογίας.

-εφαρμογή μπορεί να εξυπηρετήσει και σκοπούς εκπαιδευτικούς για την εκμάθηση της γλώσσας java

Πιθανές επεκτάσεις :

- Σύστημα ενημέρωσης (update) της βάσης δεδομένων μέσω διαδικτύου.
- Εμπλουτισμός της γραμματικής με πιο οργανωμένη δομή και πληροφορία
- Ενσωμάτωση Online ερμηνευτή

10/11

