



ΤΕΙ Κρήτης
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πτυχιακή Εργασία
Πληροφοριακό Σύστημα Αντιπροσωπείας
Αυτοκινήτων**

Νάκου Ασημίνα

Επιβλέπων Καθηγητής: Αίβαλης Κωστής

Ημερομηνία Παρουσίασης: 2/12/2016

Ευχαριστίες

Με την εκπόνηση της πτυχιακής αυτής με θέμα Εφαρμογή Πληροφοριακού Συστήματος Αντιπροσωπείας Αυτοκινήτων θα ήθελα να ευχαριστήσω τον επιβλέπωντα καθηγητή μου κ. Αϊβαλή Κωστή, για την ανάθεση της πτυχιακής αλλά και την ευκαιρία που μου έδωσε. Τέλος να ευχαριστήσω την μητέρα μου για όλη τη στήριξη που μου παρείχε όλα τα χρόνια της φοίτησής μου.

Abstract

The thesis refers to the implementation of an application for the Information System of a car dealership manager. So, this implementation will require the connection of the application to our database and will be installing a web server, the XAMPP to make the connection with phpMyAdmin. Also, the following programming languages will be used is Java using Java Swing. Then, in order to store data, SQL database will be needed. This application will keep customers' personal information such as name, first name, surname, IRS, VAT number, telephone, address and city. As a second tab will keep information about the cars which customer buys. Information will be kept and for that, model, year, cubic horses, color, gearbox, drive and doors. It can display a window data of all customers and individually according to the customer code and will have options to store a new client, deleting, cleaning and window closing.

Σύνοψη

Η πτυχιακή εργασία αυτή με θέμα τη δημιουργία μιας εφαρμογής για τη Διαχείριση Πληροφοριακού Συστήματος μιας Αντιπροσωπείας Αυτοκινήτων. Έτσι, για την υλοποίηση αυτή θα χρειαστεί η σύνδεση της εφαρμογής με τη βάση δεδομένων μας και για αυτό θα γίνει η εγκατάσταση ενός web server, του Xampp για να γίνει η σύνδεση με το phpMyAdmin. Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί θα είναι η Java με τη χρήση Java Swing. Η εφαρμογή αυτή θα κρατάει προσωπικά στοιχεία των πελατών όπως επωνυμία πελάτη, επώνυμο, εφορία, ΑΦΜ, τηλέφωνο και διεύθυνση και πόλη. Σα δεύτερη καρτέλα θα κρατάει στοιχεία για τα αυτοκίνητα που θα αγοράζει ο πελάτης. Πληροφορίες που θα κρατηθούν για την καρτέλα αυτή θα είναι μάρκα, μοντέλο, έτος, κυβικά, ίπποι, χρώμα, σασμάν, κίνηση και πόρτες. Στη συνέχεια, για την αποθήκευση των δεδομένων χρειάζεται η γλώσσα υπολογιστών στις βάσεις δεδομένων SQL. Θα μπορεί να εμφανίζει σε παράθυρο τα στοιχεία όλων των πελατών καθώς και μεμονωμένα βάσει του κωδικού πελάτη και θα έχει επιλογές για αποθήκευση νέου πελάτη, διαγραφή του, καθαρισμός και κλείσιμο παραθύρου.

Περιεχόμενα

Περίληψη **Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.**

9

Κεφάλαιο 1^ο Γλώσσα Προγραμματισμού Java 10

1.1 Ιστορία 10

12

12

12

13

14

15

16

17

18

18

Κεφάλαιο 2^ο Netbeans 19

2.1 Ιστορία 19

22

2.2.1 Καλύτερη υποστήριξη για τις τελευταίες Java τεχνολογίες 22

23

2.2.3 Εύκολη και Έξυπνη διαχείριση project 24

25

26

2.4 Δημιουργία project 27

Κεφάλαιο 3^ο Βάση Δεδομένων MySQL 34

3.1 Εισαγωγή 34

3.2 Ιστορική Αναδρομή της MySQL 34

38

38

38

4.2.1 Κλάση Cars 38

4.2.2 Κλάση database 39

4.2.3 Κλάση TextForm 39

40
4.2.5 Κλάση carsTab40
4.2.6 Κλάση ShowCars41
4.2.7 Κλάση MyDialog41
41
42
49
4.4.1 Κλάση Cars49
4.4.2 Κλάση database49
4.4.3 Κλάση TextForm50
4.4.3.1 Constructor51
4.4.3.2 getText51
4.4.3.3 actionPerformed52
4.4.3.4 notIsNumber53
4.4.3.5 checkValues53
4.4.3.6 getCity54
55
4.4.5 Κλάση carsTab55
4.4.5.1 Constructor56
4.4.5.2 getText56
4.4.5.3 actionPerformed57
4.4.5.4 notIsNumber58
4.4.5.5 checkValues58
4.4.5.6 getCity59
4.4.5.7 showSelected59
4.4.6 Κλάση ShowCars59
4.4.7 Κλάση MyDialog60
60
71
Κεφάλαιο 5ο Συμπεράσματα – Μελλοντική Εργασία77
77
78
80

Κατάλογος Εικόνων

Εικόνα 1: Ο Duke η μασκότ της Java (jag, 2010)10
Εικόνα 2: Java11
Εικόνα 3: Ανεξαρτησία Πλατφόρμας14
Εικόνα 4: Τρόπος εκτέλεσης της Java15
Εικόνα 5: Δημιουργία κατανεμημένων εφαρμογών17
Εικόνα 6: Η αρχική ομάδα του Netbeans19
Εικόνα 7: Υποστήριξη για τις τελευταίες Java τεχνολογίες23
Εικόνα 8: Δυνατότητες συντάκτη24
Εικόνα 9: Καλύτερη οργάνωση έργων25
Εικόνα 10: Εργαλεία βελτίωσης εφαρμογής26
Εικόνα 11: Εκδόσεις NetBeans27
Εικόνα 12: Αρχική οθόνη NetBeans28
Εικόνα 13: 1ο βήμα28
Εικόνα 14: 2ο βήμα29
Εικόνα 15: 3ο βήμα30
Εικόνα 16: 4ο βήμα - τελικό31
Εικόνα 17: Το παράθυρο projects31
Εικόνα 18: Το παράθυρο Source Editor32
Εικόνα 19: Το παράθυρο Navigator32
Εικόνα 20: Output προγράμματος33
Εικόνα 21: MySQL (Dybka, 2014)34
Εικόνα 22: Ιστορικό Ανάπτυξης της MySQL (Database Friends, 2014)36
Εικόνα 23: Πίνακας customers46
Εικόνα 24: Πίνακας cities46
Εικόνα 25: Πίνακας cars47
Εικόνα 26: Πίνακας brands48
Εικόνα 27: Πίνακας models48
Εικόνα 28: Πίνακας colors48
Εικόνα 29: Βασική φόρμα60
Εικόνα 30: Συμπλήρωση στοιχείων61
Εικόνα 31: Ολοκλήρωση προσθήκης πελάτη61
Εικόνα 32: Μήνυμα ότι υπάρχει ήδη ο κωδικός πελάτη62
Εικόνα 33: Όλοι οι πελάτες63
Εικόνα 34: Αποτέλεσμα αναζήτησης χωρίς κωδικό64
Εικόνα 35: Αποτέλεσμα αναζήτησης64
Εικόνα 36: Αναζήτηση κωδικού που δεν υπάρχει65
Εικόνα 37: Ολοκλήρωση ενημέρωσης65
Εικόνα 38: Προσπάθεια για διαγραφή χωρίς κωδικό66
Εικόνα 39: Διαγραφή πελάτη67
Εικόνα 40: Έλεγχος διαγραφής68
Εικόνα 41: Εμφάνιση αυτοκινήτων για αγορά69
Εικόνα 42: Πώληση αυτοκινήτου69
Εικόνα 43: Ενημέρωση καρτέλας πελάτη μετά από αγορά70
Εικόνα 44: Καρτέλα αυτοκίνητα70
Εικόνα 45: Εμφάνιση διαθέσιμων αυτοκινήτων71
Εικόνα 46: Εμφάνιση στοιχείων αυτοκινήτων71
Εικόνα 47: Αρχική οθόνη εγκατάστασης73
Εικόνα 48: 2ο βήμα εγκατάστασης73
Εικόνα 49: 3ο βήμα74

Εικόνα 50: 4ο βήμα εγκατάστασης75

Εικόνα 51: Ολοκλήρωση εγκατάστασης75

Εικόνα 52: Αρχική οθόνη Χαμpp76

Περίληψη

Ο άνθρωπος από τη στιγμή που ξεκίνησε να έχει εμπορικές δραστηριότητες, αναζητούσε βοήθεια τόσο στην εκτέλεση υπολογισμών όσο και στην τήρηση αρχείου και παραγγελιών.

Από τη στιγμή που εμφανίστηκαν οι πρώτες υπολογιστικές μηχανές, αξιοποιήθηκαν στο έπακρο για την εκτέλεση υπολογιστικών πράξεων και την απελευθέρωση του ανθρώπου από τις επίπονες και χρονοβόρες εργασίες.

Με την εμφάνιση των υπολογιστών των γλωσσών προγραμματισμού και των βάσεων δεδομένων, οι προγραμματιστές ξεκίνησαν την υλοποίηση προγραμμάτων για την κάλυψη των αναγκών σε διάφορους χώρους. Οι εφαρμογές αυτές χρησιμοποιήθηκαν και βελτίωσαν σημαντικά την εργασία των ανθρώπων και τα αποτελέσματά της.

Ένας κλάδος που χρησιμοποίησε τους υπολογιστές και τις εφαρμογές τους ήταν ο κλάδος των αντιπροσωπειών αυτοκινήτων. Στον κλάδο αυτό, ήταν απαραίτητη η χρήση Τεχνολογιών Πληροφορικής και Επικοινωνιών γιατί υπήρχαν αρκετά δεδομένα τα οποία έπρεπε να επεξεργάζονται ταχύτατα και να μπορεί εύκολα να αναζητηθεί πληροφορία σε αυτά.

Στην παρούσα πτυχιακή εργασία θα δημιουργήσουμε μια απλή εφαρμογή στη γλώσσα προγραμματισμού Java για μια αντιπροσωπία αυτοκινήτων.

Εισαγωγή

Σκοπός της παρούσας πτυχιακής είναι η δημιουργία με χρήση της γλώσσας προγραμματισμού Java και της βάσης δεδομένων MySQL μιας εφαρμογής για μια αντιπροσωπεία αυτοκινήτων.

Η εφαρμογή αυτή θα πρέπει να υποστηρίζει δυνατότητες όπως προσθήκης πελατών, αναζήτησης και εμφάνισης στοιχείων τους, διόρθωσης στοιχείων πελατών, διαγραφής πελατών και εμφάνισης όλων των πελατών στη βάση δεδομένων. Επίσης θα πρέπει να υποστηρίζει αντίστοιχες δυνατότητες και για τα αυτοκίνητα και δυνατότητα αγοράς αυτοκινήτου από πελάτη.

Στο 1^ο κεφάλαιο θα παρουσιάσουμε εν συντομία τη γλώσσα προγραμματισμού Java, την ιστορία της και τα χαρακτηριστικά που την κάνουν να ξεχωρίζει.

Στο κεφάλαιο 2 θα μελετήσουμε το NetBeans, το εργαλείο που χρησιμοποιήθηκε για τη συγγραφή της εργασίας. Θα δούμε την ιστορία του, τα πλεονεκτήματά του. Στη συνέχεια θα περιγράψουμε την εγκατάστασή του βήμα – βήμα καθώς και τον τρόπο δημιουργίας ενός project.

Στο κεφάλαιο 3 θα παρουσιάσουμε εν συντομία την ιστορική εξέλιξη της βάσης δεδομένων MySQL.

Στο κεφάλαιο 4 θα δούμε τα διαγράμματα κλάσεων της εφαρμογής μας, θα περιγράψουμε το σχεδιασμό της βάσης δεδομένων. Στη συνέχεια θα δούμε το σχεδιασμό και την υλοποίηση της εφαρμογής. Το επόμενο θέμα είναι η διαδικασία εκτέλεσης της εφαρμογής και τέλος ακολουθούν οι διαδικασίες εγκατάστασης των απαιτούμενων προγραμμάτων.

Το 5^ο Κεφάλαιο περιέχει τα συμπεράσματα και τις προτάσεις μας για μελλοντική εργασία.

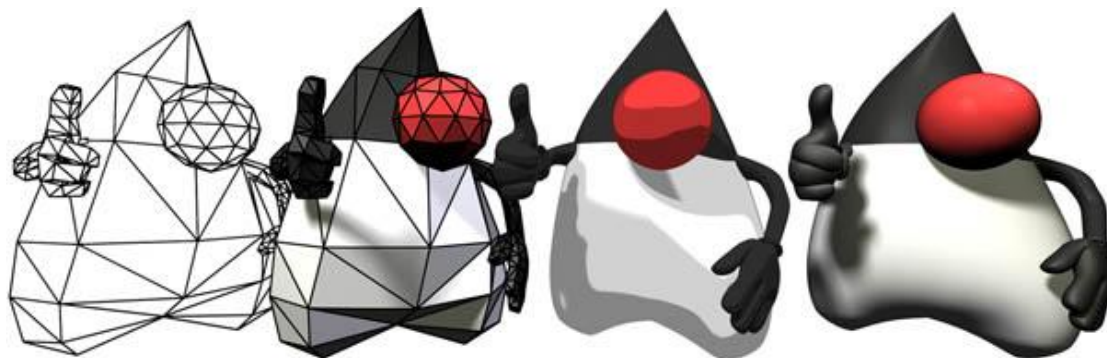
Κεφάλαιο 1^ο Γλώσσα Προγραμματισμού Java

Στο κεφάλαιο αυτό θα παρουσιάσουμε (εν συντομία) τη γλώσσα προγραμματισμού Java. Θα ξεκινήσουμε αναφέροντας λίγα λόγια για την ιστορία της και στη συνέχεια θα δούμε τα χαρακτηριστικά της που την κάνουν να ξεχωρίζει.

1.1 Ιστορία

Η ιστορία της Java ξεκινά από την Green Team της Sun Microsystems το 1991. Η Sun τη χρονιά εκείνη έκανε μια έρευνα για το μέλλον των υπολογιστικών εργασιών και προέκυψε σαν αποτέλεσμα ότι τα επόμενα χρόνια θα κυριαρχούσαν οι ευφυείς συσκευές. Για να καλύψει το συγκεκριμένο στόχο, δημιούργησε την Green Team. Τα μέλη της ομάδας (James Gosling, Mike Sheridan, και Patrick Naughton) ξεκίνησαν ένα επαναστατικό έργο, να αναπτύξουν μια γλώσσα για τις ψηφιακές συσκευές όπως set-top boxes, τηλεοράσεις κ.α.

Μετά από 18 μήνες εργασίας η ομάδα παρουσίασε μια συσκευή που ονομαζόταν Star7 (σαν τα PDA αλλά αρκετά μεγαλύτερη) η οποία εμφάνιζε μια φιγούρα με το όνομα Duke που αντιδρούσε στις προτροπές του χρήστη. Η φιγούρα αυτή σήμερα έχει γίνει μασκότες της γλώσσας προγραμματισμού Java.



Εικόνα 1: Ο Duke η μασκότες της Java (jag, 2010)

Η συσκευή αυτή βασιζόταν σε μια νέα γλώσσα προγραμματισμού η οποία είχε δημιουργηθεί από τον James Gosling και ονομάστηκε αρχικά Greentalk με επέκταση αρχείου gt. Στη συνέχεια μετονομάστηκε σε Oak, επειδή υπήρχε μια βελανιδιά έξω από το παράθυρο του γραφείου του και η βελανιδιά απεικονίζει τη δύναμη.

Η συγκεκριμένη γλώσσα διατηρούσε μεγάλη συγγένεια με τη C++ (την οποία δοκίμασε ο Gosling για γλώσσα προγραμματισμού μικροσυσκευών, αλλά την απέρριψε ως ακατάλληλη). Είχε όμως πιο έντονο αντικειμενοστρεφή χαρακτήρα και χαρακτηριζόταν από την απλότητά της. Λίγο καιρό αργότερα η ομάδα ανακάλυψε ότι το όνομα Oak είναι ήδη κατοχυρωμένο αναγκάστηκε να ψάξει για ένα νέο όνομα. Οι προτάσεις περιλάμβαναν τα ονόματα: "dynamic", "revolutionary", "Silk", "jolt", "DNA". Ήθελαν κάτι που αντικατοπτρίζει την ουσία της τεχνολογίας: επαναστατική, δυναμική, ζωντανή, δροσερή, μοναδική και ταυτόχρονα με εύκολο και διασκεδαστικό όνομα. Από αυτά δύο επιλέχθηκαν το Java και το Silk. Επειδή το Java ήταν μοναδικό, προτιμήθηκε. Java είναι ένα νησί στην Ινδονησία στο οποίο έγινε η πρώτη παραγωγή καφέ (JavaTPoint, n.a).



Εικόνα 2: Java

Μέχρι το 1993 η Java δεν είχε την αποδοχή που όλοι περίμεναν. Όμως με την έκρηξη του World Wide Web και την κατανόηση από την Sun ότι το προϊόν που είχαν σχεδιάσει μπορούσε να χρησιμοποιηθεί στο νέο περιβάλλον για να παρέχει δυναμικό περιεχόμενο, επιτάχυνε τις εξελίξεις.

Το 1996 η Sun παρουσιάζει, επίσημα την πρώτη έκδοση της Java. Η έκδοση αυτή θα εμπλουτιστεί με νέες προσθήκες (ενσωμάτωση Swing

graphical API κ.λ.π) και το 1998 θα παρουσιαστεί η δεύτερη έκδοση. Στη συνέχεια θα ακολουθήσουν πολλές νέες εκδόσεις και update στις εκδόσεις αυτές. Σήμερα είμαστε στην έκδοση 8 και στο update 40 της συγκεκριμένης έκδοσης.

1.2 Χαρακτηριστικά της Java

Υπάρχει μια μεγάλη λίστα από χαρακτηριστικά που κάνουν την Java να ξεχωρίζει ανάμεσα στις άλλες γλώσσες προγραμματισμού. Στην ενότητα αυτή θα παρουσιάσουμε τα πιο σημαντικά.

1.2.1 Απλή

Σύμφωνα με την Sun, η γλώσσα Java είναι απλή, διότι:

- Η σύνταξη βασίζεται στη C++ (έτσι είναι ευκολότερη για τους προγραμματιστές να τη μάθουν μετά από τη C++).
- Έχουν αφαιρεθεί πολλά στοιχεία που προκαλούσαν σύγχυση ή χρησιμοποιούνταν σπάνια όπως explicit pointers, operator overloading κ.λπ.
- Δεν χρειάζεται να αφαιρεθούν unreferenced αντικείμενα γιατί υπάρχει αυτόματος Garbage Collector της Java.

1.2.2 Object Oriented

Αντικειμενοστραφής σημαίνει η οργάνωση του λογισμικού μας σαν ένα συνδυασμό διαφορετικών τύπων αντικειμένων που ενσωματώνουν δεδομένα και συμπεριφορές.

Ο Αντικειμενοστραφής προγραμματισμός είναι μια μεθοδολογία που

απλοποιεί την ανάπτυξη και συντήρηση του λογισμικού, παρέχοντας κάποιους κανόνες.

Οι βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού είναι:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

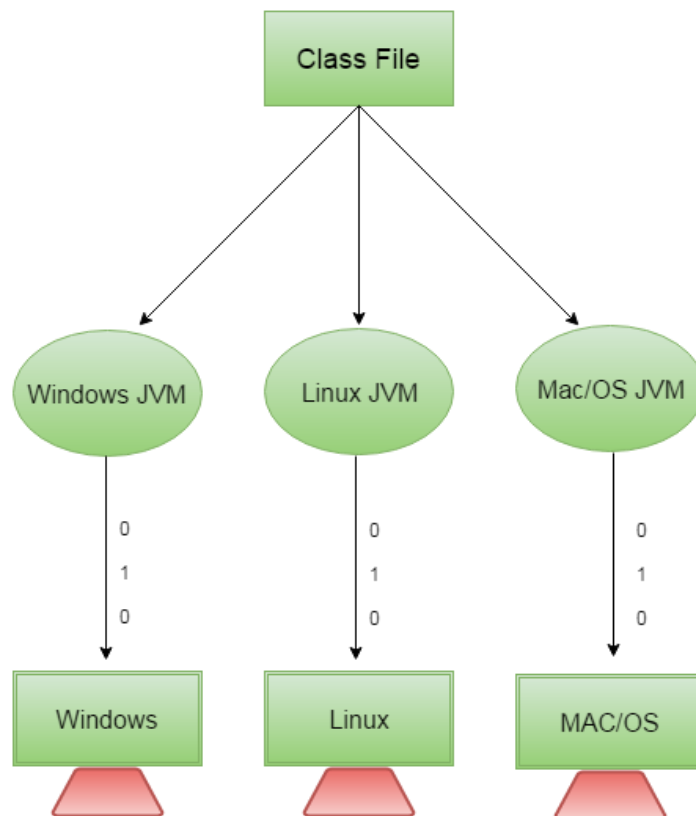
1.2.3 Ανεξάρτητη Πλατφόρμας

Μια πλατφόρμα είναι το υλικό ή το λογισμικό περιβάλλον στο οποίο τρέχει ένα πρόγραμμα. Υπάρχουν δύο τύποι πλατφορμών που βασίζονται σε λογισμικό και σε υλικό. Η Java παρέχει την πλατφόρμα που βασίζεται σε λογισμικό.

Η πλατφόρμα Java διαφέρει από τις άλλες πλατφόρμες, με την έννοια ότι είναι μια πλατφόρμα λογισμικού που τρέχει στην κορυφή άλλων πλατφορμών που βασίζονται σε υλικό. Αποτελείται από δύο στοιχεία:

- Runtime Environment
- API (Application Programming Interface)

Ο κώδικας Java μπορεί να τρέξει σε πολλές πλατφόρμες, π.χ. Windows, Linux, Sun Solaris, Mac / OS κλπ κώδικα Java μεταγλωττίζεται από τον compiler και μετατρέπεται σε bytecode. Αυτό το bytecode είναι μια πλατφόρμα-ανεξάρτητη κώδικα, επειδή μπορεί να τρέξει σε πολλαπλές πλατφόρμες δηλαδή Write Once and Run Anywhere (WORA).

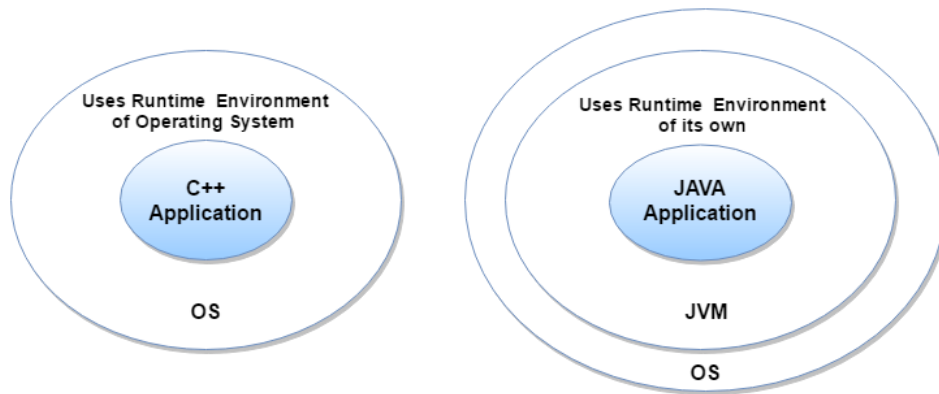


Εικόνα 3: Ανεξαρτησία Πλατφόρμας

1.2.4 Ασφαλής

Η Java είναι ασφαλής γιατί:

- Δεν περιλαμβάνει explicit pointer
- Εκτελείται σε ένα virtual machine sandbox



Εικόνα 4: Τρόπος εκτέλεσης της Java

Προσθέτει ασφάλεια

- Με τον Classloader που ξεχωρίζει τις κλάσεις που ανήκουν στο package σε αυτές που ανήκουν στο τοπικό file system και σε αυτές από πηγές του δικτύου.
- Με τον ελεγκτή bytecode που ελέγχει τον κώδικα για τμήματά του που μπορεί να παραβιάζουν τα δικαιώματα πρόσβασης σε αντικείμενα.
- Με τον security manager που καθορίζει σε ποιους πόρους μπορεί να έχει πρόσβαση μια κλάση.

1.2.5 Στιβαρή

Η Java διαθέτει μια σειρά από χαρακτηριστικά τα οποία την κάνουν πιο στιβαρή γλώσσα, σε σύγκριση με άλλες γλώσσες προγραμματισμού. Ορισμένα από αυτά τα χαρακτηριστικά είναι:

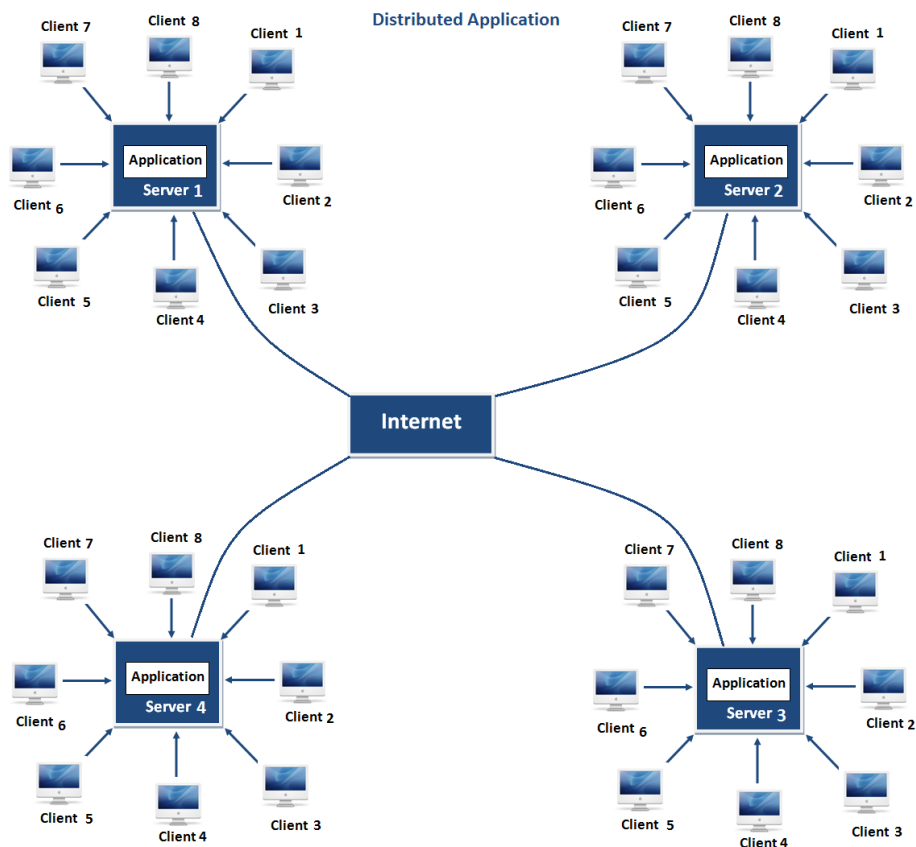
- Αυστηρή τυποποίηση: Η Java είναι αυστηρή ως προς την εγκυρότητα του κώδικα και ο μεταγλωττιστής δεν επιτρέπει να γίνουν πολλά από τα λάθη που ταλαιπωρούν τους προγραμματιστές σε άλλες γλώσσες.
- Αυτόματη διαχείριση μνήμης: Η διαχείριση της μνήμης γίνεται αυτόματα από τη JVM, με αποτέλεσμα οι προγραμματιστές να

μην ασχολούνται με πολύπλοκες διαδικασίες που συχνά οδηγούν σε λάθη (όπως η χρήση pointer).

- Χρησιμοποιεί συλλογή απορριμμάτων. Ο συλλέκτης απορριμμάτων (garbage collector) είναι ένα τμήμα της JVM που παρακολουθεί τη μνήμη και καθορίζει αν υπάρχουν απορρίμματα που πρέπει να αφαιρεθούν. Με προηγούμενες γλώσσες προγραμματισμού (π.χ. C++), ο προγραμματιστής ήταν υπεύθυνος να αφαιρεί από τη μνήμη τα αντικείμενα που δεν χρειαζόταν. Αυτό στη Java γίνεται αυτόματα.
- Παρέχει χειρισμό μηχανισμού εξαιρέσεων: Η Java επιτρέπει να ορίσουμε τύπους εξαιρέσεων, οι οποίοι δηλώνουν συγκεκριμένες συνθήκες σφαλμάτων. Αυτές οι συνθήκες σφαλμάτων μπορούμε να τις χειριστούμε μέσα από τα προγράμματα ή να τις παρακάμψουμε (Greanier, 2005).

1.2.6 Δημιουργία κατανεμημένων εφαρμογών

Τα ευρέως χρησιμοποιούμενα πρωτόκολλα όπως το HTTP και FTP έχουν αναπτυχθεί σε Java. Οι προγραμματιστές διαδικτυακών εφαρμογών, μπορούν να καλούν μεθόδους των πρωτοκόλλων αυτών και μπορούν να αποκτήσουν πρόσβαση σε αρχεία από οποιοδήποτε απομακρυσμένο μηχάνημα στο διαδίκτυο από τη σύνταξη προγραμμάτων στο τοπικό σύστημά τους.



Εικόνα 5: Δημιουργία κατανεμημένων εφαρμογών

1.2.7 Πολυνηματική

Μια θεμελιώδης έννοια του προγραμματισμού ηλεκτρονικών υπολογιστών είναι η ιδέα της διαχείρισης περισσότερων από μιας εργασιών σε μια στιγμή. Στο πλαίσιο ενός προγράμματος αυτά τα ξεχωριστά τμήματα ονομάζονται νήματα και η γενική ιδέα ονομάζεται multithreading.

Η Virtual Machine της Java επιτρέπει σε μια εφαρμογή να έχει πολλαπλά threads εκτέλεσης που εκτελούνται ταυτόχρονα. Κάθε νήμα έχει μια προτεραιότητα. Τα νήματα με μεγαλύτερη προτεραιότητα εκτελούνται κατά προτίμηση σε σχέση με τα νήματα με χαμηλότερη προτεραιότητα. Όταν μια Java Virtual Machine ξεκινά, συνήθως υπάρχει ένα απλό νήμα. Η Virtual Machine Java συνεχίζει να εκτελεί τα νήματα μέχρι η μέθοδος εξόδου της τάξης Runtime να κληθεί και ο security manager να επιτρέψει τη λειτουργία εξόδου.

1.2.8 Φορητή

Ο κώδικας Java είναι φορητός. Η φορητότητα ήταν ένας σημαντικός στόχος του σχεδιασμού της Java, έτσι ώστε αν οι νέες αρχιτεκτονικές (λόγω υλικού αλλά και λογισμικού - λειτουργικού συστήματος), αναπτύσσονταν, το περιβάλλον java θα μπορούσε να μεταφερθεί σε αυτές.

Στη java, όλοι οι primitive τύποι (integer, long, float, double κ.λπ.) έχουν καθορισμένο μέγεθος, ανεξάρτητα από το υλικό ή το λογισμικό στα οποία εκτελείται το πρόγραμμα. Αυτό έρχεται σε άμεση αντίθεση με γλώσσες όπως η C και η C ++ που αφήνουν το μέγεθος των primitive τύπων στον μεταγλωττιστή.

Επιπλέον, η Java είναι φορητή, γιατί ο ίδιος ο μεταγλωττιστής είναι γραμμένος σε Java.

1.2.9 Υψηλή απόδοση

Η Java είναι πιο γρήγορη σε σύγκριση με τις παραδοσιακές interpreted γλώσσες, γιατί ο bytecode είναι «κοντά» σε εγγενή κώδικα. Εξακολουθεί όμως να είναι κάπως πιο αργή σε σχέση με τις γλώσσες που μεταγλωττίζονται (π.χ., C++) – αν και παρατηρούνται αρκετές βελτιώσεις με την εξέλιξη της Java Virtual Machine (JavaTPoint, n.a).

Κεφάλαιο 2^ο Netbeans

2.1 Ιστορία

Το NetBeans ξεκίνησε ως ένα φοιτητό project (αρχικά ονομαζόταν Xelfi) στην Τσεχοσλοβακία (τώρα η Τσεχική Δημοκρατία), το 1996. Ο στόχος ήταν να υλοποιηθεί ένα Java IDE (Ολοκληρωμένο Περιβάλλον Ανάπτυξης) σε Java, σαν την Delphi. Το όνομά του ήταν Xelfi και ήταν ο πρώτος Java IDE γραμμένο σε Java. Οι πρώτες προ-εκδόσεις έγιναν το 1997.

Στην εικόνα που ακολουθεί μπορούμε να δούμε την ομάδα μαθητών που αποτέλεσε την αρχική ομάδα του NetBeans:



Εικόνα 6: Η αρχική ομάδα του Netbeans

Το έργο προσέλκυσε αρκετό ενδιαφέρον, με αποτέλεσμα οι φοιτητές μετά από την αποφοίτησή τους, αποφάσισαν ότι θα μπορούσαν να το προωθήσουν στην αγορά σαν εμπορικό προϊόν. Προσέλκυσαν πόρους από συγγενείς και φίλους και ίδρυσαν μια εταιρεία και ένα δικτυακό τόπο.

Λίγο αργότερα ήρθαν σε επαφή με ένα επιχειρηματία, τον Roman Stanek, ο οποίος είχε εμπλακεί σε πολλές startup επιχειρήσεις και έψαχνε μια καλή ιδέα για να επενδύσει. Συναντήθηκε με τους ιδρυτές, οι οποίοι τον έπεισαν και ξεκίνησαν να καταστρώνουν το επιχειρηματικό σχέδιο. Το σχέδιο αυτό περιλάμβανε την υλοποίηση JavaBeans components με ενεργοποιημένα χαρακτηριστικά για δικτυακές υπηρεσίες. Με αφορμή την υλοποίηση των JavaBeans, ο Jaroslav Tulach που σχεδίασε τη βασική αρχιτεκτονική του IDE, πρότεινε το όνομα NetBeans προσπαθώντας να περιγράψει τη λειτουργία των components, που θα διατίθονταν μέσω της πλατφόρμας.

Την άνοιξη του 1999, κυκλοφόρησε το NetBeans DeveloperX2, υποστηρίζοντας Swing. Οι βελτιώσεις απόδοσης που προέκυψαν με την κυκλοφορία του JDK 1.3, το φθινόπωρο του 1999, έκανε NetBeans μια βιώσιμη επιλογή για τα εργαλεία ανάπτυξης. Μέχρι το καλοκαίρι του 1999, η ομάδα εργαζόταν σκληρά επανασχεδιάζοντας την αρχιτεκτονική του DeveloperX2 σε μια πιο σπονδυλωτή μορφή που αποτελεί τη βάση του λογισμικού σήμερα. Στην πορεία, διάφοροι άνθρωποι ξεκίνησαν την υλοποίηση και τη δημιουργία εφαρμογών με τη χρήση του NetBeans.

Το καλοκαίρι του 1999 η Sun Microsystems ήθελε καλύτερα εργαλεία ανάπτυξης της Java και είχε δείξει το ενδιαφέρον της για την πλατφόρμα NetBeans. Το όνειρο της αρχικής ομάδας γινόταν πραγματικότητα, το NetBeans θα γινόταν η ναυαρχίδα του ίδιου του κατασκευαστή της Java. Το φθινόπωρο είχε επέλθει η συμφωνία μεταξύ της Sun και του NetBeans. Όμως η Sun είχε αποκτήσει άλλη μια εταιρεία, τη Forte και αποφάσισε να μετονομάσει το NetBeans σε Forte για Java, προσωρινά το όνομα NetBeans έφευγε από το προσκήνιο.

Κατά τη διάρκεια της εξαγοράς, νέοι προγραμματιστές που είχαν εργαστεί σε έργα ανοικτού κώδικα πρότειναν τη μετατροπή του NetBeans σε ανοικτό κώδικα. Σε λιγότερο από έξι μήνες, είχε ληφθεί η απόφαση για μετατροπή του NetBeans σε ανοικτό κώδικα μέσω του δικτυακού τόπου netbeans.org τον Ιούνιο του 2000.

Όταν η Oracle απέκτησε Sun το 2010, το NetBeans έγινε μέρος της Oracle, η οποία συνεχίζει να το υποστηρίζει. Η Oracle επιδιώκει ενεργά νέους

προγραμματιστές να εργαστούν για την ομάδα του NetBeans και βλέπει το NetBeans IDE ως επίσημο IDE για την πλατφόρμα Java.

Ας δούμε τώρα ορισμένα από τα πιο βασικά χαρακτηριστικά των εκδόσεων του NetBeans:

- Έκδοση 3.2: η πρώτη έκδοση ανοικτού κώδικα.
- Έκδοση 3.5: έγιναν μεγάλα άλματα στην απόδοση και προβλέφθηκαν δοκιμές και διαδικασίες για αποφυγή παλινδρομήσεων στις επόμενες εκδόσεις.
- Έκδοση 3.6: το σύστημα παραθύρων και το φύλλο ιδιοτήτων έχουν επαναυλοποιηθεί και έχει απλοποιηθεί το user interface.
- Έκδοση 4: ήταν μια πλήρης αλλαγή του τρόπου λειτουργίας του IDE, παρουσιάστηκαν ανανεωμένες δομές που κρατούσαν το NetBeans πίσω σε σχέση με τον ανταγωνισμό.
- Έκδοση 4.1: πρόσθεσε νέες δυνατότητες και πλήρη υποστήριξη του J2EE.
- Έκδοση 5: παρουσιάζει ολοκληρωμένη υποστήριξη για ανάπτυξη μονάδων IDE και εφαρμογών πελάτη που βασίζονται στην πλατφόρμα NetBeans. Επίσης περιλαμβάνει έναν έξυπνο δημιουργό GUI (Matisse), νέα και επανασχεδιασμένη υποστήριξη CVS και υποστήριξη για τον Sun Application Server 8.2, WebLogic 9 και JBoss 4.
- Έκδοση 6: ήταν εστιασμένη στη βελτίωση της παραγωγικότητας με ένα νέο – επανασχεδιασμένο επεξεργαστή κειμένου με την ταυτόχρονα ενσωμάτωση εξωτερικών προϊόντων του NetBeans.
- Έκδοση 7: κυκλοφόρησε μαζί με το JDK 7. Η έκδοση 7.1 παρέχει εργαλεία επεξεργασίας για την εργασία με τις νέες γλωσσικές δομές του JDK 7, μαζί με υποστήριξη για JavaFX 2.0. Η έκδοση 7.2 έχει μια σειρά από βελτιώσεις στην απόδοση σε όλο τον IDE. Η έκδοση 7.3 παρέχει υποστήριξη για εφαρμογές HTML5.
- Έκδοση 8: παρέχει ενσωματωμένους τους GlassFish και Tomcat server. Επίσης παρέχει πλήρη υποστήριξη στις τεχνολογίες της Java 8 (Java

SE 8, Java SE Embedded 8, and Java ME Embedded 8). Επίσης παρέχονται μια σειρά από εργαλεία για ανάπτυξη HTML 5/Javascript έργων όπως Node.js, KnockoutJS, και AngularJS (Oracle, 2016a).

2.2 Πλεονεκτήματα

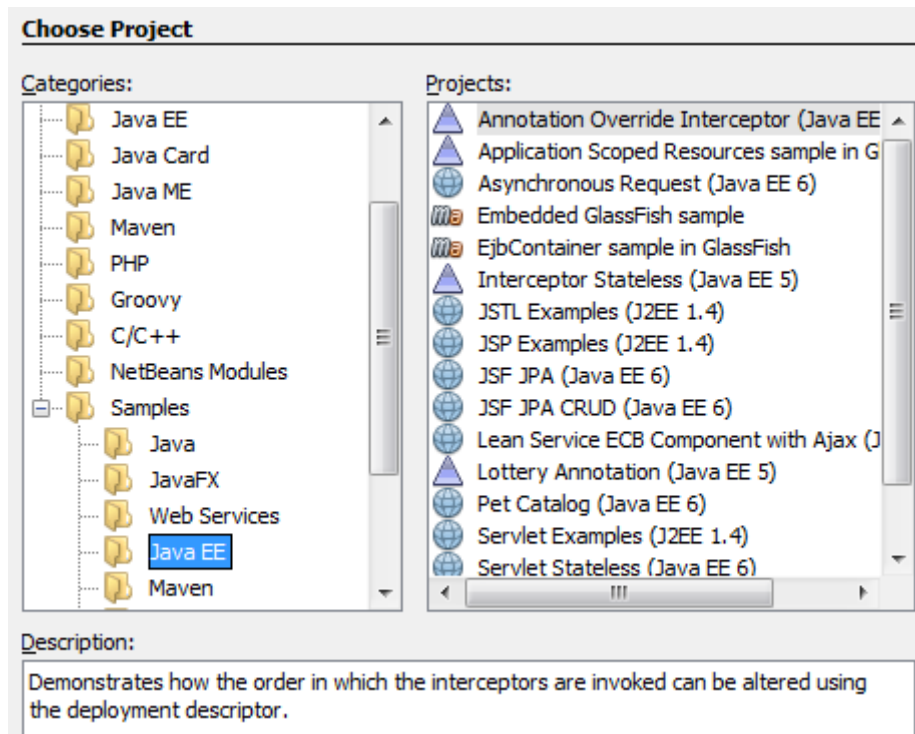
Στην ενότητα αυτή θα παρουσιάσουμε τα σημαντικότερα πλεονεκτήματα που παρουσιάζει το Netbeans σε σχέση με τον ανταγωνισμό

2.2.1 Καλύτερη υποστήριξη για τις τελευταίες Java τεχνολογίες

Το NetBeans είναι το επίσημο IDE για τη Java 8. Με τους επεξεργαστές κειμένου, τους αναλυτές κώδικα και μετατροπείς, μπορεί κάποιος γρήγορα και εύκολα να αναβαθμίσει τις εφαρμογές του ώστε να χρησιμοποιούν τις νέες γλωσσικές δομές της Java 8, όπως λειτουργικές δραστηριότητες και οι αναφορές σε μεθόδους.

Οι αναλυτές και μετατροπείς που παρέχονται βοηθούν στην αναζήτηση μέσα από πολλαπλές εφαρμογές ταυτόχρονα, ταιριάζοντας μοτίβα για τη μετατροπή σε νέα Java 8 γλωσσικές δομές.

Το NetBeans θέτει το πρότυπο για την ανάπτυξη και την υλοποίηση εφαρμογών με τη συνεχή βελτίωση του επεξεργαστή κειμένου της, τα πολλά χαρακτηριστικά και το ευρύ φάσμα εργαλείων, προτύπων και δειγμάτων που διαθέτει.

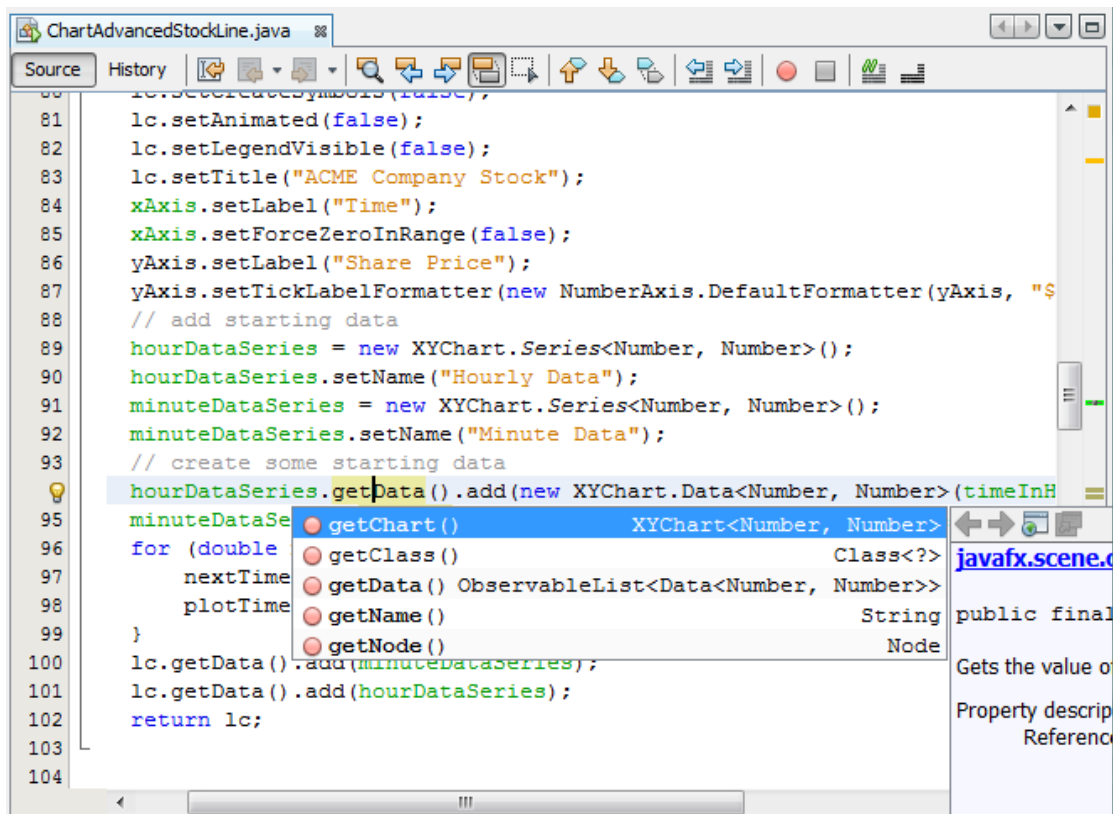


Εικόνα 7: Υποστήριξη για τις τελευταίες Java τεχνολογίες

2.2.2 Έξυπνη και Γρήγορη επεξεργασία κειμένου

Ένα IDE είναι κάτι πολύ περισσότερο από ένα πρόγραμμα επεξεργασίας κειμένου. Ο NetBeans Editor, ταιριάζει τις παρενθέσεις, τις λέξεις και τονίζει τον πηγαίο κώδικα συντακτικά και σημασιολογικά. Επίσης, ανιχνεύει σφάλματα κατά την πληκτρολόγηση και βοηθά με pop up για αυτόματη συμπλήρωση του κώδικα. Επιτρέπει στο χρήστη να κάνει refactor εύκολα τον κώδικα, με μια σειρά από εύχρηστα και ισχυρά εργαλεία, ενώ παρέχει επίσης πρότυπα κώδικα, συμβουλές κωδικοποίησης, και γεννήτριες παραγωγής κώδικα.

Ο συντάκτης υποστηρίζει πολλές γλώσσες από Java, C/C++, XML και HTML, PHP, Groovy, Javadoc, JavaScript και JSP. Επειδή ο συντάκτης είναι επεκτάσιμος, μπορείτε να προσθέσετε υποστήριξη για πολλές άλλες γλώσσες.

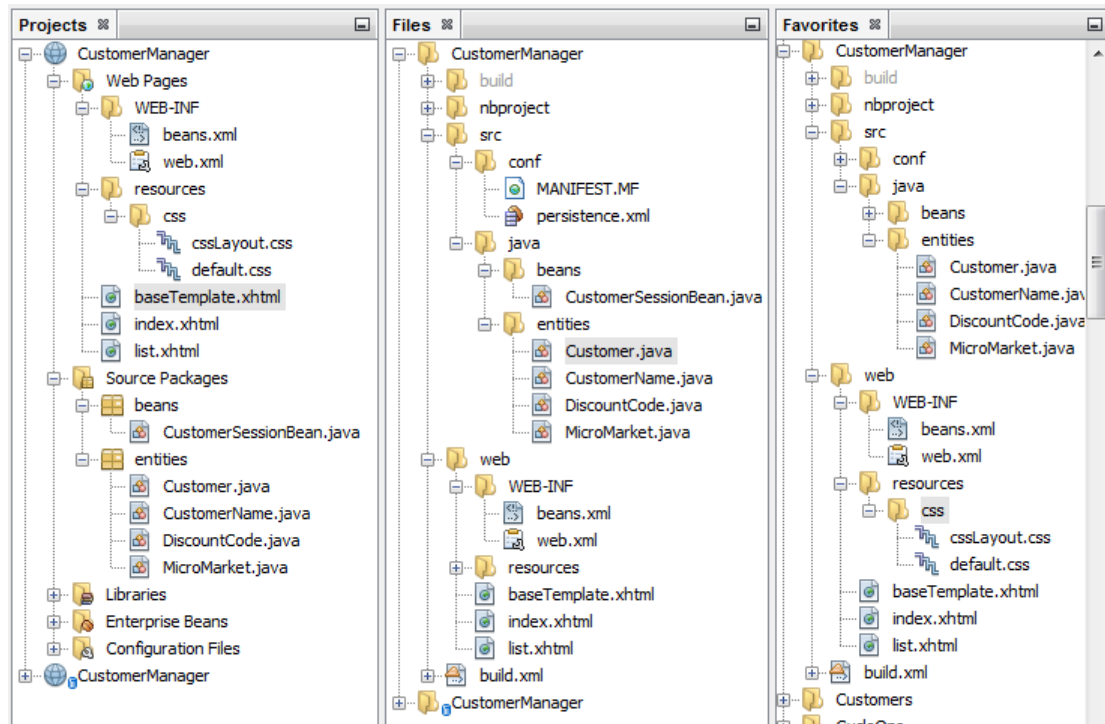


Εικόνα 8: Δυνατότητες συντάκτη

2.2.3 Εύκολη και Έξυπνη διαχείριση project

Το να κρατηθεί μια σαφή εικόνα των μεγάλων εφαρμογών, με χιλιάδες φακέλους και αρχεία, και εκατομμύρια γραμμές κώδικα, είναι ένα δύσκολο έργο. Ο NetBeans IDE παρέχει διαφορετικές προβολές των δεδομένων, από πολλαπλά παράθυρα του έργου με χρήσιμα εργαλεία για τη δημιουργία εφαρμογών και τη διαχείρισή τους αποτελεσματικά, επιτρέποντάς να εμβαθύνετε στα δεδομένα γρήγορα και εύκολα, ενώ σας δίνει εργαλεία εκδόσεων (versioning) μέσω της ενσωμάτωσης των εργαλείων Subversion, το Mercurial, Git.

Όταν καινούριοι προγραμματιστές ξεκινούν να συμμετάσχουν σε ένα ήδη υπάρχον έργο, μπορούν να κατανοήσουν τη δομή της εφαρμογής, γιατί ο κώδικας είναι καλά οργανωμένος.

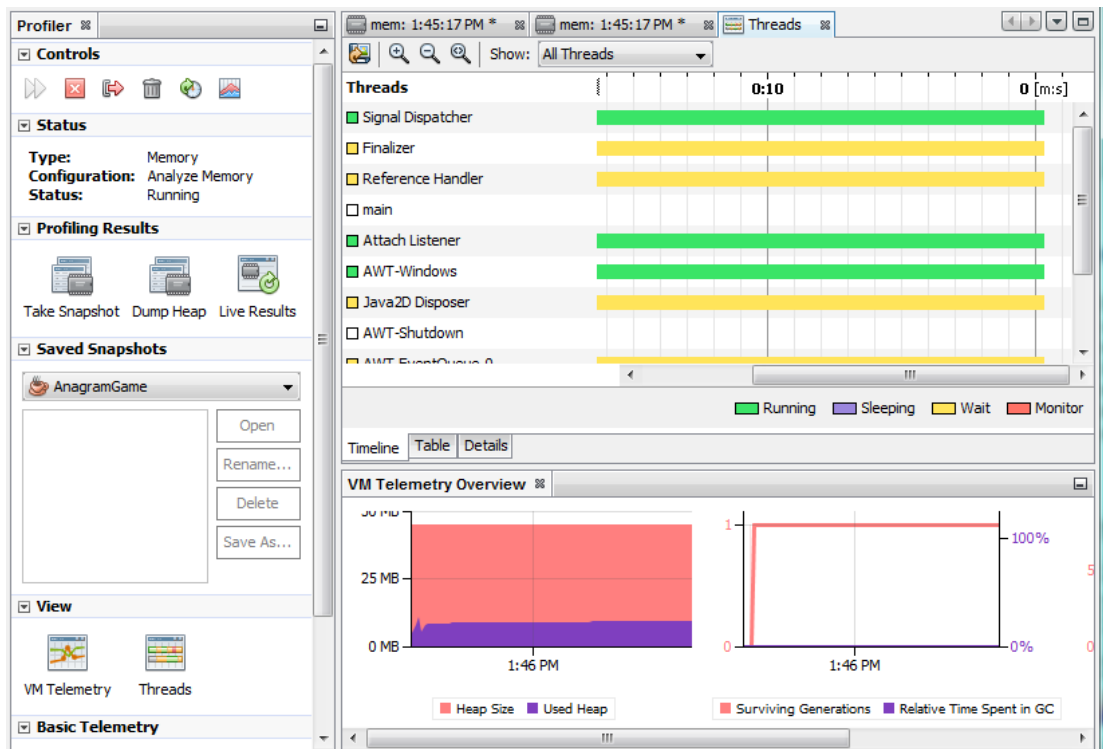


Εικόνα 9: Καλύτερη οργάνωση έργων

2.2.4 Συγγραφή κώδικα χωρίς λάθη

Το κόστος του κώδικα με λάθη αυξάνει όσο περισσότερο παραμένει μη διορθωμένα. Το NetBeans παρέχει στατικά εργαλεία ανάλυσης, ιδιαίτερα το ενσωματωμένο εργαλείο που χρησιμοποιείται ευρέως το FindBugs, για τον εντοπισμό και τον καθορισμό κοινών προβλημάτων στον κώδικα Java. Επιπλέον, το NetBeans Debugger επιτρέπει να τοποθετηθούν σημεία παύσης (break point) στον πηγαίο κώδικα, να προστεθούν μεταβλητές ελέγχου, να εκτελεστεί βηματικά ο κώδικας, να εκτελεστούν οι μέθοδοι σαν μια εντολή ή βηματικά, να λαμβάνει στιγμιότυπα του προγράμματος και να παρακολουθεί την εκτέλεση καθώς αυτή συμβαίνει.

Το NetBeans Profiler παρέχει εξειδικευμένη βοήθεια για τη βελτιστοποίηση της ταχύτητας και της χρήσης μνήμης της εφαρμογής και καθιστά ευκολότερο να δημιουργηθούν εφαρμογές αξιόπιστες και επεκτάσιμες με Java SE, JavaFX και Java EE (Oracle, 2016b).



Εικόνα 10: Εργαλεία βελτίωσης εφαρμογής

2.3 Εγκατάσταση

Για να εκτελεστεί ένα πρόγραμμα Java, θα πρέπει να υπάρχει εγκατεστημένο το πρόγραμμα μεταγλώττισης Java στον υπολογιστή. Το πρόγραμμα μεταγλώττισης μπορεί να κατέβει από το δικτυακό τόπο της oracle.

Στη συνέχεια θα πρέπει να κατεβάσουμε το NetBeans από τη διεύθυνση

Όπως μπορούμε να δούμε από την εικόνα που ακολουθεί, υπάρχουν διάφορες εκδόσεις του NetBeans ανάλογα με το τι αυτό θα περιλαμβάνει και τι δυνατότητες θα προσφέρει.

NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME						•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

Download	Download	Download x86	Download x86	Download x86	Download
		Download x64	Download x64	Download x64	
Free, 95 MB	Free, 192 MB	Free, 104 - 107 MB	Free, 104 - 107 MB	Free, 106 - 110 MB	Free, 215 MB

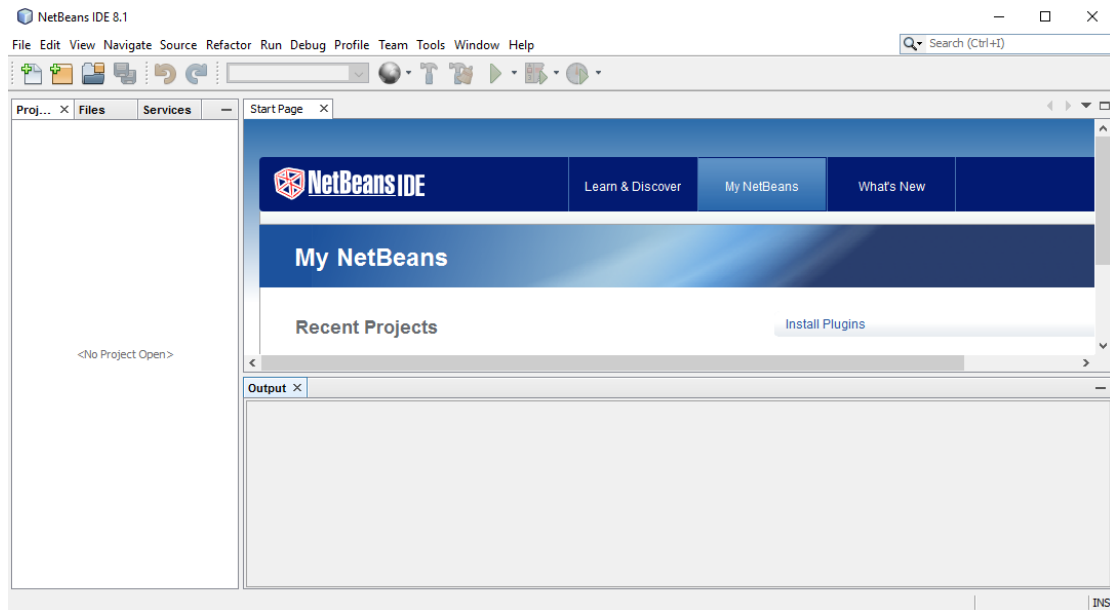
Εικόνα 11: Εκδόσεις NetBeans

τη συνέχεια θα πρέπει να κατεβάσουμε το NetBeans από τη διεύθυνση

Για τη δική μας περίπτωση, μπορούμε να επιλέξουμε να κατεβάσουμε το Java SE ή JAVA EE. Αφού ολοκληρωθεί η λήψη του, ακολουθούμε τα βήματα της εγκατάστασης.

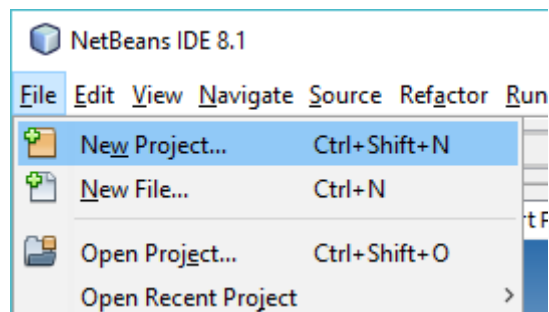
2.4 Δημιουργία project

Όταν ξεκινήσουμε το NetBeans θα δούμε την αρχική του οθόνη, η οποία στο αριστερό τμήμα θα μας εμφανίσει το τελευταίο project με το οποίο έχουμε εργαστεί, την αρχική σελίδα και την οθόνη εξόδου.



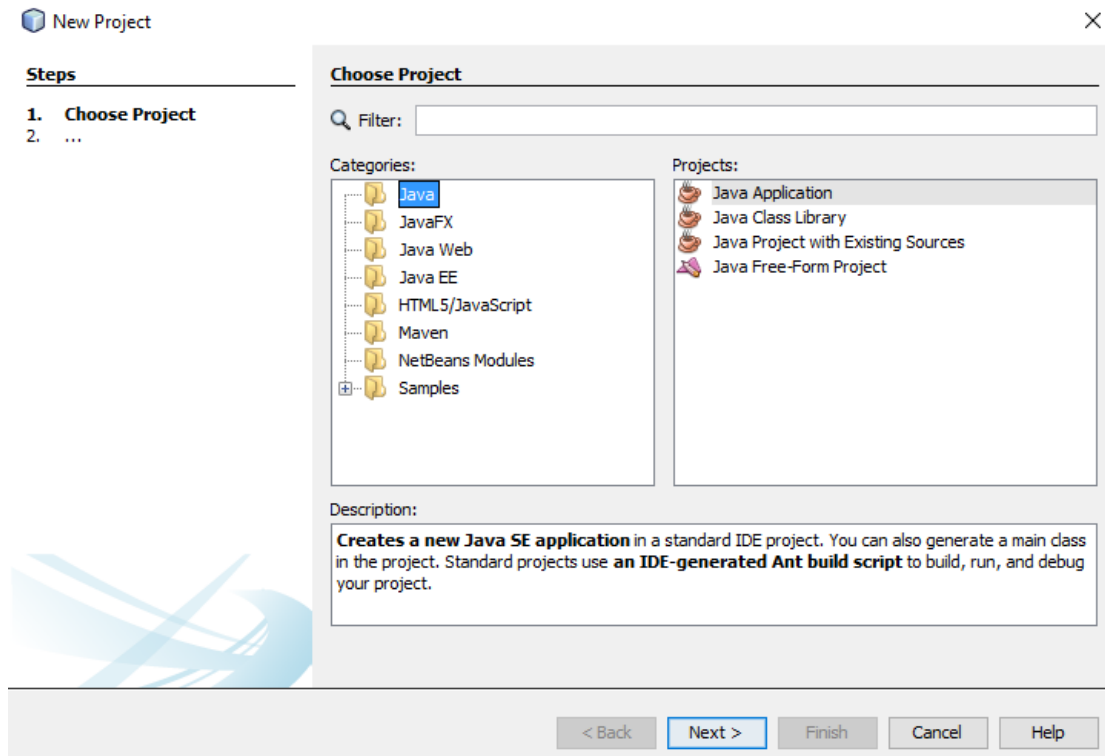
Εικόνα 12: Αρχική οθόνη NetBeans

Για να δημιουργήσουμε ένα νέο project επιλέγουμε από το μενού File -> New Project, όπως βλέπουμε στην εικόνα που ακολουθεί:



Εικόνα 13: 1ο βήμα

Όταν το επιλέξουμε θα ανοίξει ένας wizard όπου θα μπορούμε να επιλέξουμε τον τύπο του project. Στην περίπτωση μας θα επιλέξουμε Java Application από το Java Categories και θα πατήσουμε το Next.

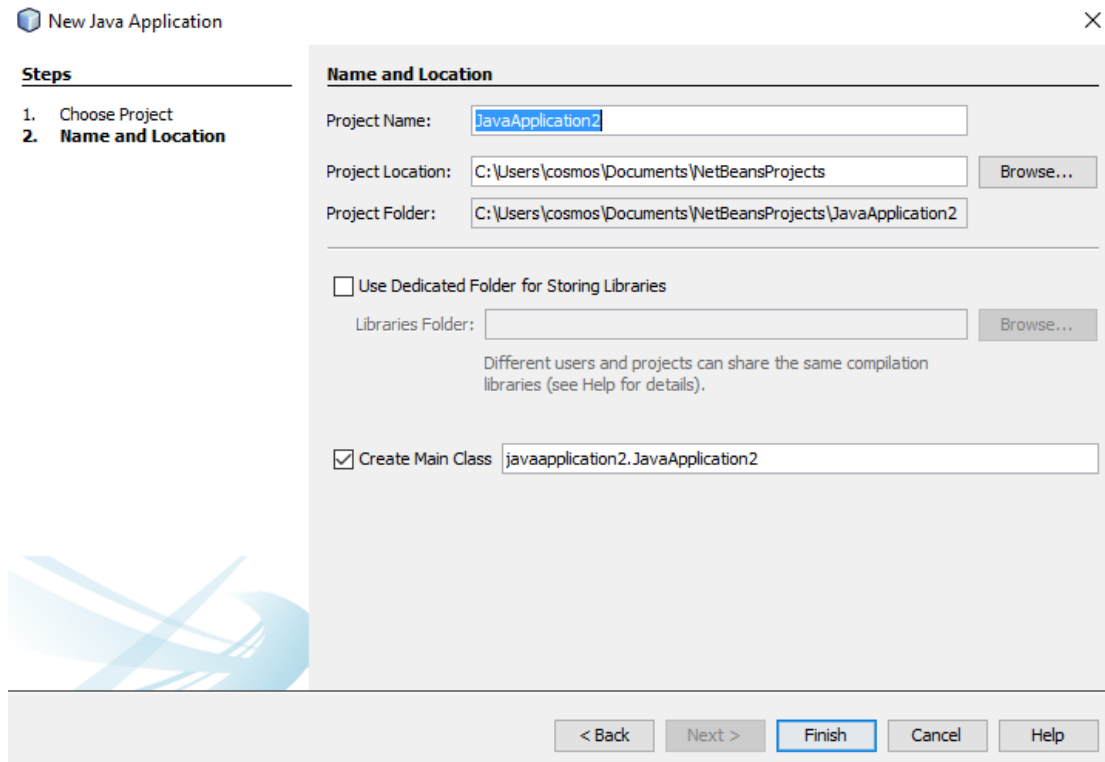


Εικόνα 14: 2ο βήμα

Στο επόμενο βήμα θα πρέπει να εισάγουμε πληροφορίες για το project.

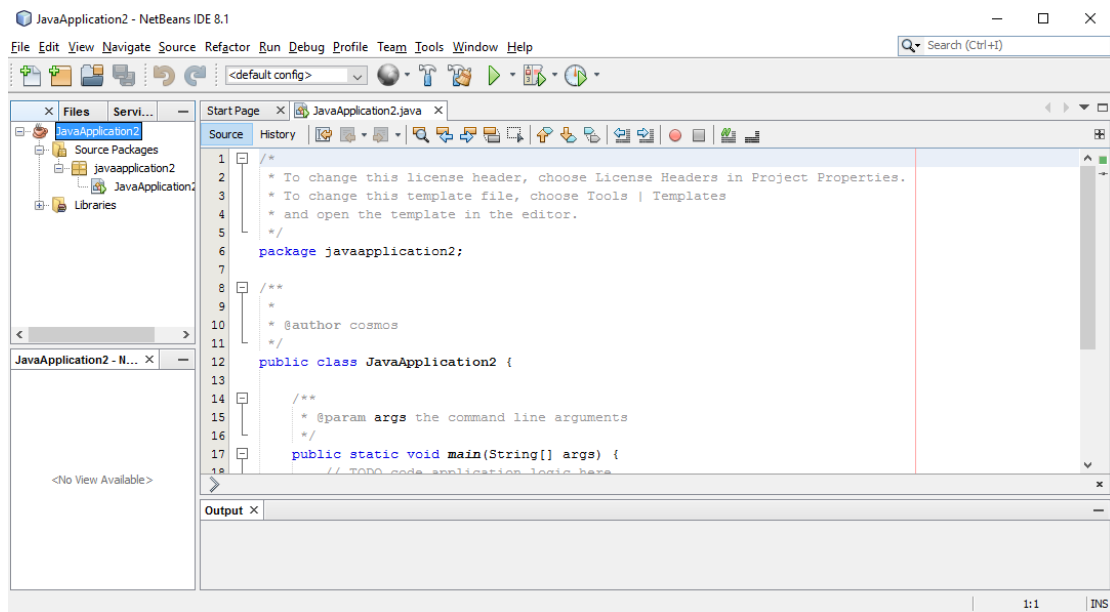
- Στο πεδίο Project Name θα εισάγουμε το όνομα του project.
- Στο πεδίο Project Location μπορούμε να επιλέξουμε τον κατάλογο που θα περιέχει τον κατάλογο του project.
- Το πεδίο Project Folder δεν μπορούμε να το τροποποιήσουμε και δείχνει την τελική διαδρομή που θα αποθηκευτεί το project.
- Το πεδίο Use Dedicated Folder for Storing Libraries χρησιμοποιείται όταν θέλουμε να υπάρχουν σε έναν κατάλογο όλες οι βιβλιοθήκες που μπορεί να χρησιμοποιήσουμε στα διάφορα project που θα υλοποιούμε. Στη δική μας περίπτωση δε θα το χρησιμοποιήσουμε.
- Το πεδίο Create Main Class χρησιμοποιείται αν θέλουμε το NetBeans να δημιουργήσει τη Main Class του project.

- To Set as Main Project το αφήνουμε επιλεγμένο αφού θέλουμε αυτό να είναι το βασικό μας project.



Εικόνα 15: 3ο βήμα

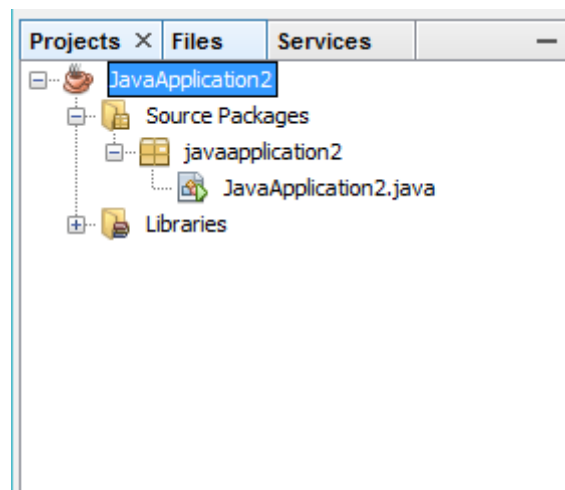
Αν πατήσουμε το Finish, μετά από λίγο, θα δημιουργηθεί το project και θα εμφανιστεί στο NetBeans. Θα βλέπουμε στην οθόνη μας την ακόλουθη εικόνα:



Εικόνα 16: 4ο βήμα - τελικό

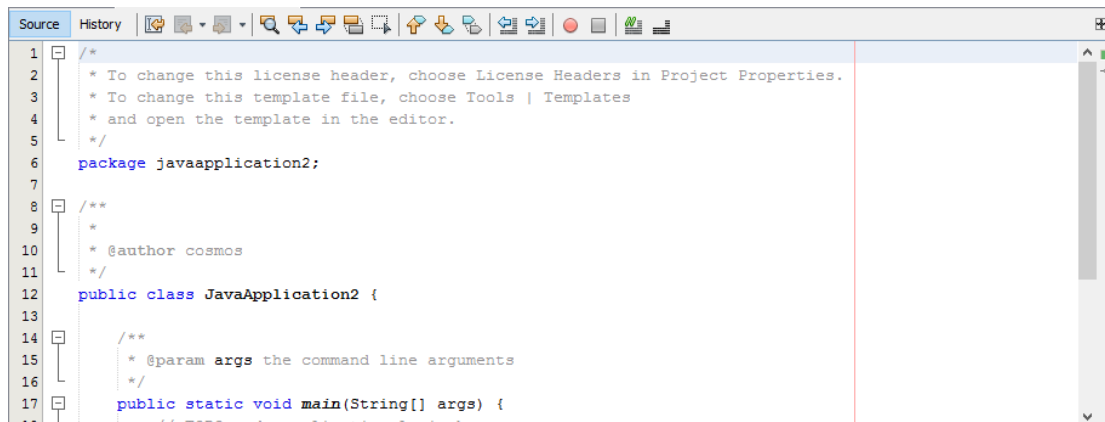
Στην οθόνη μπορούμε να ξεχωρίσουμε τις ακόλουθες περιοχές:

- Το παράθυρο projects: περιέχει σε μια δομή δέντρου όλα τα στοιχεία του project, συμπεριλαμβανομένων των source αρχείων, των βιβλιοθηκών από τις οποίες εξαρτάται κ.λπ.



Εικόνα 17: Το παράθυρο projects

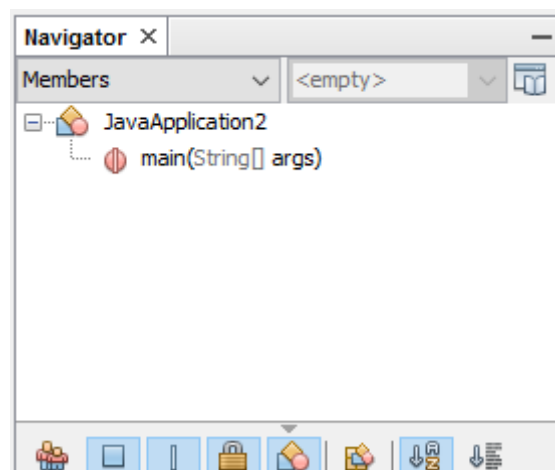
- Το παράθυρο Source Editor: δείχνει τα περιεχόμενα του αρχείου javaapplication2, στο παράθυρο αυτό θα περιέχονται τα αρχεία που επιλέγουμε από το project και θα προσθέτουμε τις εντολές που θέλουμε.



```
1  /**
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package javaapplication2;
7
8  /**
9  *
10 * @author cosmos
11 */
12 public class JavaApplication2 {
13
14     /**
15     * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20 }
```

Εικόνα 18: Το παράθυρο Source Editor

- Το παράθυρο Navigator: μπορούμε να μεταβούμε γρήγορα και εύκολα μεταξύ των στοιχείων που αποτελούν την κλάση



Εικόνα 19: Το παράθυρο Navigator

- Η έξοδος (Output): περιέχει την έξοδο ενός προγράμματος ή

μηνύματα λάθους.

Στο σημείο αυτό, μπορούμε να ξεκινήσουμε τη δημιουργία του προγράμματός μας. Αν προσθέσουμε την ακόλουθη εντολή μέσα στη main της κλάσης:

```
System.out.println("Hello World!");
```

Μπορούμε να αποθηκεύσουμε την κλάση μας επιλέγοντας από το μενού File -> Save και για να τρέξουμε το πρόγραμμα να επιλέξουμε από το μενού Run -> Run Project. Τότε στο Output θα δούμε την ακόλουθη εικόνα:



Εικόνα 20: Output προγράμματος

Κεφάλαιο 3ο Βάση Δεδομένων MySQL

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα κάνουμε μια σύντομη ιστορική αναδρομή στη MySQL που θα είναι η βάση δεδομένων που θα χρησιμοποιήσουμε.

3.2 Ιστορική Αναδρομή της MySQL

Η MySQL δημιουργήθηκε από μια Σουηδική εταιρία, τη MySQL AB το 1995. Οι προγραμματιστές που τη δημιούργησαν ήταν δύο Σουηδοί και ένας Φινλανδός, οι Σουηδοί David Axmark και Allan Larsson και ο Φινλανδός Michael Widenius (Monty). Ο κυριότερος σκοπός ήταν να παρέχει αποτελεσματικές και αξιόπιστες επιλογές στη διαχείριση δεδομένων για οικιακή αλλά και επαγγελματική χρήση. Τα πρώτα χρόνια εκπροσωπείται από μια εταιρία την TCX (Buytaert, 2016).

Το όνομά της προέρχεται από το όνομα της κόρης του Μόντυ Μy. Ο οποίος δημιούργησε και άλλες βάσεις δεδομένων δίνοντας ονόματα των άλλων παιδιών του MariaDB και MaxDB (Dybka, 2014).



Εικόνα 21: MySQL (Dybka, 2014)

Κατά το έτος 2000, κυκλοφόρησαν πολλές αρχικές εκδόσεις (alpha και beta) οι οποίες ήταν συμβατές με σχεδόν όλες τις διαδεδομένες πλατφόρμες εκείνης της εποχής.

Την ίδια χρονιά πάρθηκε απόφαση να γίνει ανοικτού κώδικα, ακολουθώντας τους όρους του GPL, κάτι το οποίο οδήγησε σε σημαντική πτώση των εσόδων της εταιρίας – περίπου στο 80%-. Σε ένα χρόνο ανέκτησε γρήγορα το χαμένο έδαφος, σε αυτό βοήθησε η βελτίωση των χαρακτηριστικών της που προήλθε από τη συνεισφορά τρίτων προγραμματιστών.

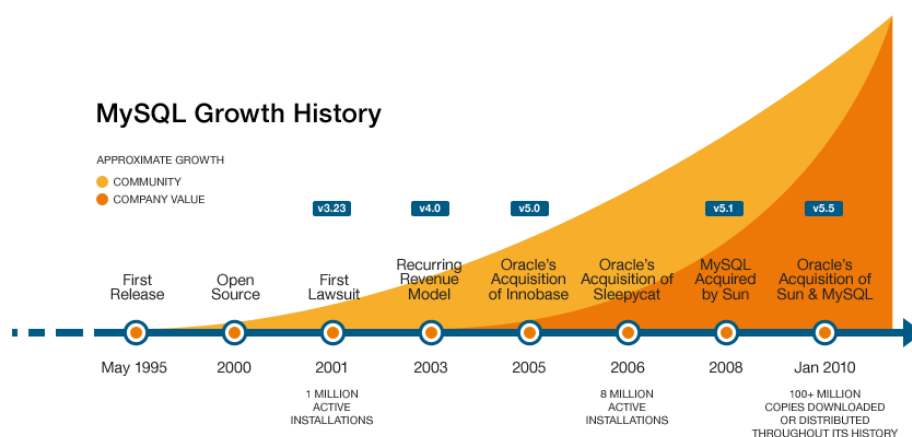
Η MySQL απέκτησε σταθερή δημοτικότητα τόσο στους οικιακούς χρήστες, όσο και σε επαγγελματίες και το έτος 2001, η πλατφόρμα είχε φτάσει τα 2 εκατομμύρια ενεργές εγκαταστάσεις. Την ίδια χρονιά, κυρίως εξαιτίας της αυξανόμενης δημοτικότητάς της, η MySQL μηνύθηκε από την Progress για παράβαση της άδειας GPL. Το 2002, η εταιρεία άνοιξε κεντρικά γραφεία στις Η.Π.Α ταυτόχρονα με τα κεντρικά γραφεία της στη Σουηδία. Επίσης ανακοίνωσε ότι τα ενεργά μέλη της πλατφόρμας ξεπέρασαν τα 3 εκατομμύρια χρήστες με έσοδα που ανέρχονται σε \$6,5 εκατομμύρια από 1000 πελάτες επί πληρωμή.

Η πλατφόρμα συνέχισε να κερδίζει δημοτικότητα και μέχρι το τέλος του 2003, είχαν φτάσει τα έσοδά της σε 12 εκατομμύρια δολάρια με 4 εκατομμύρια ενεργές εγκαταστάσεις και πάνω από 30.000 λήψεις ανά ημέρα. Ταυτόχρονα ξεκινά μια συνεργασία με τη SAP, η οποία θα ολοκληρωθεί το 2007. Το 2004 η εταιρεία αποφάσισε να επικεντρωθεί περισσότερο στο μοντέλο επαναλαμβανόμενων χρεώσεων στους χρήστες, αντί για χρέωση μόνο στην αρχική αδειοδότηση και να κινηθεί περισσότερο στην αγορά των επιχειρήσεων. Το έτος έκλεισε με καθαρά έσοδα ύψους \$20 εκατομμυρίων.

Το 2005, η MySQL ξεκίνησε το MySQL Network το οποίο δημιουργήθηκε μετά από το RedHat Network. Το MySQL Network είναι μια συνδρομητική υπηρεσία που απευθύνεται σε τελικούς χρήστες και παρέχει ενημερώσεις, ειδοποιήσεις, κοινοποιήσεις, και υποστήριξη σε επίπεδο προϊόντος που αποσκοπεί να καταστήσει ευκολότερο για τις εταιρείες να διαχειρίζονται εκατοντάδες servers. Η Oracle αγόρασε την Innobase, την εταιρεία που διαχειρίζεται την MySQL Innobase storage backend, τη μηχανή αποθήκευσης η οποία προσέφερε πολλές σημαντικές λειτουργίες στη MySQL (transactions και foreign keys). Το ίδιο έτος, κυκλοφόρησε η MySQL 5, μία έκδοση με νέα

χαρακτηριστικά (ιδιαίτερα για εταιρικούς χρήστες) όπως stored procedures, triggers, views, cursors, distributed transactions, federated storage engines κ.λπ.

Το 2008, η MySQL εξαγοράστηκε από την Sun Microsystems για 1 δις δολάρια. Οι Michael Widenius (Monty) και David Axmark, δύο από τους συνιδρυτές της MySQL AB, άφησαν τη Sun λίγο μετά την εξαγορά.



στορικό Ανάπτυξης της MySQL (Database Friends, 2014)

Μέχρι το τέλος του έτους η Sun είχε χάσει τα περισσότερα στελέχη που είχαν κάνει επιτυχημένη τη MySQL. Το 2009 η Sun ανακοίνωσε ότι συμφώνησε με την Oracle για την εξαγορά της (μαζί με την MySQL).

Στην σύγχρονη εποχή η MySQL συνεχίζει να χρησιμοποιείται από δημοφιλείς δικτυακούς τόπους όπως το Facebook, Yahoo, Wikipedia και έμμεσα από κάθε δικτυακό τόπου που είναι φτιαγμένος σε Joomla, Drupal και WordPress (Buytaert, 2016). Με την έλευση του Cloud Computing, η MySQL έγινε και αυτή συμβατή με τις διάφορες υπηρεσίες cloud computing, όπως η Amazon EC2. Διάφορα μοντέλα ανάπτυξης έχουν χρησιμοποιηθεί για την εφαρμογή της MySQL σε πλατφόρμες cloud computing. Ίσως το πιο δημοφιλές από αυτά, είναι το «Virtual Machine Image», το οποίο είναι ένα image μιας μηχανής που περιέχει τη MySQL προ-εγκατεστημένη. Ένα άλλο μοντέλο (όχι τόσο διαδεδομένο) είναι η βάση δεδομένων να μην είναι διαθέσιμη σαν υπηρεσία αλλά να φιλοξενείται και να διαχειρίζεται για

λογαριασμό του ιδιοκτήτη (DataBase Friends, 2014).

Κεφάλαιο 4ο Σχεδιασμός και Υλοποίηση Εφαρμογής

4.1 Εισαγωγή

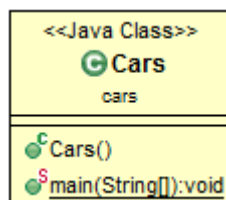
Στο κεφάλαιο αυτό θα παρουσιάσουμε τα βήματα που ακολουθήσαμε για το σχεδιασμό και την υλοποίηση της εφαρμογής. Θα ξεκινήσουμε περιγράφοντας τα διαγράμματα κλάσεων, θα προχωρήσουμε στη σχεδίαση της βάσης δεδομένων, στη συνέχεια θα αναλύσουμε τις μεθόδους των κλάσεων που υλοποιήσαμε και τέλος θα περιγράψουμε τον τρόπο εκτέλεσης.

4.2 Διαγράμματα Κλάσεων

Στην ενότητα αυτή θα παρουσιάσουμε τα UML διαγράμματα κλάσεων της εφαρμογής. Θα αφιερώσουμε μια παράγραφο για κάθε μια κλάση παρουσιάζοντας το UML διάγραμμα και εξηγώντας τη λειτουργία της.

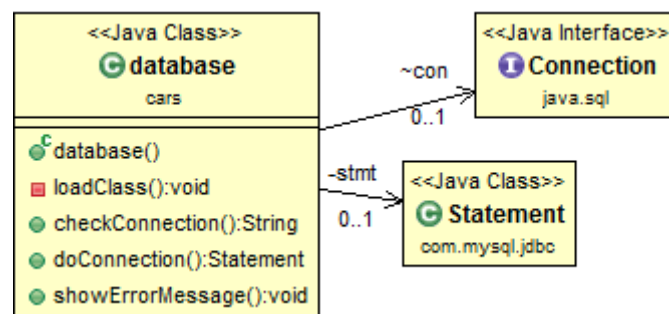
4.2.1 Κλάση Cars

Η κλάση αυτή είναι η κλάση που περιέχει τη main μέθοδο του project μας. Από την κλάση αυτή ξεκινά η εφαρμογή μας και δημιουργεί ένα στιγμιοτύπο του αντικειμένου TextForm.



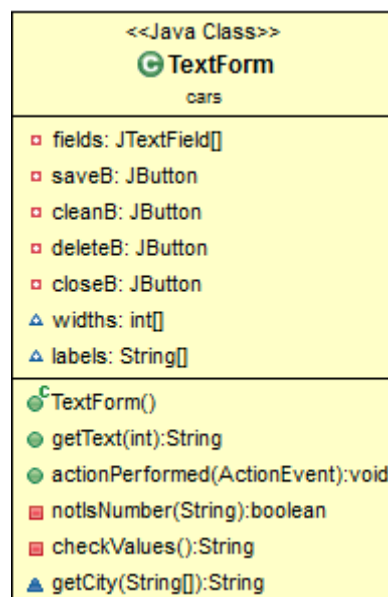
4.2.2 Κλάση database

Η κλάση αυτή χρησιμοποιείται για τη δημιουργία της σύνδεσης με τη βάση δεδομένων. Περιέχει τις απαραίτητες μεθόδους για να φορτωθεί η βιβλιοθήκη της mysql, τη σύνδεση με τη βάση δεδομένων (δίνοντας το όνομα χρήστη και τον κωδικό πρόσβασης).



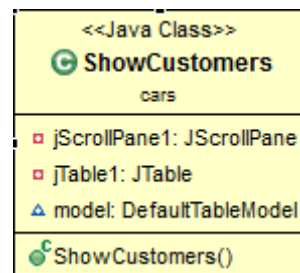
4.2.3 Κλάση TextForm

Η κλάση αυτή είναι η βασική κλάση της πτυχιακής μας. Εμφανίζει το GUI, και ελέγχει όλες τις λειτουργίες που έχουμε σχεδιάσει.



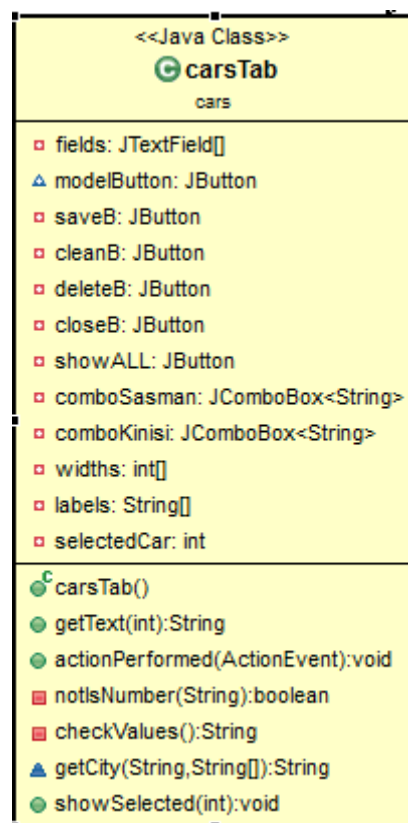
4.2.4 Κλάση ShowCustomers

Η κλάση ShowCustomers, είναι υπεύθυνη για να εμφανίζει ένα JTable το οποίο να περιέχει τα βασικά στοιχεία των πελατών.



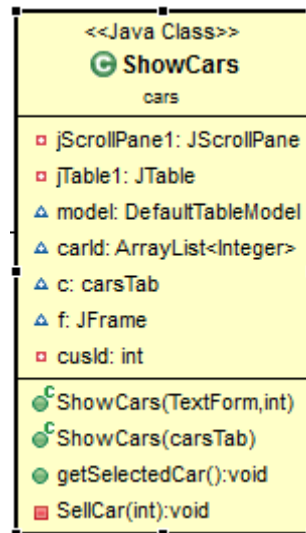
4.2.5 Κλάση carsTab

Η κλάση carsTab, εμφανίζει την καρτέλα για τους πελάτες και προσφέρει όλες τις λειτουργίες διαχείρισής τους.



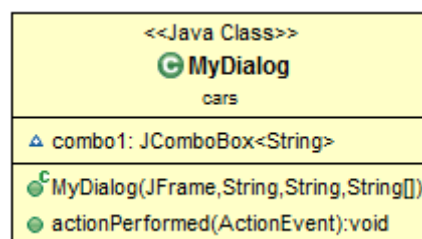
4.2.6 Κλάση ShowCars

Η κλάση ShowCars, είναι υπεύθυνη για να εμφανίζει ένα jTable το οποίο να περιέχει τα βασικά στοιχεία των αυτοκινήτων.



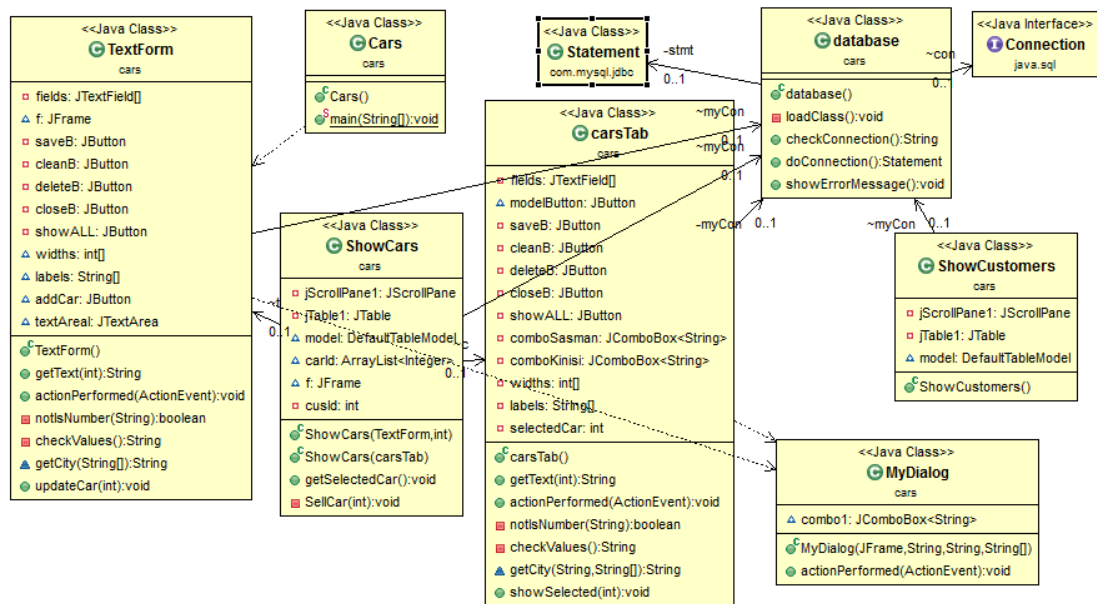
4.2.7 Κλάση MyDialog

Η κλάση MyDialog, εμφανίζει ένα Combo box με βάση τον πίνακα των String που έχει λάβει.



4.2.8 Συνολικό διάγραμμα κλάσεων

Το συνολικό διάγραμμα των κλάσεων της εφαρμογής μας είναι:



4.3 Σχεδιασμός Βάσης Δεδομένων

Στην ενότητα αυτή θα περιγράψουμε τη βάση δεδομένων που δημιουργήσαμε για να αποθηκεύονται τα δεδομένα του προγράμματος.

Ο βασικός πίνακας που έπρεπε να περιέχεται στη βάση δεδομένων ήταν ο πίνακας των πελατών (customers). Ο πίνακας αυτός θα είχε τα ακόλουθα πεδία για να περιγράφεται πλήρως ένας πελάτης:

Πεδίο	Περιγραφή	Τύπος
cId	Κωδικός Πελάτη	Ακέραιος
cSurname	Επώνυμο Πελάτη	Χαρακτήρες
cName	Όνομα Πελάτη	Χαρακτήρες
afm	ΑΦΜ Πελάτη	Ακέραιος
eforia	Εφορία Πελάτη	Χαρακτήρες
address	Διεύθυνση Πελάτη	Χαρακτήρες
city	Πόλη Πελάτη	Χαρακτήρες

celphone	Κινητό Τηλέφωνο Πελάτη	Χαρακτήρες
phone	Τηλέφωνο Πελάτη	Χαρακτήρες
carId	Κωδικός αυτοκινήτου που έχει αγοράσει	Ακέραιος

Παρατηρώντας τα στοιχεία του πελάτη που συγκεντρώσαμε στον πίνακα customers είδαμε ότι το πεδίο city (που περιγράφει την πόλη του πελάτη) αν αφεθεί, στο χειριστή του προγράμματος, να συμπληρωθεί, μπορεί να συμπληρωθεί με πολλούς τρόπους π.χ. όλα κεφαλαία, όλα πεζά με τόνο, όλα πεζά χωρίς τόνο κ.λπ. Για το λόγο αυτό θεωρήσαμε σκόπιμο να δημιουργήσουμε έναν ακόμη πίνακα, όπου ο διαχειριστής της βάσης δεδομένων θα έχει ήδη καταχωρίσει τις πόλεις που θέλουμε και ο χειριστής του προγράμματος απλά θα επιλέγει μια από αυτές.

Το νέο πίνακα που δημιουργήσαμε τον ονομάσαμε cities:

Πεδία	Περιγραφή	Τύπος
cityId	Μοναδικός κωδικός πόλης	Ακέραιος
city	Όνομα πόλης	Χαρακτήρες

Ένας άλλος βασικός πίνακας της εφαρμογής είναι ο πίνακας cars ο οποίος περιέχει το στόλο των αυτοκινήτων της αντιπροσωπίας. Ο πίνακας αυτός έχει τα ακόλουθα πεδία για να περιγράφεται πλήρως ένα αυτοκίνητο:

Πεδίο	Περιγραφή	Τύπος
carId	Κωδικός Αυτοκινήτου	Ακέραιος
brand	Μάρκα αυτοκινήτου	Χαρακτήρες
Model	Μοντέλο αυτοκινήτου	Χαρακτήρες
yearRelease	Έτος Κυκλοφορίας	Ακέραιος
cc	Κυβικά	Ακέραιος

hp	Ίπποι	Ακέραιος
color	Χρώμα	Χαρακτήρες
sasman	Σασμάν	Χαρακτήρες
kinisi	Κίνηση	Χαρακτήρες
portes	Αριθμός πορτών	Ακέραιος
sold	Bit που είναι 0 αν δεν έχει αγοραστεί και 1 αν έχει αγοραστεί	bit

Παρατηρώντας τα πεδία αυτά βλέπουμε ότι για τα πεδία Μάρκα, μοντέλο και Χρώμα πρέπει να φτιάξουμε ξεχωριστούς πίνακες για να περιέχουν προεπιλεγμένες τιμές από εμάς.

Για το λόγο αυτό δημιουργήσαμε τους ακόλουθους πίνακες:

Τον πίνακα brands για τις μάρκες των αυτοκινήτων:

Πεδία	Περιγραφή	Τύπος
brandId	Μοναδικός κωδικός μάρκας	Ακέραιος
brand	Μάρκα αυτοκινήτου	Χαρακτήρες

Τον πίνακα models για τα μοντέλα:

Πεδία	Περιγραφή	Τύπος
modelId	Μοναδικός κωδικός μοντέλου	Ακέραιος
bid	Κωδικός μάρκας που αντιστοιχεί το μοντέλο	Ακέραιος
modelName	Όνομα μοντέλου	Χαρακτήρες

Τον πίνακα colors

Πεδία	Περιγραφή	Τύπος
-------	-----------	-------

colorId	Μοναδικός κωδικός χρώματος	Ακέραιος
color	Όνομα χρώματος	Χαρακτήρες

Έχοντας σχεδιάσει τους πίνακες και τα πεδία τους, προχωρήσαμε στη δημιουργία της βάσης δεδομένων. Η βάση δεδομένων ονομάστηκε cars και έχει επιλεγεί σύνθεση utf8_bin, για να απεικονίζονται σωστά και οι Ελληνικοί χαρακτήρες.

Δημιουργώντας τους δύο πίνακες λάβαμε και τις τελικές αποφάσεις για τον τύπο τους, όπως το πλήθος των χαρακτήρων που θα περιέχουν:

Για τον πίνακα customers:

Πεδίο	Τύπος
cId	int(11)
cSurname	varchar(25)
cName	varchar(20)
afm	int(11)
eforia	varchar(20)
address	varchar(40)
city	varchar(20)
celphone	varchar(10)
phone	varchar(10)
carId	int(11)

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1	cId 🗝️	int(11)			Όχι	Καμία	
<input type="checkbox"/>	2	cSurname	varchar(25)			Όχι	Καμία	
<input type="checkbox"/>	3	cName	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	4	afm	int(11)			Όχι	Καμία	
<input type="checkbox"/>	5	eforia	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	6	address	varchar(40)			Όχι	Καμία	
<input type="checkbox"/>	7	city	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	8	celphone	varchar(10)			Όχι	Καμία	
<input type="checkbox"/>	9	phone	varchar(10)			Όχι	Καμία	
<input type="checkbox"/>	10	carId	int(11)			Όχι	0	

Εικόνα 23: Πίνακας customers

Για τον πίνακα cities:

Πεδία	Τύπος
cityId	Int(11), autoIncrement
city	Varchar(20)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα	
<input type="checkbox"/>	1	cityId 🗝️	int(11)			Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	city	varchar(20)			Όχι	Καμία	

Εικόνα 24: Πίνακας cities

Για τον πίνακα cars:

Πεδίο	Τύπος
-------	-------

carId	int(11) , autoIncrement
brand	Varchar(20)
Model	Varchar(20)
yearRelease	int(11)
cc	int(11)
hp	int(11)
color	Varchar(25)
sasman	Varchar(20)
kinisi	Varchar(20)
portes	int(11)
sold	bit(1)

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1	carId	int(11)			Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	brand	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	3	Model	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	4	yearRelease	int(11)			Όχι	Καμία	
<input type="checkbox"/>	5	cc	int(11)			Όχι	Καμία	
<input type="checkbox"/>	6	hp	int(11)			Όχι	Καμία	
<input type="checkbox"/>	7	color	varchar(25)			Όχι	Καμία	
<input type="checkbox"/>	8	sasman	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	9	kinisi	varchar(20)			Όχι	Καμία	
<input type="checkbox"/>	10	portes	int(11)			Όχι	Καμία	
<input type="checkbox"/>	11	sold	bit(1)			Όχι	0	

Εικόνα 25: Πίνακας cars

Για τον πίνακα brands:

Πεδία	Τύπος
brandId	int(11) , autoIncrement
brand	Varchar(20)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1	brandId		int(11)	Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	brand		varchar(20)	Όχι	Καμία	

Εικόνα 26: Πίνακας brands

Για τον πίνακα models:

Πεδία	Τύπος
mId	int(11) , autoIncrement
bid	int(11)
modelName	Varchar(20)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1	mId		int(11)	Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	bid		int(11)	Όχι	Καμία	
<input type="checkbox"/>	3	modelName		varchar(20)	Όχι	Καμία	

Εικόνα 27: Πίνακας models

Για τον πίνακα colors:

Πεδία	Τύπος
colorId	int(11) , autoIncrement
color	Varchar(20)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
<input type="checkbox"/>	1	colorId		int(11)	Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	color		varchar(20)	Όχι	Καμία	

Εικόνα 28: Πίνακας colors

4.4 Σχεδιασμός και Υλοποίηση Εφαρμογής

Μετά από την ολοκλήρωση των UML διαγραμμάτων και του σχεδιασμού της βάσης δεδομένων, θα προχωρήσουμε στην υλοποίηση της εφαρμογής. Η γλώσσα που επιλέχτηκε είναι η Java. Στις ενότητες που ακολουθούν θα περιγράψουμε τα βήματα υλοποίησης των κλάσεων.

4.4.1 Κλάση Cars

Όπως έχουμε αναφέρει η κλάση αυτή περιέχει τη `main` του `project`. Στη μέθοδο `main`, δημιουργείται ένα αντικείμενο του τύπου `TextForm` και στη συνέχεια συνεχίζει η εκτέλεση με τη βοήθεια των μεθόδων του αντικειμένου αυτού.

4.4.2 Κλάση database

Ξεκινήσαμε την υλοποίηση με τη δημιουργία μια κλάσης που θα αναλάμβανε να κάνει τη διασύνδεση με τη βάση δεδομένων. Για να γίνει η διασύνδεση με τη MySQL έπρεπε να χρησιμοποιήσουμε τον `connector mysql-connector-java-5.0.8-bin.jar` που παρέχεται από τη MySQL. Ο `connector` αυτός μπορεί να αντικατασταθεί από κάποιο νεότερο, αρκεί να υποστηρίζει τις μεθόδους του `connector` που επιλέξαμε και να τοποθετηθεί στον κατάλογο `Libraries` του `project`.

Δημιουργήσαμε τη μέθοδο `loadClass`, η οποία αναλαμβάνει να φορτώσει τον `connector`. Η φόρτωση του `connector` γίνεται με τον έλεγχο της δομής `try catch`, έτσι ώστε να αναγνωρίσουμε αν συμβεί κάποιο λάθος.

Η επόμενη μέθοδος που δημιουργήσαμε είναι η `checkConnection` η οποία αναλαμβάνει να καλέσει τη `loadClass`, για να φορτωθεί ο `connector` και στη συνέχεια να συνδεθεί με τη βάση δεδομένων MySQL χρησιμοποιώντας

σαν connection string το ακόλουθο:

```
"jdbc:mysql://localhost:3306/cars?characterEncoding=utf8","root", ""
```

Σε αυτό το connection string περιγράφουμε:

- τη διεύθυνση που βρίσκεται η βάση δεδομένων μας(*localhost:3306*),
- τη βάση δεδομένων που θέλουμε να συνδεθούμε (*cars*),
- το Encoding που θα χρησιμοποιήσουμε(*utf8*),
- το username (*root*) και το password (κενό) για τη σύνδεση σε αυτή.

Αν δεν προκύψει κάποιο σφάλμα, τότε η μέθοδος θα επιστρέψει το κενό string με το οποίο δηλώνει ότι δεν προέκυψε κάποιο μήνυμα λάθους. Αν όμως προκύψει κάποιο σφάλμα (το οποίο θα αναγνωριστεί – συλληφθεί) με τη δομή (try catch) τότε εμφανίζουμε ένα μήνυμα λάθους και επιστρέφουμε ένα string, με το οποίο δηλώνουμε το μήνυμα λάθους που προέκυψε.

Η μέθοδος `getConnection` είναι παρόμοια με την `checkConnection`. Προσπαθεί να συνδεθεί με τη βάση δεδομένων και αν επιτύχει, αποθηκεύει τη σύνδεση σε μια μεταβλητή τύπου `Connection` η οποία είναι `public`. Επίσης δημιουργεί μια μεταβλητή τύπου `Statement` (για τη λήψη των αποτελεσμάτων ερωτημάτων `select`) που είναι και αυτή `public`.

Η μέθοδος `showErrorMessage` χρησιμοποιείται για να εμφανίζει ένα μήνυμα λάθους.

4.4.3 Κλάση `TextForm`

Η κλάση `TextForm`, είναι υπεύθυνη για την όλες τις λειτουργίες που σχετίζονται με τους πελάτες. Εμφανίζει μια φόρμα, στο αντίστοιχο `tab`, με τα κατάλληλα πεδία και με τη βοήθεια των `event` που συμβαίνουν με το πάτημα των `button`, συνεχίζεται η εκτέλεση του προγράμματος.

4.4.3.1 Constructor

Ο constructor της κλάσης αυτής, είναι υπεύθυνος για τη δημιουργία της φόρμας. Αρχικά δημιουργεί ένα αντικείμενο του τύπου database για σύνδεση με τη βάση δεδομένων. Στη συνέχεια καλεί τη μέθοδο checkConnection του αντικειμένου database για να ελέγξει αν υπάρχει κάποιο πρόβλημα. Αν υπάρχει (π.χ. λάθος username ή password) τότε η μέθοδος θα επιστρέψει ένα μήνυμα λάθους και ο constructor θα σταματήσει την εκτέλεση του προγράμματος καλώντας την εντολή System.exit().

Στη συνέχεια δημιουργεί τα button που θα χρησιμοποιήσουμε στη φόρμα και προσθέτει στο καθένα και ένα Action Listener, δηλαδή δηλώνει ότι αν πατηθεί το button τότε πρέπει να κληθεί η μέθοδος actionPerformed που βρίσκεται στην ίδια κλάση και γίνεται Override (αφού κληρονομείται από τον ActionListener που υλοποιούμε). Για τα δύο button που θα περιέχουν την εικόνα με το φακό για αναζήτηση, βρίσκουμε την εικόνα (search_1.png) και την αναθέτουμε στο button με τη μέθοδό του setIcon. Επίσης και στα δύο αυτά button προσθέτουμε Action Listener και τους ορίζουμε ξεχωριστά ονόματα για να μπορούμε να τα ξεχωρίσουμε στη μέθοδο actionPerformed.

Κατόπιν δημιουργούμε τα απαραίτητα JPanel για να τοποθετήσουμε μέσα σε αυτά τα JLabel, JTextField και JButton και τέλος ένα JFrame το οποίο τα περιέχει όλα αυτά. Για τη δημιουργία τους έχουμε χρησιμοποιήσει τρεις πίνακες, τον πίνακα labels που περιέχει τα λεκτικά που θα απεικονίζονται στα JLabel, ο πίνακας widths που περιέχει το μέγιστο μήκος των JTextFields και ο πίνακας fields που περιέχει τα JTextField που θα δημιουργήσουμε. Κάθε ένα στοιχείο του πίνακα fields είναι ένα JTextField.

4.4.3.2 getText

Η μέθοδος αυτή δέχεται μια παράμετρο έναν ακέραιο που αντιστοιχεί στο JTextField για το οποίο θέλουμε να διαβάσουμε το κείμενο που περιέχει.

Επιστρέφει το κείμενο του JTextField που επιλέξαμε.

4.4.3.3 actionPerformed

Είναι η βασική μέθοδος που περιέχει όλους τους ελέγχους, για το ποιο κουμπί έχει επιλέξει ο χρήστης και την ενέργεια που πρέπει να εκτελεστεί.

Για το διαχωρισμό των κουμπιών έχουμε επιλέξει να αποδώσουμε όνομα (με τη μέθοδο setName) στα κουμπιά που είναι για αναζήτηση (τα δύο κουμπιά με την εικόνα) και στα άλλα κουμπιά κάνουμε το διαχωρισμό, ανάλογα με το λεκτικό που αναγράφουν.

Έτσι αν έχει πατηθεί το κουμπί για αναζήτηση με βάση κωδικού πελάτη, θα ελεγχθεί πρώτα αν έχει συμπληρωθεί κωδικός πελάτη και

- αν δεν έχει συμπληρωθεί θα εμφανιστεί ενημερωτικό μήνυμα,
- αν έχει συμπληρωθεί, θα εκτελεστεί μια αναζήτηση στη βάση δεδομένων με βάση τον κωδικό πελάτη. Αν δε βρεθεί πελάτης θα εμφανιστεί ενημερωτικό μήνυμα, διαφορετικά θα συμπληρωθούν τα στοιχεία του στα πεδία και θα αλλάξει το όνομα του κουμπιού από Προσθήκη σε Αποθήκευση (αφού πλέον θέλουμε να τροποποιήσουμε τον πελάτη).

Αν πατηθεί το κουμπί για αναζήτηση πόλης, θα γίνει μια αναζήτηση στον πίνακα cities και τα αποτελέσματα θα προστεθούν σε ένα ArrayList. Στη συνέχεια καλείται η μέθοδος getCity για να εμφανιστούν οι πόλεις και ο χρήστης να επιλέξει μια. Αυτή που επιλέγει ο χρήστης τοποθετείται στο αντίστοιχο JTextField.

Αν έχει πατηθεί το κουμπί Κλείσιμο, τότε θα τερματίσει το πρόγραμμα.

Αν έχει πατηθεί το Εμφάνιση πελατών, τότε καλείται η κλάση ShowCustomers, η οποία και θα δείξει, σε μια νέα φόρμα όλους τους πελάτες που υπάρχουν στη βάση δεδομένων.

Αν έχει πατηθεί το κουμπί Καθαρισμός, τότε διαγράφονται τα κείμενα που, μπορεί, να έχουν πληκτρολογηθεί στα JTextField.

Αν πατήθηκε το κουμπί Αποθήκευση νέου, τότε αρχικά ελέγχεται αν υπάρχει πελάτης με τον ίδιο κωδικό στη βάση δεδομένων και αν ναι τότε εμφανίζεται ενημερωτικό μήνυμα ότι υπάρχει ήδη πελάτης με αυτό τον κωδικό. Αν όμως δεν υπάρχει πελάτης, τότε θα εκτελεσθεί ένα ερώτημα insert στη βάση δεδομένων και θα προστεθεί ο πελάτης σε αυτή. Μετά από την εκτέλεση του ερωτήματος θα εμφανιστεί ενημερωτικό μήνυμα.

Αν έχει επιλεγεί το κουμπί Αποθήκευση, σημαίνει ότι έχει αναζητηθεί κάποιος πελάτης στη βάση δεδομένων, έχει βρεθεί – εμφανίστηκαν τα στοιχεία του, ο χρήστης του προγράμματος έχει τροποποιήσει κάποιο από αυτά και θέλει να ενημερώσει τη βάση δεδομένων. Τότε εκτελείται ένα ερώτημα update στα στοιχεία του πελάτη και μετά από την εκτέλεση του ερωτήματος, εμφανίζεται ενημερωτικό μήνυμα.

Τέλος αν έχει πατηθεί το κουμπί Διαγραφή, τότε θα εκτελεστεί ένα ερώτημα delete στη βάση δεδομένων για τη διαγραφή της εγγραφής του πελάτη. Μετά από την εκτέλεση του ερωτήματος θα εμφανιστεί ενημερωτικό μήνυμα.

4.4.3.4 notIsNumber

Η μέθοδος αυτή δέχεται μια μεταβλητή τύπου String και προσπαθεί να τη μετατρέψει σε ακέραιο με την εντολή Integer.parseInt. Αν επιτύχει η μετατροπή, τότε επιστρέφει false, ενώ αν συμβεί Exception NumberFormatException (το οποίο θα συλληφθεί από τη δομή try-catch) επιστρέφει true.

4.4.3.5 checkValues

Η μέθοδος αυτή, χρησιμοποιείται για να ελέγξουμε αν ο χρήστης έχει συμπληρώσει στα JTextField τιμές μεγαλύτερες από αυτές που δέχεται η βάση δεδομένων, πριν προστεθεί μια νέα εγγραφή στη βάση δεδομένων. Οι

έλεγχοι που κάνει είναι:

- αν είναι αριθμός ο κωδικός του πελάτη
- αν η επωνυμία του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν το όνομα του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν η εφορία του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν το ΑΦΜ του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν η διεύθυνση του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν το κινητό τηλέφωνο του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί
- αν το τηλέφωνο του πελάτη έχει περισσότερους χαρακτήρες από το μέγεθος που έχει οριστεί

Αν έχει συμβεί κάποιο λάθος (περισσότεροι χαρακτήρες από αυτούς που πρέπει) τότε επιστρέφει το κατάλληλο μήνυμα σε String το οποίο και θα εμφανιστεί.

4.4.3.6 *getCity*

Η μέθοδος αυτή δέχεται ένα πίνακα από String και εμφανίζει την κλάση MyDialog με περιεχόμενα τα στοιχεία του πίνακα String, για την επιλογή της πόλης που θέλει ο χρήστης

4.4.4 Κλάση ShowCustomers

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση μια φόρμας που περιέχει ένα JTable και εμφανίζει όλους τους πελάτες που υπάρχουν στη βάση δεδομένων.

Περιέχει μόνο τον constructor ο οποίος και είναι υπεύθυνος για τη δημιουργία της φόρμας και την εμφάνισή της. Αρχικά δημιουργείται ένα αντικείμενο του τύπου database και ελέγχεται ότι υπάρχει σύνδεση με τη βάση δεδομένων. Αν δεν υπάρχει τότε θα εμφανιστεί κατάλληλο μήνυμα και θα τερματίσει η λειτουργία του προγράμματος.

Στη συνέχεια δημιουργούμε ένα αντικείμενο του τύπου DefaultTableModel και κάνουμε Override τη μέθοδό του isCellEditable έτσι ώστε όταν εμφανίζεται το JTable να μην είναι επεξεργάσιμα τα κελιά.

Κατόπιν δημιουργούμε ένα αντικείμενο του τύπου JTable, εκτελούμε ένα ερώτημα αναζήτησης στη βάση δεδομένων για να επιλέξουμε όλους τους πελάτες και τους προσθέτουμε στο DefaultTableModel.

Για να μπορέσουμε όμως να έχουμε μπάρες κύλισης (κατακόρυφα) πρέπει να προσθέσουμε το JTable σε ένα JScrollPane. Τέλος προσθέτουμε το JScrollPane στο JPanel και το JPanel στο JFrame και θέτουμε την ιδιότητα visible (με την setVisible) σε true.

4.4.5 Κλάση carsTab

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση φόρμας με πληροφορίες για τα αυτοκίνητα στο κατάλληλο tab. Υποστηρίζει τις αντίστοιχες εργασίες όπως π.χ. προσθήκη αυτοκινήτου, αλλαγή στοιχείων αυτοκινήτου κ.λπ.

4.4.5.1 Constructor

Ο constructor της κλάσης αυτής, είναι υπεύθυνος για τη δημιουργία της φόρμας. Αρχικά δημιουργεί ένα αντικείμενο του τύπου database για σύνδεση με τη βάση δεδομένων. Στη συνέχεια καλεί τη μέθοδο `checkConnection` του αντικειμένου database για να ελέγξει αν υπάρχει κάποιο πρόβλημα. Αν υπάρχει (π.χ. λάθος username ή password) τότε η μέθοδος θα επιστρέψει ένα μήνυμα λάθους και ο constructor θα σταματήσει την εκτέλεση του προγράμματος καλώντας την εντολή `System.exit()`.

Στη συνέχεια δημιουργεί τα button που θα χρησιμοποιήσουμε στη φόρμα – τα οποία θα υποστηρίζουν τις αντίστοιχες λειτουργίες - και προσθέτει στο καθένα και ένα Action Listener. Με την προσθήκη του Action Listener, δηλώνει ότι αν πατηθεί το button τότε πρέπει να κληθεί η μέθοδος `actionPerformed` που βρίσκεται στην ίδια κλάση και γίνεται Override (αφού κληρονομείται από τον ActionListener που υλοποιούμε). Για τα button που θα περιέχουν την εικόνα με το φακό για αναζήτηση, βρίσκουμε την εικόνα (`search_1.png`) και την αναθέτουμε στο button με τη μέθοδό του `setIcon`. Για να μπορέσουν τα button αυτά να υποστηρίξουν ενέργειες, προσθέτουμε Action Listener και τους ορίζουμε ξεχωριστά ονόματα, έτσι ώστε να τα ξεχωρίσουμε στη μέθοδο `actionPerformed`.

Κατόπιν δημιουργούμε τα απαραίτητα JPanel για να τοποθετήσουμε μέσα σε αυτά τα JLabel, JTextField, JComboBox και JButton και τέλος ένα JFrame το οποίο τα περιέχει όλα αυτά. Για τη δημιουργία τους έχουμε χρησιμοποιήσει τρεις πίνακες, τον πίνακα labels που περιέχει τα λεκτικά που θα απεικονίζονται στα JLabel, ο πίνακας widths που περιέχει το μέγιστο μήκος των JTextFields και ο πίνακας fields που περιέχει τα JTextField που θα δημιουργήσουμε. Κάθε ένα στοιχείο του πίνακα fields είναι ένα JTextField. Σε κατάλληλα σημεία π.χ. για την επιλογή του σασμάν, έχουμε προσθέσει JComboBox, στο οποίο νωρίτερα του έχουμε δώσει τιμές.

4.4.5.2 getText

Η μέθοδος αυτή είναι ίδια με τη μέθοδο `getText` της κλάσης `TextForm`.

4.4.5.3 `actionPerformed`

Είναι η βασική μέθοδος που περιέχει όλους τους ελέγχους, για το ποιο κουμπί έχει επιλέξει ο χρήστης και την ενέργεια που πρέπει να εκτελεστεί.

Για το διαχωρισμό των κουμπιών έχουμε επιλέξει να αποδώσουμε όνομα (με τη μέθοδο `setName`) στα κουμπιά που περιέχουν εικόνα και στα άλλα κουμπιά κάνουμε το διαχωρισμό, ανάλογα με το λεκτικό που αναγράφουν.

Έτσι αν έχει πατηθεί το κουμπί για την επιλογή της μάρκας του αυτοκινήτου, θα εκτελεστεί μια αναζήτηση στη βάση δεδομένων για να βρούμε όλες τις διαθέσιμες μάρκες. Οι μάρκες αυτές θα προστεθούν σε ένα `ArrayList` και στη συνέχεια θα εμφανιστούν στο χρήστη για να επιλέξει.

Αν πατηθεί το κουμπί για επιλογή χρώματος του αυτοκινήτου, θα εκτελεστεί μια αναζήτηση στη βάση δεδομένων για να βρούμε όλα τα διαθέσιμα χρώματα. Τα χρώματα αυτά θα προστεθούν σε ένα `ArrayList` και στη συνέχεια θα εμφανιστούν στο χρήστη για να επιλέξει.

Όσον αφορά το κουμπί για την επιλογή μοντέλου, θα πρέπει πρώτα να έχει επιλεγεί μάρκα αυτοκινήτου για να είναι ενεργό. Όταν πατηθεί, ελέγχει αρχικά αν υπάρχει μάρκα αυτοκινήτου:

- αν δεν υπάρχει, εμφανίζει μήνυμα ότι πρέπει πρώτα να επιλέξουμε μάρκα.
- Αν υπάρχει, θα εκτελεστεί μια αναζήτηση στη βάση δεδομένων για όλα τα διαθέσιμα μοντέλα με βάση τη μάρκα που έχουμε επιλέξει. Τα μοντέλα που θα βρεθούν θα προστεθούν σε ένα `ArrayList` και στη συνέχεια θα εμφανιστούν στο χρήστη για να επιλέξει.

Αν έχει πατηθεί το κουμπί Κλείσιμο, τότε θα τερματίσει το πρόγραμμα.

Αν έχει πατηθεί το Εμφάνιση Αυτοκινήτων, τότε καλείται η κλάση

ShowCars, η οποία και θα δείξει, σε μια νέα φόρμα όλα τα αυτοκίνητα που υπάρχουν στη βάση δεδομένων.

Αν έχει πατηθεί το κουμπί Καθαρισμός, τότε διαγράφονται τα κείμενα που, μπορεί, να έχουν πληκτρολογηθεί στα JTextField.

Αν πατήθηκε το κουμπί Αποθήκευση νέου, τότε αρχικά ελέγχεται αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία στη φόρμα και εκτελείται ένα ερώτημα insert στη βάση δεδομένων για προσθήκη του αυτοκινήτου σε αυτή. Μετά από την εκτέλεση του ερωτήματος θα εμφανιστεί ενημερωτικό μήνυμα.

Αν έχει επιλεγεί το κουμπί Αποθήκευση, σημαίνει ότι έχει επιλεγεί κάποιο αυτοκίνητο από τη λίστα των αυτοκινήτων και ο κωδικός του είναι στη μεταβλητή selectedCar. Τότε εκτελείται ένα ερώτημα στη βάση δεδομένων για ενημέρωση των στοιχείων του αυτοκινήτου και μετά από την εκτέλεση του ερωτήματος, εμφανίζεται ενημερωτικό μήνυμα.

Αν έχει πατηθεί το κουμπί Διαγραφή, σημαίνει ότι έχουμε επιλέξει κάποιο αυτοκίνητο από τη λίστα των αυτοκινήτων και ο κωδικός του είναι στη μεταβλητή selectedCar. Τότε θα εκτελεστεί ένα ερώτημα delete στη βάση δεδομένων για τη διαγραφή του αυτοκινήτου. Μετά από την εκτέλεση του ερωτήματος θα εμφανιστεί ενημερωτικό μήνυμα.

4.4.5.4 notIsNumber

Η μέθοδος αυτή είναι ίδια με τη μέθοδο στην κλάση TextForm.

4.4.5.5 checkValues

Η μέθοδος αυτή είναι παρόμοια με αυτή στην κλάση TextForm, η οποία ελέγχει τις τιμές που έχουν συμπληρωθεί στα πεδία και επιστρέφει κατάλληλο μήνυμα.

4.4.5.6 *getCity*

Η συγκεκριμένη μέθοδος χρησιμοποιείται για να εμφανίζεται μια επιλογή στο χρήστη με συγκεκριμένες τιμές, με τη χρήση της κλάσης MyDialog.

4.4.5.7 *showSelected*

Η μέθοδος χρησιμοποιείται για να εμφανίσει στη φόρμα τα στοιχεία ενός αυτοκινήτου που έχει επιλεγεί. Δέχεται σαν παράμετρο τον κωδικό του αυτοκινήτου, εκτελεί το ερώτημα στη βάση δεδομένων και συμπληρώνει τα κατάλληλα στοιχεία στα πεδία.

4.4.6 Κλάση ShowCars

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση μια φόρμας που περιέχει ένα JTable και εμφανίζει όλα τα αυτοκίνητα που υπάρχουν στη βάση δεδομένων.

Έχουν υλοποιηθεί δύο κατασκευαστές, οι οποίοι χρησιμοποιούνται ανάλογα από ποιο σημείο της εφαρμογής καλείται η κλάση. Αν η κλάση κληθεί από την καρτέλα των αυτοκινήτων για την εμφάνιση των διαθέσιμων αυτοκινήτων θα εμφανίσει ένα JTable με τέσσερις στήλες (Μάρκα, Μοντέλο, Έτος και Κυβικά) για κάθε ένα αυτοκίνητο που υπάρχει στη βάση δεδομένων. Αν γίνει διπλό κλικ σε μια σειρά του JTable τότε θα κληθεί η μέθοδος ShowSelected από την κλάση carsTab (για να εμφανιστεί το αυτοκίνητο που επιλέγει) και θα κλείσει το JFrame που περιέχει την ShowCars.

Αν όμως η κλάση κληθεί από την καρτέλα των πελατών για επιλογή αυτοκινήτου προς αγορά, τότε θα εμφανίσει ένα JTable με όλα τα στοιχεία των αυτοκινήτων. Αν γίνει διπλό κλικ σε μια γραμμή του JTable, σημαίνει ότι έχουμε επιλέξει το συγκεκριμένο αυτοκίνητο για πώληση και θα ενημερωθεί ο πίνακας των αυτοκινήτων και των πελατών της βάσης δεδομένων. Στη

συνέχεια θα ενημερωθεί η καρτέλα του πελάτη και θα κλείσει το JFrame που περιέχει την ShowCars.

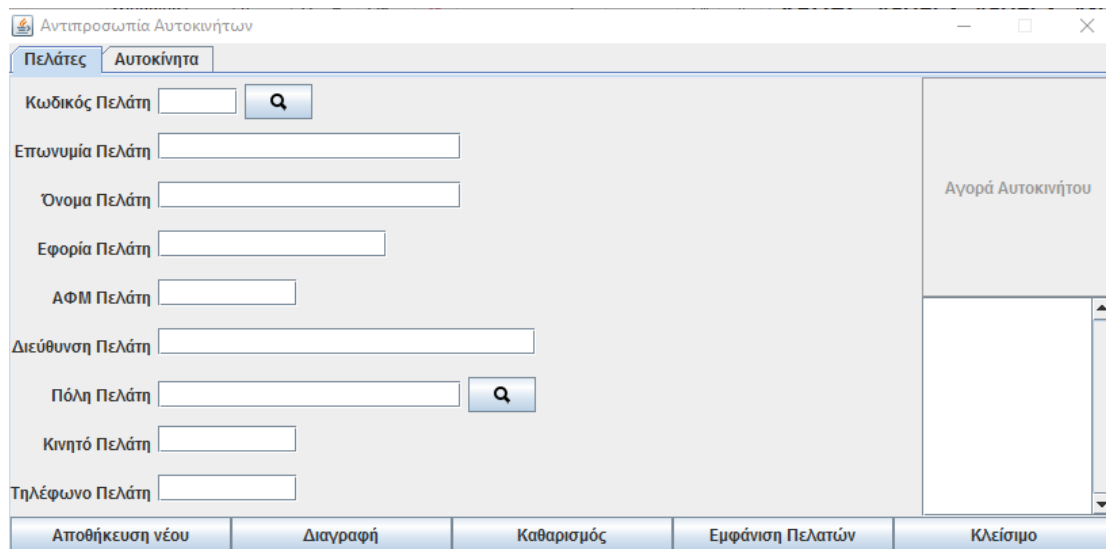
4.4.7 Κλάση MyDialog

Η κλάση αυτή δέχεται τέσσερις παραμέτρους και εμφανίζει ένα JDialog μπροστά από την τρέχουσα φόρμα. Οι παράμετροι που δέχεται είναι το τρέχον JFrame, ο τίτλος και το κείμενο που θα περιέχει καθώς και έναν πίνακα από String που θα είναι τα περιεχόμενα ενός JComboBox.

4.5 Εκτέλεση εφαρμογής

Στην ενότητα αυτή θα παρουσιάσουμε (μέσω screenshot) τα βήματα εκτέλεσης της εφαρμογής.

Όταν εκτελούμε την εφαρμογή, τότε εμφανίζεται η κεντρική φόρμα της.



Εικόνα 29: Βασική φόρμα

Αν συμπληρώσουμε τα στοιχεία κάποιου πελάτη:

Αντιπροσωπία Αυτοκινήτων

Πελάτες Αυτοκίνητα

Κωδικός Πελάτη 2332

Επωνυμία Πελάτη Παππά

Όνομα Πελάτη Δήμητρα

Εφορία Πελάτη Α Αθήνας

ΑΦΜ Πελάτη 989878132

Διεύθυνση Πελάτη Βουραϊκού 20

Πόλη Πελάτη Αθήνα

Κινητό Πελάτη 6987764534

Τηλέφωνο Πελάτη 2109878765

Αποθήκευση νέου Διαγραφή Καθαρισμός Εμφάνιση Πελατών Κλείσιμο

Αγορά Αυτοκινήτου

Εικόνα 30: Συμπλήρωση στοιχείων

Αν πατήσουμε το Αποθήκευση νέου, θα εμφανιστεί το ακόλουθο ενημερωτικό μήνυμα ότι ολοκληρώθηκε η προσθήκη:

Αντιπροσωπία Αυτοκινήτων

Πελάτες Αυτοκίνητα

Κωδικός Πελάτη 2332

Επωνυμία Πελάτη Παππά

Όνομα Πελάτη Δήμητρα

Εφορία Πελάτη Α Αθήνας

ΑΦΜ Πελάτη 989878132

Διεύθυνση Πελάτη Βουραϊκού 20

Πόλη Πελάτη Αθήνα

Κινητό Πελάτη 6987764534

Τηλέφωνο Πελάτη 2109878765

Αποθήκευση νέου Διαγραφή Καθαρισμός Εμφάνιση Πελατών Κλείσιμο

Αγορά Αυτοκινήτου

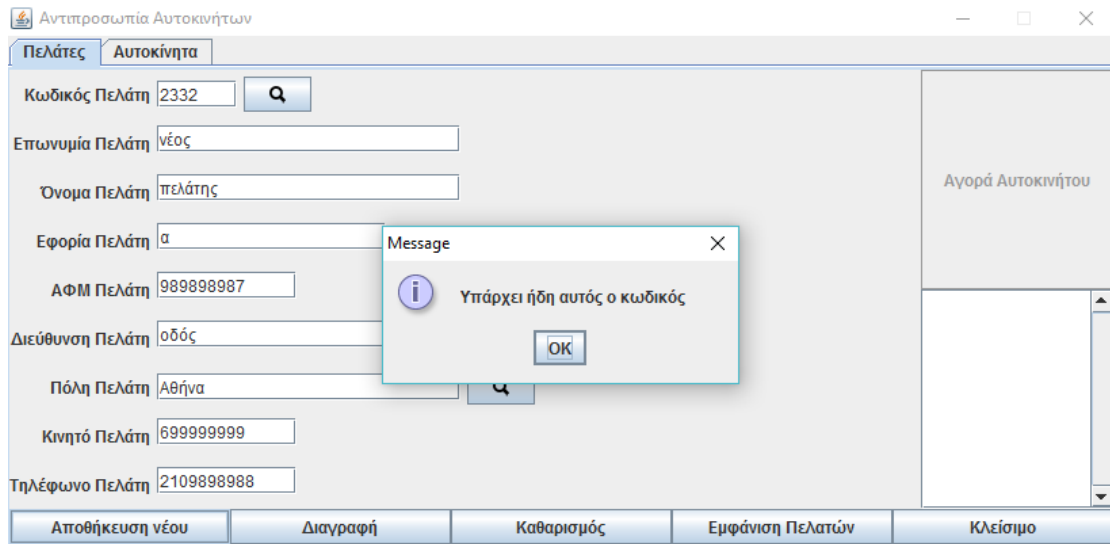
Message

Η προσθήκη ολοκληρώθηκε

OK

Εικόνα 31: Ολοκλήρωση προσθήκης πελάτη

Αν προσπαθήσουμε να εισάγουμε έναν πελάτη με κωδικό που υπάρχει ήδη στη βάση δεδομένων, τότε θα εμφανιστεί μήνυμα ότι υπάρχει ήδη αυτός ο κωδικός στη βάση δεδομένων.



Εικόνα 32: Μήνυμα ότι υπάρχει ήδη ο κωδικός πελάτη

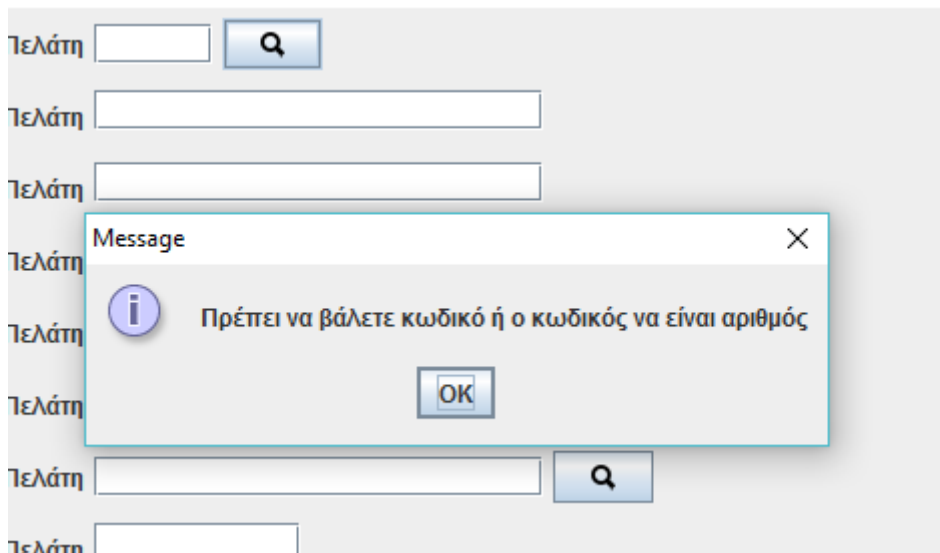
Αν επιλέξουμε το Εμφάνιση πελατών θα εμφανιστεί ένα νέο παράθυρο, το οποίο θα περιέχει τα στοιχεία των πελατών (Id, Επώνυμο, Όνομα και ΑΦΜ) με όλους τους πελάτες της βάσης δεδομένων.

Το παράθυρο αυτό θα παραμένει στην οθόνη μέχρι να το κλείσει ο χρήστης.

Id	Επώνυμο	Όνομα	ΑΦΜ
12	παπαδόπουλος	κώστας1	12378868
123	Ιωάννου	Δημήτρης	123456789
124	Βασίλης	Βασιλείου	123456780
128	Μαρία	Παύλου	988987652
143	Δήμητρα	Γιαννίκου	918787653
147	Αφροδίτη	Δημητρίου	9676765
156	Παπαδημητρίου	Δημήτρης	9876845
158	Παπάς	Γιάννης	97673459
175	Παπαϊωάννου	Παναγιώτα	9768345
12345	Δημήτρης	Δημητρίου	123458712

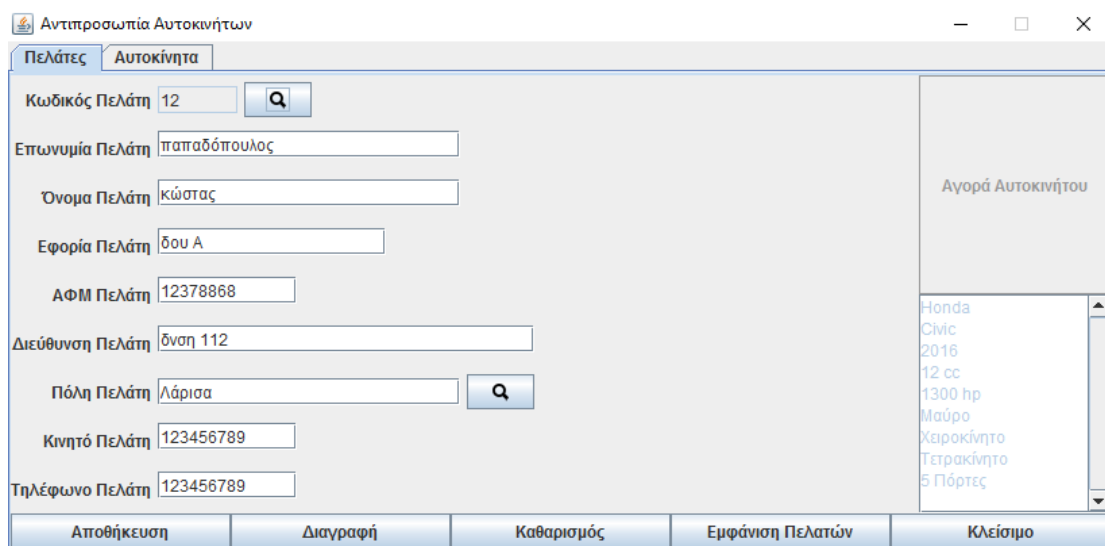
Εικόνα 33: Όλοι οι πελάτες

Αν πατήσουμε το πλήκτρο της αναζήτησης πελάτη από τον κωδικό του, χωρίς να εισάγουμε κωδικό ή έχοντας εισάγει κωδικό που δεν είναι αριθμός, τότε θα εμφανιστεί το ακόλουθο μήνυμα:



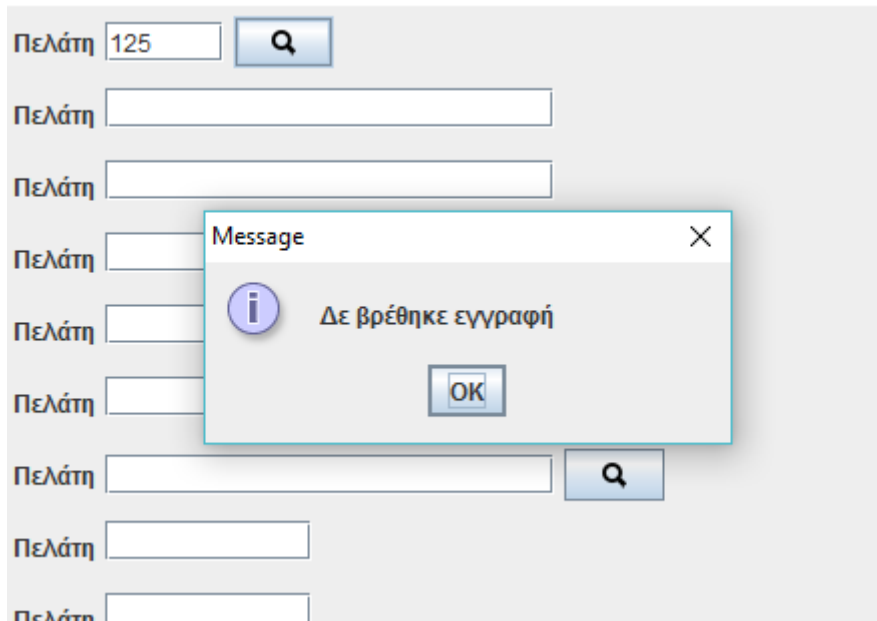
Εικόνα 34: Αποτέλεσμα αναζήτησης χωρίς κωδικό

Αν αναζητήσουμε ένα κωδικό που υπάρχει στη βάση δεδομένων (συμπληρώσουμε τον κωδικό πελάτη στο κατάλληλο πεδίο και πατήσουμε το φακό για αναζήτηση), τότε θα εμφανιστούν τα στοιχεία του πελάτη και θα αλλάξει λεκτικό το κουμπί από προσθήκη νέου σε αποθήκευση.



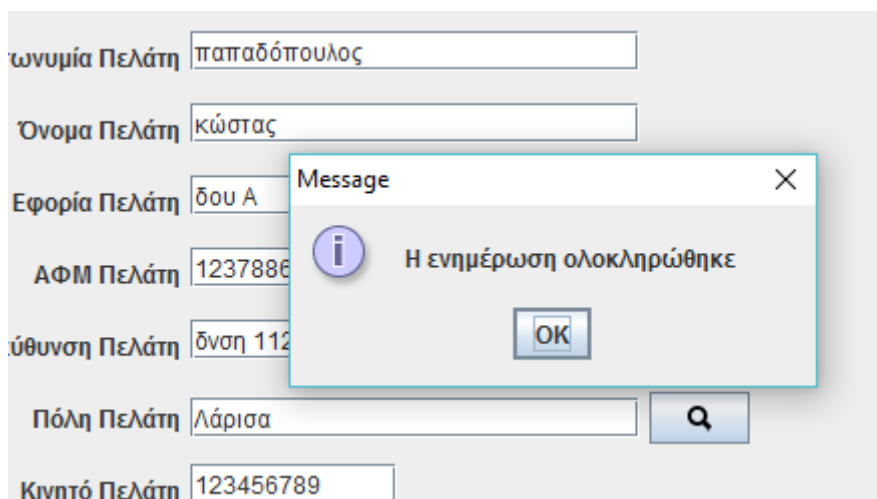
Εικόνα 35: Αποτέλεσμα αναζήτησης

Αν αναζητήσουμε έναν κωδικό πελάτη που δεν υπάρχει στη βάση δεδομένων τότε θα εμφανιστεί κατάλληλο ενημερωτικό μήνυμα:



Εικόνα 36: Αναζήτηση κωδικού που δεν υπάρχει

Αν τροποποιήσουμε την τιμή κάποιου πεδίου και πατήσουμε το Αποθήκευση θα εμφανιστεί ενημερωτικό μήνυμα ότι η ενημέρωση ολοκληρώθηκε.

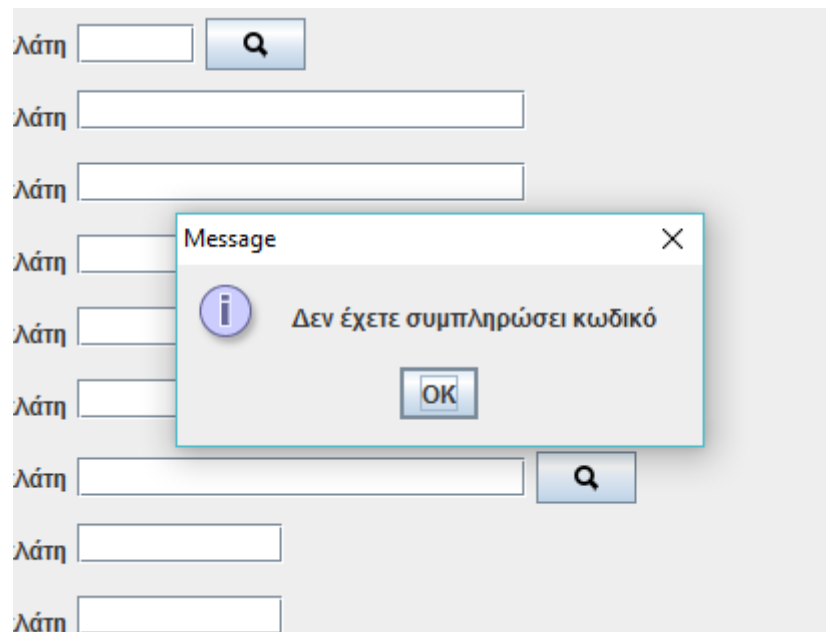


Εικόνα 37: Ολοκλήρωση ενημέρωσης

Αν αναζητήσουμε πάλι τον πελάτη, θα δούμε ότι η ενημέρωση υπάρχει στη βάση δεδομένων.

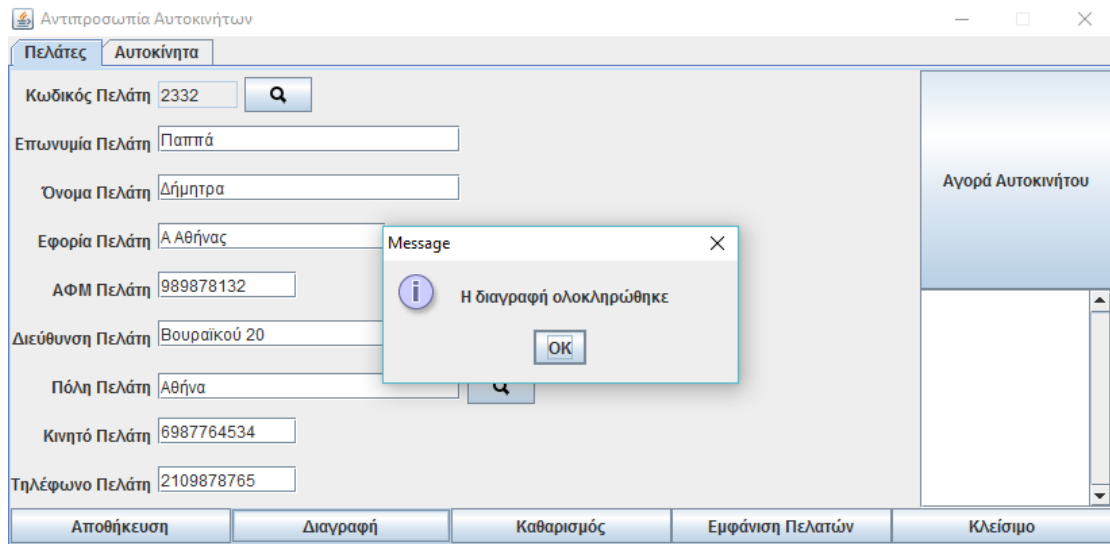
Αν πατήσουμε το κουμπί Διαγραφή, χωρίς να έχει γίνει αναζήτηση σε κάποιο κωδικό τότε θα εμφανιστεί ενημερωτικό μήνυμα ότι δεν έχει

συμπληρωθεί κωδικός πελάτη.



Εικόνα 38: Προσπάθεια για διαγραφή χωρίς κωδικό

Αν έχουμε κάνει Αναζήτηση ενός πελάτη και πατήσουμε στη συνέχεια διαγραφή τότε θα εμφανιστεί ενημερωτικό μήνυμα ότι η διαγραφή ολοκληρώθηκε.



Εικόνα 39: Διαγραφή πελάτη

Για να ελέγξουμε ότι έγινε η διαγραφή στη βάση δεδομένων, μπορούμε να πατήσουμε το Εμφάνιση Πελατών

Id	Επώνυμο	Όνομα	ΑΦΜ
12	παπαδόπουλος	κώστας	12378868
123	Ιωάννου	Δημήτρης	123456789
128	Μαρία	Παύλου	988987652
143	Δήμητρα	Γιαννίκου	918787653
147	Αφροδίτη	Δημητρίου	9676765
156	Παπαδημητρίου	Δημήτρης	9876845
158	Παπάς	Γιάννης	97673459
175	Παπαϊωανου	Παναγιώτα	9768345
12345	Δημήτρης	Δημητρίου	123458712

Εικόνα 40: Έλεγχος διαγραφής

Όπως μπορούμε να δούμε η συγκεκριμένη εγγραφή δεν υπάρχει στη βάση δεδομένων.

Όταν αναζητούμε έναν πελάτη, ενεργοποιείται και το κουμπί Αγορά Αυτοκινήτου. Αν το πατήσουμε, τότε θα εμφανιστούν όλα τα αυτοκίνητα που υπάρχουν στη βάση δεδομένων:

Μάρκα	Μοντέλο	Έτος	Κυβικά	Ίπποι	Χρώμα	Σασμάν	Κίνηση	Πόρτες
Audi	90	2016	120	1800	Κόκκινο	Χειροκίνητο	Τετρακίνητο	5
Hyundai	Accent	2016	9	1300	Κόκκινο	Χειροκίνητο	Προσθιοκίνητο	5
Seat	Ibiza	2015	10	1390	Κόκκινο	Χειροκίνητο	Προσθιοκίνητο	5

Εικόνα 41: Εμφάνιση αυτοκινήτων για αγορά

Κάνοντας διπλό κλικ σε μια γραμμή τότε δηλώνουμε ότι έχει επιλεγεί το συγκεκριμένο αυτοκίνητο για αγορά. Για παράδειγμα με διπλό κλικ πάνω στο Seat θα δούμε:

Μάρκα	Μοντέλο	Έτος	Κυβικά	Ίπποι	Χρώμα	Σασμάν	Κίνηση	Πόρτες
Audi	90	2016	120	1800	Κόκκινο	Χειροκίνητο	Τετρακίνητο	5
Hyundai	Accent	2016	9	1300	Κόκκινο	Χειροκίνητο	Προσθιοκίνητο	5
Seat	Ibiza	2015	10	1390	Κόκκινο	Χειροκίνητο	Προσθιοκίνητο	5

Message ×

i Η πώληση ολοκληρώθηκε

Εικόνα 42: Πώληση αυτοκινήτου

Όταν πατήσουμε το OK τα στοιχεία του αυτοκινήτου έχουν μεταφερθεί στην καρτέλα του πελάτη.

Εικόνα 43: Ενημέρωση καρτέλας πελάτη μετά από αγορά

Αν επιλέξουμε την καρτέλα αυτοκίνητα τότε θα εμφανιστεί μια φόρμα όπου μπορούμε να συμπληρώσουμε τις πληροφορίες που θέλουμε και να προσθέσουμε, διαγράψουμε, εμφανίσουμε τα αυτοκίνητα που υπάρχουν στη βάση δεδομένων μας.

Εικόνα 44: Καρτέλα αυτοκίνητα

Αν επιλέξουμε το Εμφάνιση Αυτοκινήτων θα εμφανιστεί ένα νέο παράθυρο με όλα τα διαθέσιμα αυτοκίνητα.

Μάρκα	Μοντέλο	Έτος	Κυβικά
Audi	90	2016	120
Hyundai	Accent	2016	9

Εικόνα 45: Εμφάνιση διαθέσιμων αυτοκινήτων

Αν κάνουμε διπλό κλικ σε μια γραμμή του θα εμφανιστούν τα στοιχεία του επιλεγμένου αυτοκινήτου στην καρτέλα των αυτοκινήτων.

Εικόνα 46: Εμφάνιση στοιχείων αυτοκινήτων

4.6 Εγκατάσταση απαιτούμενων προγραμμάτων

Για την συγγραφή και εκτέλεση του προγράμματος εγκαταστάθηκαν στον υπολογιστή τα ακόλουθα προγράμματα:

- Java
- Netbeans
- Xampp

Για να μπορέσουμε να εκτελέσουμε το πρόγραμμα θα πρέπει να κατεβάσουμε το Java Runtime Environment. Αυτό μπορεί να γίνει από τη σελίδα και να επιλέξουμε JRE. Στη συνέχεια πρέπει να επιλέξουμε το λειτουργικό σύστημα του υπολογιστή μας (Linux, Solaris, Windows Mac) και να κατεβάσουμε τα αντίστοιχα πακέτα. Με την εκτέλεσή του θα εγκατασταθεί το Java Runtime Environment στον υπολογιστή μας και θα μπορούμε να τρέχουμε αρχεία jar.

Όσον αφορά το NetBeans έχουμε αναφέρει πληροφορίες στο σχετικό κεφάλαιο.

Το XAMPP είναι ένα πακέτο προγραμμάτων, ανεξαρτήτου πλατφόρμας, ελεύθερου - ανοικτού λογισμικού, το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων HTTP Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για σενάρια γραμμένα σε γλώσσες προγραμματισμού PHP και Perl. Το όνομά του είναι ακρωνύμιο των :

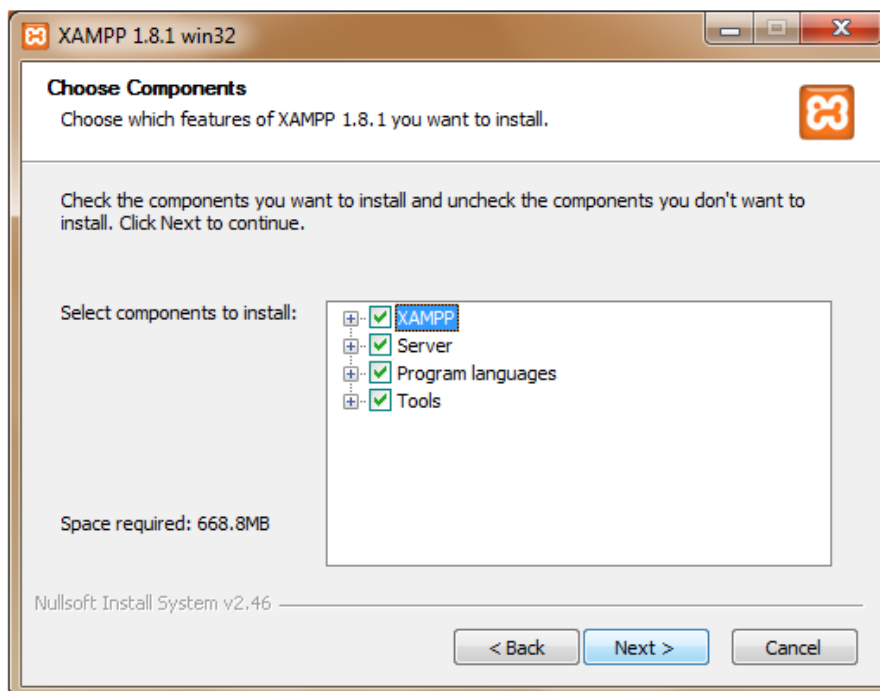
- X (cross-platform δηλαδή λειτουργεί σε πολλές πλατφόρμες)
- Apache HTTP Server
- MySQL
- PHP
- Perl

Το XAMPP διατίθεται δωρεάν από την σελίδα <http://www.apachefriends.org> για διάφορα λειτουργικά συστήματα (Linux, Windows, Solaris, Mac). Στις ακόλουθες εικόνες θα περιγράψουμε τα στάδια εγκατάστασης του xampp.



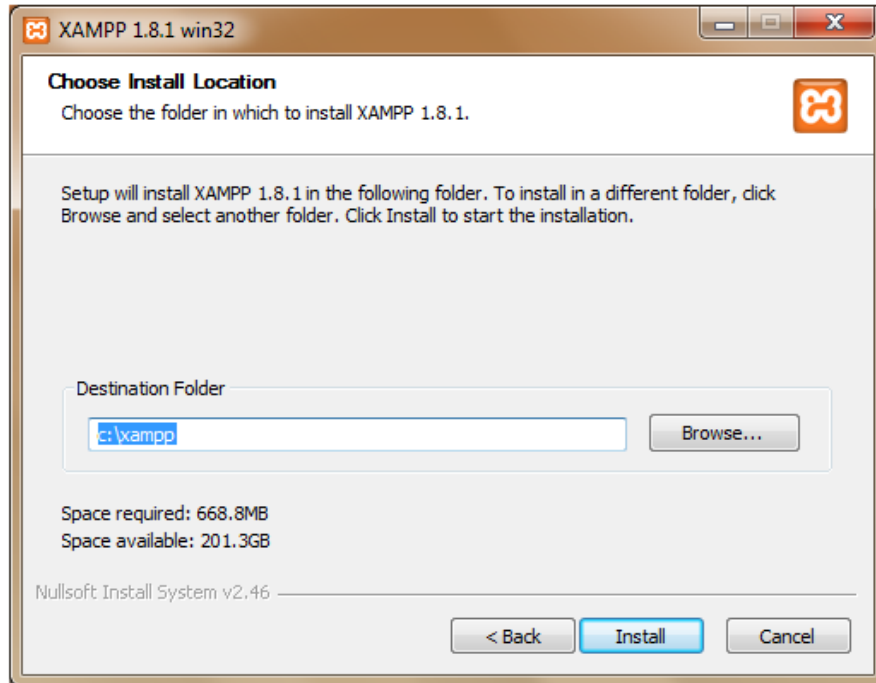
Εικόνα 47: Αρχική οθόνη εγκατάστασης

Πατώντας το κουμπί Next θα μεταβούμε στην επόμενη οθόνη όπου θα πρέπει να επιλέξουμε τί θέλουμε να εγκαταστήσουμε από το xampp.



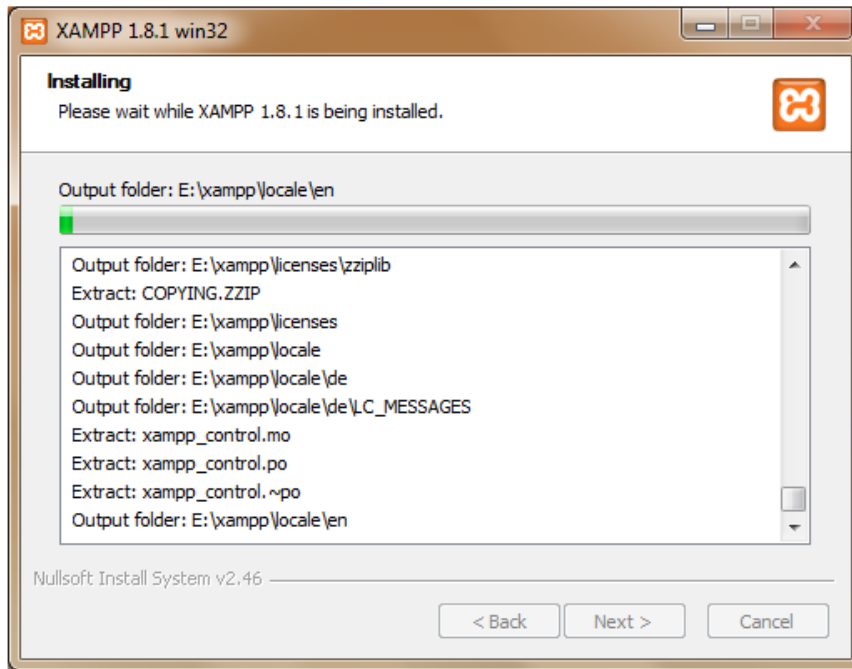
Εικόνα 48: 2ο βήμα εγκατάστασης

Αν πατήσουμε το κουμπί Next θα μεταβούμε στο επόμενο βήμα όπου θα επιλέξουμε τον κατάλογο στον οποίο θα γίνει η εγκατάσταση.



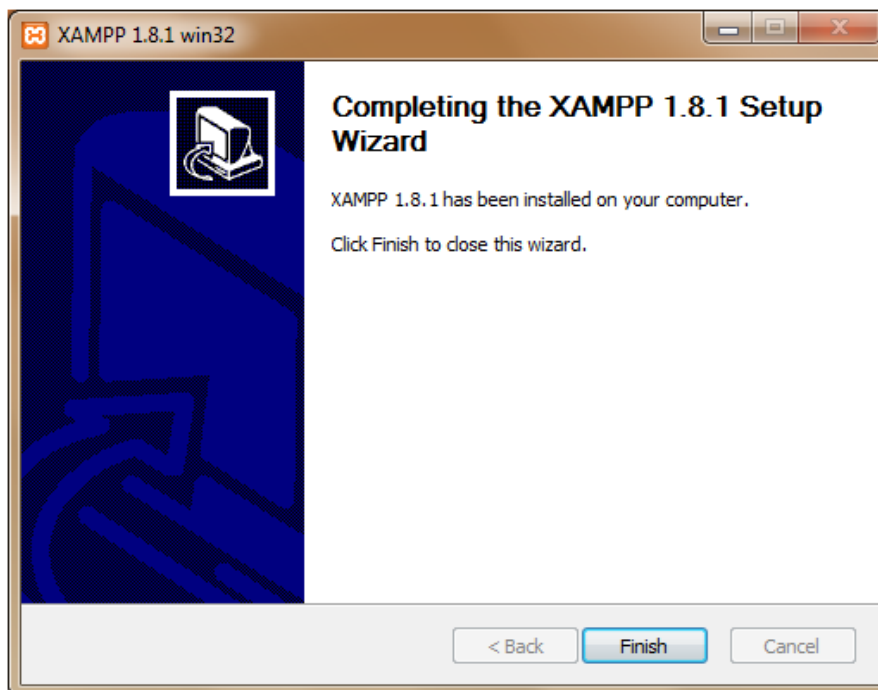
Εικόνα 49: 3ο βήμα

Πατώντας το κουμπί Install ξεκινά η εγκατάσταση, ένα στιγμιότυπό της μπορούμε να δούμε στην οθόνη που ακολουθεί:



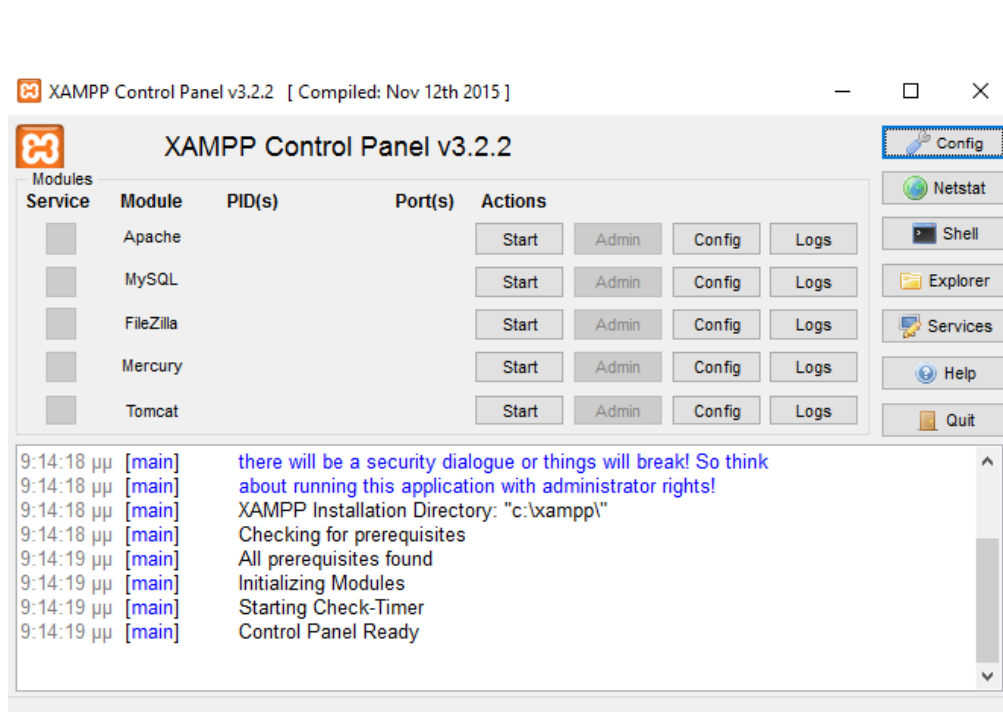
Εικόνα 50: 4ο βήμα εγκατάστασης

Όταν ολοκληρωθεί η εγκατάσταση, θα δούμε ένα ενημερωτικό μήνυμα στην οθόνη μας.



Εικόνα 51: Ολοκλήρωση εγκατάστασης

Όταν ξεκινήσουμε την εφαρμογή θα δούμε την ακόλουθη εικόνα, όπου μπορούμε να καθορίσουμε τα στοιχεία που θα ξεκινήσουμε (Apache για πρόσβαση μέσω web interface της βάσης δεδομένων και τη MySQL).



Εικόνα 52: Αρχική οθόνη Xampp

Κεφάλαιο 5ο Συμπεράσματα – Μελλοντική Εργασία

5.1 Συμπεράσματα

Με την παρούσα πτυχιακή εργασία κάναμε το πρώτο βήμα για τη δημιουργία ενός πληροφοριακού συστήματος για μια αντιπροσωπία αυτοκινήτων με χρήση Java. Μετά από την υλοποίηση και τον έλεγχο της εφαρμογής διαπιστώσαμε ότι καλύψαμε πλήρως τις απαιτήσεις που είχαν τεθεί. Οι απαιτήσεις αυτές περιλάμβαναν:

- Την αποθήκευση σε μια βάση δεδομένων των ακόλουθων στοιχείων πελατών:
 - Κωδικός
 - Επωνυμία
 - Όνομα
 - Διεύθυνση
 - Εφορία
 - ΑΦΜ
 - Πόλη
 - Κινητό Τηλέφωνο
 - Τηλέφωνο
- Την αποθήκευση σε μια βάση δεδομένων των ακόλουθων στοιχείων των αυτοκινήτων:
 - Μάρκα
 - Μοντέλο
 - Έτος
 - Κυβικά

- Ίππιοι
 - Χρώμα
 - Σασμάν
 - Κίνηση
 - Πόρτες
- Τη δυνατότητα προσθήκης νέων πελάτων και αυτοκινήτων
 - Τη δυνατότητα αναζήτησης πελατών και αυτοκινήτων και εμφάνισης των στοιχείων τους
 - Τη δυνατότητα διόρθωσης των στοιχείων των πελατών και αυτοκινήτων και αποθήκευσής τους στη βάση δεδομένων
 - Τη δυνατότητα διαγραφής πελατών και αυτοκινήτων
 - Τη δυνατότητα εμφάνισης όλων των πελατών και αυτοκινήτων από τη βάση δεδομένων
 - Τη δυνατότητα αγοράς αυτοκινήτου από πελάτη

5.2 Μελλοντική Εργασία

Μετά από την ολοκλήρωση του πρώτου βήματος και τη δημιουργία ενός λειτουργικού προγράμματος, το επόμενο βήμα θα μπορούσε να προσφέρει περισσότερες δυνατότητες οι οποίες να πλησιάζουν αυτές των εμπορικών εφαρμογών:

- Προσθήκη περισσότερων στοιχείων πελάτη και προσθήκη περισσότερων δυνατοτήτων αναζήτησης π.χ. ΑΦΜ, επωνυμία κ.λπ.
- Προσθήκη οικονομικών στοιχείων στην καρτέλα του πελάτη

- Εμφάνιση φωτογραφιών αυτοκινήτων στην καρτέλα του πελάτη
- Τήρηση ιστορικού αγοράς για κάθε πελάτη και δυνατότητα σύνδεσης με συνεργείο αυτοκινήτων ή ασφαλιστική εταιρεία
- Κράτηση παραγγελίας και ενημέρωσης πελάτη με email όταν το αυτοκίνητο είναι διαθέσιμο

Βιβλιογραφία

apache.org (n.a) Java Persistence API, διαθέσιμο από http://openjpa.apache.org/builds/1.2.3/apache-openjpa/docs/jpa_overview.html

Buytaert D. (2016) The history of MySQL AB, διαθέσιμο από

DataBase Friends (2014) History of MySQL, διαθέσιμο από <http://databasefriends.blogspot.gr/2014/02/history-of-mysql.html>

Dybka P. (2014) MySQL: Why My?, διαθέσιμο από <http://www.vertabelo.com/blog/notes-from-the-lab/mysql-history>

Greanier T. (2005) Java Εισαγωγή στη Σύγχρονη Τεχνολογία, Αθήνα: Μ. Γκιούρδας

Jag (2010) Duke, the Java mascot, διαθέσιμο από

JavaTpoint (n.a) History of Java, διαθέσιμο από <http://www.javatpoint.com/history-of-java#>

Mohammed, A. (2014) EclipseLink – JPA Installation And Configuration Using Eclipse, διαθέσιμο από <http://www.javabeat.net/eclipselink-jpa-installation-configuration/>

Ostema, R.P. (2004) The History Of Databases, διαθέσιμο από people.cis.ksu.edu/~hankley/d560/Topics/Rogers-Ostema_History.ppt

tutorialspoint.com (n.a) JPA Tutorial, διαθέσιμο από <http://www.tutorialspoint.com/jpa/index.htm>

Oracle (n.a.) Learn About Java Technology, <https://java.com/en/about/>

Oracle (2014) Duke, the Java mascot, <https://kenai.com/projects/duke/pages/Home>

Oracle (2015) About the Java Technology, <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>

Oracle (2016a) A Brief History of NetBeans, διαθέσιμο από

Oracle (2016b) NetBeans IDE - The Smarter and Faster Way to Code,
διαθέσιμο από

Miller R. & Kasparian R. (2006), Java For Artists, Virginia: Pulp Free
Press