



## **Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**

**Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



### **Πτυχιακή εργασία**

**Τίτλος: Ανάπτυξη Εφαρμογής Για Επιχείρηση  
Συντήρησης Φίλτρων Καθαρισμού Νερού**

**Κωνσταντίνος Καλαφατάκης (ΑΜ: 1399)**

**Επιβλέπων καθηγητής : Χατζάκης Ηλίας**

**Επιτροπή Αξιολόγησης : Μανιφάβας Χαράλαμπος**

**Παπαδάκης Νικόλαος**

**Ημερομηνία παρουσίασης: 30/11/2011**



## **Abstract**

The subject of this paper is to develop an application to serve the needs of a water filter company. The application gives you the opportunity to manage purchases, sales and maintenance of water filters. More specifically, the potentials of the application are:

- Management of the business clients
- Sales management of water filter equipment
- Management of company's products
- Management of water filter maintenance
- To do the database backup and restore.

The application was developed in Sun Java Net Beans a language that provides independence of the operating system. The database was made in MySQL a very popular database managing system open code that runs in more than 20 platforms.

## Σύνοψη

Το αντικείμενο της Πτυχιακής Εργασίας είναι η ανάπτυξη εφαρμογής για την εξυπηρέτηση αναγκών εταιρίας φίλτρων νερού. Η εφαρμογή δίνει τη δυνατότητα διαχείρισης πωλήσεων και συντηρήσεων φίλτρων νερού. Πιο συγκεκριμένα οι δυνατότητες που παρέχει η εφαρμογή είναι:

- Διαχείρισης πελατών της επιχείρησης
- Διαχείριση πωλήσεων συστημάτων-συσκευών φίλτρων νερού
- Διαχείριση προϊόντων εταιρίας
- Διαχείριση - οργάνωση συντηρήσεων φίλτρων νερού
- Δημιουργία αντιγράφων ασφαλείας και επαναφορά την βάση δεδομένων.

Η ανάπτυξη της εφαρμογής έγινε σε Sun Java με Net Beans μια γλώσσα που μπορεί να εκτελεστεί από οποιοδήποτε λειτουργικό σύστημα. Η βάση δεδομένων έγινε σε MySQL ένα δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα η οποία λειτουργεί σε περισσότερες από 20 πλατφόρμες.

# Πίνακας Περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>1</b>
1.1	Περίληψη.....	3
1.2	Κίνητρο για την Διεξαγωγή της Εργασίας.....	3
1.3	Σκοπός και Στόχοι Εργασίας.....	4
1.4	Δομή Εργασίας.....	4
<b>2</b>	<b>Αντικειμενοστραφής προγραμματισμός- Μεθοδολογία Υλοποίησης.....</b>	<b>5</b>
2.1	Εισαγωγή στον αντικειμενοστραφή προγραμματισμό.....	5
2.1.1	Χαρακτηριστικά αντικειμενοστραφούς προγραμματισμού.....	5
2.2	Java.....	6
2.3	Τι κάνει τη Java να ξεχωρίζει.....	7
2.4	Περιβάλλον ανάπτυξης και δημιουργία εφαρμογής.....	8
2.5	Δομή ενός προγράμματος σε Java.....	10
2.5.1	Τα αντικείμενα της Java.....	11
2.5.2	Ορισμός κλάσεων στην Java.....	11
2.6	Κατασκευαστές (constructors).....	12
2.7	Καταστροφή αντικειμένων (Finalization).....	12
2.8	Προσδιοριστές πρόσβασης (access specifiers).....	12
2.9	Τροποποιητές (Modifiers).....	13
2.10	Διασυνδέσεις (Interfaces).....	13
2.11	Εξαιρέσεις (Exceptions).....	14
2.12	Τα βασικά των Applets στην Java.....	14
2.12.1	Βασικές λειτουργίες των Applets.....	15
2.13	Abstract Windowing Toolkit (AWT).....	16
2.13.1	Βασικά στοιχεία ενός GUI.....	16
2.13.2	Δομή ενός GUI.....	16
2.14	Τα εργαλεία της Java.....	17
2.15	Τι είναι το εργαλείο NetBeans.....	17
<b>3</b>	<b>Βάσεις Δεδομένων.....</b>	<b>19</b>
3.1	Τα τρία βασικά μοντέλα.....	21
3.1.1	Το Ιεραρχικό Μοντέλο Βάσεων Δεδομένων.....	21
3.1.2	Το Δικτυωτό Μοντέλο Βάσεων Δεδομένων.....	22
3.1.3	Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων.....	22
3.2	Στάδια Σχεδίασης - Μοντέλο Οντοτήτων – Συσχετίσεων.....	22
3.2.1	Οι Οντότητες.....	23
3.2.2	Οι Ιδιότητες (Χαρακτηριστικά) των Οντοτήτων.....	23
3.2.3	Τα κλειδιά.....	23
3.2.4	Συσχετίσεις.....	24
3.2.5	Χαρακτηριστικά τύπων συσχέτισης.....	24
3.2.6	Αδύναμοι τύποι οντότητας (Weak Entity Types).....	24
3.3	Χρήστες και δικαιώματα.....	25
3.4	Η αρχιτεκτονική των ΣΔΒΔ.....	25
3.5	Τα σχεσιακά ΣΔΒΔ (RDBMS).....	26
<b>4</b>	<b>My SQL.....</b>	<b>27</b>
4.1	Γιατί MySQL.....	27
4.2	Δημιουργία Βάσης.....	29
4.2.1	Εισαγωγή Δεδομένων.....	30
4.2.2	Επιλογή Δεδομένων.....	30
4.2.3	Ενημερώσεις και Διαγραφές.....	30
4.3	JDBC.....	31

4.3.1	Χρήση του Οδηγού.....	31
<b>5</b>	<b>Ανάλυση λογισμικού.....</b>	<b>32</b>
5.1	Περιγραφή Συστήματος.....	32
5.2	Απαιτήσεις.....	33
5.3	Περιπτώσεις Χρήσης και Σενάρια.....	39
5.4	Σχεδίαση Βάσης Δεδομένων.....	40
5.4.1	Διάγραμμα Βάσης Δεδομένων.....	41
5.4.2	Δημιουργία Βάσης Δεδομένων.....	42
5.5	Διάγραμμα κλάσεων.....	50
5.6	Οι Κλάσεις και οι σημαντικότερες μέθοδοι.....	54
<b>6</b>	<b>Οδηγίες χρήσεως της εφαρμογής.....</b>	<b>59</b>
<b>7</b>	<b>Αποτελέσματα.....</b>	<b>72</b>
7.1	Συμπεράσματα.....	72
7.2	Μελλοντική εργασία και επεκτάσεις.....	73
	<b>Βιβλιογραφία.....</b>	<b>74</b>
	<b>ΠΑΡΑΡΤΗΜΑ.....</b>	<b>75</b>
	<b>ΠΑΡΑΡΤΗΜΑ Α : Εγκαταστάσεις προγραμμάτων σε λειτουργικά.....</b>	<b>76</b>
	Παράρτημα Α1: Εγκαθιστώντας τη java σε Windows.....	77
	Παράρτημα Α2: Εφαρμογή του NetBeans.....	78
	Παράρτημα Α3: Εγκατάσταση του NetBeans σε Windows.....	83
	Παράρτημα Α4: Εγκατάσταση του NetBeans σε Ubuntu.....	85
	Παράρτημα Α5: Downloading MySQL.....	88
	Παράρτημα Α6: Εγκατάσταση της MySQL σε Windows.....	89
	Παράρτημα Α7: Εγκατάσταση της MySQL σε Ubuntu.....	93
	<b>ΠΑΡΑΡΤΗΜΑ Β : Κώδικας Εφαρμογής.....</b>	<b>96</b>
	<b>ΠΑΡΑΡΤΗΜΑ Γ : Παρουσίαση .....</b>	<b>138</b>

## Πίνακας Εικόνων

### Κεφάλαιο 2

Εικόνα 2.α : Επισκόπηση της διαδικασίας ανάπτυξης λογισμικού.....	8
Εικόνα 2.β : Μέσω της Java VM, η ίδια αίτηση μπορεί να λειτουργεί σε πολλαπλές πλατφόρμες.....	9
Εικόνα 2.γ : Το API και Java Virtual Machine μονώνουν το πρόγραμμα από το υποκείμενο υλικό.....	10

### Κεφάλαιο 5

Εικόνα 5. α : Διάγραμμα Βάσης Δεδομένων της εφαρμογής.....	41
Εικόνα 5. β: Πίνακας νομοί.....	42
Εικόνα 5. γ :Περιεχόμενα πίνακα νομοί.....	42
Εικόνα 5. δ : Πίνακας customers.....	43
Εικόνα 5. ε : Περιεχόμενα πίνακα customers.....	44
Εικόνα 5. ζ: Πίνακας products.....	45
Εικόνα 5. η: Περιεχόμενα πίνακα products.....	45
Εικόνα 5. θ: Πίνακας customersproducts.....	46
Εικόνα 5. ι: Περιεχόμενα πίνακα customersproducts.....	47
Εικόνα 5. κ: Πίνακας sintirisis.....	48
Εικόνα 5. λ: Περιεχόμενα πίνακα sintirisis.....	49
Εικόνα 5. μ: Διάγραμμα Κλάσεων.....	51
Εικόνα 5. ν: Κλάσεις Διαγράμματος α.....	52
Εικόνα 5. ξ: Κλάσεις Διαγράμματος β.....	53

### Κεφάλαιο 6

Εικόνα 6. α : Καρτέλα συσκευές-προϊόντα.....	60
Εικόνα 6. β : Καρτέλα πελάτες και οι συσκευές τους.....	61
Εικόνα 6. γ : Παράθυρο προσθήκης πελάτη.....	62
Εικόνα 6. δ : Παράθυρο προσθήκης συσκευής του πελάτη.....	64
Εικόνα 6. ε : Παράθυρο προσθήκης εργασιών συντήρησης.....	66
Εικόνα 6. ζ : Καρτέλα εργασίες συντήρησης.....	67
Εικόνα 6. η : Παράθυρο φόρμας εκτύπωσης.....	68
Εικόνα 6. θ : Drop down μενού.....	69
Εικόνα 6. ι : Προειδοποιητικό μήνυμα αποθήκευσης backup.....	69
Εικόνα 6. κ : Παράθυρο ολοκλήρωσης backup.....	69
Εικόνα 6. λ : Προειδοποιητικό μήνυμα βημάτων επαναφοράς.....	70
Εικόνα 6. μ : Παράθυρο ολοκλήρωσης επαναφοράς.....	70
Εικόνα 6. ν : Μενού Βοήθεια.....	71
Εικόνα 6. ξ : Παράθυρο πληροφοριών κατασκευαστή.....	71

## ΠΑΡΑΡΤΗΜΑ

### Παράρτημα Α2

Εικόνα Α2.α : Περιβάλλον του NetBeans μετά την δημιουργία ενός project.....	79
Εικόνα Α2.β : Παράθυρο σύνταξης κώδικα.....	80
Εικόνα Α2.γ : Επιλογή Compile του προγράμματος.....	81
Εικόνα Α2.δ : Παράθυρο αποτελέσματος ελέγχου.....	81
Εικόνα Α2.ε : Φάκελος αποθηκευμένων αρχείων.....	82

### Παράρτημα Α3

Εικόνα Α3.α : Εικονίδιο του εργαλείου NetBeans.....	84
---	----

### Παράρτημα Α4

Εικόνα Α4.α : Παράθυρα Terminal και Accessories μενού.....	85
Εικόνα Α4.β : Διαδικασία αλλαγής δικαιωμάτων.....	86
Εικόνα Α4.γ : Παράθυρο Terminal του Ubuntu.....	86

### Παράρτημα Α6

Εικόνα Α6.α : Προετοιμασία εγκατάστασης.....	89
Εικόνα Α6.β : Παράθυρο διαδικασίας εγκατάστασης του προγράμματος.....	90
Εικόνα Α6.γ : Διαδικασία επικοινωνίας μας με την MySQL.....	92

### Παράρτημα Α7

<i>Εικόνα Α7.α : Παράθυρο του Update Manager.....</i>	<i>93</i>
<i>Εικόνα Α7.β: Διαδικασία εγκατάστασης της MySQL.....</i>	<i>94</i>
<i>Εικόνα Α7.γ : Διαδικασία επιβεβαίωσης του συστήματος για τον κωδικό πρόσβασης.....</i>	<i>94</i>
<i>Εικόνα Α7.δ : Ενεργοποίηση ενός session.....</i>	<i>95</i>

## **Λίστα Πινάκων**

<i>Πίνακας 2.13.1.α : Περιγραφή των κλάσεων για την κατασκευή GUI-components.....</i>	<i>16</i>
<i>Πίνακας 2.13.2.α : Περιγραφή των κλάσεων των Layout Managers.....</i>	<i>17</i>
<i>Πίνακας 5.3.α: Περιπτώσεις Χρήσης και Σενάρια.....</i>	<i>39</i>
<i>Πίνακας 5.5 α: Επεξήγηση συμβόλων πρόσβασης διαγράμματος.....</i>	<i>50</i>
<i>Πίνακας 5.5 β: Επεξήγηση συμβόλων πολλαπλότητας διαγράμματος.....</i>	<i>50</i>
<i>Πίνακας 5.6 α: Σύνοψη κλάσεων.....</i>	<i>54</i>



## 1. Εισαγωγή

Η αλματώδης ανάπτυξη των πληροφοριακών συστημάτων και των δυνατοτήτων τους θέτει νέες απαιτήσεις από τα προγραμματιστικά εργαλεία και δημιουργεί απαιτήσεις για ένα νέο και συνεχώς αυξανόμενο σύνολο εφαρμογών. Επειδή την ανάπτυξη αυτή ακολουθούν και οι διάφορες κατασκευάστριες εταιρίες υπολογιστικών συστημάτων, οι εφαρμογές που κατασκευάζονται θα πρέπει να λειτουργούν σε πολλές μηχανές και λειτουργικά συστήματα με την ίδια αξιοπιστία και λειτουργικότητα.

Το αναπτυσσόμενο λογισμικό απαιτείται συνήθως να λειτουργεί σωστά σε πολλά διαφορετικά περιβάλλοντα και αρχιτεκτονικές υπολογιστών, καθώς και να συνεργάζεται με πολλές εφαρμογές. Το λογισμικό πρέπει να εκμεταλλεύεται τις δυνατότητες της σύγχρονης δομής υπολογιστικών συστημάτων και να είναι σε θέση να προσπελαύνει τις καταναμημένες πηγές πληροφοριών, να συνδυάζει τις αντλούμενες πληροφορίες και να τις παρουσιάζει στο χρήστη σε κατάλληλη μορφή.

Το πρόβλημα που δημιουργείται είναι ότι οι απαιτήσεις για μεταφερσιμότητα είναι συνήθως αντικρουόμενες, ενώ στο θέμα της ασφάλειας δεν έχει δοθεί η απαραίτητη σημασία. Υπάρχουν γλώσσες προγραμματισμού οι οποίες είναι μεταφέρσιμες αλλά αργές λόγω του ότι τα προγράμματα ερμηνεύονται αντί να μεταγλωττίζονται. Οι γλώσσες αυτές είναι αρκετά διαδεδομένες τόσο λόγω της υψηλής λειτουργικότητάς τους όσο και τη μεταφερσιμότητά τους. Επίσης, υπάρχουν γλώσσες προγραμματισμού οι οποίες είναι γρήγορες αλλά η ταχύτητά τους απορρέει από το ότι είναι σχεδιασμένες για συγκεκριμένες υπολογιστικές αρχιτεκτονικές.

Από μεθοδολογικής πλευράς η ανάπτυξη λογισμικού τα τελευταία χρόνια έχει προσανατολιστεί κυρίως προς τον αντικειμενοστραφή προγραμματισμό (object-oriented programming), ο οποίος επιχειρεί να δαμάσει τη συνεχώς αυξανόμενη πολυπλοκότητα ανάπτυξης λογισμικού. Σύμφωνα με τον αντικειμενοστραφή προγραμματισμό το λογισμικό δομείται σε αυτόνομες μονάδες οι οποίες έχουν σαφή λειτουργικότητα και διαπροσωπεία.

Η Java είναι μια γλώσσα προγραμματισμού και επιχειρεί να δώσει λύση στα προβλήματα που αναφέρθηκαν παραπάνω. Αναπτύχθηκε από την εταιρεία Sun και στο χρόνο από την ανάπτυξή της έχει γνωρίσει αρκετά μεγάλη διάδοση. Αρχικά ήταν προσανατολισμένη στην ανάπτυξη λογισμικού για ηλεκτρονικές συσκευές οικιακής χρήσης, στη συνέχεια όμως εξελίχθηκε σε μια ολοκληρωμένη γλώσσα η οποία έχει αρκετά από τα χαρακτηριστικά των μοντέρνων γλωσσών προγραμματισμού και υποστηρίζει τον αντικειμενοστραφή προγραμματισμό. Η επιτυχία της γλώσσας έγκειται στο ότι μπορεί να χρησιμοποιηθεί για προγραμματισμό ασφαλών και υψηλής απόδοσης εφαρμογών οι οποίες μπορούν να τρέξουν αυτούσιες σε διαφορετικά προγραμματιστικά περιβάλλοντα και αρχιτεκτονικές, καθώς και ότι παρέχει τη δυνατότητα μεταφοράς δυναμικού περιεχομένου σε εφαρμογές πολυμέσων. Επίσης ένας πολύ σημαντικός παράγοντας της Java είναι ότι διατίθεται δωρεάν.

Σκοπός όμως της java δεν είναι η διατήρηση μεγάλου όγκου πληροφοριών, γι' αυτό το λόγο έχουν δημιουργηθεί άλλοι μηχανισμοί για να γίνει αυτή η διεργασία, όπως είναι οι βάσεις δεδομένων που με τον όρο αυτό εννοούμε ένα σύστημα τήρησης εγγράφων με υπολογιστή, που ο γενικός σκοπός του είναι να τηρεί πληροφορίες και να δίνει αυτές τις πληροφορίες όταν του ζητούνται. Οι πληροφορίες που τηρούνται σε ένα τέτοιο σύστημα μπορεί να είναι οτιδήποτε χρειάζεται για την υποβοήθηση των εργασιών ενός ατόμου ή οργανισμού.

Ένας από τους πιο αξιόπιστους τρόπους καταγραφής πληροφοριών, μπορεί να γίνει από την **MySQL** η οποία είναι η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα λόγω της αυξημένης απόδοσης και ευκολίας στην διαχείριση της.

Θεωρείται μια από τις πιο αξιόπιστες πλατφόρμες για την συγκέντρωση δεδομένων. Καθίσταται η απόλυτη επιλογή μιας νέας γενιάς εφαρμογών που υλοποιούνται σε περιβάλλον LAMP (Linux, Apache, MySQL, PHP / Perl / Python).

Η MySQL ανήκοντας στην κατηγορία των DBMS μπορεί να αναλάβει την διαχείριση πολλών βάσεων μαζί.

Όλα τα παραπάνω εργαλεία χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής που αναφέρεται στην εξυπηρέτηση αναγκών εταιρίας φίλτρων νερού.

Η σωστή λειτουργία μιας εταιρίας είναι ένα ζήτημα που ανέκαθεν απασχολούσε τους επαγγελματίες και θα συνεχίσει να τους απασχολεί και στο μέλλον. Το γεγονός αυτό σε συνδυασμό με την προσωπική εμπειρία στον εργασιακό χώρο και τις γνώσεις που απέκτησα από τη σχολή

Εφαρμοσμένης Πληροφορικής & Πολυμέσων του ΤΕΙ Κρήτης, μου έδωσαν το έναυσμα ώστε να διεκπεραιώσω τη συγκεκριμένη πτυχιακή εργασία.

Αν και σήμερα κυκλοφορεί ένα τεράστιο πλήθος εφαρμογών που είναι φτιαγμένες για να εξυπηρετούν και να διευκολύνουν τις λειτουργικές ανάγκες των επιχειρήσεων πολλές φορές δεν είναι αρκετές για να τις καλύψουν πλήρως. Η συγκεκριμένη πτυχιακή εργασία ασχολείται με την δημιουργία μιας εφαρμογής προσανατολισμένης στις συγκεκριμένες ανάγκες μιας εταιρίας με συστήματα και συσκευές φιλτραρίσματος του νερού. Η συγκεκριμένη εφαρμογή θα διευκολύνει τη διατήρηση πληροφοριών σχετικά με στοιχεία πελατών, στοιχεία προϊόντων της εταιρίας, ιστορικό εργασιών συντήρησης και τη δημιουργία λίστας εργασιών συντήρησης η οποία θα βοηθήσει στον προγραμματισμό των εργασιών του τεχνικού τμήματος της εταιρίας.

Παρακάτω αναπτύσσονται αναλυτικά όλα τα στάδια από την ανάλυση, την ανάπτυξη μέχρι την ολοκλήρωση της εφαρμογής καθώς και τα εργαλεία που χρησιμοποίησα, με αποτέλεσμα η εν λόγω πτυχιακή να αποτελεί ένα εκπαιδευτικό βοήθημα για τον αναγνώστη.

## 1.1 Περίληψη

Η εφαρμογή που αναπτύχθηκε και αναλύθηκε είναι για την εξυπηρέτηση αναγκών εταιρίας με συσκευές φίλτρων νερού. Η εφαρμογή αυτή θα είναι καθημερινό εργαλείο του τεχνικού τμήματος, η οποία ουσιαστικά θα δημιουργεί και το πλάνο εργασιών των τεχνικών.

Παρέχει πολλές δυνατότητες στον χρήστη όπως για παράδειγμα να καταχωρεί πελάτες στο σύστημα με διάφορα πεδία για τον κάθε ένα από αυτούς όπως ( κωδικό, όνομα, επώνυμο, τηλέφωνο σταθερό κατοικίας κ.α.). Επίσης θα κάνει αναζήτηση των στοιχείων των πελατών που είναι καταχωρημένοι γράφοντας οποιοδήποτε στοιχείο γνωρίζει είτε είναι κάποιο όνομα είτε είναι κάποιος αριθμός η κάποια αρχικά της λέξης. Ακόμα να κάνει επεξεργασία σε όλα τα πεδία των στοιχείων των πελατών που είναι καταχωρημένοι και να διαγράφει τους καταχωρημένους πελάτες. Είναι επίσης πολύ σημαντικό το γεγονός ότι ο χρήστης θα μπορεί να προβάλλει ηλεκτρονικό χάρτη με την τοποθεσία του κάθε πελάτη με βάση τις γεωγραφικές συντεταγμένες. Με αυτόν τον τρόπο θα είναι πολύ εύκολο να βρεθεί ο πελάτης ακόμα και σε απομακρυσμένες περιοχές που δεν έχουν ούτε ονομασία οδού.

Όσο αφορά τα προϊόντα-συσκευές της εταιρίας, αυτά θα καταχωρούνται σε διάφορα πεδία όπως (κωδικό, μοντέλο κ.α.), να επεξεργάζονται τα στοιχεία τους και να διαγράφονται .

Η εφαρμογή θα δίνει τη δυνατότητα στο χρήστη να καταχωρεί πλήθος συσκευών που έχουν αγοραστεί από κάθε πελάτη με συγκεκριμένα χαρακτηριστικά για κάθε μια από αυτές, καθώς επίσης να επεξεργάζεται όλα τα πεδία κάθε αγορασμένης συσκευής και να διαγράφει την αγορασμένη συσκευή από τη λίστα.

Εξίσου σημαντικό είναι το γεγονός ότι με αυτό το πρόγραμμα θα δίνεται η δυνατότητα να καταγράφονται οι εργασίες συντήρησης (service) για κάθε αγορασμένη συσκευή και να γίνεται επεξεργασία σε όλα τα πεδία εργασίας συντήρησης που έχουν καταχωρηθεί. Με αυτές τις καταχωρήσεις η εταιρία θα έχει διαθέσιμο όλο το ιστορικό των εργασιών που έχουν γίνει σε κάθε συσκευή κάθε πελάτη.

Τέλος, αλλά με την μεγαλύτερη σπουδαιότητα, θεωρείται το γεγονός ότι αυτή η εφαρμογή θα έχει σαν αποτέλεσμα την προβολή στην οθόνη και εκτύπωση λίστας με τις συσκευές των πελατών που πρέπει σύντομα να συντηρηθούν. Η λίστα αυτή που θα δημιουργείται βάση νομού θα παρέχει τις πληροφορίες επικοινωνίας με τον πελάτη, για τα φίλτρα και τα αναλώσιμα υλικά που θα χρειαστούν και την τοποθεσία (διεύθυνση) της συσκευής καθαρισμού του νερού.

Η ανάπτυξη έγινε σε Sun Java μια απλή αντικειμενοστραφής γλώσσα, που ενδείκνυται για προγραμματισμό ασφαλών και υψηλής απόδοσης εφαρμογών οι οποίες μπορούν να τρέξουν αυτούσιες σε διαφορετικά λειτουργικά συστήματα και αρχιτεκτονικές. Η βάση δεδομένων έγινε σε MySQL ένα δημοφιλές Σύστημα Διαχείρισης βάσεων δεδομένων ανοικτού κώδικα.

## 1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Αν και οι εφαρμογές εμπορικής διαχείρισης που κυκλοφορούν στην αγορά είναι πάρα πολλές και οι δυνατότητες τους ποικίλουν, σε κάποιες περιπτώσεις δεν είναι αρκετές. Οι ανάγκες μιας επιχείρησης με φίλτρα καθαρισμού νερού για μια αποτελεσματική διαχείριση των εργασιών συντήρησης στις συσκευές που πουλάει, ήταν το αρχικό κίνητρο για την εκπόνηση της συγκεκριμένης πτυχιακής εργασίας. Οι ήδη υπάρχουσες εφαρμογές δεν ήταν σε θέση να καλύψουν απόλυτα τις ανάγκες αυτές.

Ακόμα ένας λόγος, ήταν η προσωπική μου διάθεση να ασχοληθώ περισσότερο με τον αντικειμενοστραφή προγραμματισμό, πιο συγκεκριμένα με την java και τις βάσεις δεδομένων. Ήθελα να δω στην πράξη, σε μια χρήσιμη εφαρμογή, την λειτουργία και τις δυνατότητες της java σε συνδυασμό με την MySQL.

### **1.3 Σκοπός και Στόχοι Εργασίας**

Η εργασία αυτή έχει σαν στόχο την δημιουργία μιας εφαρμογής που θα βοηθήσει την λειτουργία επιχείρησης με φίλτρα καθαρισμού νερού στην οργάνωση των εργασιών συντήρησης με στοιχεία πελατών και προϊόντων. Η εφαρμογή αυτή θα αντικαταστήσει τις παλιές μεθόδους που χρησιμοποιούσε μέχρι τώρα η επιχείρηση, με απώτερο σκοπό να βελτιωθεί η αποδοτικότητα του προσωπικού και ποιότητα υπηρεσιών που προσφέρουν στους πελάτες της.

### **1.4 Δομή Εργασίας**

Η πτυχιακή εργασία είναι χωρισμένη σε έξι κεφάλαια και τρία παραρτήματα.

Το πρώτο κεφάλαιο αναφέρεται σε κάποια εισαγωγικά για την πτυχιακή. Τα επόμενα τρία κεφάλαια ασχολούνται με την μεθοδολογία που χρησιμοποίησα για την δημιουργία της εφαρμογής. Στο δεύτερο για τον αντικειμενοστραφή προγραμματισμό και την JAVA, το τρίτο για τις βάσεις δεδομένων και το τέταρτο για την MySQL. Στο επόμενο κεφάλαιο, το πέμπτο είναι η ανάλυση για την εφαρμογή και πώς φτιάχτηκε. Στο τελευταίο, έκτο κεφάλαιο, είναι οι οδηγίες χρήσης της εφαρμογής.

Το πρώτο παράρτημα περιέχει οδηγίες για εγκατάσταση εργαλείων. Το δεύτερο έχει το περιεχόμενο όλων των κλάσεων της εφαρμογής. Τέλος το τρίτο παράρτημα έχει τις διαφάνειες παρουσίασης της πτυχιακής.

## 2. Αντικειμενοστραφής προγραμματισμός - Μεθοδολογία Υλοποίησης

### 2.1 Εισαγωγή στον αντικειμενοστραφή προγραμματισμό

Ο αντικειμενοστραφής προγραμματισμός (OOP – Object Oriented Programming), είναι μια πραγματική φιλοσοφία όπως και ο προστακτικός ή ο λογικός προγραμματισμός. Σύμφωνα με τον OOP ένα πρόγραμμα δεν αποτελείται από τα δεδομένα και τον κώδικα που τα επεξεργάζεται αλλά από αντικείμενα (objects) τα οποία εμπεριέχουν τα δεδομένα και τα οποία (αντικείμενα) ανταλλάσσουν μεταξύ τους πληροφορίες και μηνύματα προκειμένου να επιτευχθεί ο στόχος του προγράμματος. Για παράδειγμα έστω ότι έχουμε ένα σύστημα που περιλαμβάνει μια βάση δεδομένων καθώς και user interfaces με τα οποία οι χρήστες επικοινωνούν και αλληλεπιδρούν με την Β.Δ. (Βάση Δεδομένων).



Τα UIs (User Interfaces) στέλνουν τις εντολές των χρηστών στην ΒΔ και παρουσιάζουν στην οθόνη τις απαντήσεις που λαμβάνουν. Δηλαδή όλο το πρόγραμμα αποτελείται από ένα αντικείμενο Β.Δ. και 2 αντικείμενα UI τα οποία είναι ολόιδια ( αλλά εξυπηρετούν άλλους χρήστες). Συνεπώς ο κώδικας που θα γράφαμε θα περιείχε τον ορισμό ενός αντικειμένου ΒΔ, τον ορισμό ενός αντικειμένου UI και την κατασκευή μιας ΒΔ και 2 UI αντικειμένων.

#### Παρατήρηση

Ο κώδικας που ορίζει ένα αντικείμενο λέγεται **κλάση** (class) του αντικειμένου αυτού. Η κλάση χρησιμοποιείται σαν μήτρα για την κατασκευή πανομοιότυπων αντιγράφων – αντικειμένων. Τα αντικείμενα – αντίγραφα, λέγονται στιγμιότυπα (instances) της κλάσης αυτής και η κατασκευή και αρχικοποίηση ενός από αυτά λέγεται instantiation.

#### Σημείωση

Το αντικείμενο αυτό είναι μια οντότητα στη μνήμη η οποία περιέχει δεδομένα καθώς και μεθόδους μέσω των οποίων μπορούμε να αλλάξουμε τα δεδομένα ή να επικοινωνήσουμε με το αντικείμενο. Αντίθετα η κλάση είναι απλώς ένα πρότυπο για την δημιουργία αντιγράφων του ίδιου αντικειμένου. Είναι δηλαδή ο Τύπος Δεδομένων για τις μεταβλητές.

### 2.1.1 Χαρακτηριστικά αντικειμενοστραφούς προγραμματισμού

#### **Encapsulation ( )**

Η διαδικασίες κρύβονται από τον χρήστη και τα ίδια τα δεδομένα προσδιορίζουν τους τρόπους διαχείρισής τους.

## Polymorphism (Πολυμορφισμός)

Αντικείμενα που ανήκουν σε παρόμοιες κλάσεις μπορούν να έχουν κοινό τρόπο προσπέλασης, με αποτέλεσμα ο χρήστης να μπορεί να τα χειριστεί με τον ίδιο τρόπο χωρίς να χρειάζεται να μάθει νέες διαδικασίες.

Οι τύποι πολυμορφισμού είναι:

2. Η υπερκάλυψη (overriding)
3. Η υπερφόρτωση (overloading)
4. Η δυναμική συσχέτιση μεθόδων (dynamic method binding)

## Inheritance (Κληρονομικότητα)

Η κληρονομικότητα είναι ένα ακόμη χαρακτηριστικό του αντικειμενοστραφούς προγραμματισμού. Μπορούμε να δημιουργήσουμε ένα νέο αντικείμενο παίρνοντας ως βάση ένα άλλο ήδη υπάρχον. Το νέο αντικείμενο θα έχει τα χαρακτηριστικά του παλιού ενώ θα μπορεί να τα τροποποιήσει, να τα επεκτείνει και να προσθέσει καινούργια για να καλύψει συγκεκριμένες ανάγκες. Πρακτικά, μια κλάση κληρονομεί τα χαρακτηριστικά μιας υπάρχουσας κλάσης και προσθέτει καινούργια ή τροποποιεί τα ήδη υπάρχοντα.

## 2.2 Java

Η **Java** είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού που εμπεριέχει όλα τα χαρακτηριστικά της εξέλιξης της επιστήμης των υπολογιστών και χρησιμοποιείται σε ένα ευρύ φάσμα εργασιών. Επίσης επιτρέπει σε οποιοδήποτε υπολογιστή, όπου κι αν βρίσκεται αυτός, να έχει πρόσβαση και να χρησιμοποιεί μια εφαρμογή εγκατεστημένη σε κάποιο δίκτυο. Οι σχεδιαστές της Java ξεκίνησαν από την C++, αφαίρεσαν πολλά χαρακτηριστικά, τροποποίησαν μερικά αλλά και πρόσθεσαν ορισμένα καινούρια. Το αποτέλεσμα διατηρεί την ισχύ και ευελιξία της C++, παρέχοντας όμως μια γλώσσα απλούστερη και ασφαλέστερη. Ακόμη προκάλεσε ίσως το μεγαλύτερο ενδιαφέρον σε σύγκριση με οποιαδήποτε άλλη εξέλιξη στον κόσμο του Internet.

Είναι η πρώτη που κατάφερε να συμπεριλάβει ήχο και κίνηση σε μια ιστοσελίδα. Η Java επιπλέον επιτρέπει στους χρήστες να αλληλεπιδρούν (interact) με την ιστοσελίδα. Εκτός από το να διαβάξει απλά και ίσως να συμπληρώνει μία φόρμα, ο χρήστης μπορεί τώρα να παίζει παιχνίδια, να συνομιλήσει, να λαμβάνει συνεχώς τις πιο πρόσφατες πληροφορίες και πολλά άλλα.

Ακολουθούν μερικές από τις πολλές δυνατότητες της Java:

- Ήχος ο οποίος εκτελείται όποτε ο χρήστης φορτώνει μία σελίδα
- Μουσική που παίζει στο background μιας σελίδας
- Δημιουργία κινουμένων σχεδίων
- Βίντεο
- Παιχνίδια με πολυμέσα

Η Java δεν είναι απλά μια γλώσσα προγραμματισμού του δικτύου με ειδικά χαρακτηριστικά. Παρόλο που η HotJava ήταν η πρώτη γλώσσα που συμπεριέλαβε ήχο και κίνηση, ο Microsoft Internet Explorer και ο Netscape Navigator υποστηρίζουν αυτά τα χαρακτηριστικά με πολλούς και διαφορετικούς τρόπους.

Επίσης είναι **ασφαλής**. Έχει σχεδιαστεί από την αρχή με τέτοιο τρόπο, ώστε να παρέχει ασφάλεια εκτέλεσης του κώδικα σε δίκτυο. Αυτό είχε ως αποτέλεσμα τον περιορισμό αρκετών από τα χαρακτηριστικά που έχουν η C και η C++. Έτσι δεν υπάρχουν δείκτες, ούτε μπορεί να γίνει αυθαίρετη προσπέλαση διευθύνσεων της μνήμης. Οπότε παρέχεται προστασία από την δράση των ιών με κάποια ειδικά προγράμματα που δημιουργεί ο προγραμματιστής και λέγονται **applets**.

Τα applet είναι προγράμματα, τα οποία “κατεβάζει” από τον παγκόσμιο ιστό (WWW) και εκτελεί το πρόγραμμα πλοήγησης (browser) ενός χρήστη ο οποίος είναι συνδεδεμένος στο Internet. Αυτά δεν μπορούν να γράψουν στο σκληρό δίσκο αν δε ζητήσουν πρώτα την άδεια, ούτε και να

μπλοκάρουν τον υπολογιστή μας. Χρησιμοποιεί επίσης έναν ισχυρό μηχανισμό για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων (exception handling).

Υποστηρίζει **πολυνημάτωση** (multithreading). Ένα πρόγραμμα **Java** μπορεί να περιλαμβάνει πολλές ξεχωριστές διαδικασίες, οι οποίες να εκτελούνται συνεχώς ανεξάρτητα η μία από την άλλη. Μπορεί π.χ. από το πρόγραμμα να μεταδίδεται μία εικόνα, και συγχρόνως ο χρήστης να εισαγάγει στοιχεία από το πληκτρολόγιο.

Κάνει **συλλογή αχρήστων (garbage collection)**. Σύλληξη 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 η στοίβα (heap) της μνήμης (στη Java η συντριπτική πλειοψηφία των αντικειμένων αποθηκεύονται στη στοίβα σε αντίθεση με τη C++ που αποθηκεύονται κυρίως στο σωρό - 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 εστιάζουν σε άδειο κουτάκι μνήμης. Αυτό είναι ιδιαίτερα σημαντικό αν σκεφτούμε ότι ένα μεγάλο ποσοστό κατάρρευσης των προγραμμάτων οφείλονται σε λάθος χειρισμό της μνήμης.

Μπορεί να δημιουργήσει **γρήγορο κώδικα** με κάποιους ειδικούς μεταγλωττιστές οι οποίοι μετατρέπουν τον κώδικα byte σε κώδικα γλώσσας μηχανής, ο οποίος ανταγωνίζεται σε ταχύτητα τον κώδικα της C++ ή άλλων γλωσσών. Οι μεταγλωττιστές αυτοί ονομάζονται “just in time compilers” (JIT).

Παρέχει **βιβλιοθήκες κώδικα** για διάφορες χρήσεις, όπως τη δημιουργία γραφικών, μαθηματικές πράξεις, χειρισμό σχεσιακών βάσεων δεδομένων, κλήση απομακρυσμένων αντικειμένων κτλ.

Με την Java μπορούμε να **δημιουργήσουμε δυναμικές ιστοσελίδες (dynamic web pages)**.

Επίσης η Java είναι μια γενική γλώσσα προγραμματισμού. Θα μπορούσε να χρησιμοποιηθεί ως γλώσσα γενικού σκοπού για τη δημιουργία συνηθισμένου τύπου εφαρμογών, όπως για παράδειγμα για την κατασκευή ενός οδηγού (driver) ή για έναν εκτυπωτή. Στην ορολογία της sun ένα κανονικό πρόγραμμα σε Java χαρακτηρίζεται με τον όρο “application” (εφαρμογή).

## 2.3 Τι κάνει τη Java να ξεχωρίζει;

Η Java είναι μια γλώσσα προγραμματισμού για ποικίλες εφαρμογές. Δεν προσφέρει απλά τη δυνατότητα να προσθέσει ο χρήστης νέο περιεχόμενο στις σελίδες του (όπως συμβαίνει στο Netscape και στον Internet Explorer) αλλά επιτρέπει να προσθέσουμε και τον κώδικα που είναι απαραίτητος. Δεν χρειάζεται πλέον να περιμένουμε για να κυκλοφορήσει ο browser που θα υποστηρίξει τον συγκεκριμένο τύπο εικόνας ή το ειδικό πρωτόκολλο παιχνιδιού (special game protocol). Με τη Java εμείς στέλνουμε στους browsers το περιεχόμενο που χρειάζεται και το πρόγραμμα για να δούμε αυτό το περιεχόμενο την ίδια στιγμή.

Μέχρι τώρα έπρεπε να περιμένουμε τους αναγνώστες μας να ενημερώσουν τους browsers τους προτού χρησιμοποιήσουμε ένα νέο τύπο περιεχομένου (content type). Η ανταγωνιστικότητα της Java βρίσκεται στο ότι μπορεί να εφαρμοστεί σε οποιονδήποτε browser. Για παράδειγμα, θέλουμε να χρησιμοποιήσουμε τα αρχεία EPS στο site μας. Προηγουμένως, έπρεπε να περιμένουμε μέχρι ένας τουλάχιστον web browser να εφάρμοζε την υποστήριξη EPS. Τώρα πια δεν περιμένουμε. Αντίθετα, μπορούμε να γράψουμε τον δικό μας κώδικα για να δούμε τα αρχεία EPS και να το στείλουμε σε οποιονδήποτε πελάτη ζητά τη σελίδα μας τον ίδιο χρόνο που ζητά το αρχείο EPS.

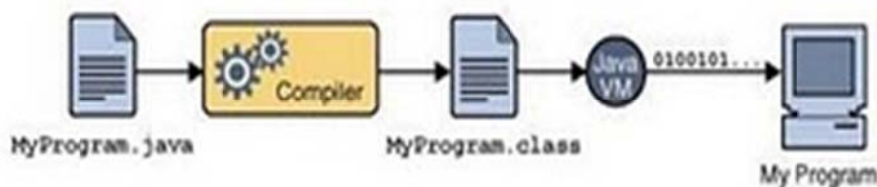
Υποθέτουμε ότι θέλουμε άτομα που να μπορούν να ψάχνουν τον ηλεκτρονικό μας κατάλογο (electronic card catalog). Η βάση δεδομένων του καταλόγου όμως υπάρχει σ’ ένα μεγάλο σύστημα που δεν αναγνωρίζει την HTTP. Πριν τη Java θα μπορούσαμε να ελπίζουμε ότι κάποιος browser θα εφάρμοζε το πρωτόκολλο της κάρτας ή θα μπορούσαμε να προσπαθήσουμε να προγραμματίσουμε κάποιο ενδιάμεσο cgi-bin σε ένα UNIX BOX που θα αναγνώριζε HTTP, πράγμα που δεν είναι

καθόλου εύκολο. Με τη Java, όταν ένας πελάτης θέλει να μιλήσει στον κατάλόγό μας μπορούμε να του στείλουμε τον κώδικα που χρειάζεται. Η Java δεν είναι γλώσσα μόνο για τα web sites. Η Java είναι μια γλώσσα προγραμματισμού που μας επιτρέπει να κάνουμε ότι και οι παραδοσιακές γλώσσες, όπως η Fortran και η C++.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής.

Ο compiler της java, μετατρέπει τα αρχεία μας σε bytecodes τα οποία είναι η γλώσσα μηχανής της java (προσοχή εδώ! Σε γλώσσα μηχανής της java και όχι του υπολογιστή), και δημιουργεί ένα νέο αρχείο με την κατάληξη .class. Εδώ βρίσκεται το όλο μυστικό στην συγκεκριμένη γλώσσα. Η java δημιουργεί ένα ενδιάμεσο επίπεδο μεταξύ του προγράμματος και του λειτουργικού συστήματος του υπολογιστή που το ονομάζει JVM (Java Virtual Machine). Αυτό, κατά κάποιο τρόπο ξεγελάει το πρόγραμμα μας για το ποιος είναι το CPU, η μνήμη και το λειτουργικό σύστημα. Οπότε εμείς πάντα γράφουμε κώδικα για το JVM (χωρίς να το καταλάβουμε) και όχι για την συγκεκριμένη πλατφόρμα στην οποία εργαζόμαστε.

Η παρακάτω φωτογραφία προήλθε από το site της Sun (<http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>)



Εικόνα 2.α : Επισκόπηση της διαδικασίας ανάπτυξης λογισμικού.

## 2.4 Περιβάλλον ανάπτυξης και δημιουργία εφαρμογής

Για να γίνει ευκολότερα κατανοητή η διαδικασία δημιουργίας μια εφαρμογής, ακολουθούμε τα παρακάτω βήματα:

- Κατεβάζουμε το JDK από το site της SUN και το εγκαθιστούμε στον υπολογιστή μας. Αυτό περιέχει τα εργαλεία εντολών που μας δίνουν την δυνατότητα να κάνουμε compile (έλεγχος σύνταξης εντολών) και να τρέξουμε το πρόγραμμά μας για να δοκιμάσουμε το αποτέλεσμα του κώδικά μας.
- Μετά γράφουμε το πρόγραμμά μας και το αποθηκεύουμε με κατάληξη .java. Μέχρι αυτή την στιγμή δεν γνωρίζουμε εάν υπάρχουν συντακτικά λάθη.
- Ενεργοποιούμε την διαδικασία του compilation για να εγκριθεί από το περιβάλλον της java το πρόγραμμά μας. Αν υπάρχουν λάθη τότε πρέπει να επιστρέψουμε στο αρχείο και να κάνουμε τις απαραίτητες αλλαγές πριν συνεχίσουμε. Εάν όχι, τότε δημιουργείται ένα καινούργιο αρχείο με το ίδιο όνομα όπως το πρόγραμμά μας αλλά με κατάληξη .class.

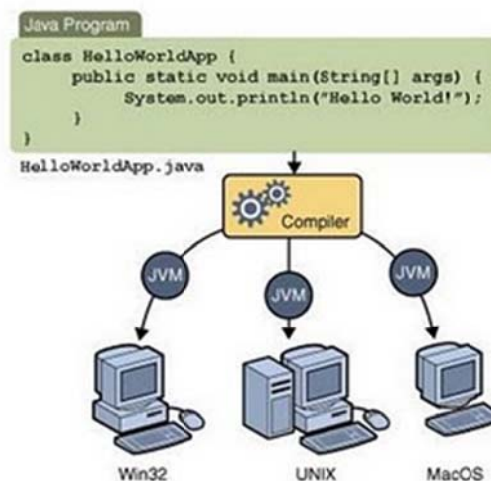


- Στον ίδιο φάκελο που έχουμε το αρχείο με την κατάληξη .java τώρα υπάρχει και ένα δεύτερο με την κατάληξη .class.
- Το δεύτερο αρχείο με την κατάληξη .class περιέχει τον κώδικα μας αλλά κωδικοποιημένο σε bytecodes. Αυτός είναι ο κώδικας που καταλαβαίνει να διαβάσει το JVM.
- Πρακτικά το JVM είναι το JRE πρόγραμμα που περιέχεται μέσα στο JDK πακέτο της Java.
- Τα bytecodes διαβάζονται από το JVM και μεταφράζονται στο τρέχων λειτουργικό σύστημα. Δηλαδή αναλαμβάνει την μετάφραση από εντολές java σε κώδικα μηχανής και ολοκληρώνει την εκτέλεση του προγράμματος.

Όπως ήδη αναφέρθηκε πιο πάνω, ο πρακτικός αντιπρόσωπος του JVM είναι το πρόγραμμα JRE (Java Runtime Environment) το οποίο χρειάζεται κάθε υπολογιστής για να τρέξει java προγράμματα. Το JRE περιέχεται μέσα στον compiler ή καλύτερα στο JDK πακέτο το οποίο έχουμε κατεβάσει έτσι ώστε να έχουμε την δυνατότητα και να γράψουμε αλλά και να τρέξουμε τα προγράμματα μας. Μπορούμε όμως να εγκαταστήσουμε το JRE και αυτόνομα και μάλιστα αυτή η διαδικασία είναι αναγκαία όταν είμαστε έτοιμοι να εγκαταστήσουμε την εφαρμογή μας και σε άλλους υπολογιστές. Κάθε υπολογιστής που θα τρέξει την εφαρμογή μας, θα πρέπει να έχει ήδη εγκαταστημένο το JRE πρόγραμμα για να μπορεί να δημιουργήσει το JVM. Πολλοί προγραμματιστές απλά ενσωματώνουν με την εφαρμογή τους και το JRE έτσι ώστε ο τελικός χρήστης να μην αντιμετωπίσει κανένα απολύτως πρόβλημα.

Επειδή θέλουμε το αρχείο .class να τρέχει σε οποιαδήποτε πλατφόρμα, τότε θα πρέπει να κατεβάσουμε και το ανάλογο JRE. Ας υποθέσουμε ότι έχουμε γράψει ένα πρόγραμμα σε java στον υπολογιστή μας που έχει εγκατεστημένα τα Windows σαν λειτουργικό. Ας υποθέσουμε επίσης ότι έχουμε γράψει το πρόγραμμα με τις σωστές εντολές και μετά την διαδικασία του compilation δημιουργήθηκε το αρχείο με κατάληξη .class. Για να μπορεί ένας άλλος χρήστης να τρέξει το πρόγραμμα μας στον δικό του υπολογιστή, θα πρέπει να του δώσουμε μόνο το αρχείο με την κατάληξη .class. Για να έχει την δυνατότητα να το εκτελέσει απλά θα πρέπει να κατεβάσει το σωστό JRE για την πλατφόρμα του υπολογιστή του.

Με άλλα λόγια, αν έχει LINUX τότε χρειάζεται το JRE για το LINUX ενώ εάν έχει Windows τότε χρειάζεται το Windows version του JRE. Ο λόγος που χρειάζεται μόνο το JRE και όχι όλο το πακέτο της java (δηλαδή το JDK) είναι γιατί απλά θα τρέξει το πρόγραμμα, και δεν χρειάζεται να το κάνει compile. Το .class αρχείο είναι ήδη το compile αρχείο του προγράμματος. Η παρακάτω εικόνα προήλθε από το site της SUN από την ηλεκτρονική διεύθυνση <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>.

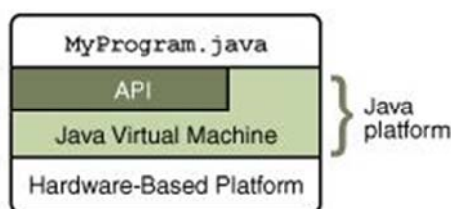


Εικόνα 2.β : Μέσω της Java VM, η ίδια αίτηση μπορεί να λειτουργεί σε πολλαπλές πλατφόρμες.

Ας ορίσουμε και την έννοια “java platform”. Πλατφόρμα στον κόσμο της πληροφορικής ονομάζεται το περιβάλλον (μπορεί να είναι software ή hardware) πάνω στο οποίο “τρέχει” ένα πρόγραμμα. Ολοκληρωμένες πλατφόρμες στις οποίες στηρίζονται πολλές εφαρμογές είναι τα λειτουργικά συστήματα όπως Windows, Linux, Solaris και MAC OS. Για την java όμως η έννοια της πλατφόρμας αποκτά συγκεκριμένη έννοια. Ορίζουμε σαν πλατφόρμα την δημιουργία μιας software μόνο υποδομής η οποία στηρίζεται και εκτελείται επάνω σε hardware πλατφόρμες. Η java μάλιστα έχει χωρίσει την πλατφόρμα της σε δύο κατηγορίες:

1. Στην Java Virtual Machine για την οποία ήδη έχουμε μιλήσει και
2. Στο Java Application Programming Interface (API).

Το API είναι μια συλλογή από έτοιμα προγραμματισμένα στοιχεία που μας δίνουν την δυνατότητα να τα χρησιμοποιήσουμε και να αξιοποιήσουμε τις ικανότητες τους χωρίς να γνωρίζουμε την υποδομή τους. Τα στοιχεία αυτά είναι ομαδοποιημένα σε βιβλιοθήκες κατηγοριών για να γίνεται ευκολότερη η ταξινόμηση τους που είναι γνωστές σαν packages. Η παρακάτω εικόνα προέρχεται από το site της Sun (<http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>)



**Εικόνα 2.γ : Το API και Java Virtual Machine μονώνουν το πρόγραμμα από το υποκείμενο υλικό.**

Είναι αντικειμενοστραφής δηλαδή χρησιμοποιεί τις κλάσεις (classes) για να οργανώσει τον κώδικα σε λογικές ενότητες. Όταν εκτελείται το πρόγραμμα δημιουργεί από τις κλάσεις αντικείμενα. Τα αντικείμενα αυτά έχουν δύο συνιστώσες: τα πεδία και τις μεθόδους. Τα πεδία περιγράφουν τι είναι το αντικείμενο ενώ οι μέθοδοι περιγράφουν τι κάνει το αντικείμενο. Οι κλάσεις μπορούν να κληρονομήσουν ιδιότητες από άλλες κλάσεις. Όμως δεν επιτρέπεται η πολλαπλή κληρονομικότητα (multiple inheritance), όπου μία κλάση έχει τη δυνατότητα να κληρονομήσει πεδία και μεθόδους από περισσότερες από μία άλλες κλάσεις.

## 2.5 Δομή ενός προγράμματος σε Java

```
<ΕΙΣΑΓΩΓΗ ΕΤΟΙΜΩΝ ΚΛΑΣΕΩΝ ΑΠΟ ΤΗ ΒΙΒΛΙΟΘΗΚΗ>  
<ΟΡΙΣΜΟΣ ΝΕΑΣ ΚΛΑΣΗΣ>  
<ΟΡΙΣΜΟΣ ΝΕΑΣ ΚΛΑΣΗΣ>  
κλπ.
```

Μία από όλες τις κλάσεις θα παίζει το ρόλο της αφετηρίας του προγράμματος, δηλαδή θα είναι σαν την main() της C. Αναλυτικότερα, θα αναλάβει να κατασκευάσει τα αντικείμενα που χρειάζεται το πρόγραμμα, με τη χρήση των κλάσεων που ορίσαμε ή φέραμε από την βιβλιοθήκη. Βέβαια την αρχική κλάση θα την υποδείξουμε εμείς όταν θα ξεκινήσουμε τον interpreter της γλώσσας.

## 2.5.1 Τα αντικείμενα της Java

Όπως αναφέρθηκε και παραπάνω τα αντικείμενα περιέχουν δεδομένα και παρέχουν μεθόδους για την επεξεργασία των δεδομένων αυτών καθώς και για την επικοινωνία με άλλα αντικείμενα.

Τα δεδομένα αυτά μπορεί να είναι άλλα αντικείμενα που περιέχονται μέσα σε ένα άλλο αντικείμενο ή μπορεί να είναι κοινές μεταβλητές όπως τις ξέρουμε από την C και την PASCAL. Τις μεταβλητές και τα αντικείμενα αυτά τα ονομάζουμε πεδία ή instance variables του αντικειμένου. Σε αυτό το σημείο να παρατηρήσουμε ότι μία κλάση μοιάζει με ένα structure της C το οποίο μπορεί να περιέχει κοινές μεταβλητές ή μεταβλητές που προέκυψαν από άλλα structures.

Επίσης μία κλάση (ή ένα αντικείμενο) περιέχει και μεθόδους για την επικοινωνία του με τον έξω κόσμο, δηλαδή τα άλλα αντικείμενα. Οι μέθοδοι αυτοί υλοποιούνται σαν συναρτήσεις παρόμοιες με αυτές της C. Όταν λοιπόν κάποιος θέλει να ζητήσει κάτι από ένα αντικείμενο, (ή εναλλακτικά να στείλει ένα μήνυμα – αίτημα), δεν έχει πάρα να καλέσει – εκτελέσει την αντίστοιχη μέθοδο του αντικειμένου. Αυτό γίνεται ως εξής :

```
<Αντικείμενο>.<μέθοδος>(<Λίστα παραμέτρων >)
```

Επίσης να σημειωθεί ότι μία μέθοδος μπορεί να καλεί άλλες μεθόδους του ίδιου αντικειμένου ή να επεξεργάζεται τα πεδία του. Φυσικά μπορεί να καλεί και μεθόδους ξένων αντικειμένων εφόσον έχει κάποιον δείκτη σε αυτά.

### ΣΗΜΕΙΩΣΗ

Εδώ είδαμε ένα σπουδαίο χαρακτηριστικό του OOP. Αυτό είναι το να μπορεί να κρατά τα δεδομένα του κρυφά από τα άλλα αντικείμενα αλλά και να προσφέρει μεθόδους με τις οποίες μπορούν άλλα αντικείμενα να επικοινωνούν και να αλληλεπιδρούν μαζί του. Επιπλέον ο κώδικας που υλοποιεί αυτές τις μεθόδους είναι άγνωστος σε άλλες κλάσεις ή αντικείμενα. Έτσι έχουμε την δημιουργία ενός interface επικοινωνίας ανεξάρτητου από την υλοποίηση και την εσωτερική δομή του αντικειμένου. Αυτό λέγεται implementation hiding που είναι μία μορφή data abstraction.

Ύστερα από τα παραπάνω μπορούμε να δούμε πώς ορίζουμε ένα αντικείμενο ή με άλλα λόγια πώς ορίζουμε την κλάση του.

## 2.5.2 Ορισμός κλάσεων στην Java

Μία κλάση της java μοιάζει αρκετά με ένα structure της C. Δηλαδή έχει την ακόλουθη δομή :

```
Class<class_name>{  
  
    <data_type or class_name>    <variable or objects>;  
  
    <returned type> <method_name> (<parameter list>){  
        <method body>  
    }  
    <data_type or class_name>    <variables or objects>;  
  
    <returned type> <method_name> (<parameter list>){  
        <method body>  
    }  
  
    κ.ο.κ.  
}
```

### Γενικά

Μπορούμε να δηλώνουμε τα πεδία και τις μεθόδους με όποια σειρά θέλουμε. Επίσης μπορούμε να τα χρησιμοποιούμε πριν ακόμη τα δηλώσουμε. Ωστόσο συνιστάται τα πεδία να τοποθετούνται στην αρχή της κλάσης ώστε ο κώδικας να είναι πιο ευανάγνωστος. Ακόμη η τοποθέτηση κατατοπιστικών σχολίων συμβάλλει στην βελτίωση της αναγνωσιμότητας του προγράμματος και διευκολύνει την διαδικασία αποσφαλμάτωσης (debugging).

## 2.6 Κατασκευαστές (constructors)

Κάθε κλάση διαθέτει τουλάχιστον μία μέθοδο η οποία εκτελείται κατά τη δημιουργία ενός στιγμιότυπου της, (δηλ. κατά τη δημιουργία ενός object της κλάσης αυτής). Αυτές οι μέθοδοι λέγονται **κατασκευαστές** (constructors) της κλάσης. Σκοπός των constructors είναι η δέσμευση μνήμης για την κατασκευή του αντικειμένου καθώς και η διενέργεια κατάλληλων αρχικοποιήσεων. Μπορούν να οριστούν πολλοί constructors για μία κλάση αλλά για τη δημιουργία ενός αντικειμένου (στιγμιότυπου) μπορεί να καλέσει μόνο έναν από αυτούς. Εάν δεν οριστεί κανένας constructor τότε η γλώσσα καλεί έναν constructor της υπερκλάσης (κάποιον που δεν θέλει παραμέτρους) ή δίνει μήνυμα λάθους αν δεν βρει κάποιον τέτοιο.

## 2.7 Καταστροφή αντικειμένων (Finalization)

Σε ορισμένες γλώσσες όπως η C++ και η Object PASCAL υποστηρίζεται και η χρήση των **καταστροφών** (destructors) μιας κλάσης. Προφανώς οι destructors είναι μέθοδοι που τερματίζουν την ύπαρξη ενός στιγμιότυπου της αντίστοιχης κλάσης. Ωστόσο στην JAVA το χαρακτηριστικό αυτό δεν υποστηρίζεται αφού είναι αιτία λαθών. Έχει όμως έναν άλλο πιο ασφαλές τρόπο για τον τερματισμό (finalization) ενός αντικειμένου. Ο τρόπος αυτός έχει ως εξής. Το runtime system της java εξετάζει κατά διαστήματα αν για κάθε αντικείμενο υπάρχει τουλάχιστο ένας δείκτης σε αυτό. Αν δεν υπάρχει τότε το αντικείμενο είναι άχρηστο (garbage) και η μνήμη που κατέχει απελευθερώνεται.

Όμως προτού γίνει αυτό καλείται η μέθοδος void finalize() η οποία πρέπει να περιέχει κώδικα για clean-up. Για παράδειγμα αν το αντικείμενο έχει ακόμα ανοιχτές συνδέσεις στο δίκτυο τότε θα τις κλείσει. Μετά το τέλος της finalize() το αντικείμενο καταστρέφεται. Η διαδικασία του εντοπισμού των αντικειμένων – σκουπιδιών λέγεται garbage collection και ο μηχανισμός που το πραγματοποιεί είναι ενσωματωμένος στο JAVA runtime system.

## 2.8 Προσδιοριστές πρόσβασης (access specifiers)

Συμβαίνει πολλές φορές να θέλουμε να έχουμε κάποια δεδομένα ή μεθόδους μας ορατά στα αντικείμενα όλων των κλάσεων ή να θέλουμε κάποια πεδία να είναι ορατά μόνο στα αντικείμενα της τρέχουσας κλάσης και όχι και στις υποκλάσεις της. Σε όλες αυτές τις περιπτώσεις χρησιμοποιούμε τους access specifiers. Οι access specifiers είναι δεσμευμένες λέξεις της JAVA οι οποίες καθορίζουν το ποιος θα έχει πρόσβαση, δηλαδή το ποιος θα μπορεί να χρησιμοποιεί και να επεξεργάζεται, τις μεθόδους ή/και τα πεδία που έχουν κάποιον τέτοιο προσδιοριστή. Για αρχή ας δούμε ποιοι είναι :

- **public** - Σημαίνει ότι το συγκεκριμένο πεδίο/μέθοδος μπορεί να χρησιμοποιηθεί από οποιονδήποτε, ακόμα και από ξένο αντικείμενο.
- **protected** - Σημαίνει ότι το πεδίο/μέθοδος που έχει είναι ορατό μόνο στις μεθόδους της κλάσης αυτής καθώς και στις υποκλάσεις.
- **Private** - Σημαίνει ότι το πεδίο/μέθοδος είναι ορατό μόνο στις μεθόδους αυτής της κλάσης αλλά όχι στις υποκλάσεις της.
- **Τίποτα** - Είναι public αλλά μόνο για το τρέχων package.

## 2.9 Τροποποιητές (Modifiers)

Είναι δεσμευμένες λέξεις της JAVA οι οποίες προσδίδουν συγκεκριμένες ιδιότητες ή χαρακτηρίζουν τα πεδία/μεθόδους που τα έχουν. Οι πιο σημαντικοί από τους modifiers είναι οι ακόλουθοι:

- **final** - Για τα πεδία σημαίνει ότι είναι σταθερές και όχι μεταβλητές. Άρα δεν μπορούμε να τους αναθέσουμε νέα τιμή, παρά μόνο στην αρχικοποίηση της σταθεράς όταν αυτή δηλώνεται. Τέλος για τις κλάσεις σημαίνει ότι δεν μπορούμε να κατασκευάσουμε υποκλάσεις από αυτές.

- **Synchronized** - Για μεθόδους σημαίνει ότι το αντικείμενο ιδιοκτήτης της μεθόδου αυτής είναι threadsafe. Δηλαδή αν κάποιος χρησιμοποιεί αυτή την μέθοδο τότε, αποκλείει την ταυτόχρονη χρήση του αντικειμένου και από τρίτον. Ο τρίτος θα περιμένει μέχρι να ελευθερωθεί το αντικείμενο και έπειτα θα το χρησιμοποιήσει και αυτός.

ΠΡΟΣΟΧΗ: αν κάποιος (τρίτος) θέλει να χρησιμοποιήσει την ίδια μέθοδο ενός άλλου αντικειμένου, (της ίδιας κλάσης), δεν θα έχει κανένα πρόβλημα να το κάνει αρκεί να μην χρησιμοποιείται ήδη και αυτό.

- **abstract** - Για μεθόδους σημαίνει ότι δεν θα υλοποιηθούν εδώ αλλά σε κάποια υποκλάση, έτσι δεν παρέχεται το σώμα της μεθόδου {...} αλλά στη θέση του βάζουμε ένα semicolon (;). Επίσης μία κλάση με έστω μία abstract μέθοδο θα πρέπει να δηλωθεί και αυτή ως abstract.
- **static** - Το πεδίο/μέθοδος που το έχει είναι μοναδικό για όλα τα στιγμιότυπα της κλάσης. Όλα τα αντικείμενα της κλάσης αυτής έχουν ένα τέτοιο πεδίο σαν κοινή μεταβλητή. Επίσης οι static μεταβλητές και μέθοδοι μπορούν να χρησιμοποιηθούν και χωρίς να κατασκευαστεί κάποιο αντικείμενο (στιγμιότυπο) της κλάσης. Όμως οι static μέθοδοι δεν μπορούν να δουν ή να αλλάξουν τα πεδία που δεν είναι static. Ωστόσο το αντίθετο είναι επιτρεπτό.
- **native** - Όσες μέθοδοι δηλωθούν native δεν πρέπει να έχουν σώμα. Ωστόσο ο κώδικας τους θα πρέπει να γράφει σε C και να συνδεθεί με το bytecode κατά την εκτέλεση του προγράμματος.

## 2.10 Διασυνδέσεις (Interfaces)

Μια διασύνδεση (interface) ορίζει έναν τρόπο συμπεριφοράς που μπορεί να υλοποιηθεί από οποιαδήποτε κλάση. Δηλώνει ένα σύνολο μεθόδων αλλά δεν προσφέρει την υλοποίησή τους. Αυτές οι μέθοδοι θεωρούνται αυτόματα εικονικές και οποιαδήποτε κλάση υλοποιεί το interface πρέπει να δώσει και υλοποίηση για όλες τις μεθόδους του. Ο λόγος για τον οποίο υπάρχουν τα interfaces είναι ότι κάθε κλάση μπορεί να έχει μόνο μία υπερκλάση αλλά μπορεί να υλοποιήσει άπειρο αριθμό interfaces. Ένα interface μοιάζει με μία abstract κλάση αλλά έχει τις εξής σημαντικές διαφορές :

- Ένα interface δεν μπορεί να παρέχει υλοποίηση για καμία μέθοδο του.
- Μια κλάση μπορεί να υλοποιήσει πολλά interfaces αλλά να κληρονομήσει μία μόνο κλάση.
- Ένα interface μπορεί επίσης να περιέχει σταθερές στον ορισμό του ή να κληρονομεί από άλλα interfaces.

## 2.11 Εξαιρέσεις (Exceptions)

Όταν σε ένα πρόγραμμα σε JAVA συμβεί κάποιο λάθος, για παράδειγμα περαστεί κάποια λάθος παράμετρος, τότε ο κώδικας που θα το ανιχνεύσει μπορεί να εγείρει (throw= πετάω) μία εξαίρεση. Η έγερση εξαίρεσης θα έχει ως αποτέλεσμα τον τερματισμό του thread στο οποίο συνέβη το λάθος και θα τυπωθεί κάποιο μήνυμα. Ωστόσο τα προγράμματα μπορούν να ορίσουν χειριστές εξαίρεσεων (exception handlers) οι οποίοι θα πιάνουν την εξαίρεση και θα φροντίζουν ώστε το πρόγραμμα να ανανήψει το λάθος. Μερικές από τις εξαιρέσεις προκαλούνται από το runtime system, όπως στην περίπτωση διαίρεσης με το μηδέν.

Ωστόσο, οποιαδήποτε κλάση μπορεί να ορίσει και να εγείρει δικές της εξαιρέσεις. Αυτό γίνεται ως εξής. Πρώτα δημιουργεί (με new) ένα αντικείμενο εξαίρεσης το οποίο θα πρέπει να είναι στιγμιότυπο της κλάσης Exception ή κάποιας υποκλάσης της. Έπειτα με την εντολή throw και το αντικείμενο εξαίρεσης προκαλείται exception. Η εκτέλεση του κώδικα θα διακοπεί στην εντολή throw και ο υπόλοιπος κώδικας που ακολουθεί δεν θα εκτελεστεί. Επίσης η μέθοδος μέσα στην οποία συνέβη η εξαίρεση δεν θα επιστρέψει κάποια τιμή. Το αντικείμενο τώρα, της εξαίρεσης θα δοθεί στον κατάλληλο exception handler, απ' όπου και συνεχίζεται η εκτέλεση του προγράμματος.

Για να ορίσουμε ένα χειριστή εξαίρεσεων (exception handler) θα πρέπει να κλείσουμε τον κώδικα που μπορεί να προκαλέσει την εξαίρεση μέσα σε μια εντολή try. Μετά την try θα πρέπει να βάλουμε μία ή περισσότερες εντολές catch. Κάθε catch μπορεί να πιάνει, μία μόνο κλάση εξαίρεσης. Επίσης σε κάθε catch θα υπάρχει ο κατάλληλος κώδικας για τον χειρισμό της εξαίρεσης.

## 2.12 Τα βασικά των Applets στην Java

Στην Java τα Applets εκτελούνται μέσα από κάποιον Java WWW Browser. Η αναφορά σε ένα Applet γίνεται σε μια WEB σελίδα μέσω ενός ειδικού HTML tag. Όταν ο χρήστης σηκώσει σε κάποιον Browser μια WEB σελίδα που περιέχει κάποιο Applet, ο Browser κατεβάζει το Applet από τον Web Server και το εκτελεί στον τοπικό υπολογιστή.

Επειδή τα Java Applets τρέχουν μέσα από κάποιον Java Browser, έχουν το πλεονέκτημα της δομής που παρέχει ο Browser: ένα υπάρχον παράθυρο, έννοιες γραφικών και γεγονότων, και το interface που τα περιβάλλει. Επιπλέον επειδή τα Applets μπορούν να κατεβούν από οπουδήποτε και να εκτελούνται τοπικά στον υπολογιστή του χρήστη, υπάρχουν περιορισμοί που εμποδίζουν τα Applets να προκαλέσουν ζημιά στο τοπικό σύστημα όπως:

- Τα Applets δεν μπορούν να γράφουν ή να διαβάσουν στο τοπικό σύστημα αρχείων, εκτός από καταλόγους που πρέπει να έχει προκαθορίσει ο τοπικός χρήστης.
- Τα Applets μπορούν να επικοινωνήσουν μόνο με τον Server στον οποίο το Applet είχε αποθηκευτεί.
- Τα Applets δεν μπορούν να τρέξουν προγράμματα που υπάρχουν στο σύστημα του τοπικού χρήστη.

Επίσης για να δημιουργηθεί ένα Applet πρόγραμμα, πρέπει να δημιουργηθεί μια υποκλάση της κλάσης Applet, του java.applet πακέτου:

```
Public class my Class extends java.applet.Applet {  
.....  
}
```

Η κλάση Applet παρέχει συμπεριφορά που επιτρέπει στο Applet πρόγραμμα όχι μόνο να λειτουργεί μέσα στον Browser αλλά να έχει και δυνατότητες AWT για ενσωμάτωση User Interface στοιχείων, διαχείριση γεγονότων ποντικίου και πληκτρολογίου, καθώς και ζωγραφικής στην οθόνη. Παρόλο που ένα Applet πρόγραμμα μπορεί να αποτελείται από επιπλέον βοηθητικές κλάσεις, η υποκλάση της κλάσης Applet είναι αυτή που ενεργοποιεί την εκτέλεση του Applet προγράμματος.

## 2.12.1 Βασικές λειτουργίες των Applets

Για την δημιουργία Java εφαρμογών, η κλάση της εφαρμογής πρέπει να διαθέτει την μέθοδο `main()`. Όταν η εφαρμογή αρχίζει να τρέχει, εκτελείται η μέθοδος `main()` η οποία καθορίζει την συμπεριφορά του προγράμματος. Αντιθέτως στα Applets προγράμματα υπάρχουν διαφορετικές λειτουργίες που αντιστοιχούν σε γεγονότα που συμβαίνουν κατά την διάρκεια της ζωής του Applet (π.χ. γεγονότα αρχικοποίησης, ζωγραφικής, ποντικιού κλπ.).

Σε κάθε λειτουργία αντιστοιχεί και κάποια μέθοδος η οποία καλείται από τον Browser όταν ένα γεγονός συμβεί. Η μέθοδοι των λειτουργιών έτσι όπως ορίζονται στην Applet κλάση της Java δεν κάνουν τίποτε. Για να δώσουμε κάποια συμπεριφορά σε κάποιο γεγονός της Applet εφαρμογή μας πρέπει να ξανά ορίσουμε την μέθοδο που αντιστοιχεί στο γεγονός μέσα στην υποκλάση της Applet που δημιουργήσαμε. Οι πέντε πιο βασικές μέθοδοι μιας applet είναι:

```
1.  
Public void init () {  
.....  
}
```

Η μέθοδος αυτή εκτελείται όταν η applet εφαρμογή κατεβαίνει στο τοπικό υπολογιστή για εκτέλεση. Η μέθοδος αυτή μπορεί να περιλαμβάνει την δημιουργία κάποιων αντικειμένων, τον καθορισμό παραμέτρων, το φόρτωμα εικόνων ή font κλπ.

```
2.  
Public void start () {  
.....  
}
```

Η μέθοδος αυτή καλείται αμέσως μετά την `init()`. Η `start()` καλείται επίσης όταν η applet εφαρμογή είχε προηγουμένως σταματήσει την εκτέλεση της. Για παράδειγμα μια applet εφαρμογή σταματά την εκτέλεση της όταν ο χρήστης αλλάξει μέσω ενός συνδέσμου HTML σελίδα και ξαναρχίζει την εκτέλεση της όταν ο χρήστης γυρίσει πίσω στη σελίδα της Applet εφαρμογής.

```
3.  
Public void stop () {  
.....  
}
```

Η `stop()` σταματά την εκτέλεση της Applet εφαρμογής και είναι το συμπλήρωμα της `start()`. Ο χρήστης μπορεί επίσης να καλέσει από μόνος του την `stop` για να σταματήσει την Applet εφαρμογή.

```
4.  
Public void destroy () {  
.....  
}
```

Δίνει την δυνατότητα στη Applet εφαρμογή να ελευθερώσει τους πόρους του συστήματος που της είχαν διατεθεί. Η μέθοδος εκτελείται λίγο πριν η Applet εφαρμογή πάψει οριστικά την εκτέλεση της ή όταν ο Browser κλείσει. Συνήθως η μέθοδος αυτή δεν χρειάζεται να οριστεί στην Applet εφαρμογή εκτός από πολύ ειδικές περιπτώσεις.

```
5.  
Public void paint (Graphics g) {  
.....  
}
```

Με την μέθοδο αυτή η Applet εφαρμογή εμφανίζει κάτι στην οθόνη πχ. κείμενα, γραφικά, εικόνες. Η μέθοδος αυτή καλείται σε διάφορες περιπτώσεις όπως όταν η Applet εφαρμογή αρχικοποιείται, όταν μετακινείται ο Browser ή τοποθετείται πίσω από άλλο παράθυρο και μετά έρχεται πάλι μπροστά, όταν θέλουμε να δημιουργήσουμε animation οπότε και καλείται συνεχώς. Για να χρησιμοποιήσουμε την μέθοδο αυτή πρέπει προηγουμένως η κλάση των γραφικών να περιληφθεί στον κώδικα της Applet εφαρμογής.

## 2.13 Abstract Windowing Toolkit (AWT)

Οι κλάσεις του API της Java που χρησιμοποιούνται για την ανάπτυξη γραφικών διαπροσωπειών ανθρώπου-μηχανής αποτελούν το AWT (Abstract Windowing Toolkit). Όπως υποδηλώνει το όνομα του είναι ένα αφηρημένο (δηλαδή ανεξάρτητο υλοποίησης) σύνολο εργαλείων για τη δημιουργία μιας απλής Γραφικής Διεπαφής Χρήστη (Graphical User Interface - GUI). Το AWT παρέχει στους προγραμματιστές της Java μια διαπροσωπεία αρκετά υψηλού επιπέδου, αντίστοιχη του MFC στα Microsoft Windows. Χρησιμοποιεί ένα σύνολο συστατικών (components), που του παρέχει η πλατφόρμα στην οποία τρέχει κάθε φορά η εφαρμογή. Μερικά χρήσιμα components όπως θα δούμε παρακάτω είναι τα κουμπιά, τα μενού, κ.λπ. Οι εφαρμογές που βασίζονται στο AWT έχουν την ίδια συμπεριφορά και εμφάνιση με τις υπόλοιπες εφαρμογές της πλατφόρμας στην οποία τρέχουν. Επίσης δεν υπάρχει καμία εξάρτηση της γλώσσας Java από το AWT και μπορούμε αν θέλουμε να υλοποιήσουμε και να χρησιμοποιήσουμε άλλα εργαλεία και βιβλιοθήκες γραφικών.

### 2.13.1 Βασικά στοιχεία ενός GUI

Ένα Graphical User Interface - GUI (Γραφική Διεπαφή με το Χρήστη) είναι το μέρος του προγράμματος, που φροντίζει για τον τρόπο εμφάνισης και χειρισμού του προγράμματος από τον χρήστη. Ένα GUI αποτελείται από GUI-components. Οι κλάσεις που χρησιμοποιούνται για την κατασκευή αντικειμένων τύπου GUI-components ανήκουν στο java.awt (Abstract Windowing Toolkit) package. Οι βασικότερες από αυτές είναι η κλάση Component και η κλάση Container. Επίσης κάθε κλάση που κληρονομεί την κλάση Container είναι και αυτή ένα Container. Οι κλάσεις που συνηθέστερα χρησιμοποιούνται για την κατασκευή GUI-components περιγράφονται παρακάτω :

<b>Label</b>	Εμφανίζει κείμενο που δεν μπορεί να τροποποιηθεί από τον χρήστη.
<b>Button</b>	Περιοχή που προξενεί ένα γεγονός (event) όταν επιλέγεται με το ποντίκι.
<b>TextField</b>	Περιοχή όπου ο χρήστης εισάγει δεδομένα από το πληκτρολόγιο. Σε ένα TextField μπορούμε επίσης να εμφανίζουμε πληροφορίες.
<b>TextArea</b>	Περιοχή όπου ο χρήστης εισάγει δεδομένα από το πληκτρολόγιο. Σε ένα TextArea μπορούμε να έχουμε πολλές γραμμές κειμένου σε αντίθεση με το TextField που μπορούμε να έχουμε μόνο μία.
<b>Choice</b>	Μια λίστα στοιχείων από τα οποία ο χρήστης μπορεί να επιλέξει ένα από αυτά με το ποντίκι.
<b>Checkbox</b>	Ένα boolean component που είναι επιλεγμένο ή μη επιλεγμένο. Με αυτό υλοποιούνται και τα Radio buttons (αμοιβαίως αποκλειόμενες επιλογές).
<b>List</b>	Λίστα στοιχείων από τα οποία ο χρήστης μπορεί να επιλέξει ένα από αυτά με το ποντίκι. Διπλό κλικ σε ένα στοιχείο της λίστας προξενεί ένα action event.
<b>Panel</b>	Είναι ένα container αντικείμενο στο οποίο μπορούν να τοποθετηθούν component αντικείμενα.
<b>ScrollPane</b>	Είναι ένα container αντικείμενο στο οποίο μπορεί να τοποθετηθεί ένα component, συνήθως ένα Panel, το οποίο θα εμφανίζεται στον χρήστη αλλά κι όταν αυτό δεν είναι εφικτό θα εμφανίζονται Scrollbars που θα επιτρέψουν την διολίσθηση του component έτσι ώστε να γίνεται ορατό και το τμήμα του που δεν φαίνεται.

Πίνακας 2.13.1.α Περιγραφή των κλάσεων για την κατασκευή GUI-components.

### 2.13.2 Δομή ενός GUI

Σύνθετα GUIs χωρίζονται σε υποπεριοχές. Για να απλοποιηθεί, λοιπόν, η ανάπτυξη ενός GUI στην Java, χωρίζεται συνήθως σε πολλά Panel components. Αξίζει να αναφερθεί ότι η κλάση Panel



κληρονομεί την κλάση Container και η κλάση Applet την κλάση Panel. Έτσι τα Panels και τα Applets είναι Containers και μπορούν να έχουν Components, ακόμα και τύπου Panel. Ο κατασκευαστής της κλάσης Panel δεν παίρνει παραμέτρους. Τελικά προκύπτει μια δενδρική δόμηση του GUI που λέγεται component hierarchy ή containment hierarchy.

Σε κάθε Panel τα Components που περιέχει είναι τοποθετημένα με ένα συγκεκριμένο πλάνο ή σχέδιο (layout). Η Java παρέχει διαχειριστές πλάνων (Layout Managers) για την διευθέτηση των Component αντικειμένων μέσα σε ένα Applet ή Panel. Οι πιο συνηθισμένες κλάσεις των Layout Managers περιγράφονται παρακάτω :

<b>FlowLayout</b>	Είναι ο default Layout Manager για τα Applets και τα Panels. Τοποθετεί τα Component αντικείμενα από αριστερά προς τα δεξιά με την σειρά που προστίθενται
<b>BorderLayout</b>	Τοποθετεί τα Component αντικείμενα σε πέντε περιοχές: Βόρεια, Νότια, Ανατολική, Δυτική και Κεντρική (North, South, East, West, Center).
<b>GridLayout</b>	Τοποθετεί τα Component αντικείμενα σε γραμμές και στήλες με καθορισμένη σειρά (πρώτα γεμίζει η πρώτη γραμμή μετά η δεύτερη γραμμή κλπ).
<b>CardLayout</b>	Τοποθετεί τα Component αντικείμενα σε στοίβα. Κάθε Container αντικείμενο στη στοίβα μπορεί να χρησιμοποιήσει οποιονδήποτε Layout Manager. Μόνο το Container αντικείμενο που βρίσκεται στην κορυφή της στοίβας είναι ορατό.
<b>GridBagLayout</b>	Είναι παρόμοιος με τον GridLayout Manager. Διαφέρει από αυτόν στο ότι κάθε Component αντικείμενο που του προστίθεται μπορεί να έχει οποιοδήποτε μέγεθος (δηλαδή να καλύπτει περισσότερες από μια γραμμές και στήλες). Επίσης η σειρά με την οποία προστίθενται τα Component αντικείμενα στον GridBagLayout Manager μπορεί να είναι οποιαδήποτε.

**Πίνακας 2.13.2.α : Περιγραφή των κλάσεων των Layout Managers.**

## 2.14 Τα εργαλεία της Java

Ακολούθως παρουσιάζονται εν συντομία όλα τα εργαλεία της Java :

- **javac** Είναι ο compiler της Java. Η χρήση του στο command-line είναι javac<όνομα αρχείου>. Το javac δεν παράγει ένα αρχείο με όλο τον κώδικα, αλλά χωριστό αρχείο για κάθε κλάση. Τα αρχεία των κλάσεων ονομάζονται: <όνομα κλάσης>.class.
- **java** Είναι ο interpreter της Java. Η χρήση του είναι η εξής : java <κλάση>, π.χ. java myClass και όχι java myClass.class.
- **jdb** Είναι ο Java debugger.
- **javah** Κατασκευάζει C files και stub files για κάποια κλάση. Αυτά τα αρχεία είναι απαραίτητα όταν θέλουμε να υλοποιήσουμε κάποιες από τις μεθόδους της κλάσης σε C, πράγμα πολύ σπάνιο.
- **javap** Είναι ο Java disassembler.
- **javadoc** Είναι ένα πρόγραμμα για αυτόματη κατασκευή documentation. Είναι αρκετά χρήσιμο στην κατασκευή βοηθημάτων και τεχνικών αναφορών για εφαρμογές οποιουδήποτε μεγέθους.
- **applet viewer** Είναι ένα πρόγραμμα το οποίο μας επιτρέπει να τρέχουμε και να χρησιμοποιούμε τα διάφορα applets σε Java. Οι stand-alone εφαρμογές, ωστόσο, δεν τρέχουν σε applet viewer αλλά κατευθείαν στον java ή javaw.

## 2.15 Τι είναι το εργαλείο NetBeans

Το εργαλείο NetBeans παρέχει μια πλατφόρμα για την ανάπτυξη διάφορων εφαρμογών που είτε θα χρησιμοποιηθούν στα πλαίσια κάποιου δικτύου είτε θα χρησιμοποιηθούν ως ανεξάρτητες (Stand-alone) εφαρμογές.

Το εργαλείο NetBeans υποστηρίζει διάφορες γλώσσες προγραμματισμού. (Java, C, C++, PHP, Python, JavaScript). Παρέχει ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών (IDE: Integrated Development Enviroment). Το περιβάλλον ανάπτυξης γενικά παρέχει βοηθήματα στον προγραμματιστή τα οποία κάνουν ευκολότερη και αποδοτικότερη την ανάπτυξη των εφαρμογών. Επίσης τα βασικότερα από τα βοηθήματα αυτά αφορούν κυρίως τον Source Code Editor (συντακτικού στον κώδικα, live επισήμανση και live parsing, pop up παράθυρα βοήθειας). Ακόμη παρέχονται βοηθήματα για κατασκευή GUI άλλα και βοηθήματα διαχείρισης βάσεων δεδομένων.

### 3. Βάσεις Δεδομένων

Με τον όρο **βάση δεδομένων** εννοείται μία συλλογή από *συστηματικά οργανωμένα* (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων.

Οι βάσεις δεδομένων είναι σημαντικές διότι δίνουν σε μια σελίδα για παράδειγμα την έννοια της “εφαρμογής”. Δηλαδή η σελίδα δεν είναι απλά μία στατική σελίδα που απλά μας πληροφορεί για κάτι αλλά μπορούμε να αποθηκεύουμε δεδομένα, να τα επεξεργαζόμαστε να τα χρησιμοποιούμε γενικότερα. Έτσι, η σελίδα γίνεται μία δυναμική “σελίδα”. Εκατομμύρια sites χρησιμοποιούν βάσεις δεδομένων. Για παράδειγμα amazon, ebay, facebook κλπ. Ολόκληρες πλατφόρμες βασίζονται σε βάσεις δεδομένων όπως η πλατφόρμα blogging “blogspot” ,”wordpress” κλπ.

Αφού δημιουργήσουμε μια βάση δεδομένων με πολλές εγγραφές, τότε μπορούμε σε αυτήν να ψάξουμε και να βρούμε εύκολα και γρήγορα το/τα στοιχείο/α που επιθυμούμε. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων αποκαλείται **Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, DataBase Management System)** το οποίο είναι ένα σύνολο από προγράμματα που επιτρέπουν τον χειρισμό των δεδομένων μιας ή περισσότερων βάσεων δεδομένων που ανήκουν στο ίδιο σύστημα, και με την βοήθειά του μπορούμε να υποθηκεύσουμε, προσθέσουμε, τροποποιήσουμε, εμφανίσουμε ή και διαγράψουμε τα αποθηκευμένα δεδομένα.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ’ αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ’ έναν μόνο υπολογιστή ή και σ’ ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

Ένα *Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System)* αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το **υλικό (hardware)** αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το **λογισμικό (software)** είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.

Η *βάση δεδομένων (data base)* αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ’ έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή.

Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι :

- **Ολοκληρωμένα (Integrated)**, δηλ. τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός (redundancy) των ίδιων στοιχείων.
- **Καταμεριζόμενα (Shared)**, δηλ. να μπορούν περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Οι στόχοι μιας βάσης δεδομένων είναι οι εξής :

- **Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).** Εάν τα ίδια δεδομένα καταχωρηθούν στη βάση δυο φορές, τότε ανακύπτουν δυο σοβαρά προβλήματα. Το πρώτο πρόβλημα είναι ότι σπαταλούμε άσκοπα αποθηκευτικό χώρο στο σκληρό δίσκο, αφού την ίδια πληροφορία την αποθηκεύουμε δυο φορές. Το δεύτερο και σοβαρότερο πρόβλημα, είναι ότι υπάρχει ο κίνδυνος δημιουργίας ασυνεπών δεδομένων (inconsistent data). Εάν κρατάμε δύο φορές την ίδια πληροφορία και η πληροφορία αυτή σε κάποια χρονική στιγμή υποστεί κάποιο είδος επεξεργασίας τότε η επεξεργασία αυτή θα πρέπει να εφαρμοσθεί και στις δύο καταχωρήσεις που αφορούν το ίδιο δεδομένο, διότι διαφορετικά, η βάση θα περιέχει δεδομένα που δεν είναι συνεπή. Για το λόγο αυτό, ένας από τους πρώτους ελέγχους που πραγματοποιούμε στη βάση αμέσως μετά το σχεδιασμό της, είναι ο **έλεγχος παρουσίας επαναλαμβανόμενων πεδίων, και η απομάκρυνσή τους, εφ' όσον υπάρχουν.**
- **Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.**
- **Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.**
- **Η επιβολή κανόνων ασφαλείας (security).** Όπως να απαγορεύει την πρόσβαση στα δεδομένα μη εξουσιοδοτημένων ατόμων. Αυτό ισχύει κυρίως σε μεγάλες βάσεις δεδομένων με πολλούς χρήστες, και η τεχνική που συνίσταται είναι ο καθορισμός ομάδων χρηστών (user groups) με διαφορετικά δικαιώματα πρόσβασης στον καθένα από αυτούς. Ο κάθε χρήστης λαμβάνει ένα κωδικό πρόσβασης (password) και τα καθήκοντα που μπορεί να επιτελέσει είναι εντελώς συγκεκριμένα και καθορισμένα εκ των προτέρων.
- **Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.** Η βάση θα πρέπει να διαθέτει σύστημα δημιουργίας αντιγράφων ασφαλείας των δεδομένων που είναι καταχωρημένα σε αυτή (backups). Η ταυτόχρονη αποθήκευση των δεδομένων σε περισσότερους από ένα δίσκους, είναι μια εργασία επιβεβλημένη, προκειμένου να είναι δυνατή η ανάκτησή τους σε περιπτώσεις κατάρρευσης της βάσης για οποιοδήποτε λόγο.
- **Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.**

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται (οργανώνονται) στις εξής στοιχειώδεις μορφές :

- **Πεδίο (Field),** είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.

Ένα πεδίο χαρακτηρίζεται ακόμη και από το είδος των δεδομένων που μπορεί να περιέχει, όπως :

- ❖ **Αλφαριθμητικό (alphanumeric),** μπορεί να περιέχει γράμματα, ψηφία ή και ειδικούς χαρακτήρες.
- ❖ **Αριθμητικό (numeric),** μπορεί να περιέχει μόνο αριθμούς.
- ❖ **Αλφαβητικό (alphabetic),** μπορεί να περιέχει μόνο γράμματα (αλφαβητικούς χαρακτήρες).
- ❖ **Ημερομηνίας (date),** μπορεί να περιέχει μόνο ημερομηνίες.
- ❖ **Διαδικό (binary),** μπορεί να περιέχει ειδικού τύπου δεδομένα, όπως εικόνες, ήχους κ.ά.
- ❖ **Λογικό (logical),** μπορεί να περιέχει μόνο μία από δύο τιμές, οι οποίες αντιστοιχούν σε δύο διακριτές καταστάσεις και μπορούν να χαρακτηρισθούν σαν 0 και 1 ή σαν αληθές (true) και ψευδές (false).
- ❖ **Σημειώσεων (memo),** μπορεί να περιέχει κείμενο με μεταβλητό μήκος, το οποίο μπορεί να είναι και αρκετά μεγάλο και συνήθως αποθηκεύεται σαν ξεχωριστό αρχείο από το κύριο αρχείο.

- **Εγγραφή (Record)**, είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.

Όσον αφορά τις εγγραφές, χρήσιμοι ορισμοί είναι οι εξής :

- ❖ **Μήκος εγγραφής (record length)** καλείται το άθροισμα που προκύπτει από τα μήκη των πεδίων που την αποτελούν.
- ❖ **Δομή εγγραφής (record layout)** ή **γραμματογράφηση** καλείται ο τρόπος με τον οποίο οργανώνουμε τα πεδία μιας εγγραφής.
- ❖ **Διάβασμα (read)** από αρχείο σημαίνει τη μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από το μέσο αποθήκευσης (σκληρό δίσκο ή δισκέτα) στην κεντρική μνήμη του υπολογιστή για επεξεργασία.
- ❖ **Γράψιμο (write)** σε αρχείο σημαίνει μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από την κεντρική μνήμη του υπολογιστή στο μέσο αποθήκευσης (σκληρό δίσκο ή δισκέτα).

- **Αρχείο (File)**, είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.
- **Πρωτεύον Κλειδί (Primary Key)**, είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.
- **Κλειδί (Key)**, είναι ένα πεδίο που δεν έχει κατ' ανάγκη μοναδική τιμή και που μπορούμε να το χρησιμοποιήσουμε για να κάνουμε αναζήτηση σ' ένα αρχείο.
- **Ξένο Κλειδί (Foreign Key)**, είναι μια ιδιότητα (πεδίο) που είναι πρωτεύον κλειδί σε μια οντότητα (πίνακας) αλλά που υπάρχει και σε μια άλλη οντότητα (πίνακας) σαν απλή ιδιότητα. Τα ξένα κλειδιά είναι απαραίτητα για να μπορέσουμε να κάνουμε τις συσχετίσεις (συνδέσεις, επικοινωνίες) ανάμεσα στις οντότητες (πίνακες).

### 3.1 Τα τρία βασικά μοντέλα

Υπάρχουν τρία βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το ιεραρχικό, το δικτυωτό και το σχεσιακό, και τα οποία αναπτύχθηκαν με βάση αντίστοιχες δομές.

#### 3.1.1 Το Ιεραρχικό Μοντέλο Βάσεων Δεδομένων

Το ιεραρχικό μοντέλο (hierarchical) έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες.

Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μία και μόνο μία οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίστηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

### 3.1.2 Το Δικτυωτό Μοντέλο Βάσεων Δεδομένων

Και στο δικτυωτό (network) μοντέλο, τα στοιχεία τοποθετούνται σ' ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σ' ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

### 3.1.3 Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων

Το σχεσιακό (relational) μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές (relational databases). Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε πολύ μεγάλες βάσεις δεδομένων. Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα.

Στις **Σχεσιακές (Relational)** βάσεις δεδομένων, τα δεδομένα συνδέονται μεταξύ τους με **σχέσεις (relations)**, οι οποίες προκύπτουν από τα κοινά πεδία που υπάρχουν σε διαφορετικά αρχεία. Τα αρχεία αποκαλούνται **πίνακες (tables)**, οι εγγραφές **γραμμές (rows)** και τα πεδία **στήλες (columns)**. Η ύπαρξη μιας κοινής τιμής στα πεδία δύο αρχείων καθορίζει και μια σχέση μεταξύ των γραμμών διαφορετικών πινάκων. Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται **πεδίο ορισμού (domain)**. Οι σχεσιακές βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι λογικά κατανοητές και πολύ ευέλικτες και δεκτικές σε αλλαγές.

Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία που έχουμε να κάνουμε κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι να ορίσουμε τους πίνακες που θα χρησιμοποιήσουμε καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται κατασκευή του **σχήματος (schema)** μιας βάσης δεδομένων.

Οι κανόνες που πρέπει να ακολουθούμε πιστά κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας.
- Η κάθε στήλη του πίνακα αντιστοιχεί σε μια ιδιότητα της οντότητας.
- Η κάθε γραμμή του πίνακα αντιστοιχεί σε μια εμφάνιση της οντότητας.
- Η κάθε γραμμή πρέπει να είναι μοναδική, δηλ. αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.

## 3.2 Στάδια σχεδίασης - Μοντέλο Οντοτήτων Συσχετίσεων

Το μοντέλο που έχει επικρατήσει σήμερα για να παραστήσει τις έννοιες ή τη δομή μιας βάσης δεδομένων είναι το **Μοντέλο Οντοτήτων -Συσχετίσεων (ΟΣ)**. Οι βασικές (θεμελιώδεις) έννοιες του μοντέλου αυτού είναι οι εξής :

- Οντότητες
- Ιδιότητες ή Χαρακτηριστικά
- Συσχετίσεις

Για να αναπαραστήσουμε ένα Μοντέλο Οντοτήτων – Συσχετίσεων χρησιμοποιούμε ειδικά διαγράμματα, όπου τα ορθογώνια συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις και οι ελλείψεις τις ιδιότητες. Με ευθείες γραμμές συνδέουμε τις οντότητες που συσχετίζονται με κάποιο τρόπο μεταξύ τους. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων, μια εργασία που είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των στοιχείων (πληροφοριών) της βάσης δεδομένων.

Το μοντέλο οντοτήτων – συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα δεδομένα (εννοιολογικό), χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.

### 3.2.1 Οι Οντότητες

Με τον όρο *οντότητα (entity)* αναφερόμαστε σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση που έχει μια ανεξάρτητη ύπαρξη. Είναι κάτι που ξεχωρίζει και μπορούμε να συγκεντρώσουμε πληροφορίες (στοιχεία) γι' αυτό. Η οντότητα είναι αντίστοιχη με την έννοια της εγγραφής που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια του αντικειμένου στις σύγχρονες αντικειμενοστραφείς γλώσσες προγραμματισμού.

Παραδείγματα οντοτήτων είναι τα εξής : ΜΕΛΟΣ, ΕΣΟΔΑ, ΕΞΟΔΑ, ΚΙΝΗΣΕΙΣ ΜΕΛΟΥΣ κ.ά.

Μια βάση δεδομένων μπορεί να περιέχει πολλές διαφορετικές οντότητες, οι οποίες απεικονίζονται με ορθογώνια παραλληλόγραμμα και συσχετίζονται μεταξύ τους ανά δύο.

### 3.2.2 Οι Ιδιότητες (Χαρακτηριστικά) των Οντοτήτων

Με τον όρο *ιδιότητες (properties)* ή *χαρακτηριστικά (attributes)* αναφερόμαστε στα συστατικά (δομικά) στοιχεία που προσδιορίζουν (αποτελούν) μια οντότητα. Η ιδιότητα είναι αντίστοιχη με την έννοια του πεδίου που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια της μεταβλητής στις γλώσσες προγραμματισμού.

Για παράδειγμα, η οντότητα ΜΕΛΟΣ μπορεί να αποτελείται από τις ιδιότητες (χαρακτηριστικά) Αριθμός Μητρώου, Επώνυμο, Όνομα, Πατρώνυμο, Ειδικότητα, Έτος Γέννησης, Διεύθυνση, ΑΦΗ, Τηλέφωνο, Κινητό κ.ά.

Απ' όλες τις ιδιότητες μιας οντότητας, υπάρχει μία μόνο ιδιότητα, και σπανιότερα ένας συνδυασμός δύο ή και περισσότερων ιδιοτήτων, η τιμή της οποίας είναι μοναδική και προσδιορίζει την κάθε εμφάνιση (στιγμιότυπο) της οντότητας και αποκαλείται *πρωτεύον κλειδί (primary key)*. Το πρωτεύον κλειδί εμφανίζεται στα διαγράμματα με υπογράμμιση ή με έντονη γραφή ή έχει ως πρόθεμα τον χαρακτήρα #.

### 3.2.3 Τα κλειδιά

Όπως είδαμε και νωρίτερα, με τον όρο *κλειδί (key)* ή πιο σωστά *πρωτεύον κλειδί (primary key)* αναφερόμαστε σε μια ιδιότητα (πεδίο), ή σπανιότερα σ' ένα σύνολο ιδιοτήτων (πεδίων), η τιμή της οποίας είναι μοναδική σ' ολόκληρη την οντότητα (πίνακα). Στην πράξη, το πρωτεύον κλειδί έχει

διαφορετική τιμή για κάθε εμφάνιση της οντότητας ή για κάθε γραμμή (εγγραφή) του πίνακα και ποτέ δεν μπορεί να έχει μηδενική (κενή) τιμή (null). Προσοχή, άλλο πράγμα είναι ο αριθμός 0 και άλλο πράγμα είναι η κενή τιμή (null), δηλ. η μη ύπαρξη τιμής.

Ο συνδυασμός δύο ή και περισσότερων ιδιοτήτων (πεδίων) για τη δημιουργία ενός πρωτεύοντος κλειδιού αποκαλείται *σύνθετο κλειδί*. Ένα παράδειγμα σύνθετου κλειδιού θα μπορούσε να είναι ο συνδυασμός των ιδιοτήτων Επώνυμο, Όνομα και Πατρώνυμο, εφόσον φυσικά είμαστε απολύτως βέβαιοι ότι δεν υπάρχουν δύο ή και περισσότερα άτομα με κοινές τιμές στις παραπάνω ιδιότητες.

*Ξένο κλειδί* αποκαλείται μια ιδιότητα (πεδίο) που είναι πρωτεύον κλειδί σε μια οντότητα (πίνακας) αλλά που υπάρχει και σε μια άλλη οντότητα (πίνακας) σαν απλή ιδιότητα. Τα ξένα κλειδιά είναι απαραίτητα για να μπορέσουμε να κάνουμε τις συσχετίσεις (συνδέσεις, επικοινωνίες) ανάμεσα στις οντότητες (πίνακες).

### 3.2.4 Συσχετίσεις

Το τρίτο χαρακτηριστικό γνώρισμα που περιγράφει τη δομή ενός μοντέλου οντοτήτων συσχετίσεων, είναι η συσχέτιση (relationship), η οποία καθορίζει με ποιο τρόπο οι τύποι οντότητας μιας βάσης δεδομένων, συσχετίζονται μεταξύ τους.

Κάθε τύπος οντότητας που συμμετέχει σε κάποιο τύπο συσχέτισης, παίζει και ένα συγκεκριμένο **ρόλο** σε αυτή τη συσχέτιση. Το **όνομα** αυτού του ρόλου (role name) καθορίζει σε πολύ μεγάλο βαθμό **το πώς οι διάφοροι τύποι οντότητας αλληλεπιδρούν μεταξύ τους**.

### 3.2.5 Χαρακτηριστικά τύπων συσχέτισης

Ο κάθε τύπος συσχέτισης που ορίζεται ανάμεσα σε δύο τύπους οντότητας κατά τη λογική σχεδίαση μιας βάσης δεδομένων, χαρακτηρίζεται από δύο ιδιότητες που καθορίζουν το πλήθος και το είδος της συμμετοχής των δύο οντοτήτων στα στιγμιότυπα αυτού του τύπου συσχέτισης. Αυτές οι δύο ιδιότητες ορίζονται με τον ακόλουθο τρόπο:

Η **πολλαπλότητα (cardinality)** ενός τύπου συσχέτισης, καθορίζει **το πλήθος των στιγμιότυπων αυτού του τύπου συσχέτισης, στον οποίο μια οντότητα μπορεί να συμμετάσχει**.

Ένα είδος πολλαπλότητας του τύπου συσχέτισης που ορίζεται ανάμεσα στους τύπους οντότητας μπορεί να είναι **1:N (ένα προς πολλά)**. Εκτός από αυτού του είδους την πολλαπλότητα ένας τύπος συσχέτισης μπορεί να χαρακτηρίζεται από πολλαπλότητα **1:1 (ένα προς ένα)** ή **M:N (πολλά προς πολλά)**.

Η δεύτερη σημαντική ιδιότητα που χαρακτηρίζει ένα τύπο συσχέτισης έχει να κάνει με το εάν η συμμετοχή των δύο τύπων οντότητας σε αυτή τη συσχέτιση είναι **ολική ή μερική**. Ένας τύπος οντότητας χαρακτηρίζεται από **ολική συμμετοχή (total participation)** σε ένα τύπο συσχέτισης, **εάν η ύπαρξή του εξαρτάται από τη συμμετοχή του ή όχι σε αυτή τη συσχέτιση**.

Είναι τέλος σημαντικό να αναφερθεί, πως ένας τύπος συσχέτισης, εκτός από τους τύπους οντότητας μεταξύ των οποίων ορίζεται, μπορεί να χαρακτηρίζεται και από την ύπαρξη πεδίων (attributes).

### 3.2.6 Αδύναμοι τύποι οντότητας (Weak Entity Types)

Σε ορισμένες περιπτώσεις είναι δυνατό να ορίσουμε τύπους οντότητας οι οποίοι **δεν έχουν κάποιο πεδίο κλειδί έτσι ώστε να ξεχωρίζουν τα διάφορα στιγμιότυπά τους, αλλά αυτή η διάκριση γίνεται δια της συσχέτισής τους με κάποιον από τους υπόλοιπους τύπους οντότητας** της βάσης. Αυτοί οι τύποι οντότητας ονομάζονται αδύναμοι τύποι οντότητας (weak entity types) και ορίζονται μόνο από ένα συνδυασμό κάποιων από τα πεδία που περιέχουν καθώς και από τη συσχέτισή



τους (identifying relationship) με κάποιον τύπο οντότητας (identifying owner). Ας σημειωθεί πως σε αυτή τη συσχέτιση, οι εν λόγω τύποι οντότητας έχουν ολική συμμετοχή (total participation).

Εάν δύο ή περισσότεροι αδύναμοι τύποι οντότητας έχουν τον ίδιο **κάτοχο (owner entity)** υπάρχει ένας συνδυασμός πεδίων ο οποίος βοηθά να ξεχωρίζουμε τα στιγμιότυπα αυτού του τύπου οντότητας μεταξύ τους. Αυτός ο συνδυασμός των πεδίων λέγεται **μερικό κλειδί (partial key)**.

### 3.3 Χρήστες και δικαιώματα

Οι *χρήστες (users)* μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :

- *Τελικοί χρήστες (end users)*. Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τούς επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- *Προγραμματιστές εφαρμογών (application programmers)*. Αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
- *Διαχειριστής δεδομένων (data administrator – DA)*. Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- *Διαχειριστής βάσης δεδομένων (database administrator – DBA)*. Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ.

### 3.4 Η αρχιτεκτονική των ΣΔΒΔ

Όπως είδαμε νωρίτερα, ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τις διεκπεραιώσει.

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία αποκαλείται *λογική διεπαφή*. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν κοκ και αυτή η ιδιότητα είναι γνωστή ως *ανεξαρτησία δεδομένων*.

Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές: ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιότερων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

- *Εσωτερικό επίπεδο (internal level)*, έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους.
- *Εξωτερικό επίπεδο (external level)*, έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και οι διαχειριστές της βάσης δεδομένων.
- *Εννοιολογικό επίπεδο (conceptual level)*, είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων.

### 3.5 Τα σχεσιακά ΣΔΒΔ (RDBMS)

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational DataBase Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων.

Για να μπορέσουμε να προσθέσουμε, διαγράψουμε ή τροποποιήσουμε τα στοιχεία που περιέχονται σε μια βάση δεδομένων, χρησιμοποιούμε ειδικές γλώσσες προγραμματισμού. Η γλώσσα που αποτελεί σήμερα ένα διεθνές πρότυπο για την επικοινωνία των χρηστών με τα Σχεσιακά ΣΔΒΔ είναι η *MYSQL*.

Τα Σχεσιακά ΣΔΒΔ τα διακρίνουμε στα *μεγάλα*, τα οποία αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα, και τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά. και τα *μικρά*, τα οποία αφορούν κυρίως απλούς χρήστες, όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

## 4. MySQL

Η βάση δεδομένων **MySQL®** έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα λόγω της αυξημένης απόδοσης, υψηλής αξιοπιστίας και ευκολίας στην διαχείριση της. Η MySQL θεωρείται μια από τις πιο αξιόπιστες πλατφόρμες για την συγκέντρωση δεδομένων και για αυτό έχει κερδίσει την εκτίμηση τόσο των administrators όσο και των programmers. Διατίθεται εντελώς δωρεάν και θα την συναντήσουμε να υποστηρίζει εφαρμογές από το τομέα των τηλεπικοινωνιών, και ηλεκτρονικών καταστημάτων στο διαδίκτυο ως και συστήματα εξυπηρέτησης πελατών και μικροσυσκευές όπως PDAs. Χρησιμοποιείται σε περισσότερες από 6 εκατομμύρια εγκαταστάσεις κλιμακούμενες από μεγάλες εταιρίες μέχρι εξειδικευμένες εφαρμογές με ενσωματωμένες βάσεις δεδομένων σε όλο τον κόσμο.

Η **MySQL** καθίσταται η απόλυτη επιλογή μιας νέας γενιάς εφαρμογών που υλοποιούνται σε περιβάλλον LAMP (Linux, Apache, MySQL, PHP / Perl / Python).

Επίσης η MySQL λειτουργεί σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων των Linux, Windows, OS/X, HP - UX, AIX, Netware. Προϋπόθεση για να έχουμε και να λειτουργούμε MySQL βάσεις δεδομένων είναι να μας παρέχετε ένας MySQL server. Η MySQL είναι το λογισμικό του διακομιστή βάσεων δεδομένων (database server software) που χρησιμοποιούμε.

Χρήσιμα εργαλεία είναι ο MySQL Administrator που έχει σχεδιαστεί για την διαχείριση ενός MySQL εξυπηρετητή και ο MySQL Query Browser που έχει σχεδιαστεί για να μας βοηθήσει να θέτουμε αιτήματα και να αναλύουμε δεδομένα, τα οποία είναι αποθηκευμένα στην MySQL βάση δεδομένων μας.

Η MySQL ανήκει στην γενική κατηγορία των DBMS (Database Management Systems) που έχουν σαν πρωταρχικό τους ρόλο την αποθήκευση των δεδομένων, την ελεγχόμενη και ασφαλή πρόσβαση στις πληροφορίες, και την διαχείριση των δικαιωμάτων με τα οποία οι χρήστες μπορούν να επέμβουν και να αλλάξουν στοιχεία πληροφοριών. Κάθε φορά που ένας χρήστης προσπαθεί να διαχειριστεί ένα όγκο δεδομένων προσθέτοντας νέα στοιχεία ή αφαιρώντας κάποια άλλα που δεν ισχύουν πια, το DBMS είναι υπεύθυνο να ακολουθήσει αυτή την διαδικασία από την αρχή της ενεργοποίησης της μέχρι το τέλος της. Παίζει το ρόλο του “μεσάζων” γιατί απλά το DBMS δεν αφήνει το χρήστη να έχει άμεση πρόσβαση στα δεδομένα. Οι εντολές απευθύνονται αποκλειστικά στο DBMS και αφού ελεγχθούν για την εγκυρότητα τους τότε το σύστημα αναλαμβάνει την μεταφορά της πληροφορίας στο χρήστη που την αναζητήσε.

Η MySQL ανήκοντας στην κατηγορία των DBMS μπορεί να αναλάβει την διαχείριση πολλών βάσεων μαζί. Κατά την εγκατάσταση έχει ήδη δημιουργηθεί μια βάση με το όνομα mysql η οποία χρησιμοποιείται αποκλειστικά για την καταγραφή πληροφοριών του συστήματος.

Αυτές οι πληροφορίες καταγράφουν την δημιουργία νέων βάσεων, νέων πινάκων, ή ακόμα ενέργειες που έχουν ζητηθεί από τον χρήστη. Με άλλα λόγια, για οποιοδήποτε νέο στοιχείο που δημιουργείται μετά την εγκατάσταση της MySQL, πρέπει το σύστημα να ενημερώνεται και να καταγράφει την ύπαρξη του.

### 4.1 Γιατί MySQL

Παρέχει απλοποίηση στην πρόσβαση, βελτίωση στις επιδόσεις, την επεκτασιμότητα και την αξιοπιστία και ικανοποίηση στην εκθετική αύξηση των απαιτήσεων σε αποθήκευση με δραματικά χαμηλότερο κόστος. Επωφελοούμαστε από την εξειδίκευση, τις βέλτιστες πρακτικές, την εξυπηρέτηση, την υποστήριξη και τη διαχείριση της MySQL για σύνθετα περιβάλλοντα.

Η MySQL μειώνει το συνολικό κόστος κτήσης του λογισμικού βάσης δεδομένων. Επίσης υπάρχει μείωση του κόστους αδειοδότησης της βάσης δεδομένων κατά 90 τοις εκατό, μείωση του χρόνου εκτός λειτουργίας των συστημάτων κατά 60 τοις εκατό, μείωση των δαπανών για υλικό εξοπλισμού κατά 70 τοις εκατό και μείωση του κόστους διαχείρισης, τεχνικού σχεδιασμού και υποστήριξης έως και 50 τοις εκατό.

Παρακάτω αναφέρονται κάποια χαρακτηριστικά και κάποια πλεονεκτήματα της MySQL :

### **Επεκτασιμότητα και ευελιξία**

Τα συστήματα της Sun για MySQL μπορούν να επεκταθούν, για να ικανοποιήσουν την εκθετική αύξηση του όγκου δεδομένων που χαρακτηρίζει τα εμπλουτισμένα πολυμέσα, τις Διαδικτυακές κοινότητες και άλλες υπηρεσίες Web. Εκτελείται οτιδήποτε από βαθιά ενσωματωμένες εφαρμογές με καταλαμβανόμενο χώρο μόλις 1MB, ή μαζικές "αποθήκες" δεδομένων - με terabyte πληροφοριών. Επίσης τα μοναδικά στο είδος τους συστήματα της Sun μπορούν να "επεκταθούν μέσα στο διακομιστή, ενώ η τεχνολογία εικονικοποίησης της Sun μπορεί να μειώσει τις απαιτήσεις σε κατανάλωση ρεύματος και χώρο και να επιτύχει εξοικονόμηση κόστους και μεγαλύτερο σεβασμό προς το περιβάλλον.

### **Υψηλή διαθεσιμότητα.**

Εκτέλεση διαμορφώσεων υψηλής ταχύτητας κύριας /εξαρτώμενης αναπαραγωγής βασισμένη σε γραμμές και υβριδική αναπαραγωγή. Οι εξειδικευμένοι διακομιστές συμπλεγμάτων προσφέρουν άμεση μεταγωγή μετά από αστοχία.

### **Πλεονεκτήματα αποθήκευσης δεδομένων και web.**

Υψηλής απόδοσης μηχανισμός ερωτημάτων. Εξαιρετικά γρήγορη δυνατότητα εισαγωγής δεδομένων. Ισχυρή υποστήριξη για εξειδικευμένες λειτουργίες web - συμπεριλαμβανομένων των γρήγορων αναζητήσεων πλήρους κειμένου.

### **Ισχυρή προστασία δεδομένων.**

Πανίσχυροι μηχανισμοί διασφάλισης της προσπέλασης μόνο από εξουσιοδοτημένους χρήστες. Υποστήριξη SSH και SSL για ασφαλείς και σίγουρες συνδέσεις. Ισχυρή κρυπτογράφηση δεδομένων και λειτουργίες αποκρυπτογράφησης.

### **Ολοκληρωμένη ανάπτυξη εφαρμογών.**

Υποστήριξη για υποθηκευμένες διαδικασίες, κανόνες ενεργοποίησης, λειτουργίες, προβολές, δρομείς, γλώσσες SQL προτύπου ANSI, και πολλά περισσότερα. Βιβλιοθήκες προσθέτων για την ενσωμάτωση της υποστήριξης βάσεων δεδομένων MySQL σχεδόν σε κάθε εφαρμογή.

### **Ευκολία διαχείρισης.**

Υπάρχει χρήση του προγραμματισμού συμβάντων - αυτόματος προγραμματισμός συνήθων επαναλαμβανόμενων εργασιών βασισμένων σε SQL για εκτέλεση σε διακομιστή βάσης δεδομένων. Επίσης ο μέσος χρόνος από τη λήψη λογισμικού μέχρι την ολοκλήρωση της εγκατάστασης είναι λιγότερος από δεκαπέντε λεπτά.

### **Ελευθερία ανοικτού πηγαίου κώδικα και υποστήριξη 24 ώρες το εικοσιτετράωρο, 7 ημέρες την εβδομάδα.**

Μπορεί να προσφέρει εικοσιτετράωρη υποστήριξη και διαθέσιμη επανόρθωση μέσω του δικτύου MySQL. Εταιρική ποιότητα και έτοιμο για εταιρική χρήση από την εγκατάσταση ως την υποστήριξη.

### **Το χαμηλότερο συνολικό κόστος κτήσης.**

Εξοικονόμηση κόστους αδειοδότησης βάσεων δεδομένων και εξόδων για υλικό εξοπλισμού με ταυτόχρονη μείωση του χρόνου εκτός λειτουργίας.

### **Με γνώμονα την ταχύτητα.**

Χάρη στο συνδυασμό πρωτοποριακών τεχνολογιών, της εξειδίκευσης σε συστήματα και των προηγμένων μεθόδων ρύθμισης της Sun, τα συστήματα της Sun για MySQL μας προσφέρουν μια αποτελεσματική και ολοκληρωμένη λύση MySQL. Έτσι, είναι πλέον ευκολότερο για μας να υλοποιήσουμε MySQL και να ικανοποιήσουμε απαιτήσεις για νέες υπηρεσίες Web – με αύξηση των επιδόσεων μέχρι και 3 φορές.

### **Απλότητα πάνω απ' όλα.**

Παρέχει από τα εισαγωγικά συστήματα μέχρι τους πανίσχυρους διακομιστές, με επεξεργαστές Intel, AMD ή SPARC, μεγάλη μνήμη, ταχύτερες λειτουργίες I/O και τις πλέον πρόσφατες τεχνολογίες. Ακόμη όλα τα συστήματα της Sun για MySQL βελτιστοποιούνται και προσαρμόζονται σύμφωνα με τα εκάστοτε περιβάλλοντα MySQL. Επίσης μπορούμε να εκτελέσουμε το λειτουργικό σύστημα της επιλογής μας: Linux, Windows, Solaris 10 ή OpenSolaris.

### **Μείωση στους κινδύνους με το MySQL Experts.**

Οι βέλτιστες πρακτικές, οι αρχιτεκτονικές αναφορές και τα blueprint της Sun, που διατίθενται προς λήψη δωρεάν, σε συνδυασμό με τις δεξιότητες ρύθμισης και υλοποίησης MySQL της Sun και τον οργανισμό επαγγελματικών υπηρεσιών της Sun, μπορούν να μας βοηθήσουν να συνθέσουμε γρήγορα και με ασφάλεια τη δική μας υποδομή Web.

## **4.2 Δημιουργία Βάσης**

Για να δημιουργήσουμε μια βάση χρησιμοποιούμε την εντολή `create database`:

```
mysql> create database members;
```

Για να εργαστούμε με μία συγκεκριμένη βάση χρησιμοποιούμε την εντολή `use`:

```
mysql> use members;
```

Για να δημιουργήσουμε έναν πίνακα χρησιμοποιούμε την εντολή `create table`:

```
CREATE TABLE members (  
  Id INT (11) NULL,  
  
  name VARCHAR(255) NOT NULL,  
  city VARCHAR (255) NOT NULL,  
  state VARCHAR (50) NOT NULL,  
  PRIMARY KEY (id)  
)  
ENGINE=InnoDB;
```

Για να σβήσουμε έναν πίνακα χρησιμοποιούμε την εντολή drop table tablename.

Για να αλλάξουμε τον ορισμό ενός πίνακα χρησιμοποιούμε την εντολή alter table tablename.

#### 4.2.1 Εισαγωγή Δεδομένων

Για την εισαγωγή δεδομένων χρησιμοποιείται η εντολή insert into.

Για να εισάγουμε τρία μέλη θα δώσουμε:

```
INSERT INTO members (name, city, state)
```

```
VALUES ('Anna Rasoylh', 'Chania', 'PL')
```

```
INSERT INTO members (name, city, state)
```

```
VALUES ('Rena Zaxou', 'Heraklion', 'PL')
```

```
INSERT INTO members (name, city, state)
```

```
VALUES ('Maria Nikolaou', 'Heraklion', 'PL')
```

#### 4.2.2 Επιλογή Δεδομένων

Η επιλογή δεδομένων στη γλώσσα SQL γίνεται με τη χρήση της εντολής select.

Για να επιλέξουμε όλα τα στοιχεία για τα μέλη δίνουμε:

```
select * from members;
```

Για να επιλέξουμε όνομα μέλους δίνουμε:

```
Select member_name
```

```
From members;
```

#### 4.2.3 Ενημερώσεις και Διαγραφές

Για να ενημερώσουμε υπάρχοντα δεδομένα στη βάση χρησιμοποιούμε την εντολή update.

Για να διαγράψουμε δεδομένα χρησιμοποιούμε την εντολή delete.

## 4.3 JDBC

Η τεχνολογία JDBC (σημαίνει *Java Database Connectivity*) μας επιτρέπει να έχουμε πρόσβαση σε μία σχεσιακή βάση δεδομένων μέσα από το πρόγραμμά μας. Αυτή η τεχνολογία βασίζεται στην ύπαρξη οδηγών (*drivers*), οι οποίοι είναι αρμόδιοι για την επικοινωνία μεταξύ του προγράμματος και της βάσης. Με τη χρήση του κατάλληλου οδηγού συνδεόμαστε με τη βάση, στέλνουμε εντολές και λαμβάνουμε αποτελέσματα. Επίσης οι απαραίτητες κλάσεις ορίζονται από τη βιβλιοθήκη `java.sql`.

### 4.3.1 Χρήση του Οδηγού

Ο οδηγός JDBC για τη βάση MySQL ονομάζεται `Connector/J`. Για να τον χρησιμοποιήσουμε πρέπει να είναι διαθέσιμος στην ιδεατή μηχανή κατά τη διάρκεια της εκτέλεσης.

Όταν αποσυμπιέσουμε το αρχείο διανομής του `Connector/J` θα βρούμε τον οδηγό σε ένα αρχείο της μορφής `mysql-connector-java-[version]-stable-bin.jar`.

Τοποθετούμε το αρχείο του οδηγού σε σημείο που πρέπει να θυμόμαστε να το συμπεριλάβουμε στο `classpath` κατά την εκτέλεση του προγράμματος. Εναλλακτικά, αν `JAVA_HOME` είναι ο κατάλογος εγκατάστασης της γλώσσας Java στο σύστημά μας και `JRE_HOME` είναι ο κατάλογος εγκατάστασης του `runtime environment` (συνήθως κάτω από τον κατάλογο `Program Files`), μπορούμε να αντιγράψουμε το αρχείο του οδηγού στον κατάλογο `JAVA_HOME\jre\lib\ext` και στον κατάλογο `\lib\ext`.

## 5. Ανάλυση Λογισμικού

### 5.1 Περιγραφή συστήματος

Το σύστημα που θα κατασκευαστεί θα πρέπει να λειτουργεί με Sun Java και να αποθηκεύει τα δεδομένα που απαιτούνται σε σύστημα βάσης δεδομένων MySQL έτσι ώστε να εκμεταλλευτεί τα πλεονεκτήματα που αυτά παρέχουν.

Η εφαρμογή θα παρέχει πολλές δυνατότητες στον χρήστη όπως για παράδειγμα να καταχωρεί πελάτες στο σύστημα με διάφορα πεδία για τον κάθε ένα από αυτούς. Επίσης θα κάνει αναζήτηση των στοιχείων των πελατών που είναι καταχωρημένοι γράφοντας οποιοδήποτε στοιχείο γνωρίζει για αυτόν . Ακόμα να κάνει επεξεργασία σε όλα τα πεδία των στοιχείων των πελατών που είναι καταχωρημένοι και να διαγράφει τους καταχωρημένους πελάτες.

Όσο αφορά τα προϊόντα-συσκευές της εταιρίας, αυτά θα πρέπει να καταχωρούνται, να επεξεργάζονται και να διαγράφονται τα στοιχεία τους.

Η εφαρμογή θα δίνει τη δυνατότητα στο χρήστη να καταχωρεί πλήθος συσκευών που έχουν αγοραστεί από κάθε πελάτη με συγκεκριμένα χαρακτηριστικά για κάθε μια από αυτές, καθώς επίσης να επεξεργάζεται όλα τα πεδία κάθε αγορασμένης συσκευής και να διαγράφει την αγορασμένη συσκευή από τη λίστα.

Εξίσου σημαντικό είναι το γεγονός ότι με αυτό το πρόγραμμα θα δίνεται η δυνατότητα να καταγράφονται οι εργασίες συντήρησης (service) για κάθε αγορασμένη συσκευή και να γίνεται επεξεργασία σε όλα τα πεδία εργασίας συντήρησης που έχουν καταχωρηθεί.

Τέλος, αλλά με την μεγαλύτερη σπουδαιότητα, θεωρείται το γεγονός ότι αυτή η εφαρμογή θα έχει σαν αποτέλεσμα την προβολή στην οθόνη και εκτύπωση λίστας με τις συσκευές των πελατών που πρέπει σύντομα να συντηρηθούν. Η λίστα αυτή που θα δημιουργείται βάση νομού θα παρέχει τις πληροφορίες επικοινωνίας με τον πελάτη, για τα φίλτρα και τα αναλώσιμα υλικά που θα χρειαστούν και την τοποθεσία (διεύθυνση) της συσκευής καθαρισμού του νερού. Για κάθε μία από τις καταχωρίσεις αυτές θα υπάρχει η δυνατότητα προβολής χάρτη με βάση τις συντεταγμένες της τοποθεσίας το πελάτη.



## 5.2 Απαιτήσεις

Οι απαιτήσεις είναι ένα πλήθος με τις υπηρεσίες που απαιτεί ο χρήστης από το σύστημα που θα κατασκευαστεί έτσι ώστε να καλύψει τις συγκεκριμένες ανάγκες του. Αφού αναλυθούν και καταγραφούν αναλυτικά οι απαιτήσεις, το σύστημα που θα κατασκευαστεί θα πρέπει κατά το δυνατόν να μην αποκλίνει από αυτές. Οι απαιτήσεις που ακολουθούν είναι από τα πρώτα βήματα για την δημιουργία της εφαρμογής.

<b>Κωδικός Απαίτησης: A1</b>
<b>Τίτλος Απαίτησης: Καταχώρηση Πελατών</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα έχει την δυνατότητα να καταχωρεί νέους πελάτες. Τα στοιχεία του πελάτη που απαιτούνται είναι τα παρακάτω: Κωδικός, Όνομα, Εταιρία, Τηλέφωνο σπιτιού, Τηλέφωνο εργασίας, FAX, Τηλέφωνο κινητό, E-mail, Νομός, Πόλη, Περιοχή, Διεύθυνση, Ταχ.Κώδ, Γεωγραφικό συντεταγμένες και σχόλια.  Κωδικό με τον οποίο ο χρήστης θα μπορεί να βρίσκει την αντιστοιχία του πελάτη στο ήδη υπάρχων πρόγραμμα εμπορικής διαχείρισης της εταιρίας. Γεωγραφικό μήκος και πλάτος με τα οποία ο χρήστης θα έχει τη δυνατότητα να βρίσκει την τοποθεσία του πελάτη στο Google maps ή για να χρησιμοποιηθούν σε συσκευή GPS. Σημειώσεις με τις οποίες θα μπορεί ο χρήστης να βρίσκει κάποιες περεταίρω πληροφορίες για τον πελάτη.

<b>Κωδικός Απαίτησης: A2</b>
<b>Τίτλος Απαίτησης: Αναζήτηση Πελατών</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα μπορεί να αναζητά πελάτες γράφοντας οτιδήποτε στοιχείο γνωρίζει είτε είναι κάποιο όνομα είτε είναι κάποιος αριθμός η κάποια αρχικά της λέξης. Η αναζήτηση θα λειτουργεί άσχετα αν γράψει με κεφαλαία ή μικρά γράμματα και θα έχει τη δυνατότητα να αναζητά όχι μόνο γράφοντας το όνομα και το τηλέφωνο αλλά και γράφοντας οποιοδήποτε προαναφερθέν στοιχείο.

<b>Κωδικός Απαίτησης: A3</b>
<b>Τίτλος Απαίτησης: Επεξεργασία Στοιχείων Πελατών</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα μπορεί να επεξεργάζεται όλα τα προαναφερθέντα στοιχεία του καταχωρημένου πελάτη όταν θελήσει να προσθέσει να αφαιρέσει ή να αλλάξει κάτι από αυτά.

<b>Κωδικός Απαίτησης: A4</b>
<b>Τίτλος Απαίτησης: Διαγραφή Πελατών</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποιο επιλεγμένο πελάτη με κάποιο ασφαλής τρόπο έτσι ώστε να αποφευχθεί η διαγραφή δεδομένων από λάθος. Το πρόγραμμα θα επιτρέπει να διαγραφεί πελάτης σε περίπτωση που έχει αγοράσει κάποια συσκευή, αλλά θα διαγράφει και τα στοιχεία που είναι συνδεδεμένα με αυτόν. Θα διαγράφει δηλαδή και τη λίστα με τα αγορασμένα προϊόντα αλλά και τις εργασίες συντήρησης αυτών</p>

<b>Κωδικός Απαίτησης: A5</b>
<b>Τίτλος Απαίτησης: Καταχώρηση Συσκευών της Εταιρίας</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει τη δυνατότητα να καταχωρεί νέες συσκευές. Τα στοιχεία της συσκευής που απαιτούνται είναι τα παρακάτω: Κωδικό με τον οποίο ο χρήστης θα μπορεί να βρίσκει την αντιστοιχία της συσκευής στο ήδη υπάρχων πρόγραμμα εμπορικής διαχείρισης της εταιρίας, περιγραφή με την οποία θα γίνεται περιγραφή του μοντέλου της συσκευής και πληροφορίες στις οποίες θα γράφονται διάφορες σημειώσεις για την συσκευή.</p>

<b>Κωδικός Απαίτησης: A6</b>
<b>Τίτλος Απαίτησης: Επεξεργασία στοιχείων συσκευής</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα μπορεί να επεξεργάζεται όλα τα προαναφερθέντα στοιχεία της καταχωρημένης συσκευής όταν θελήσει να προσθέσει να αφαιρέσει ή να αλλάξει κάτι από αυτά.</p>

<b>Κωδικός Απαίτησης: A7</b>
<b>Τίτλος Απαίτησης: Διαγραφή Συσκευών</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποια επιλεγμένη συσκευή με κάποιο ασφαλής τρόπο έτσι ώστε να αποφευχθεί η διαγραφή δεδομένων από λάθος. Δεν θα επιτρέπει το πρόγραμμα να διαγραφεί η συσκευή αν αυτή έχει αγοραστεί από κάποιο πελάτη.</p>

<b>Κωδικός Απαίτησης: A8</b>
<b>Τίτλος Απαίτησης: Συσχετισμός Συσκευών με Πελάτες</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα μπορεί να επιλέγει μια συσκευή από τις διαθέσιμες συσκευές και να καταχωρεί τα στοιχεία παρακάτω: Ημερομηνία αγοράς της συσκευής από τον πελάτη, τοποθεσία, στην οποία ο χρήστης θα μπορεί να βλέπει σε ποιο σημείο του χώρου του πελάτη έχει τοποθετηθεί η συσκευή, φίλτρα, μήνες μέχρι την συντήρηση, ανά πόσους μήνες δηλαδή θα γίνεται η συντήρηση της συσκευής, πληροφορίες της συσκευής του πελάτη που θα θέλει να δει ο χρήστης και σημειώσεις.</p>

<b>Κωδικός Απαίτησης: A9</b>
<b>Τίτλος Απαίτησης: Επεξεργασία Συσκευών του Πελάτη</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα μπορεί να επεξεργάζεται τα προαναφερθέντα στοιχεία της συσκευής που έχει αγοράσει ο πελάτης όταν θελήσει να προσθέσει να αφαιρέσει ή να αλλάξει κάτι από αυτά.</p>

<b>Κωδικός Απαίτησης: A10</b>
<b>Τίτλος Απαίτησης: Προβολή Συσκευών του Πελάτη</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα μπορεί να δει λίστα με τις συσκευές που έχουν αγοράσει οι πελάτες με τα στοιχεία που έχουν καταχωρηθεί για την κάθε μία από αυτές. Επίσης θα μπορεί να δει και το ιστορικό συντηρήσεων και επισκευών για κάθε μία από αυτές.</p>

<b>Κωδικός Απαίτησης: A11</b>
<b>Τίτλος Απαίτησης: Διαγραφή Συσκευών του Πελάτη</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποια συσκευή που έχει αγοράσει ο πελάτης με κάποιο ασφαλές τρόπο έτσι ώστε να αποφευχθεί η διαγραφή δεδομένων από λάθος. Επίσης κατά την διαγραφή της συσκευής θα πρέπει να διαγράφετε αυτόματα και το ιστορικό συντήρησης της.</p>

<b>Κωδικός Απαίτησης: A12</b>
<b>Τίτλος Απαίτησης: Καταχώρηση Εργασιών Συντήρησης</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει την δυνατότητα να καταχωρεί τις εργασίες συντήρησης. Τα στοιχεία των εργασιών συντήρησης που απαιτούνται είναι τα παρακάτω: η ημερομηνία, η ώρα, το κόστος, τα υλικά-ανταλλακτικά που χρησιμοποιήθηκαν για την εργασία συντήρησης, θα καταγράφει τυχόν πληροφορίες για την εργασία συντήρησης και θα καταγράφει αν είναι εξοφλημένη η εργασία συντήρησης.</p>

<b>Κωδικός Απαίτησης: A13</b>
<b>Τίτλος Απαίτησης: Επεξεργασία Εργασιών Συντήρησης</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Το πρόγραμμα θα πρέπει να δίνει τη δυνατότητα στο χρήστη να επεξεργάζεται τα προαναφερθέντα στοιχεία των εργασιών συντήρησης της συσκευής που έχει αγοράσει ο πελάτης όταν θελήσει να προσθέσει να αφαιρέσει ή να αλλάξει κάτι από αυτά</p>

<b>Κωδικός Απαίτησης: A14</b>
<b>Τίτλος Απαίτησης: Διαγραφή Εργασιών Συντήρησης</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποια επιλεγμένη εργασία συντήρησης σε συσκευή που έχει αγοράσει ο πελάτης με κάποιο ασφαλές τρόπο έτσι ώστε να αποφευχθεί η διαγραφή δεδομένων από λάθος.</p>

<b>Κωδικός Απαίτησης: A15</b>
<b>Τίτλος Απαίτησης: Καταχώρηση Εργασιών Επισκευής</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Ο χρήστης θα έχει την δυνατότητα να καταχωρεί τις εργασίες επισκευής. Τα στοιχεία των εργασιών επισκευής που απαιτούνται είναι τα παρακάτω: η ημερομηνία, η ώρα, το κόστος</p> <p>Τα υλικά-ανταλλακτικά που χρησιμοποιήθηκαν για την εργασία επισκευής, θα καταγράφει τυχόν πληροφορίες για την εργασία επισκευής και θα καταγράφει αν είναι εξοφλημένη η εργασία επισκευής.</p>

<b>Κωδικός Απαίτησης: A16</b>
<b>Τίτλος Απαίτησης: Επεξεργασία Εργασιών Επισκευής</b>
<p><b>Περιγραφή Απαίτησης</b></p> <p>Το πρόγραμμα θα πρέπει να δίνει τη δυνατότητα στο χρήστη να επεξεργάζεται τα προαναφερθέντα στοιχεία των εργασιών επισκευής της συσκευής που έχει αγοράσει ο πελάτης όταν θελήσει να προσθέσει να αφαιρέσει ή να αλλάξει κάτι από αυτά</p>

<b>Κωδικός Απαίτησης: A17</b>
<b>Τίτλος Απαίτησης: Διαγραφή Εργασιών Επισκευής</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα έχει τη δυνατότητα να διαγράψει κάποια επιλεγμένη εργασία επισκευής σε συσκευή που έχει αγοράσει ο πελάτης με κάποιο ασφαλές τρόπο έτσι ώστε να αποφευχθεί η διαγραφή δεδομένων από λάθος.

<b>Κωδικός Απαίτησης: A18</b>
<b>Τίτλος Απαίτησης: Λίστα Επόμενων Συντηρήσεων</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα έχει τη δυνατότητα να βλέπει τις επόμενες εργασίες συντήρησης που θα πρέπει να γίνουν στις συσκευές των πελατών σε μορφή λίστας – πίνακα. Η λίστα θα δημιουργείται βάση νομού (και με δυνατότητα απεικόνισης πελατών από όλους τους νομούς) και της ημερομηνίας που θα πρέπει να γίνουν οι συντηρήσεις. Θα αναγράφονται οι παρακάτω πληροφορίες: Ονοματεπώνυμο, το κόστος, τηλέφωνο εργασίας, νομός, μοντέλο συσκευής, τα φίλτρα, τα ανταλλακτικά, τηλέφωνο σπιτιού.  Το σημείο στο χώρο του πελάτη που είναι τοποθετημένη η συσκευή Διάφορες πληροφορίες που αφορούν τις εργασίες συντήρησης Θα αναγράφεται επίσης αν είναι εξοφλημένο το ποσό της τελευταίας συντήρησης. Την ημερομηνία αγοράς της συσκευής του πελάτη. Την ημερομηνία της τελευταίας συντήρησης που έγινε στη συσκευή του πελάτη. Την ημερομηνία που θα πρέπει να γίνει η επόμενη συντήρηση.

<b>Κωδικός Απαίτησης: A19</b>
<b>Τίτλος Απαίτησης: Εκτυπώσιμη Λίστα Συντηρήσεων</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα έχει τη δυνατότητα να βλέπει και να εκτυπώνει τις επόμενες εργασίες συντήρησης που θα πρέπει να γίνουν στις συσκευές των πελατών σε μορφή λίστας η οποία θα δημιουργείται βάση νομού (και με δυνατότητα απεικόνισης πελατών από όλους τους νομούς). Ο χρήστης θα έχει τη δυνατότητα να αλλάζει την σειρά των στηλών, να τις αφαιρεί από τον πίνακα και να τις επανεμφανίζει. Θα αναγράφονται οι παρακάτω πληροφορίες: Ονοματεπώνυμο, το κόστος, τηλέφωνο εργασίας, νομός, μοντέλο συσκευής, τα φίλτρα, τα ανταλλακτικά, τηλέφωνο σπιτιού.  Το σημείο στο χώρο του πελάτη που είναι τοποθετημένη η συσκευή Διάφορες πληροφορίες που αφορούν τις εργασίες συντήρησης Θα αναγράφεται επίσης αν είναι εξοφλημένο το ποσό της τελευταίας συντήρησης. Την ημερομηνία αγοράς της συσκευής του πελάτη. Την ημερομηνία της τελευταίας συντήρησης που έγινε στη συσκευή του πελάτη. Την ημερομηνία που θα πρέπει να γίνει η επόμενη συντήρηση.

<b>Κωδικός Απαίτησης: A20</b>
<b>Τίτλος Απαίτησης: Προβολή Χάρτη</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα μπορεί επιλέγοντας ένα πελάτη από τη λίστα των εργασιών συντήρησης των συσκευών να βλέπει την ακριβή τοποθεσία του πελάτη στο Google maps. Να μπορεί να εκτυπωθεί ο χάρτης και να έχει ο τεχνικός τις ακριβείς συντεταγμένες του πελάτη για την διευκόλυνση του. Μαζί με το χάρτη να προβάλλεται το γεωγραφικό μήκος και πλάτος της τοποθεσίας του πελάτη για να μπορεί να χρησιμοποιηθεί σε εξωτερική συσκευή πλοήγησης (GPS).

<b>Κωδικός Απαίτησης: A21</b>
<b>Τίτλος Απαίτησης: Backup Βάσης Δεδομένων</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα μπορεί με αυτή την επιλογή να κρατάει αντίγραφο ασφαλείας της βάσης δεδομένων της εφαρμογής. Το αρχείο που θα δημιουργεί, θα έχει σαν όνομα την ημερομηνία και ώρα της δημιουργίας του

<b>Κωδικός Απαίτησης: A22</b>
<b>Τίτλος Απαίτησης: Επαναφορά Βάσης Δεδομένων</b>
<b>Περιγραφή Απαίτησης</b> Ο χρήστης θα μπορεί με αυτή την επιλογή να κάνει επαναφορά της βάσης δεδομένων της εφαρμογής. Θα πρέπει η διαδικασία επαναφοράς να γίνεται με τέτοιο τρόπο ώστε να αποφευχθούν απόλυτες δεδομένων που υπάρχουν στην βάση.

<b>Κωδικός Απαίτησης: A23</b>
<b>Τίτλος Απαίτησης: Πολλαπλή Προβολή</b>
<b>Περιγραφή Απαίτησης</b> Το πρόγραμμα θα πρέπει να δίνει τη δυνατότητα στο χρήστη βλέπει ταυτόχρονα πληροφορίες που αφορούν τον πελάτη, τις συσκευές του πελάτη και τις συντηρήσεις και επισκευές για κάθε μία από αυτές.

### 5.3 Περιπτώσεις Χρήσης και Σενάρια

Οι περιπτώσεις χρήσεις είναι οι λειτουργίες που πρέπει να εκτελεί το λογισμικό για την ικανοποίηση των απαιτήσεων. Τα σενάρια είναι περιγραφές με τις πιθανές λειτουργίες που μπορεί να προσφέρει το σύστημα στην πράξη.

Περίπτωση Χρήσης	Σενάρια
Διαχείριση Πελατών (A1,A2,A3,A4)	Καταχώριση Πελάτη. Αναζήτηση Πελάτη. Μεταβολή στοιχείων Πελάτη. Διαγραφή Πελάτη.
Διαχείριση Προϊόντων (A5,A6,A7)	Καταχώριση προϊόντων. Αναζήτηση προϊόντων. Μεταβολή στοιχείων προϊόντων. Διαγραφή προϊόντων.
Διαχείριση Προϊόντων Πελατών (A8,A9,A10,A11)	Καταχώριση Προϊόντων Πελατών. Προβολή Προϊόντων Πελατών. Μεταβολή στοιχείων Προϊόντων Πελατών. Διαγραφή Προϊόντων Πελατών.
Διαχείριση Εργασιών Συντήρησης (A12,A13,A14)	Καταχώριση Εργασιών Συντήρησης. Προβολή Εργασιών Συντήρησης. Μεταβολή Εργασιών Συντήρησης. Διαγραφή Εργασιών Συντήρησης.
Διαχείριση Εργασιών Επισκευής (A15,A16,A17)	Καταχώριση Εργασιών Επισκευής. Προβολή Εργασιών Επισκευής. Μεταβολή Εργασιών Επισκευής. Διαγραφή Εργασιών Επισκευής.
Προβολή λίστας Επόμενων Εργασιών Συντήρησης (A18)	Προβολή Επόμενων Εργασιών Συντήρησης από όλους τους νομούς. Προβολή Επόμενων Εργασιών Συντήρησης από το νομό Ηρακλείου. Προβολή Επόμενων Εργασιών Συντήρησης από το νομό Λασιθίου. Προβολή Επόμενων Εργασιών Συντήρησης από το νομό Ρεθύμνου. Προβολή Επόμενων Εργασιών Συντήρησης από το νομό Χανίων.
Εκτύπωση λίστας Επόμενων Εργασιών Συντήρησης (A19)	Εκτύπωση Επόμενων Εργασιών Συντήρησης από όλους τους νομούς. Εκτύπωση Επόμενων Εργασιών Συντήρησης από το νομό Ηρακλείου. Εκτύπωση Επόμενων Εργασιών Συντήρησης από το νομό Λασιθίου. Εκτύπωση Επόμενων Εργασιών Συντήρησης από το νομό Ρεθύμνου. Εκτύπωση Επόμενων Εργασιών Συντήρησης από το νομό Χανίων.
Προβολή χάρτη (A20)	Προβολή Χάρτη μέσω Google Maps. Προβολή γεωγραφικών συντεταγμένων.
Ασφάλεια λειτουργίας (A21,A22)	Δημιουργία Αντιγράφων της Βάσης. Επαναφορά δεδομένων στη βάση.
Πολλαπλή προβολή πληροφοριών (A23)	Προβολή στοιχείων πελάτη. Προβολή στοιχείων πελάτη και όλων των συσκευών του. Προβολή στοιχείων πελάτη, όλων των συσκευών του και όλες τις συντηρήσεις για μια από τις συσκευές.

Πίνακας 5.3.α: Περιπτώσεις Χρήσης και Σενάρια

## 5.4 Σχεδίαση Βάσης Δεδομένων

Με βάση την ανάλυση που έγινε μέχρι τώρα, προκύπτουν και οι απαραίτητες πληροφορίες ώστε να ορίσουμε τις οντότητες που υπάρχουν στο σύστημα μας.

Μια οντότητα είναι ο πελάτης που έχει τα χαρακτηριστικά: Κωδικός, Όνομα, Εταιρία, Τηλέφωνο σπιτιού, Τηλέφωνο εργασίας, FAX, Τηλέφωνο κινητό, E-mail, Νομός, Πόλη, Περιοχή, Διεύθυνση, Ταχ.Κώδ, Γεωγραφικό συντεταγμένες και σχόλια. Θα έχει επίσης και ένα μοναδικό χαρακτηριστικό αριθμό ώστε να είναι δυνατός ο διαχωρισμός των πλειάδων.

Τα προϊόντα της επιχείρησης είναι άλλη μία οντότητα με χαρακτηριστικά: περιγραφή, πληροφορίες, σημειώσεις. Θα έχει επίσης και ένα μοναδικό χαρακτηριστικό αριθμό ώστε να είναι δυνατός ο διαχωρισμός των πλειάδων.

Τα αγορασμένα προϊόντα που διαμορφώνονται σύμφωνα με τις ανάγκες του πελάτη που τα αγοράζει είναι μία ακόμα οντότητα με χαρακτηριστικά: χώρος τοποθέτησης, μοντέλο συσκευής, φίλτρα, μήνες μέχρι την συντήρηση, πληροφορίες και σημειώσεις. . Θα έχει επίσης και ένα μοναδικό χαρακτηριστικό αριθμό ώστε να είναι δυνατός ο διαχωρισμός των πλειάδων.

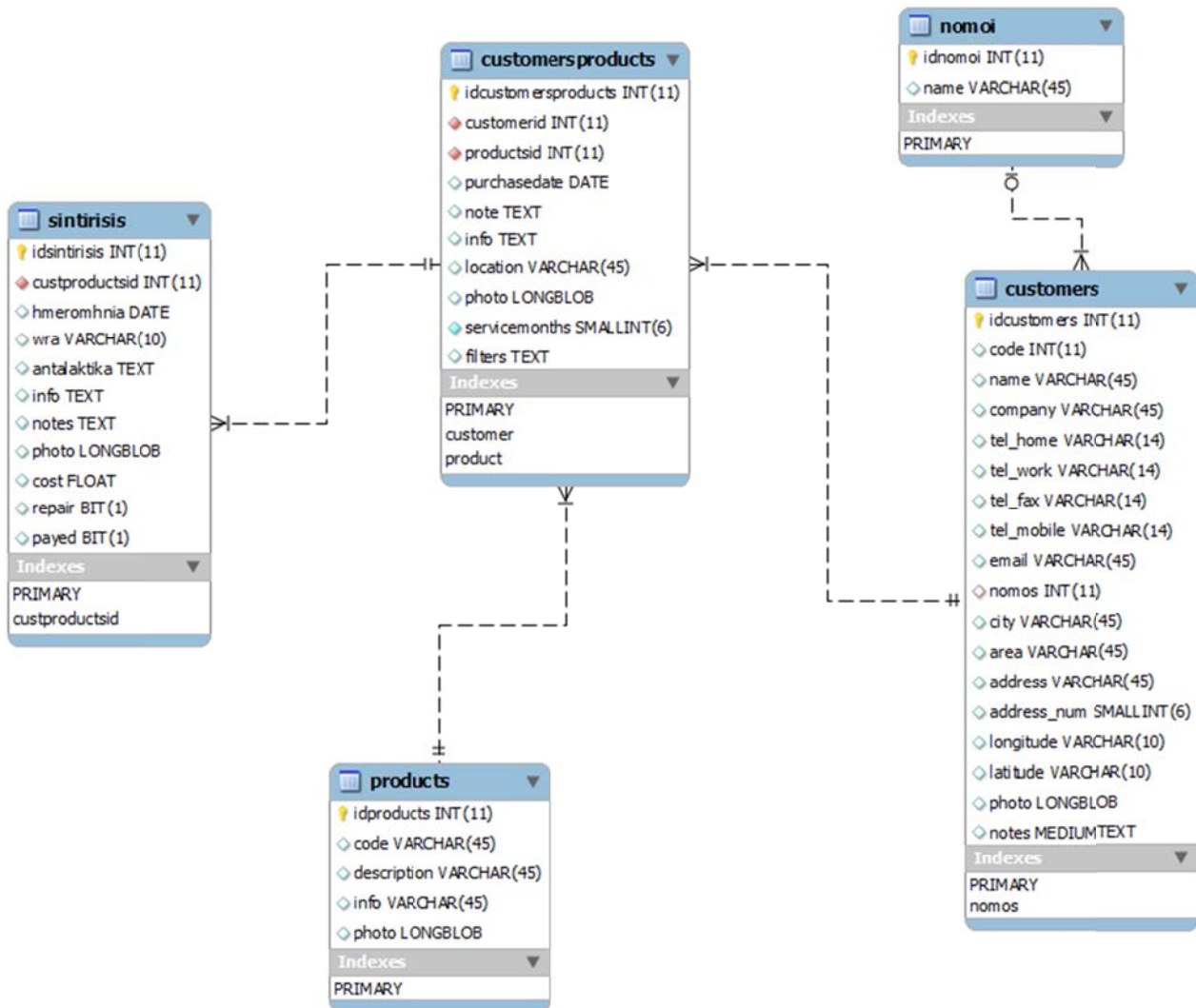
Τέλος, οι εργασίες συντήρησης είναι μια οντότητα που θα έχει τα χαρακτηριστικά: ημερομηνία, ώρα, κόστος, υλικά-ανταλλακτικά που χρησιμοποιήθηκαν, πληροφορίες, εξόφληση ή όχι των εργασιών. Θα έχει επίσης και ένα μοναδικό χαρακτηριστικό αριθμό ώστε να είναι δυνατός ο διαχωρισμός των πλειάδων.

Οι οντότητες αυτές έχουν μεταξύ τους κάποιες συσχετίσεις που επίσης προκύπτουν από την ανάλυση στα προηγούμενα κεφάλαια. Η οντότητα πελάτης έχει συσχέτιση «έχει αγοράσει» από την οντότητα με τα αγορασμένα και διαμορφωμένα προϊόντα, η οποία οντότητα έχει συσχέτιση «εμπεριέχει» την οντότητα προϊόντα της επιχείρησης. Επίσης για την οντότητα αγορασμένα και διαμορφωμένα προϊόντα υπάρχει η συσχέτιση «διενεργείται» με την οντότητα εργασία συντήρησης.

Στο επόμενο κεφάλαιο χρησιμοποιώντας τις οντότητες αυτές με τις συσχετίσεις τους θα προσδιοριστεί το διάγραμμα της βάσης δεδομένων και οι πίνακες αναλυτικά.



## 5.4.1 Διάγραμμα Βάσης Δεδομένων



Εικόνα 5. α : Διάγραμμα Βάσης Δεδομένων της εφαρμογής

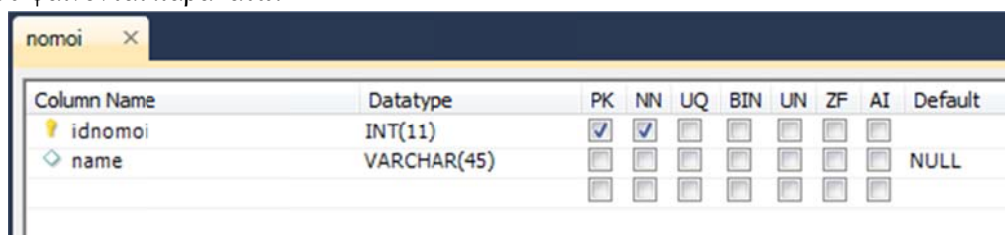
	Primary Key
	Foreign Key (NOT NULL)
	Foreign Key (DEFAULT NULL)
	Column (NOT NULL)
	Column (DEFAULT NUL)

## 5.4.2 Δημιουργία Βάσης Δεδομένων

Παρακάτω ακολουθούν οι πίνακες και ο κώδικας που χρειάστηκε για την δημιουργία της βάσης και των πινάκων στην MySQL, αλλά και εισαγωγή δοκιμαστικών δεδομένων σε αυτούς. Τα screen shoots είναι από το πρόγραμμα MySQL Workbench 5.2 CE.

- **Πίνακας νομοί.**

Στον πίνακα αυτό καταχωρούνται οι νομοί με τα ονόματα τους. Ο πίνακας έχει την μορφή και τα πεδία που φαίνονται παρακάτω:



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idnomoi	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Εικόνα 5. β: Πίνακας νομοί

**Ο κώδικας για την δημιουργία του πίνακα είναι:**

```
DROP TABLE IF EXISTS `klf`.`nomoi`;  
CREATE TABLE IF NOT EXISTS `klf`.`nomoi` (  
  `idnomoi` INT(11) NOT NULL,  
  `name` VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (`idnomoi`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

**Ο πίνακας με δοκιμαστικά δεδομένα:**

	idnomoi	name
▶	1	Οποιοδήποτε
	2	Ηρακλείου
	3	Χανίων
	4	Ρεθύμνου
	5	Λασιθίου
*	NULL	NULL

Εικόνα 5. γ :Περιεχόμενα πίνακα νομοί

**Για την εισαγωγή δοκιμαστικών δεδομένων στον πίνακα εκτελούμε τον παρακάτω κώδικα:**

```
LOCK TABLES `nomoi` WRITE;  
/*!40000 ALTER TABLE `nomoi` DISABLE KEYS */;  
INSERT INTO `nomoi` VALUES (1,'Οποιοδήποτε'),(2,'Ηρακλείου'),(3,'Χανίων'),(4,'Ρεθύμνου'),(5,'Λασιθίου');  
/*!40000 ALTER TABLE `nomoi` ENABLE KEYS */;  
UNLOCK TABLES;
```

- Πίνακας customers

Στον πίνακα αυτό καταχωρούνται οι πελάτες με τα στοιχεία τους. Ο πίνακας έχει την μορφή και τα πεδία που φαίνονται παρακάτω:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idcustomers	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
code	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
company	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tel_home	VARCHAR(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tel_work	VARCHAR(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tel_fax	VARCHAR(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
tel_mobile	VARCHAR(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
email	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nomos	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
city	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
area	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
address	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
address_num	SMALLINT(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
longitude	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
latitude	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
photo	LONGBLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
notes	MEDIUMTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Εικόνα 5. δ : Πίνακας customers

Ο κώδικας για την δημιουργία του πίνακα είναι:

```
DROP TABLE IF EXISTS `klf`.`customers` ;
CREATE TABLE IF NOT EXISTS `klf`.`customers` (
  `idcustomers` INT(11) NOT NULL AUTO_INCREMENT ,
  `code` INT(11) NULL DEFAULT NULL ,
  `name` VARCHAR(45) NULL DEFAULT NULL ,
  `company` VARCHAR(45) NULL DEFAULT NULL ,
  `tel_home` VARCHAR(14) NULL DEFAULT NULL ,
  `tel_work` VARCHAR(14) NULL DEFAULT NULL ,
  `tel_fax` VARCHAR(14) NULL DEFAULT NULL ,
  `tel_mobile` VARCHAR(14) NULL DEFAULT NULL ,
  `email` VARCHAR(45) NULL DEFAULT NULL ,
  `nomos` INT(11) NULL DEFAULT NULL ,
  `city` VARCHAR(45) NULL DEFAULT NULL ,
  `area` VARCHAR(45) NULL DEFAULT NULL ,
  `address` VARCHAR(45) NULL DEFAULT NULL ,
  `address_num` SMALLINT(6) NULL DEFAULT NULL ,
  `longitude` VARCHAR(10) NULL DEFAULT NULL ,
  `latitude` VARCHAR(10) NULL DEFAULT NULL ,
  `photo` LONGBLOB NULL DEFAULT NULL ,
  `notes` MEDIUMTEXT NULL DEFAULT NULL ,
  PRIMARY KEY (`idcustomers`),
  CONSTRAINT `nomos`
  FOREIGN KEY (`nomos`)
  REFERENCES `klf`.`nomoi` (`idnomoi`)
  ON DELETE NO ACTION
```

```

ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 21
DEFAULT CHARACTER SET = utf8;
CREATE INDEX `nomos` ON `klf`.`customers` (`nomos` ASC);

```

### Ο πίνακας με δοκιμαστικά δεδομένα:

idcustomers	code	name	company	tel_home	tel_work	tel_fax	tel_mobile	email	nomos	city	area	address	address_num	longitude	latitude	photo	notes
13	123	Όνοματεπώνυμο1	Εταιρία1	1111111111	1111111111	1111111111	1111111111	e-mail@number1.gr	2	Πόλη1	Περιοχή1	Διεύθυνση 1	1	25.102955	35.327319	NULL	σχόλια1
14	222	Όνοματεπώνυμο2	Εταιρία2	2222222222	2222222222	2222222222	2222222222	e-mail@number2.gr	5	Ηράκλειο	Περιοχή2	Διεύθυνση2	2	25.102955	35.327319	NULL	22222
15	333	Όνοματεπώνυμο3	Εταιρία3	3333333333	3333333333	3333333333	3333333333	e-mail@number3.gr	5	Πόλη3	Περιοχή3	Διεύθυνση3	3	25.102955	35.327319	NULL	33333
16	444	Όνοματεπώνυμο4	Εταιρία4	4444444444	4444444444	4444444444	4444444444	e-mail@number4.gr	3	Πόλη4	Περιοχή4	Διεύθυνση4	4	25.102955	35.327319	NULL	444444
17	555	Όνοματεπώνυμο5	Εταιρία5	5555555555	5555555555	5555555555	5555555555	e-mail@number5.gr	4	Πόλη5	Περιοχή5	Διεύθυνση5	5	25.102955	35.327319	NULL	55555
19	666	Όνοματεπώνυμο6	Εταιρία6	6666666666	6666666666	6666666666	6666666666	e-mail@number8.gr	2	Πόλη6	Περιοχή6	Διεύθυνση6	6	25.102955	35.327319	NULL	666666
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα 5. ε : Περιεχόμενα πίνακα customers

Για την εισαγωγή δοκιμαστικών δεδομένων στον πίνακα εκτελούμε τον παρακάτω κώδικα:

```

LOCK TABLES `customers` WRITE;
/*!40000 ALTER TABLE `customers` DISABLE KEYS */;
INSERT INTO `customers` VALUES
(13,123,'Όνοματεπώνυμο1','Εταιρία1','1111111111','1111111111','1111111111','1111111111', 'e-mail@number1.gr',2,'Πόλη1','Περιοχή1','Διεύθυνση 1',1,'25.102955','35.327319', NULL,'σχόλια1'),
(14,222,'Όνοματεπώνυμο2','Εταιρία2','2222222222','2222222222','2222222222','2222222222', 'e-mail@number2.gr',5,'Ηράκλειο','Περιοχή2','Διεύθυνση2',2,'25.102955','35.327319', NULL,'22222'),
(15,333,'Όνοματεπώνυμο3','Εταιρία3','3333333333','3333333333','3333333333','3333333333', 'e-mail@number3.gr',5,'Πόλη3','Περιοχή3','Διεύθυνση3',3,'25.102955','35.327319', NULL,'33333'),
(16,444,'Όνοματεπώνυμο4','Εταιρία4','4444444444','4444444444','4444444444','4444444444', 'e-mail@number4.gr',3,'Πόλη4','Περιοχή4','Διεύθυνση4',4,'25.102955','35.327319', NULL,'444444'),
(17,555,'Όνοματεπώνυμο5','Εταιρία5','5555555555','5555555555','5555555555','5555555555', 'e-mail@number5.gr',4,'Πόλη5','Περιοχή5','Διεύθυνση5',5,'25.102955','35.327319', NULL,'55555'),
(19,666,'Όνοματεπώνυμο6','Εταιρία6','6666666666','6666666666','6666666666','6666666666', 'e-mail@number8.gr',2,'Πόλη6','Περιοχή6','Διεύθυνση6',6,'25.102955','35.327319', NULL,'666666');
/*!40000 ALTER TABLE `customers` ENABLE KEYS */;
UNLOCK TABLES;

```

## • Πίνακας products

Στον πίνακα αυτό καταχωρούνται τα προϊόντα με τα στοιχεία τους. Ο πίνακας έχει την μορφή και τα πεδία που φαίνονται παρακάτω:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idproducts	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
code	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
description	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
info	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
photo	LONGBLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Εικόνα 5. ζ: Πίνακας products

Ο κώδικας για την δημιουργία του πίνακα είναι:

```
DROP TABLE IF EXISTS `klf`.`products`;
CREATE TABLE IF NOT EXISTS `klf`.`products` (
  `idproducts` INT(11) NOT NULL AUTO_INCREMENT,
  `code` VARCHAR(45) NULL DEFAULT NULL,
  `description` VARCHAR(45) NULL DEFAULT NULL,
  `info` VARCHAR(45) NULL DEFAULT NULL,
  `photo` LONGBLOB NULL DEFAULT NULL,
  PRIMARY KEY (`idproducts`))
ENGINE = InnoDB
AUTO_INCREMENT = 19
DEFAULT CHARACTER SET = utf8;
```

Ο πίνακας με δοκιμαστικά δεδομένα:

	idproducts	code	description	info	photo
▶	1	34534	WLHC750	πληροφορίες μοντέλου	NULL
	2	45667	WLHC350	πληροφορίες μοντέλου	NULL
	3	45709	WLHC850	πληροφορίες μοντέλου	NULL
	4	234	WLHC3000	πληροφορίες μοντέλου	NULL
	9	323	WL4	πληροφορίες μοντέλου	NULL
	10	42342	WLHC650	πληροφορίες μοντέλου	NULL
	12	13435	WLHC950	πληροφορίες μοντέλου	NULL
*	NULL	NULL	NULL	NULL	NULL

Εικόνα 5. η: Περιεχόμενα πίνακα products

Για την εισαγωγή δοκιμαστικών δεδομένων στον πίνακα εκτελούμε τον παρακάτω κώδικα:

```
LOCK TABLES `products` WRITE;
/*!40000 ALTER TABLE `products` DISABLE KEYS */;
```

```

INSERT INTO `products` VALUES (1,'34534','WLHC750','πληροφορίες
μοντέλου',NULL),(2,'45667','WLHC350','πληροφορίες μοντέλου',NULL),(3,'45709','WLHC850','πληροφορίες
μοντέλου',NULL),(4,'234','WLHC3000','πληροφορίες μοντέλου',NULL),(9,'323','WL4','πληροφορίες
μοντέλου',NULL),(10,'42342','WLHC650','πληροφορίες μοντέλου',NULL),(12,'13435','WLHC950','πληροφορίες
μοντέλου',NULL);
/*!40000 ALTER TABLE `products` ENABLE KEYS */;
UNLOCK TABLES;

```

## • Πίνακας customersproducts

Στον πίνακα αυτό καταχωρούνται τα προϊόντα των πελατών με τα στοιχεία τους. Ο πίνακας έχει την μορφή και τα πεδία που φαίνονται παρακάτω:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idcustomersproducts	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
customerid	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
productsid	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
purchasedate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
note	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
info	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
location	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
photo	LOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
servicemonths	SMALLINT(6)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
filters	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Εικόνα 5. 0: Πίνακας customersproducts

### Ο κώδικας για την δημιουργία του πίνακα είναι:

```

DROP TABLE IF EXISTS `klf`.`customersproducts` ;
CREATE TABLE IF NOT EXISTS `klf`.`customersproducts` (
  `idcustomersproducts` INT(11) NOT NULL AUTO_INCREMENT ,
  `customerid` INT(11) NOT NULL ,
  `productsid` INT(11) NOT NULL ,
  `purchasedate` DATE NULL DEFAULT NULL ,
  `note` TEXT NULL DEFAULT NULL ,
  `info` TEXT NULL DEFAULT NULL ,
  `location` VARCHAR(45) NULL DEFAULT NULL ,
  `photo` LONGBLOB NULL DEFAULT NULL ,
  `servicemonths` SMALLINT(6) NOT NULL ,
  `filters` TEXT NULL DEFAULT NULL ,
  PRIMARY KEY (`idcustomersproducts`),
  CONSTRAINT `customer`
  FOREIGN KEY (`customerid`)
  REFERENCES `klf`.`customers` (`idcustomers`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `product`
  FOREIGN KEY (`productsid`)
  REFERENCES `klf`.`products` (`idproducts`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 26
DEFAULT CHARACTER SET = utf8;
CREATE INDEX `customer` ON `klf`.`customersproducts` (`customerid` ASC) ;
CREATE INDEX `product` ON `klf`.`customersproducts` (`productsid` ASC) ;

```

### Ο πίνακας με δοκιμαστικά δεδομένα:

idcustomersproducts	customerid	productsid	purchasedate	note	info	location	photo	servicemonths	filters
5	13	3	2011-10-26	σημειώσεις πελάτη 1 συσκευή 1	πληροφορίες πελάτη 1 συσκευή 1	κουζίνα	NULL	6	σωματιδίων
6	14	2	2011-10-26	σημειώσεις πελάτη 2 συσκευή 2	πληροφορίες πελάτη 2 συσκευή 2	σαλόνι	NULL	4	αντίστροφη όσμωση
7	13	1	2011-10-26	σημειώσεις πελάτη 1 συσκευή 1	πληροφορίες πελάτη 1 συσκευή 1	σαλόνι	NULL	6	αντίστροφη όσμωση
9	13	3	2011-10-26	σημειώσεις πελάτη 1 συσκευή 1	πληροφορίες πελάτη 1 συσκευή 1	κουζίνα	NULL	5	άνθρακα
12	14	10	2011-10-26	σημειώσεις πελάτη 2 συσκευή 2	πληροφορίες πελάτη 2 συσκευή 2	κουζίνα	NULL	6	άνθρακα
15	16	2	2011-10-28	σημειώσεις πελάτη 4 συσκευή 4	πληροφορίες πελάτη 4 συσκευή 4	κουζίνα 2	NULL	8	σωματιδίων
16	17	4	2011-10-28	σημειώσεις πελάτη 5 συσκευή 5	πληροφορίες πελάτη 5 συσκευή 5	σαλόνι	NULL	6	αντίστροφη όσμωση
18	15	2	2011-05-04	σημειώσεις πελάτη 3 συσκευή 3	πληροφορίες πελάτη 3 συσκευή 3	Bar	NULL	6	αντίστροφη όσμωση
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα 5. ι: Περιεχόμενα πίνακα customersproducts

### Για την εισαγωγή δοκιμαστικών δεδομένων στον πίνακα εκτελούμε τον παρακάτω κώδικα:

```

LOCK TABLES `customersproducts` WRITE;
/*!40000 ALTER TABLE `customersproducts` DISABLE KEYS */;
INSERT INTO `customersproducts` VALUES
(5,13,3,'2011-10-26','σημειώσεις πελάτη 1 συσκευή 1','πληροφορίες πελάτη 1 συσκευή 1','κουζίνα',NULL,6,'σωματιδίων'),
(6,14,2,'2011-10-26','σημειώσεις πελάτη 2 συσκευή 2','πληροφορίες πελάτη 2 συσκευή 2','σαλόνι',NULL,4,'αντίστροφη
όσμωση'),
(7,13,1,'2011-10-26','σημειώσεις πελάτη 1 συσκευή 1','πληροφορίες πελάτη 1 συσκευή 1','σαλόνι',NULL,6,'αντίστροφη
όσμωση'),
(9,13,3,'2011-10-26','σημειώσεις πελάτη 1 συσκευή 1','πληροφορίες πελάτη 1 συσκευή 1','κουζίνα',NULL,5,'άνθρακα'),
(12,14,10,'2011-10-26','σημειώσεις πελάτη 2 συσκευή 2','πληροφορίες πελάτη 2 συσκευή 2','κουζίνα',NULL,6,'άνθρακα'),
(15,16,2,'2011-10-28','σημειώσεις πελάτη 4 συσκευή 4','πληροφορίες πελάτη 4 συσκευή 4','κουζίνα 2',NULL,8,'σωματιδίων'),
(16,17,4,'2011-10-28','σημειώσεις πελάτη 5 συσκευή 5','πληροφορίες πελάτη 5 συσκευή 5','σαλόνι',NULL,6,'αντίστροφη
όσμωση'),
(18,15,2,'2011-05-04','σημειώσεις πελάτη 3 συσκευή 3','πληροφορίες πελάτη 3 συσκευή 3','Bar',NULL,6,'αντίστροφη
όσμωση');
/*!40000 ALTER TABLE `customersproducts` ENABLE KEYS */;
UNLOCK TABLES;

```

- **Πίνακας sintirisis**

Στον πίνακα αυτό καταχωρούνται οι συντηρήσεις και επισκευές των συσκευών με τα στοιχεία τους. Ο πίνακας έχει την μορφή και τα πεδία που φαίνονται παρακάτω:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idsintirisis	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
custproductsid	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
hmeromhnia	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
wra	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
antalaktika	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
info	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
notes	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
photo	LONGBLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
cost	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
repair	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
payed	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Εικόνα 5. κ:Πίνακας sintirisis

**Ο κώδικας για την δημιουργία του πίνακα είναι:**

```

DROP TABLE IF EXISTS `klf`.`sintirisis` ;
CREATE TABLE IF NOT EXISTS `klf`.`sintirisis` (
  `idsintirisis` INT(11) NOT NULL AUTO_INCREMENT ,
  `custproductsid` INT(11) NOT NULL ,
  `hmeromhnia` DATE NULL DEFAULT NULL ,
  `wra` VARCHAR(10) NULL DEFAULT NULL ,
  `antalaktika` TEXT NULL DEFAULT NULL ,
  `info` TEXT NULL DEFAULT NULL ,
  `notes` TEXT NULL DEFAULT NULL ,
  `photo` LONGBLOB NULL DEFAULT NULL ,
  `cost` FLOAT NULL DEFAULT NULL ,
  `repair` BIT(1) NULL DEFAULT NULL ,
  `payed` BIT(1) NULL DEFAULT NULL ,
  PRIMARY KEY (`idsintirisis`),
  CONSTRAINT `custproductsid`
    FOREIGN KEY (`custproductsid`)
    REFERENCES `klf`.`customersproducts` (`idcustomersproducts`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 19
DEFAULT CHARACTER SET = utf8;
CREATE INDEX `custproductsid` ON `klf`.`sintirisis` (`custproductsid` ASC);

```



**Ο πίνακας με δοκιμαστικά δεδομένα:**

	idsintirisis	custproductsid	hmeromhnia	wra	antalaktika	info	notes	photo	cost	repair	payed
▶	8	5	2011-11-05	5:50	άνθρακας	πληροφορίες πελάτη 1 συντήρηση 1	σημειώσεις πελάτη 1 συντήρηση 1	NULL	NULL	0	1
	9	18	2011-11-05	NULL	λάμπα UV	πληροφορίες πελάτη 3 συντήρηση 3	σημειώσεις πελάτη 3 συντήρηση 3	NULL	NULL	0	0
	10	7	2011-11-05	NULL	άνθρακας	πληροφορίες πελάτη 1 συντήρηση 1	σημειώσεις πελάτη 1 συντήρηση 1	NULL	NULL	0	0
	11	12	2011-11-05	5:50 μμ	λάμπα UV	πληροφορίες πελάτη 2 συντήρηση 2	σημειώσεις πελάτη 2 συντήρηση 2	NULL	NULL	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Εικόνα 5. λ: Περιεχόμενα πίνακα sintirisis**

**Για την εισαγωγή δοκιμαστικών δεδομένων στον πίνακα εκτελούμε τον παρακάτω κώδικα:**

*LOCK TABLES `sintirisis` WRITE;*

*/\*!40000 ALTER TABLE `sintirisis` DISABLE KEYS \*/;*

*INSERT INTO `sintirisis` VALUES (8,5,'2011-11-05','5:50','άνθρακας','πληροφορίες πελάτη 1 συντήρηση 1','σημειώσεις πελάτη 1 συντήρηση 1',NULL,NULL,'\0',''),(9,18,'2011-11-05',NULL,'λάμπα UV','πληροφορίες πελάτη 3 συντήρηση 3','σημειώσεις πελάτη 3 συντήρηση 3',NULL,NULL,'\0','\0'),(10,7,'2011-11-05',NULL,'άνθρακας','πληροφορίες πελάτη 1 συντήρηση 1','σημειώσεις πελάτη 1 συντήρηση 1',NULL,NULL,'\0','\0'),(11,12,'2011-11-05','5:50 μμ','λάμπα UV','πληροφορίες πελάτη 2 συντήρηση 2','σημειώσεις πελάτη 2 συντήρηση 2',NULL,NULL,'\0','');*

*/\*!40000 ALTER TABLE `sintirisis` ENABLE KEYS \*/;*

*UNLOCK TABLES;*

## 5.5 Διάγραμμα Κλάσεων

Τα διαγράμματα κλάσεων χρησιμοποιούνται για να περιγράψουν τα αντικείμενα και τις σχέσεις τους σε ένα σύστημα. Μοντελοποιούν τις δομές των κλάσεων και το περιεχόμενο τους χρησιμοποιώντας σχεδιαστικά στοιχεία όπως κλάσεις, πακέτα και αντικείμενα. Επίσης εμπεριέχουν: τις κλάσεις του συστήματος, τις σχέσεις μεταξύ των κλάσεων και τις ιδιότητες & λειτουργίες των κλάσεων. Τα διαγράμματα κλάσεων περιγράφουν τρεις διαφορετικές διαστάσεις κατά τον σχεδιασμό ενός συστήματος: την εννοιολογική διάσταση (conceptual), την διάσταση τεχνικών προδιαγραφών (specification) και την διάσταση της υλοποίησης (implementation).

Παρακάτω ακολουθεί το διάγραμμα κλάσεων με τις συσχετίσεις όλων των κλάσεων και έπειτα σε επόμενες σελίδες οι σημαντικότερες κλάσεις με τα ολοκληρωμένα διαγράμματα τους λόγω δυσκολίας παρουσίασης του διαγράμματος σε εκτυπώσιμη μορφή. Τα διαγράμματα δημιουργήθηκαν στο πρόγραμμα Visual Paradigm for UML 8.3.

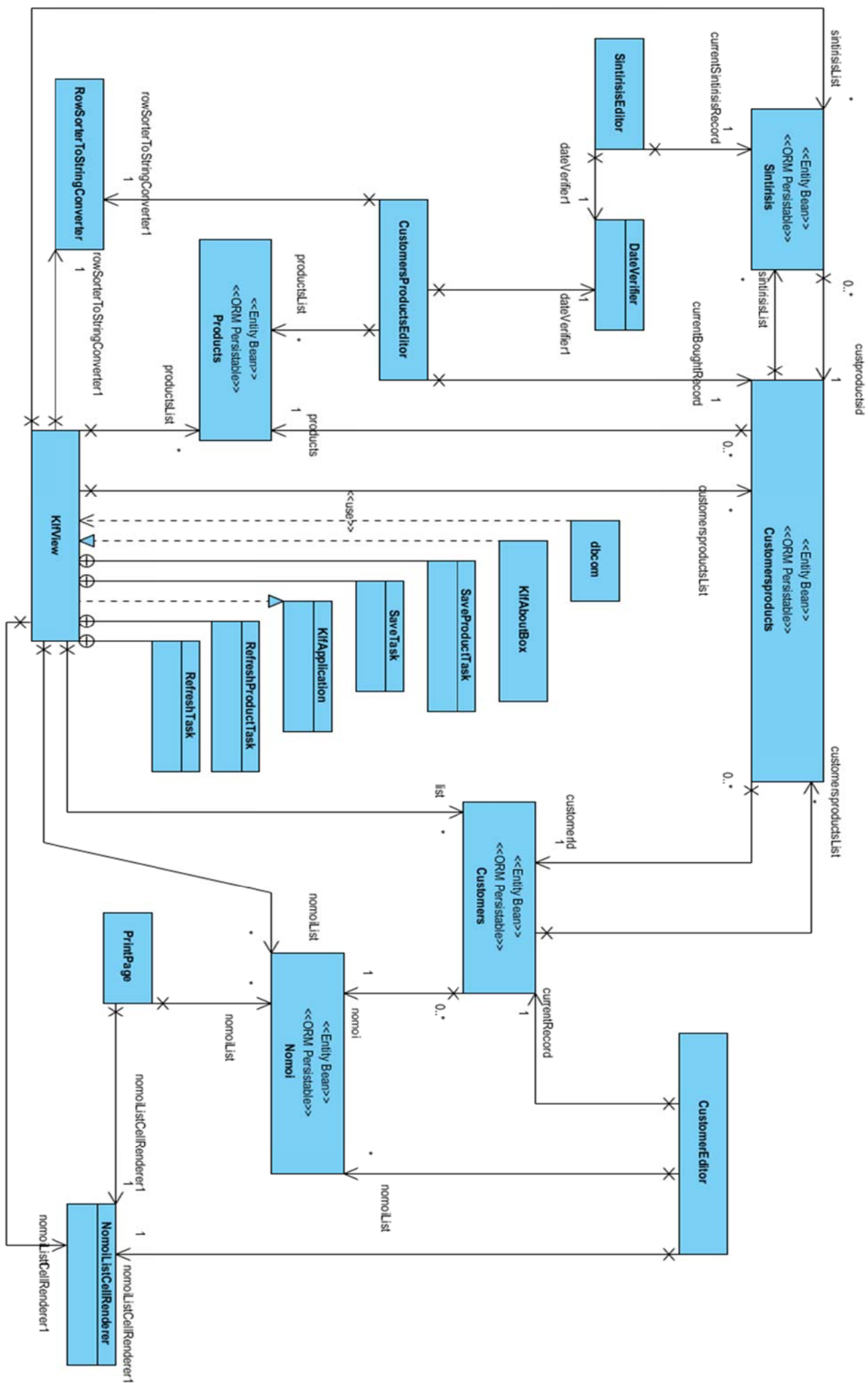
### Επεξήγηση συμβόλων διαγράμματος

ΣΥΜΒΟΛΟ	ΠΡΟΣΒΑΣΗ
+	Public
-	Private
#	Protected
~	Package

Πίνακας 5.5 α: Επεξήγηση συμβόλων πρόσβασης διαγράμματος

Πιθανές τιμές Πολλαπλότητας	
Indicator	Meaning
1	One only
0..*	Zero or more
*	One or more

Πίνακας 5.5 β: Επεξήγηση συμβόλων πολλαπλότητας διαγράμματος



Εικόνα 5. μ: Διάγραμμα Κλάσεων

```

KifView
-customersproductsQuery : Query
-entityManager : EntityManager
-jSeparator1 : JSeparator
-jSeparator3 : JSeparator
-jSeparator4 : JSeparator
-jTabbedPane1 : JTabbedPane
-mainPanel : JPanel
-menuBar : JMenuBar
-nomoiQuery : Query
-productsQuery : Query
-productsQuery1 : Query
-productsShowScrollPane : JScrollPane
-productsShowTable : JTable
-progressBar : JProgressBar
-query : Query
-servicesScrollPane : JScrollPane
-servicesTable : JTable
-sintirisQuery : Query
-sintirisScrollPane : JScrollPane
-sintirisTable : JTable
-statusAnimationLabel : JLabel
-statusMessageLabel : JLabel
-statusPanel : JPanel
-bindingGroup : BindingGroup
-messageTimer : Timer
-busyIconTimer : Timer
-idleIcon : Icon
-busyIcons : Icon[] = new Icon[15]
-busyIconIndex : int = 0
-aboutBox : JDialog
<<Property>> -saveNeeded : boolean
<<Property>> -saveProductNeeded : boolean
-customersproductsList : List<Customersproducts>
-list : List<Customers>
-nomoiList : List<Nomoi>
-nomoiListCellRenderer1 : NomoiListCellRenderer
-productsList : List<Products>
-rowSorterToStringConverter1 : RowSorterToStringConv...
-sintirisList : List<Sintiris>
-addProductButton : JButton
-addSintirisButton : JButton
-backupMenuItem : JMenuItem
-codeProductLabel : JLabel
-codeProductTextField : JTextField
-custProdLabel : JLabel
-customProdPanel : JPanel
-customerTable : JTable
-customersLabel : JLabel
-customersProdScrollPane : JScrollPane
-customersProdTable : JTable
-customersScrollPane : JScrollPane
-deleteCustProdButton : JButton
-deleteCustomerButton : JButton
-deleteProductButton : JButton
-deleteSintirisButton : JButton
-editCustProdButton : JButton
-editCustomerButton : JButton
-editSintirisButton : JButton
-infoProductLabel : JLabel
-infoProductScrollPane : JScrollPane
-infoProductTextArea : JTextArea
-manualMenuItem : JMenuItem
-mapButton : JButton
-modelProductLabel : JLabel
-modelProductTextField : JTextField
-newCustProdButton : JButton
-newCustomerButton : JButton
-nomoiComboBox : JComboBox
-nomoiLabel : JLabel
-printButton : JButton
-productsLabel : JLabel
-productsPanel : JPanel
-refreshButton : JButton
-refreshProductButton : JButton
-restoreMenuItem : JMenuItem
-saveProductButton : JButton
-searchLabel : JLabel
-searchTextField : JTextField
-servicesPanel : JPanel
-sintirisLabel : JLabel
-productsList1 : List<Products>
+KifView(app : JFrameApplication)
-setSaveNeeded(saveNeeded : boolean) : void
+isSintirisRecordSelected() : boolean
+isServicesSelected() : boolean
+newRecord() : void
+deleteRecord() : void
+newDetailRecord() : void
+deleteDetailRecord() : void
+newSintirisRecord() : void
+deleteSintirisRecord() : void
+save() : Task
-setSaveProductNeeded(saveProductNeeded : boolean) ...
+isProductRecordSelected() : boolean
+newProductRecord() : void
+deleteProductRecord() : void
+saveProduct() : Task
+refreshProduct() : Task
+refresh() : Task
+showAboutBox() : void
-initComponents() : void
+editCustomer() : void
+editBought() : void
+editSintiris() : void
+backupDB() : void
+restoreDB() : void
+showMap(source1 : String, source2 : String) : void
+showMapPressed() : void
+nomosSelected() : void
+print() : void
+isCustomerRecordSelected() : boolean
+isCustProdRecordSelected() : boolean
+refreshButtonActionPerformed(evt : ActionEvent) : void
+isSaveNeeded() : boolean
+isSaveProductNeeded() : boolean
+newSimpleSintirisRecord() : void
+autoSel(evt : MouseEvent) : void

```

```

<<Entity Bean>>
<<ORM Persistable>>
Customers
-changeSupport : PropertyChangeSupport = new PropertyChangeS...
-serialVersionUID : long = 1L
<<Property>> -idcustomers : Integer
<<Property>> -code : Integer
<<Property>> -name : String
<<Property>> -company : String
<<Property>> -telHome : String
<<Property>> -telWork : String
<<Property>> -telFax : String
<<Property>> -telMobile : String
<<Property>> -email : String
<<Property>> -city : String
<<Property>> -area : String
<<Property>> -address : String
<<Property>> -addressNum : Short
<<Property>> -longitude : String
<<Property>> -latitude : String
<<Property>> -photo : byte[]
<<Property>> -notes : String
<<Property>> -nomoi : Nomoi
<<Property>> -customersproductsList : Customersproducts
+Customers()
+Customers(idcustomers : Integer)
+hashCode() : int
+equals(object : Object) : boolean
+toString() : String
+addPropertyChangeListener(listener : PropertyChangeListener) : ...
+removePropertyChangeListener(listener : PropertyChangeListene...

```

```

<<Entity Bean>>
<<ORM Persistable>>
Products
-changeSupport : PropertyChangeSupport = new PropertyChangeS...
-serialVersionUID : long = 1L
<<Property>> -idproducts : Integer
<<Property>> -code : String
<<Property>> -description : String
<<Property>> -info : String
<<Property>> -photo : byte[]
+Products()
+Products(idproducts : Integer)
+hashCode() : int
+equals(object : Object) : boolean
+toString() : String
+addPropertyChangeListener(listener : PropertyChangeListener) : ...
+removePropertyChangeListener(listener : PropertyChangeListene...

```

```

<<Entity Bean>>
<<ORM Persistable>>
Sintiris
-changeSupport : PropertyChangeSupport = new PropertyChangeS...
-serialVersionUID : long = 1L
<<Property>> -idsintiris : Integer
<<Property>> -hmeromhnia : Date
<<Property>> -wra : String
<<Property>> -antalaktika : String
<<Property>> -info : String
<<Property>> -notes : String
<<Property>> -photo : byte[]
<<Property>> -cost : Float
<<Property>> -repair : Boolean
<<Property>> -payed : Boolean
<<Property>> -custproductsid : Customersproducts
+Sintiris()
+Sintiris(idsintiris : Integer)
+hashCode() : int
+equals(object : Object) : boolean
+toString() : String
+addPropertyChangeListener(listener : PropertyChangeListener) : ...
+removePropertyChangeListener(listener : PropertyChangeListene...

```

```

<<Entity Bean>>
<<ORM Persistable>>
Customersproducts
-changeSupport : PropertyChangeSupport = new PropertyChangeS...
-serialVersionUID : long = 1L
<<Property>> -idcustomersproducts : Integer
<<Property>> -purchasedate : Date
<<Property>> -note : String
<<Property>> -info : String
<<Property>> -location : String
<<Property>> -photo : byte[]
<<Property>> -servicemonths : short
<<Property>> -filters : String
<<Property>> -sintirisList : Sintiris
<<Property>> -products : Products
<<Property>> -customerid : Customers
+Customersproducts()
+Customersproducts(idcustomersproducts : Integer)
+Customersproducts(idcustomersproducts : Integer, products : Pro...
+hashCode() : int
+equals(object : Object) : boolean
+toString() : String
+addPropertyChangeListener(listener : PropertyChangeListener) : ...
+removePropertyChangeListener(listener : PropertyChangeListene...

```

```

PrintPage
-columnRemoveMenuItem : JMenuItem
-kifPUEntityManager : EntityManager
-nomoiQuery : Query
-printPopupMenu : JPopupMenu
-printScrollPane : JScrollPane
-printTable : JTable
-restoreColumnsMenuItem : JMenuItem
-bindingGroup : BindingGroup
-nomoiList : List<Nomoi>
-nomoiListCellRenderer1 : NomoiListCellRenderer
-cancelButton : JButton
-nomoiComboBox : JComboBox
-nomoiLabel : JLabel
-printButton : JButton
-titleLabel : JLabel
+PrintPage(parent : Frame, modal : boolean)
+isRecordSelected() : boolean
+isColumnSelected() : boolean
-initComponents() : void
-cancelPrint(evt : ActionEvent) : void
+main(args : String []) : void
+removeColumn() : void
+restoreColumns() : void
-printButtonActionPerformed(evt : ActionEvent) : void
-nomoiComboBoxActionPerformed(evt : ActionEvent) : void

```

```

<<Entity Bean>>
<<ORM Persistable>>
Nomoi
-changeSupport : PropertyChangeSupport = new PropertyChangeS...
-serialVersionUID : long = 1L
<<Property>> -idnomoi : Integer
<<Property>> -name : String
+Nomoi()
+Nomoi(idnomoi : Integer)
+hashCode() : int
+equals(object : Object) : boolean
+toString() : String
+addPropertyChangeListener(listener : PropertyChangeListener) : ...
+removePropertyChangeListener(listener : PropertyChangeListene...

```

Εικόνα 5. ν: Κλάσεις Διαγράμματος α

```

CustomerEditor
<<Property>> #customerConfirmed : boolean
-propertyChangeSupport : PropertyChangeSupport = new PropertyChange...
-TeWorkTextField : JTextField
-addNumLabel : JLabel
-addNumTextField : JTextField
-addressLabel : JLabel
-addressTextField : JTextField
-areaLabel : JLabel
-areaTextField : JTextField
-cancelCustButton : JButton
-cityLabel : JLabel
-cityTextField : JTextField
-codeLabel : JLabel
-codeTextField : JTextField
-companyLabel : JLabel
-companyTextField : JTextField
-customersEditorLabel : JLabel
-geotALabel : JLabel
-geotATextField : JTextField
-geotOLabel : JLabel
-geotOTextField : JTextField
-kifPUEntityManager : EntityManager
-mailLabel : JLabel
-mailTextField : JTextField
-nameLabel : JLabel
-nameTextField : JTextField
-nomoiQuery : Query
-nomosComboBox : JComboBox
-nomosLabel : JLabel
-notesLabel : JLabel
-notesScrollPane : JScrollPane
-notesTextArea : JTextArea
-saveCustButton : JButton
-showMapCustButton : JButton
-telFaxLabel : JLabel
-telFaxTextField : JTextField
-telHomeLabel : JLabel
-telHomeTextField : JTextField
-telMobLabel : JLabel
-telMobTextField : JTextField
-telWorkLabel : JLabel
-bindingGroup : BindingGroup
<<Property>> #currentRecord : Customers
-nomoiList : List<Nomoi>
-nomoiListCellRenderer1 : NomoiListCellRenderer
+CustomerEditor(parent : Frame, modal : boolean)
+addPropertyChangeListener(listener : PropertyChangeListener) : void
+removePropertyChangeListener(listener : PropertyChangeListener) : void
-initComponents() : void
-saveCustomer(evt : ActionEvent) : void
-cancelCustomer(evt : ActionEvent) : void
+main(args : String []): void
+showTheMap() : void
+isCustomerConfirmed() : boolean

```

```

SintirisisEditor
<<Property>> #sintirisisConfirmed : boolean
-propertyChangeSupport : PropertyChangeSupport = new PropertyChange...
-bindingGroup : BindingGroup
<<Property>> #currentSintirisisRecord : Sintirisis
-dateVerifier1 : DateVerifier
-antalaktikaLabel : JLabel
-antalaktikaScrollPane : JScrollPane
-antalaktikaTextArea : JTextArea
-cancelButton : JButton
-costLabel : JLabel
-costTextField : JTextField
-dateFormattedTextField : JFormattedTextField
-dateLabel : JLabel
-infoLabel : JLabel
-infoScrollPane : JScrollPane
-infoTextArea : JTextArea
-notesLabel : JLabel
-notesScrollPane : JScrollPane
-notesTextArea : JTextArea
-payedCheckBox : JCheckBox
-repairCheckBox : JCheckBox
-saveButton : JButton
-timeLabel : JLabel
-timeTextField : JTextField
-titleLabel : JLabel
+SintirisisEditor(parent : Frame, modal : boolean)
+addPropertyChangeListener(listener : PropertyChangeListener) : void
+removePropertyChangeListener(listener : PropertyChangeListener) : void
-initComponents() : void
-saveSintirisis(evt : ActionEvent) : void
-cancelSintirisis(evt : ActionEvent) : void
+main(args : String []): void
+isSintirisisConfirmed() : boolean

```

```

CustomersProductsEditor
<<Property>> #productConfirmed : boolean
-propertyChangeSupport : PropertyChangeSupport = new PropertyChange...
-JSeparator1 : JSeparator
-JSeparator2 : JSeparator
-kifPUEntityManager : EntityManager
-productScrollPane : JScrollPane
-productTable : JTable
-productsQuery : Query
-bindingGroup : BindingGroup
<<Property>> #currentBoughtRecord : Customersproducts
-dateVerifier1 : DateVerifier
-productsList : List<Products>
-rowSorterToStringConverter1 : RowSorterToStringConverter
-cancelButton : JButton
-dateFormattedTextField : JFormattedTextField
-dateInputLabel : JLabel
-dateLabel : JLabel
-filtersLabel : JLabel
-filtersTextField : JTextField
-helpLabel : JLabel
-infoLabel : JLabel
-infoScrollPane : JScrollPane
-infoTextArea : JTextArea
-locationLabel : JLabel
-locationTextField : JTextField
-monthsLabel : JLabel
-monthsTextField : JTextField
-notesLabel : JLabel
-notesScrollPane : JScrollPane
-notesTextArea : JTextArea
-productInputLabel : JLabel
-productInfoScrollPane : JScrollPane
-productInfoTextArea : JTextArea
-saveButton : JButton
-searchLabel : JLabel
-searchTextField : JTextField
-titleLabel : JLabel
+CustomersProductsEditor(parent : Frame, modal : boolean)
+addPropertyChangeListener(listener : PropertyChangeListener) : void
+removePropertyChangeListener(listener : PropertyChangeListener) : void
-initComponents() : void
-saveCProduct(evt : ActionEvent) : void
-cancelCProduct(evt : ActionEvent) : void
+main(args : String []): void
+isProdutConfirmed() : boolean

```

Εικόνα 5. ζ: Κλάσεις Διαγράμματος β

## 5.6 Οι Κλάσεις και οι Σημαντικότερες Μέθοδοι

Class Summary	
<b>CustomerEditor</b>	Το παράθυρο προθήκης / επεξεργασίας πελατών
<b>Customers</b>	Κλάση του πίνακα customers της βάσης
<b>Customersproducts</b>	Κλάση του πίνακα customersproducts της βάσης
<b>CustomersProductsEditor</b>	Το παράθυρο προσθήκης / επεξεργασίας συσκευών πελατών
<b>DateVerifier</b>	Κλάση για έλεγχο ημερομηνίας
<b>dbcom</b>	Κλάση για εργασίες που εκτελούνται με τη βάση δεδομένων πέρα των βασικών πινάκων
<b>KlfAboutBox</b>	Το παράθυρο περί εφαρμογής
<b>KlfApplication</b>	The main class of the application.
<b>KlfView</b>	Το κεντρικό παράθυρο της εφαρμογής
<b>Nomoi</b>	Κλάση του πίνακα nomoi της βάσης
<b>NomoiListCellRenderer</b>	Κλάση για render των combobox με το όνομα νομού
<b>PrintPage</b>	Το παράθυρο εκτύπωσης εργασιών συντήρησης
<b>Products</b>	Κλάση του πίνακα products της βάσης
<b>RowSorterToStringConverter</b>	Binding converter ανάμεσα σε String και regex RowFilter
<b>Sintirisis</b>	Κλάση του πίνακα sintirisis της βάσης
<b>SintirisisEditor</b>	Το παράθυρο προσθήκης / επεξεργασίας εργασιών συντήρησης

Πίνακας 5.6 α: Σύνοψη κλάσεων

### ▪ CustomerEditor.java

Η κλάση αυτή περιέχει τις κατάλληλες δηλώσεις μεταβλητών (τύπου customer) και μεθόδους (set-get) για την αποθήκευση και επεξεργασία δεδομένων από τον πίνακα customer. Επίσης δημιουργεί το παράθυρο διαλόγου προβολής και επεξεργασίας με ξεχωριστά πεδία (jTextFields) για κάθε μια από τις στήλες του πίνακα customer εκτός του νομού που επιλέγεται από ένα μενού πολλαπλών επιλογών

(nomosCombobox-jComboBox). Τα δεδομένα του μενού αυτού αντλούνται από τον πίνακα νομοί με βάση το ID τους αλλά εμφανίζει μόνο την περιγραφή χρησιμοποιώντας τον μετατροπέα της κλάσης `NomoiListCellRenderer`. Ακόμα υπάρχει και μια κλήση της μεθόδου “ShowTheMap” από την κλάση `KlfView` που θα περιγράψουμε παρακάτω τη λειτουργία της. Τέλος, υπάρχουν δυο κουμπιά (`jButton`) για αποθήκευση ή ακύρωση της φόρμας.

### ▪ **Customers.java**

Η κλάση αυτή περιέχει τις κατάλληλες μεθόδους για την ανάγνωση, εγγραφή και επεξεργασία όλων των πεδίων του πίνακα `customers` της βάσης δεδομένων. Επίσης υπάρχει η ανάλογη δήλωση συσχετισμού του πίνακα `customers` με τους πίνακες `customersproducts`, `nomoi` και οι μέθοδοι για την ανάγνωση και εγγραφή των δεδομένων από τους συσχετισμούς αυτούς.

### ▪ **Customersproducts.java**

Η κλάση αυτή περιέχει τις κατάλληλες μεθόδους για την ανάγνωση, εγγραφή και επεξεργασία όλων των πεδίων του πίνακα `customersproducts` της βάσης δεδομένων. Επίσης υπάρχει η ανάλογη δήλωση συσχετισμού του πίνακα `customersproducts` με τους πίνακες `products`, `sintirisis` και οι μέθοδοι για την ανάγνωση και εγγραφή των δεδομένων από τους συσχετισμούς αυτούς.

### ▪ **CustomersProductsEditor.java**

Η κλάση αυτή περιέχει τις κατάλληλες δηλώσεις μεταβλητών (τύπου `customersproducts`) και μεθόδους (`set-get`) για την αποθήκευση και επεξεργασία δεδομένων από τον πίνακα `customersproducts`. Επίσης δημιουργεί το παράθυρο διαλόγου προβολής και επεξεργασίας με ξεχωριστά πεδία (`jTextFields`) για κάθε μια από τις στήλες του πίνακα `customersproducts` εκτός του `product ID` που επιλέγεται από ένα πίνακα πολλαπλών επιλογών (`productTable-jTable`). Το πεδίο της ημερομηνίας αγοράς ελέγχεται για την σωστή μορφή του από την κλάση `DateVerifier`. Επιπλέον υπάρχει ένα πεδίο (`searchTextField-jTextField`) αναζήτησης για την γρήγορη εύρεση του επιθυμητού προϊόντος με τη χρήση της κλάσης `RowSorterToStringConverter`. Τέλος, υπάρχουν δυο κουμπιά (`jButton`) για αποθήκευση ή ακύρωση της φόρμας.

### ▪ **DateVerifier.java**

Η κλάση αυτή χρησιμοποιείται για τον έλεγχο της ημερομηνίας που πληκτρολογείται σε κάποιο `jTextField` ώστε να έχει τη μορφή `HH/MM/XXXX` και προβάλλει μήνυμα λάθους σε περίπτωση που δεν έχει καταχωρηθεί η σωστή μορφή ημερομηνίας.

## ▪ **dbcom.java**

Στην κλάση αυτή υπάρχουν όλες οι μέθοδοι για την εκτέλεση όλων των περαιτέρω λειτουργιών (εκτός των βασικών λειτουργιών αποθήκευσης, επεξεργασίας και διαγραφής δεδομένων στους πίνακες customers, customersproducts, products, sintirisis) που απαιτούν επικοινωνία με τη βάση δεδομένων. Οι σημαντικότερες μέθοδοι της κλάσης αυτής είναι:

- connect: Δημιουργεί σύνδεση με τη βάση δεδομένων.
- disconnect: Καταργεί τη σύνδεση με τη βάση δεδομένων.
- fillPrintTable: Γεμίζει πίνακα jTable με τα αποτελέσματα μιας query που συνδυάζει τα δεδομένα από 3 πίνακες της βάσης για να χρησιμοποιηθεί για εκτύπωση.
- fillServiceTable: Γεμίζει πίνακα jTable με τα αποτελέσματα μιας query που συνδυάζει τα δεδομένα από 3 πίνακες της βάσης για να χρησιμοποιηθεί για ενημέρωση σχετικά με τις συντηρήσεις που πρέπει να γίνουν.

## ▪ **KlfAboutBox.java**

Η κλάση αυτή δημιουργεί ένα παράθυρο με πληροφορίες σχετικά με την εφαρμογή

## ▪ **KlfApplication.java**

Η Κλάση αυτή είναι η κεντρική κλάση της εφαρμογής που ξεκινά την λειτουργία της και ανοίγει το κεντρικό παράθυρο.

## ▪ **KlfView.java**

Η κλάση αυτή είναι η σημαντικότερη κλάση της εφαρμογής η οποία είναι και το κεντρικό παράθυρο της. Στο παράθυρο αυτό εμφανίζονται τα μενού της εφαρμογής, ένα dropdown μενού και 3 καρτέλες.

Στην πρώτη καρτέλα είναι ο πίνακας με τις εργασίες συντήρησης και δυο κουμπιά, για την εκτύπωση της λίστας (printButton-jButton) και προβολή χάρτη (mapButton-jButton). Επίσης υπάρχει ένα κουμπί ανανέωσης (refreshButton-jButton) και ένα μενού επιλογής νομού (nomoiComboBox-jComboBox). Οι κυριότερες μέθοδοι που αφορούν τη λειτουργία αυτής της καρτέλας είναι:

- nomosSelected: Καλεί μέθοδο από την κλάση dbcom για να γεμίσει τον πίνακα.
- showMapPressed: Καλεί μέθοδο από την κλάση dbcom με όρισμα το Id του επιλεγμένου πελάτη και εμφανίζει τον χάρτη με τη διεύθυνση του.

Στην δεύτερη καρτέλα υπάρχουν 3 πίνακες (customerTable, customersProdTable, sintirisisTable – jTable) με 3 κουμπιά (jButton) κάτω από τον κάθε πίνακα για προσθήκη επεξεργασία και διαγραφή δεδομένων. Επιπλέον υπάρχει ένα πεδίο (searchTextField-jTextField) αναζήτησης για την γρήγορη εύρεση του επιθυμητού πελάτη από τον πρώτο πίνακα με τη χρήση της κλάσης RowSorterToStringConverter. Οι κυριότερες μέθοδοι που αφορούν τη λειτουργία αυτής της καρτέλας είναι:

- newRecord: Δημιουργεί νέο παράθυρο της κλάσης CustomerEditor για προσθήκη δεδομένων στον πίνακα customer της βάσης.
- deleteRecord: Διαγράφει τα δεδομένα της επιλεγμένης καταχώρισης από το πίνακα customerTable αν επιβεβαιωθεί στο παράθυρο ειδοποίησης που εμφανίζει.
- editCustomer: Δημιουργεί νέο παράθυρο της κλάσης CustomerEditor με δεδομένα της επιλεγμένης καταχώρισης από το πίνακα customerTable για επεξεργασία.
- newDetailRecord: Δημιουργεί νέο παράθυρο της κλάσης CustomersProductsEditor για προσθήκη δεδομένων στον πίνακα customersproducts της βάσης.



- deleteDetailRecord: Διαγράφει τα δεδομένα της επιλεγμένης καταχώρισης από το πίνακα customersProdTable αν επιβεβαιωθεί στο παράθυρο ειδοποίησης που εμφανίζει.
- editBought: Δημιουργεί νέο παράθυρο της κλάσης CustomersProductsEditor με δεδομένα της επιλεγμένης καταχώρισης από το πίνακα customersProdTable για επεξεργασία.
- newSintirisisRecord: Δημιουργεί νέο παράθυρο της κλάσης SintirisisEditor για προσθήκη δεδομένων στον πίνακα sintirisis της βάσης.
- deleteSintirisisRecord: Διαγράφει τα δεδομένα της επιλεγμένης καταχώρισης από το πίνακα sintirisisTable αν επιβεβαιωθεί στο παράθυρο ειδοποίησης που εμφανίζει.
- edit Sintirisi: Δημιουργεί νέο παράθυρο της κλάσης SintirisisEditor με δεδομένα της επιλεγμένης καταχώρισης από το πίνακα sintirisisTable για επεξεργασία.
- Save: Καλείτε για αποθήκευση δεδομένων στη βάση

Στην τρίτη καρτέλα υπάρχει ο πίνακας προϊόντων (productsShowTable – jTable) με 3 πεδία κειμένου (codeProductTextField, modelProductTextField και infoProductTextArea) συνδεδεμένα με την επιλεγμένη σειρά από τον πίνακα. Επίσης υπάρχουν 4 κουμπιά (jButton) για αποθήκευση, ανανέωση, προσθήκη και διαγραφή. Οι κυριότερες μέθοδοι που αφορούν τη λειτουργία αυτής της καρτέλας είναι:

- newProductRecord: Δημιουργεί νέα σειρά στον πίνακα της κλάσης Products για προσθήκη δεδομένων στον πίνακα products της βάσης.
- deleteProductRecord: Διαγράφει τα δεδομένα της επιλεγμένης καταχώρισης από το πίνακα productsShowTable αν επιβεβαιωθεί στο παράθυρο ειδοποίησης που εμφανίζει.
- saveProduct: Αποθηκεύει τις αλλαγές που έχουν γίνει είτε από επεξεργασία είτε από καινούρια καταχώρηση.
- RefreshProductTask: Επαναφέρει τα δεδομένα του πίνακα productsShowTable διαβάζοντας τα ξανά από την βάση

## ▪ **Nomoi.java**

Η κλάση αυτή περιέχει τις κατάλληλες μεθόδους για την ανάγνωση, εγγραφή και επεξεργασία όλων των πεδίων του πίνακα nomoi της βάσης δεδομένων.

## ▪ **NomoiListCellRenderer.java**

Η κλάση αυτή δημιουργεί μια συσχέτιση της ονομασίας του νομού με το id του πίνακα nomoi της βάσης για την χρήση του id από την εφαρμογή για τις λειτουργίες της αλλά προβολή της ονομασίας στο χρήστη.

## ▪ **PrintPage.java**

Η κλάση αυτή ενεργοποιείται μέσα από την KlfView. Εμφανίζει το παράθυρο διαλόγου στο οποίο υπάρχει ο πίνακας printTable, που εμφανίζει δεδομένα χρήσιμα για το τεχνικό τμήμα. Υπάρχει επίσης ένα μενού επιλογής νομού (nomoiComboBox-jComboBox) το οποίο διαμορφώνει τα αποτελέσματα του πίνακα. Ακόμα υπάρχει και το κουμπί (printButton-jButton) που στέλνει τον πίνακα για εκτύπωση. Οι κυριότερες μέθοδοι που αφορούν τη λειτουργία αυτής της κλάσης είναι:

- printButtonActionPerformed: Ανοίγει το παράθυρο διαλόγου εκτύπωσης του πίνακα με τις ρυθμίσεις του εκτυπωτή.

- `nomoiComboBoxActionPerformed`: Διαμορφώνει τα αποτελέσματα του πίνακα σύμφωνα με αυτό που έχει επιλεγθεί από το `nomoiComboBox`.

### ▪ **Products.java**

Η κλάση αυτή περιέχει τις κατάλληλες μεθόδους για την ανάγνωση, εγγραφή και επεξεργασία όλων των πεδίων του πίνακα `products` της βάσης δεδομένων.

### ▪ **RowSorterToStringConverter.java**

Η κλάση αυτή χρησιμοποιείτε για αναζήτηση σε πίνακες της εφαρμογής. Μετατρέπει τις σειρές με τα δεδομένα κάποιου πίνακα (`JTable`) σε ένα `string` και και αναζητά εκεί το περιεχόμενο κάποιου πεδίου κειμένου (`JTextLabel`).

### ▪ **Sintirisis.java**

Η κλάση αυτή περιέχει τις κατάλληλες μεθόδους για την ανάγνωση, εγγραφή και επεξεργασία όλων των πεδίων του πίνακα `sintirisis` της βάσης δεδομένων. Επίσης υπάρχει η ανάλογη δήλωση συσχετισμού του πίνακα `sintirisis` με του πίνακα `customersproducts` και οι μέθοδοι για την ανάγνωση και εγγραφή των δεδομένων από το συσχετισμό αυτό.

### ▪ **SintirisisEditor.java**

Η κλάση αυτή περιέχει τις κατάλληλες δηλώσεις μεταβλητών (τύπου `sintirisis`) και μεθόδους (`set-get`) για την αποθήκευση και επεξεργασία δεδομένων από τον πίνακα `sintirisis`. Επίσης δημιουργεί το παράθυρο διαλόγου προβολής και επεξεργασίας με ξεχωριστά πεδία (`JTextFields`) για κάθε μια από τις στήλες του πίνακα `sintirisis` εκτός του `payed` και του `repair` που έχουν τη μορφή επιλογής με “ν” (`payedCheckBox`, `repairCheckBox`). Το πεδίο της ημερομηνίας συντήρησης ελέγχεται για την σωστή μορφή του από την κλάση `DateVerifier`. Τέλος, υπάρχουν δυο κουμπιά (`JButton`) για αποθήκευση ή ακύρωση της φόρμας.

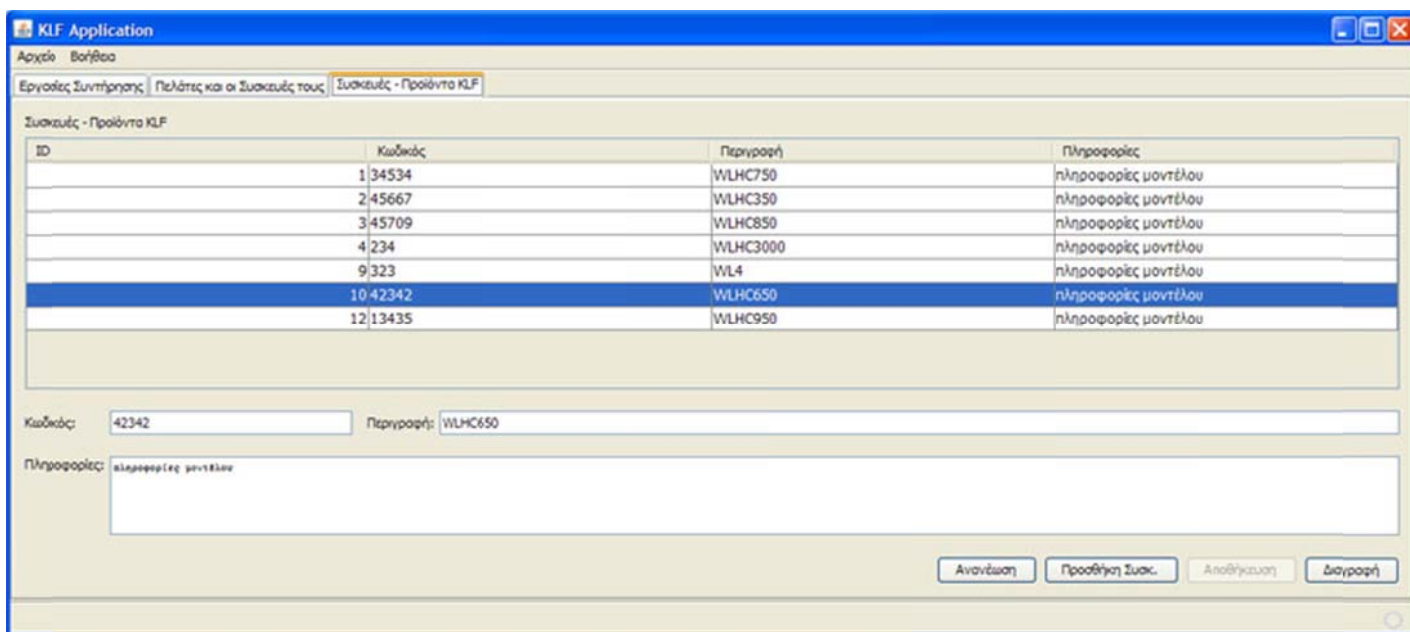
## 6. Οδηγίες χρήσεως της εφαρμογής

Οι λειτουργίες του προγράμματος είναι κατανεμημένες σε καρτέλες και ένα drop down μενού με επιλογές.

### **Καρτέλες:**

1. **Εργασίες συντήρησης**
2. **Πελάτες και οι συσκευές τους**
3. **Συσκευές-προϊόντα KLF**

Για λειτουργικούς λόγους και για να γίνουν οι οδηγίες χρήσεως ευκολότερα κατανοητές στο χρήστη θα ξεκινήσουμε την περιγραφή από την τελευταία καρτέλα και θα φτάσουμε στην πρώτη.



**Εικόνα 6. α : Καρτέλα συσκευές-προϊόντα**

**Συσκευές-προϊόντα KLF:** Σε αυτή την καρτέλα βλέπουμε τον πίνακα που είναι καταχωρημένες οι συσκευές και τα προϊόντα της εταιρίας. Αναγράφεται ο κωδικός του προϊόντος, το μοντέλο της συσκευής ή του προϊόντος και διάφορες πληροφορίες που αφορούν το προϊόν ή τη συσκευή.

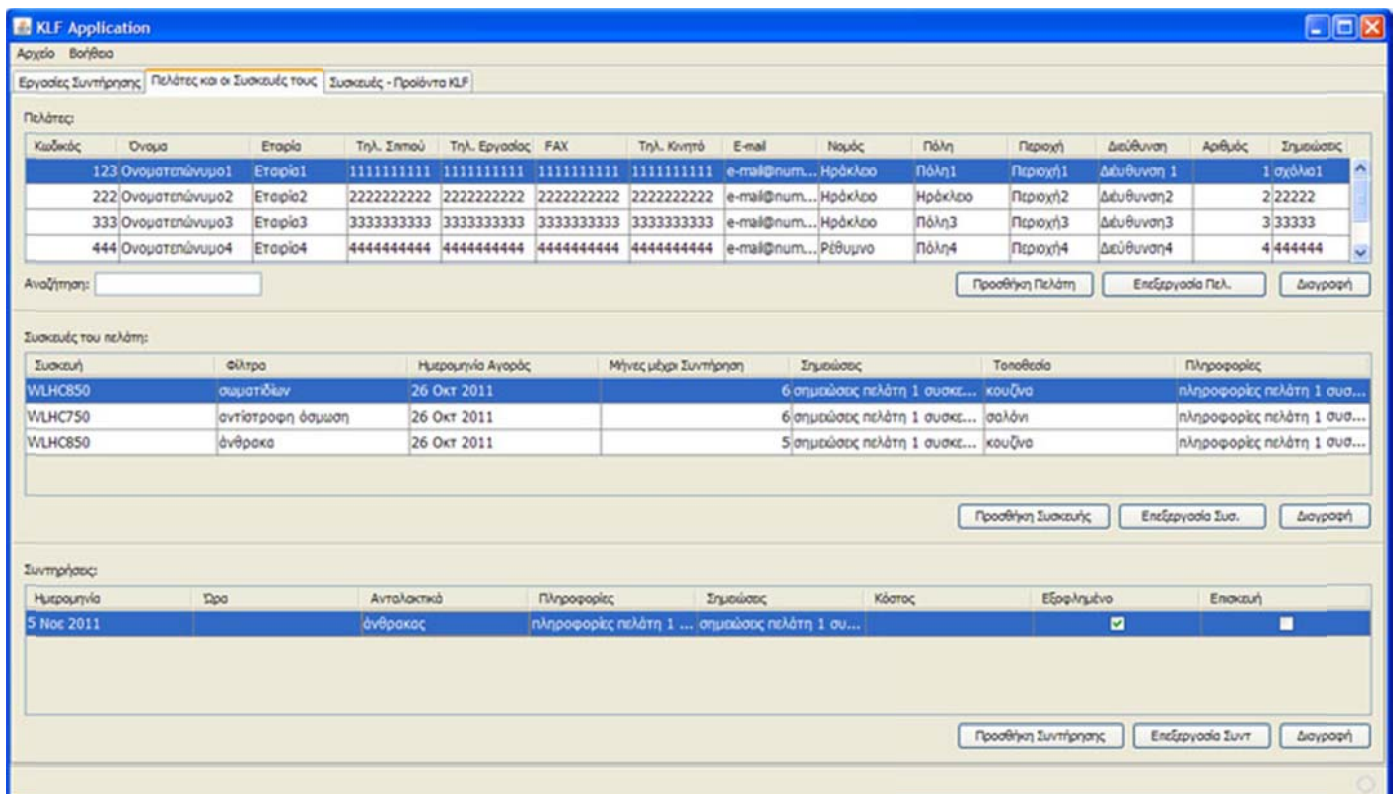
Παρακάτω θα δούμε τρία πεδία για τον κωδικό, το μοντέλο του προϊόντος ή της συσκευής και τις πληροφορίες. Πατώντας ένα «κλικ» πάνω σε κάποιο προϊόν της εταιρίας αμέσως εμφανίζονται τα στοιχεία του κάθε προϊόντος στα τρία πεδία κάτω από τον πίνακα από όπου δίνεται η δυνατότητα στον χρήστη να κάνει μεταβολές, να τις αποθηκεύσει πατώντας το πλήκτρο «Αποθήκευση» ή να επαναφέρει το προϊόν όπως ήταν πριν τις αλλαγές πατώντας το κουμπί «Ανανέωση»

Πατώντας την επιλογή «Προσθήκη συσκευής» εμφανίζεται στον πίνακα άλλο ένα πλαίσιο και πηγαίνοντας κάτω στα τρία πεδία συμπληρώνουμε τα στοιχεία που χρειάζονται για το κάθε προϊόν και πατάμε την επιλογή «Αποθήκευση» για να αποθηκευτούν τα στοιχεία του προϊόντος.

Για να επεξεργαστούμε κάποια συσκευή ή προϊόν της εταιρίας επιλέγουμε από τον πίνακα ένα προϊόν πατώντας ένα «κλικ» και πηγαίνουμε παρακάτω στις μπάρες, αλλάζουμε τα στοιχεία που θέλουμε και πατάμε αποθήκευση.

Η επιλογή «Διαγραφή» διαγράφει την συσκευή που έχουμε επιλέξει εμφανίζοντας ένα προειδοποιητικό μήνυμα. Σε περίπτωση που το συγκεκριμένο προϊόν έχει αγοραστεί από κάποιο πελάτη εμφανίζεται κατάλληλο μήνυμα και αποτρέπεται η διαγραφή.

Όταν θελήσουμε να κάνουμε ανανέωση των δεδομένων του πίνακα χωρίς να αποθηκεύσουμε τυχόν αλλαγές που έχουμε κάνει απλά πατάμε με ένα «κλικ» την επιλογή «Ανανέωση» και αυτόματα γίνεται η διεργασία επαναφοράς των δεδομένων.



Εικόνα 6. β : Καρτέλα πελάτες και οι συσκευές τους

**Πελάτες και οι συσκευές τους:** Σε αυτή την καρτέλα μας παρουσιάζονται τα στοιχεία των πελατών, τα στοιχεία των συσκευών των πελατών και το ιστορικό συντηρήσεως κάθε συσκευής του πελάτη.

Στον πρώτο πίνακα αναγράφονται τα στοιχεία των πελατών με τον κωδικό, το όνομα, την εταιρία, το τηλ. σπιτιού, το τηλ. εργασίας, Fax, τηλ κινητό, E-mail, νομό, πόλη, περιοχή, διεύθυνση, αριθμό και σημειώσεις.

Κάτω από αυτό τον πίνακα υπάρχει η μπάρα της «αναζήτησης» οπου εκεί γράφουμε οποιοδήποτε στοιχείο γνωρίζουμε είτε είναι κάποιο όνομα είτε είναι κάποιος αριθμός η κάποια αρχικά της λέξης και καθώς πληκτρολογούμε αφαιρούνται από τον πίνακα οι καταχωρήσεις που δεν περιέχουν αυτά που πληκτρολογούμε. Άρα μένουν στον πίνακα οι καταχωρήσεις που περιέχουν αυτό που έχουμε πληκτρολογήσει.

Οι στήλες μπορούν να μεγαλώσουν και να μικρύνουν και πατώντας με ένα «κλικ» πάνω σε οποιαδήποτε στήλη γίνεται ταξινόμηση σε αλφαβητική σειρά και με το δεύτερο «κλικ» γίνεται αντίστροφη αλφαβητική ταξινόμηση.

Όταν θελήσουμε να προσθέσουμε κάποιο νέο πελάτη, απλά πηγαίνουμε στην επιλογή «Προσθήκη πελάτη».

The screenshot shows a window titled "Στοιχεία Πελάτη" (Customer Details). The form contains the following fields:

- Κωδικός: [Text input]
- Όνοματεπώνυμο: [Text input]
- Εταιρία: [Text input]
- Τηλ. Σπιτιού: [Text input]
- Τηλ. Εργασίας: [Text input]
- FAX: [Text input]
- Κινητό: [Text input]
- E-Mail: [Text input]
- Νομός: [Dropdown menu]
- Πόλη: [Text input]
- Περιοχή: [Text input]
- Διεύθυνση: [Text input]
- Αριθμός: [Text input]
- Γεωγρ. Μήκος: [Text input]
- Γεωγρ. Πλάτος: [Text input]
- Προβολή Χάρτη: [Button]
- Σημειώσεις: [Large text area]
- Αποθήκευση: [Button]
- Άκυρο: [Button]

**Εικόνα 6. γ : Παράθυρο προσθήκης πελάτη**

Τότε μας εμφανίζεται το παραπάνω παράθυρο στο οποίο συμπληρώνουμε τα στοιχεία που αναγράφονται σε αυτό όπως τον κωδικό, το όνομα, την εταιρία, το τηλ. σπιτιού, το τηλ. εργασίας, Fax, τηλ κινητό, E-mail, νομό, πόλη, περιοχή, διεύθυνση, αριθμό, γεωγρ. μήκος, γεωγρ. πλάτος, προβολή χάρτη και σημειώσεις.

Ακόμα εμφανίζονται δυο επιπλέον στοιχεία, το γεωγρ. μήκος και πλάτος τα οποία θα χρησιμοποιηθούν για να εμφανίζεται η τοποθεσία του πελάτη πατώντας την επιλογή «**Προβολή χάρτη**» στο Google maps ή για χρήση τους σε συσκευή GPS. Και πατάμε αποθήκευση αν θέλουμε τη αποθήκευση των δεδομένων που καταχωρήσαμε και άκυρο αν θέλουμε να επιστρέψουμε στην προηγούμενη σελίδα χωρίς αλλαγές. Σε περίπτωση που δεν έχουμε βάλει τις συντεταγμένες στο γεωγραφικό μήκος και πλάτος τότε εμφανίζεται προειδοποιητικό μήνυμα έλλειψης συντεταγμένων.

Όταν θελήσουμε να επεξεργαστούμε τα στοιχεία ενός πελάτη τότε πηγαίνουμε στην επιλογή «**Επεξεργασία πελάτη**» οπου και εκεί εμφανίζεται το ίδιο παράθυρο όπως της επιλογής «**Προσθήκη πελάτη**» και απλά προσθέτουμε, αφαιρούμε ή αλλάζουμε τα στοιχεία του πελάτη και πατάμε αποθήκευση. Μπορεί ακόμα και να μην χρειαστεί να κάνουμε κάποια αλλαγή στα στοιχεία οπότε πατάμε άκυρο.

Πατώντας την επιλογή «**Διαγραφή**» εμφανίζεται αρχικά το μήνυμα «Θέλετε σίγουρα να διαγράψετε τον επιλεγμένο πελάτη;» και πατώντας «**ναι**» γίνεται η διαγραφή όλων των στοιχείων του πελάτη, των συσκευών που έχει αγοράσει και το ιστορικό των συντηρήσεων των συσκευών. Αν δεν θέλουμε να διαγραφούν αυτά τα στοιχεία πατάμε «**όχι**».

Στον δεύτερο πίνακα της καρτέλας αναγράφονται τα στοιχεία των συσκευών των πελατών με την συσκευή, τα φίλτρα, την ημερομηνία αγοράς, τους μήνες μέχρι την συντήρηση, σημειώσεις, τοποθεσία δηλαδή σε ποιο σημείο του σπιτιού είναι τοποθετημένη η συσκευή και πληροφορίες που τυχόν θα χρειαστεί να σημειώσουμε.

Μια βασική λεπτομέρεια που μπορούμε να αναφέρουμε είναι ότι αφού έχουμε συμπληρώσει και στον πρώτο και στο δεύτερο πίνακα της καρτέλας τα απαραίτητα στοιχεία, μπορούν πλέον πατώντας ένα «κλικ» πάνω στο όνομα ή σε κάποιο άλλο στοιχείο του πελάτη από τον πρώτο πίνακα, αυτομάτως να εμφανιστούν οι συσκευές που έχει αγοράσει ο πελάτης με όλες τις λεπτομέρειες που αναγράφονται πάνω στον πίνακα.

Όταν θελήσουμε να προσθέσουμε κάποια νέα συσκευή που αγόρασε ο πελάτης επιλέγουμε αρχικά τον πελάτη και στη συνέχεια πατάμε την επιλογή **«Προσθήκη συσκευής»**.

Συσκευή του Πελάτη

Συσκευή του Πελάτη

Επιλέξτε μια από τις παρακάτω διαθέσιμες συσκευές:

ID	Κωδικός	Περιγραφή	Πληροφορίες
1	34534	WLHC750	πληροφορίες μοντέλου
2	45667	WLHC350	πληροφορίες μοντέλου
3	45709	WLHC850	πληροφορίες μοντέλου
4	234	WLHC3000	πληροφορίες μοντέλου
9	323	WL4	πληροφορίες μοντέλου
10	42342	WLHC650	πληροφορίες μοντέλου
12	13435	WLHC950	πληροφορίες μοντέλου

Αναζήτηση:

Πληροφορίες Συσκευής:

Μορφή: Η/Μ/XXXX

Ημερομηνία αγοράς:  Τοποθετημένη:

Μήνες για συντήρηση:  Φίλτρα:

Πληροφορίες:

Σημειώσεις:

Αποθήκευση  Άκυρο

Εικόνα 6. δ : Παράθυρο προσθήκης συσκευής του πελάτη

Τότε μας εμφανίζεται ένα παράθυρο με ένα πίνακα ο οποίος εμφανίζει όλα τα μοντέλα των συσκευών της εταιρίας και εμείς επιλέγουμε το μοντέλο που έχει αγοράσει ο πελάτης . Ακόμα, με την επιλογή «Αναζήτηση» μπορούμε εκεί να γράψουμε οποιοδήποτε στοιχείο γνωρίζουμε είτε είναι κάποιο όνομα είτε είναι κάποιος αριθμός η κάποια αρχικά της λέξης και καθώς πληκτρολογούμε αφαιρούνται από τον πίνακα οι καταχωρήσεις που δεν περιέχουν αυτά που πληκτρολογούμε. Άρα μένουν στον πίνακα οι καταχωρήσεις που περιέχουν αυτό που έχουμε πληκτρολογήσει.

Λόγω του ότι ο πίνακας είναι μικρός και μπορεί κάποιες πληροφορίες του μοντέλου ή της συσκευής να μην φαίνονται, θα εμφανίζονται στο πλαίσιο «πληροφορίες συσκευής». Επιπλέον υπάρχουν και άλλα πεδία όπως η ημερομηνία αγοράς της συσκευής οπότε εκεί θα εμφανίζει το πρόγραμμα αυτόματα την σημερινή ημερομηνία καταχώρησης εκτός και αν δεν είναι η επιθυμητή και θελήσουμε εμείς να την αλλάξουμε, οι μήνες για συντήρηση οπότε σε αυτό το πεδίο συμπληρώνουμε ανά πόσους μήνες θα γίνεται η συντήρηση της συσκευής, σε ποιο σημείο στο χώρο του πελάτη είναι τοποθετημένη η συσκευή και ποια φίλτρα χρησιμοποιήθηκαν στη συσκευή. Και πατάμε αποθήκευση αν θέλουμε τη αποθήκευση των δεδομένων που καταχωρήσαμε και άκυρο αν θέλουμε να επιστρέψουμε στην προηγούμενη σελίδα χωρίς αλλαγές.

Όταν θελήσουμε να επεξεργαστούμε τα στοιχεία μιας συσκευής του πελάτη τότε πηγαίνουμε στην επιλογή «Επεξεργασία συσκευής» οπότε και εκεί εμφανίζεται το ίδιο παράθυρο όπως της επιλογής «Προσθήκη συσκευής» και απλά προσθέτουμε, αφαιρούμε ή αλλάζουμε τα στοιχεία της συσκευής του πελάτη και πατάμε αποθήκευση. Μπορεί ακόμα και να μην χρειαστεί να κάνουμε κάποια αλλαγή στα στοιχεία οπότε πατάμε άκυρο.



Πατώντας την επιλογή «**Διαγραφή**» εμφανίζεται αρχικά το μήνυμα «Θέλετε σίγουρα να διαγράψετε την επιλεγμένη συσκευή του πελάτη;» και πατώντας «**ναι**» γίνεται η διαγραφή της συσκευής που επιλέξαμε. Αν δεν θέλουμε να διαγραφτεί πατάμε «**όχι**».

Στον τρίτο πίνακα της καρτέλας αναγράφονται τα στοιχεία των εργασιών συντήρησης που έχουν γίνει σε μια συγκεκριμένη συσκευή που έχει αγοράσει ο πελάτης με την ημερομηνία που είχε γίνει η εργασία συντήρησης, την ώρα που έγινε, τα ανταλλακτικά που χρησιμοποιήθηκαν για τη συντήρηση, πληροφορίες που τυχόν θα χρειαστεί να σημειώσουμε, σημειώσεις, το κόστος των εργασιών της συντήρησης, αν είναι εξοφλημένη η εργασία στην οποία θα το καταλαβαίνουμε βάζοντας ένα «**τικ**» μέσα στο τετραγωνάκι που υπάρχει και τέλος αν έχει γίνει κάποια εργασία επισκευής στην οποία κι εδώ θα το καταλαβαίνουμε αν έχουμε βάλει στο τετραγωνάκι που υπάρχει ένα «**τικ**».

Μια σημαντική πληροφορία που μπορούμε να αναφέρουμε στο πεδίο «**Εργασία επισκευής**» είναι ότι αν είχε γίνει επισκευή σε κάποια συσκευή στην τελευταία ημερομηνία των εργασιών συντήρησης και όχι συντήρηση, τότε το πρόγραμμα δεν θα λάβει υπόψη του σαν εργασία συντήρησης την τελευταία ημερομηνία αλλά σαν επισκευή.

Όταν θελήσουμε να προσθέσουμε την εργασίας συντήρησης που έχει γίνει στη συσκευής που έχει αγοράσει ο πελάτης αρχικά επιλέγουμε τον πελάτη από τον πρώτο πίνακα στη συνέχεια στο δεύτερο πίνακα εμφανίζονται οι συσκευές που έχει αγοράσει ο πελάτης και επιλέγουμε την συσκευή στην οποία έγιναν οι εργασίες συντήρησης και στη συνέχεια στον τρίτο πίνακα της καρτέλας παρακάτω πατάμε την επιλογή «**Προσθήκη συντήρησης**».

Εικόνα 6. ε : Παράθυρο προσθήκης εργασιών συντήρησης

Τότε εμφανίζεται ένα νέο παράθυρο και συμπληρώνουμε τα στοιχεία που αναγράφονται εκεί όπως την ημερομηνία που είχε γίνει η εργασία συντήρησης, την ώρα που έγινε, τα ανταλλακτικά που χρησιμοποιήθηκαν για τη συντήρηση, πληροφορίες που τυχόν θα χρειαστεί να σημειώσουμε, σημειώσεις, το κόστος των εργασιών της συντήρησης, αν είναι εξοφλημένη η εργασία στην οποία θα το καταλαβαίνουμε βάζοντας ένα «**τικ**» μέσα στο τετραγωνάκι που υπάρχει και τέλος αν έχει γίνει κάποια εργασία επισκευής στην οποία κι εδώ θα το καταλαβαίνουμε αν έχουμε βάλει στο τετραγωνάκι που υπάρχει ένα «**τικ**» και πατάμε αποθήκευση αν θέλουμε τη αποθήκευση των δεδομένων που καταχωρήσαμε και άκυρο αν θέλουμε να επιστρέψουμε στην προηγούμενη σελίδα χωρίς αλλαγές.

Όταν θελήσουμε να επεξεργαστούμε τα στοιχεία μιας εργασίας συντήρησης σε μια συσκευή του πελάτη τότε επιλέγουμε από τον πίνακα τη συντήρηση που θέλουμε να επεξεργαστούμε πηγαίνουμε στην επιλογή «**Επεξεργασία συντήρησης**» οπού και εκεί εμφανίζεται το ίδιο παράθυρο όπως της επιλογής «**Προσθήκη συντήρησης**» και απλά προσθέτουμε, αφαιρούμε ή αλλάζουμε τα στοιχεία της συντήρησης και πατάμε αποθήκευση. Μπορεί ακόμα και να μην χρειαστεί να κάνουμε κάποια αλλαγή στα στοιχεία οπότε πατάμε άκυρο.

Πατώντας την επιλογή «**Διαγραφή**» αφού πρώτα έχουμε επιλέξει από τον πίνακα ποια συντήρηση θέλουμε να διαγράψουμε, εμφανίζεται το μήνυμα «Θέλετε σίγουρα να διαγράψετε την επιλεγμένη καταχωρημένη συντήρηση;» και πατώντας «**ναι**» γίνεται η διαγραφή της συντήρησης που επιλέξαμε. Αν δεν θέλουμε να διαγραφεί πατάμε «**όχι**».

Τέλος υπάρχει άλλη μια επιλογή «**Ανανέωση**» οπού πατώντας το γίνεται ταυτόχρονα η ανανέωση των καταχωρημένων στοιχείων που υπάρχουν στο πρόγραμμα.

KLF Application

Αρχείο Βοήθεια

Εργασίες Συντήρησης | Πελάτες και οι Συσκευές τους | Συσκευές - Προϊόντα KLF

Επόμενες εργασίες συντήρησης που πρέπει να γίνουν στο:

ID	Όνοματεπώνυμο	Τηλέφωνο...	Τηλέφωνο...	Νομός	Μοντέλο Σ...	Τοποθεσία	Φίλτρο Συσκ.	Πληρωφ. α...	Ανταλλακτ...	Αγορά	Τελ. Συντ.	Επόμενη Σ...	Εξοφλημ...	Κόστος
14	Όνοματεπώνυμο2	2222222222	2222222222	Λασθίου	WLHC350	σαλόνι	αντίστροφη...			2011-10-26		2012-02-26	<input type="checkbox"/>	0
13	Όνοματεπώνυμο1	1111111111	1111111111	Ηρακλείου	WLHC850	κουζίνα	άνθρακα			2011-10-26		2012-03-26	<input type="checkbox"/>	0
17	Όνοματεπώνυμο5	5555555555	5555555555	Ρεθύμνου	WLHC3000	σαλόνι	αντίστροφη...			2011-10-28		2012-04-28	<input type="checkbox"/>	0
19	Όνοματεπώνυμο6	6666666666	6666666666	Ηρακλείου	WLHC850	bar	άνθρακα			2011-11-05		2012-05-05	<input type="checkbox"/>	0
13	Όνοματεπώνυμο1	1111111111	1111111111	Ηρακλείου	WLHC850	κουζίνα	συσταδίων	σημετώσας ...	άνθρακας	2011-10-26	2011-11-05	2012-05-05	<input checked="" type="checkbox"/>	0
13	Όνοματεπώνυμο1	1111111111	1111111111	Ηρακλείου	WLHC750	σαλόνι	αντίστροφη...	σημετώσας ...	άνθρακας	2011-10-26	2011-11-05	2012-05-05	<input type="checkbox"/>	0
15	Όνοματεπώνυμο3	3333333333	3333333333	Λασιθίου	WL4	Bar	αντίστροφη...	σημετώσας ...	λάμπα UV	2011-05-01	2011-11-05	2012-05-05	<input type="checkbox"/>	0
14	Όνοματεπώνυμο2	2222222222	2222222222	Λασιθίου	WLHC650	κουζίνα	άνθρακα	σημετώσας ...	λάμπα UV	2011-10-26	2011-11-05	2012-05-05	<input checked="" type="checkbox"/>	0
16	Όνοματεπώνυμο4	4444444444	4444444444	Χανίων	WLHC350	κουζίνα 2	συσταδίων			2011-10-28		2012-06-28	<input type="checkbox"/>	0

Εικόνα 6. ζ : Καρτέλα εργασίες συντήρησης

**Εργασίες συντήρησης:** Ουσιαστικά, αυτή η καρτέλα έχει ενημερωτικό χαρακτήρα. Εδώ μας παρουσιάζονται οι εργασίες συντήρησης που πρέπει να γίνουν για μια συσκευή που έχει αγοράσει κάποιος πελάτης.

Πάνω από αυτόν τον πίνακα υπάρχουν οι επιλογές «**Νομός**» και «**Ανανέωση**». Πατώντας ένα «κλικ» στην επιλογή «**Νομός**» εμφανίζεται η λίστα των νομών οι οποίοι θα είναι προκαταχωρημένοι στη βάση δεδομένων και πατώντας σε ένα συγκεκριμένο νομό θα εμφανίζονται αυτόματα οι πελάτες που βρίσκονται στον συγκεκριμένο νομό και χρειάζονται εργασίες συντήρησης για την συσκευή που έχουν αγοράσει. Επίσης υπάρχει η επιλογή «**Οποιοδήποτε**» που εμφανίζει όλους τους νομούς. Η επιλογή «**Ανανέωση**» κάνει ανανέωση στα δεδομένα της βάσης σε περίπτωση αλλαγής των δεδομένων.

Αρχικά εμφανίζεται ένας πίνακας με το ονοματεπώνυμο του πελάτη, το τηλέφωνο του σπιτιού, το τηλέφωνο της εργασίας, το νομό που βρίσκεται ο πελάτης, το μοντέλο της συγκεκριμένης συσκευής που έχει αγοράσει ο συγκεκριμένος πελάτης την οποία είχαμε επιλέξει από τον πίνακα συσκευών όταν καταχωρήσαμε τη συσκευή του πελάτη. Στη συνέχεια έχουμε το σημείο που είναι τοποθετημένη η συσκευή που έχει αγοράσει ο πελάτης την οποία είχαμε καταχωρήσει στο πεδίο «τοποθεσία», τα φίλτρα που χρειάστηκαν να τοποθετηθούν στη συσκευή του πελάτη τα οποία τα είχαμε καταχωρήσει στο πεδίο «**φίλτρα**». Ακόμα διάφορες πληροφορίες συντήρησης (οι οποίες θα εμφανίζονται μόνο σε περίπτωση που έχει γίνει κάποια συντήρηση διαφορετικά θα υπάρχει κενό.) που αναγράφονται για την συσκευή του πελάτη και αφορούν τις πληροφορίες συντήρησης που είχαν σημειωθεί από την τελευταία συντήρηση έτσι ώστε να επιτραπεί στον τεχνικό να έχει μια πλήρη εικόνα του τι έχει γίνει, τα ανταλλακτικά που χρησιμοποιήθηκαν στην τελευταία συντήρηση της συσκευής του πελάτη, την ημερομηνία που αγόρασε ο πελάτης τη συσκευή, την ημερομηνία τελευταίας συντήρησης της συσκευής του πελάτη, την ημερομηνία της επόμενης συντήρησης της συσκευής του πελάτη, αν είναι εξοφλημένο το κόστος της τελευταίας συντήρησης για τη συσκευή του πελάτη και τέλος πόσο ήταν το κόστος αυτό.

**Φόρμα Εκτύπωσης**

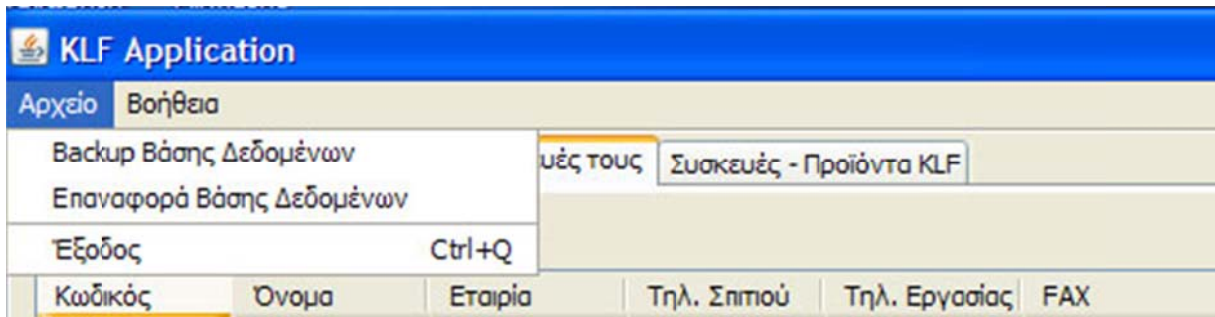
Όνοματεπώνυμο	Τηλ. Σπρ.	Τηλ. Εργ.	Διεύθυνση	Νο...	Μοντέλο	Τοποθεσία	Φίλτρο Συσκ.	Πληροφ. Σ...	Αντολ. Τελ...	Τελ. Συντ.	Επάμ. Συντ.	Κό...
Όνοματεπώνυμο2	2222222222	2222222222	Διεύθυνση2, 2	Λασ...	WLHC350	σαλόνι	αντίτρο...				2012-02-26	0
Όνοματεπώνυμο1	1111111111	1111111111	Διεύθυνση 1, 1	Ηρα...	WLHC850	κουζίνα	άνθρακα				2012-03-26	0
Όνοματεπώνυμο5	5555555555	5555555555	Διεύθυνση5, 5	Ρεθ...	WLHC3000	σαλόνι	αντίτρο...				2012-04-28	0
Όνοματεπώνυμο6	6666666666	6666666666	Διεύθυνση6, 6	Ηρα...	WLHC850	bar	άνθρακα				2012-05-05	0
Όνοματεπώνυμο1	1111111111	1111111111	Διεύθυνση 1, 1	Ηρα...	WLHC850	κουζίνα	σωματιδίων	σημαώσας...	άνθρακας	2011-11-05	2012-05-05	0
Όνοματεπώνυμο1	1111111111	1111111111	Διεύθυνση 1, 1	Ηρα...	WLHC750	σαλόνι	αντίτρο...	σημαώσας...	άνθρακας	2011-11-05	2012-05-05	0
Όνοματεπώνυμο3	3333333333	3333333333	Διεύθυνση3, 3	Λασ...	WL4	Bar	αντίτρο...	σημαώσας...	λάμπα UV	2011-11-05	2012-05-05	0
Όνοματεπώνυμο2	2222222222	Αφαίρεση στήλης	Διεύθυνση2, 2	Λασ...	WLHC650	κουζίνα	άνθρακα	σημαώσας...	λάμπα UV	2011-11-05	2012-05-05	0
Όνοματεπώνυμο4	4444444444	Επαναφορά στήλων	Διεύθυνση4, 4	Χαν...	WLHC350	κουζίνα 2	σωματιδίων				2012-06-28	0

Εκτύπωση του πίνακα    Για το νομό: Οποιοδήποτε    Άκυρο

**Εικόνα 6. η : Παράθυρο φόρμας εκτύπωσης**

Παραπάνω έχουμε την επιλογή «**Εκτύπωση**». Μέσα σε αυτό το πεδίο γράφουμε πόσες καταχωρήσεις θέλουμε να εκτυπωθούν και πατώντας την επιλογή «**Εκτύπωση**» εμφανίζεται η φόρμα εκτύπωσης με τη λίστα και εκτυπώνεται η λίστα με το επιθυμητό πλήθος καταχωρήσεων που έχουμε βάλει και η πληροφορίες που είναι χρήσιμες για τους τεχνικούς. Ακόμα όταν δεν θέλουμε να φαίνονται κάποια δεδομένα στη λίστα εκτύπωσης, μπορούμε πατώντας «**δεξί κλικ**» μέσα στον πίνακα σε οποιοδήποτε σημείο του πίνακα να μας εμφανίσει τις επιλογές «**Αφαίρεση στήλης**» και πατώντας στην αφαίρεση στήλης αφαιρείται η στήλη που δεν θέλουμε να φαίνεται. Όταν πατήσουμε την «**Επαναφορά στήλης**» τότε επαναφέρουμε τη στήλη που είχαμε αφαιρέσει.

Τέλος υπάρχει η επιλογή «**Προβολή χάρτη**» οπου όταν επιλέξουμε κάποιο πελάτη από τον πίνακα εργασιών συντήρησης και πατήσουμε την επιλογή «**Προβολή χάρτη**» τότε εμφανίζεται η τοποθεσία του πελάτη στο Google maps.



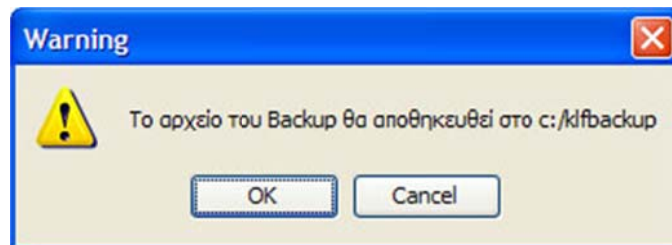
Εικόνα 6. θ : Drop down μενού

Drop down μενού: Το μενού drop down έχει τις παρακάτω επιλογές

- Αρχείο
- Βοήθεια

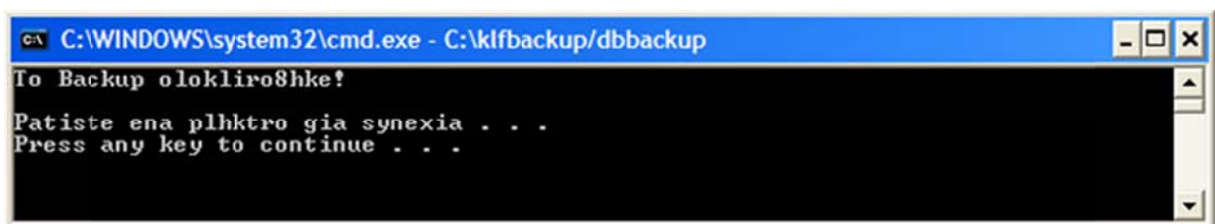
Το μενού «**Αρχείο**» έχει τις εξής επιλογές: «**Backup βάσης δεδομένων**», «**Επαναφορά βάσης δεδομένων**» και «**Έξοδος**».

Πατώντας την επιλογή **Backup βάσης δεδομένων** από το μενού «**Αρχείο**» εμφανίζεται ένα προειδοποιητικό μήνυμα που λέει « Το αρχείο του backup θα αποθηκευτεί στο c:/klfbackup» και έχει τις επιλογές ok και άκυρο.



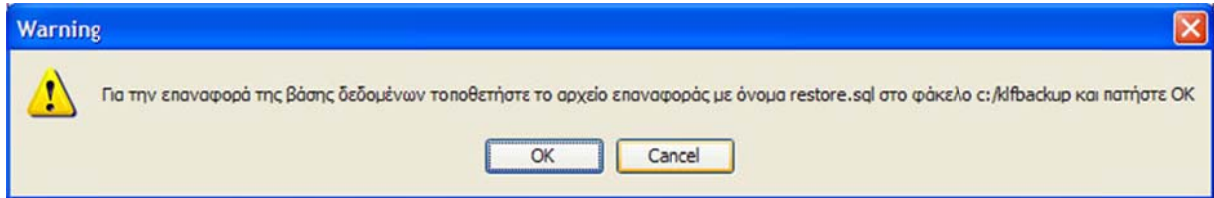
Εικόνα 6. ι : Προειδοποιητικό μήνυμα αποθήκευσης backup

Πατώντας την επιλογή «ok» τότε εμφανίζεται ένα άλλο παράθυρο το οποίο ενημερώνει ότι το backup ολοκληρώθηκε και θα πρέπει να πατήσουμε κάποιο πλήκτρο για συνέχεια. Αν δεν θέλουμε να κάνουμε backup πατάμε την επιλογή «**Άκυρο**».



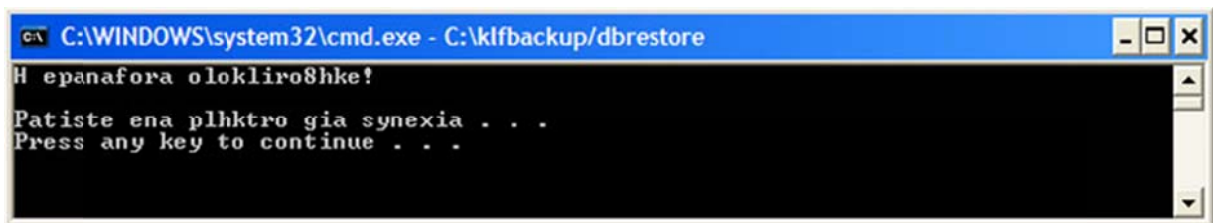
Εικόνα 6. κ : Παράθυρο ολοκλήρωσης backup

Για να επαναφέρουμε την βάση δεδομένων πατάμε την επιλογή «**Επαναφορά βάσης δεδομένων**» και εμφανίζεται κι εδώ ένα προειδοποιητικό μήνυμα που λέει: «Για την επαναφορά της βάσης δεδομένων τοποθετήστε το αρχείο επαναφοράς με όνομα restore.sql στο φάκελο c:/klfbackup και πατήστε «**ok**». Ακολουθούμε τα βήματα που μας λέει το προειδοποιητικό μήνυμα και στη συνέχεια πατάμε «**ok**».



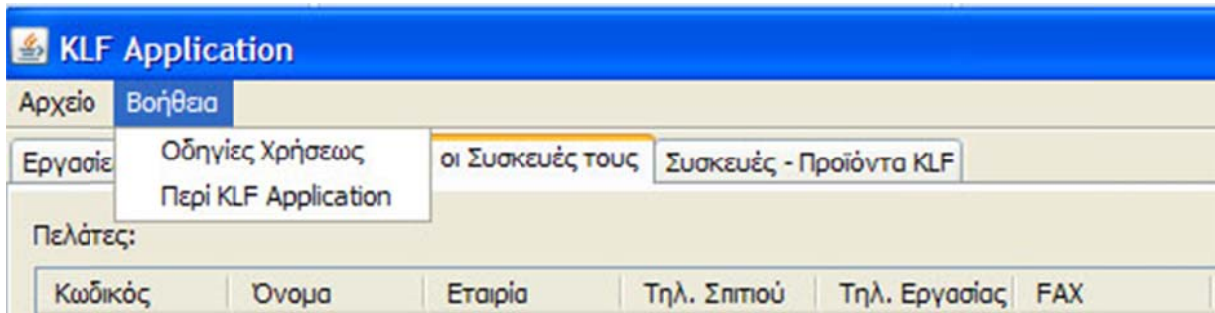
**Εικόνα 6. λ : Προειδοποιητικό μήνυμα βημάτων επαναφοράς**

Στη συνέχεια εμφανίζεται το μήνυμα ότι « η επαναφορά ολοκληρώθηκε πατήστε ένα πλήκτρο για συνέχεια».



**Εικόνα 6. μ : Παράθυρο ολοκλήρωσης επαναφοράς**

Τέλος, η επιλογή του dropdown menu, «**Εξόδος**» μας επιτρέπει να βγούμε από το πρόγραμμά μας.



**Εικόνα 6. ν : Μενού Βοήθεια**

Το μενού «Βοήθεια» έχει την επιλογή «Περί KLF Application» οπού εδώ αναφέρονται οι πληροφορίες για τον κατασκευαστή του προγράμματος



**Εικόνα 6. ξ : Παράθυρο πληροφοριών κατασκευαστή**

και την επιλογή «Οδηγίες χρήσεως» οπού εδώ θα μπορούμε να βλέπουμε τις οδηγίες χρήσεως της εφαρμογής.

Για τη λειτουργία της εφαρμογή, απαιτείται να είναι εγκατεστημένα στον υπολογιστή τα προγράμματα Java και MySQL. Συνίσταται η έκδοση του MySQL να είναι από 5.1 και άνω, ενώ η έκδοση της Java από 6 και άνω.

## 7. Αποτελέσματα

### 7.1 Συμπεράσματα

Από την αρχή που ξεκίνησα την πτυχιακή μου εργασία, σκοπός μου ήταν να ασχοληθώ με κάτι που να είναι χρήσιμο, αποτελεσματικό, αλλά και απόλυτα λειτουργικό. Τελειώνοντας την εργασία αυτή κατάφερα να πετύχω τους αρχικούς μου στόχους αλλά και να διατηρώ κάποιες επιπλέον φιλοδοξίες για μελλοντικές βελτιώσεις. Η δημιουργία αυτής της πτυχιακής εργασίας με βοήθησε να κατανοήσω καλύτερα τα βήματα που θα πρέπει να γίνουν ώστε να κατασκευαστεί μια εφαρμογή, απέκτησα την εμπειρία μιας πρώτης ολοκληρωμένης εφαρμογής από την αναζήτηση απαιτήσεων μέχρι το debugging της εφαρμογής. Είχα την δυνατότητα να δω πόσο χρόνο μπορεί να χριστεί κάθε κομμάτι αυτής της διαδικασίας και πόσο εύκολο είναι να αποκλίνει ο προγραμματιστής από το χρονοδιάγραμμα του. Κατά την διαδικασία συγγραφής της εργασίας αλλά και δημιουργίας της εφαρμογής είχα την δυνατότητα να αναγνωρίσω τις ελλείψεις γνώσεων που είχα σε κάποια θέματα, να τα καλύψω και να εμπλουτίσω τις γνώσεις που είχα. Με την συνεχή αναζήτηση πληροφοριών, κατάφερα να βελτιώσω τις μεθόδους μου για αναζήτηση πληροφοριών και σε κάποιες περιπτώσεις να τις αλλάξω προς το καλύτερο, πράγμα που θα με βοηθήσει και σε μελλοντικές εργασίες. Επίσης είδα στην πράξη τις δυνατότητες μου στον προγραμματισμό και ότι είναι εφικτή μια συνεχής βελτίωση όσο περισσότερο εξασκούμε και προβληματίζομαι στην εύρεση αποτελεσματικότερου κώδικα για κάποια διεργασία.

Ακόμα απέκτησα μια πρώτη εμπειρία στην επικοινωνία με τον πελάτη και τους υποψήφιους χρήστες της εφαρμογής που με βοήθησε να κατανοήσω την δυσκολία συνδυασμού πολλών λειτουργιών που να καλύψουν ανάγκες διαφορετικών χρηστών ταυτόχρονα.

Από την πλευρά της εταιρίας με φίλτρα καθαρισμού νερού για την οποία κατασκευάστηκε η εφαρμογή, κατάφερε να καλύψει τις άμεσες ανάγκες για καλύτερη οργάνωση των υπηρεσιών μετά την πώληση. Πιο συγκεκριμένα έχει καλύτερη οργάνωση των στοιχείων των πελατών και των συντηρήσεων που έχουν γίνει και που θα γίνουν, σε σχέση με τις μεθόδους που χρησιμοποιούσε μέχρι σήμερα. Στο προσωπικό του τμήματος πωλήσεων είναι πλέον πολύ εύκολη η διαδικασία καταχώρισης πελατών με τα προϊόντα που αγοράζουν με γρήγορο και αποτελεσματικό τρόπο. Στο τεχνικό τμήμα υπάρχει πλέον η δυνατότητα αυτόματης δημιουργίας λίστας με τις εργασίες που πρέπει να εκτελέσουν, τα στοιχεία των πελατών ώστε να επικοινωνήσουν με αυτούς, αλλά και να τους υπενθυμίσουν τι είχαν κάνει στην τελευταία επίσκεψη τους στον πελάτη αυτό. Άρα με λίγα λόγια η εταιρία καταφέρνει να βελτιώσει τις υπηρεσίες που παρέχει στους πελάτες της, να μειώσει το κόστος λειτουργίας της και να αυξήσει την αποδοτικότητα του προσωπικού της.

Η χρήση αυτής της εφαρμογής θα μπορούσε επίσης να καλύψει ανάγκες διαφορετικών εταιριών που ασχολούνται με προϊόντα που απαιτούν συστηματική συντήρηση και έχουν και μοναδικά χαρακτηριστικά για τον κάθε ένα από τους πελάτες που τα αγοράζει.



## 7.2 Μελλοντική εργασία και επεκτάσεις

Μελλοντικά σχέδια για την βελτίωση της εφαρμογής υπάρχουν αρκετά, θα διαμορφωθούν όμως ανάλογα τις ανάγκες και απαιτήσεις που μεταβάλλονται και αυτές με την πάροδο του χρόνου.

Μια μελλοντική επέκταση της εφαρμογής, είναι η προσθήκη διαχείρισης φωτογραφιών των συσκευών του κάθε πελάτη. Έτσι ώστε με μια ματιά να μπορεί ο τεχνικός να γνωρίζει την ακριβή τοποθεσία, τον τρόπο, τόπο εγκατάστασης και να προετοιμαστεί κατάλληλα.

Θα μπορούσα ακόμα να διαμορφώσω λίγο την διεπαφή της εφαρμογής ώστε να μειωθεί ο χρόνος διαχείρισης δεδομένων. Για παράδειγμα, να προσθέσω κάποιο μικρό κουμπί «+» και πατώντας το να μην ανοίγει νέο παράθυρο προσθήκης δεδομένων, αλλά να πληκτρολογεί ο χρήστης απευθείας στον πίνακα τα δεδομένα.

Μία ακόμα χρήσιμη λειτουργία που θα μπορούσε να προστεθεί, είναι η δυνατότητα επιλογής βάσης δεδομένων κατά το άνοιγμα της εφαρμογής. Σε περίπτωση που υπάρχουν περισσότερα από ένα αντικείμενα εργασίας της επιχείρησης, δηλαδή κάτι διαφορετικό από τα συστήματα φίλτρων. Έτσι για κάθε ξεχωριστό αντικείμενο που επίσης απαιτεί συστηματική συντήρηση και θέλουμε να το διαχωρίσουμε, θα έχουμε μια διαφορετική βάση. Αυτό φυσικά απαιτεί και άλλες αλλαγές στην εφαρμογή για μια γενικότερη χρήση (αλλαγές σε ετικέτες, κάποιους πίνακες κλπ).

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Cadenhead Rogers, Lemay Laura «Πλήρες Εγχειρίδιο της Java 6» (Α. Γκιούρδα & ΣΙΑ ΟΕ 2007)
2. Grealier, Todd «Java Εισαγωγή στη σύγχρονη τεχνολογία» (Γκιούρδας Μ. 2005).
3. Λιακέας Γ. «Εισαγωγή στην Java 2» (ΚΛΕΙΔΑΡΙΘΜΟΣ 2003)
4. Silberschatz, Korth, Suda «Συστήματα Βάσεων Δεδομένων» (Α. Γκιούρδα & ΣΙΑ ΟΕ 2004)
5. C.J. DATE «Εισαγωγή στα συστήματα βάσεων δεδομένων» Τόμος Α (ΚΛΕΙΔΑΡΙΘΜΟΣ 2008).
6. Dubois, Paul, Hinz, Stefan, Pedersen, Carsten « Ο επίσημος οδηγός MySQL 5.0» (ΓΚΙΟΥΡΔΑΣ Μ 2006)
7. Δημοσθένης Ακουμιανάκης «ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ - ΥΠΟΛΟΓΙΣΤΗ: ΜΙΑ ΣΥΓΧΡΟΝΗ ΠΡΟΣΕΓΓΙΣΗ» (ΚΛΕΙΔΑΡΙΘΜΟΣ 2006).
8. IAN SOMMERVILLE «ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ» (ΚΛΕΙΔΑΡΙΘΜΟΣ 2009).

### Πηγές από το Διαδίκτυο:

1. <http://java.sun.com/docs/books/tutorial/>
2. <http://java.sun.com/j2se/docs/api/index.html>
3. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
4. <http://javalessons.com>
5. <http://www.mysql-tools.com/en/mysql-lessons.html>
6. <http://www.mysql.com/training/>
7. <http://download.oracle.com/javase/tutorial/>
8. <http://www.it.uom.gr/project/java/tutorial.htm>
9. <http://www.jhug.gr/> (Java Hellenic User Group)

# ΠΑΡΑΡΤΗΜΑ

**ΠΑΡΑΡΤΗΜΑ Α : Εγκαταστάσεις εργαλείων ανάπτυξης  
εφαρμογής**

## Παράρτημα Α1: Εγκαθιστώντας τη java σε Windows

Εκδόσεις της Java σε διαφορετικά στάδια ολοκλήρωσης διατίθενται από τη Sun για Windows, Unix και MacOS. Μέχρι στιγμής δεν υπάρχουν εκδόσεις της Java για τα MIPS, Alpha or PowerPC based NT, Windows , Amiga. Το βασικό περιβάλλον της Java αποτελείται από έναν web browser, ο οποίος μπορεί να εκτελεί τις μίνι εφαρμογές της Java, έναν compiler που μετατρέπει τον πηγαίο κώδικα της Java σε κώδικα byte, κι έναν μεταφραστή της Java για να εκτελεί τα προγράμματα.

Αυτά είναι τα τρία συστατικά-κλειδιά ενός περιβάλλοντος Java. Επίσης απαραίτητος είναι ένας text editor όπως το Brief ή το BBEdit. Η Sun διαθέτει το Java Developers Kit (JDK). Περιέχει έναν applet viewer όπου θα μπορούμε να δούμε και να ελέγξουμε τις εφαρμογές μας. Το JDK περιλαμβάνει επίσης τον javac compiler, τον java interpreter, τον javaprof profiler, τον Java debugger και περιορισμένα κείμενα. Τα περισσότερα από τα κείμενα για το API και τη βιβλιοθήκη κλάσης είναι στο web site της Sun.

### Οδηγίες εγκατάστασης σε Windows

Θα χρειαστούμε περίπου 6 MB ελεύθερα στο δίσκο για την εγκατάσταση του JDK. Εκτελούμε το αρχείο κάνοντας διπλό κλικ πάνω στο File Manager ή επιλέγοντας Run... από το Program Manager's File menu και πληκτρολογώντας το μονοπάτι στο αρχείο. Προτείνεται να το εγκαταστήσουμε στο C:drive. Σ' αυτήν την περίπτωση τα αρχεία θα βρίσκονται στο C:\java. Θα πρέπει να προσθέσουμε το C:\java\bin directory στο path environment.

Η αρχειοθέτηση περιλαμβάνει δύο κοινά DLL's:

- MSVCRT20.DLL
- MFC30.DLL

Αυτά τα 2 αρχεία θα εγκατασταθούν στο java directory. Αν δεν έχουμε ήδη αντίγραφα αυτών στο σύστημά μας, τα αντιγράφουμε στο C:\java\bin directory. Αν τα έχουμε απλά διαγράφουμε τα επιπλέον αντίγραφα.

## Παράρτημα Α2: Εφαρμογή του NetBeans

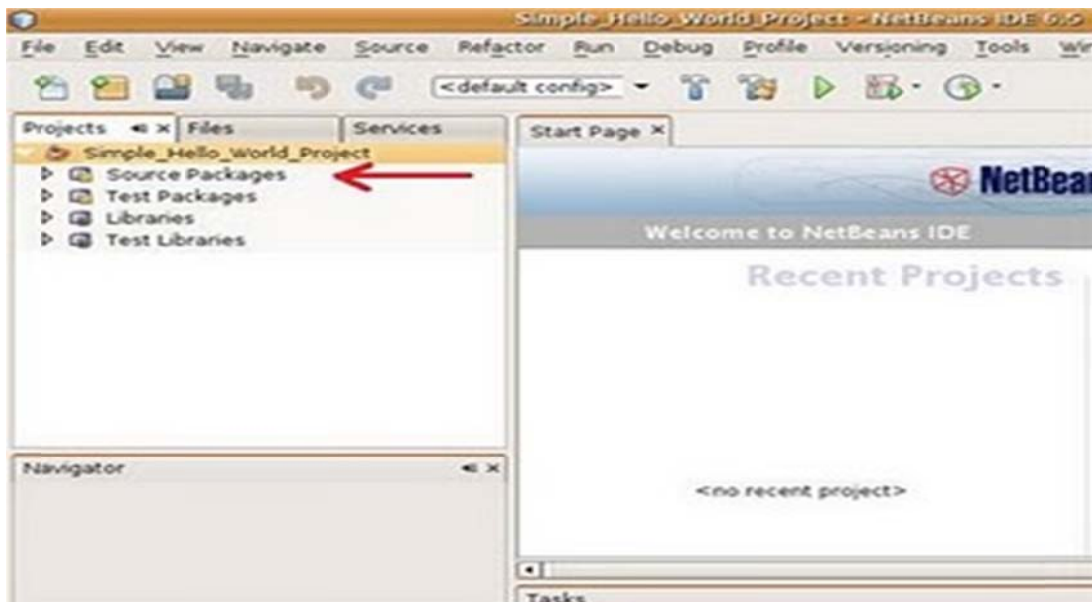
Το NetBeans δουλεύει κάτω από την έννοια του project. Δηλαδή πριν ξεκινήσουμε την δημιουργία του προγράμματος μας, πρέπει να ορίσουμε ένα project μέσα στο οποίο θα περιέχονται όλα τα αρχεία που συσχετίζονται με την δημιουργία της εφαρμογής. Έτσι, κατά την διαδικασία του compilation, όσα αρχεία συνδέονται με το συγκεκριμένο project θα συμπεριληφθούν στο τελικό αρχείο που θα δημιουργηθεί και θα έχει την κατάληξη jar. Το jar είναι το εκτελέσιμο τελικό πρόγραμμα. Η java δεν προσφέρει καταλήξεις .exe στα προγράμματα της. Αυτό όμως δεν σημαίνει ότι δεν έχουν και την ικανότητα αυτόνομης εκτέλεσης.

Όσο αφορά όμως το πρακτικό κομμάτι αυτής της διαδικασίας, με την δημιουργία ενός project, έχουμε την αντίστοιχη δημιουργία ενός φακέλου με το ίδιο όνομα στο σκληρό μας δίσκο. Κάνουμε λοιπόν κλικ στην επιλογή **File** από το κύριο μενού, επιλέγουμε **New Project** και στο καινούργιο παράθυρο που εμφανίζεται επιλέγουμε το φάκελο **Java** από τα **“Categories”**, και **Java Application** από τα **“Projects”**, και μετά συνεχίζουμε με το κουμπί **next**.

Στο επόμενο παράθυρο που εμφανίζεται έχουμε να κάνουμε κάποιες τελευταίες επιλογές πριν δημιουργηθεί ο χώρος (Project) μέσα στον οποίο θα περικλείεται το πρόγραμμά μας. Στο **Project Name** γράφουμε το όνομα που επιθυμούμε να δώσουμε στο Project μας. Δεν υπάρχουν περιορισμοί για το πως θα το ονομάσουμε, όμως αν δώσουμε ονόματα όπως Application\_1, Application\_2 ή Project\_1 και Project\_2 αργότερα δεν θα θυμόμαστε τι ακριβώς κάνει κάθε μια από τις εφαρμογές μας. Το project name να αντιπροσωπεύει ξεκάθαρα το σκοπό για τον οποίο δημιουργήθηκε η εφαρμογή. Από το **Project Location** έχουμε την δυνατότητα να αλλάξουμε την προκαθορισμένη τοποθεσία στο σκληρό δίσκο στην οποία θα αποθηκευτεί το project με όλα τα αρχεία του. Ενώ από το **Project Folder** ενημερωνόμαστε ότι ο φάκελος στον οποίο μπορούμε να αναφερόμαστε για να αντιγράψουμε, ή να αποθηκεύσουμε τα αρχεία μας θα έχει το ίδιο όνομα όπως και το project.

Πριν συνεχίσουμε με το κουμπί **FINISH**, βεβαιωνόμαστε ότι οι επιλογές **“Use Dedicated Folder for Storing Libraries”**, **“Create Main Class”**, και **“Set as Main Project”** δεν είναι επιλεγμένες, όπως δείχνει η παρακάτω εικόνα. Ο λόγος που αυτές τις τρεις επιλογές δεν τις θέλουμε ενεργοποιημένες είναι γιατί στα πρώτα βήματα του προγραμματισμού, δεν θέλουμε την βοήθεια από προκαθορισμένα templates.

Όταν δημιουργηθεί το Project με το όνομα **“Simple\_Hello\_World\_Project”**, θα εμφανιστούν τέσσερις υποκατηγορίες εκ τις οποίες η μία ονομάζεται Source Packages. Μέσα σε αυτή την κατηγορία θα δημιουργήσουμε τα πακέτα μας στα οποία θα περικλείονται τα αρχεία της εφαρμογής μας. Τα πακέτα χρησιμεύουν για την κατηγοριοποίηση και ομαδοποίηση αρχείων που εξυπηρετούν συγκεκριμένο σκοπό μέσα στην εφαρμογή μας. Για παράδειγμα, μέσα σε ένα πακέτο μπορούμε να έχουμε τις κλάσεις (αρχεία) που αφορούν την ένωση του προγράμματος μας με μια βάση δεδομένων, ενώ σε ένα δεύτερο πακέτο να έχουμε τις κλάσεις που αφορούν το γραφικό μέρος της εφαρμογής μας. Πρακτικά, στο σκληρό μας δίσκο η δημιουργία ενός πακέτου αντιστοιχεί με την δημιουργία ενός φακέλου με το ίδιο όνομα. Στην παρακάτω εικόνα βλέπουμε πως θα είναι το περιβάλλον του NetBeans μετά την δημιουργία ενός project.



**Εικόνα Α2.α : Περιβάλλον του NetBeans μετά την δημιουργία ενός project.**

Αναφέραμε πιο πάνω, ότι ένα πρόγραμμα, θα πρέπει να βρίσκεται μέσα σε ένα πακέτο για να μπορεί εκτελεστεί. Επίσης ένα πακέτο είναι δυνατόν να περιέχει περισσότερα από ένα αρχείο και η δημιουργία ενός πακέτου ισοβαθμεί με την δημιουργία φακέλου μέσα στον ήδη υπάρχων φάκελο του project. Οπότε εάν θέλουμε να έχουμε πάντα ένα αντίγραφο του προγράμματός μας, απλά αντιγράφουμε το φάκελο του project σε κάποιο αποθηκευτικό μέσο (flash drive ή hard drive).

Είμαστε αναγκασμένοι να δημιουργήσουμε τουλάχιστον ένα πακέτο για να μπορούμε να τρέξουμε την εφαρμογή μας. Για να δημιουργήσουμε ένα πακέτο, επιλέγουμε δεξί κλικ στο εικονίδιο **Source Packages**, επιλέγουμε **New**, και μετά **Java Package**.

Στο νέο παράθυρο που ανοίγει, μας ζητείται να δώσουμε ένα όνομα στο πακέτο που πρόκειται να δημιουργήσουμε. Αν και για το πως θα ονομάσουμε τα πακέτα δεν υπάρχει κανένας περιορισμός, ας τα ονομάζουμε με το ίδιο όνομα όπως την java εκτελέσιμη κλάση που περιέχουν.

Ονομάζουμε το πακέτο μας "**Hello\_World**" (με underscore ανάμεσα στις δύο λέξεις) και αμέσως μετά συνεχίζουμε με το κουμπί **FINISH**.

Επιλέγοντας τώρα δεξί κλικ το πακέτο Hello\_World, επιλέγουμε **New**, και μετά **Java class**. Στην επιλογή **Class Name** γράφουμε το όνομα του προγράμματος το οποίο είναι Hello\_World (χωρίς κενό ανάμεσα στις δύο λέξεις). Τα ονόματα ανάμεσα στην κλάση και στο πακέτο δεν είναι απαραίτητο να είναι ακριβώς τα ίδια. Όμως υπάρχει μια μεγάλη διαφορά – Ενώ το NetBeans αδιαφορεί για το πως θα γράψουμε το όνομα του πακέτου, δίνει όμως μεγάλη βαρύτητα στον τρόπο εγγραφής του ονόματος της κλάσης.

Η Java είναι "ευαίσθητη" με τα πεζά και κεφαλαία γράμματα. Η λέξη Hello\_World δεν είναι η ίδια με βάση τους κανόνες της java με την λέξη hello\_world. Το όνομα που θα δώσουμε στην κλάση πρέπει να συμφωνεί στην γραφή με το όνομα το οποίο θα πρέπει να ξεκινήσουμε τον κώδικα μας. Όπως γράφουμε το όνομα του προγράμματος στο παράθυρο, έτσι πρέπει να το γράψουμε και μέσα στο κώδικα μας. Ο λόγος για αυτό τον αυστηρό κανόνα είναι απλός – το αρχείο θα αποθηκευτεί με το ίδιο όνομα που θα δώσουμε στην κλάση. Αφήνουμε τις υπόλοιπες επιλογές όπως είναι και συνεχίζουμε με το **FINISH**.

Κάποιες βασικές γραμμές κώδικα που εμφανίζονται στα δεξιά της οθόνης μας, είναι απαραίτητες για οποιοδήποτε πρόγραμμα java και αν γράψουμε. Απλά το NetBeans θέλει να βοηθήσει τον προγραμματιστή ετοιμάζοντας το σκελετό του προγράμματος εξοικονομώντας έτσι χρόνο και μειώνοντας την πιθανότητα λάθους – το λάθος εδώ μπορεί να γίνει όταν γράψουμε το όνομα της κλάσης μέσα στο κώδικα μας με διαφορετικό όνομα από αυτό που δώσαμε κατά την δημιουργία της. Αυτή η πρωτοβουλία που παίρνει το περιβάλλον του Netbeans μας προφυλάσσει από πολλά προβλήματα που μπορεί να αντιμετωπίσουμε αργότερα. Δεν είναι ανάγκη να διαγράψουμε καμία από αυτές τις γραμμές κώδικα. Εμείς απλά τώρα μέσα στην κλάση Hello\_World θα συμπληρώσουμε τις γραμμές κώδικα που θα κάνουν αυτή την κλάση εκτελέσιμη (γιατί από μόνη της δεν είναι) και επίσης να ορίσουμε μια απλή μέθοδο η οποία θα μας δείξει ένα σύντομο μήνυμα στην οθόνη μας.

```
public class Hello_World  
  
{  
  
public static void main(String [ ] args)  
  
{  
  
System.out.println("Hello, World");  
  
}  
  
}
```

Το ολοκληρωμένο πρόγραμμα παρουσιάζεται στην παρακάτω εικόνα.

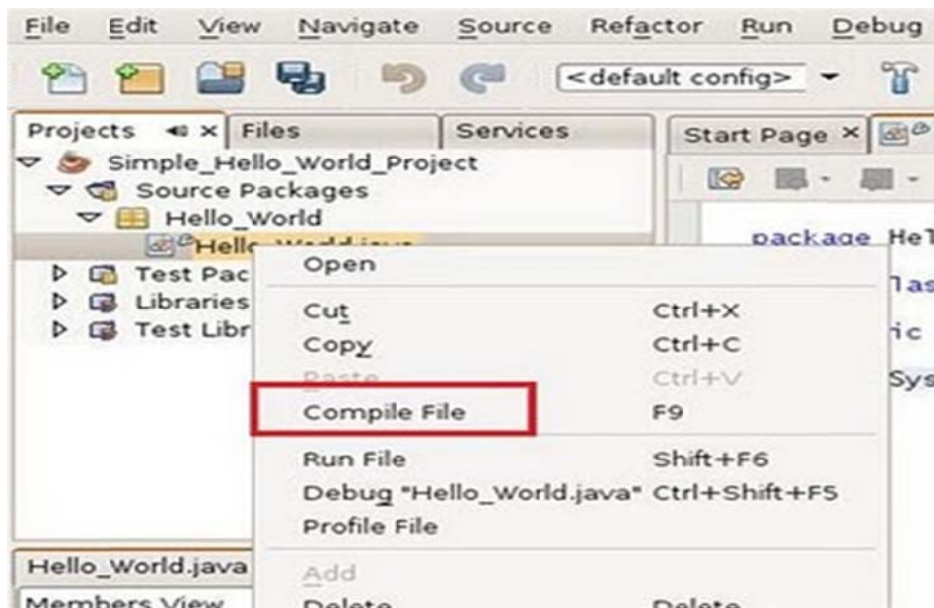


**Εικόνα Α2.β : Παράθυρο σύνταξης κώδικα.**

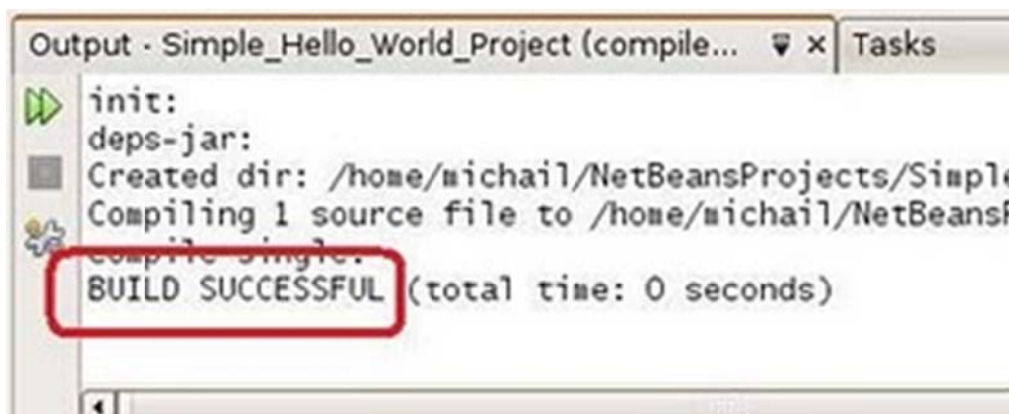
Αν κοιτάξουμε στο αριστερό μέρος της οθόνης μας, κάτω από το πακέτο Hello\_World, υπάρχει ένα αρχείο Hello\_World.java το οποίο συνοδεύεται από ένα μικρό εικονίδιο που μοιάζει σαν μηχανικό κλειδί. Αυτό σημαίνει ότι το πρόγραμμα μας γράφτηκε, όμως δεν έχει ελεγχθεί επίσημα για λάθη από τον Compiler της Java. Πριν όμως προχωρήσουμε σε αυτή την διαδικασία πάμε στο κύριο μενού Window και από την επιλογή Output να επιλέξουμε πάλι Output. Τώρα στο κάτω μέρος της οθόνη μας έχουμε ένα καινούργιο πλαίσιο ανοιχτό μέσα από το οποίο μπορούμε να βλέπουμε τα αποτελέσματα των προγραμμάτων μας ή ακόμα τυχόν λάθη για τα οποία ο compiler έχει διαμαρτυρηθεί.



Επιλέγοντας λοιπόν με δεξί κλικ το java αρχείο και από τις επιλογές που εμφανίζονται, επιλέγουμε το **Compile File**. Αν δεν υπάρχουν συντακτικά λάθη στο πρόγραμμα μας, τότε στο πλαίσιο Output θα παρατηρήσουμε ότι η διαδικασία ελέγχου ήταν επιτυχής ( BUILD SUCCESSFUL) και το πρόγραμμα μας είναι άριστα γραμμένο.



Εικόνα Α2.γ : Επιλογή Compile του προγράμματος.



Εικόνα Α2.δ : Παράθυρο αποτελέσματος ελέγχου.

Το γεγονός ότι το πρόγραμμά μας έγινε compile επίσης συμβολίζει την δημιουργία και του αρχείου .class. Επειδή όμως το NetBeans θέλει να ξεχωρίσει τα δύο αρχεία, κρατάει τα αρχεία .java στο φάκελο src ενώ δημιουργεί τα αρχεία .class στο φάκελο build.



**Εικόνα Α2.ε : Φάκελος αποθηκευμένων αρχείων.**

Αν γυρίσουμε πίσω στην κύρια οθόνη του NetBeans, θα προσέξουμε ότι το εικονίδιο Hello\_World.java στα αριστερά της οθόνης μας, κάτω από το πακέτο Hello\_World, έχει ένα καινούργιο σύμβολο που μοιάζει σαν το play κουμπί που έχουμε συνηθίσει στις ηλεκτρικές συσκευές. Αυτό σημαίνει ότι το αρχείο java είναι έτοιμο για εκτέλεση. Αυτό το σύμβολο εμφανίζεται μόνο στις κλάσεις που περιέχουν την μέθοδο main - την μέθοδο που μετατρέπει την κλάση σε εκτελέσιμη. Ο τρόπος που θα πραγματοποιήσουμε αυτή την ενέργεια είναι να κάνουμε δεξί κλικ για ακόμα μια φορά επάνω στο αρχείο, αλλά αυτή τη φορά θα επιλέξουμε Run File. Το αποτέλεσμα της εκτέλεσης του προγράμματος ("Hello, World") θα το δούμε στο παράθυρο output.

## Παράρτημα Α3: Εγκατάσταση του NetBeans σε Windows

Μετά την ανάλυση γύρω από το JVM (Java Virtual Machine) είμαστε έτοιμοι να εγκαταστήσουμε το περιβάλλον του NetBeans μέσα από το οποίο θα αναπτύξουμε την εφαρμογή μας. Το πρώτο και απαραίτητο βήμα είναι να κατεβάσουμε από την Sun Microsystems το JDK πακέτο το οποίο περιέχει τον Java compiler μαζί με το γραφικό περιβάλλον του NetBeans (εάν έχουμε ήδη το JDK κατεβάζουμε μόνο το NetBeans) . Το πλήρες πρόγραμμα θα το βρούμε στην ακόλουθη ηλεκτρονική διεύθυνση: <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>

Επιλέγοντας την επιλογή **DOWNLOAD** (στα δεξιά της οθόνης), θα μεταφερθούμε σε μια νέα σελίδα που μας ζητάει με απλό τρόπο να ορίσουμε την πλατφόρμα στην οποία πρόκειται να εγκαταστήσουμε το NetBeans περιβάλλον και σε ποια γλώσσα επιθυμούμε να παρουσιάζονται τα μενού του συγκεκριμένου προγράμματος. Πριν επιλέξουμε το κουμπί **CONTINUE** θα πρέπει να συμφωνήσουμε με τους όρους άδειας που συνοδεύουν το πρόγραμμα. Αυτό πραγματοποιείται επιλέγοντας το κενό κουτάκι δίπλα από το “I agree to...”.

Στο επόμενο παράθυρο που θα εμφανιστεί, απλά επιλέγουμε τον κόκκινο σύνδεσμο (link) και όταν εμφανιστεί η επιλογή αποθήκευσης από το browser, επιλέγουμε μια τοποθεσία στο σκληρό μας δίσκο στο οποίο θα αποθηκευτεί το αρχείο.

Πηγαίνουμε στο φάκελο μέσα στο οποίο βρίσκεται αποθηκευμένο το αρχείο και ξεκινάμε την διαδικασία εγκατάστασης του προγράμματος κάνοντας διπλό κλικ επάνω στο εικονίδιο.

Το πρώτο παράθυρο που θα εμφανιστεί σχεδόν αμέσως μετά από την ενεργοποίηση του προγράμματος μας ειδοποιεί ότι στα επόμενα βήματα που θα ακολουθήσουν θα εγκατασταθούν δύο προγράμματα – το Java SE Development Kit (JDK) και αμέσως μετά το περιβάλλον του NetBeans. Σε αυτό το παράθυρο απλά επιλέγουμε το **NEXT**.

Η διαδικασία εγκατάστασης μας ενημερώνει στο αμέσως επόμενο παράθυρο ότι θα δημιουργηθεί ένας φάκελος με το όνομα Java κάτω από το directory Program Files των Windows μέσα στον οποίο θα γίνει εγκατάσταση το JDK. Δεν έχουμε κανένα απολύτως λόγο να διαφωνήσουμε με αυτή την προκαθορισμένη επιλογή και απλά επιλέγουμε το κουμπί **NEXT** για να συνεχιστεί η διαδικασία.

Μετά από το JDK, σειρά έχει το πρόγραμμα NetBeans να δημιουργήσει ένα φάκελο στον υπολογιστή μας μέσα στον οποίο θα εγκαταστήσει τα απαραίτητα αρχεία του. Παρατηρούμε ότι το NetBeans χρειάζεται απαραίτητα το JDK για να ολοκληρώσει την εγκατάσταση του και αυτός είναι άλλωστε ο λόγος που η δεύτερη γραμμή παρουσιάζει το directory μέσα στο οποίο θα μπορέσει να το βρει. Αυτό σημαίνει ότι αν πάμε στην ιστοσελίδα του NetBeans (<http://www.netbeans.org>) και κατεβάσουμε το Netbeans, για να μπορέσουμε να το εγκαταστήσουμε θα πρέπει να υπάρχει ήδη μια εγκατάσταση του JDK στον υπολογιστή μας. Το NetBeans δεν προσφέρεται σαν πακέτο από το δικό του site για αυτό άλλωστε και επιλέξαμε την ORACLE για να κάνουμε μια ολοκληρωμένη και χωρίς προαπαιτήσεις εγκατάσταση του προγράμματος. Στην δική μας περίπτωση και τα δύο προγράμματα βρίσκονται στο ίδιο πακέτο εγκατάστασης και η σειρά έχει αυτόματα οριστεί. Δεχόμαστε λοιπόν και αυτή την επιλογή και συνεχίζουμε με το κουμπί **NEXT**.

Αμέσως μετά θα εμφανιστεί μια σύντομη περίληψη της τοποθεσίας των φακέλων στους οποίους θα εγκατασταθούν τα αρχεία των δύο προγραμμάτων. Απλά επιλέγουμε **INSTALL** για να ολοκληρωθεί η εγκατάσταση.

Στο τελευταίο παράθυρο εγκατάστασης του NetBeans, απλά ενημερωνόμαστε για το επιτυχές τέλος της διαδικασίας και κατά πόσο θέλουμε να συνεισφέρουμε στο NetBeans project αφήνοντας την εταιρεία να μαζεύει κάποια στατιστικά στοιχεία τα οποία θα αξιοποιήσει για την περαιτέρω βελτίωση του προγράμματος.

Αφού αποφασίσουμε ποιες επιλογές θέλουμε να υποστηρίξουμε (όλες είναι προαιρετικές), απλά συνεχίζουμε με το κουμπί **FINISH** για να επισημοποιήσουμε την ολοκλήρωση της εγκατάστασης. Στην επιφάνεια εργασίας του υπολογιστή μας, θα πρέπει τώρα να εμφανίζεται ένα εικονίδιο σε σχήμα κύβου με το όνομα NetBeans.



**Εικόνα Α3.α : Εικονίδιο του εργαλείου NetBeans.**

Κάνοντας διπλό κλικ επάνω στο εικονίδιο του NetBeans, θα βρεθούμε στο περιβάλλον εργασίας του.

## Παράρτημα Α4: Εγκατάσταση του NetBeans σε Ubuntu

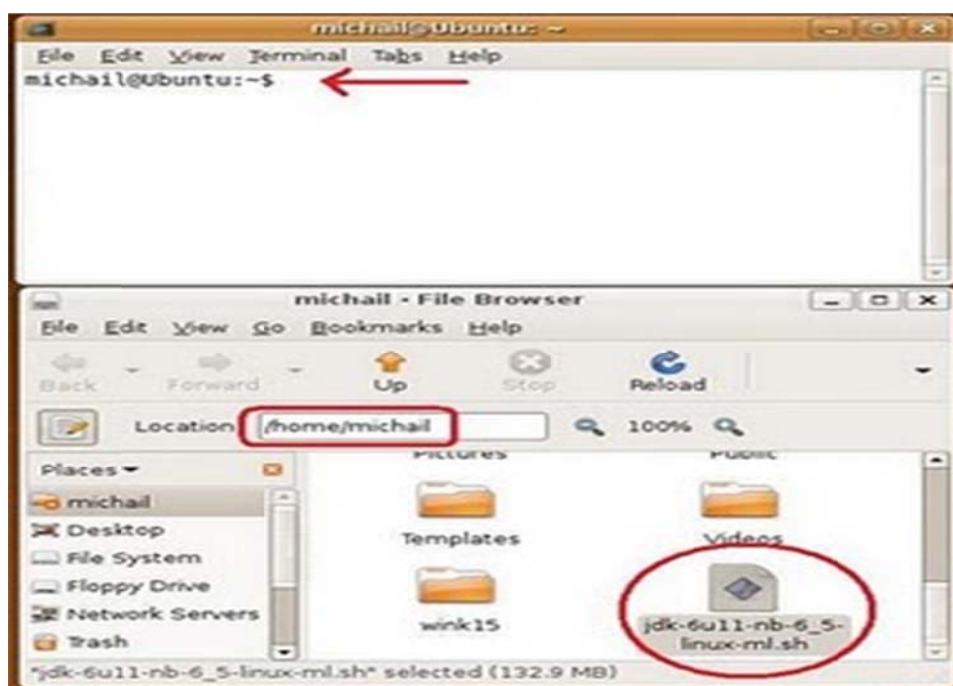
Κατεβάσουμε από την Sun Microsystems το JDK πακέτο το οποίο παρέχει τον java compiler μαζί με το γραφικό περιβάλλον του NetBeans. Το πλήρες πακέτο θα το βρούμε στην ακόλουθη ηλεκτρονική διεύθυνση: <http://java.sun.com/javase/downloads/netbeans.html>

Πατώντας την επιλογή **DOWNLOAD** θα μεταφερθούμε σε μια καινούργια σελίδα που μας ζητάει με απλό τρόπο να ορίσουμε την πλατφόρμα στην οποία πρόκειται να εγκαταστήσουμε το NetBeans και σε ποια γλώσσα επιθυμούμε να παρουσιάζονται τα μενού του συγκεκριμένου προγράμματος. Πριν επιλέξουμε το κουμπί **Continue**, θα πρέπει να συμφωνήσουμε με τους όρους της άδειας για το συγκεκριμένο πρόγραμμα επιλέγοντας μέσα στο κενό κουτάκι δίπλα από το “I agree...”. Έχουμε επιλέξει για πλατφόρμα ανάπτυξης προγραμμάτων το λειτουργικό LINUX και σαν γλώσσα τα αγγλικά (περιέχεται μέσα στο Multi-language πακέτο).

Στην επόμενη ιστοσελίδα που μας εμφανίζεται, απλά επιλέγουμε το μπλε link του πακέτου jdk και όταν εμφανιστεί η επιλογή αποθήκευσης του browser, επιλέγουμε την τοποθεσία στο σκληρό δίσκο στο οποίο θα αποθηκεύσουμε το αρχείο.

Για τους χρήστες που δεν γνωρίζουν την διαδικασία εγκατάστασης ενός προγράμματος σε περιβάλλον LINUX, μπορούν να δουν τα απαραίτητα βήματα από τις πιο κάτω εικόνες.

Σαν πρώτο βήμα, τοποθετούμε το αρχείο που κατεβάσαμε στο home folder (δηλαδή στο φάκελο με το όνομα μας). Έτσι θα είναι ευκολότερο για μας να το βρούμε. Σαν δεύτερο βήμα ανοίγουμε το πρόγραμμα Terminal από το Accessories μενού. Η πιο κάτω εικόνα μας δείχνει πως θα πρέπει να μοιάζει η οθόνη μας με τα δύο προγράμματα ενεργοποιημένα.



Εικόνα Α4.α : Παράθυρα Terminal και Accessories μενού.

Για να μπορέσουμε να δώσουμε την εντολή εγκατάστασης, θα πρέπει πρώτα να αλλάξουμε τα δικαιώματα κάτω από τα οποία ορίζεται η πρόσβαση μας στο αρχείο.



```
michail@Ubuntu: ~  
File Edit View Terminal Tabs Help  
michail@Ubuntu:~$ chmod +x jdk-6u11-nb-6_5-linux-ml.sh  
michail@Ubuntu:~$
```

**Εικόνα A4.β : Διαδικασία αλλαγής δικαιωμάτων.**

Τέλος μπορούμε να καλέσουμε την εντολή εγκατάστασης.



```
michail@Ubuntu: ~  
File Edit View Terminal Tabs Help  
michail@Ubuntu:~$ chmod +x jdk-6u11-nb-6_5-linux-ml.sh  
michail@Ubuntu:~$ ./jdk-6u11-nb-6_5-linux-ml.sh
```

**Εικόνα A4.γ: Παράθυρο Terminal του Ubuntu.**

Αμέσως μετά από την εκτέλεση της εντολής θα εμφανιστεί το πρώτο παράθυρο που μας ενημερώνει ότι πρόκειται να γίνει εγκατάσταση του JDK (δηλαδή του compiler της java), αλλά και του περιβάλλοντος Netbeans. Για να μπορέσουμε να συνεχίσουμε την εγκατάσταση του προγράμματος, δεχόμαστε το μήνυμα με το κουμπί NEXT.

Στο αμέσως επόμενο παράθυρο, πρέπει να συμφωνήσουμε με τους όρους της άδειας κάτω από τους οποίους λειτουργεί το συγκεκριμένο πρόγραμμα. Τσεκάρουμε το κουτί δίπλα από το “I accept the ...” και επιλέγουμε NEXT.

Η διαδικασία εγκατάστασης μας ενημερώνει ότι θα πρέπει να δημιουργήσει ένα φάκελο στο home folder του λογαριασμού μας για να μπορέσει να πραγματοποιήσει το πρώτο βήμα που είναι η εγκατάσταση της java. Επιλέγουμε το κουμπί NEXT για να προχωρήσουμε στο επόμενο βήμα.

Μετά από την java, είναι η σειρά του Netbeans να δημιουργήσει ένα φάκελο στο σύστημα μας μέσα στον οποίο θα εγκατασταθεί. Παρατηρούμε ότι το Netbeans χρειάζεται απαραίτητως το JDK για να ολοκληρώσει την εγκατάστασή του. Αυτό σημαίνει ότι αν πάμε στο <http://www.netbeans.org> και προμηθευτούμε μόνοι μας το Netbeans, για να μπορέσουμε να το εγκαταστήσουμε θα πρέπει να υπάρχει ήδη μια εγκατάστασή της java JDK στον υπολογιστή μας. Χωρίς αυτή την συνθήκη, και επειδή το Netbeans θα το αναζητήσει και δεν θα το βρει, θα ακυρωθεί η διαδικασία εγκατάστασης. Στην δική μας περίπτωση όμως και τα δύο προγράμματα βρίσκονται στο ίδιο πακέτο και η σειρά έχει αυτόματα οριστεί. Δεχόμαστε λοιπόν και αυτή την επιλογή και συνεχίζουμε με NEXT.

Μια σύντομη περίληψη της τοποθεσίας των φακέλων που θα γίνουν εγκατάστασή του JDK και το Netbeans εμφανίζεται στο επόμενο παράθυρο. Απλά επιλέγουμε INSTALL για να αρχίσει η διαδικασία εγκατάστασης του πακέτου.

Στο τελευταίο παράθυρο της διαδικασίας αυτής απλά ενημερωνόμαστε για το επιτυχές τέλος της εγκατάστασης και κατά πόσο θέλουμε να συνεισφέρουμε στο Netbeans project αφήνοντας την εταιρεία να μαζεύει κάποια στατιστικά στοιχεία τα οποία θα αξιοποιήσει για την βελτίωση του προγράμματος.

Αφού αποφασίσουμε, απλά επιλέγουμε το κουμπί FINISH για να επισημοποιήσουμε την ολοκλήρωση εγκατάστασης. Στην επιφάνεια εργασίας του υπολογιστή μας, θα πρέπει τώρα να εμφανίζεται ένα εικονίδιο με το όνομα του Netbeans.

Εδώ ολοκληρώνεται η διαδικασία εγκατάστασης του JDK και του περιβάλλοντος ανάπτυξης Java εφαρμογών Netbeans.

## Παράρτημα A5: Downloading MySQL

Το πρώτο βήμα για την εγκατάσταση του MySQL για Windows είναι να κατεβάσουμε την τελευταία και πιο σταθερή έκδοση που είναι διαθέσιμη από το site της MySQL. Οι νέες εκδόσεις κυκλοφορούν συχνά, αλλά η πραγματική διαδικασία ρύθμισης θα πρέπει να παραμείνει η ίδια. Ανοίγουμε το πρόγραμμα περιήγησής μας και αναζητούμε το αρχείο εγκατάστασης στο <http://dev.mysql.com/downloads/mysql/>. Η οποία είναι η MySQL σελίδα για τα downloads. Από την κατηγορία Windows downloads επιλέγουμε το Windows ZIP/Setup.EXE (x86).

Για την εγκατάσταση της MySQL σε περιβάλλον Windows υπάρχουν τρία διαφορετικά πακέτα που μπορούμε να χρησιμοποιήσουμε.

Η πρώτη μας επιλογή είναι το Windows Essentials(x86) αρχείο που έχει κατάληξη .msi (η στάνταρ κατάληξη πακέτων εγκατάστασης σε λειτουργικά συστήματα της Microsoft). Αυτό περιέχει τα βασικά στοιχεία που χρειάζεται ο μέσος χρήστης σε μια βάση δεδομένων περιλαμβανομένου και ενός γραφικού Configuration Wizard που μας βοηθάει με τις επιλογές που διαθέτει να τροποποιήσουμε την εγκατάσταση όπως εμείς θέλουμε.

Το δεύτερο πακέτο (Windows ZIP/Setup.EXE(x86)) περιλαμβάνει ότι και το πρώτο, μαζί με πολλά έξτρα στοιχεία όπως μια συλλογή από εργαλεία για μετρήσεις απόδοσης της βάσης. Το αρχείο αυτό είναι σε μορφή ZIP που σημαίνει ότι πρέπει να έχουμε ένα απλό πρόγραμμα συμπίεσης / αποσυμπίεσης αρχείων εγκαταστημένο στον υπολογιστή μας (π.χ. WinZIP, WinRAR) για να μπορέσουμε να το αποσυμπιέσουμε πριν το εγκαταστήσουμε. Τα περισσότερα από αυτά τα έξτρα εργαλεία δεν θα χρειαστούν για τον απλό χρήστη ή προγραμματιστή.

Το τελευταίο πακέτο Without installer(unzip in C:\) περιλαμβάνει ότι και τα δύο προηγούμενα εκτός του Configuration Wizard. Αυτό σημαίνει ότι δεν θα έχουμε καμιά γραφική βοήθεια κατά την εγκατάσταση και θα πρέπει να γνωρίζουμε τις κατάλληλες εντολές για να μπορέσουμε να ολοκληρώσουμε επιτυχώς την διαδικασία. Και αυτό το πακέτο έχει κατάληξη ZIP οπότε και εδώ θα χρειαστεί να αποσυμπιέσουμε το αρχείο πριν το εγκαταστήσουμε.

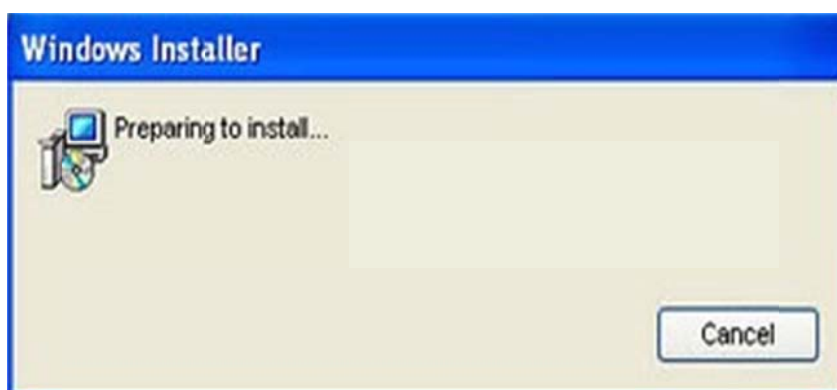
Κατεβάζουμε το αρχείο για την εγκατάσταση επιλέγοντας το σύνδεσμο.



## Παράρτημα Α6 : Εγκατάσταση της MySQL σε Windows

Αφού αποσυμπιέσουμε το ZIP αρχείο. Εκτελούμε το πρόγραμμα εγκατάστασης, κάνοντας διπλό κλικ επάνω στο Setup (εκτελούμενο αρχείο) για να αρχίσουμε την διαδικασία εγκατάστασης. Η MySQL έχει προγραμματιστεί να τρέχει σταθερά και χωρίς ιδιαίτερα προβλήματα απόδοσης σε πλατφόρμα Windows XP και Windows Server. Αν τυχόν υπάρχει κάποια προηγούμενη εγκατάσταση της MySQL στον υπολογιστή μας, πρέπει να την απεγκαταστήσουμε πριν συνεχίσουμε με τα παρακάτω βήματα. Μόνο έτσι θα μπορέσουμε να εξαλείψουμε εξ αρχής οποιαδήποτε προβλήματα που μπορεί να υπάρξουν αργότερα.

Το πρώτο παράθυρο που θα εμφανιστεί στην διαδικασία της εγκατάστασης θα είναι το παρακάτω:



Εικόνα Α6.α : Προετοιμασία εγκατάστασης.

Μετά την εμφάνιση της πρώτης εικόνας, σε λίγα δευτερόλεπτα θα εμφανιστεί το μήνυμα καλωσορίσματος της εγκατάστασης του MySQL. Σε αυτό το παράθυρο επιλέγουμε το κουμπί **NEXT**.

Έπειτα έχουμε να επιλέξουμε ανάμεσα σε τρεις επιλογές εγκατάστασης. Η πρώτη επιλογή (Typical) θα εγκαταστήσει την κύρια βάση μας με πολλά έξτρα command-line εργαλεία (δηλαδή έξτρα δυνατότητες που μπορούμε να αξιοποιήσουμε μέσα από εντολές μόνο). Αυτή είναι και η πιο συνηθισμένη μορφή εγκατάστασης για τις περισσότερες εφαρμογές για τις οποίες χρειαζόμαστε υποστήριξη από μια βάση δεδομένων.

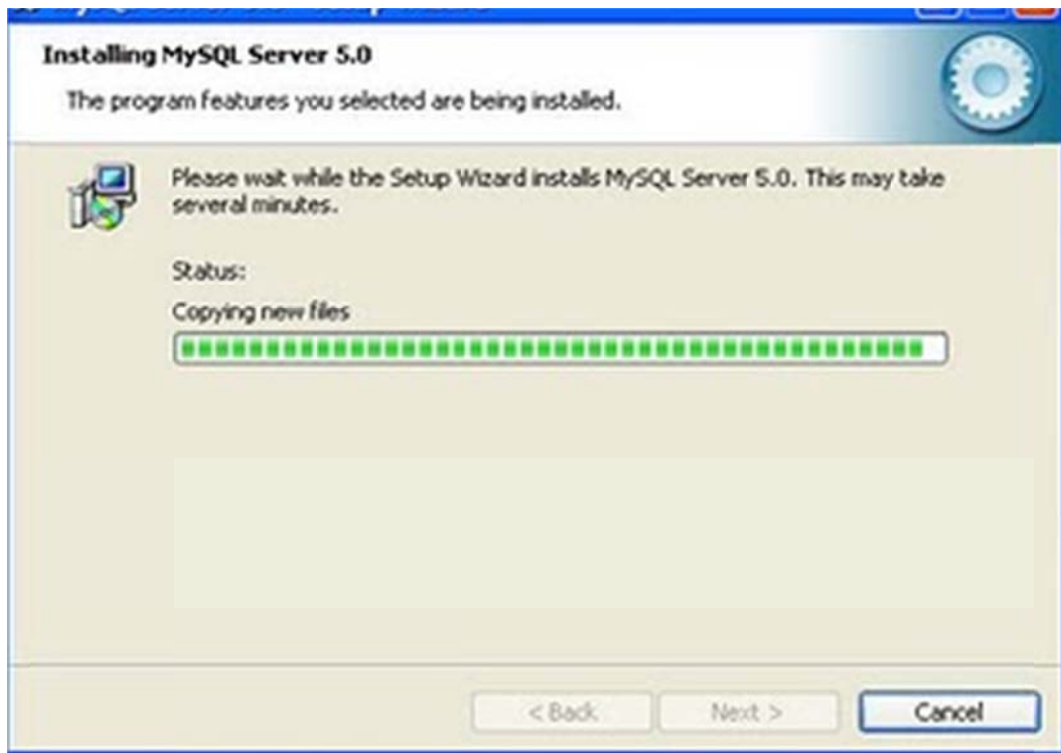
Η δεύτερη επιλογή (Complete) θα προχωρήσει σε μια πλήρη εγκατάσταση του προγράμματος η οποία περιλαμβάνει μια πληθώρα από scripts και ένα ολοκληρωμένο documentation για να μπορούμε να ανατρέξουμε ανά πάσα στιγμή όταν χρειαστούμε κάποια βοήθεια με τις εντολές.

Η τρίτη και τελευταία επιλογή (Custom) αφήνει στην κρίση μας την επιλογή των διαφόρων εργαλείων και δυνατοτήτων που θέλουμε να εγκαταστήσουμε.

Εμείς θα επιλέξουμε την δεύτερη επιλογή η οποία περιλαμβάνει την ολοκληρωμένη εγκατάσταση και θα συνεχίσουμε με το κουμπί **NEXT**.

Στο επόμενο παράθυρο το πρόγραμμα μας ειδοποιεί ότι είναι έτοιμο να ξεκινήσει την διαδικασία εγκατάστασης. Στην περιγραφή που εμφανίζεται στο παραθυρικό περιβάλλον μπορούμε να δούμε μια συνοπτική εικόνα των επιλογών μας και την τοποθεσία εγκατάστασης του προγράμματος στον σκληρό μας δίσκο. Συνιστάται να αφήσουμε αυτόν τον κατάλογο, όπως είναι η προεπιλογή. Αν το αλλάξουμε, θα πρέπει να ρυθμίσουμε τη MySQL αργότερα να επιλέγει την πορεία μας. Απλά επιλέγουμε NEXT και συνεχίζουμε.

Το επόμενο παράθυρο που θα εμφανιστεί θα είναι η διαδικασία εγκατάστασης του προγράμματος.



**Εικόνα Α6.β : Παράθυρο διαδικασίας εγκατάστασης του προγράμματος.**

Μετά την ολοκλήρωση του πρώτου μέρους της διαδικασίας εγκατάστασης, θα ενημερωθούμε για τα οφέλη που μπορούμε να αποκτήσουμε κάνοντας μια συνδρομή στο MySQL Enterprise. Για να ενημερωθούμε σχετικά με την συνδρομή επιλέγουμε το κουμπί **MORE**, ειδάλλως συνεχίζουμε με την διαδικασία ολοκλήρωσης της εγκατάστασης επιλέγοντας το κουμπί **NEXT**.

Μετά τις δυο ανακοινώσεις και τα οφέλη της συνδρομής, θα μας παρουσιαστεί ένα καινούργιο παράθυρο που μας ενημερώνει ότι η εγκατάσταση ολοκληρώθηκε και μπορούμε τώρα να ορίσουμε το port στο οποίο θα “ακούει” το πρόγραμμα όπως το password για τον administrator. Αφήνοντας λοιπόν την επιλογή όπως είναι συνεχίζουμε με το κουμπί **FINISH**. Έπειτα θα μεταφερθούμε στο επόμενο περιβάλλον της MySQL από το οποίο μπορούμε να ενεργοποιήσουμε διάφορες επιλογές για την συγκεκριμένη βάση που εγκαταστήσαμε.

Αφού δεχτούμε ένα ακόμα μήνυμα που μας ειδοποιεί ότι τώρα έχουμε την ευκαιρία να ορίσουμε τις δικές μας προτιμήσεις στο πως θα “τρέχει” η βάση, απλά επιλέγουμε το κουμπί **NEXT**.

Στο επόμενο παράθυρο θα βρεθούμε πάλι μπροστά σε δυο επιλογές. Η πρώτη (Detailed Configuration) δίνει την ευελιξία στον χρήστη να διαλέξει τις καταλληλότερες ρυθμίσεις της βάσης, ενώ η δεύτερη επιλογή (Standard Configuration) απλά δημιουργεί ένα περιβάλλον κατάλληλο για τις περισσότερες απαιτήσεις του εργασιακού περιβάλλοντος. Εμείς επιλέγουμε την πρώτη επιλογή.

Οι επόμενες τρεις επιλογές που θα εμφανιστούν είναι αρκετά επεξηγηματικές από μόνες τους. Αν διαλέξουμε την πρώτη επιλογή (Developer Machine), η MySQL βάση θα χρησιμοποιήσει όσο το δυνατόν λιγότερη μνήμη από τον υπολογιστή μας. Η επιλογή αυτή είναι κατάλληλη για προγραμματιστές που απλά χρειάζονται μια βάση δεδομένων για να δοκιμάσουν την λειτουργικότητα κάποιας εφαρμογής που αναπτύσσουν.

Η δεύτερη επιλογή (Server Machine) θα χειριστεί την MySQL σαν server που όμως υποθετικά δεν θα είναι ο μόνος που τρέχει στον συγκεκριμένο υπολογιστή. Συνήθως αυτή η επιλογή χρησιμοποιείται όταν στον ίδιο υπολογιστή υπάρχουν και άλλοι servers που έχουν εγκατασταθεί όπως π.χ. ένας Web Server (είναι υπεύθυνος για την διαχείριση και την παρουσίαση ιστοσελίδων). Ο λόγος που ίσως εγκαταστήσαμε την MySQL στον ίδιο υπολογιστή είναι γιατί πιθανόν υπάρχει μια Web εφαρμογή (π.χ. ένα ηλεκτρονικό κατάστημα e-commerce) το οποίο χρειάζεται μια βάση δεδομένων για την αποθήκευση στοιχείων όπως προϊόντα, πωλήσεις, και πελατολόγιο. Οπότε γίνεται μια καλύτερη κατανομή της μνήμης και του CPU όσο αφορά την MySQL σε σχέση με τους υπόλοιπους servers.

Η τρίτη επιλογή (Dedicated MySQL Server Machine) δίνει την ελευθερία στην MySQL να χρησιμοποιήσει όσους πόρους χρειάζεται από τον υπολογιστή για να ανταποκριθεί όσο το δυνατόν πιο γρήγορα στις απαιτήσεις του χρήστη. Κυρίως θα απορροφήσει ένα μεγάλο κομμάτι της μνήμης. Αυτή η επιλογή είναι κατάλληλη όταν ο υπολογιστής θα τρέχει αποκλειστικά και μόνο την MySQL και κανένα άλλο server πρόγραμμα.

Εμείς θα επιλέξουμε την δεύτερη επιλογή για τον απλό λόγο ότι πολλοί χρήστες του Internet πειραματίζονται με την εγκατάσταση Web servers όπως τον Apache και με προγράμματα ανοιχτού κώδικα (Open Source) που απαιτούν εγκατάσταση της MySQL για να προσφέρουν τις υπηρεσίες τους, όπως π.χ. την δημιουργία ενός forum. Επιλέγουμε λοιπόν το κουμπί **NEXT** με την δεύτερη επιλογή τσεκαρισμένη και προχωράμε στο επόμενο βήμα.

Αμέσως μετά θα βρεθούμε πάλι μπροστά σε τρεις επιλογές. Οι επιλογές επιδρούν άμεσα στην ενεργοποίηση/απενεργοποίηση των δυο μηχανών αποθήκευσης δεδομένων της MySQL οι οποίες είναι η InnoDB και MyISAM.

Στην πρώτη επιλογή μας, και οι δυο αυτές μηχανές είναι ενεργές και οι πόροι όσο και το φορτίο εργασίας μοιράζεται μεταξύ τους. Εάν τώρα θέλουμε και τις δυο μηχανές ενεργοποιημένες αλλά όμως μια μόνο να κάνει την δουλειά τότε διαλέγουμε την δεύτερη επιλογή. Ενώ στην τρίτη η InnoDB είναι εντελώς απενεργοποιημένη.

Επειδή στους περισσότερους αρέσει να ενεργοποιούν κάθε δυνατό εργαλείο ενός προγράμματος, θα επιλέξουμε την πρώτη επιλογή. Έτσι η βάση μας θα είναι ικανή να ανταποκριθεί σε οποιοδήποτε πρόγραμμα, εφαρμογή ή παράδειγμα αναλύσουμε.

Με την επιλογή μας δώσαμε εντολή να εγκαταστήσουμε και τις δυο μηχανές διαχείρισης στοιχείων που προσφέρει η MySQL. Σαν αποτέλεσμα αυτής της επιλογής μας, η MySQL μας δίνει την δυνατότητα να μπορέσουμε να μεταφέρουμε το InnoDB σε διαφορετικό σκληρό δίσκο για ακόμα μεγαλύτερη απόδοση. Εμείς δεν χρειάζεται να αλλάξουμε το μονοπάτι εγκατάστασης του InnoDB, οπότε απλά επιλέγουμε το κουμπί **NEXT**.

Για να μην φτάσει ο server στα όρια του τόσο ώστε να μην μπορεί να εξυπηρετήσει τα εισερχόμενα αιτήματα, στο επόμενο παράθυρο μπορούμε να καθορίσουμε τον αριθμό των ενεργών ενώσεων που θα κάνουν ταυτόχρονα κάποια διεργασία επάνω στην βάση. Αν και μπορούμε να θέσουμε μόνοι μας το όριο επιλέγοντας την τρίτη επιλογή (Manual Setting), εμείς όμως θα επιλέξουμε την πρώτη επιλογή (επιτρέπει μέχρι 20 ταυτόχρονες ενώσεις να κάνουν ερωτήματα στην βάση). Αφού λοιπόν επιλέξουμε την πρώτη επιλογή συνεχίζουμε με το κουμπί **NEXT**.

Στο επόμενο παράθυρο έχουμε την επιλογή να δεχτούμε την ενεργοποίηση του TCP/IP και της θύρας (port) την οποία «ακούει» η MySQL. Αν και μπορούμε να αλλάξουμε την θύρα σε ότι αριθμό είναι κατάλληλος για το δικό μας δίκτυο, απλά θα δεχτούμε το TCP/IP και το port 3306 για την δική μας βάση (αυτές είναι οι προεπιλεγμένες ρυθμίσεις που προτείνει πάντα η MySQL). Όσο αφορά την επιλογή Enable Strict Mode, αν και το επιθυμητό είναι να την ενεργοποιήσουμε έτσι ώστε ο MySQL να συμπεριφέρεται όπως μια κοινή βάση δεδομένων, εξ αρχής δεν είναι επιλεγμένη. Ενεργοποιούμε και αυτή την επιλογή και μετά συνεχίζουμε με το κουμπί **NEXT**.

Έπειτα μας ζητείται να διαλέξουμε ποιο σετ χαρακτήρων θέλουμε η βάση μας να υποστηρίξει. Εμείς επιλέγουμε την πρώτη επιλογή (Standard Character Set) και πατάμε το κουμπί **NEXT**.

Εδώ δεχόμαστε και τις δυο επιλογές - να δηλωθεί η MySQL σαν Windows Service αλλά και να αναβαθμιστεί το Windows PATH με το BIN Directory του MySQL - έτσι ώστε από οποιοδήποτε directory του command line και αν γράφουμε τις εντολές να είναι δυνατή η εκτέλεση τους και να αναγνωρίζονται από τα Windows.

Για την ασφάλεια του Server, μας ζητείται στο επόμενο βήμα να βάλουμε το password το οποίο προστατεύει τις ρυθμίσεις και την λειτουργικότητα της βάσης. Μόνο βάζοντας το σωστό κωδικό πρόσβασης, θα αποκτήσει κάποιος το δικαίωμα να προσθέσει επιπλέον βάσεις, να προσθέσει ή να αφαιρέσει πίνακες με δεδομένα, να τροποποιήσει αποθηκευμένες πληροφορίες και να κάνει γενικώς διαχείριση όλης της MySQL βάσης. Επιλέγουμε την πρώτη επιλογή, βάζουμε το password που μας ικανοποιεί και συνεχίζουμε με το κουμπί **NEXT**.

Στο επόμενο παράθυρο εμφανίζεται μια επιβεβαίωση των επιλογών μας και το περιβάλλον περιμένει να πατήσουμε το κουμπί EXECUTE για να ολοκληρωθεί και αυτό το τελευταίο στάδιο της εγκατάστασης.

Όταν πατήσουμε το κουμπί EXECUTE, θα δούμε ότι το παράθυρο παρουσιάζει το μήνυμα Processing configuration και απλά περιμένουμε να ολοκληρωθούν οι ρυθμίσεις που επιλέξαμε.

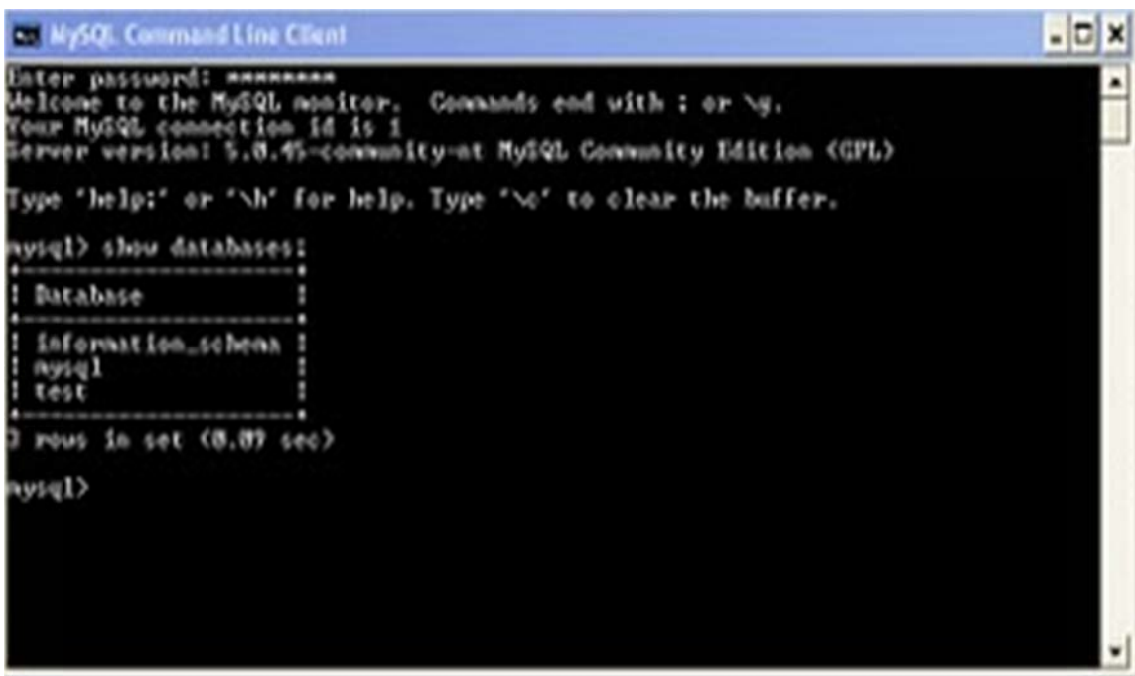
Εάν όλη η διαδικασία ολοκληρώθηκε με επιτυχία, τότε απλά θα λάβουμε ένα παράθυρο που επιβεβαιώνει αυτό ακριβώς το γεγονός.

Κλείνοντας το Wizard, θα επιθυμούσαμε να επιβεβαιώσουμε ότι η βάση λειτουργεί σωστά. Οπότε επιλέγουμε το κουμπί **START** (στα Windows), μετά **ALL PROGRAMS -->MYSQL-->MYSQL SERVER 5.0-->MYSQL COMMAND LINE CLIENT**. Λογικό είναι να μην έχουμε πρόσβαση επάνω στην βάση αφού ορίσαμε κωδικό πρόσβασης κατά την διάρκεια της εγκατάστασης. Γράφουμε λοιπόν το password και επιλέγουμε **ENTER**. Εάν έχουμε κάνει την εγκατάσταση σωστά (δεν ξεχάσαμε το password), τότε θα πρέπει η MySQL να μας παραχωρήσει την πρόσβαση στο σύστημα.

Για να δοκιμάσουμε να δούμε την επικοινωνία μας με την MySQL, απλά γράφουμε την εντολή

**show databases;**

Η εντολή αυτή θα μας εμφανίσει την παρακάτω εικόνα. Στα σίγουρα θα πρέπει να βλέπουμε στο αποτέλεσμα της εντολής τις databases με τα ονόματα mysql και test.



```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
2 rows in set (0.07 sec)

mysql>
```

**Εικόνα Α6.γ : Διαδικασία επικοινωνίας μας με την MySQL .**

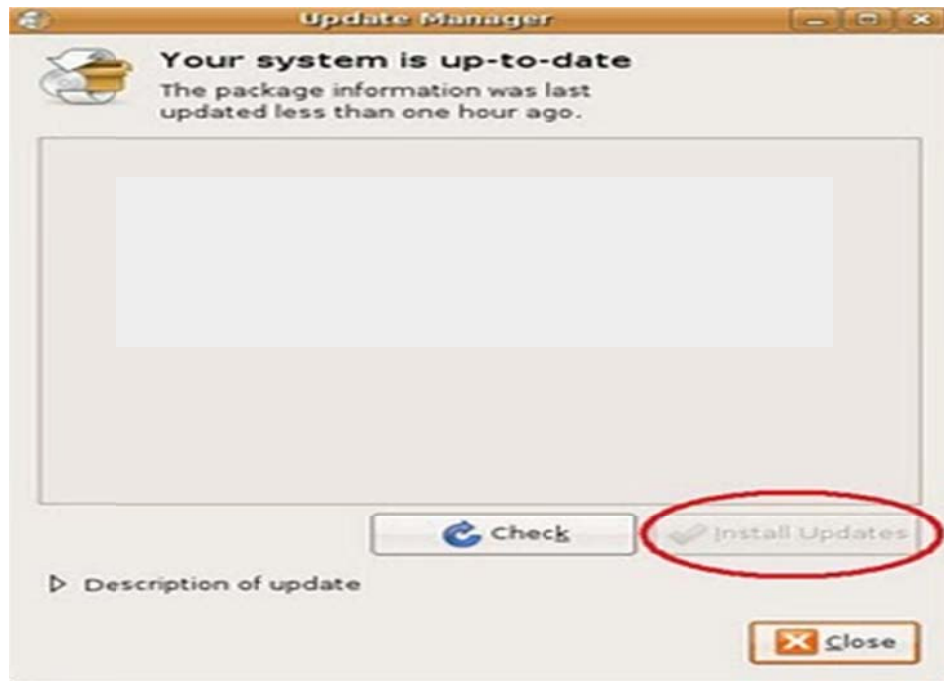
Αυτή η απλή διαδικασία είναι μια πρόχειρη απόδειξη ότι η MySQL έχει ενεργοποιηθεί χωρίς κανένα πρόβλημα, και τα Windows έχουν επιτυχημένα αναβαθμίσει το PATH.

## Παράρτημα Α7: Εγκατάσταση της MySQL σε Ubuntu

Παρακάτω θα δείξουμε ποια είναι τα βήματα εγκατάστασης της MySQL και σε λειτουργικό Ubuntu.

Το πρώτο βήμα που χρειάζεται να πραγματοποιηθεί στο Ubuntu (εδώ μιλάμε με την υπόθεση ότι είναι ήδη εγκαταστημένο το λειτουργικό) είναι να έχουμε κατεβάσει και εγκαταστήσει όλα τα τελευταία updates. Η διαδικασία αυτή είναι πολύ εύκολη. Πηγαίνουμε στο κεντρικό μενού **System**, επιλέγουμε **Administration** και μετά **Update Manager**.

Στην συνέχεια θα μας εμφανιστεί το παράθυρο του Update Manager. Εάν υπάρχουν διαθέσιμα updates, τότε θα εμφανιστούν μέσα στο τετράγωνο πλαίσιο του παραθύρου και το κουμπί Install Updates θα είναι ενεργοποιημένο. Απλά το επιλέγουμε, και το σύστημα θα κατεβάσει και θα εγκαταστήσει όλα τα διαθέσιμα updates. Εάν μας ζητηθεί να κάνουμε επανεκκίνηση στο λειτουργικό, τότε απλά δεχόμαστε το αίτημα αυτό, και μετά την επαναφορά του συστήματος θα είμαστε έτοιμοι να συνεχίσουμε με την εγκατάσταση της MySQL. Στην πιο κάτω εικόνα η οποία προέρχεται από τον υπολογιστή μου, το σύστημα είναι ενημερωμένο και δεν υπάρχουν διαθέσιμα updates οπότε δεν υπάρχει λόγος για επανεκκίνηση.



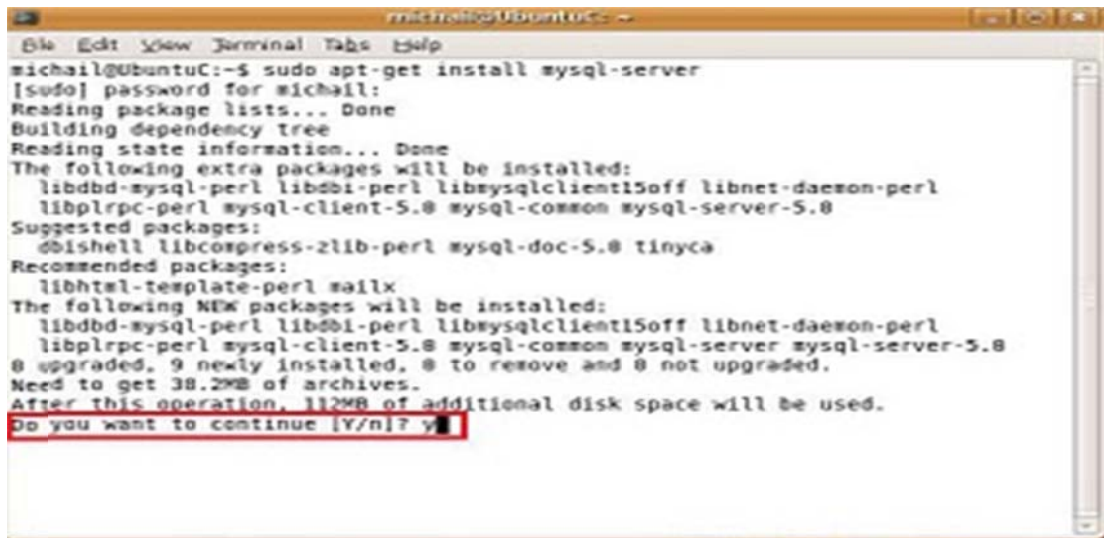
Εικόνα Α7.α : Παράθυρο του Update Manager.

Για να εγκαταστήσουμε την MySQL χρειάζεται να καλέσουμε μέσα από το terminal την πιο κάτω εντολή:

```
sudo apt-get install mysql-server
```

Αυτό θα ενεργοποιήσει τον μηχανισμό εγκατάστασης της MySQL στο Ubuntu, το οποίο σε πολύ μικρό χρονικό διάστημα θα έχει ολοκληρωθεί. Όταν μας ζητηθεί το password, απλά γράφουμε το κωδικό πρόσβασης που έχουμε σαν sudo χρήστης στο Ubuntu. Είναι λογικό το σύστημα να μην αφήνει κανέναν άλλο εκτός από τον Administrator να έχει δικαιώματα εγκατάστασης προγραμμάτων στο λειτουργικό.

Απαντάμε θετικά (γράφοντας απλά το γράμμα “y” και επιλέγοντας ENTER) στην ερώτηση που μας ρωτάει το σύστημα, εάν δεχόμαστε να χρησιμοποιηθούν περίπου 112MB από τον σκληρό δίσκο για την εγκατάσταση της MySQL.



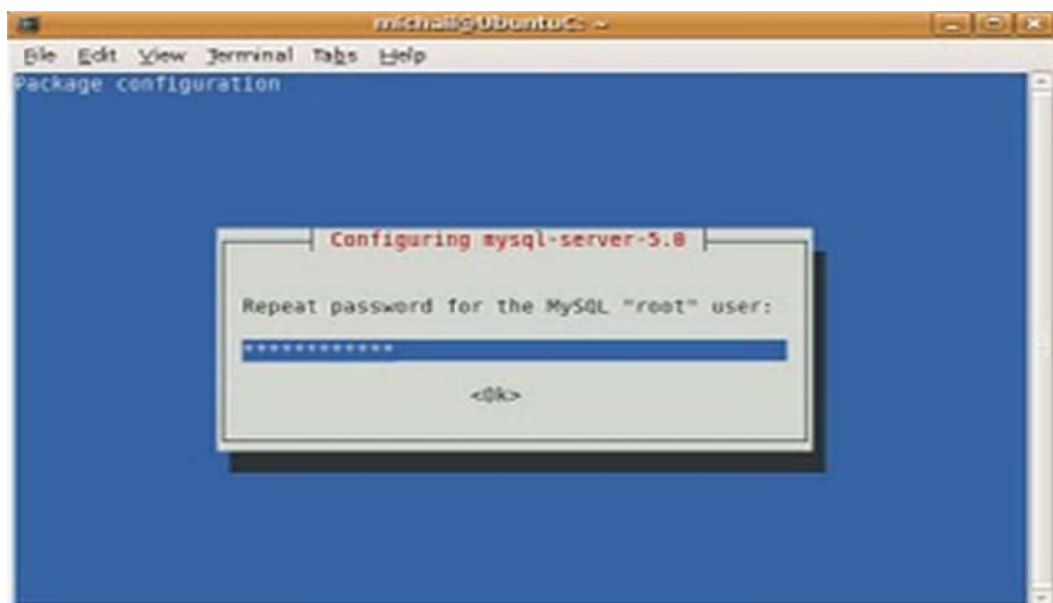
```
michail@ubuntuC:~$ sudo apt-get install mysql-server
[sudo] password for michail:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libdbd-mysql-perl libdbi-perl libmysqlclient15off libnet-daemon-perl
  libplrpc-perl mysql-client-5.0 mysql-common mysql-server-5.0
Suggested packages:
  dbshell libcups-zebra-perl mysql-doc-5.0 tinyca
Recommended packages:
  libhtml-template-perl mailx
The following NEW packages will be installed:
  libdbd-mysql-perl libdbi-perl libmysqlclient15off libnet-daemon-perl
  libplrpc-perl mysql-client-5.0 mysql-common mysql-server mysql-server-5.0
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 38.2MB of archives.
After this operation, 112MB of additional disk space will be used.
Do you want to continue [Y/n]? y
```

**Εικόνα Α7.β:** Διαδικασία εγκατάστασης της MySQL.

Αμέσως μετά την απάντησή μας, το Ubuntu ενώνεται με το Ίντερνετ για να κατεβάσει τα απαραίτητα αρχεία και να τα εγκαταστήσει.

Όταν τελειώσει με αυτή την διαδικασία, θα μας ζητηθεί να καταχωρήσουμε ένα password για το root λογαριασμό της βάσης, το οποίο φυσικά θα μας δίνει απεριόριστη πρόσβαση στις ιδιότητες της. Όταν γράψουμε τον κωδικό συνεχίζουμε με ENTER.

Για να βεβαιωθεί το σύστημα ότι γράψαμε σωστά τον κωδικό πρόσβασης μας ζητάει να το γράψουμε ακόμα μια φορά. Οπότε επαναλαμβάνουμε τον ίδιο κωδικό και πατάμε ENTER.



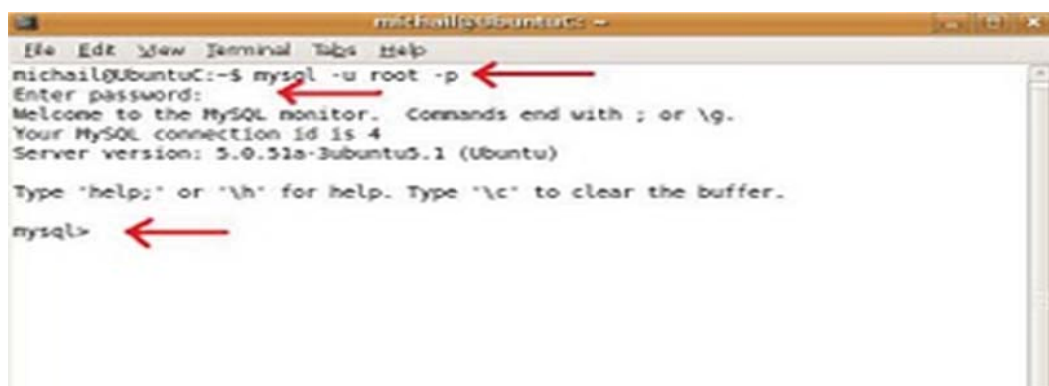
**Εικόνα Α7.γ:** Διαδικασία επιβεβαίωσης του συστήματος για τον κωδικό πρόσβασης.

Αφού επιβεβαιωθεί ότι ο κωδικός που γράψαμε είναι σωστός και τις δύο φορές, τότε το Ubuntu ολοκληρώνει την εγκατάσταση και μας επαναφέρει πίσω στο terminal prompt.

Αυτό που έχει απομείνει τώρα, είναι να δοκιμάσουμε την σύνδεση μας με την βάση. Αυτό γίνεται δίνοντας την εντολή:

**mysql -u root -p**

Με το mysql δηλώνουμε ότι θέλουμε να ενεργοποιήσουμε ένα session με την **mysql**, το **-u** δηλώνει ότι θα δώσουμε αμέσως μετά το όνομα του χρήστη που θέλει να συνδεθεί (root) και το **-p** δηλώνει ότι θα μας ζητηθεί ο κωδικός σύνδεσης όταν πατήσουμε ENTER. Πραγματικά, στην επόμενη γραμμή στο terminal μας ζητείται ο κωδικός πρόσβασης. Αφού επιτυχημένα γράψουμε και τον κωδικό, τότε το prompt αλλάζει σε mysql που συμβολίζει την επιτυχής ένωση.



```
michael@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.51a-3ubuntu5.1 (Ubuntu)

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

mysql>
```

**Εικόνα Α7.δ : Ενεργοποίηση ενός session.**

Όπως και με την εγκατάσταση στην πλατφόρμα των Windows έτσι και εδώ θα χρησιμοποιήσουμε την γνωστή εντολή **SHOW DATABASES** απλά για να βεβαιωθούμε ότι μπορούμε να επικοινωνήσουμε σωστά με την MySQL.

Για να διακόψουμε την ένωση μας με την MySQL (να σταματήσουμε το session) απλά γράφουμε την εντολή EXIT και πατάμε ENTER.

Εδώ ολοκληρώσαμε την διαδικασία εγκατάστασης της MySQL και σε λειτουργικό Ubuntu.

## **ΠΑΡΑΡΤΗΜΑ Β : Κώδικας Εφαρμογής**



## CustomerEditor.java

```

package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import org.jdesktop.application.Action;

/**
 * Το παράθυρο προσθήκης / επεξεργασίας πελατών
 * @author Kostas
 */
public class CustomerEditor extends javax.swing.JDialog {

    /** Creates new form CustomerEditor */
    public CustomerEditor(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();

        protected Customers currentRecord;

        public Customers getCurrentRecord() {
            return currentRecord;
        }

        public void setCurrentRecord(Customers currentRecord) {
            /*this.currentRecord = currentRecord;*/
            Customers oldRecord = this.currentRecord;
            this.currentRecord = currentRecord;
            propertyChangeSupport.firePropertyChange("currentRecord", oldRecord, currentRecord);
        }

        /*Gia thn apo8hkeush h akurwsh allagwn sth forma pelath*/
        protected boolean customerConfirmed;

        public boolean isCustomerConfirmed() {
            return customerConfirmed;
        }

        public void setCustomerConfirmed(boolean customerConfirmed) {
            this.customerConfirmed = customerConfirmed;
        }

        private PropertyChangeSupport propertyChangeSupport = new PropertyChangeSupport(this);

        public void addPropertyChangeListener(PropertyChangeListener listener) {
            propertyChangeSupport.addPropertyChangeListener(listener);
        }

        public void removePropertyChangeListener(PropertyChangeListener listener) {
            propertyChangeSupport.removePropertyChangeListener(listener);
        }

        /** This method is called from within the constructor to
         * initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is
         * always regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
        // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
        private void initComponents() {
            bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

            klfPUEntityManager = java.beans.Beans.isDesignTime() ? null :
            javax.persistence.Persistence.createEntityManagerFactory("klfPU").createEntityManager();
            nomoiQuery = java.beans.Beans.isDesignTime() ? null :
            klfPUEntityManager.createQuery("SELECT n FROM Nomoi n");
            nomoiList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
            nomoiQuery.getResultList();
            nomoiListCellRenderer1 = new klfapplication.NomoiListCellRenderer();
            nameLabel = new javax.swing.JLabel();
            nameTextField = new javax.swing.JTextField();
            codeLabel = new javax.swing.JLabel();
            codeTextField = new javax.swing.JTextField();
            companyLabel = new javax.swing.JLabel();
            companyTextField = new javax.swing.JTextField();
            telHomeLabel = new javax.swing.JLabel();
            telHomeTextField = new javax.swing.JTextField();
            telWorkLabel = new javax.swing.JLabel();
            telWorkTextField = new javax.swing.JTextField();
            telFaxLabel = new javax.swing.JLabel();
            telFaxTextField = new javax.swing.JTextField();
            telMobLabel = new javax.swing.JLabel();
            telMobTextField = new javax.swing.JTextField();
            mailLabel = new javax.swing.JLabel();
            mailTextField = new javax.swing.JTextField();

            nomosLabel = new javax.swing.JLabel();
            nomosComboBox = new javax.swing.JComboBox();
            cityLabel = new javax.swing.JLabel();
            cityTextField = new javax.swing.JTextField();
            areaLabel = new javax.swing.JLabel();
            areaTextField = new javax.swing.JTextField();
            addressLabel = new javax.swing.JLabel();
            addressTextField = new javax.swing.JTextField();
            addrNumLabel = new javax.swing.JLabel();
            addrNumTextField = new javax.swing.JTextField();
            geoLaLabel = new javax.swing.JLabel();
            geoLoLabel = new javax.swing.JLabel();
            geoLoTextField = new javax.swing.JTextField();
            geoLaTextField = new javax.swing.JTextField();
            notesLabel = new javax.swing.JLabel();
            notesScrollPane = new javax.swing.JScrollPane();
            notesTextArea = new javax.swing.JTextArea();
            saveCustButton = new javax.swing.JButton();
            cancelCustButton = new javax.swing.JButton();
            customersEditorLabel = new javax.swing.JLabel();
            showMapCustButton = new javax.swing.JButton();

            org.jdesktop.application.ResourceMap resourceMap =
            org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
            ResourceMap(CustomerEditor.class);
            nomoiListCellRenderer1.setText(resourceMap.getString("nomoiListCellRenderer1.text")); // NOI18N
            nomoiListCellRenderer1.setName("nomoiListCellRenderer1"); // NOI18N

            setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
            setTitle(resourceMap.getString("Form.title")); // NOI18N
            setModal(true);
            setName("Form"); // NOI18N

            nameLabel.setText(resourceMap.getString("nameLabel.text")); // NOI18N
            nameLabel.setName("nameLabel"); // NOI18N

            nameTextField.setName("nameTextField"); // NOI18N

            org.jdesktop.beansbinding.Binding binding =
            org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
            eStrategy.READ_WRITE, this,
            org.jdesktop.beansbinding.ELProperty.create("${currentRecord.name}"), nameTextField,
            org.jdesktop.beansbinding.BeanProperty.create("text"));
            bindingGroup.addBinding(binding);

            codeLabel.setText(resourceMap.getString("codeLabel.text")); // NOI18N
            codeLabel.setName("codeLabel"); // NOI18N

            codeTextField.setName("codeTextField"); // NOI18N

            binding =
            org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
            eStrategy.READ_WRITE, this,
            org.jdesktop.beansbinding.ELProperty.create("${currentRecord.code}"), codeTextField,
            org.jdesktop.beansbinding.BeanProperty.create("text"));
            bindingGroup.addBinding(binding);

            companyLabel.setText(resourceMap.getString("companyLabel.text")); // NOI18N
            companyLabel.setName("companyLabel"); // NOI18N

            companyTextField.setName("companyTextField"); // NOI18N

            binding =
            org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
            eStrategy.READ_WRITE, this,
            org.jdesktop.beansbinding.ELProperty.create("${currentRecord.company}"), companyTextField,
            org.jdesktop.beansbinding.BeanProperty.create("text"));
            bindingGroup.addBinding(binding);

            telHomeLabel.setText(resourceMap.getString("telHomeLabel.text")); // NOI18N
            telHomeLabel.setName("telHomeLabel"); // NOI18N

            telHomeTextField.setName("telHomeTextField"); // NOI18N

            binding =
            org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
            eStrategy.READ_WRITE, this,
            org.jdesktop.beansbinding.ELProperty.create("${currentRecord.telHome}"), telHomeTextField,
            org.jdesktop.beansbinding.BeanProperty.create("text"));
            bindingGroup.addBinding(binding);

            telWorkLabel.setText(resourceMap.getString("telWorkLabel.text")); // NOI18N
            telWorkLabel.setName("telWorkLabel"); // NOI18N

            telWorkTextField.setName("telWorkTextField"); // NOI18N

            binding =
            org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
            eStrategy.READ_WRITE, this,
            org.jdesktop.beansbinding.ELProperty.create("${currentRecord.telWork}"), telWorkTextField,
            org.jdesktop.beansbinding.BeanProperty.create("text"));
            bindingGroup.addBinding(binding);

            telFaxLabel.setText(resourceMap.getString("telFaxLabel.text")); // NOI18N
            telFaxLabel.setName("telFaxLabel"); // NOI18N

            telFaxTextField.setName("telFaxTextField"); // NOI18N
        }
    }

```

```

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.telFax}"), telFaxTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

telMobLabel.setText(resourceMap.getString("telMobLabel.text")); // NOI18N
telMobLabel.setName("telMobLabel"); // NOI18N

telMobTextField.setName("telMobTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.telMobile}"), telMobTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

mailLabel.setText(resourceMap.getString("mailLabel.text")); // NOI18N
mailLabel.setName("mailLabel"); // NOI18N

mailTextField.setName("mailTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.email}"), mailTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

nomosLabel.setText(resourceMap.getString("nomosLabel.text")); // NOI18N
nomosLabel.setName("nomosLabel"); // NOI18N

nomosComboBox.setName("nomosComboBox"); // NOI18N
nomosComboBox.setRenderer(nomoiListCellRenderer1);

org.jdesktop.swingbinding.JComboBoxBinding.jComboBoxBinding =
org.jdesktop.swingbinding.Bindings.createJComboBoxBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, nomoiList, nomosComboBox);
bindingGroup.addBinding(jComboBoxBinding);
binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.nomoi}"), nomosComboBox,
org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
bindingGroup.addBinding(binding);

cityLabel.setText(resourceMap.getString("cityLabel.text")); // NOI18N
cityLabel.setName("cityLabel"); // NOI18N

cityTextField.setName("cityTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.city}"), cityTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

areaLabel.setText(resourceMap.getString("areaLabel.text")); // NOI18N
areaLabel.setName("areaLabel"); // NOI18N

areaTextField.setName("areaTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.area}"), areaTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

addressLabel.setText(resourceMap.getString("addressLabel.text")); // NOI18N
addressLabel.setName("addressLabel"); // NOI18N

addressTextField.setName("addressTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.address}"), addressTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

addrNumLabel.setText(resourceMap.getString("addrNumLabel.text")); // NOI18N
addrNumLabel.setName("addrNumLabel"); // NOI18N

addrNumTextField.setName("addrNumTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.addressNum}"), addrNumTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

geoLaLabel.setText(resourceMap.getString("geoLaLabel.text")); // NOI18N
geoLaLabel.setName("geoLaLabel"); // NOI18N

geoLoLabel.setText(resourceMap.getString("geoLoLabel.text")); // NOI18N
geoLoLabel.setName("geoLoLabel"); // NOI18N

```

```

geoLoTextField.setName("geoLoTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.longitude}"), geoLoTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

geoLaTextField.setName("geoLaTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.latitude}"), geoLaTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

notesLabel.setText(resourceMap.getString("notesLabel.text")); // NOI18N
notesLabel.setName("notesLabel"); // NOI18N

notesScrollPane.setName("notesScrollPane"); // NOI18N

notesTextArea.setColumns(20);
notesTextArea.setRows(5);
notesTextArea.setName("notesTextArea"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentRecord.notes}"), notesTextArea,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

notesScrollPane.setViewportView(notesTextArea);

saveCustButton.setText(resourceMap.getString("saveCustButton.text")); // NOI18N
saveCustButton.setName("saveCustButton"); // NOI18N
saveCustButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
saveCustomer(evt);
}
});

cancelCustButton.setText(resourceMap.getString("cancelCustButton.text")); // NOI18N
cancelCustButton.setName("cancelCustButton"); // NOI18N
cancelCustButton.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
cancelCustomer(evt);
}
});

customersEditorLabel.setFont(resourceMap.getFont("customersEditorLabel.font")); // NOI18N
customersEditorLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
customersEditorLabel.setText(resourceMap.getString("customersEditorLabel.text")); // NOI18N
customersEditorLabel.setName("customersEditorLabel"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().getActionMap(customersEditor.class, this);
showMapCustButton.setAction(actionMap.get("showTheMap")); // NOI18N
showMapCustButton.setName("showMapCustButton"); // NOI18N

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup().addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(layout.createSequentialGroup().addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup().addGap(179, 179, 179)
.addComponent(customersEditorLabel, javax.swing.GroupLayout.DEFAULT_SIZE,
409, Short.MAX_VALUE)
.addGroup(layout.createSequentialGroup().addGap(168, 168, 168)
.addGroup(layout.createSequentialGroup().addContainerGap()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup().addContainerGap()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(nameLabel)
.addComponent(companyLabel)
.addComponent(telHomeLabel)
.addComponent(telWorkLabel)
.addComponent(telMobLabel)
.addComponent(telFaxLabel)
.addComponent(mailLabel)
.addComponent(codeLabel)
.addComponent(notesLabel))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(layout.createSequentialGroup().addContainerGap(10, 10, 10)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(mailTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)

```

```

        .addComponent(telHomeTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(telMobTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(telFaxTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(TelWorkTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(companyTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(nameTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE)
        .addComponent(codeTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 251, Short.MAX_VALUE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(geoLaLabel)
        .addComponent(geoLoLabel)
        .addComponent(addrNumLabel)
        .addComponent(addressLabel)
        .addComponent(areaLabel)
        .addComponent(cityLabel)
        .addComponent(nomosLabel))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(showMapCustButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(geoLoTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(addrNumTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(addressTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(areaTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(cityTextField,
javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 288,
Short.MAX_VALUE)
        .addComponent(geoLaTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 288, Short.MAX_VALUE)
        .addComponent(nomosComboBox, 0, 288, Short.MAX_VALUE)))
        .addGroup(layout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(notesScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 658, Short.MAX_VALUE)))
        .addGroup(layout.createParallelGroup.Alignment.TRAILING,
layout.createSequentialGroup())
        .addComponent(saveCustButton, javax.swing.GroupLayout.PREFERRED_SIZE,
121, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(35, 35, 35)
        .addComponent(cancelCustButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()
);
layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addComponent(customersEditorLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(codeLabel)
        .addComponent(nomosComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(codeTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(nomosLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(nameTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(nameLabel)
        .addComponent(cityTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cityLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(companyTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(companyLabel)
        .addComponent(areaTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(areaLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(telHomeTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(telHomeLabel)
        .addComponent(addressTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(addressLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(telWorkTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(telWorkLabel)
        .addComponent(addrNumTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(addrNumLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(telFaxTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(telFaxLabel)
        .addComponent(geoLoTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(geoLaLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(telMobTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(telMobLabel)
        .addComponent(geoLaTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(geoLoLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(notesLabel)
        .addComponent(notesScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 89,
Short.MAX_VALUE)
        .addGap(20, 20, 20)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(saveCustButton)
        .addComponent(cancelCustButton))
        .addContainerGap()
);

bindingGroup.bind();

pack();
} // </editor-fold> // GEN-END: initComponents

/* Gia to save this formas pelath */
private void saveCustomer(java.awt.event.ActionEvent evt) // GEN-FIRST:event_saveCustomer

    setCustomerConfirmed(true);
    setVisible(false);

} // GEN-LAST:event_saveCustomer
/* Gia akyrwrsh formas pelath */
private void cancelCustomer(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_cancelCustomer
    setCustomerConfirmed(false);
    setVisible(false);
} // GEN-LAST:event_cancelCustomer

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            CustomerEditor dialog = new CustomerEditor(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

// kalei thn show map gia na emfanisei to xarh toy pelath
@Action
public void showTheMap() {
    KlfView.showMap(currentRecord.getLatitude(), currentRecord.getLongitude());
}

// Variables declaration - do not modify // GEN-BEGIN:variables
private javax.swing.JTextField TelWorkTextField;
private javax.swing.JLabel addrNumLabel;
private javax.swing.JTextField addrNumTextField;
private javax.swing.JLabel addressLabel;
private javax.swing.JTextField addressTextField;
private javax.swing.JLabel areaLabel;
private javax.swing.JTextField areaTextField;
private javax.swing.JButton cancelCustButton;
private javax.swing.JLabel cityLabel;
private javax.swing.JTextField cityTextField;
private javax.swing.JLabel codeLabel;
private javax.swing.JTextField codeTextField;
private javax.swing.JLabel companyLabel;
private javax.swing.JTextField companyTextField;
private javax.swing.JLabel customersEditorLabel;
private javax.swing.JLabel geoLaLabel;
private javax.swing.JTextField geoLaTextField;
private javax.swing.JLabel geoLoLabel;
private javax.swing.JTextField geoLoTextField;

```

```
private javax.persistence.EntityManager klfPUEntityManager;
private javax.swing.JLabel mailLabel;
private javax.swing.JTextField mailTextField;
private javax.swing.JLabel nameLabel;
private javax.swing.JTextField nameTextField;
private java.util.List<klfapplication.Nomoi> nomoiList;
private klfapplication.NomoiListCellRenderer nomoiListCellRenderer1;
private javax.persistence.Query nomoiQuery;
private javax.swing.JComboBox nomosComboBox;
private javax.swing.JLabel nomosLabel;
private javax.swing.JLabel notesLabel;
private javax.swing.JScrollPane notesScrollPane;
private javax.swing.JTextArea notesTextArea;
private javax.swing.JButton saveCustButton;
private javax.swing.JButton showMapCustButton;
private javax.swing.JLabel telFaxLabel;
private javax.swing.JTextField telFaxTextField;
private javax.swing.JLabel telHomeLabel;
private javax.swing.JTextField telHomeTextField;
private javax.swing.JLabel telMobLabel;
private javax.swing.JTextField telMobTextField;
private javax.swing.JLabel telWorkLabel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration//GEN-END:variables
```

```
}
```

## Customers.java

```
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.io.Serializable;
import java.util.List;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToMany;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Transient;

/**
 * Κλάση του πίνακα customers της βάσης
 * @author Kostas
 */
@Entity
@Table(name = "customers", catalog = "klf", schema = "")
@NamedQueries({
    @NamedQuery(name = "Customers.findAll", query = "SELECT c FROM Customers c"),
    @NamedQuery(name = "Customers.findByIdcustomers", query = "SELECT c FROM Customers c
WHERE c.idcustomers = :idcustomers"),
    @NamedQuery(name = "Customers.findByIdCode", query = "SELECT c FROM Customers c
WHERE c.code = :code"),
    @NamedQuery(name = "Customers.findByName", query = "SELECT c FROM Customers c
WHERE c.name = :name"),
    @NamedQuery(name = "Customers.findByCompany", query = "SELECT c FROM Customers c
WHERE c.company = :company"),
    @NamedQuery(name = "Customers.findByTelHome", query = "SELECT c FROM Customers c
WHERE c.telHome = :telHome"),
    @NamedQuery(name = "Customers.findByTelWork", query = "SELECT c FROM Customers c
WHERE c.telWork = :telWork"),
    @NamedQuery(name = "Customers.findByTelFax", query = "SELECT c FROM Customers c
WHERE c.telFax = :telFax"),
    @NamedQuery(name = "Customers.findByTelMobile", query = "SELECT c FROM Customers c
WHERE c.telMobile = :telMobile"),
    @NamedQuery(name = "Customers.findByEmail", query = "SELECT c FROM Customers c
WHERE c.email = :email"),
    @NamedQuery(name = "Customers.findByCity", query = "SELECT c FROM Customers c
WHERE c.city = :city"),
    @NamedQuery(name = "Customers.findByArea", query = "SELECT c FROM Customers c
WHERE c.area = :area"),
    @NamedQuery(name = "Customers.findByAddress", query = "SELECT c FROM Customers c
WHERE c.address = :address"),
    @NamedQuery(name = "Customers.findByAddressNum", query = "SELECT c FROM Customers c
WHERE c.addressNum = :addressNum"),
    @NamedQuery(name = "Customers.findByLongitude", query = "SELECT c FROM Customers c
WHERE c.longitude = :longitude"),
    @NamedQuery(name = "Customers.findByLatitude", query = "SELECT c FROM Customers c
WHERE c.latitude = :latitude");
})
public class Customers implements Serializable {
    @Transient
    private PropertyChangeSupport changeSupport = new PropertyChangeSupport(this);
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idcustomers", nullable = false)
    private Integer idcustomers;
    @Column(name = "code")
    private Integer code;
    @Column(name = "name", length = 45)
    private String name;
    @Column(name = "company", length = 45)
    private String company;
    @Column(name = "tel_home", length = 14)
    private String telHome;
    @Column(name = "tel_work", length = 14)
    private String telWork;
    @Column(name = "tel_fax", length = 14)
    private String telFax;

    @Column(name = "tel_mobile", length = 14)
    private String telMobile;
    @Column(name = "email", length = 45)
    private String email;
    @Column(name = "city", length = 45)
    private String city;
    @Column(name = "area", length = 45)
    private String area;
    @Column(name = "address", length = 45)
    private String address;
    @Column(name = "address_num")
    private Short addressNum;
    @Column(name = "longitude", length = 10)
    private String longitude;
    @Column(name = "latitude", length = 10)
    private String latitude;
    @Lob
    @Column(name = "photo")
    private byte[] photo;
    @Lob
    @Column(name = "notes", length = 1677215)
    private String notes;
    @JoinColumn(name = "nomos", referencedColumnName = "idnomoi")
    @ManyToOne
    private Nomoi nomoi;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "customerId")
    private List<Customersproducts> customersproductsList;
    //private Collection<Customersproducts> CustomersproductsCollection;

    public Customers() {
    }

    public Customers(Integer idcustomers) {
        this.idcustomers = idcustomers;
    }

    public Integer getIdcustomers() {
        return idcustomers;
    }

    public void setIdcustomers(Integer idcustomers) {
        Integer oldIdcustomers = this.idcustomers;
        this.idcustomers = idcustomers;
        changeSupport.firePropertyChange("idcustomers", oldIdcustomers, idcustomers);
    }

    public Integer getCode() {
        return code;
    }

    public void setCode(Integer code) {
        Integer oldCode = this.code;
        this.code = code;
        changeSupport.firePropertyChange("code", oldCode, code);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        String oldName = this.name;
        this.name = name;
        changeSupport.firePropertyChange("name", oldName, name);
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        String oldCompany = this.company;
        this.company = company;
        changeSupport.firePropertyChange("company", oldCompany, company);
    }

    public String getTelHome() {
        return telHome;
    }

    public void setTelHome(String telHome) {
        String oldTelHome = this.telHome;
        this.telHome = telHome;
        changeSupport.firePropertyChange("telHome", oldTelHome, telHome);
    }

    public String getTelWork() {
        return telWork;
    }
}
```

```

}

public void setTelWork(String telWork) {
    String oldTelWork = this.telWork;
    this.telWork = telWork;
    changeSupport.firePropertyChange("telWork", oldTelWork, telWork);
}

public String getTelFAX() {
    return telFAX;
}

public void setTelFAX(String telFAX) {
    String oldTelFAX = this.telFAX;
    this.telFAX = telFAX;
    changeSupport.firePropertyChange("telFAX", oldTelFAX, telFAX);
}

public String getTelMobile() {
    return telMobile;
}

public void setTelMobile(String telMobile) {
    String oldTelMobile = this.telMobile;
    this.telMobile = telMobile;
    changeSupport.firePropertyChange("telMobile", oldTelMobile, telMobile);
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    String oldEmail = this.email;
    this.email = email;
    changeSupport.firePropertyChange("email", oldEmail, email);
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    String oldCity = this.city;
    this.city = city;
    changeSupport.firePropertyChange("city", oldCity, city);
}

public String getArea() {
    return area;
}

public void setArea(String area) {
    String oldArea = this.area;
    this.area = area;
    changeSupport.firePropertyChange("area", oldArea, area);
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    String oldAddress = this.address;
    this.address = address;
    changeSupport.firePropertyChange("address", oldAddress, address);
}

public Short getAddressNum() {
    return addressNum;
}

public void setAddressNum(Short addressNum) {
    Short oldAddressNum = this.addressNum;
    this.addressNum = addressNum;
    changeSupport.firePropertyChange("addressNum", oldAddressNum, addressNum);
}

public String getLongitude() {
    return longitude;
}

public void setLongitude(String longitude) {
    String oldLongitude = this.longitude;
    this.longitude = longitude;
    changeSupport.firePropertyChange("longitude", oldLongitude, longitude);
}

```

```

public String getLatitude() {
    return latitude;
}

public void setLatitude(String latitude) {
    String oldLatitude = this.latitude;
    this.latitude = latitude;
    changeSupport.firePropertyChange("latitude", oldLatitude, latitude);
}

public byte[] getPhoto() {
    return photo;
}

public void setPhoto(byte[] photo) {
    byte[] oldPhoto = this.photo;
    this.photo = photo;
    changeSupport.firePropertyChange("photo", oldPhoto, photo);
}

public String getNotes() {
    return notes;
}

public void setNotes(String notes) {
    String oldNotes = this.notes;
    this.notes = notes;
    changeSupport.firePropertyChange("notes", oldNotes, notes);
}

public Nomoi getNomoi() {
    return nomoi;
}

public void setNomoi(Nomoi nomoi) {
    Nomoi oldNomoi = this.nomoi;
    this.nomoi = nomoi;
    changeSupport.firePropertyChange("nomoi", oldNomoi, nomoi);
}

public List<Customersproducts> getCustomersproductsList() {
    return customersproductsList;
}

public void setCustomersproductsList(List<Customersproducts> customersproductsList) {
    this.customersproductsList = customersproductsList;
}

/*public Collection<Customersproducts> getCustomersproductsCollection() {
    return CustomersproductsCollection;
}

public void setCustomersproductsCollection(Collection<Customersproducts>
CustomersproductsCollection) {
    this.CustomersproductsCollection = CustomersproductsCollection;
}*/

@Override
public int hashCode() {
    int hash = 0;
    hash += (idcustomers != null ? idcustomers.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Customers)) {
        return false;
    }
    Customers other = (Customers) object;
    if ((this.idcustomers == null && other.idcustomers != null) || (this.idcustomers != null &&
!this.idcustomers.equals(other.idcustomers))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "kfapplication.Customers[idcustomers=" + idcustomers + "]";
}

public void addPropertyChangeListener(PropertyChangeListener listener) {
    changeSupport.addPropertyChangeListener(listener);
}

```

```

public void removePropertyChangeListener(PropertyChangeListener listener) {
    changeSupport.removePropertyChangeListener(listener);
}

```

## Customersproducts.java

```

}
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.io.Serializable;
import java.util.Date;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;

/**
 * Κλάση του πίνακα customersproducts της βάσης
 * @author Kostas
 */
@Entity
@Table(name = "customersproducts", catalog = "klf", schema = "")
@NamedQueries({
    @NamedQuery(name = "Customersproducts.findAll", query = "SELECT c FROM Customersproducts c"),
    @NamedQuery(name = "Customersproducts.findByIdcustomersproducts", query = "SELECT c FROM Customersproducts c WHERE c.idcustomersproducts = :idcustomersproducts"),
    @NamedQuery(name = "Customersproducts.findByPurchasedate", query = "SELECT c FROM Customersproducts c WHERE c.purchasedate = :purchasedate"),
    @NamedQuery(name = "Customersproducts.findByLocation", query = "SELECT c FROM Customersproducts c WHERE c.location = :location"),
    @NamedQuery(name = "Customersproducts.findByServicemonths", query = "SELECT c FROM Customersproducts c WHERE c.servicemonths = :servicemonths");})
public class Customersproducts implements Serializable {
    @Transient
    private PropertyChangeSupport changeSupport = new PropertyChangeSupport(this);
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idcustomersproducts", nullable = false)
    private Integer idcustomersproducts;
    @Column(name = "purchasedate")
    @Temporal(TemporalType.DATE)
    private Date purchasedate;
    @Lob
    @Column(name = "note", length = 65535)
    private String note;
    @Lob
    @Column(name = "info", length = 65535)
    private String info;
    @Column(name = "location", length = 45)
    private String location;
    @Lob
    @Column(name = "photo")
    private byte[] photo;
    @Basic(optional = false)
    @Column(name = "servicemonths", nullable = false)
    private short servicemonths;
    @Lob
    @Column(name = "filters", length = 65535)
    private String filters;
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "custproductsid")
    private List<Sintirisis> sintirisisList;
    @JoinColumn(name = "productsid", referencedColumnName = "idproducts", nullable = false)
    @ManyToOne(optional = false)
    private Products products;

```

```

@JoinColumn(name = "customerid", referencedColumnName = "idcustomers", nullable = false)
@ManyToOne(optional = false)
private Customers customerId;

```

```

public Customersproducts() {
}

public Customersproducts(Integer idcustomersproducts) {
    this.idcustomersproducts = idcustomersproducts;
}

public Customersproducts(Integer idcustomersproducts, Products products, short servicemonths) {
    this.idcustomersproducts = idcustomersproducts;
    this.products = products;
    this.servicemonths = servicemonths;
}

public Integer getIdcustomersproducts() {
    return idcustomersproducts;
}

public void setIdcustomersproducts(Integer idcustomersproducts) {
    Integer oldIdcustomersproducts = this.idcustomersproducts;
    this.idcustomersproducts = idcustomersproducts;
    changeSupport.firePropertyChange("idcustomersproducts", oldIdcustomersproducts, idcustomersproducts);
}

public Date getPurchasedate() {
    return purchasedate;
}

public void setPurchasedate(Date purchasedate) {
    Date oldPurchasedate = this.purchasedate;
    this.purchasedate = purchasedate;
    changeSupport.firePropertyChange("purchasedate", oldPurchasedate, purchasedate);
}

public String getNote() {
    return note;
}

public void setNote(String note) {
    String oldNote = this.note;
    this.note = note;
    changeSupport.firePropertyChange("note", oldNote, note);
}

public String getInfo() {
    return info;
}

public void setInfo(String info) {
    String oldInfo = this.info;
    this.info = info;
    changeSupport.firePropertyChange("info", oldInfo, info);
}

public String getLocation() {
    return location;
}

public void setLocation(String location) {
    String oldLocation = this.location;
    this.location = location;
    changeSupport.firePropertyChange("location", oldLocation, location);
}

public byte[] getPhoto() {
    return photo;
}

public void setPhoto(byte[] photo) {
    byte[] oldPhoto = this.photo;
    this.photo = photo;
    changeSupport.firePropertyChange("photo", oldPhoto, photo);
}

public short getServicemonths() {
    return servicemonths;
}

public void setServicemonths(short servicemonths) {
    short oldServicemonths = this.servicemonths;
    this.servicemonths = servicemonths;
    changeSupport.firePropertyChange("servicemonths", oldServicemonths, servicemonths);
}

```

```

public String getfilters() {
    return filters;
}

public void setfilters(String filters) {
    String oldfilters = this.filters;
    this.filters = filters;
    changeSupport.firePropertyChange("filters", oldfilters, filters);
}

public List<Sintiris> getSintirisList() {
    return sintirisList;
}

public void setSintirisList(List<Sintiris> sintirisList) {
    this.sintirisList = sintirisList;
}

public Products getProducts() {
    return products;
}

public void setProducts(Products products) {
    Products oldProducts = this.products;
    this.products = products;
    changeSupport.firePropertyChange("products", oldProducts, products);
}

public Customers getCustomerId() {
    return customerId;
}

public void setCustomerId(Customers customerId) {
    Customers oldCustomerId = this.customerId;
    this.customerId = customerId;
    changeSupport.firePropertyChange("customerId", oldCustomerId, customerId);
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (idcustomersproducts != null ? idcustomersproducts.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Customersproducts)) {
        return false;
    }
    Customersproducts other = (Customersproducts) object;
    if ((this.idcustomersproducts == null && other.idcustomersproducts != null) ||
(this.idcustomersproducts != null && !this.idcustomersproducts.equals(other.idcustomersproducts))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "klfapplication.Customersproducts[idcustomersproducts=" + idcustomersproducts + "]";
}

public void addPropertyChangeListener(PropertyChangeListener listener) {
    changeSupport.addPropertyChangeListener(listener);
}

public void removePropertyChangeListener(PropertyChangeListener listener) {
    changeSupport.removePropertyChangeListener(listener);
}
}

```



## CustomersProductsEditor.java

```

package klffapplication;

import java.awt.Color;
import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;

/**
 * Το παράθυρο προσθήκης / επεξεργασίας συσκευών πελατών
 * @author Kostas
 */
public class CustomersProductsEditor extends javax.swing.JDialog {

    /** Creates new form CustomersProductsEditor */
    public CustomersProductsEditor(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();

        protected Customersproducts currentBoughtRecord;

        public Customersproducts getCurrentBoughtRecord() {
            return currentBoughtRecord;
        }

        public void setCurrentBoughtRecord(Customersproducts currentBoughtRecord) {

            Customersproducts oldRecord = this.currentBoughtRecord;
            this.currentBoughtRecord = currentBoughtRecord;
            if (getCurrentBoughtRecord().getProducts() != null) {
                int
row=dbcom.getProductIndex(getCurrentBoughtRecord().getProducts().getIdproducts()-1;
                productTable.setRowSelectionInterval(row,row);
            }
            else {productTable.setRowSelectionInterval(0,0);}
            propertyChangeSupport.firePropertyChange("currentBoughtRecord", oldRecord,
currentBoughtRecord);
        }

        //Gia apo8hkeush h akyrwsh kataxorisis
        protected boolean productConfirmed;

        public boolean isProductConfirmed() {
            return productConfirmed;
        }

        public void setProductConfirmed(boolean productConfirmed) {
            this.productConfirmed = productConfirmed;
        }

        private PropertyChangeSupport propertyChangeSupport = new PropertyChangeSupport(this);

        public void addPropertyChangeListener(PropertyChangeListener listener) {
            propertyChangeSupport.addPropertyChangeListener(listener);
        }

        public void removePropertyChangeListener(PropertyChangeListener listener) {
            propertyChangeSupport.removePropertyChangeListener(listener);
        }

        /** This method is called from within the constructor to
         * initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is
         * always regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
        // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
        private void initComponents() {

            bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

            klfPUEntityManager = java.beans.Beans.isDesignTime() ? null :
javax.persistence.Persistence.createEntityManagerFactory("klfPU").createEntityManager();

            productsQuery = java.beans.Beans.isDesignTime() ? null :
klfPUEntityManager.createQuery("SELECT p FROM Products p");
            productsList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
productsQuery.getResultList();
            dateVerifier1 = new klffapplication.DateVerifier();
            rowSorterToStringConverter1 = new klffapplication.RowSorterToStringConverter();
            titleLabel = new javax.swing.JLabel();
            helpLabel = new javax.swing.JLabel();
            productScrollPane = new javax.swing.JScrollPane();
            productTable = new javax.swing.JTable();
            searchTextField = new javax.swing.JTextField();
            searchLabel = new javax.swing.JLabel();
            jSeparator2 = new javax.swing.JSeparator();
            productInfoLabel = new javax.swing.JLabel();
            productInfoScrollPane = new javax.swing.JScrollPane();
            productInfoTextArea = new javax.swing.JTextArea();
            jSeparator1 = new javax.swing.JSeparator();
            dateHelpLabel = new javax.swing.JLabel();
            dateLabel = new javax.swing.JLabel();
            dateFormattedTextField = new javax.swing.JFormattedTextField();
            locationLabel = new javax.swing.JLabel();
            locationTextField = new javax.swing.JTextField();
            monthsLabel = new javax.swing.JLabel();
            monthsTextField = new javax.swing.JTextField();
            filtersLabel = new javax.swing.JLabel();
            filtersTextField = new javax.swing.JTextField();
            infoLabel = new javax.swing.JLabel();
            infoScrollPane = new javax.swing.JScrollPane();
            infoTextArea = new javax.swing.JTextArea();
            notesLabel = new javax.swing.JLabel();
            notesScrollPane = new javax.swing.JScrollPane();
            notesTextArea = new javax.swing.JTextArea();
            cancelButton = new javax.swing.JButton();
            saveButton = new javax.swing.JButton();

            rowSorterToStringConverter1.setTable(productTable);

            setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
            org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(klffapplication.KlfApplication.class).getContext().get
ResourceMap(CustomersProductsEditor.class);
            setTitle(resourceMap.getString("Form.title")); // NOI18N
            setModal(true);
            setName("Form"); // NOI18N

            titleLabel.setFont(resourceMap.getFont("titleLabel.font")); // NOI18N
            titleLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
            titleLabel.setText(resourceMap.getString("titleLabel.text")); // NOI18N
            titleLabel.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
            titleLabel.setName("titleLabel"); // NOI18N

            helpLabel.setText(resourceMap.getString("helpLabel.text")); // NOI18N
            helpLabel.setName("helpLabel"); // NOI18N

            productScrollPane.setName("productScrollPane"); // NOI18N

            productTable.setName("productTable"); // NOI18N

            org.jdesktop.swingbinding.JTableBinding jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBindi
ng.UpdateStrategy.READ_WRITE, productsList, productTable);
            org.jdesktop.swingbinding.JTableBinding.ColumnBinding columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${idproducts}"));
            columnBinding.setColumnName("Idproducts");
            columnBinding.setColumnClass(String.class);
            columnBinding.setEditable(false);
            columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${code}"));
            columnBinding.setColumnName("Code");
            columnBinding.setColumnClass(String.class);
            columnBinding.setEditable(false);
            columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${description}"));
            columnBinding.setColumnName("Description");
            columnBinding.setColumnClass(String.class);
            columnBinding.setEditable(false);
            columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${info}"));
            columnBinding.setColumnName("Info");
            columnBinding.setColumnClass(String.class);
            columnBinding.setEditable(false);
            bindingGroup.addBinding(jTableBinding);
            jTableBinding.bind();org.jdesktop.beansbinding.Binding binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentBoughtRecord.products}"), productTable,
org.jdesktop.beansbinding.BeanProperty.create("selectedElement_IGNORE_ADJUSTING"));
            bindingGroup.addBinding(binding);

```

```

        productScrollPane.setViewportView(productTable);
        productTable.getColumnModel().getColumn(0).setMaxWidth(50);

productTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("jTable1.colu
mnModel.title2")); // NOI18N
        productTable.getColumnModel().getColumn(1).setPreferredWidth(100);
        productTable.getColumnModel().getColumn(1).setMaxWidth(150);

productTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("jTable1.colu
mnModel.title0")); // NOI18N

productTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("jTable1.colu
mnModel.title1")); // NOI18N

productTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("productTable
.columnModel.title3")); // NOI18N

        searchTextField.setName("searchTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, productTable,
org.jdesktop.beansbinding.ELProperty.create("{rowSorter}"), searchTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        binding.setConverter(rowSorterToStringConverter1);
        bindingGroup.addBinding(binding);

        searchLabel.setText(resourceMap.getString("searchLabel.text")); // NOI18N
        searchLabel.setName("searchLabel"); // NOI18N

jSeparator2.setName("jSeparator2"); // NOI18N

productInfoLabel.setText(resourceMap.getString("productInfoLabel.text")); // NOI18N
productInfoLabel.setName("productInfoLabel"); // NOI18N

productInfoScrollPane.setName("productInfoScrollPane"); // NOI18N

productInfoTextArea.setColumns(20);
productInfoTextArea.setEditable(false);
productInfoTextArea.setFont(resourceMap.getFont("productInfoTextArea.font")); // NOI18N
productInfoTextArea.setRows(3);
productInfoTextArea.setName("productInfoTextArea"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, productTable,
org.jdesktop.beansbinding.ELProperty.create("{selectedElement.info}"), productInfoTextArea,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

productInfoScrollPane.setViewportView(productInfoTextArea);

jSeparator1.setName("jSeparator1"); // NOI18N

dateHelpLabel.setFont(resourceMap.getFont("dateHelpLabel.font")); // NOI18N
dateHelpLabel.setText(resourceMap.getString("dateHelpLabel.text")); // NOI18N
dateHelpLabel.setName("dateHelpLabel"); // NOI18N

dateLabel.setText(resourceMap.getString("dateLabel.text")); // NOI18N
dateLabel.setName("dateLabel"); // NOI18N

dateFormattedTextField.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.DateFormatter(java.text.DateFormat.getDateInstance(java.text.DateFormat.SHORT)
));
        dateFormattedTextField.setInputVerifier(dateVerifier1);
        dateFormattedTextField.setName("dateFormattedTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.purchasedate}"),
dateFormattedTextField, org.jdesktop.beansbinding.BeanProperty.create("value"));
        bindingGroup.addBinding(binding);

locationLabel.setText(resourceMap.getString("locationLabel.text")); // NOI18N
locationLabel.setName("locationLabel"); // NOI18N

locationTextField.setName("locationTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.location}"), locationTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

monthsLabel.setText(resourceMap.getString("monthsLabel.text")); // NOI18N

monthsLabel.setName("monthsLabel"); // NOI18N

monthsTextField.setName("monthsTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.servicemonths}"),
monthsTextField, org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

filtersLabel.setText(resourceMap.getString("filtersLabel.text")); // NOI18N
filtersLabel.setName("filtersLabel"); // NOI18N

filtersTextField.setName("filtersTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.filters}"), filtersTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

infoLabel.setText(resourceMap.getString("infoLabel.text")); // NOI18N
infoLabel.setName("infoLabel"); // NOI18N

infoScrollPane.setName("infoScrollPane"); // NOI18N

infoTextArea.setColumns(20);
infoTextArea.setRows(3);
infoTextArea.setName("infoTextArea"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.info}"), infoTextArea,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

infoScrollPane.setViewportView(infoTextArea);

notesLabel.setText(resourceMap.getString("notesLabel.text")); // NOI18N
notesLabel.setName("notesLabel"); // NOI18N

notesScrollPane.setName("notesScrollPane"); // NOI18N

notesTextArea.setColumns(20);
notesTextArea.setRows(3);
notesTextArea.setName("notesTextArea"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("{currentBoughtRecord.note}"), notesTextArea,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

notesScrollPane.setViewportView(notesTextArea);

cancelButton.setText(resourceMap.getString("cancelButton.text")); // NOI18N
cancelButton.setName("cancelButton"); // NOI18N
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cancelCProduct(evt);
    }
});

saveButton.setText(resourceMap.getString("saveButton.text")); // NOI18N
saveButton.setName("saveButton"); // NOI18N
saveButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        saveCProduct(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(299, 299, 299)
            .addComponent(titleLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 192,
Short.MAX_VALUE)
            .addGap(300, 300, 300))
        .addGroup(layout.createSequentialGroup()
            .addGap(613, 613, 613)
            .addComponent(saveButton)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))

```

```

        .addComponent(cancelButton)
        .addContainerGap())
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
        .addGap(10, 10, 10)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(productScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 771,
Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(productInfoLabel)
            .addComponent(searchLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
156, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(productInfoScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 651, Short.MAX_VALUE)))
        .addContainerGap()
        .addComponent(jSeparator1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 791, Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(helpLabel)
            .addContainerGap(532, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(dateLabel)
                .addComponent(monthsLabel)
                .addComponent(infoLabel)
                .addGroup(layout.createSequentialGroup())
                    .addGap(48, 48, 48)
                    .addComponent(notesLabel)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(dateHelpLabel)
                .addComponent(notesScrollPane, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 654, Short.MAX_VALUE)
                .addComponent(infoScrollPane, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 654, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(dateFormattedTextField)
            .addComponent(monthsTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
115, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(30, 30, 30)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(filtersLabel)
        .addComponent(locationLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(filtersTextField, javax.swing.GroupLayout.DEFAULT_SIZE, 432,
Short.MAX_VALUE)
        .addComponent(locationTextField, javax.swing.GroupLayout.DEFAULT_SIZE,
432, Short.MAX_VALUE)))
        .addContainerGap()
        .addComponent(jSeparator2, javax.swing.GroupLayout.DEFAULT_SIZE, 791,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(titleLabel)
            .addGap(7, 7, 7)
            .addComponent(helpLabel)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(productScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(9, 9, 9)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(searchLabel)
                .addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(2, 2, 2)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(productInfoLabel)
                .addComponent(productInfoScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(dateLabel)
        .addComponent(monthsLabel)
        .addComponent(infoLabel)
        .addGroup(layout.createSequentialGroup())
            .addGap(48, 48, 48)
            .addComponent(notesLabel)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(locationTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(locationLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(filtersTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(filtersLabel)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(infoLabel)
            .addComponent(infoScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(12, 12, 12)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(notesLabel)
            .addComponent(notesScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cancelButton)
            .addComponent(saveButton))
        .addContainerGap()
    );

    bindingGroup.bind();

    pack();
} // </editor-fold> // GEN-END: initComponents

private void saveCProduct(java.awt.event.ActionEvent evt) { // GEN-FIRST:event_saveCProduct
    setProductConfirmed(true);
    setVisible(false);
} // GEN-LAST:event_saveCProduct

private void cancelCProduct(java.awt.event.ActionEvent evt) { // GEN-FIRST:event_cancelCProduct
    setProductConfirmed(false);
    setVisible(false);
} // GEN-LAST:event_cancelCProduct

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(CustomersProductsEditor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(CustomersProductsEditor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CustomersProductsEditor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CustomersProductsEditor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        CustomersProductsEditor dialog = new CustomersProductsEditor(new
javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton cancelButton;
private javax.swing.JFormattedTextField dateFormattedTextField;
private javax.swing.JLabel dateHelpLabel;
private javax.swing.JLabel dateLabel;
private klfapplication.DateVerifier dateVerifier1;
private javax.swing.JLabel filtersLabel;
private javax.swing.JTextField filtersTextField;
private javax.swing.JLabel helpLabel;
private javax.swing.JLabel infoLabel;
private javax.swing.JScrollPane infoScrollPane;
private javax.swing.JTextArea infoTextArea;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.persistence.EntityManager klfPUEntityManager;
private javax.swing.JLabel locationLabel;
private javax.swing.JTextField locationTextField;
private javax.swing.JLabel monthsLabel;
private javax.swing.JTextField monthsTextField;
private javax.swing.JLabel notesLabel;
private javax.swing.JScrollPane notesScrollPane;
private javax.swing.JTextArea notesTextArea;
private javax.swing.JLabel productInfoLabel;
private javax.swing.JScrollPane productInfoScrollPane;
private javax.swing.JTextArea productInfoTextArea;
private javax.swing.JScrollPane productScrollPane;
private javax.swing.JTable productTable;
private java.util.List<klfapplication.Products> productsList;
private javax.persistence.Query productsQuery;
private klfapplication.RowSorterToStringConverter rowSorterToStringConverter1;
private javax.swing.JButton saveButton;
private javax.swing.JLabel searchLabel;
private javax.swing.JTextField searchTextField;
private javax.swing.JLabel titleLabel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration//GEN-END:variables
}

```

## DateVerifier.java

```
package klfapplication;

import javax.swing.InputVerifier;
import javax.swing.JComponent;
import javax.swing.JFormattedTextField;
import javax.swing.JOptionPane;

/**
 * Κλάση για έλεγχο ημερομηνίας
 * @author Kostas
 */
public class DateVerifier extends InputVerifier {

    @Override
    public boolean shouldYieldFocus(JComponent input) {
        boolean inputOK = verify(input);

        if (inputOK) {
            return true;
        }

        String failedVerificationMessage = "Η Ημερομηνία πρέπει να είναι της μορφής H/M/XXXX.
        Για παράδειγμα: 27/11/2011";
        JOptionPane.showMessageDialog(null, //no owner frame
            failedVerificationMessage,
            "Λανθασμένη μορφή Ημερομηνίας", //title
            JOptionPane.WARNING_MESSAGE);

        //Reinstall the input verifier.
        input.setInputVerifier(this);
        return false;
    }

    public boolean verify(JComponent input) {
        if (!(input instanceof JFormattedTextField))
            return true;
        return ((JFormattedTextField) input).isEditValid();
    }
}
```

## dbcom.java

```
import java.sql.*;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.DefaultTableModel.*;

/**
 * Κλάση για εργασίες που εκτελούνται με τη βάση δεδομένων πέρα των βασικών πινάκων
 * @author Kostas
 */
public class dbcom {

    public static String DBname = "klf";
    public static String DBip = "127.0.0.1";
    public static String DBuser = "klfuser";
    public static String DBpass = "klfpass";
    public static Connection conn;
    public static String msg;

    /**
     * Σύνδεση με τη βάση
     */
    public static Statement connect() throws SQLException {

        try {

            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://" + DBip + "/" + DBname +
            "?useUnicode=true&characterEncoding=utf8", DBuser, DBpass);
            System.out.println("connection ok");
            Statement stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_UPDATABLE);

            return (stmt);
        } catch (Exception ex) {
            System.out.println(ex);
            JOptionPane.showMessageDialog(null, ex.toString(), "Σφάλμα Σύνδεσης με Β.Δ. σφάλμα 01",
            2);
            System.exit(0);
            return (null);
        }
    }

    /**
     * Αποσύνδεση απο τη βάση
     */
    public static void disconnect() {

        try {
            conn.close();
            System.out.println("disconnected");
        } catch (SQLException ex) {
            Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    /**
     * Επιστρέφει το ResultSet απο ένα Query
     */
    public static ResultSet getRS(String q) {
        ResultSet rs;
        try {
            rs = connect().executeQuery(q);
            //disconnect();
            return (rs);
        } catch (SQLException ex) {
            Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
            return (null);
        }
    }
}
```

```
/**
 * Κάνει Backup στη Βάση
 */

public void backup() {
    int n = JOptionPane.showConfirmDialog(null, "Το αρχείο του Backup θα αποθηκευθεί στο c:",
    "Warning",
    JOptionPane.OK_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE, null);
    if (n == JOptionPane.OK_OPTION) {
        try {
            Runtime rt = Runtime.getRuntime();
            rt.exec("cmd.exe /c start C:\\dbbackup");
            System.exit(0);
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

/**
 * Διαβάζει και δίνει τις συντεταγμένες απο το id κάποιου πελάτη
 */
public static void getgps(Object data[], int custid) {
    try {
        ResultSet rs = getRS("SELECT customers.longitude, customers.latitude from customers where
        customers.idcustomers="+custid);
        while (rs.next()) {
            data[0] = rs.getString("latitude");
            data[1] = rs.getString("longitude");
        }
    }
    catch (SQLException ex) {
        Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
        disconnect();
    }
}

/**
 * Επιστρέφει T/F αν ένα προϊόν είναι καταχωρημένο σε κάποιο πελάτη ή όχι
 */
public static boolean getProductSold(int id) {
    boolean index=false;

    try {
        int idt=0;
        ResultSet rs = getRS("SELECT customersproducts.productsid FROM customersproducts");

        while (idt!=id & rs.next()){
            idt =rs.getInt("productsid");

            if (idt==id) {index=true;}
        }

        disconnect();
    }

    catch (SQLException ex) {

        Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
        disconnect();
    }
    return index;
}

/**
 * Επιστρέφει το index του πίνακα products με βάση το idproduct
 */
public static int getProductIndex(int id) {
    int index= -1;

    try {
        int count = 0, idt=0;
        ResultSet rs = getRS("SELECT products.idproducts FROM products");

        while (idt!=id & rs.next()){
            idt =rs.getInt("idproducts");
            count++;
            if (idt==id) {index=count;}
        }

        disconnect();
    }
}
```

```

catch (SQLException ex) {

Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
disconnect();
}
return index;
}

/**
 * Αδιάζει τον πίνακα που δίνουμε
 */
public static void emptyTable(JTable ResTable) {
DefaultTableModel myModel = (DefaultTableModel) ResTable.getModel();
int num = myModel.getRowCount();
for (int i = 0; i < num; i++) {
myModel.removeRow(0);
}
}

/**
 * Γεμίζει τον πίνακα εκτύπωσης των επόμενων εργασιών συντήρησης για συγκεκριμένο νομό
 */
public static void fillPrintTable(JTable ResTable, int nomos) {
int count = 0;
String nomo = Integer.toString(nomos);

if (nomos==1){nomo="customers.nomos";}

DefaultTableModel ourModel = (DefaultTableModel) ResTable.getModel();
emptyTable(ResTable);
try {
ResultSet rs = getRS(
"(SELECT distinct "
+ "customers.name,"
+ "customers.tel_home,"
+ "customers.tel_work,"
+ "CONCAT(customers.address, ', ', customers.address_num) as fulladdress,"
+ "nomoi.name AS nomos,"
+ "products.description,"
+ "customersproducts.location,"
+ "customersproducts.filters,"
+ "NULL as notes,"
+ "NULL as antalaktika,"
+ "NULL as lastservice,"
+ "customersproducts.purchaseatdate + INTERVAL customersproducts.servicemonths
MONTH AS nextservice,"
+ "NULL as cost"
+ " FROM "
+ "nomoi,"
+ "customers,"
+ "customersproducts,"
+ "products,"
+ "sintirisis"
+ " WHERE "
+ "customers.nomos=nomoi.idnomoi"
+ " and "
+ "customers.nomos=" + nomo
+ " and "
+ "customers.idcustomers=customersproducts.customerid"
+ " and "
+ "customersproducts.productsid=products.idproducts"
+ " and not exists"
+ " (SELECT "
+ "sintirisis.custproductsid"
+ " FROM "
+ "sintirisis"
+ " WHERE "
+ "customersproducts.idcustomersproducts=sintirisis.custproductsid"
+ " and "
+ "sintirisis.repair!=true)"
+ "UNION"
+ "(SELECT "
+ "customers.name, "
+ "customers.tel_home,"
+ "customers.tel_work,"
+ "CONCAT(customers.address, ', ', customers.address_num) as fulladdress,"
+ "nomoi.name AS nomos,"
+ "products.description,"
+ "customersproducts.location,"
+ "customersproducts.filters,"
+ "sintirisis.notes, "
+ "sintirisis.antalaktika, "
+ "sintirisis.hmeromhnia as lastservice, "
+ "sintirisis.hmeromhnia + INTERVAL customersproducts.servicemonths MONTH AS
nextservice, "
+ "sintirisis.cost"
+ " FROM "
+ "nomoi, "
+ "customers, "
+ "customersproducts, "
+ "products, "
+ "sintirisis"
+ " WHERE "
+ "customers.nomos=nomoi.idnomoi"
+ " and "
+ "customers.nomos=" + nomo
+ " and "
+ "customers.idcustomers=customersproducts.customerid"
+ " and "
+ "customersproducts.productsid=products.idproducts"
+ " and "
+ "customersproducts.idcustomersproducts=sintirisis.custproductsid"
+ " and "
+ "sintirisis.hmeromhnia in"
+ " (SELECT "
+ "max(sintirisis.hmeromhnia)"
+ " from "
+ "sintirisis"
+ " where "
+ "customersproducts.idcustomersproducts=sintirisis.custproductsid"
+ " and "
+ "sintirisis.repair!=true))"
+ " ORDER BY nextservice"
);

while (rs.next()) {count++;}
rs.beforeFirst();
Object data[] = new Object[count][13];
count=0;
while (rs.next()) {

data[count][0] = rs.getString("name");
data[count][1] = rs.getString("tel_home");
data[count][2] = rs.getString("tel_work");
data[count][3] = rs.getString("fulladdress");
data[count][4] = rs.getString("nomos");
data[count][5] = rs.getString("description");
data[count][6] = rs.getString("location");
data[count][7] = rs.getString("filters");
data[count][8] = rs.getString("notes");
data[count][9] = rs.getString("antalaktika");
data[count][10] = rs.getDate("lastservice");
data[count][11] = rs.getDate("nextservice");
data[count][12] = rs.getInt("cost");

ourModel.addRow(data[count]);
count++;
}
ResTable.setModel(ourModel);
ResTable.setColumnSelectionAllowed(false);
disconnect();

} catch (SQLException ex) {
Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
disconnect();
}
}

/**
 * Γεμίζει τον πίνακα των επόμενων εργασιών συντήρησης για συγκεκριμένο νομό
 */
public static void fillServiceTable(JTable ResTable, int nomos) {
int i = 0, count = 0;
Calendar cal = Calendar.getInstance();
String nomo = Integer.toString(nomos);

if (nomos==1){nomo="customers.nomos";}

DefaultTableModel ourModel = (DefaultTableModel) ResTable.getModel();
emptyTable(ResTable);
try {
ResultSet rs = getRS(
"(SELECT distinct "
+ "customers.idcustomers,"
+ "customers.name,"
+ "customers.tel_home,"
+ "customers.tel_work,"
+ "nomoi.name AS nomos,"
+ "products.description,"
+ "customersproducts.idcustomersproducts,"
+ "customersproducts.location,"

```

```

+ "customersproducts.filters,"
+ "NULL as idsintiris, "
+ "NULL as notes,"
+ "NULL as antalaktika,"
+ "customersproducts.purchasedate,"
+ "NULL as lastservice,"
+ "customersproducts.purchasedate + INTERVAL customersproducts.servicemonths
MONTH AS nextservice,"
+ "NULL as paid,"
+ "NULL as cost"
+ " FROM "
+ "nomoi,"
+ "customers,"
+ "customersproducts,"
+ "products,"
+ "sintiris"
+ " WHERE "
+ "customers.nomos=nomoi.idnomoi"
+ " and "
+ "customers.nomos=" + nomo
+ " and "
+ "customers.idcustomers=customersproducts.customerid"
+ " and "
+ "customersproducts.productsid=products.idproducts"
+ " and not exists"
+ " (SELECT "
+ "sintiris.custproductsid"
+ " FROM "
+ "sintiris"
+ " WHERE "
+ "customersproducts.idcustomersproducts=sintiris.custproductsid"
+ " and "
+ "sintiris.repair!=true)"
+ "UNION"
+ "(SELECT "
+ "customers.idcustomers, "
+ "customers.name, "
+ "customers.tel_home,"
+ "customers.tel_work,"
+ "nomoi.name AS nomos,"
+ "products.description,"
+ "customersproducts.idcustomersproducts, "
+ "customersproducts.location,"
+ "customersproducts.filters,"
+ "sintiris.idsintiris, "
+ "sintiris.notes, "
+ "sintiris.antalaktika, "
+ "customersproducts.purchasedate, "
+ "sintiris.hmeromhnia as lastservice, "
+ "sintiris.hmeromhnia + INTERVAL customersproducts.servicemonths MONTH AS
nextservice, "
+ "sintiris.paid, "
+ "sintiris.cost"
+ " FROM "
+ "nomoi, "
+ "customers, "
+ "customersproducts, "
+ "products, "
+ "sintiris"
+ " WHERE "
+ "customers.nomos=nomoi.idnomoi"
+ " and "
+ "customers.nomos=" + nomo
+ " and "
+ "customers.idcustomers=customersproducts.customerid"
+ " and "
+ "customersproducts.productsid=products.idproducts"
+ " and "
+ "customersproducts.idcustomersproducts=sintiris.custproductsid"
+ " and "
+ "sintiris.hmeromhnia in"
+ " (SELECT "
+ "max(sintiris.hmeromhnia)"
+ " from "
+ "sintiris"
+ " where "
+ "customersproducts.idcustomersproducts=sintiris.custproductsid"
+ " and "
+ "sintiris.repair!=true)"
+ " ORDER BY nextservice"
);

while (rs.next()) {count++;}
rs.beforeFirst();
Object data[] = new Object[count][17];
count=0;

while (rs.next()) {
    data[count][0] = rs.getInt("idcustomers");
    data[count][1] = rs.getString("name");
    data[count][2] = rs.getString("tel_home");
    data[count][3] = rs.getString("tel_work");
    data[count][4] = rs.getString("nomos");
    data[count][5] = rs.getString("description");
    data[count][6] = rs.getString("location");
    data[count][7] = rs.getString("filters");
    data[count][8] = rs.getString("notes");
    data[count][9] = rs.getString("antalaktika");
    data[count][10] = rs.getDate("purchasedate");
    data[count][11] = rs.getDate("lastservice");
    data[count][12] = rs.getDate("nextservice");
    data[count][13] = rs.getBoolean("paid");
    data[count][14] = rs.getInt("cost");
    data[count][15] = rs.getString("idcustomersproducts");
    data[count][16] = rs.getString("idsintiris");

    ourModel.addRow(data[count]);
    count++;
}
ResTable.setModel(ourModel);
ResTable.setColumnSelectionAllowed(false);
disconnect();

} catch (SQLException ex) {
    Logger.getLogger(dbcom.class.getName()).log(Level.SEVERE, null, ex);
    disconnect();
}
}
}
}

```



## KIfAboutBox.java

```

package kifapplication;

import org.jdesktop.application.Action;

/**
 * Το παράθυρο περι εφαρμογής
 * @author Kostas
 */
public class KIfAboutBox extends javax.swing.JDialog {

    public KIfAboutBox(java.awt.Frame parent) {
        super(parent);
        initComponents();
        getRootPane().setDefaultButton(closeButton);
    }

    @Action public void closeAboutBox() {
        dispose();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        closeButton = new javax.swing.JButton();
        javax.swing.JLabel appTitleLabel = new javax.swing.JLabel();
        javax.swing.JLabel versionLabel = new javax.swing.JLabel();
        javax.swing.JLabel appVersionLabel = new javax.swing.JLabel();
        javax.swing.JLabel vendorLabel = new javax.swing.JLabel();
        javax.swing.JLabel appVendorLabel = new javax.swing.JLabel();
        javax.swing.JLabel homepageLabel = new javax.swing.JLabel();
        javax.swing.JLabel appHomepageLabel = new javax.swing.JLabel();
        javax.swing.JLabel appDescLabel = new javax.swing.JLabel();
        javax.swing.JLabel imageLabel = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(kifapplication.KIfApplication.class).getContext().get
ResourceMap(KIfAboutBox.class);
        setTitle(resourceMap.getString("title")); // NOI18N
        setModal(true);
        setName("aboutBox"); // NOI18N
        setResizable(false);

        javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(kifapplication.KIfApplication.class).getContext().get
ActionMap(KIfAboutBox.class, this);
        closeButton.setAction(actionMap.get("closeAboutBox")); // NOI18N
        closeButton.setName("closeButton"); // NOI18N

        appTitleLabel.setFont(appTitleLabel.getFont().deriveFont(appTitleLabel.getFont().getStyle() |
java.awt.Font.BOLD, appTitleLabel.getFont().getSize()+4));
        appTitleLabel.setText(resourceMap.getString("Application.title")); // NOI18N
        appTitleLabel.setName("appTitleLabel"); // NOI18N

        versionLabel.setFont(versionLabel.getFont().deriveFont(versionLabel.getFont().getStyle() |
java.awt.Font.BOLD));
        versionLabel.setText(resourceMap.getString("versionLabel.text")); // NOI18N
        versionLabel.setName("versionLabel"); // NOI18N

        appVersionLabel.setText(resourceMap.getString("Application.version")); // NOI18N
        appVersionLabel.setName("appVersionLabel"); // NOI18N

        vendorLabel.setFont(vendorLabel.getFont().deriveFont(vendorLabel.getFont().getStyle() |
java.awt.Font.BOLD));
        vendorLabel.setText(resourceMap.getString("vendorLabel.text")); // NOI18N
        vendorLabel.setName("vendorLabel"); // NOI18N

        appVendorLabel.setText(resourceMap.getString("Application.vendor")); // NOI18N
        appVendorLabel.setName("appVendorLabel"); // NOI18N

        homepageLabel.setFont(homepageLabel.getFont().deriveFont(homepageLabel.getFont().getStyle() |
java.awt.Font.BOLD));
        homepageLabel.setText(resourceMap.getString("homepageLabel.text")); // NOI18N
        homepageLabel.setName("homepageLabel"); // NOI18N

        appHomepageLabel.setText(resourceMap.getString("Application.homepage")); // NOI18N
appHomepageLabel.setName("appHomepageLabel"); // NOI18N

        appDescLabel.setText(resourceMap.getString("appDescLabel.text")); // NOI18N
        appDescLabel.setName("appDescLabel"); // NOI18N

        imageLabel.setIcon(resourceMap.getIcon("imageLabel.icon")); // NOI18N
        imageLabel.setName("imageLabel"); // NOI18N

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(closeButton)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appTitleLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(versionLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appVersionLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(vendorLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appVendorLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(homepageLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appHomepageLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appDescLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(imageLabel)
                )
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(closeButton)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appTitleLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(versionLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appVersionLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(vendorLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appVendorLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(homepageLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appHomepageLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(appDescLabel)
                    .addGap(10, 10, 0, 0)
                    .addComponent(imageLabel)
                )
        );

        pack();
    } // </editor-fold> //GEN-END: initComponents

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JButton closeButton;
    // End of variables declaration //GEN-END:variables
}

```

## KlfApplication.java

```
package klfapplication;

import org.jdesktop.application.Application;
import org.jdesktop.application.SingleFrameApplication;

/**
 * The main class of the application.
 */
public class KlfApplication extends SingleFrameApplication {

    /**
     * At startup create and show the main frame of the application.
     */
    @Override protected void startup() {
        show(new KlfView(this));
    }

    /**
     * This method is to initialize the specified window by injecting resources.
     * Windows shown in our application come fully initialized from the GUI
     * builder, so this additional configuration is not needed.
     */
    @Override protected void configureWindow(java.awt.Window root) {
    }

    /**
     * A convenient static getter for the application instance.
     * @return the instance of KlfApplication
     */
    public static KlfApplication getApplication() {
        return Application.getInstance(KlfApplication.class);
    }

    /**
     * Main method launching the application.
     */
    public static void main(String[] args) {
        launch(KlfApplication.class, args);
    }
}
```

## KlfView.java

```

package klfapplication;

//import org.openide.awt.HtmlBrowser.URLDisplayer;
import java.awt.Desktop;
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import org.jdesktop.application.Task;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import javax.swing.undo.RollbackException;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import org.jdesktop.beansbinding.AbstractBindingListener;
import org.jdesktop.beansbinding.Binding;
import org.jdesktop.beansbinding.PropertyChangeEvent;

/**
 * Το κεντρικό παράθυρο της εφαρμογής
 * @author Kostas
 */
public class KlfView extends FrameView {

    public KlfView(SingleFrameApplication app) {
        super(app);

        initComponents();

        // status bar initialization - message timeout, idle icon and busy animation, etc
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                statusBarMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                busyIconIndex = (busyIconIndex + 1) % busyIcons.length;
                statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
            }
        });
        idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
        statusAnimationLabel.setIcon(idleIcon);
        progressBar.setVisible(false);

        // connecting action tasks to status bar via TaskMonitor
        TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());
        taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
            public void propertyChange(java.beans.PropertyChangeEvent evt) {
                String propertyName = evt.getPropertyName();
                if ("started".equals(propertyName)) {
                    if (!busyIconTimer.isRunning()) {
                        statusAnimationLabel.setIcon(busyIcons[0]);
                        busyIconIndex = 0;
                        busyIconTimer.start();
                    }
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(true);
                } else if ("done".equals(propertyName)) {
                    busyIconTimer.stop();
                    statusAnimationLabel.setIcon(idleIcon);
                    progressBar.setVisible(false);

                    progressBar.setValue(0);
                } else if ("message".equals(propertyName)) {
                    String text = (String)(evt.getNewValue());
                    statusBarMessageLabel.setText((text == null) ? "" : text);
                    messageTimer.restart();
                } else if ("progress".equals(propertyName)) {
                    int value = (Integer)(evt.getNewValue());
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(false);
                    progressBar.setValue(value);
                }
            }
        });

        //Gemisma pinaka syntirisewn
        dbcom.fillServiceTable(servicesTable, nomoiComboBox.getSelectedIndex()+1);

        // tracking table selection
        customerTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("customerRecordSelected", !isCustomerRecordSelected(),
                        isCustomerRecordSelected());
                }
            }
        );

        customersProdTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("custProdRecordSelected", !isCustProdRecordSelected(),
                        isCustProdRecordSelected());
                }
            }
        );

        sintirisisTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("sintirisisRecordSelected", !isSintirisisRecordSelected(),
                        isSintirisisRecordSelected());
                }
            }
        );

        productsShowTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("productRecordSelected", !isProductRecordSelected(),
                        isProductRecordSelected());
                }
            }
        );

        servicesTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("servicesSelected", !isServicesSelected(), isServicesSelected());
                }
            }
        );

        // tracking changes to save
        bindingGroup.addBindingListener(new AbstractBindingListener() {
            @Override
            public void targetChanged(Binding binding, PropertyChangeEvent event) {
                // save action observes saveNeeded property
                setSaveNeeded(true);
                setSaveProductNeeded(true);
            }
        });

        // have a transaction started
        entityManager.getTransaction().begin();
    }

    public boolean isSaveNeeded() {
        return saveNeeded;
    }

    private void setSaveNeeded(boolean saveNeeded) {
        if (saveNeeded != this.saveNeeded) {
            this.saveNeeded = saveNeeded;
            firePropertyChange("saveNeeded", !saveNeeded, saveNeeded);
        }
    }

    public boolean isSaveProductNeeded() {

```

```

return saveProductNeeded;
}

private void setSaveProductNeeded(boolean saveProductNeeded) {
    if (saveProductNeeded != this.saveProductNeeded) {
        this.saveProductNeeded = saveProductNeeded;
        firePropertyChange("saveProductNeeded", !saveProductNeeded, saveProductNeeded);
    }
}

public boolean isServicesSelected() {
    return servicesTable.getSelectedRow() != -1;
}

public boolean isCustomerRecordSelected() {
    return customerTable.getSelectedRow() != -1;
}

public boolean isCustProdRecordSelected() {
    return customersProdTable.getSelectedRow() != -1;
}

public boolean isSintirisRecordSelected() {
    return sintirisTable.getSelectedRow() != -1;
}

public boolean isProductRecordSelected() {
    return productsShowTable.getSelectedRow() != -1;
}

//public void autoSelect(){
//
//}

if(customersProdTable.getRowCount()<=0){customersProdTable.setRowSelectionInterval(0,0);}
//}

@Action
public void newRecord() {
    klfapplication.Customers C = new klfapplication.Customers();
    entityManager.persist(C);
    list.add(C);
    int row = customerTable.getRowCount() - 1;
    customerTable.setRowSelectionInterval(row, row);
    customerTable.scrollRectToVisible(customerTable.getCellRect(row, 0, true));
    setSaveNeeded(true);
    /*anoigma ths formas stoixeiwv tou pelath*/
    JFrame mainFrame = KIfApplication.getApplication().getMainFrame();
    CustomerEditor ce = new CustomerEditor(mainFrame, false);
    ce.setCurrentRecord(C);
    ce.setVisible(true);

    /*Gia apo8hkeush h akvrosh apo th forma pelath */
    if (ce.isCustomerConfirmed()) {
        save().run();
    } else {
        refresh().run();
    }
}

@Action(enabledProperty = "customerRecordSelected")
public void deleteRecord() {
    int n = JOptionPane.showConfirmDialog(null, "Θέλετε σίγουρα να διαγράψετε τον επιλεγμένο
πελάτη;", "Warning",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE, null);
    if (n == JOptionPane.YES_OPTION) {
        int[] selected = customerTable.getSelectedRows();
        List<klfapplication.Customers> toRemove = new
ArrayList<klfapplication.Customers>(selected.length);
        for (int idx = 0; idx < selected.length; idx++) {
            klfapplication.Customers C =
list.get(customerTable.convertRowIndexToModel(selected[idx]));
            toRemove.add(C);
            entityManager.remove(C);
        }
        list.removeAll(toRemove);
        save().run();
    } else {
        refresh().run();
    }
}

@Action(enabledProperty = "customerRecordSelected")
public void newDetailRecord() {
    int index = customerTable.getSelectedRow();
    klfapplication.Customers C = list.get(customerTable.convertRowIndexToModel(index));

```

```

Collection<klfapplication.Customersproducts> cs = C.getCustomersproductsList();
//Collection<klfapplication.Customersproducts> cs = C.getCustomersproductsCollection();
if (cs == null) {
    cs = new LinkedList<klfapplication.Customersproducts>();
    C.setCustomersproductsList((List)cs);
    //C.setCustomersproductsCollection(cs);
}
klfapplication.Customersproducts cp = new klfapplication.Customersproducts();
cp.setPurchasedate(new java.util.Date()); //shmerinh hmeromhnia sth forma
entityManager.persist(cp);
cp.setCustomerId(C);
cs.add(cp);
customerTable.clearSelection();
customerTable.setRowSelectionInterval(index, index);
//int row = cs.size()-1;
int row = customersProdTable.getRowCount() - 1;
customersProdTable.setRowSelectionInterval(row, row);
customersProdTable.scrollRectToVisible(customersProdTable.getCellRect(row, 0, true));
setSaveNeeded(true);
/*anoigma ths formas agorasmenwn proioutwn*/
JFrame mainFrame = KIfApplication.getApplication().getMainFrame();
CustomersProductsEditor cpe = new CustomersProductsEditor(mainFrame, false);
cpe.setCurrentBoughtRecord(cp);
cpe.setVisible(true);
// Gia apo8hkeush h akvrosh apo th forma suskeyhs pelath
if (cpe.isProductConfirmed()) {
    save().run();
} else {
    refresh().run();
}
}

@Action(enabledProperty = "custProdRecordSelected")
public void deleteDetailRecord() {
    Object[] options = {"OK", "Cancel"};
    int n = JOptionPane.showConfirmDialog(null, "Θέλετε σίγουρα να διαγράψετε την επιλεγμένη
συσκευή του πελάτη;", "Warning",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE, null);
    if (n == JOptionPane.YES_OPTION) {
        int index = customerTable.getSelectedRow();
        klfapplication.Customers C = list.get(customerTable.convertRowIndexToModel(index));
        List<klfapplication.Customersproducts> os = C.getCustomersproductsList();

        int[] selected = customersProdTable.getSelectedRows();
        List<klfapplication.Customersproducts> toRemove = new
ArrayList<klfapplication.Customersproducts>(selected.length);
        for (int idx = 0; idx < selected.length; idx++) {
            selected[idx] = customersProdTable.convertRowIndexToModel(selected[idx]);
            int count = 0;
            Iterator<klfapplication.Customersproducts> iter = os.iterator();
            while (count++ < selected[idx]) {
                iter.next();
            }
            klfapplication.Customersproducts o = iter.next();
            toRemove.add(o);
            entityManager.remove(o);
        }
        os.removeAll(toRemove);
        customerTable.clearSelection();
        customerTable.setRowSelectionInterval(index, index);
        list.removeAll(toRemove);
        save().run();
    } else {
        refresh().run();
    }
}

@Action(enabledProperty = "custProdRecordSelected")
public void newSintirisRecord() {
    int index = customerTable.getSelectedRow();
    klfapplication.Customers C = list.get(customerTable.convertRowIndexToModel(index));
    List<klfapplication.Customersproducts> cpl = C.getCustomersproductsList();

    int index2 = customersProdTable.getSelectedRow();
    klfapplication.Customersproducts cp =
cpl.get(customersProdTable.convertRowIndexToModel(index2));
//Collection<klfapplication.Sintiris> ss = cp.getSintirisList();
List<klfapplication.Sintiris> ss = cp.getSintirisList();
if (ss == null) {
    ss = new LinkedList<klfapplication.Sintiris>();
    cp.setSintirisList((List)ss);
}
klfapplication.Sintiris s = new klfapplication.Sintiris();
entityManager.persist(s);
s.setCustproductsid(cp);

```

```

ss.add(s);
customersProdTable.clearSelection();
customersProdTable.setRowSelectionInterval(index2, index2);
int row = sintirisisTable.getRowCount() - 1;
sintirisisTable.setRowSelectionInterval(row, row);
sintirisisTable.scrollRectToVisible(sintirisisTable.getCellRect(row, 0, true));
setSaveNeeded(true);
//anoigma ths formas sintirisewn
JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
SintirisisEditor se = new SintirisisEditor(mainFrame, false);
s.setHmeromhnia(new java.util.Date());
s.setPaid(false);
s.setRepair(false);
//System.out.println("to s tou currentSintirisisRecord: -----" + s.getHmeromhnia());
se.setCurrentSintirisisRecord(s);
se.setVisible(true);

if (se.isSintirisisConfirmed()) {
    save().run();
} else {
    refresh().run();
}
}

@Action(enabledProperty = "sintirisisRecordSelected")
public void deleteSintirisisRecord() {
    Object[] options = {"OK", "Cancel"};
    int n = JOptionPane.showConfirmDialog(null,
        "Θέλετε σίγουρα να διαγράψετε την επιλεγμένη καταχώρηση συντήρησης;", "Warning",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE, null);
    if (n == JOptionPane.YES_OPTION) {

        int index = customerTable.getSelectedRow();
        klfapplication.Customers C = list.get(customerTable.convertRowIndexToModel(index));
        List<klfapplication.Customersproducts> cpl = C.getCustomersproductsList();

        int index2 = customersProdTable.getSelectedRow();
        klfapplication.Customersproducts c =
        cpl.get(customersProdTable.convertRowIndexToModel(index2));
        List<klfapplication.Sintirisis> os = c.getSintirisisList();
        int[] selected = sintirisisTable.getSelectedRows();
        List<klfapplication.Sintirisis> toRemove = new
        ArrayList<klfapplication.Sintirisis>(selected.length);
        for (int idx = 0; idx < selected.length; idx++) {
            selected[idx] = sintirisisTable.convertRowIndexToModel(selected[idx]);
            int count = 0;
            Iterator<klfapplication.Sintirisis> iter = os.iterator();
            while (count++ < selected[idx]) {
                iter.next();
            }
            klfapplication.Sintirisis s = iter.next();
            toRemove.add(s);
            entityManager.remove(s);
        }
        os.removeAll(toRemove);
        customersProdTable.clearSelection();
        customersProdTable.setRowSelectionInterval(index2, index2);
        list.removeAll(toRemove);
        save().run();
    } else {
        refresh().run();
    }
}

@Action
public void newProductRecord() {
    klfapplication.Products p = new klfapplication.Products();
    entityManager.persist(p);
    productsList1.add(p);
    int row = productsShowTable.getRowCount() - 1;
    productsShowTable.setRowSelectionInterval(row, row);
    productsShowTable.scrollRectToVisible(productsShowTable.getCellRect(row, 0, true));
    setSaveProductNeeded(true);
}

@Action(enabledProperty = "productRecordSelected")
public void deleteProductRecord() {

    Object b = productsShowTable.getValueAt(productsShowTable.getSelectedRow(), 0);
    int id = Integer.parseInt(b.toString());

    if (dbcom.getProductSold(id))
    {
        JOptionPane.showConfirmDialog(null,
            "Το προϊόν δεν μπορεί να διαγραφεί γιατί έχει αγορασθεί απο πελάτη!",
            "Warning", JOptionPane.PLAIN_MESSAGE, JOptionPane.ERROR_MESSAGE, null);
    } else {
        Object[] options = {"OK", "Cancel"};
        int n = JOptionPane.showConfirmDialog(null, "Θέλετε σίγουρα να διαγράψετε το προϊόν;",
            "Warning",
            JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE, null);
        if (n == JOptionPane.YES_OPTION) {
            int[] selected = productsShowTable.getSelectedRows();
            List<klfapplication.Products> toRemove = new
            ArrayList<klfapplication.Products>(selected.length);
            for (int idx=0; idx<selected.length; idx++) {
                klfapplication.Products p =
                productsList1.get(productsShowTable.convertRowIndexToModel(selected[idx]));
                toRemove.add(p);
                entityManager.remove(p);
            }
            productsList1.removeAll(toRemove);
            setSaveProductNeeded(true);
            saveProduct().run();
        } else {
            //refreshProduct().run();
        }
    }
}

@Action(enabledProperty = "saveNeeded")
public Task save() {
    return new SaveTask(getApplication());
}

private class SaveTask extends Task {
    SaveTask(org.jdesktop.application.Application app) {
        super(app);
    }
    @Override protected Void doInBackground() {
        try {
            entityManager.getTransaction().commit();
            entityManager.getTransaction().begin();
        } catch (RollbackException rex) {
            rex.printStackTrace();
            entityManager.getTransaction().begin();
            List<klfapplication.Customers> merged = new
            ArrayList<klfapplication.Customers>(list.size());
            for (klfapplication.Customers C : list) {
                merged.add(entityManager.merge(C));
            }
            list.clear();
            list.addAll(merged);
        }
        return null;
    }
    @Override protected void finished() {
        setSaveNeeded(false);
    }
}

@Action(enabledProperty = "saveProductNeeded")
public Task saveProduct() {
    return new SaveProductTask(getApplication());
}

private class SaveProductTask extends Task {
    SaveProductTask(org.jdesktop.application.Application app) {
        super(app);
    }
    @Override protected Void doInBackground() {
        try {
            entityManager.getTransaction().commit();
            entityManager.getTransaction().begin();
        } catch (RollbackException rex) {
            rex.printStackTrace();
            entityManager.getTransaction().begin();
            List<klfapplication.Products> merged = new
            ArrayList<klfapplication.Products>(productsList1.size());
            for (klfapplication.Products p : productsList1) {
                merged.add(entityManager.merge(p));
            }
            productsList1.clear();
            productsList1.addAll(merged);
        }
        return null;
    }
    @Override protected void finished() {
}
}

```

```

        setSaveProductNeeded(false);
    }
}

@Action
public Task refreshProduct() {
    return new RefreshProductTask(getApplication());
}

private class RefreshProductTask extends Task {
    RefreshProductTask(org.jdesktop.application.Application app) {
        super(app);
    }
    @SuppressWarnings("unchecked")
    @Override protected void doInBackground() {

        setProgress(0, 0, 4);
        setMessage("Rolling back the current changes...");
        setProgress(1, 0, 4);
        entityManager.getTransaction().rollback();
        setProgress(2, 0, 4);
        setMessage("Starting a new transaction...");
        entityManager.getTransaction().begin();
        setProgress(3, 0, 4);
        setMessage("Fetching new data...");
        java.util.Collection data = productsQuery1.getResultList();
        for (Object entity : data) {
            entityManager.refresh(entity);
        }
        setProgress(4, 0, 4);

        productsList1.clear();
        productsList1.addAll(data);
        return null;
    }
    @Override protected void finished() {
        setMessage("Done.");
        setSaveProductNeeded(false);
    }
}

/**
 * An example action method showing how to create asynchronous tasks
 * (running on background) and how to show their progress. Note the
 * artificial 'Thread.sleep' calls making the task long enough to see the
 * progress visualization - remove the sleeps for real application.
 */
@Action
public Task refresh() {
    return new RefreshTask(getApplication());
}

private class RefreshTask extends Task {
    RefreshTask(org.jdesktop.application.Application app) {
        super(app);
    }
    @SuppressWarnings("unchecked")
    @Override protected void doInBackground() {
        setProgress(0, 0, 4);
        setMessage("Rolling back the current changes...");
        setProgress(1, 0, 4);
        entityManager.getTransaction().rollback();
        setProgress(2, 0, 4);
        setMessage("Starting a new transaction...");
        entityManager.getTransaction().begin();
        setProgress(3, 0, 4);
        setMessage("Fetching new data...");
        java.util.Collection data = query.getResultList();
        for (Object entity : data) {
            entityManager.refresh(entity);
            setProgress(4, 0, 4);
            list.clear();
            list.addAll(data);
        }
        return null;
    }
    @Override protected void finished() {
        setMessage("Done.");
        setSaveNeeded(false);
    }
}

@Action
public void showAboutBox() {
    if (aboutBox == null) {
        JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
        aboutBox = new KlfAboutBox(mainFrame);
        aboutBox.setLocationRelativeTo(mainFrame);
    }
}

}
KlfApplication.getApplication().show(aboutBox);
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
private void initComponents() {

    bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

    mainPanel = new javax.swing.JPanel();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    servicesPanel = new javax.swing.JPanel();
    servicesScrollPane = new javax.swing.JScrollPane();
    servicesTable = new javax.swing.JTable();
    nomoiLabel = new javax.swing.JLabel();
    refreshButton = new javax.swing.JButton();
    nomoiComboBox = new javax.swing.JComboBox();
    printButton = new javax.swing.JButton();
    mapButton = new javax.swing.JButton();
    customProdPanel = new javax.swing.JPanel();
    customersScrollPane = new javax.swing.JScrollPane();
    customerTable = new javax.swing.JTable();
    newCustomerButton = new javax.swing.JButton();
    deleteCustomerButton = new javax.swing.JButton();
    editCustomerButton = new javax.swing.JButton();
    customersProdScrollPane = new javax.swing.JScrollPane();
    customersProdTable = new javax.swing.JTable();
    deleteCustProdButton = new javax.swing.JButton();
    newCustProdButton = new javax.swing.JButton();
    editCustProdButton = new javax.swing.JButton();
    customersLabel = new javax.swing.JLabel();
    custProdLabel = new javax.swing.JLabel();
    jSeparator3 = new javax.swing.JSeparator();
    jSeparator4 = new javax.swing.JSeparator();
    sintirisisLabel = new javax.swing.JLabel();
    sintirisisScrollPane = new javax.swing.JScrollPane();
    sintirisisTable = new javax.swing.JTable();
    addSintirisisButton = new javax.swing.JButton();
    deleteSintirisisButton = new javax.swing.JButton();
    editSintirisisButton = new javax.swing.JButton();
    searchTextField = new javax.swing.JTextField();
    searchLabel = new javax.swing.JLabel();
    productsPanel = new javax.swing.JPanel();
    productsLabel = new javax.swing.JLabel();
    productsShowScrollPane = new javax.swing.JScrollPane();
    productsShowTable = new javax.swing.JTable();
    deleteProductButton = new javax.swing.JButton();
    saveProductButton = new javax.swing.JButton();
    addProductButton = new javax.swing.JButton();
    refreshProductButton = new javax.swing.JButton();
    codeProductLabel = new javax.swing.JLabel();
    codeProductTextField = new javax.swing.JTextField();
    modelProductLabel = new javax.swing.JLabel();
    modelProductTextField = new javax.swing.JTextField();
    infoProductLabel = new javax.swing.JLabel();
    infoProductScrollPane = new javax.swing.JScrollPane();
    infoProductTextArea = new javax.swing.JTextArea();
    menuBar = new javax.swing.JMenuBar();
    javax.swing.JMenu fileMenu = new javax.swing.JMenu();
    backupMenuItem = new javax.swing.JMenuItem();
    restoreMenuItem = new javax.swing.JMenuItem();
    jSeparator1 = new javax.swing.JSeparator();
    javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
    javax.swing.JMenu helpMenu = new javax.swing.JMenu();
    manualMenuItem = new javax.swing.JMenuItem();
    javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
    statusPanel = new javax.swing.JPanel();
    javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
    statusMessageLabel = new javax.swing.JLabel();
    statusAnimationLabel = new javax.swing.JLabel();
    progressBar = new javax.swing.JProgressBar();
    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
ResourceMap(KlfView.class);
    entityManager = java.beans.Beans.isDesignTime() ? null :
javax.persistence.Persistence.createEntityManagerFactory(resourceMap.getString("entityManager.pers
istenceUnit")).createEntityManager(); // NOI18N
    query = java.beans.Beans.isDesignTime() ? null :
entityManager.createQuery(resourceMap.getString("query.query")); // NOI18N
    list = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
org.jdesktop.observablecollections.ObservableCollections.observableList(query.getResultList());
    sintirisisQuery = java.beans.Beans.isDesignTime() ? null : entityManager.createQuery("SELECT
s FROM Sintirisis s");
}

```

```

        sintirisisList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
org.jdesktop.observablecollections.ObservableCollections.observableList(sintirisisQuery.getResultList
());
        productsQuery = java.beans.Beans.isDesignTime() ? null : entityManager.createQuery("SELECT
p FROM Products p");
        productsList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
productsQuery.getResultList();
        nomoiQuery = java.beans.Beans.isDesignTime() ? null : entityManager.createQuery("SELECT n
FROM Nomoi n");
        nomoiList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
nomoiQuery.getResultList();
        nomoiListCellRenderer1 = new klfapplication.NomoiListCellRenderer();
        rowSorterToStringConverter1 = new klfapplication.RowSorterToStringConverter();
        productsQuery1 = java.beans.Beans.isDesignTime() ? null :
entityManager.createQuery("SELECT p FROM Products p");
        productsList1 = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
org.jdesktop.observablecollections.ObservableCollections.observableList(productsQuery1.getResultLi
st());
        customersproductsQuery = java.beans.Beans.isDesignTime() ? null :
entityManager.createQuery(resourceMap.getString("customersproductsQuery.query"));
        customersproductsList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
org.jdesktop.observablecollections.ObservableCollections.observableList(customersproductsQuery.get
ResultList());

        mainPanel.setName("mainPanel"); // NOI18N

        jTablebedPane1.setName("jTablebedPane1"); // NOI18N

        servicesPanel.setName("servicesPanel"); // NOI18N

        servicesScrollPane.setBorder(null);
        servicesScrollPane.setName("servicesScrollPane"); // NOI18N

        servicesTable.setAutoCreateRowSorter(true);
        servicesTable.setFont(resourceMap.getFont("servicesTable.font")); // NOI18N
        servicesTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                ,
                new String [] {
                    "ID", "Όνοματεπώνυμο", "Τηλέφωνο Σπ.", "Τηλέφωνο Εργ.", "Νομός", "Μοντέλο Συσκ.",
                    "Τοποθεσία", "Φίλτρα Συσκ.", "Πλήροφο. συντήρ.", "Ανταλλάκ. Τελ. Συντ.", "Αγορά", "Τελ. Συντ.",
                    "Επόμενη Συντ.", "Εξοφλημένο", "Κόστος"
                }
            }
        ) {
            Class[] types = new Class [] {
                java.lang.Integer.class, java.lang.Object.class, java.lang.Object.class, java.lang.Object.class,
                java.lang.Object.class, java.lang.Object.class, java.lang.Object.class, java.lang.Object.class,
                java.lang.Object.class, java.lang.Object.class, java.lang.Object.class, java.lang.Object.class,
                java.lang.Object.class, java.lang.Boolean.class, java.lang.Object.class
            };
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false, false, false, false, false, false, false, false, false
            };
        }

        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    servicesTable.setName("servicesTable"); // NOI18N
    servicesTable.setRowHeight(22);
    servicesScrollPane.setViewportView(servicesTable);

    servicesTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title14")); // NOI18N

    servicesTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title0")); // NOI18N

    servicesTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title1")); // NOI18N

    servicesTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("jTable1.colu
mnModel.title3")); // NOI18N

    servicesTable.getColumnModel().getColumn(4).setHeaderValue(resourceMap.getString("jTable1.colu
mnModel.title2")); // NOI18N

    servicesTable.getColumnModel().getColumn(5).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title4")); // NOI18N

    servicesTable.getColumnModel().getColumn(6).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title5")); // NOI18N

    servicesTable.getColumnModel().getColumn(7).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title6")); // NOI18N

    servicesTable.getColumnModel().getColumn(8).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title7")); // NOI18N

    servicesTable.getColumnModel().getColumn(9).setHeaderValue(resourceMap.getString("servicesTabl
e.columnModel.title8")); // NOI18N

    servicesTable.getColumnModel().getColumn(10).setHeaderValue(resourceMap.getString("servicesTa
ble.columnModel.title9")); // NOI18N

    servicesTable.getColumnModel().getColumn(11).setHeaderValue(resourceMap.getString("servicesTa
ble.columnModel.title10")); // NOI18N

    servicesTable.getColumnModel().getColumn(12).setHeaderValue(resourceMap.getString("servicesTa
ble.columnModel.title11")); // NOI18N

    servicesTable.getColumnModel().getColumn(13).setHeaderValue(resourceMap.getString("servicesTa
ble.columnModel.title12")); // NOI18N

    servicesTable.getColumnModel().getColumn(14).setHeaderValue(resourceMap.getString("servicesTa
ble.columnModel.title13")); // NOI18N

    nomoiLabel.setText(resourceMap.getString("nomoiLabel.text")); // NOI18N
    nomoiLabel.setName("nomoiLabel"); // NOI18N

    refreshButton.setText(resourceMap.getString("refreshButton.text")); // NOI18N
    refreshButton.setName("refreshButton"); // NOI18N
    refreshButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            refreshButtonActionPerformed(evt);
        }
    });

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
ActionMap(KlfView.class, this);
    nomoiComboBox.setAction(actionMap.get("nomosSelected")); // NOI18N
    nomoiComboBox.setName("nomoiComboBox"); // NOI18N
    nomoiComboBox.setRenderer(nomoiListCellRenderer1);

    org.jdesktop.swingbinding.JComboBoxBinding jComboBoxBinding =
org.jdesktop.swingbinding.SwingBindings.createJComboBoxBinding(org.jdesktop.beansbinding.Auto
Binding.UpdateStrategy.READ_WRITE, nomoiList, nomoiComboBox);
    bindingGroup.addBinding(jComboBoxBinding);

    printButton.setAction(actionMap.get("print")); // NOI18N
    printButton.setText(resourceMap.getString("printButton.text")); // NOI18N
    printButton.setName("printButton"); // NOI18N

    mapButton.setAction(actionMap.get("showMapPressed")); // NOI18N
    mapButton.setText(resourceMap.getString("mapButton.text")); // NOI18N
    mapButton.setName("mapButton"); // NOI18N

    javax.swing.GroupLayout servicesPanelLayout = new javax.swing.GroupLayout(servicesPanel);
    servicesPanel.setLayout(servicesPanelLayout);
    servicesPanelLayout.setHorizontalGroup(
        servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(servicesPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addGroup(servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(servicesPanelLayout.createSequentialGroup()
                        .addContainerGap()
                    )
                )
            );
    servicesPanelLayout.setVerticalGroup(
        servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(servicesPanelLayout.createSequentialGroup()
                .addGroup(servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(servicesPanelLayout.createSequentialGroup()
                        .addComponent(servicesScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 1075,
Short.MAX_VALUE)
                        .addGroup(servicesPanelLayout.createSequentialGroup()
                            .addComponent(nomoiLabel)
                            .addGap(18, 18, 18)
                            .addComponent(nomoiComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
115, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(refreshButton)
                            .addGroup(servicesPanelLayout.createSequentialGroup()
                                .addComponent(printButton)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 853,
Short.MAX_VALUE)
                                .addComponent(mapButton))
                            .addContainerGap()
                        )
                )
            );
    servicesPanelLayout.setVerticalGroup(
        servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(servicesPanelLayout.createSequentialGroup()
                .addGroup(servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addGroup(servicesPanelLayout.createSequentialGroup()
                        .addContainerGap()
                    )
                )
            );
    servicesPanelLayout.addGroup(servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addGroup(servicesPanelLayout.createSequentialGroup()
            .addContainerGap()
        )
    );

```

```

        .addComponent(nomoiLabel)
        .addComponent(nomoiComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(refreshButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(servicesScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 540,
Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(servicesPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
)
        .addComponent(printButton)
        .addComponent(mapButton))
        .addContainerGap()
);

jTabbedPane1.addTab(resourceMap.getString("servicesPanel.TabConstraints.tabTitle"),
servicesPanel); // NOI18N

customProdPanel.setName("customProdPanel"); // NOI18N

customersScrollPane.setName("customersScrollPane"); // NOI18N

customerTable.setFont(resourceMap.getFont("customerTable.font")); // NOI18N
customerTable.setName("customerTable"); // NOI18N
customerTable.setRowHeight(22);

customerTable.setSelectionMode(javax.swing.ListSelectionMode.MULTIPLE_INTERVAL_SELECTION);

org.jdesktop.swingbinding.JTableBinding jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, list, customerTable);
org.jdesktop.swingbinding.JTableBinding.ColumnBinding columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${code}"));
columnBinding.setColumnClass(Integer.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${name}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${company}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${telHome}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${telWork}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${telFax}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${telMobile}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${email}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${nomoi.name}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${city}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${area}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);

columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${address}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${addressNum}"));
columnBinding.setColumnClass(Short.class);
columnBinding.setEditable(false);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${notes}"));
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
bindingGroup.addBinding(jTableBinding);
jTableBinding.bind();
customersScrollPane.setViewportView(customerTable);

customerTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("customerTable.columnModel.title0")); // NOI18N

customerTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("customerTable.columnModel.title1")); // NOI18N

customerTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("customerTable.columnModel.title2")); // NOI18N

customerTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("customerTable.columnModel.title3")); // NOI18N

customerTable.getColumnModel().getColumn(4).setHeaderValue(resourceMap.getString("customerTable.columnModel.title4")); // NOI18N

customerTable.getColumnModel().getColumn(5).setHeaderValue(resourceMap.getString("customerTable.columnModel.title5")); // NOI18N

customerTable.getColumnModel().getColumn(6).setHeaderValue(resourceMap.getString("customerTable.columnModel.title6")); // NOI18N

customerTable.getColumnModel().getColumn(7).setHeaderValue(resourceMap.getString("customerTable.columnModel.title7")); // NOI18N

customerTable.getColumnModel().getColumn(8).setHeaderValue(resourceMap.getString("customerTable.columnModel.title8")); // NOI18N

customerTable.getColumnModel().getColumn(9).setHeaderValue(resourceMap.getString("customerTable.columnModel.title9")); // NOI18N

customerTable.getColumnModel().getColumn(10).setHeaderValue(resourceMap.getString("customerTable.columnModel.title10")); // NOI18N

customerTable.getColumnModel().getColumn(11).setHeaderValue(resourceMap.getString("customerTable.columnModel.title11")); // NOI18N

customerTable.getColumnModel().getColumn(12).setHeaderValue(resourceMap.getString("customerTable.columnModel.title12")); // NOI18N

customerTable.getColumnModel().getColumn(13).setHeaderValue(resourceMap.getString("customerTable.columnModel.title13")); // NOI18N

newCustomerButton.setAction(actionMap.get("newRecord")); // NOI18N
newCustomerButton.setText(resourceMap.getString("newCustomerButton.text")); // NOI18N
newCustomerButton.setName("newCustomerButton"); // NOI18N

deleteCustomerButton.setAction(actionMap.get("deleteRecord")); // NOI18N
deleteCustomerButton.setText(resourceMap.getString("deleteCustomerButton.text")); // NOI18N
deleteCustomerButton.setName("deleteCustomerButton"); // NOI18N

editCustomerButton.setAction(actionMap.get("editCustomer")); // NOI18N
editCustomerButton.setText(resourceMap.getString("editCustomerButton.text")); // NOI18N
editCustomerButton.setName("editCustomerButton"); // NOI18N

customersProdScrollPane.setName("customersProdScrollPane"); // NOI18N

customersProdTable.setFont(resourceMap.getFont("customersProdTable.font")); // NOI18N
customersProdTable.setName("customersProdTable"); // NOI18N
customersProdTable.setRowHeight(22);

customersProdTable.setSelectionMode(javax.swing.ListSelectionMode.MULTIPLE_INTERVAL_SELECTION);

org.jdesktop.beansbinding.ELProperty eLProperty =
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.customersproductsList}");
jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, customerTable, eLProperty, customersProdTable);

```



```

columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${products.description}"));
columnBinding.setColumnName("Products.description");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${filters}"));
columnBinding.setColumnName("Filters");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${purchasedate}"));
columnBinding.setColumnName("Purchasedate");
columnBinding.setColumnClass(java.util.Date.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${servicemonths}"));
columnBinding.setColumnName("Servicemonths");
columnBinding.setColumnClass(Short.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${note}"));
columnBinding.setColumnName("Note");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${location}"));
columnBinding.setColumnName("Location");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${info}"));
columnBinding.setColumnName("Info");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
 jTableBinding.setSourceUnreadableValue(java.util.Collections.emptyList());
bindingGroup.addBinding(jTableBinding);
 jTableBinding.bind();
customersProdScrollPane.setViewportViewView(customersProdTable);

customersProdTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("detailTable.columnModel.title6")); // NOI18N

customersProdTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("detailTable.columnModel.title7")); // NOI18N

customersProdTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("detailTable.columnModel.title0")); // NOI18N

customersProdTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("detailTable.columnModel.title4")); // NOI18N

customersProdTable.getColumnModel().getColumn(4).setHeaderValue(resourceMap.getString("detailTable.columnModel.title1")); // NOI18N

customersProdTable.getColumnModel().getColumn(5).setHeaderValue(resourceMap.getString("detailTable.columnModel.title3")); // NOI18N

customersProdTable.getColumnModel().getColumn(6).setHeaderValue(resourceMap.getString("detailTable.columnModel.title2")); // NOI18N

deleteCustProdButton.setAction(actionMap.get("deleteDetailRecord")); // NOI18N
deleteCustProdButton.setText(resourceMap.getString("deleteCustProdButton.text")); // NOI18N
deleteCustProdButton.setName("deleteCustProdButton"); // NOI18N

newCustProdButton.setAction(actionMap.get("newDetailRecord")); // NOI18N
newCustProdButton.setName("newCustProdButton"); // NOI18N

editCustProdButton.setAction(actionMap.get("editBought")); // NOI18N
editCustProdButton.setName("editCustProdButton"); // NOI18N

customersLabel.setText(resourceMap.getString("customersLabel.text")); // NOI18N
customersLabel.setName("customersLabel"); // NOI18N

custProdLabel.setText(resourceMap.getString("custProdLabel.text")); // NOI18N
custProdLabel.setName("custProdLabel"); // NOI18N

jSeparator3.setName("jSeparator3"); // NOI18N

jSeparator4.setName("jSeparator4"); // NOI18N

sintirisLabel.setText(resourceMap.getString("sintirisLabel.text")); // NOI18N
sintirisLabel.setName("sintirisLabel"); // NOI18N

sintirisScrollPane.setName("sintirisScrollPane"); // NOI18N

sintirisTable.setFont(resourceMap.getFont("sintirisTable.font")); // NOI18N
sintirisTable.setName("sintirisTable"); // NOI18N
sintirisTable.setRowHeight(22);

sintirisTable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);

eLProperty = org.jdesktop.beansbinding.ELProperty.create("${selectedElement.sintirisList}");
 jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, customersProdTable, eLProperty, sintirisTable);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${hmeromhnia}"));
columnBinding.setColumnName("Hmeromhnia");
columnBinding.setColumnClass(java.util.Date.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${wra}"));
columnBinding.setColumnName("Wra");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${antalaktika}"));
columnBinding.setColumnName("Antalaktika");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${info}"));
columnBinding.setColumnName("Info");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${notes}"));
columnBinding.setColumnName("Notes");
columnBinding.setColumnClass(String.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cost}"));
columnBinding.setColumnName("Cost");
columnBinding.setColumnClass(Float.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${payed}"));
columnBinding.setColumnName("Payed");
columnBinding.setColumnClass(Boolean.class);
columnBinding.setEditable(false);
columnBinding =
 jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${repair}"));
columnBinding.setColumnName("Repair");
columnBinding.setColumnClass(Boolean.class);
columnBinding.setEditable(false);
 jTableBinding.setSourceUnreadableValue(java.util.Collections.emptyList());
bindingGroup.addBinding(jTableBinding);
 jTableBinding.bind();
sintirisScrollPane.setViewportViewView(sintirisTable);

sintirisTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title0")); // NOI18N

sintirisTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title1")); // NOI18N

sintirisTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title2")); // NOI18N

sintirisTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title3")); // NOI18N

sintirisTable.getColumnModel().getColumn(4).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title4")); // NOI18N

sintirisTable.getColumnModel().getColumn(5).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title5")); // NOI18N

sintirisTable.getColumnModel().getColumn(6).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title6")); // NOI18N

sintirisTable.getColumnModel().getColumn(7).setHeaderValue(resourceMap.getString("sintirisTable.columnModel.title7")); // NOI18N

addSintirisButton.setAction(actionMap.get("newSintirisRecord")); // NOI18N
addSintirisButton.setText(resourceMap.getString("addSintirisButton.text")); // NOI18N
addSintirisButton.setName("addSintirisButton"); // NOI18N

deleteSintirisButton.setAction(actionMap.get("deleteSintirisRecord")); // NOI18N
deleteSintirisButton.setText(resourceMap.getString("deleteSintirisButton.text")); // NOI18N
deleteSintirisButton.setName("deleteSintirisButton"); // NOI18N

```

```

editSintirisButton.setAction(actionMap.get("editSintirisi")); // NOI18N
editSintirisButton.setText(resourceMap.getString("editSintirisButton.text")); // NOI18N
editSintirisButton.setMaximumSize(new java.awt.Dimension(119, 23));
editSintirisButton.setMinimumSize(new java.awt.Dimension(119, 23));
editSintirisButton.setName("editSintirisButton"); // NOI18N
editSintirisButton.setPreferredSize(new java.awt.Dimension(119, 23));

searchTextField.setName("searchTextField"); // NOI18N

org.jdesktop.beansbinding.Binding binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, customerTable,
org.jdesktop.beansbinding.ELProperty.create("${rowSorter}"), searchTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
binding.setConverter(rowSorterToStringConverter1);
bindingGroup.addBinding(binding);

searchLabel.setText(resourceMap.getString("searchLabel.text")); // NOI18N
searchLabel.setName("searchLabel"); // NOI18N

javax.swing.GroupLayout customProdPanelLayout = new
javax.swing.GroupLayout(customProdPanel);
customProdPanel.setLayout(customProdPanelLayout);
customProdPanelLayout.setHorizontalGroup(

customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()

.addGroup(customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)
.addGroup(customProdPanelLayout.createSequentialGroup())
.addComponent(customersLabel)
.addGap(661, 661, 661))
.addGroup(customProdPanelLayout.createSequentialGroup())
.addComponent(customersScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE,
1075, Short.MAX_VALUE)
.addContainerGap()))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
customProdPanelLayout.createSequentialGroup())
.addContainerGap(732, Short.MAX_VALUE)
.addComponent(new CustProdButton)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(editCustProdButton, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(deleteCustProdButton)
.addContainerGap())
.addComponent(jSeparator4, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 1095, Short.MAX_VALUE)
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(sintirisisLabel)
.addContainerGap(1022, Short.MAX_VALUE))
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(sintirisisScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 1075,
Short.MAX_VALUE)
.addContainerGap()
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
customProdPanelLayout.createSequentialGroup())
.addContainerGap(732, Short.MAX_VALUE)
.addComponent(addSintirisButton)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(editSintirisButton, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(deleteSintirisButton)
.addContainerGap())
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(custProdLabel)
.addContainerGap(979, Short.MAX_VALUE))
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(customersProdScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE,
1075, Short.MAX_VALUE)
.addContainerGap()
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(searchLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE, 142,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 518,
Short.MAX_VALUE)
.addComponent(new CustomerButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(editCustomerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(deleteCustomerButton)
.addContainerGap())
.addComponent(jSeparator3, javax.swing.GroupLayout.DEFAULT_SIZE, 1095,
Short.MAX_VALUE)
);
customProdPanelLayout.setVerticalGroup(

customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(customProdPanelLayout.createSequentialGroup())
.addContainerGap()
.addComponent(customersLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(customersScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
Short.MAX_VALUE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(deleteCustomerButton)
.addComponent(new CustomerButton)
.addComponent(editCustomerButton, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(searchLabel)
.addComponent(searchTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(custProdLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(customersProdScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE,
142, Short.MAX_VALUE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(deleteCustProdButton)
.addComponent(new CustProdButton)
.addComponent(editCustProdButton))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jSeparator4, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(sintirisisLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(sintirisisScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
Short.MAX_VALUE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(customProdPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(deleteSintirisButton)
.addComponent(editSintirisButton, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(addSintirisButton))
.addContainerGap()
);

jTabbedPane1.addTab(resourceMap.getString("customProdPanel.TabConstraints.tabTitle"),
customProdPanel); // NOI18N

productsPanel.setName("productsPanel"); // NOI18N

productsLabel.setText(resourceMap.getString("productsLabel.text")); // NOI18N
productsLabel.setName("productsLabel"); // NOI18N

productsShowScrollPane.setName("productsShowScrollPane"); // NOI18N

productsShowTable.setFont(resourceMap.getFont("productsShowTable.font")); // NOI18N
productsShowTable.setName("productsShowTable"); // NOI18N
productsShowTable.setRowHeight(20);

jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, productsList1, productsShowTable);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${idproducts}"));
columnBinding.setColumnName("Idproducts");
columnBinding.setColumnClass(Integer.class);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${code}"));
columnBinding.setColumnName("Code");
columnBinding.setColumnClass(String.class);

```

```

columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${description}"));
columnBinding.setColumnName("Description");
columnBinding.setColumnClass(String.class);
columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${info}"));
columnBinding.setColumnName("Info");
columnBinding.setColumnClass(String.class);
bindingGroup.addBinding(jTableBinding);
jTableBinding.bind();
productsShowScrollPane.setViewportView(productsShowTable);

productsShowTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("productsShowTable.columnModel.title0")); // NOI18N

productsShowTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("productsShowTable.columnModel.title1")); // NOI18N

productsShowTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("productsShowTable.columnModel.title2")); // NOI18N

productsShowTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("productsShowTable.columnModel.title3")); // NOI18N

deleteProductButton.setAction(actionMap.get("deleteProductRecord")); // NOI18N
deleteProductButton.setText(resourceMap.getString("deleteProductButton.text")); // NOI18N
deleteProductButton.setName("deleteProductButton"); // NOI18N

saveProductButton.setAction(actionMap.get("saveProduct")); // NOI18N
saveProductButton.setText(resourceMap.getString("saveProductButton.text")); // NOI18N
saveProductButton.setName("saveProductButton"); // NOI18N

addProductButton.setAction(actionMap.get("newProductRecord")); // NOI18N
addProductButton.setText(resourceMap.getString("addProductButton.text")); // NOI18N
addProductButton.setName("addProductButton"); // NOI18N

refreshProductButton.setAction(actionMap.get("refreshProduct")); // NOI18N
refreshProductButton.setText(resourceMap.getString("refreshProductButton.text")); // NOI18N
refreshProductButton.setName("refreshProductButton"); // NOI18N

codeProductLabel.setText(resourceMap.getString("codeProductLabel.text")); // NOI18N
codeProductLabel.setName("codeProductLabel"); // NOI18N

codeProductTextField.setName("codeProductTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, productsShowTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.code}"), codeProductTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

modelProductLabel.setText(resourceMap.getString("modelProductLabel.text")); // NOI18N
modelProductLabel.setName("modelProductLabel"); // NOI18N

modelProductTextField.setName("modelProductTextField"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, productsShowTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.description}"),
modelProductTextField, org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

infoProductLabel.setText(resourceMap.getString("infoProductLabel.text")); // NOI18N
infoProductLabel.setName("infoProductLabel"); // NOI18N

infoProductScrollPane.setName("infoProductScrollPane"); // NOI18N

infoProductTextArea.setColumns(20);
infoProductTextArea.setRows(5);
infoProductTextArea.setName("infoProductTextArea"); // NOI18N

binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE, productsShowTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.info}"), infoProductTextArea,
org.jdesktop.beansbinding.BeanProperty.create("text"));
bindingGroup.addBinding(binding);

infoProductScrollPane.setViewportView(infoProductTextArea);

javax.swing.GroupLayout productsPanelLayout = new javax.swing.GroupLayout(productsPanel);
productsPanel.setLayout(productsPanelLayout);
productsPanelLayout.setHorizontalGroup(
productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
productsPanelLayout.createSequentialGroup().createSequentialGroup()

.addContainerGap()

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(productsShowScrollPane,
javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 1075, Short.MAX_VALUE)
.addGroup(productsPanelLayout.createSequentialGroup().createSequentialGroup()
.addComponent(refreshProductButton)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(addProductButton)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(saveProductButton)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(deleteProductButton)
.addComponent(productsLabel, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 141, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
productsPanelLayout.createSequentialGroup().createSequentialGroup()

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(infoProductLabel)
.addComponent(codeProductLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(productsPanelLayout.createSequentialGroup().createSequentialGroup()
.addComponent(codeProductTextField,
javax.swing.GroupLayout.PREFERRED_SIZE, 203, javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(13, 13, 13)
.addComponent(modelProductLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(modelProductTextField,
javax.swing.GroupLayout.DEFAULT_SIZE, 728, Short.MAX_VALUE)
.addComponent(infoProductScrollPane,
javax.swing.GroupLayout.DEFAULT_SIZE, 1004, Short.MAX_VALUE))))
.addContainerGap()
);
productsPanelLayout.setVerticalGroup(
productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(productsPanelLayout.createSequentialGroup().createSequentialGroup()
.addContainerGap()
.addComponent(productsLabel)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(productsShowScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 420, Short.MAX_VALUE)
.addGap(18, 18, 18)

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(modelProductTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(codeProductTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(modelProductLabel)
.addComponent(codeProductLabel)
.addGap(18, 18, 18)

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(infoProductLabel)
.addComponent(infoProductScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)

.addGroup(productsPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(deleteProductButton)
.addComponent(saveProductButton)
.addComponent(addProductButton)
.addComponent(refreshProductButton))
.addContainerGap()
);
jTabbedPane1.addTab(resourceMap.getString("productsPanel.TabConstraints.tabTitle"),
productsPanel); // NOI18N

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jTabbedPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 1100, Short.MAX_VALUE)
);
mainPanelLayout.setVerticalGroup(
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jTabbedPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 653,
Short.MAX_VALUE)
);

menuBar.setName("menuBar"); // NOI18N

fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
fileMenu.setName("fileMenu"); // NOI18N

backupMenuItem.setAction(actionMap.get("backupDB")); // NOI18N
backupMenuItem.setText(resourceMap.getString("backupMenuItem.text")); // NOI18N
backupMenuItem.setName("backupMenuItem"); // NOI18N
fileMenu.add(backupMenuItem);

restoreMenuItem.setAction(actionMap.get("restoreDB")); // NOI18N
restoreMenuItem.setText(resourceMap.getString("restoreMenuItem.text")); // NOI18N
restoreMenuItem.setName("restoreMenuItem"); // NOI18N
fileMenu.add(restoreMenuItem);

jSeparator1.setName("jSeparator1"); // NOI18N
fileMenu.add(jSeparator1);

exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
exitMenuItem.setText(resourceMap.getString("exitMenuItem.text")); // NOI18N
exitMenuItem.setName("exitMenuItem"); // NOI18N
fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

helpMenu.setText(resourceMap.getString("helpMenu.text")); // NOI18N
helpMenu.setName("helpMenu"); // NOI18N

manualMenuItem.setText(resourceMap.getString("manualMenuItem.text")); // NOI18N
manualMenuItem.setName("manualMenuItem"); // NOI18N
helpMenu.add(manualMenuItem);

aboutMenuItem.setAction(actionMap.get("showAboutBox")); // NOI18N
aboutMenuItem.setText(resourceMap.getString("aboutMenuItem.text")); // NOI18N
aboutMenuItem.setName("aboutMenuItem"); // NOI18N
helpMenu.add(aboutMenuItem);

menuBar.add(helpMenu);

statusPanel.setName("statusPanel"); // NOI18N

statusPanelSeparator.setName("statusPanelSeparator"); // NOI18N

statusMessageLabel.setName("statusMessageLabel"); // NOI18N

statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
statusAnimationLabel.setName("statusAnimationLabel"); // NOI18N

progressBar.setName("progressBar"); // NOI18N

javax.swing.GroupLayout statusPanelLayout = new javax.swing.GroupLayout(statusPanel);
statusPanel.setLayout(statusPanelLayout);
statusPanelLayout.setHorizontalGroup(
    statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(statusPanelSeparator, javax.swing.GroupLayout.DEFAULT_SIZE, 1100,
Short.MAX_VALUE)
        .addGroup(statusPanelLayout.createSequentialGroup()
            .addComponent(statusMessageLabel)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 930,
Short.MAX_VALUE)
            .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(statusAnimationLabel)
            .addContainerGap())
        );
statusPanelLayout.setVerticalGroup(
    statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(statusPanelLayout.createSequentialGroup()
            .addComponent(statusPanelSeparator, javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(statusMessageLabel)
                .addComponent(statusAnimationLabel)
                .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(3, 3, 3))
        );

nomoiListCellRenderer1.setText(resourceMap.getString("nomoiListCellRenderer1.text")); //
NOI18N
nomoiListCellRenderer1.setName("nomoiListCellRenderer1"); // NOI18N

rowSorterToStringConverter1.setTable(customerTable);

setComponent(mainPanel);
setMenuBar(menuBar);
setStatusBar(statusPanel);

bindingGroup.bind();
} // </editor-fold> // GEN-END: initComponents

private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_refreshButtonActionPerformed

    dbcom.fillServiceTable(servicesTable, nomoiComboBox.getSelectedIndex()+1);
} // GEN-LAST:event_refreshButtonActionPerformed

@Action
public void nomosSelected() {
    dbcom.fillServiceTable(servicesTable, nomoiComboBox.getSelectedIndex()+1);
}

@Action(enabledProperty = "customerRecordSelected")
public void editCustomer() {
    setSaveNeeded(true);
    JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
    CustomerEditor ce = new CustomerEditor(mainFrame, false);

    ce.setCurrentRecord(list.get(customerTable.convertRowIndexToModel(customerTable.getSelectedRo
w())));
    ce.setVisible(true);
    if (ce.isCustomerConfirmed()) {
        save().run();
    } else {
        refresh().run();
    }
}

@Action(enabledProperty = "custProdRecordSelected")
public void editBought() {
    setSaveNeeded(true);
    int index = customerTable.getSelectedRow();
    klfapplication.Customers c = list.get(customerTable.convertRowIndexToModel(index));
    List<klfapplication.Customersproducts> os = c.getCustomersproductsList();
    JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
    CustomersProductsEditor oe = new CustomersProductsEditor(mainFrame, false);
    oe.setCurrentBoughtRecord(os.get(customerProdTable.getSelectedRow()));
    oe.setVisible(true);
    if (oe.isProductConfirmed()) {
        save().run();
    } else {
        refresh().run();
    }
}

@Action(enabledProperty = "sintirisRecordSelected")
public void editSintiris() {
    setSaveNeeded(true);
    int index = customerTable.getSelectedRow();
    klfapplication.Customers c = list.get(customerTable.convertRowIndexToModel(index));
    List<klfapplication.Customersproducts> cpl = c.getCustomersproductsList();
    int index2 = customerProdTable.getSelectedRow();
    klfapplication.Customersproducts cp =
cpl.get(customerProdTable.convertRowIndexToModel(index2));
    List<klfapplication.Sintiris> os = cp.getSintirisList();

    JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
    SintirisEditor se = new SintirisEditor(mainFrame, false);
    se.setCurrentSintirisRecord(os.get(sintirisTable.getSelectedRow()));
    se.setVisible(true);
    if (se.isSintirisConfirmed()) {
        save().run();
    } else {
        refresh().run();
    }
}

@Action
public void backupDB() {
    int n = JOptionPane.showConfirmDialog(null, "Το αρχείο του Backup θα αποθηκευθεί στο
c:/klfbackup", "Warning",
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE, null);

```

```

if (n == JOptionPane.OK_OPTION) {
    try {
        Runtime rt = Runtime.getRuntime();
        rt.exec("cmd.exe /c start C:\\klfbakup\\dbbackup");
        //System.exit(0);
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}

}

@Action
public void restoreDB() {
    int n = JOptionPane.showConfirmDialog(null, "Για την επαναφορά της βάσης δεδομένων
τοποθετήστε το αρχείο"
        + " επαναφοράς με όνομα restore.sql στο φάκελο c:\\klfbakup και πατήστε OK", "Warning",
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE, null);
    if (n == JOptionPane.OK_OPTION) {
        try {
            Runtime rt = Runtime.getRuntime();
            rt.exec("cmd.exe /c start C:\\klfbakup\\dbrestore");
            //System.exit(0);
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

public static void showMap(String source1, String source2) {

    if (source1 == null || source1 == "" || source2 == "" || source2 == null)
    {
        JOptionPane.showConfirmDialog(null, "Δεν Υπάρχουν Συντεταγμένες για Προβολή χάρτη ",
            "Warning", JOptionPane.PLAIN_MESSAGE, JOptionPane.ERROR_MESSAGE, null);
    }
    else {
        try {
            URI uri = new
            URI("http://maps.google.com/maps?daddr="+source1+"&hl=el&t=m&z=17&vpsrc=0");
            //URI uri = new URI("http://maps.google.com/?ll="+source1+"&hl=el&t=m&z=17&vpsrc=0");
            Desktop desktop = null;
            //Test whether Desktop class is supported on the current platform.
            if (Desktop.isDesktopSupported()) {
                // If it's supported, retrieve an instance.
                desktop = Desktop.getDesktop();
            }
            if (desktop != null)
                // Launch the default browser to display a URI.
                desktop.browse(uri);
        }
        catch (IOException e) {
            e.printStackTrace();
        }
        catch (URISyntaxException e) {
            e.printStackTrace();
        }
    }
}

@Action(enabledProperty = "servicesSelected")
public void showMapPressed() {
    Object a = servicesTable.getValueAt(servicesTable.getSelectedRow(), 0);
    int id = Integer.parseInt(a.toString());
    Object gps[] = new Object[2];
    dbcom.getgps(gps, id);
    showMap(gps[0].toString(), gps[1].toString());
}

@Action
public void print() {
    JFrame mainFrame = KlfApplication.getApplication().getMainFrame();
    PrintPage pp = new PrintPage(mainFrame, false);
    pp.setVisible(true);
}

}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton addProductButton;
private javax.swing.JButton addSintirisisButton;
private javax.swing.JMenuItem backupMenuItem;
private javax.swing.JLabel codeProductLabel;
private javax.swing.JTextField codeProductTextField;
private javax.swing.JLabel custProdLabel;
private javax.swing.JPanel customProdPanel;
private javax.swing.JTable customerTable;
private javax.swing.JLabel customersLabel;
private javax.swing.JScrollPane customersProdScrollPane;
private javax.swing.JTable customersProdTable;
private javax.swing.JScrollPane customerscrollPane;
private java.util.List<klfapplication.Customersproducts> customersproductsList;
private javax.persistence.Query customersproductsQuery;
private javax.swing.JButton deleteCustProdButton;
private javax.swing.JButton deleteCustomerButton;
private javax.swing.JButton deleteProductButton;
private javax.swing.JButton deleteSintirisisButton;
private javax.swing.JButton editCustProdButton;
private javax.swing.JButton editCustomerButton;
private javax.swing.JButton editSintirisisButton;
private javax.persistence.EntityManager entityManager;
private javax.swing.JLabel infoProductLabel;
private javax.swing.JScrollPane infoProductScrollPane;
private javax.swing.JTextArea infoProductTextArea;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JTabbedPane jTabbedPane1;
private java.util.List<klfapplication.Customers> list;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuItem manualMenuItem;
private javax.swing.JButton mapButton;
private javax.swing.JMenuBar menuBar;
private javax.swing.JLabel modelProductLabel;
private javax.swing.JTextField modelProductTextField;
private javax.swing.JButton newCustProdButton;
private javax.swing.JButton newCustomerButton;
private javax.swing.JComboBox nomoiComboBox;
private javax.swing.JLabel nomoiLabel;
private java.util.List<klfapplication.Nomoi> nomoiList;
private klfapplication.NomoiListCellRenderer nomoiListCellRenderer1;
private javax.persistence.Query nomoiQuery;
private javax.swing.JButton printButton;
private javax.swing.JLabel productsLabel;
private java.util.List<klfapplication.Products> productsList;
private java.util.List<klfapplication.Products> productsList1;
private javax.swing.JPanel productsPanel;
private javax.persistence.Query productsQuery;
private javax.persistence.Query productsQuery1;
private javax.swing.JScrollPane productsShowScrollPane;
private javax.swing.JTable productsShowTable;
private javax.swing.JProgressBar progressBar;
private javax.persistence.Query query;
private javax.swing.JButton refreshButton;
private javax.swing.JButton refreshProductButton;
private javax.swing.JMenuItem restoreMenuItem;
private klfapplication.RowSorterToStringConverter rowSorterToStringConverter1;
private javax.swing.JButton saveProductButton;
private javax.swing.JLabel searchLabel;
private javax.swing.JTextField searchTextField;
private javax.swing.JPanel servicesPanel;
private javax.swing.JScrollPane servicesScrollPane;
private javax.swing.JTable servicesTable;
private javax.swing.JLabel sintirisisLabel;
private java.util.List<klfapplication.Sintirisis> sintirisisList;
private javax.persistence.Query sintirisisQuery;
private javax.swing.JScrollPane sintirisisScrollPane;
private javax.swing.JTable sintirisisTable;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration//GEN-END:variables

private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
private int busyIconIndex = 0;

private JDialog aboutBox;

private boolean saveNeeded;
private boolean saveProductNeeded;
}

```

## Nomoi.java

```
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Transient;

/**
 * Κλάση του πίνακα nomoi της βάσης
 * @author Kostas
 */
@Entity
@Table(name = "nomoi")
@NamedQueries({
    @NamedQuery(name = "Nomoi.findAll", query = "SELECT n FROM Nomoi n"),
    @NamedQuery(name = "Nomoi.findByIdnomoi", query = "SELECT n FROM Nomoi n WHERE n.idnomoi = :idnomoi"),
    @NamedQuery(name = "Nomoi.findByName", query = "SELECT n FROM Nomoi n WHERE n.name = :name")})
public class Nomoi implements Serializable {
    @Transient
    private PropertyChangeSupport changeSupport = new PropertyChangeSupport(this);
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "idnomoi")
    private Integer idnomoi;
    @Column(name = "name")
    private String name;

    public Nomoi() {
    }

    public Nomoi(Integer idnomoi) {
        this.idnomoi = idnomoi;
    }

    public Integer getIdnomoi() {
        return idnomoi;
    }

    public void setIdnomoi(Integer idnomoi) {
        Integer oldIdnomoi = this.idnomoi;
        this.idnomoi = idnomoi;
        changeSupport.firePropertyChange("idnomoi", oldIdnomoi, idnomoi);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        String oldName = this.name;
        this.name = name;
        changeSupport.firePropertyChange("name", oldName, name);
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idnomoi != null ? idnomoi.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {

        if (!(object instanceof Nomoi)) {
            return false;
        }
        Nomoi other = (Nomoi) object;
        if ((this.idnomoi == null && other.idnomoi != null) || (this.idnomoi != null &&
!this.idnomoi.equals(other.idnomoi))) {
            return false;
        }
        return true;
    }
}
```

## NomoiListCellRendererer.java

```
package klapplication;

import java.awt.Component;
import javax.swing.DefaultListCellRenderer;
import javax.swing.JList;

/**
 * Κλάση για render των combobox με το όνομα νομού
 * @author Kostas
 */
public class NomoiListCellRendererer extends DefaultListCellRenderer {

    @Override
    public Component getListCellRendererComponent(
        JList list, Object value, int index, boolean isSelected, boolean cellHasFocus) {
        super.getListCellRendererComponent(list, value, index, isSelected, cellHasFocus);
        if (value instanceof Nomoi) {
            Nomoi c = (Nomoi) value;
            setText(c.getName());
        }
        return this;
    }
}
```

## PrintPage.java

```
package klfapplication;

import java.awt.Font;
import java.text.MessageFormat;
import javax.swing.JTable;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import org.jdesktop.application.Action;

/**
 * Το παράθυρο εκτύπωσης εργασιών συντήρησης
 * @author Kostas
 */
public class PrintPage extends javax.swing.JDialog {

    /** Creates new form PrintPage */
    public PrintPage(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();

        dbcom.fillPrintTable(printTable, nomoiComboBox.getSelectedIndex()+1);

        printTable.getSelectionModel().addListSelectionListener(
            new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent e) {
                    firePropertyChange("recordSelected", !isRecordSelected(), isRecordSelected());
                }
            }
        );

        // tracking table selection
        public boolean isRecordSelected() {
            return printTable.getSelectedRow() != -1;
        }
        public boolean isColumnSelected() {
            return printTable.getSelectedColumn() != -1;
        }

        /** This method is called from within the constructor to
         * initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is
         * always regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
        // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
        private void initComponents() {
            bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

            klfPUEntityManager = java.beans.Beans.isDesignTime() ? null :
            javax.persistence.Persistence.createEntityManagerFactory("klfPU").createEntityManager();
            nomoiQuery = java.beans.Beans.isDesignTime() ? null :
            klfPUEntityManager.createQuery("SELECT n FROM Nomoi n");
            nomoiList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
            nomoiQuery.getResultList();
            nomoiListCellRenderer1 = new klfapplication.NomoiListCellRenderer();
            printPopupMenu = new javax.swing.JPopupMenu();
            columnRemoveMenuItem = new javax.swing.JMenuItem();
            restoreColumnsMenuItem = new javax.swing.JMenuItem();
            printScrollPane = new javax.swing.JScrollPane();
            printTable = new javax.swing.JTable();
            printButton = new javax.swing.JButton();
            titleLabel = new javax.swing.JLabel();
            nomoiComboBox = new javax.swing.JComboBox();
            nomoiLabel = new javax.swing.JLabel();
            cancelButton = new javax.swing.JButton();

            org.jdesktop.application.ResourceMap resourceMap =
            org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
            ResourceMap(PrintPage.class);
            nomoiListCellRenderer1.setText(resourceMap.getString("nomoiListCellRenderer1.text")); // NOI18N
            nomoiListCellRenderer1.setName("nomoiListCellRenderer1"); // NOI18N

            printPopupMenu.setName("printPopupMenu"); // NOI18N

            javax.swing.ActionMap actionMap =
            org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
            ActionMap(PrintPage.class, this);
            columnRemoveMenuItem.setAction(actionMap.get("removeColumn")); // NOI18N
            columnRemoveMenuItem.setName("columnRemoveMenuItem"); // NOI18N
            printPopupMenu.add(columnRemoveMenuItem);

            restoreColumnsMenuItem.setAction(actionMap.get("restoreColumns")); // NOI18N
            restoreColumnsMenuItem.setText(resourceMap.getString("restoreColumnsMenuItem.text")); // NOI18N
            NOI18N
            restoreColumnsMenuItem.setName("restoreColumnsMenuItem"); // NOI18N
            printPopupMenu.add(restoreColumnsMenuItem);

            setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
            setName("Form"); // NOI18N

            printScrollPane.setName("printScrollPane"); // NOI18N

            printTable.setFont(resourceMap.getFont("printTable.font")); // NOI18N
            printTable.setModel(new javax.swing.table.DefaultTableModel(
                new Object [][] {
                    {null, null, null, null, null, null, null, null, null, null, null, null},
                    {null, null, null, null, null, null, null, null, null, null, null, null},
                    {null, null, null, null, null, null, null, null, null, null, null, null},
                    {null, null, null, null, null, null, null, null, null, null, null, null}
                },
                new String [] {
                    "Όνοματεπώνυμο", "Τηλ. Σπιτ.", "Τηλ. Εργ.", "Διεύθυνση", "Νομός", "Μοντέλο",
                    "Τοποθεσία", "Φίλτρα Συσκ.", "Πληροφ. Συντ", "Ανταλ. Τελ. Συντ.", "Τελ. Συντ.", "Επόμ. Συντ.",
                    "Κόστος"
                }
            ));
            printTable.setComponentPopupMenu(printPopupMenu);
            printTable.setName("printTable"); // NOI18N
            printTable.setRowHeight(40);
            printScrollPane.setViewportView(printTable);
            printTable.getColumnModel().getColumn(0).setMinWidth(150);

            printTable.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title0")); // NOI18N
            printTable.getColumnModel().getColumn(1).setMinWidth(100);

            printTable.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title1")); // NOI18N
            printTable.getColumnModel().getColumn(2).setMinWidth(100);

            printTable.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title2")); // NOI18N
            printTable.getColumnModel().getColumn(3).setMinWidth(150);

            printTable.getColumnModel().getColumn(3).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title3")); // NOI18N
            printTable.getColumnModel().getColumn(4).setMaxWidth(40);

            printTable.getColumnModel().getColumn(4).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title4")); // NOI18N

            printTable.getColumnModel().getColumn(5).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title5")); // NOI18N

            printTable.getColumnModel().getColumn(6).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title6")); // NOI18N

            printTable.getColumnModel().getColumn(7).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title7")); // NOI18N

            printTable.getColumnModel().getColumn(8).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title8")); // NOI18N

            printTable.getColumnModel().getColumn(9).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title9")); // NOI18N

            printTable.getColumnModel().getColumn(10).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title10")); // NOI18N

            printTable.getColumnModel().getColumn(11).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title11")); // NOI18N
            printTable.getColumnModel().getColumn(12).setMaxWidth(40);

            printTable.getColumnModel().getColumn(12).setHeaderValue(resourceMap.getString("printTable.colu
            mnModel.title12")); // NOI18N

            printButton.setText(resourceMap.getString("printButton.text")); // NOI18N
            printButton.setName("printButton"); // NOI18N
            printButton.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    printButtonActionPerformed(evt);
                }
            });

            titleLabel.setFont(resourceMap.getFont("titleLabel.font")); // NOI18N
            titleLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
            titleLabel.setText(resourceMap.getString("titleLabel.text")); // NOI18N
            titleLabel.setName("titleLabel"); // NOI18N
        }
    }
}
```



```

nomoiComboBox.setName("nomoiComboBox"); // NOI18N
nomoiComboBox.setRenderer(nomoiListCellRenderer1);

org.jdesktop.swingbinding.JComboBoxBinding jComboBoxBinding =
org.jdesktop.swingbinding.SwingBindings.createJComboBoxBinding(org.jdesktop.beansbinding.Auto
Binding.UpdateStrategy.READ_WRITE, nomoiList, nomoiComboBox);
bindingGroup.addBinding(jComboBoxBinding);

nomoiComboBox.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        nomoiComboBoxActionPerformed(evt);
    }
});

nomoiLabel.setText(resourceMap.getString("nomoiLabel.text")); // NOI18N
nomoiLabel.setName("nomoiLabel"); // NOI18N

cancelButton.setText(resourceMap.getString("cancelButton.text")); // NOI18N
cancelButton.setName("cancelButton"); // NOI18N
cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cancelPrint(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(279, 279, 279)
            .addComponent(titleLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 600,
Short.MAX_VALUE)
            .addGap(302, 302, 302))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(printButton)
            .addGap(18, 18, 18)
            .addComponent(nomoiLabel)
            .addGap(18, 18, 18)
            .addComponent(nomoiComboBox, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(737, 737, 737)
            .addComponent(cancelButton)
            .addGap(18, 18, 18)
            .addComponent(printScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 1161,
Short.MAX_VALUE)
            .addGap(18, 18, 18))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(printScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 407,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(nomoiLabel)
            .addComponent(nomoiComboBox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(cancelButton)
            .addGap(18, 18, 18))
);

bindingGroup.bind();

pack();
} // <editor-fold> // GEN-END: initComponents

private void printButtonActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_printButtonActionPerformed
// TODO add your handling code here:
MessageFormat header = new MessageFormat("Αίστα Εργασιών Συντήρησης Σελ
{0,number,integer}");
MessageFormat footer = new MessageFormat("KLF SYSTEMS");
try {

    printTable.print(JTable.PrintMode.FIT_WIDTH, header, footer);
} catch (java.awt.print.PrinterException e) {
    System.err.format("Cannot print %s%n", e.getMessage());
}
} // GEN-LAST:event_printButtonActionPerformed

private void nomoiComboBoxActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_nomoiComboBoxActionPerformed
// TODO add your handling code here:
dbcom.fillPrintTable(printTable, nomoiComboBox.getSelectedIndex()+1);
} // GEN-LAST:event_nomoiComboBoxActionPerformed

private void cancelPrint(java.awt.event.ActionEvent evt) // GEN-FIRST:event_cancelPrint
// TODO add your handling code here:
setVisible(false);
} // GEN-LAST:event_cancelPrint

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /** Set the Nimbus look and feel */
    // <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /** If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {}

    java.util.logging.Logger.getLogger(PrintPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {}

    java.util.logging.Logger.getLogger(PrintPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {}

    java.util.logging.Logger.getLogger(PrintPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {}

    java.util.logging.Logger.getLogger(PrintPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    // </editor-fold>

    /** Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            PrintPage dialog = new PrintPage(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {

                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

@Action(enabledProperty = "recordSelected")
public void removeColumn() {

printTable.removeColumn(printTable.getColumnModel().getColumn(printTable.getSelectedColumn()
));
}

@Action
public void restoreColumns() {
    printTable.createDefaultColumnsFromModel();
}

// Variables declaration - do not modify // GEN-BEGIN:variables
private javax.swing.JButton cancelButton;
private javax.swing.JMenuItem columnRemoveMenuItem;
private javax.persistence.EntityManager klfPUEntityManager;
private javax.swing.JComboBox nomoiComboBox;
private javax.swing.JLabel nomoiLabel;

```

```
private java.util.List<klfapplication.Nomoi> nomoiList;
private klfapplication.NomoiListCellRenderer nomoiListCellRenderer1;
private javax.persistence.Query nomoiQuery;
private javax.swing.JButton printButton;
private javax.swing.JPopupMenu printPopupMenu;
private javax.swing.JScrollPane printScrollPane;
private javax.swing.JTable printTable;
private javax.swing.JMenuItem restoreColumnsMenuItem;
private javax.swing.JLabel titleLabel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration//GEN-END:variables
}
```

## Products.java

```
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.io.Serializable;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Transient;

/**
 * Κλάση του πίνακα products της βάσης
 * @author Kostas
 */
@Entity
@Table(name = "products", catalog = "klf", schema = "")
@NamedQueries({
    @NamedQuery(name = "Products.findAll", query = "SELECT p FROM Products p"),
    @NamedQuery(name = "Products.findByIdproducts", query = "SELECT p FROM Products p
WHERE p.idproducts = :idproducts"),
    @NamedQuery(name = "Products.findByCode", query = "SELECT p FROM Products p WHERE
p.code = :code"),
    @NamedQuery(name = "Products.findByDescription", query = "SELECT p FROM Products p
WHERE p.description = :description"),
    @NamedQuery(name = "Products.findByInfo", query = "SELECT p FROM Products p WHERE
p.info = :info")})
public class Products implements Serializable {
    @Transient
    private PropertyChangeSupport changeSupport = new PropertyChangeSupport(this);
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idproducts", nullable = false)
    private Integer idproducts;
    @Column(name = "code", length = 45)
    private String code;
    @Column(name = "description", length = 45)
    private String description;
    @Column(name = "info", length = 45)
    private String info;
    @Lob
    @Column(name = "photo")
    private byte[] photo;

    public Products() {
    }

    public Products(Integer idproducts) {
        this.idproducts = idproducts;
    }

    public Integer getIdproducts() {
        return idproducts;
    }

    public void setIdproducts(Integer idproducts) {
        Integer oldIdproducts = this.idproducts;
        this.idproducts = idproducts;
        changeSupport.firePropertyChange("idproducts", oldIdproducts, idproducts);
    }

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        String oldCode = this.code;
        this.code = code;
        changeSupport.firePropertyChange("code", oldCode, code);
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        String oldDescription = this.description;
        this.description = description;
        changeSupport.firePropertyChange("description", oldDescription, description);
    }

    public String getInfo() {
        return info;
    }

    public void setInfo(String info) {
        String oldInfo = this.info;
        this.info = info;
        changeSupport.firePropertyChange("info", oldInfo, info);
    }

    public byte[] getPhoto() {
        return photo;
    }

    public void setPhoto(byte[] photo) {
        byte[] oldPhoto = this.photo;
        this.photo = photo;
        changeSupport.firePropertyChange("photo", oldPhoto, photo);
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idproducts != null ? idproducts.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {

        if (!(object instanceof Products)) {
            return false;
        }
        Products other = (Products) object;
        if (((this.idproducts == null && other.idproducts != null) || (this.idproducts != null &&
!this.idproducts.equals(other.idproducts))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "klfapplication.Products[idproducts=" + idproducts + "]";
    }

    public void addPropertyChangeListener(PropertyChangeListener listener) {
        changeSupport.addPropertyChangeListener(listener);
    }

    public void removePropertyChangeListener(PropertyChangeListener listener) {
        changeSupport.removePropertyChangeListener(listener);
    }
}
```

## RowSorterToStringConverter.java

```
package klfapplication;

import javax.swing.JTable;
import javax.swing.RowFilter;
import javax.swing.table.TableRowSorter;
import org.jdesktop.beansbinding.Converter;

/**
 * Binding converter ανάμεσα σε String και regex RowFilter
 * @author Kostas
 */
public class RowSorterToStringConverter extends Converter {

    private JTable table;

    public JTable getTable() {
        return table;
    }

    public void setTable(JTable table) {
        this.table = table;
    }

    @Override
    public Object convertForward(Object value) {
        return value.toString();
    }

    @Override
    public Object convertReverse(Object mask) {
        TableRowSorter sorter = new TableRowSorter(table.getModel());

        //case-sensitive filter:
        //sorter.setRowFilter(RowFilter.regexFilter(".*" + mask + ".*"));

        //following 7 lines, case-insensitive filter.
        String m = mask.toString();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < m.length(); i++) {
            char c = m.charAt(i);
            sb.append(' ').append(Character.toLowerCase(c)).append(Character.toUpperCase(c)).append(' ');
        }
        sorter.setRowFilter(RowFilter.regexFilter(".*" + sb + ".*"));
        return sorter;
    }
}
```

## Sintiris.java

```
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;

/**
 * Κλάση του πίνακα sintiris της βάσης
 * @author Kostas
 */
@Entity
@Table(name = "sintiris", catalog = "klf", schema = "")
@NamedQueries({
    @NamedQuery(name = "Sintiris.findAll", query = "SELECT s FROM Sintiris s"),
    @NamedQuery(name = "Sintiris.findByIdsintiris", query = "SELECT s FROM Sintiris s
WHERE s.getIdsintiris = :idsintiris"),
    @NamedQuery(name = "Sintiris.findByHmeromhnia", query = "SELECT s FROM Sintiris s
WHERE s.hmeromhnia = :hmeromhnia"),
    @NamedQuery(name = "Sintiris.findByWra", query = "SELECT s FROM Sintiris s WHERE
s.wra = :wra"),
    @NamedQuery(name = "Sintiris.findByCost", query = "SELECT s FROM Sintiris s WHERE
s.cost = :cost"),
    @NamedQuery(name = "Sintiris.findByRepair", query = "SELECT s FROM Sintiris s WHERE
s.repair = :repair"),
    @NamedQuery(name = "Sintiris.findByPayed", query = "SELECT s FROM Sintiris s WHERE
s.payed = :payed"))
public class Sintiris implements Serializable {
    @Transient
    private PropertyChangeSupport changeSupport = new PropertyChangeSupport(this);
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idsintiris")
    private Integer idsintiris;
    @Column(name = "hmeromhnia")
    @Temporal(TemporalType.DATE)
    private Date hmeromhnia;
    @Column(name = "wra")
    private String wra;
    @Lob
    @Column(name = "antalaktika")
    private String antalaktika;
    @Lob
    @Column(name = "info")
    private String info;
    @Lob
    @Column(name = "notes")
    private String notes;
    @Lob
    @Column(name = "photo")
    private byte[] photo;
    @Column(name = "cost")
    private Float cost;
    @Column(name = "repair")
    private Boolean repair;
    @Column(name = "payed")
    private Boolean payed;
    @JoinColumn(name = "custproductsid", referencedColumnName = "idcustomersproducts")
    @ManyToOne(optional = false)
    private Customersproducts custproductsid;

    public Sintiris() {
    }

    public Sintiris(Integer idsintiris) {
        this.getIdsintiris = idsintiris;
    }
}
```

```
public Integer getIdsintiris() {
    return idsintiris;
}

public void setIdsintiris(Integer idsintiris) {
    Integer oldIdsintiris = this.getIdsintiris;
    this.getIdsintiris = idsintiris;
    changeSupport.firePropertyChange("idsintiris", oldIdsintiris, idsintiris);
}

public Date getHmeromhnia() {
    return hmeromhnia;
}

public void setHmeromhnia(Date hmeromhnia) {
    Date oldHmeromhnia = this.hmeromhnia;
    this.hmeromhnia = hmeromhnia;
    changeSupport.firePropertyChange("hmeromhnia", oldHmeromhnia, hmeromhnia);
}

public String getWra() {
    return wra;
}

public void setWra(String wra) {
    String oldWra = this.wra;
    this.wra = wra;
    changeSupport.firePropertyChange("wra", oldWra, wra);
}

public String getAntalaktika() {
    return antalaktika;
}

public void setAntalaktika(String antalaktika) {
    String oldAntalaktika = this.antalaktika;
    this.antalaktika = antalaktika;
    changeSupport.firePropertyChange("antalaktika", oldAntalaktika, antalaktika);
}

public String getInfo() {
    return info;
}

public void setInfo(String info) {
    String oldInfo = this.info;
    this.info = info;
    changeSupport.firePropertyChange("info", oldInfo, info);
}

public String getNotes() {
    return notes;
}

public void setNotes(String notes) {
    String oldNotes = this.notes;
    this.notes = notes;
    changeSupport.firePropertyChange("notes", oldNotes, notes);
}

public byte[] getPhoto() {
    return photo;
}

public void setPhoto(byte[] photo) {
    byte[] oldPhoto = this.photo;
    this.photo = photo;
    changeSupport.firePropertyChange("photo", oldPhoto, photo);
}

public Float getCost() {
    return cost;
}

public void setCost(Float cost) {
    Float oldCost = this.cost;
    this.cost = cost;
    changeSupport.firePropertyChange("cost", oldCost, cost);
}

public Boolean getRepair() {
    return repair;
}

public void setRepair(Boolean repair) {
    Boolean oldRepair = this.repair;
    this.repair = repair;
}
```

```

        changeSupport.firePropertyChange("repair", oldRepair, repair);
    }

    public Boolean getPayed() {
        return payed;
    }

    public void setPaid(Boolean payed) {
        Boolean oldPaid = this.payed;
        this.payed = payed;
        changeSupport.firePropertyChange("payed", oldPaid, payed);
    }

    public Customersproducts getCustproductsid() {
        return custproductsid;
    }

    public void setCustproductsid(Customersproducts custproductsid) {
        Customersproducts oldCustproductsid = this.custproductsid;
        this.custproductsid = custproductsid;
        changeSupport.firePropertyChange("custproductsid", oldCustproductsid, custproductsid);
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idsintirisis != null ? idsintirisis.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Sintirisis)) {
            return false;
        }
        Sintirisis other = (Sintirisis) object;
        if ((this.idsintirisis == null && other.idsintirisis != null) || (this.idsintirisis != null &&
!this.idsintirisis.equals(other.idsintirisis))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "klfapplication.Sintirisis[idsintirisis=" + idsintirisis + "]";
    }

    public void addPropertyChangeListener(PropertyChangeListener listener) {
        changeSupport.addPropertyChangeListener(listener);
    }

    public void removePropertyChangeListener(PropertyChangeListener listener) {
        changeSupport.removePropertyChangeListener(listener);
    }
}

```

## SintirisEditor.java

```
package klfapplication;

import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;

/**
 * Το παράθυρο προσθήκης / επεξεργασίας εργασιών συντήρησης
 * @author Kostas
 */
public class SintirisEditor extends javax.swing.JDialog {

    /** Creates new form SintirisEditor */
    public SintirisEditor(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    protected Sintiris currentSintirisRecord;

    public Sintiris getCurrentSintirisRecord() {
        return currentSintirisRecord;
    }

    public void setCurrentSintirisRecord(Sintiris currentSintirisRecord) {
        Sintiris oldRecord = this.currentSintirisRecord;
        this.currentSintirisRecord = currentSintirisRecord;
        propertyChangeSupport.firePropertyChange("currentSintirisRecord", oldRecord,
currentSintirisRecord);
    }

    protected boolean sintirisConfirmed;

    /**
     * Get the value of sintirisConfirmed
     *
     * @return the value of sintirisConfirmed
     */
    public boolean isSintirisConfirmed() {
        return sintirisConfirmed;
    }

    /**
     * Set the value of sintirisConfirmed
     *
     * @param sintirisConfirmed new value of sintirisConfirmed
     */
    public void setSintirisConfirmed(boolean sintirisConfirmed) {
        this.sintirisConfirmed = sintirisConfirmed;
    }

    private PropertyChangeSupport propertyChangeSupport = new PropertyChangeSupport(this);

    /**
     * Add PropertyChangeListener.
     *
     * @param listener
     */
    public void addPropertyChangeListener(PropertyChangeListener listener) {
        propertyChangeSupport.addPropertyChangeListener(listener);
    }

    /**
     * Remove PropertyChangeListener.
     *
     * @param listener
     */
    public void removePropertyChangeListener(PropertyChangeListener listener) {
        propertyChangeSupport.removePropertyChangeListener(listener);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
    private void initComponents() {
        bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

        dateVerifier1 = new klfapplication.DateVerifier();

        titleLabel = new javax.swing.JLabel();
        dateLabel = new javax.swing.JLabel();
        dateFormattedTextField = new javax.swing.JFormattedTextField();
        timeLabel = new javax.swing.JLabel();
        timeTextField = new javax.swing.JTextField();
        costLabel = new javax.swing.JLabel();
        costTextField = new javax.swing.JTextField();
        payedCheckBox = new javax.swing.JCheckBox();
        repairCheckBox = new javax.swing.JCheckBox();
        antalaktikaLabel = new javax.swing.JLabel();
        antalaktikaScrollPane = new javax.swing.JScrollPane();
        antalaktikaTextArea = new javax.swing.JTextArea();
        infoLabel = new javax.swing.JLabel();
        infoScrollPane = new javax.swing.JScrollPane();
        infoTextArea = new javax.swing.JTextArea();
        notesLabel = new javax.swing.JLabel();
        notesScrollPane = new javax.swing.JScrollPane();
        notesTextArea = new javax.swing.JTextArea();
        cancelButton = new javax.swing.JButton();
        saveButton = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(klfapplication.KlfApplication.class).getContext().get
ResourceMap(SintirisEditor.class);
        setTitle(resourceMap.getString("Form.title")); // NOI18N
        setModal(true);
        setName("Form"); // NOI18N

        titleLabel.setFont(resourceMap.getFont("titleLabel.font")); // NOI18N
        titleLabel.setText(resourceMap.getString("titleLabel.text")); // NOI18N
        titleLabel.setName("titleLabel"); // NOI18N

        dateLabel.setText(resourceMap.getString("dateLabel.text")); // NOI18N
        dateLabel.setName("dateLabel"); // NOI18N

        dateFormattedTextField.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.DateFormatter(java.text.DateFormat.getDateInstance(java.text.DateFormat.SHORT)
)));
        dateFormattedTextField.setInputVerifier(dateVerifier1);
        dateFormattedTextField.setName("dateFormattedTextField"); // NOI18N

        org.jdesktop.beansbinding.Binding binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentSintirisRecord.hmeromhnia}"),
dateFormattedTextField, org.jdesktop.beansbinding.BeanProperty.create("value"));
        bindingGroup.addBinding(binding);

        timeLabel.setText(resourceMap.getString("timeLabel.text")); // NOI18N
        timeLabel.setName("timeLabel"); // NOI18N

        timeTextField.setName("timeTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentSintirisRecord.wra}"), timeTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

        costLabel.setText(resourceMap.getString("costLabel.text")); // NOI18N
        costLabel.setName("costLabel"); // NOI18N

        costTextField.setName("costTextField"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentSintirisRecord.cost}"), costTextField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
        bindingGroup.addBinding(binding);

        payedCheckBox.setText(resourceMap.getString("payedCheckBox.text")); // NOI18N
        payedCheckBox.setName("payedCheckBox"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat
eStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("${currentSintirisRecord.payed}"), payedCheckBox,
org.jdesktop.beansbinding.BeanProperty.create("selected"));
        bindingGroup.addBinding(binding);

        repairCheckBox.setText(resourceMap.getString("repairCheckBox.text")); // NOI18N
        repairCheckBox.setName("repairCheckBox"); // NOI18N

        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.Updat

```





```

        .addComponent(saveButton))
        .addContainerGap()
    );

    bindingGroup.bind();

    pack();
} // </editor-fold> // GEN-END: initComponents

private void saveSintirisi(java.awt.event.ActionEvent evt) // GEN-FIRST: event_saveSintirisi
// TODO add your handling code here:
    setSintirisiConfirmed(true);
    setVisible(false);
} // GEN-LAST: event_saveSintirisi

private void cancelSintirisi(java.awt.event.ActionEvent evt) // GEN-FIRST: event_cancelSintirisi
// TODO add your handling code here:
    setSintirisiConfirmed(false);
    setVisible(false);
} // GEN-LAST: event_cancelSintirisi

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    // Set the Nimbus look and feel */
    // <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    // If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    // For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    //
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(SintirisiEditor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(SintirisiEditor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(SintirisiEditor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(SintirisiEditor.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
} // </editor-fold>

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        SintirisiEditor dialog = new SintirisiEditor(new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});
}
// Variables declaration - do not modify // GEN-BEGIN: variables
private javax.swing.JLabel antalaktikaLabel;
private javax.swing.JScrollPane antalaktikaScrollPane;
private javax.swing.JTextArea antalaktikaTextArea;
private javax.swing.JButton cancelButton;
private javax.swing.JLabel costLabel;
private javax.swing.JTextField costTextField;
private javax.swing.JFormattedTextField dateFormattedTextField;
private javax.swing.JLabel dateLabel;
private klfapplication.DateVerifier dateVerifier1;
private javax.swing.JLabel infoLabel;
private javax.swing.JScrollPane infoScrollPane;
private javax.swing.JTextArea infoTextArea;
private javax.swing.JLabel notesLabel;
private javax.swing.JScrollPane notesScrollPane;
private javax.swing.JTextArea notesTextArea;
private javax.swing.JCheckBox payedCheckBox;
private javax.swing.JCheckBox repairCheckBox;
private javax.swing.JButton saveButton;
private javax.swing.JLabel timeLabel;
private javax.swing.JTextField timeTextField;
private javax.swing.JLabel titleLabel;
private org.jdesktop.beansbinding.BindingGroup bindingGroup;
// End of variables declaration // GEN-END: variables
}

```

## **ΠΑΡΑΡΤΗΜΑ Γ: Παρουσίαση**

# Ανάπτυξη Εφαρμογής Για Επιχείρηση Συντήρησης Φίλτρων Καθαρισμού Νερού

Κωνσταντίνος Καλαφατάκης (ΑΜ: 1399)

Επιβλέπων καθηγητής : Χατζάκης Ηλίας

Επιτροπή Αξιολόγησης : Μανιφάβας Χαράλαμπος

Παπαδάκης Νικόλαος

## ΕΡΓΑΛΕΙΑ

- ▣ Sun Java version 6
- ▣ Net Beans 7.0.1
- ▣ My SQL server 5.5
- ▣ My SQL Workbench 5.2 CE
- ▣ Visual Paradigm for UML 8.3

## Γιατί Java

- ▣ Παρέχει ανεξαρτησία του λειτουργικού συστήματος Windows, Linux, Unix και Macintosh
- ▣ Αντικειμενοστραφής, δηλαδή τα πάντα στη Java είναι είτε κλάση, είτε μέθοδος ή αντικείμενο
- ▣ Ανερχόμενη δύναμη στον προγραμματισμό
- ▣ Απλή
- ▣ Ασφαλής
- ▣ Είναι δωρεάν

## Γιατί MySQL

- ▣ Υψηλή απόδοση, υψηλή αξιοπιστία και ευκολία στη χρήση.
- ▣ Εξοικονόμησης χρόνου και χρημάτων.
- ▣ Οι μεγαλύτεροι αναπτυσσόμενοι οργανισμοί όπως Facebook, Google, Adobe, Alcatel Lucent και Zappos βασίζονται σε MySQL
- ▣ Λειτουργεί σε περισσότερες από 20 πλατφόρμες Linux, Windows, Mac OS, Solaris κ.λ.π.
- ▣ Λογισμικό ανοικτού κώδικα

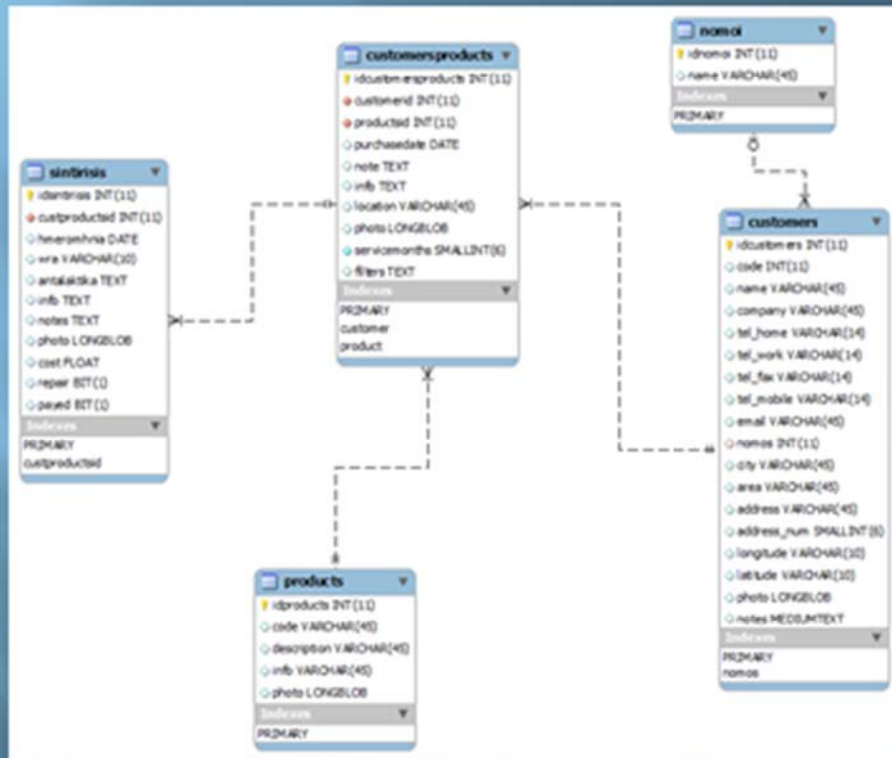
## Ανάπτυξη Εφαρμογής Για Επιχείρηση Συντήρησης Φίλτρων Καθαρισμού Νερού

- ▣ Η εφαρμογή είναι προσαρμοσμένη στις συγκεκριμένες ανάγκες μιας εταιρίας με συστήματα και συσκευές φιλτραρίσματος του νερού. Δίνει τη δυνατότητα διαχείρισης πωλήσεων και συντηρήσεων φίλτρων νερού.
- ▣ Η συγκεκριμένη εφαρμογή θα διευκολύνει τη διατήρηση πληροφοριών σχετικά με στοιχεία πελατών, στοιχεία προϊόντων της εταιρίας, ιστορικό εργασιών συντήρησης και τη δημιουργία λίστας εργασιών συντήρησης η οποία θα βοηθήσει στον προγραμματισμό των εργασιών του τεχνικού τμήματος της εταιρίας.

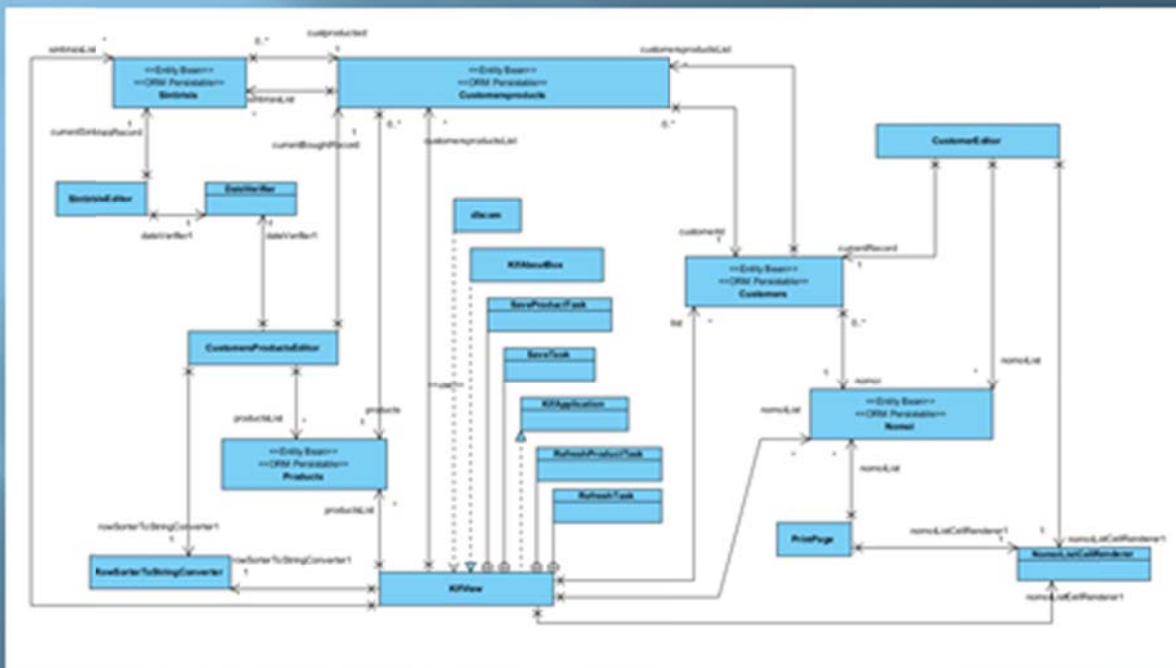
## ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

- ▣ Διαχείρισης πελατών της επιχείρησης
- ▣ Διαχείριση πωλήσεων συστημάτων-συσκευών φίλτρων νερού
- ▣ Διαχείριση προϊόντων εταιρίας
- ▣ Διαχείριση - οργάνωση συντηρήσεων φίλτρων νερού
- ▣ Χάρτης Google maps
- ▣ Backup και επαναφορά της βάσης δεδομένων

## Διάγραμμα Βάσης Δεδομένων



## Διάγραμμα Κλάσεων



ΚΑΙ ΕΠΕΙΔΗ ΜΙΑ ΕΙΚΟΝΑ  
ΕΙΝΑΙ ΧΙΛΙΕΣ ΛΕΞΕΙΣ ...

(ΕΚΤΕΛΕΣΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ)

Ευχαριστώ

**Καλαφατάκης Κωνσταντίνος**