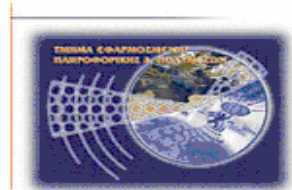




# Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Μηχανικών Πληροφορικής ΤΕ



## Πτυχιακή εργασία

Τίτλος:

Μηχανοργάνωση ιδιωτικού ιατρίου(Java)

**Κίτος Οδυσσέας 2696**  
**Ποντικάκης Νίκος 2813**

Επιβλέπων καθηγητής: Παπαδάκης Νικόλαος

**ΗΡΑΚΛΕΙΟ**  
**2015**



## Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή Νικόλαο Παπαδάκη για την πρόταση και ανάθεση αυτής της πτυχιακής, καθώς και για την υποστήριξη του κατά την διάρκεια εκπόνησης της εργασίας.



## Abstract

The purpose of the dissertation is the analysis, design, implementation and operation of an application, that provides the ability of computerization of a microbiological lab.

For the construction of the application Netbeans was used. The implementation of which was written with Java language (for the Graphical User Interface and the function of the program) and SQL for applying questions on data base that the software uses

This application gives the opportunity to the user to store information about a micro-biological lab and process it.

Also the user can input information for every new patient, and arrange a date to accomplish this inquiry and to make orders of reactors for the microbiological lab.



## Σύνοψη

Σκοπός της πτυχιακής αυτής είναι η ανάλυση, σχεδίαση, υλοποίηση και λειτουργία μια εφαρμογής, που παρέχει την δυνατότητα μηχανοργάνωσης ενός μικροβιολογικού εργαστηρίου.

Για την κατασκευή της εφαρμογής χρησιμοποιήθηκε το Netbeans. Η υλοποίηση της έγινε σε γλώσσα προγραμματισμού Java (για το γραφικό περιβάλλον και την λειτουργία του προγράμματος) και SQL για την υποβολή ερωτημάτων στην βάση δεδομένων που χρησιμοποιεί η εφαρμογή.

Η εφαρμογή αυτή δίνει την δυνατότητα στο χρήστη να αποθηκεύει πληροφορίες για ένα μικροβιολογικό εργαστήριο και να τις επεξεργάζεται.

Ειδικότερα, ο χρήστης μπορεί να εισάγει στοιχεία για κάθε νέο ασθενή, να κλείνει ημερομηνία για την πραγματοποίηση μιας εξέτασης, καθώς και να κάνει παραγγελίες αντιδραστηρίων για το μικροβιολογικό εργαστήριο.



## Περιεχόμενα

<b>1. Εισαγωγή</b> .....	<b>7</b>
1.1 Περίληψη.....	7
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	8
1.3 Σκοπός και στόχοι εργασίας .....	9
1.4 Δομή εργασίας.....	9
<b>2.Μεθοδολογία Υλοποίησης</b> .....	<b>11</b>
2.1. Μέθοδος Ανάλυσης και Ανάπτυξης Πτυχιακής .....	11
2.2 Θεωρίες-Μοντέλα .....	11
2.2.1 Java.....	11
2.2.2 SQL.....	12
2.2.3 ER model .....	12
2.2.4 UML.....	12
2.2.5 Netbeans .....	12
<b>3.Σχέδιο Δράσης για την εκπόνηση της εργασίας</b> .....	<b>12</b>
3.1 State of Art.....	13
3.1.1 Τι είναι Java .....	13
3.1.2 Τι είναι SQL.....	17
3.1.3 Τι είναι ER model .....	20
3.1.4 Σχεσιακό μοντέλο .....	22
3.1.5 Μετατροπή ΔΟΣ σε Σχεσιακό Σχήμα.....	23
3.1.6 Τι είναι UML .....	24
3.1.7 Τι είναι το Netbeans .....	26
3.1.8 Συνδιασμός Java με SQL.....	28
3.2 Σημαντικοί στόχοι για την ολοκλήρωση της πτυχιακής.....	29
<b>4.Κύριο μέρος πτυχιακής</b> .....	<b>30</b>
4.1 Ανάλυση Προβλήματος .....	30
4.1.2 Απαιτήσεις Συστήματος .....	31
4.2 Σχεδιασμός Υλοποίησης .....	31
<b>5. Υλοποίηση</b> .....	<b>32</b>
5.1 Φόρμες προγράμματος .....	34
5.1.1Φόρμα εισαγωγής.....	34
5.1.2 Στοιχεία ασθενή.....	35
5.1.3 Εξετάσεις.....	36
5.1.4 Παραγγελίες.....	38
5.2Βάση δεδομένων .....	39
5.2.1 Σχεδιασμός Βάσης (Ο/Σ και πίνακες) .....	40
5.2.2 Υλοποίηση Βάσης στο Netbeans.....	41
5.3 Συναρτήσεις προγράμματος .....	44



<b>6. Αποτελέσματα.....</b>	<b>50</b>
6.1 Συμπεράσματα .....	50
6.2 Μελλοντική εργασία και επεκτάσεις .....	50
<b>Βιβλιογραφία.....</b>	<b>51</b>



Εικόνα 1-Java.....	13
Εικόνα 2-SQL .....	17
Εικόνα 3-sql query .....	18
Εικόνα 4-er model.....	21
Εικόνα 5-σχεσιακό μοντέλο .....	22
Εικόνα 6-UML.....	24
Εικόνα 7-κολάζ .....	25
Εικόνα 8-Netbeans symbol .....	26
Εικόνα 9-Netbeans workspace .....	27
Εικόνα 10-Netbeans site.....	33
Εικόνα 11-JDK.....	33
Εικόνα 12-project.....	34
Εικόνα 13-φόρμα εισαγωγής.....	35
Εικόνα 14-στοιχεία ασθενή.....	36
Εικόνα 15-εξετάσεις.....	37
Εικόνα 16-εξετάσεις που εκκρεμούν .....	37
Εικόνα 17-παραγγελίες 1 .....	38
Εικόνα 18-παραγγελίες 2.....	39
Εικόνα 19-αποτυχία παραγγελίας .....	39
Εικόνα 20-Διάγραμμα Ο/Σ βάσης.....	40
Εικόνα 21-database .....	41
Εικόνα 22-πίνακας βάσης 1 .....	42
Εικόνα 23-πίνακας βάσης 2 .....	42
Εικόνα 24-πίνακας βάσης 3 .....	43
Εικόνα 25-πίνακας βάσης 4 .....	43
Εικόνα 26-πίνακας βάσης 5 .....	43
Εικόνα 27-πίνακας βάσης 6 .....	44
Εικόνα 28-πίνακας βάσης 7 .....	44
Εικόνα 29-Doconnect.....	45
Εικόνα 30-DoImport .....	46
Εικόνα 31-Fillexet.....	46
Εικόνα 32-parState .....	47
Εικόνα 33-meiksi .....	48
Εικόνα 34-συναρτήσεις αντικειμένων .....	49
Εικόνα 35-forma .....	49

# 1. Εισαγωγή

## 1.1 Περίληψη



Το αντικείμενο που πραγματεύεται αυτή η πτυχιακή εργασία είναι η κατασκευή προγράμματος μηχανοργάνωσης για ένα μικροβιολογικό εργαστήριο με χρήση του Netbeans. Πιο συγκεκριμένα, για την υλοποίηση χρησιμοποιήθηκε γλώσσα προγραμματισμού Java. Για την δημιουργία, αποθήκευση και εμφάνιση πληροφοριών έχει χρησιμοποιηθεί γλώσσα προγραμματισμού βάσεων δεδομένων SQL και διάφορα ερωτήματα της.

Το πρόγραμμα παρέχει στον χρήστη την δυνατότητα να αποθηκεύει τα στοιχεία των ασθενών του, να διαχειρίζεται τα ραντεβού, αλλά και τις εξετάσεις που θέλουν να πραγματοποιήσουν. Επίσης αποθηκεύει και διαχειρίζεται τα αντιδραστήρια που χρησιμοποιούνται σε ένα μικροβιολογικό εργαστήριο, καθώς και τις παραγγελίες τους.

Πιο συγκεκριμένα οι λειτουργίες που μπορεί να εκτελέσει το πρόγραμμα είναι οι ακόλουθες:

1. Εισαγωγή
  - 1.1 Εισαγωγή νέου ασθενή
  - 1.2 Κλείσιμο εξέτασης
2. Στοιχεία ασθενή
  - 2.1 Αναζήτηση ασθενή
  - 2.2 Προβολή εξετάσεων ασθενή
  - 2.3 Πίνακας στοιχείων ασθενή
  - 2.4 Πίνακας εξετάσεων ασθενή
3. Εξετάσεις
  - 3.1 Πίνακας εξετάσεων
  - 3.2 Φόρμα εισαγωγής αποτελεσμάτων
  - 3.3 Φόρμα προβολής αποτελεσμάτων
4. Παραγγελίες
  - 4.1 Φόρμα παραγγελίας αντιδραστηρίων
  - 4.2 Φόρμα παραλαβής παραγγελίας
  - 4.3 Πίνακας παραγγελιών
  - 4.4 Πίνακας αποθεμάτων

## 1.2 Κίνητρο για την διεξαγωγή της εργασίας





Στην εποχή που ζούμε η πληροφορική έχει γίνει απαραίτητο κομμάτι σε πολλούς κλάδους εργασίας, ένας απ' αυτούς είναι και της ιατρικής. Γι' αυτό το λόγο φτιάχνονται προγράμματα τα οποία διευκολύνουν στην πρόοδο, την ανάπτυξη αλλά και τη μελέτη του κλάδου.

Όσον αφορά ένα μικροβιολογικό εργαστήριο, το πρόγραμμα μας είναι απαραίτητο για την σωστή λειτουργία του, καθώς διευκολύνει την οργάνωση και διαχείριση των πληροφοριών και των δεδομένων του.

### 1.3 Σκοπός και στόχοι εργασίας

Ο κυριότερος στόχος είναι η δημιουργία ενός προγράμματος μηχανοργάνωσης μικροβιολογικού εργαστηρίου στο οποίο ο χρήστης θα μπορεί να επεξεργάζεται οποιεσδήποτε πληροφορίες του είναι απαραίτητες. Ο χρήστης μπορεί να αποθηκεύει στοιχεία για ένα ασθενή να κλείνει τα ραντεβού του για εξετάσεις καθώς και να πραγματοποιεί παραγγελίες για το μικροβιολογικό εργαστήριο. Το πρόγραμμα μας έχει σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να είναι όσο το δυνατόν πιο εύκολο και απλό στη χρήση του.

Ο απώτερος σκοπός της εργασίας είναι η εκμάθηση μιας διαδομένης γλώσσας προγραμματισμού όπως είναι η Java. Επίσης η χρήση μιας βάσης δεδομένων μας οδήγησε στη εκπλήρωση ενός ακόμη στόχου ο οποίος είναι η εκμάθηση υποβολής ερωτημάτων με τη χρήση της γλώσσας προγραμματισμού SQL.

### 1.4 Δομή εργασίας



## **Κεφάλαιο 2: Μεθοδολογία Υλοποίησης.**

Σε αυτό το κεφάλαιο γίνεται αναφορά για την μεθοδολογία που ακολουθήσαμε για την υλοποίηση της πτυχιακής εργασίας. Επίσης κάνουμε μια συνοπτική αναφορά στις εμπλεκόμενες γλώσσες προγραμματισμού.

## **Κεφάλαιο 3: Σχέδιο δράσης για την εκπόνηση της πτυχιακής εργασίας.**

Παρατηθούμε με βάση την έρευνα που πραγματοποιήσαμε, πιο εκτενής λεπτομέρειες για το Netbeans καθώς και για τις γλώσσες προγραμματισμού που χρησιμοποιήσαμε. Δίνονται επίσης επεξηγήσεις σε σχετικές βασικές έννοιες.

## **Κεφάλαιο 4: Κύριο μέρος πτυχιακής.**

Πραγματοποιείται ανάλυση του προβλήματος που καλούμαστε να αντιμετωπίσουμε, προσδιορισμός των απαιτήσεων του προγράμματος και αναφορά στον σχεδιασμό υλοποίησης.

## **Κεφάλαιο 5: Υλοποίηση.**

Σε αυτό το κεφάλαιο περιγράφονται τα διάφορα βήματα της υλοποίησης όπως η εγκατάσταση απαραίτητου λογισμικού και η ανάπτυξη του προγράμματος μας.

## **Κεφάλαιο 6: Αποτελέσματα.**

Στο έκτο και τελευταίο κεφάλαιο αναφέρονται τα αποτελέσματα αυτής της πτυχιακής εργασίας. Επίσης μοιραζόμαστε τα συμπεράσματα μας καθώς και μελλοντικές επεκτάσεις του προγράμματος.



## 2.Μεθοδολογία Υλοποίησης

### 2.1. Μέθοδος Ανάλυσης και Ανάπτυξης Πτυχιακής

Το πρόβλημα στην συγκεκριμένη περίπτωση είναι η οργάνωση όλων των στοιχείων που διαχειρίζεται ένα μικροβιολογικό εργαστήριο, με τέτοιο τρόπο έτσι ώστε να είναι προσβάσιμες και διαχειρίσιμες από τον χρήστη.

Για την επίλυση του προβλήματος, δημιουργήσαμε ένα πρόγραμμα το οποίο κατασκευάστηκε στο Netbeans με χρήση γλώσσας προγραμματισμού Java και γλώσσα διαχείρισης βάσεων δεδομένων SQL. Το πρόγραμμα μας διαχειρίζεται όλες αυτές τις πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων.

Με την Java έχουμε την δυνατότητα να φτιάξουμε το περιβάλλον του προγράμματος που θα χρησιμοποιεί ο χρήστης και με την SQL να επεξεργαστούμε και να αποθηκεύσουμε όλα τα δεδομένα που είναι απαραίτητα στη βάση.

ΜΕΘΟΔΟΛΟΓΙΑ	ΑΝΑΦΟΡΑ ΣΤΗΝ ΒΙΒΛΙΟΓΡΑΦΙΑ
Netbeans	<a href="http://en.wikipedia.org/wiki/NetBeans">http://en.wikipedia.org/wiki/NetBeans</a> , <a href="https://netbeans.org/">https://netbeans.org/</a>
Java	<a href="http://en.wikipedia.org/wiki/Java">el.wikipedia.org/wiki/Java</a> , <a href="https://java.com/en/">https://java.com/en/</a>
SQL	<a href="http://en.wikipedia.org/wiki/SQL">http://en.wikipedia.org/wiki/SQL</a> , <a href="http://www.w3schools.com/sql/">http://www.w3schools.com/sql/</a>
ER model	<a href="http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model">http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model</a>
UML	<a href="http://el.wikipedia.org/wiki/Unified_Modeling_Language">http://el.wikipedia.org/wiki/Unified_Modeling_Language</a>

### 2.2 Θεωρίες-Μοντέλα

Παρακάτω γίνεται συνοπτική αναφορά στις γλώσσες που χρησιμοποιήθηκαν στον πρόγραμμα μας (Java,SQL). Επίσης αναφέρουμε τα θεωρητικά μοντέλα που βοήθησαν στο να υλοποιήσουμε το πρόγραμμα(UML, ER model), καθώς και το Netbeans που είναι το περιβάλλον στο οποίο εργαστήκαμε.

#### 2.2.1 Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που την δημιούργησε η εταιρεία πληροφορικής Sun Microsystems. Ένα βασικό πλεονέκτημα της Java σε σχέση με τις περισσότερες άλλες γλώσσες είναι η ανεξαρτησία του λειτουργικού συστήματος και της πλατφόρμας.



## 2.2.2 SQL

Η SQL (Structured Query Language) είναι μία γλώσσα που χρησιμοποιείται σε βάσεις δεδομένων και σχεδιάστηκε για τη διαχείριση δεδομένων. Η γλώσσα αυτή περιλαμβάνει δυνατότητες ανάκτησης, επεξεργασίας και ενημέρωσης δεδομένων. Επίσης την δυνατότητα δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.

## 2.2.3 ER model

Το μοντέλο οντοτήτων-συσχετίσεων (ER model) είναι ένα ιδεατό μοντέλο δεδομένων τα οποία έχουν συγκεκριμένη δομή. Χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης είναι το διάγραμμα οντοτήτων-συσχετίσεων.

## 2.2.4 UML

Η UML πλέον είναι μια πρότυπη γλώσσα μοντελοποίησης στη κατασκευή λογισμικού. Χρησιμοποιείται για τη κατασκευή, τη γραφική απεικόνιση, τον προσδιορισμό, και την τεκμηρίωση των στοιχείων ενός συστήματος. Μπορεί να χρησιμοποιηθεί από την ανάλυση απαιτήσεων ως τον έλεγχο ενός ολοκληρωμένου συστήματος.

## 2.2.5 Netbeans

Το NetBeans IDE ένα εργαλείο που χρησιμοποιούν οι προγραμματιστές για να γράψουν, να κάνουν compile, debug και να αναπτύξουν προγράμματα. Είναι γραμμένο σε Java - αλλά μπορεί να υποστηρίξει και άλλες γλώσσες προγραμματισμού.

## 3.Σχέδιο Δράσης για την εκπόνηση της εργασίας



## 3.1 State of Art

### 3.1.1 Τι είναι Java



Εικόνα 1-Java

Η αντικειμενοστρεφής γλώσσα προγραμματισμού που θα ασχοληθούμε σε αυτό το κεφάλαιο είναι η Java. Στις αρχές τις δεκαετίας του 90, η Sun έψαχνε το κατάλληλο εργαλείο το οποίο θα ήταν η πλατφόρμα ανάπτυξης λογισμικού σε μικρό-συσκευές (όπως έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Εκείνη την εποχή υπήρχαν γλώσσες προγραμματισμού όπως η C και η C++. Μετά από πειραματισμούς προέκυψε το συμπέρασμα ότι αυτές οι γλώσσες δεν μπορούσαν πλέον να καλύψουν τις ανάγκες τους. Ο James Gosling, αποκαλούμενος ως και ο "πατέρας" της Java, που εκείνη την εποχή εργαζόταν στην Sun, είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες ως πρότυπα για το νέο εργαλείο που αναζητούσαν από την εταιρία. Τελικά μετά από καιρό κατέληξαν με μια πρόταση, η οποία ήταν η γλώσσα Oak. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε καθημερινά.

Στις 27 Απριλίου 2010 η εταιρία Oracle Corporation ανακοίνωσε ότι ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών (πνευματικά δικαιώματα/ πατέντες) που η δεύτερη είχε στην κατοχή της ή δημιουργήσει. Αυτή η συμφωνία θεωρείται σημαντική για το μέλλον της Java καθώς και του γενικότερου συστήματος τεχνολογιών γύρω από αυτή μιας και ο έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια.



Ένα από τα πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος της αλλά και της πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (τρέχουν και σε κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση ή να αλλάξει ο πηγαίος κώδικας. Για να γίνει όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα που είναι γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή αλλά και λειτουργικού συστήματος (Windows, Unix, Linux). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Mac. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

Αφού γραφεί ένα πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή javac, ο οποίος παράγει έναν αριθμό από αρχεία .class (κώδικας byte). Ο κώδικας byte είναι η μορφή που παίρνει ο κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό, θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής η οποία να μπορεί να υποστηριχθεί από το λειτουργικό σύστημα και τον επεξεργαστή, έτσι ώστε να εκτελεστεί (αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή).

Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα bytecode απευθείας σε κώδικα μηχανής με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού σε Java. Η JVM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Macintosh, , Linux, Unix κινητά, παιχνιδιομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής η οποία είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα.

## ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ JAVA



Οι ακόλουθες ενότητες περιγράφουν τις σημαντικότερες διαφορές της Java με τις άλλες γλώσσες προγραμματισμού.

- Ανεξαρτησία πλατφόρμας

Είναι από τους κύριους λόγους που η Java είναι τόσο δημοφιλής. Πράγμα το οποίο σημαίνει ότι η Java μπορεί να τρέξει σε πολλούς διαφορετικούς τύπους υπολογιστών και ένα πρόγραμμα Java τρέχει σε οποιοδήποτε υπολογιστή.

- Αντικειμενοστραφής

Η Java είναι αντικειμενοστραφής γλώσσα, πράγμα που σημαίνει ότι τα προγράμματα Java αποτελούνται από στοιχεία προγραμματισμού που ονομάζονται αντικείμενα. Τα αντικείμενα έχουν δύο βασικά χαρακτηριστικά :

1. Τα αντικείμενα έχουν δεδομένα. Για παράδειγμα, ένα αντικείμενο που αναπαριστά ένα τετράγωνο έχει στοιχεία όπως το μέγεθος και χρώμα.

2. Τα αντικείμενα έχουν συμπεριφορά, μπορούν δηλαδή να εκτελέσουν ορισμένα καθήκοντα. Στην Java, τα καθήκοντα αυτά ονομάζονται μέθοδοι.

Ορισμένες μέθοδοι απλά επιτρέπουν την πρόσβαση στα δεδομένα του αντικειμένου. Οι κλάσεις είναι στενά συνδεδεμένες με τα αντικείμενα, καθώς περιγράφουν τα δεδομένα και τις μεθόδους που καθορίζουν τη συμπεριφορά και την κατάσταση του αντικειμένου. Επίσης, όταν εκτελείται το πρόγραμμα, οι κλάσεις χρησιμοποιούνται για τη δημιουργία αντικειμένων.

- Το Java API

Η Java παρέχει μια βιβλιοθήκη κλάσεων που προσφέρουν συνηθισμένες και χρήσιμες λειτουργίες, που ονομάζεται Java API. Το API παρέχει στο εξωτερικό λογισμικό τη δυνατότητα να εκτελεί κάθε λειτουργία που υποστηρίζει η κλάση. Στην ουσία, η πραγματική πρόκληση για την χρήση της Java δεν είναι η εκμάθηση της γλώσσας, αλλά η εκμάθηση του API.

Η γλώσσα Java έχει μόνο 50 λέξεις-κλειδιά, όμως το Java API έχει αρκετές χιλιάδες κλάσεις που μπορούν να χρησιμοποιηθούν στα προγράμματά. Επίσης, για την σωστή χρήση της Java, δεν είναι απαραίτητη η εκμάθηση όλων των πτυχών του Java API. Οι περισσότεροι προγραμματιστές μπορούν να επιφέρουν άπταιστα αποτελέσματα την χρησιμοποιώντας μόνο ένα μικρό μέρος από το Java API.

## ΠΛΕΟΝΕΚΤΗΜΑΤΑ - ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ JAVA



## ΠΛΕΟΝΕΚΤΗΜΑΤΑ

- ✓ Είναι ανεξάρτητη από την πλατφόρμα: Ένα από τα πιο σημαντικά πλεονεκτήματα της Java είναι η ικανότητα της εύκολης μετακίνησης από το ένα σύστημα υπολογιστή στον άλλο. Η δυνατότητα να τρέξει το ίδιο πρόγραμμα σε πολλά διαφορετικά συστήματα είναι ζωτικής σημασίας για το λογισμικό World Wide Web, και η Java έχει πετύχει σε αυτό με το να είναι ανεξάρτητη από την πλατφόρμα.
- ✓ Είναι απλή: Η Java σχεδιάστηκε ώστε να είναι εύκολη στην χρήση της και συνεπώς εύκολη στη σύνταξη, την κατάρτιση, τον εντοπισμό σφαλμάτων, και τη μάθηση σχετικά με άλλες γλώσσες προγραμματισμού. Ο λόγος που η Java είναι πολύ πιο απλή από την C++ είναι επειδή η Java λειτουργεί με αυτόματη κατανομή μνήμης και συλλογή απορριμμάτων καθώς η C++ απαιτεί από τον προγραμματιστή την εκχώρηση μνήμης και τη συλλογή των απορριμμάτων.
- ✓ Είναι ασφαλής: Η Java, είναι μία από τις πρώτες γλώσσες προγραμματισμού που έχουν αφιερώσει μεγάλο μέρος του σχεδιασμού τους στην ασφάλεια.
- ✓ Είναι ισχυρή: Καμία γλώσσα προγραμματισμού δεν μπορεί να εξασφαλίσει την αξιοπιστία. Είναι ένα ισχυρό και αξιόπιστο μέσο που δίνει μεγάλη έμφαση στο έγκαιρο έλεγχο για τυχόν λάθη. Οι μεταγλωττιστές της είναι σε θέση να εντοπίσουν πολλά προβλήματα, τα οποία θα εμφανίζονταν σε άλλα προγράμματα.
- ✓ Διανέμεται: Ένα κατανεμημένο σύστημα περιλαμβάνει πολλούς υπολογιστές σε ένα δίκτυο οι οποίοι εργάζονται από κοινού. Η Java έχει σχεδιαστεί για την εύκολη κατανομή συστημάτων με τη δυνατότητα δικτύωσης που είναι εκ φύσεως ενσωματωμένη με τη Java.
- ✓ Ερμηνεύεται: Ο διερμηνέας είναι απαραίτητος προκειμένου να τρέξει η Java. Τα προγράμματα καταρτίζονται από κώδικα Java Virtual Machine που ονομάζεται bytecode (ενδιάμεσος κώδικας). Το bytecode είναι ανεξάρτητο και είναι σε θέση να τρέξει σε οποιαδήποτε συσκευή που έχει διερμηνέα Java. Με την Java, το πρόγραμμα χρειάζεται να συνταχθεί μόνο μία φορά, και το bytecode που παράγεται από το μεταγλωττιστή της Java, μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα.

## ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Η Java έχει μερικά μειονεκτήματα, στα οποία όμως υπάρχουν περιθώρια





βελτίωσης.

- **Επιδόσεις:** Η Java μπορεί να θεωρηθεί σημαντικά πιο αργή από άλλες γλώσσες προγραμματισμού, καθώς οι λειτουργίες τις καταναλώνουν περισσότερη μνήμη.
- **Εμφάνιση και αίσθηση:** Η προεπιλεγμένη εμφάνιση και αίσθηση του γραφικού περιβάλλοντος είναι πολύ διαφορετική από μητρικές εφαρμογές. Η Sun εργάζεται στον τομέα αυτό και μπορούμε να δούμε μερικές μεγάλες τεχνολογίες της Sun στο σύντομο μέλλον.

### 3.1.2 Τι είναι SQL



Εικόνα 2-SQL

Η SQL (Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων. Αρχικά, βασίστηκε στη σχεσιακή άλγεβρα και σχεδιάστηκε για τη διαχείριση δεδομένων σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS). Η γλώσσα περιλαμβάνει δυνατότητες δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, ανάκτησης και ενημέρωσης δεδομένων, , αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από

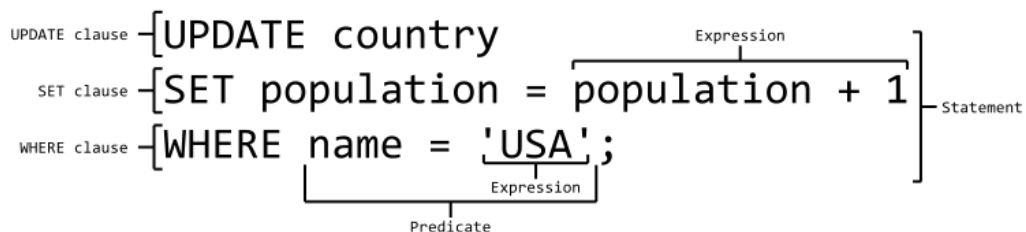


τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

Η SQL αναπτύχθηκε από τους Andrew Richardson, Donald C. Messerly και Raymond F. Boyce, στις αρχές της δεκαετίας του 1970 στην IBM. Αποκαλούμενη αρχικά SEQUEL, αυτή η έκδοση είχε σκοπό να χειριστεί και να ανακτήσει τα στοιχεία που αποθηκεύτηκαν στο πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων της IBM, το System R.

Το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων RDMBS αναπτύχθηκε στο MIT στις αρχές της δεκαετίας του 1970 και η Ingres, που αναπτύχθηκε το 1974 στο Πανεπιστήμιο Μπέρκλεϋ εφάρμοσε μία γλώσσα διατύπωσης ερωτήσεων γνωστή ως QUEL, το οποίο αντικαταστάθηκε αργότερα στην αγορά από την SQL.

Προς το τέλος της δεκαετίας του 70 η Oracle Corporation (τότε Relational Software) είδε τη δυνατότητα αυτών που περιγράφηκαν από Codd, Chamberlin, και Boyce και αναπτύξε την SQL βασισμένο στο RDBMS, με τις φιλοδοξίες πώλησης του στο Αμερικανικό ναυτικό και άλλες υπηρεσίες. Το καλοκαίρι του 1979, η Relational Software εισήγαγε την πρώτη διαθέσιμη στο εμπόριο εφαρμογή του SQL και νίκησε την IBM με τη διάθεση του πρώτου εμπορικού RDBMS για μερικές εβδομάδες.



Εικόνα 3-sql query

Η γλώσσα SQL χωρίζεται σε διάφορα γλωσσικά στοιχεία, που περιλαμβάνουν

- Clauses, οι οποίες είναι σε μερικές περιπτώσεις προαιρετικές, αλλά είναι απαραίτητα συστατικά των δηλώσεων αλλά και των ερωτήσεων.
- Expressions που μπορούν να παραγάγουν είτε πίνακες που αποτελούνται από στήλες και σειρές στοιχείων είτε τις κλιμακωτές τιμές.
- Statements που μπορούν να έχουν μια επίδραση στα σχήματα και τα στοιχεία, ή που μπορούν να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις από άλλα προγράμματα.



- Το κενό αγνοείται γενικά στις Statements και τις Queries SQL. Ένα κενό είναι όμως απαραίτητο για να ξεχωρίζει Statements όπως και στην κανονική γραφή κειμένων.
- Predicates που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν σαν σωστό ή λάθος.
- Queries που ανακτούν τα στοιχεία.

### Βασικά ερωτήματα SQL:

Με την SQL, μπορούμε να κάνουμε ένα ερώτημα (Query) σε μια βάση δεδομένων και να έχουμε ένα αποτέλεσμα (Result) σε μορφή πίνακα. Ένα ερώτημα έχει την εξής μορφή : `SELECT FirstName FROM Persons`.

Η SQL είναι μια γλώσσα για την εκτέλεση ερωτημάτων (queries). Αλλά περιλαμβάνει επίσης την ενημέρωση εγγραφών, την εισαγωγή νέων εγγραφών και τη διαγραφή υπαρχόντων εγγραφών εντολές ερωτημάτων και ενημέρωσης αποτελούν μαζί τη Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language, DML) της SQL.

Παρακάτω έχουμε κάποιες εντολές και ερωτήματα που συναντάμε στην SQL:

**SELECT** - εξάγει δεδομένα από μια βάση δεδομένων.

**UPDATE** - ενημερώνει δεδομένα σε μια βάση δεδομένων.

**DELETE** - διαγράφει δεδομένα από μια βάση δεδομένων.

**INSERT** - εισάγει νέα δεδομένα σε μια βάση δεδομένων.

**CREATE TABLE** - δημιουργεί έναν νέον πίνακα σε μια βάση δεδομένων.

**ALTER TABLE** - τροποποιεί έναν πίνακα σε μια βάση δεδομένων.

**DROP TABLE** - διαγράφει έναν πίνακα από μια βάση δεδομένων.



**CREATE INDEX** - δημιουργεί ένα index (search key).

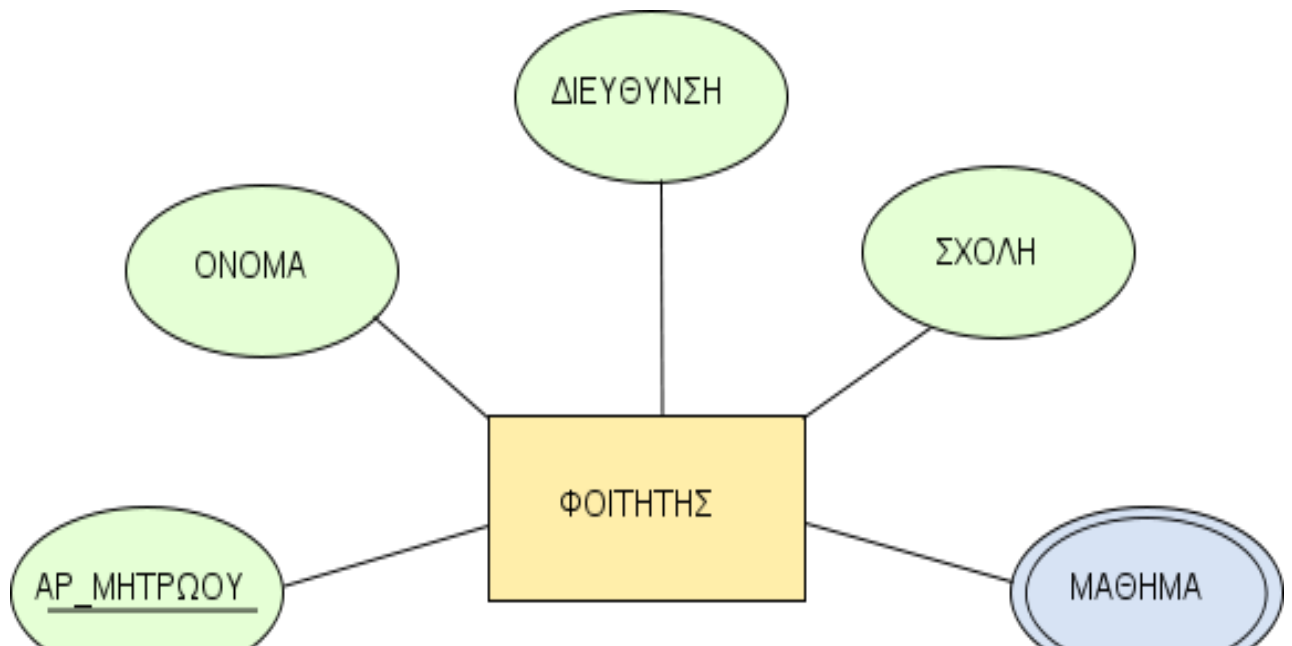
**DROP INDEX** - διαγράφει έναν index.

Η Γλώσσα Ορισμού Δεδομένων (Data Definition Language, DDL), που αποτελεί μέρος της SQL, επιτρέπει τη δημιουργία και τη διαγραφή πινάκων μιας βάσης δεδομένων. Μπορούμε επίσης να ορίσουμε indexes (keys), να καθορίσουμε συνδέσμους (links) ανάμεσα στους πίνακες και να επιβάλλουμε περιορισμούς ανάμεσα στους πίνακες μιας βάσης δεδομένων.

### 3.1.3 Τι είναι ER model

Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο Ο/Σ - ER model) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Χρησιμοποιείται στη μηχανική λογισμικού για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης λέγεται διάγραμμα οντοτήτων-συσχετίσεων, ή διάγραμμα Ο/Σ εν συντομία.

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης κατά την ανάλυση των απαιτήσεών ενός συστήματος πληροφοριών. Σκοπός του είναι να περιγράψει τις απαραίτητες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων. Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεών τους σε έναν ορισμένο τομέα ενδιαφέροντος. Κατά τον σχεδιασμό ενός συστήματος πληροφοριών, που στηρίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων χαρτογραφείται σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Το στάδιο αυτό ονομάζεται στάδιο λογικού σχεδιασμού. Έπειτα, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο χαρτογραφείται σε κάποιο φυσικό μοντέλο. Ορισμένες φορές και οι δύο φάσεις αναφέρονται ως "φυσικός σχεδιασμός".



Εικόνα 4-er model

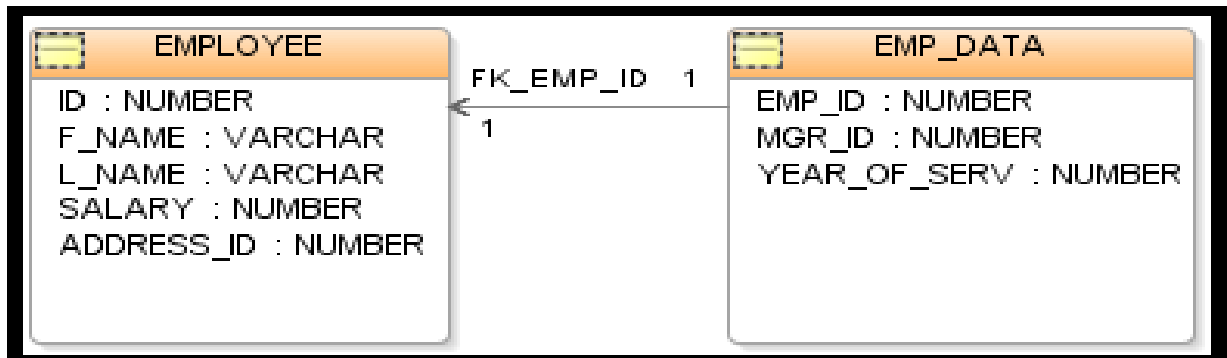
Υπάρχουν διάφοροι κανόνες για τη σημειογραφία που χρησιμοποιείται στα διαγράμματα οντοτήτων-συσχετίσεων (ERDs). Αρκετές φορές η σημειογραφία που υιοθετείται κατά το σχεδιασμό μιας βάσης δεδομένων διαφέρει ως προς τη σαφήνεια, τις δυνατότητες που έχει η γραφική γλώσσα και τα εργαλεία. Συνήθως για τις συσχετίσεις χρησιμοποιούνται ευθείες με διαφορετικές άκρες, ή με σημειώσεις που παριστάνουν την πληθικότητα του τύπου συσχέτισης.

Σημειογραφίες ιστορικής σημασίας:

- Η σημειογραφία Chen, του Peter Chen που δημιούργησε τα διαγράμματα το 1976.
- Η σημειογραφία IDEF1X ως de facto πρότυπο, χρησιμοποιούμενο για χρόνια από τις αρχές των ΗΠΑ.
- Η σημειογραφία Martin (πόδι του κόρακα) διαδεδομένη σε εργαλεία για διαγράμματα.
- Η σημειογραφία του Charles Bachman, σε εργαλεία διαγραμμάτων.
- Η σημειογραφία (Min , max), του Jean Raymond Abrial το 1974.
- Η γλώσσα UML, πρότυπο που χρησιμοποιείται ως αντικαταστάτης των διαγραμμάτων Ο/Σ.



### 3.1.4 Σχεσιακό μοντέλο



Εικόνα 5-σχεσιακό μοντέλο

Σχεσιακό μοντέλο είναι μία συλλογή δεδομένων οργανωμένη σε πίνακες που έχουν μεταξύ τους κάποια σχέση. Ταυτόχρονα παρέχει ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση ή και πιο πολύπλοκες διαδικασίες πάνω στα δεδομένα. Σκοπός μιας βάσης δεδομένων είναι η οργανωμένη αποθήκευση πληροφορίας και η δυνατότητα εξαγωγής αυτής της πληροφορίας.

Σύμφωνα με ερωτήματα που τίθενται στη σχεσιακή βάση δεδομένων παίρνουμε τα αντίστοιχα αποτελέσματα σε οργανωμένη μορφή. Τα δεδομένα είναι δυνατόν να αναδιοργανώνονται με πολλούς διαφορετικούς τρόπους, σε νοητούς πίνακες, χωρίς να είναι απαραίτητη η αναδιοργάνωση των αρχικών πινάκων. Τη σχεσιακή βάση δεδομένων επινόησε ο Αντικέρ Κεντ το 1970.

Οι ερωτήσεις προς τη βάση δεδομένων, γίνονται συνήθως μέσω της διαδεδομένης γλώσσας SQL (Structured Query Language). Εκτελώντας ερωτήματα είναι δυνατόν, ανάλογα με τα δικαιώματά που έχει ο χρήστης, να δημιουργήσει, να μεταβάλλει και να διαγράψει δεδομένα της βάσης, ή να ανασύρει πληροφορίες με σύνθετα κριτήρια αναζήτησης.

Οι πρώτοι τύποι βάσεων δεδομένων χρησιμοποίησαν ιεραρχικά μοντέλα ταξινόμησης ή μοντέλα δικτύων. Η εξέλιξη στο σχεσιακό μοντέλο σχετίζεται με τη δυνατότητα συσχετισμού των πινάκων μέσω διακριτών πεδίων. Για το συσχετισμό δύο πινάκων αρκεί ένα κοινό πεδίο, χαρακτηριστικό που κάνει το μοντέλο εύκαμπτο.



### 3.1.5 Μετατροπή ΔΟΣ σε Σχεσιακό Σχήμα

Για κάθε τύπο οντοτήτων και για κάθε τύπο συσχετίσεων δημιουργούμε ένα σχήμα σχέσης που παίρνει το όνομα του αντίστοιχου τύπου.

- Για κάθε (ισχυρό) τύπο οντοτήτων  $E$  δημιουργούμε ένα σχήμα σχέσης  $R$  με τα ίδια γνωρίσματα (ένα για κάθε απλό γνώρισμα του  $E$ ). Αν το  $E$  έχει σύνθετα γνωρίσματα, στο σχεσιακό σχήμα  $R$  έχουμε ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο.
- Για κάθε ασθενή τύπο οντοτήτων  $A$  που εξαρτάται από τον ισχυρό τύπο οντοτήτων  $B$  (ιδιοκτήτης) δημιουργούμε ένα σχήμα σχέσης  $R$  με γνωρίσματα:
  1. τα γνωρίσματα του μερικού κλειδιού του  $A$ , και
  2. τα γνωρίσματα του πρωτεύοντος κλειδιού του  $B$
- Για κάθε 1-1 δυαδική συσχέτιση  $R$  μεταξύ δύο τύπων οντοτήτων του διαγράμματος Ο/Σ που αντιστοιχούν στις σχέσεις  $T$  και  $S$ 
  1. επιλογή μιας εκ των  $T$  και  $S$ , έστω της  $S$
  2. το πρωτεύον κλειδί της  $S$  γίνεται ξένο κλειδί της  $T$
- Για κάθε 1- $N$  δυαδική συσχέτιση  $R$  μεταξύ δύο τύπων οντοτήτων του διαγράμματος Ο/Σ που αντιστοιχούν στις σχέσεις  $T$  και  $S$ 
  1. έστω  $T$  από την πλευρά 'N' στο διάγραμμα ΟΣ
  2. το πρωτεύον κλειδί της  $S$  γίνεται ξένο κλειδί της  $T$



- Για κάθε M-N δυαδική συσχέτιση R μεταξύ δύο τύπων οντοτήτων του διαγράμματος Ο/Σ που αντιστοιχούν στις σχέσεις T1 και T2
  1. δημιουργούμε μια νέα σχέση S
  2. το πρωτεύον κλειδί της T1 γίνεται ξένο κλειδί της S
  3. το πρωτεύον κλειδί της T2 γίνεται ξένο κλειδί της S
  4. ο συνδυασμός των παραπάνω θα αποτελεί το πρωτεύον κλειδί της S
  5. συμπεριλαμβάνουμε τυχόν απλά ή σύνθετα γνωρίσματα της R ως γνωρίσματα της S

### 3.1.6 Τι είναι UML



Εικόνα 6-UML

Η **Unified Modeling Language (UML)**, *Ενοποιημένη Γλώσσα Μοντελοποίησης*) πλέον είναι η πρότυπη γλώσσα μοντελοποίησης.

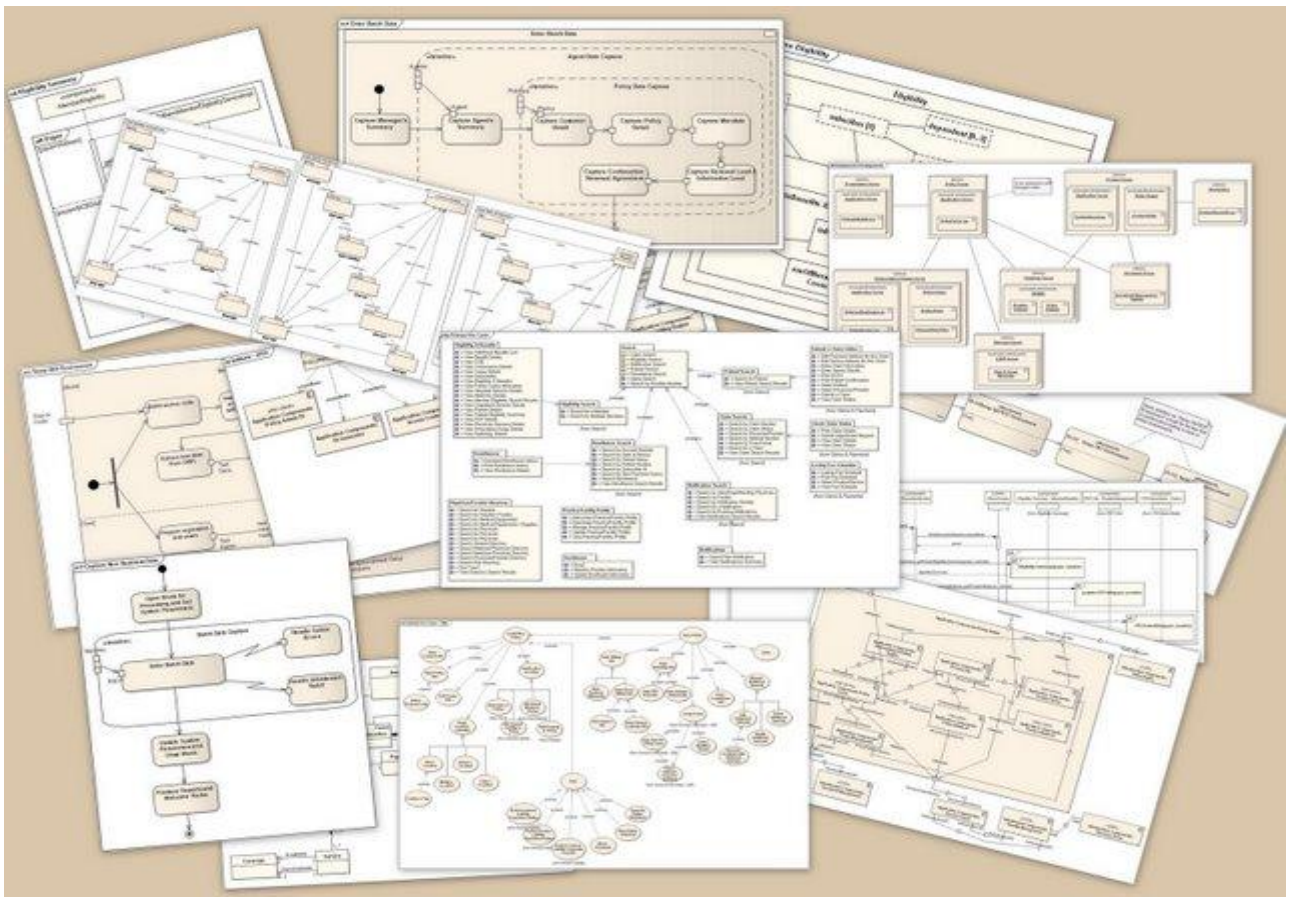
Χρησιμοποιείται για τη γραφική απεικόνιση, τον προσδιορισμό, την κατασκευή και την τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Μπορεί να χρησιμοποιηθεί σε διάφορες φάσεις ανάπτυξης, από την ανάλυση απαιτήσεων ως τον έλεγχο ενός ολοκληρωμένου συστήματος.

Αποτελείται από ένα σύνολο προσυμφωνημένων συμβόλων και διαγραμμάτων που επιτρέπουν:





- Την αναπαράσταση μιας στατικής δομής ενός συστήματος χρησιμοποιώντας διαγράμματα κλάσεων.
- Τη μοντελοποίηση της συμπεριφοράς των αντικειμένων με διαγράμματα καταστάσεων.
- Την εμφάνιση των ορίων ενός συστήματος και των βασικών λειτουργιών του, χρησιμοποιώντας περιπτώσεις χρήσης.
- Την επέκταση της λειτουργικότητας με στερεότυπα.
- Την επεξήγηση των περιπτώσεων χρήσης με διαγράμματα αλληλεπίδρασης.
- Τη μοντελοποίηση της εργασιακής ροής με διαγράμματα δραστηριοτήτων.
- Την αποκάλυψη της υλοποίησης της αρχιτεκτονικής με διαγράμματα συστατικών και ανάπτυξης.



Εικόνα 7-κολάζ

Τα διαγράμματα κλάσεων της UML χρησιμοποιούν γεωμετρικά σχήματα για να συμβολίσουν τα αντικείμενα, τις κλάσεις και τις διασυνδέσεις. Χρησιμοποιούνται για να συνδέουν αυτά τα σχήματα και να υποδηλώνουν τον τρόπο που κληρονομούν ή εξαρτώνται μεταξύ τους. Τα αντικείμενα που είναι της ίδιας κλάσης αναπαριστώνται με ένα μόνο γεωμετρικό σχήμα. Όταν ένα αντικείμενο χρησιμοποιεί κώδικα από κάποια άλλη κλάση (π.χ. καλώντας μία



μέθοδό της), σύμφωνα με το πρότυπο της UML υπάρχει μία εξάρτηση μεταξύ τους η οποία αναπαρίσταται με μία διακεκομμένη γραμμή.

Μια εξάρτηση μπορεί να είναι :

- Συσχέτιση (association), ένας τύπος εξάρτησης που υπονοεί πραγματική συνύπαρξη στη μνήμη στιγμιότυπων των συμμετεχόντων κλάσεων κατά τον χρόνο εκτέλεσης,
- Συνάθροιση (aggregation), ένας τύπος συσχέτισης ο οποίος σημαίνει ότι το ένα αντικείμενο μπορεί να περιέχει στιγμιότυπα της άλλης κλάσης ως γνωρίσματα του.
- Σύνθεση (composition), ένας πιο ισχυρός τύπος συνάθροισης που υπονοεί πως ο χρόνος ζωής των αντικειμένων είναι κοινός (δημιουργούνται και καταστρέφονται στη μνήμη ταυτόχρονα).

Καθεμία από αυτές τις σχέσεις συμβολίζεται οπτικά με έναν διαφορετικό τύπο γραμμής μεταξύ των συμμετεχόντων κλάσεων, ενώ μπορεί να υπάρχουν και εξαρτήσεις οι οποίες δεν είναι καν συσχετίσεις.

### 3.1.7 Τι είναι το Netbeans



Εικόνα 8-Netbeans symbol

Το NetBeans είναι ένα επιτυχημένο ερευνητικό έργο ανοιχτής πηγής (open source) με μεγάλο αριθμό χρηστών. Η Sun Microsystems ίδρυσε το ερευνητικό έργο

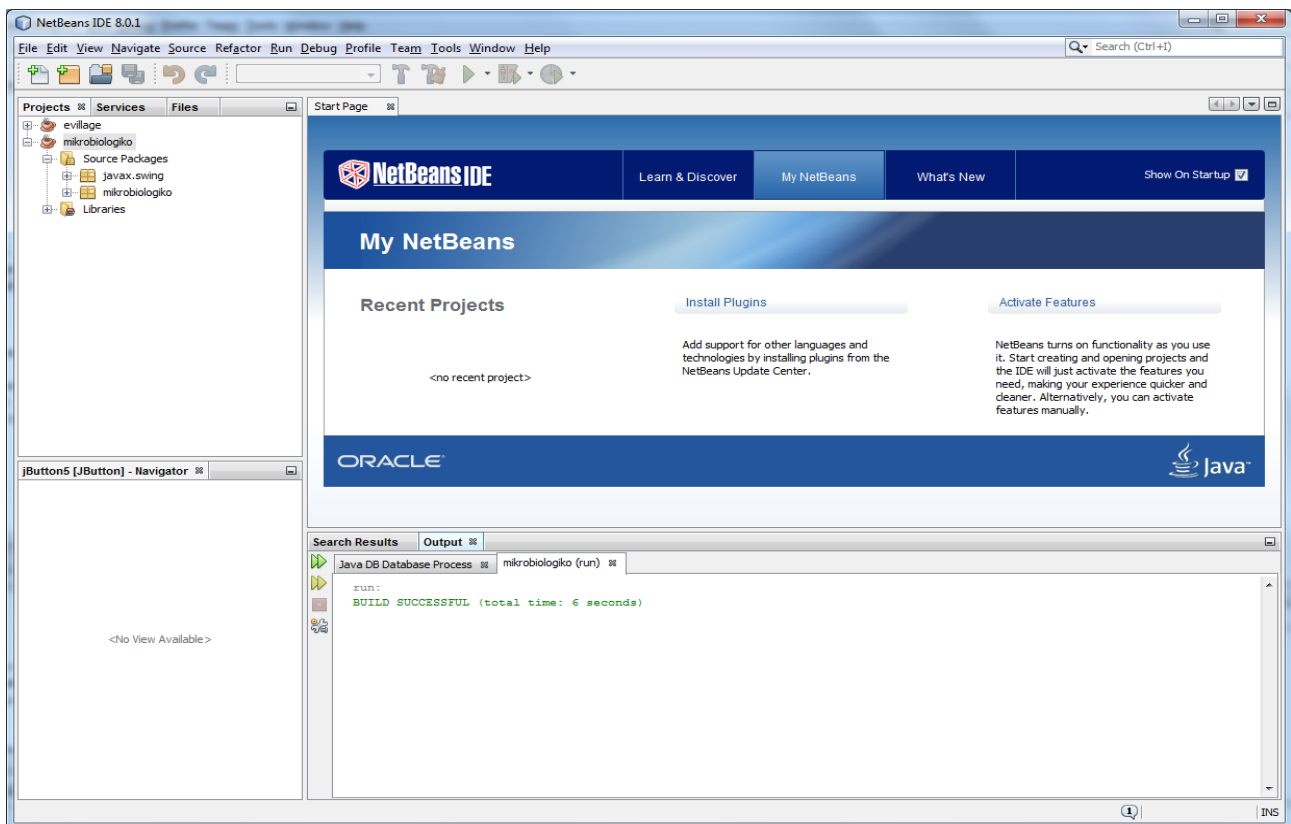


ανοιχτής πηγής NetBeans τον Ιούνιο του 2000 και συνεχίζει να είναι ο κύριος ανάδοχος.

Σήμερα δύο ερευνητικά έργα υπάρχουν: Το NetBeans IDE και το NetBeans Platform. Το NetBeans IDE είναι ένα περιβαλλοντικό ανάπτυγμα IDE - ένα εργαλείο στους προγραμματιστές για να γράψουν, να κάνουν compile, debug και να αναπτύξουν προγράμματα. Είναι γραμμένο σε Java - αλλά μπορεί να υποστηρίξει όλες τις γλώσσες προγραμματισμού. Υπάρχει επίσης ένας μεγάλος αριθμός υπομονάδων (modules) που βοηθάνε στην επέκταση της λειτουργικότητας του NetBeans IDE. Το NetBeans IDE είναι ένα ελεύθερο προϊόν δίχως περιορισμούς στον τρόπο χρησιμοποίησής του.

Διαθέσιμο επίσης είναι το NetBeans Platform; ένα εκτατό θεμέλιο αποτελούμενο από υπομονάδες (modular) που χρησιμοποιείται σαν βάση λογισμικού για τη δημιουργία μεγάλων επιτραπέζιων (desktop) εφαρμογών. Οι ISV συνεργάτες διαθέτουν προσθήκες, επιπρόσθετα προγράμματα (plug-ins) που εύκολα συνενώνονται στο Platform και μπορούν επίσης να χρησιμοποιηθούν για την ανάπτυξη άλλων εργαλείων και λύσεων.

Και τα δύο τα προϊόντα είναι ανοιχτής πηγής (open source) και ελεύθερα για εμπορική ή μη χρήση.



Εικόνα 9-Netbeans workspace



### 3.1.8 Συνδιασμός Java με SQL

#### JDBC

- Η τεχνολογία JDBC (σημαίνει Java Database Connectivity) μας επιτρέπει να έχουμε πρόσβαση σε μία σχεσιακή βάση δεδομένων μέσα από το πρόγραμμά μας.
- 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.
- Οι απαραίτητες κλάσεις ορίζονται από τη βιβλιοθήκη `java.sql`.

#### Επικοινωνία με τη Βάση

- Για να στείλουμε εντολές στη βάση χρησιμοποιούμε την κλάση `java.sql.Statement`. Αντικείμενα της κλάσης αυτής μας επιστρέφουν τα αντικείμενα της κλάσης `java.sql.Connection`.
- Για να μεταβάλλουμε τη βάση (δηλαδή για να ορίσουμε, να ανανεώσουμε ή να σβήσουμε κάποιους πίνακες ή εγγραφές) χρησιμοποιούμε τη μέθοδο `executeUpdate` της κλάσης `java.sql.Statement`. Η μέθοδος `executeUpdate` δέχεται ένα `string` με την εντολή SQL που θέλουμε να εκτελέσουμε και επιστρέφει τον αριθμό των γραμμών που επηρεάστηκαν αν η εντολή είναι `insert`, `update`, `delete`, ή 0 αν η εντολή δεν επιστρέφει τίποτα.
- Για να κάνουμε αναζήτηση στοιχείων στη βάση χρησιμοποιούμε τη μέθοδο `executeQuery` της κλάσης `java.sql.Statement`. Η μέθοδος



επιστρέφει ένα αντικείμενο της κλάσης `java.sql.ResultSet` με τα αποτελέσματα της αναζήτησής μας.

- Αφού ολοκληρώσουμε την επικοινωνία μας με τη βάση κλείνουμε το αντικείμενο `java.sql.Statement` με τη μέθοδο `close`.

### Μεθοδολογία Ανάπτυξης

- Σχεδιάζουμε τους πίνακες της βάσης, λαμβάνοντας υπόψη:  
Τύπους δεδομένων, keys, foreign keys, indices και τους περιορισμούς (null, not null, unique, κλπ.)
- Δημιουργούμε τη βάση στο σύστημα διαχείρισης βάσης δεδομένων.
- Δημιουργούμε χρήστες με συγκεκριμένες δυνατότητες πρόσβασης στη βάση.
- Κατασκευάζουμε και ελέγχουμε τον SQL κώδικα σε ένα εργαλείο της επιλογής μας.
- Γράφουμε και ελέγχουμε τον JDBC κώδικα που αντιστοιχεί στον SQL κώδικα που φτιάξαμε.
- Ενσωματώνουμε τον JDBC κώδικα στην εφαρμογή μας.

## 3.2 Σημαντικοί στόχοι για την ολοκλήρωση της πτυχιακής

- Ολοκλήρωση της έρευνας state of the art.....15
- Ολοκλήρωση της ανάλυσης του προβλήματος.....15
- Ολοκλήρωση του σχεδιασμού ανάπτυξης της πτυχιακής.....21
- Υλοποίηση του τεχνικού μέρους της πτυχιακής εργασίας.....28
- Έλεγχος λειτουργίας τεχνικού μέρους.....7
- Συγγραφή Αναφοράς Εργασίας.....21
- Υποβολή αίτησης αξιολόγησης εργασίας.....1
- Προετοιμασία παρουσίασης αναφοράς.....14
- Παρουσίαση αναφοράς.....1



ITEM	10/14	11/14	12/14	1/15	2/15
Ολοκλήρωση της έρευνας state of the art	■				
Ολοκλήρωση της ανάλυσης του προβλήματος		■			
Ολοκλήρωση του σχεδιασμού ανάπτυξης της πτυχιακής		■			
Υλοποίηση του τεχνικού μέρους της πτυχιακής εργασίας			■		
Έλεγχος λειτουργίας τεχνικού μέρους				■	
Συγγραφή Αναφοράς Εργασίας				■	
Υποβολή αίτησης αξιολόγησης εργασίας					■
Προετοιμασία παρουσίασης αναφοράς					■
Παρουσίαση αναφοράς					■

## 4.Κύριο μέρος πτυχιακής

### 4.1 Ανάλυση Προβλήματος

Θέλουμε να αποθηκεύουμε τις παρακάτω πληροφορίες για ένα μικροβιολογικό εργαστήριο.

-Τα στοιχεία κάθε ασθενή. Όνομα, επίθετο τηλέφωνο διεύθυνση και αριθμό κοινωνικών ασφαλίσεων.

-Τα διάφορα είδη εξετάσεων που υπάρχουν. Όνομα, κωδικό, αρρώστια για την οποία γίνεται.

- Τα διάφορα αντιδραστήρια που υπάρχουν. Όνομα , κωδικός και αποθέματα που υπάρχουν. Επίσης και το ελάχιστο αποθεματικό που πρέπει να έχουμε από το



κάθε αντιδραστήριο.

-Οι παραγγελίες που πραγματοποιούνται για την προμήθεια νέων ποσοτήτων αντιδραστηρίων. Έχουν κωδικό, ημερομηνία που έγινε και ημερομηνία που παραλείφθηκε.

1. Οι πελάτες κάνουν εξετάσεις. Θέλουμε να αποθηκεύουμε την ημερομηνία και τα αποτελέσματα αυτών. Επίσης θέλουμε να ξέρουμε αν ο πελάτης έχει πάρει ή όχι τα αποτελέσματα των εξετάσεων.

2. Μια εξέταση χρησιμοποιεί κάποια αντιδραστήρια. Θέλουμε με το που πραγματοποιείται μια εξέταση να ανανεώνουμε τις ποσότητες τους.

3. Μια παραγγελία μπορεί να αφορά πολλά αντιδραστήρια. Θέλουμε να αποθηκεύουμε την ποσότητα του καθενός.

Το σύστημα μας πρέπει να υποστηρίζει :

-Εισαγωγή όλης της παραπάνω πληροφορίας τμηματικά. Πρέπει να υποστηρίζεται η εισαγωγή εξετάσεων που θέλει να πραγματοποιήσει κάποιος ασθενής. Πρέπει να δίνεται η δυνατότητα ενημέρωση για τα αποτελέσματα αυτών των εξετάσεων. Επίσης κάθε φορά που γίνεται ενημέρωση για εξετάσεις που θέλει να πραγματοποιήσει κάποιος πελάτης να γίνεται αυτόματη μείωση των αποθεμάτων των αντιδραστηρίων.

-Συγκεντρωτική αναφορά των πελατών που δεν έχουν πάρει ακόμα τα αποτελέσματα κάποιων εξετάσεων που έκαναν.

- Ενημέρωση για παραλαβή κάποιας παραγγελίας.

#### 4.1.2 Απαιτήσεις Συστήματος

Βασική προϋπόθεση είναι η ύπαρξη ενός server που φιλοξενεί την βάση δεδομένων και οι υπολογιστές που θα χρησιμοποιούν το πρόγραμμα, να υποστηρίζουν Java.

Από την άλλη το πρόγραμμα μας δεν απαιτεί από τον χρήστη να έχει εξειδικευμένες γνώσεις για να το χρησιμοποιήσει.

#### 4.2 Σχεδιασμός Υλοποίησης

Παρακάτω αναφέρονται συνοπτικά τα βήματα που ακολουθήσαμε :

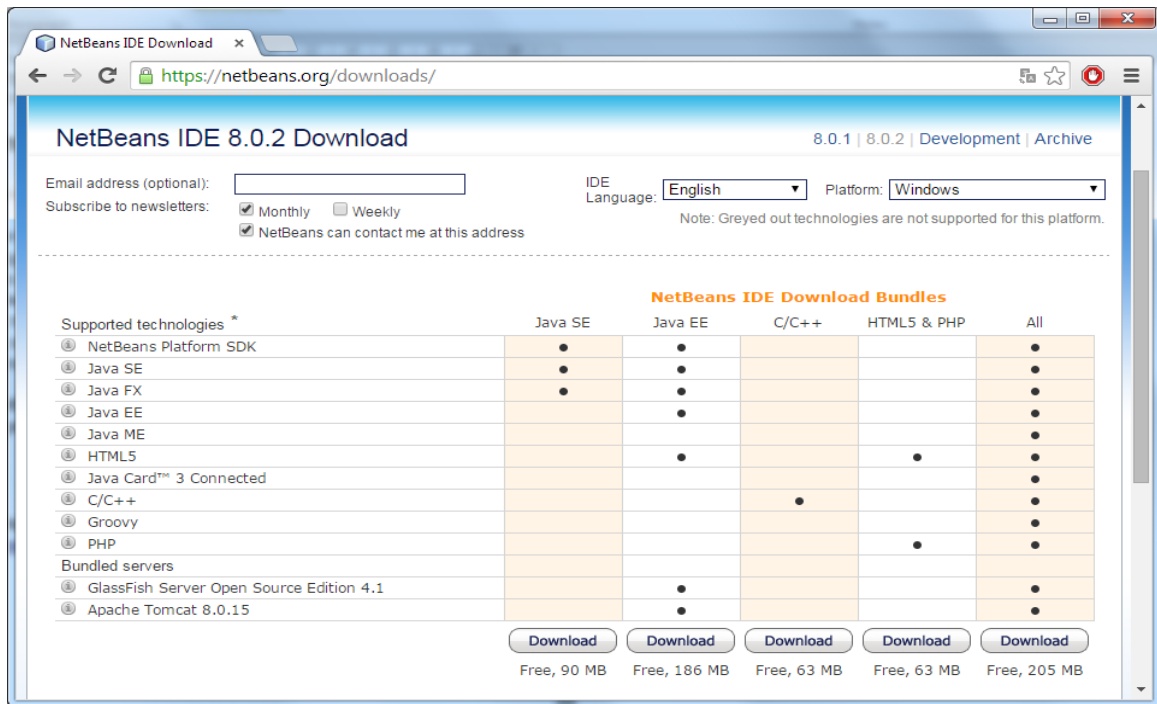


- Βασική εκμάθηση των γλωσσών προγραμματισμού SQL και Java
- Έρευνα για το ποιο πρόγραμμα θα χρησιμοποιήσουμε για την υλοποίηση
- Έρευνα για τον τρόπο λειτουργίας του Netbeans
- Αναζήτηση στο διαδίκτυο για παρόμοιες εφαρμογές
- Έρευνα για την δομή της βάσης δεδομένων που θα χρησιμοποιήσουμε
- Έρευνα για την δομή των τύπων περιεχομένων μας
- Αναζήτηση λύσεων στο διαδίκτυο προκειμένου να γίνει εύχρηστη η εφαρμογή μας
- Εγκατάσταση του Netbeans
- Υλοποίηση του προγράμματος

## 5. Υλοποίηση

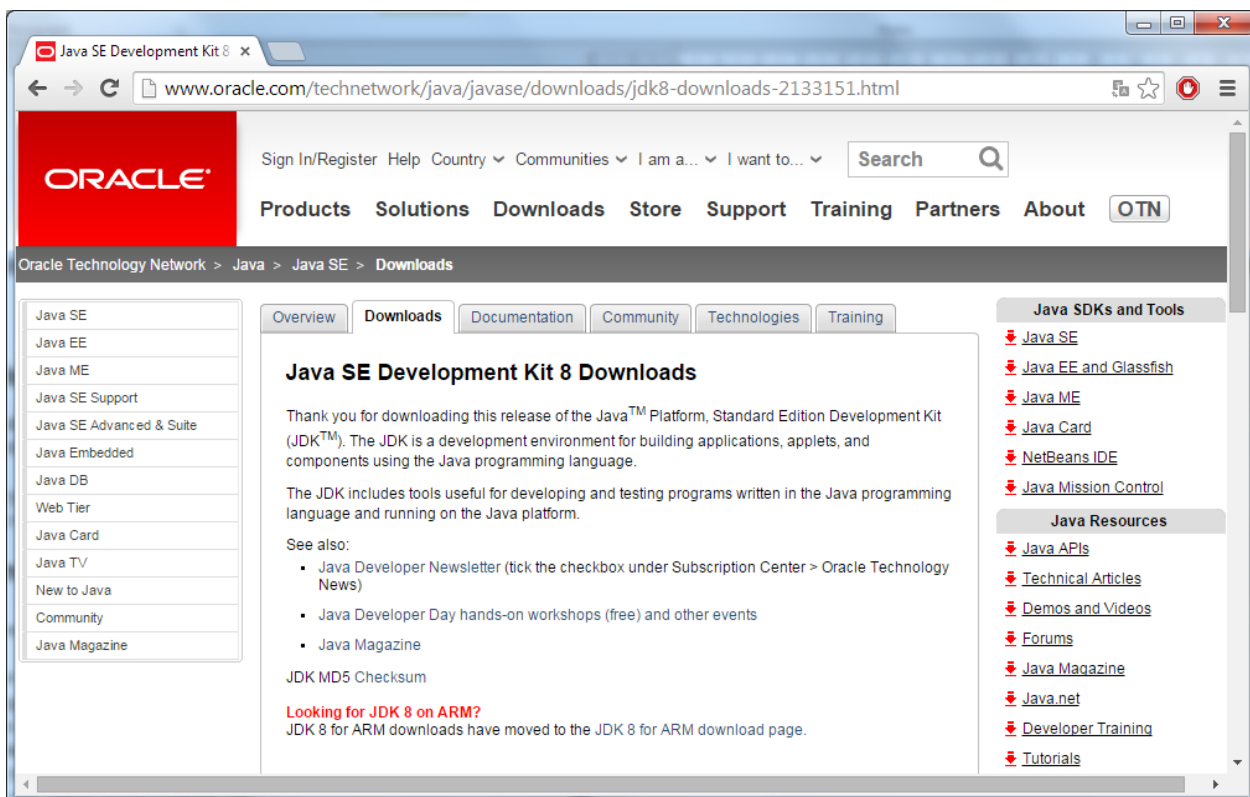
Αρχικά κατεβάσαμε και εγκαταστήσαμε το Netbeans από τη σελίδα <https://netbeans.org/downloads/> :





Εικόνα 10-Netbeans site

Προσθέσαμε και το plugin Java SE Development Kit 8 το οποίο είναι απαραίτητο για την λειτουργία του Netbeans



Εικόνα 11-JDK

Στη συνέχεια προχωρήσαμε στην υλοποίηση του προγράμματος που αποτελείται από 3 βασικά βήματα :

1. Δημιουργία της βάσης δεδομένων



- 2. Δημιουργία της φόρμας του προγράμματος σε Java
- 3. Δημιουργία συναρτήσεων και κώδικα του προγράμματος

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes Databases, Java DB, Drivers, Servers, Maven Repositories, Cloud, Hudson Builders, and Task Repositories. The code editor shows the following Java code:

```
public void DoConnect ( ) {  
    try {  
        String host = "jdbc:derby://localhost:1527/mikro";  
        String uName = "app";  
        String uPass= "app";  
        Connection con=DriverManager.getConnection(host, uName, uPass);  
  
        Statement stmt = con.createStatement ( );  
        String SQL = "SELECT * FROM ASTHENEIS";  
        String SQL1 = "SELECT * FROM STOIXEIA";  
        String SQL2 = "SELECT * FROM ANTIDRASTIRIA";  
        String SQL3 = "SELECT * FROM PARAGELIA";  
        ResultSet rs = stmt.executeQuery(SQL);  
        pinakas.setModel(DbUtils.resultSetToTableModel(rs));  
  
        ResultSet rs1 = stmt.executeQuery(SQL1);  
        pinakas1.setModel(DbUtils.resultSetToTableModel(rs1));  
        pinakas2.setModel(pinakas1.getModel());  
    }  
}
```

Εικόνα 12-project

## 5.1 Φόρμες προγράμματος

### 5.1.1 Φόρμα εισαγωγής



The screenshot shows a web application window titled "Μικροβιολογικό Εργαστήριο". It has four tabs: "Εισαγωγή", "Στοιχεία Ασθενή", "Εξετάσεις", and "Παραγγελίες". The "Εισαγωγή" tab is active, showing a form with two panels. The left panel, titled "Εισαγωγή νέου ασθενή", has a cyan background and contains five text input fields: "Όνομα", "Επίθετο", "Τηλέφωνο", "Διεύθυνση", and "ΑΚΑ". Below these fields is a note "(Συμπληρώστε απαραίτητα όλα τα πεδία)" and a "Αποθήκευση" button. The right panel, titled "Κλείσιμο εξέτασης", has a light purple background and contains three fields: "ΑΚΑ Ασθενή" (text input), "Εξέταση" (dropdown menu with "Επιλογή εξέτασης" selected), and "Ημ/νια" (date input showing "8 / 12 / 2014" with a calendar icon). A "Αποθήκευση" button is also present at the bottom right of this panel.

Εικόνα 13-φόρμα εισαγωγής

Σε αυτή τη φόρμα έχουμε την δυνατότητα να εισάγουμε τα στοιχεία ενός νέου ασθενή τα οποία αποθηκεύονται στη βάση δεδομένων μας(ονομα,επίθετο,τηλ κλπ).

Επίσης μπορούμε να κλείσουμε ημερομηνία εξέτασης για έναν ασθενή ο οποίος υπάρχει ήδη στη βάση μας χρησιμοποιώντας το ΑΚΑ του.

Ο χρήστης το μόνο που έχει να κάνει σε αυτή τη φόρμα, είναι να συμπληρώσει όλα τα απαραίτητα στοιχεία και να πατήσει το αντίστοιχο κουμπί αποθήκευσης.

### 5.1.2 Στοιχεία ασθενή



ΑΚΑ	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΤΗΛΕΦΩΝΟ	ΔΙΕΥΘΥΝΣΗ
1234.0	GEORGE			
4.0	gianis	gad	54511	hgjh 6
22.0	manos	anto	777777	ghfvghv 5
1.0	qwe	rty	0986	fgfg 65
2.0	aaa	qqqq	23245	frdetv 4
7.0	asdd	ynbn	456575	b fd 3
3.0	fasf	faffac	464566	gedr 78
9.0	jkhknb	CHGBN	9879	hjvgf 5

ID	ΑΚΑ	ΕΧΕΤΑΣΙ	ΗΜΕΡΟΜΙΝΙΑ	ΑΠΟΤΕΛΕΣΜΑ	ΠΑΡΑΛΑΒΗ
1	4	exetasi 1	07-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
2	5	exetasi 1	08-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
3	251156	exetasi 1	21-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
4	415895	Επιλογή εξέτασης	21-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
5	6546	Επιλογή εξέτασης	15-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
6	3	exetasi 1	08-11-2014	C:\Users\Niko\$\\...	DIATHESIMA
7	2	exetasi 1	05-11-2014		EKREMOYN
8	2	exetasi 1	28-11-2014		EKREMOYN
9	nikos	exetasi 1	08-11-2014	C:\Users\Niko\$\\...	DIATHESIMA

Εικόνα 14-στοιχεία ασθενή

Σε αυτή τη φόρμα μπορούμε να προβάλουμε όλα τα στοιχεία για τους ασθενείς που έχουμε καταχωρημένους στη βάση μας.

Πιο αναλυτικά, συμπληρώνοντας το ΑΚΑ του ασθενή στο αντιστοιχικό πεδίο της φόρμας και πατώντας το κουμπί αναζήτηση, ο πρώτος πίνακας μας εμφανίζει τα στοιχεία του ασθενή (ΑΚΑ, όνομα, επίθετο, τηλέφωνο και διεύθυνση)

Ο δεύτερος πίνακας (ο οποίος αφορά τα αποτελέσματα των εξετάσεων) μας εμφανίζει τις εξετάσεις του αντιστοιχικού ασθενή, την ημερομηνία εξέτασης καθώς και το αν είναι διαθέσιμα προς παραλαβή τα αποτελέσματα των εξετάσεων του. Αν τα αποτελέσματα αυτά είναι διαθέσιμα, με το κουμπί Προβολή αποτελεσμάτων, αυτά εμφανίζονται στην οθόνη μας.

### 5.1.3 Εξετάσεις



ID	ΑΚΑ	EXETASI	IMEROMINIA	APOTELESMA	PARALABI
1	4	exetasi 1	07-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
2	5	exetasi 1	08-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
3	251156	exetasi 1	21-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
4	415895	Επιλογή εξέτασης	21-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
5	6546	Επιλογή εξέτασης	15-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
6	3	exetasi 1	08-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
7	2	exetasi 1	05-11-2014		EKREMOYN
8	2	exetasi 1	28-11-2014		EKREMOYN
9	nikos	exetasi 1	08-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
10	1	exetasi 1	28-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
11	1	exetasi 1	04-11-2014	C:\Users\Niko\$WDo...	DIATHESIMA
12	2813	EXETASI-3	11-12-2014	C:\Users\Niko\$WDe...	DIATHESIMA
13	125489	EXETASI-3	27-07-2014	C:\Users\Niko\$WDo...	DIATHESIMA

Επιλογή αποτελεσμάτων:  Κωδ.Εξέτασης:    
Θέση Αρχείου:   
(Αποθήκευση και Ενημέρωση αποτελεσμάτων εξέτασης)

Εικόνα 15-εξετάσεις

ID	ΑΚΑ	EXETASI	IMEROMINIA	APOTELESMA	PARALABI
7	2	exetasi 1	05-11-2014		EKREMOYN
8	2	exetasi 1	28-11-2014		EKREMOYN

Επιλογή αποτελεσμάτων:  Κωδ.Εξέτασης:    
Θέση Αρχείου:   
(Αποθήκευση και Ενημέρωση αποτελεσμάτων εξέτασης)

Εικόνα 16-εξετάσεις που εκκρεμούν



Η φόρμα εξετάσεις, έχει να κάνει αποκλειστικά με τις εξετάσεις και τα αποτελέσματα τους. Αρχικά η φόρμα έχει έναν πίνακα στον οποίο αναγράφονται όλα τα στοιχεία των εξετάσεων. Το όνομα της εξέτασης, η ημερομηνία πραγματοποίησης της, το αρχείο με τα αποτελέσματα και το αν η εξέταση είναι διαθέσιμη προς παραλαβή.

Με το κουμπί εκκρεμούν μπορούμε να προβάλουμε στον πίνακα όλες αυτές τις εξετάσεις οι οποίες δεν είναι διαθέσιμες προς παραλαβή. Ξαναπατώντας το κουμπί εμφανίζονται στον πίνακα πάλι όλα τα αποτελέσματα (διαθέσιμα και μη).

Με το κουμπί άνοιγμα επιλέγουμε το αρχείο των αποτελεσμάτων της εξέτασης, μόλις το επιλέξουμε, αναγράφεται η διαδρομή του αρχείου που επιλέξαμε στο πεδίο θέση αρχείου. Έπειτα συμπληρώνοντας το πεδίο κωδικός εξέτασης και πατώντας το κουμπί αποθήκευση, τα αποτελέσματα αυτά καταχωρούνται στην αντιστοιχη εξέταση. Με τον ίδιο τρόπο μπορούμε να ενημερώσουμε τα ήδη υπάρχοντα αποτελέσματα.

Αν τα αποτελέσματα της εξέτασης είναι ήδη διαθέσιμα, συμπληρώνοντας το πεδίο κωδικός εξέτασης με τον αντιστοιχο κωδικό και πατήσουμε το κουμπί προβολή, τα βλέπουμε στην οθόνη.

## 5.1.4 Παραγγελίες

Μικροβιολογικό Εργαστήριο

Εισαγωγή Στοιχεία Ασθενή Εξετάσεις Παραγγελίες

Εισάγεται Ποσότητες Επιλέξτε Ημερομηνία Παραλαβή Παραγγελίας

ANT-1 0 m l 18 / 12 / 2014 Κωδ. Παραγγελίας:

ANT-2 0 m l Παραλήφθηκε

ANT-3 0 m l

ANT-4 0 m l Αποστολή Παραγγελίας

Παραγγελίες αντιδραστηρίων Αποθέματα αντιδραστηρίων

ID	DATE	PARALAVI
100	01-11-2014	PARALIFTHIKE
101	04-12-2014	PARALIFTHIKE
102	11-12-2014	EKREMEI
103	11-12-2014	PARALIFTHIKE
104	11-12-2014	EKREMEI

Εικόνα 17-παραγγελίες 1

Στην τελευταία φόρμα του προγράμματός μας, έχουμε ένα πίνακα ο οποίος

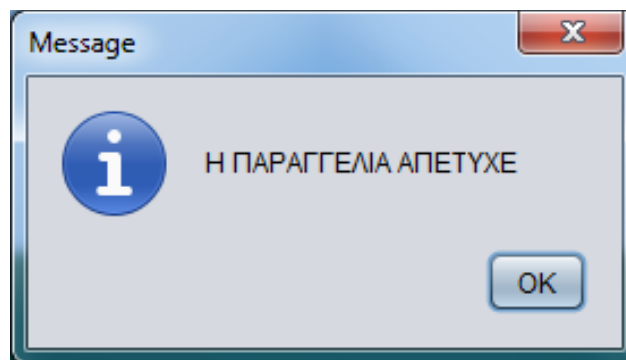


αναγράφει όλες τις παραγγελίες που έχουμε πραγματοποιήσει, την ημερομηνία που κάναμε την παραγγελία, καθώς και το αν έχει παραληφθεί ή όχι. Επίσης στη δεύτερη καρτέλα του πίνακα(αποθέματα αντιδραστηρίων) αναφέρονται τα συνολικά αποθέματα των αντιδραστηρίων που έχουμε, καθώς και το ελάχιστο ποσό που θα μπορούμε να έχουμε.

Παραγγελίες αντιδραστηρίων		Αποθέματα αντιδραστηρίων	
ID	IDOS	ΑΡΟΘΗΜΑ	ΕΛΑΧΙΣΤΟ
1	ANT-1	1200	250
2	ANT-2	290	250
3	ANT-3	700	250
4	ANT-4	380	250

Εικόνα 18-παραγγελίες 2

Αν τα αποθέματα μας πέσουν κάτω από το ελάχιστο, το πρόγραμμα πετάει μήνυμα, έτσι ώστε να ενημερωθούμε για να φροντίσουμε να παραγγείλουμε νέες ποσότητες αντιδραστηρίων. Σε περίπτωση που τα αποθέματα δεν επαρκούν, δεν μπορούμε να πραγματοποιήσουμε νέα εξέταση και μας εμφανίζεται το παρακάτω μήνυμα.



Εικόνα 19-αποτυχία παραγγελίας

Στο πάνω μέρος της φόρμας επιλέγουμε τις αντίστοιχες ποσότητες αντιδραστηρίων που επιθυμούμε να παραγγείλουμε και με το κουμπί αποστολή παραγγελίας, στέλνουμε την παραγγελία και την καταχωρούμε στη βάση μας.

Τέλος, για να παραλάβουμε μια παραγγελία συμπληρώνουμε τον κωδικό της παραγγελίας στο αντίστοιχο πεδίο και πατάμε το κουμπί παραλήφθηκε. Με το πάτημα ανανεώνονται αυτόματα τα αποθέματα των αντιδραστηρίων.

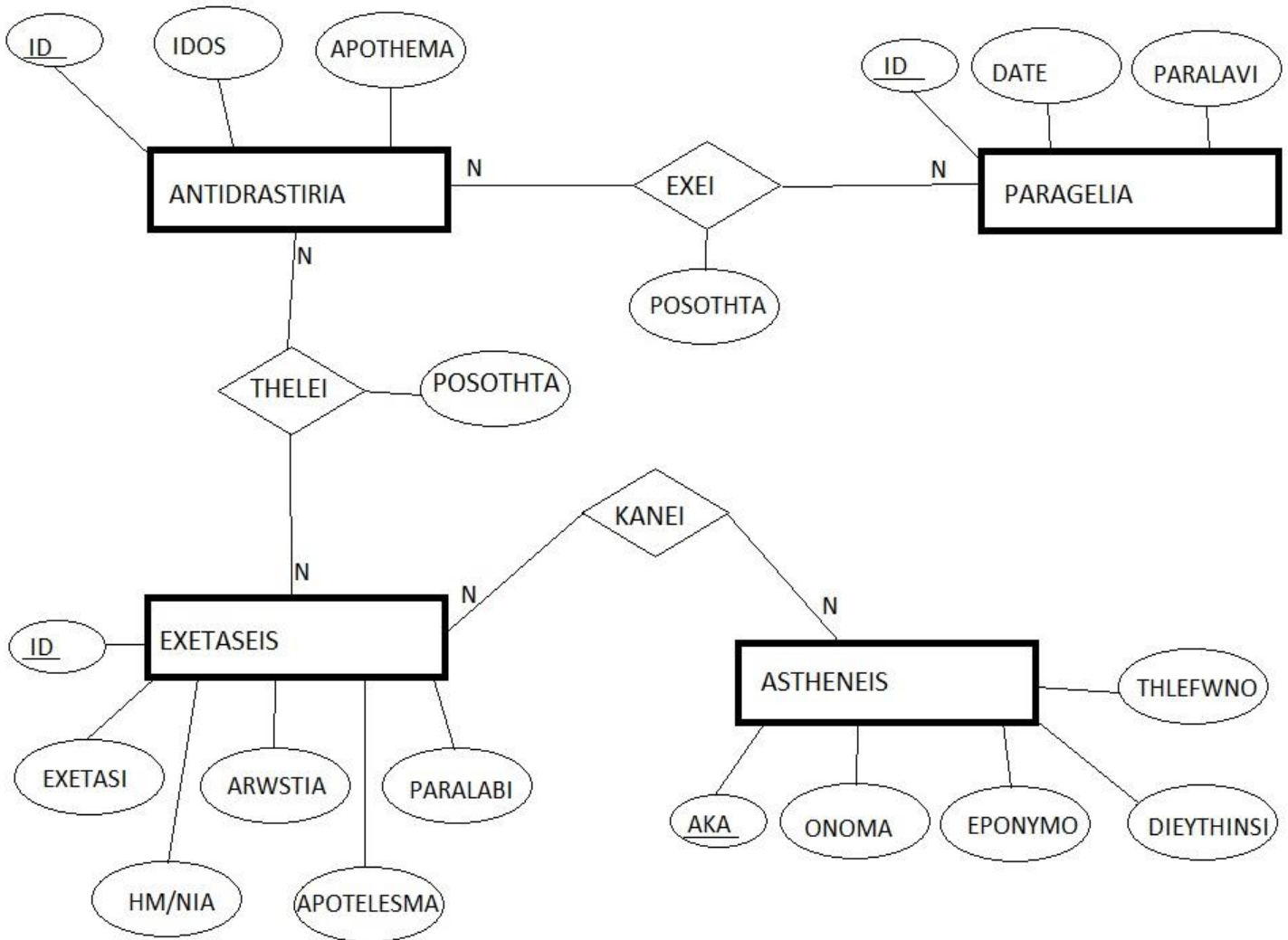
## 5.2 Βάση δεδομένων



### 5.2.1 Σχεδιασμός Βάσης (Ο/Σ και πίνακες)

Στη πρώτη φάση της σχεδίασης, φτιάξαμε ένα διάγραμμα οντοτήτων συσχετίσεων, για να παρουσιάσουμε το πως θα είναι η βάση και στη συνέχεια ακολουθώντας κάποιους κανόνες (βήματα) μεταβήκαμε στο σχεσιακό σχήμα (πίνακες).

Παρακάτω φαίνεται το μοντέλο οντοτήτων συσχετίσεων που φτιάξαμε:



Εικόνα 20-Διάγραμμα Ο/Σ βάσης





Ακολουθούν οι πίνακες που βγήκαν από το παραπάνω διάγραμμα οντοτήτων συσχετίσεων:

ANTIDRASTIRIA (ID, IDOS, APOTHEMA, ELAXISTO)

ANTX (ID, EXETASI, ANTIDRASTIRIO, POSOTHTA)

ASTENEIS (AKA, ONOMA, EPONYMO, THLEFWNO, DIEYTHINSI)

EXETASEIS (ID, EXETASI, ARWSTIA)

PARAGELEIA (ID, DATE, PARALAVI)

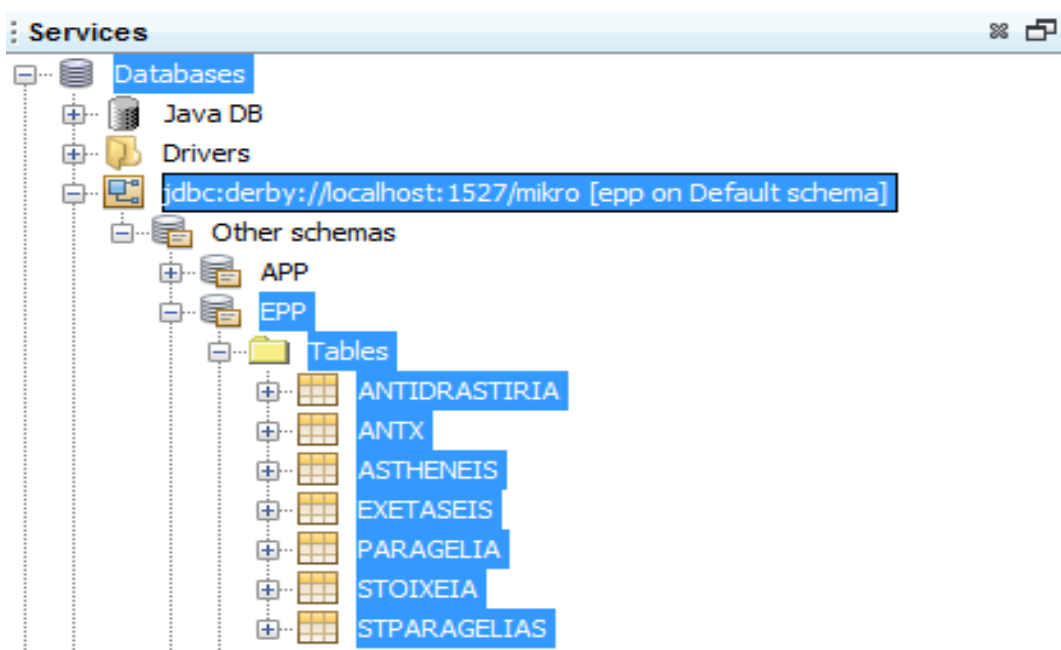
STOIXEIA (ID, AKA, EXETASI, IMEROMINIA, APOTELESMA, PARALAVI)

STPARAGELEIAS (ID, ANTIDRASTIRIO, POSOTHTA, PCODE)

Στο επόμενο στάδιο προχωράμε στη δημιουργία της βάσης δεδομένων χρησιμοποιώντας το Netbeans. Η βάση δημιουργείται σε ένα τοπικό server.

## 5.2.2 Υλοποίηση Βάσης στο Netbeans

Χρησιμοποιώντας το Netbeans δημιουργήσαμε τοπικά τη βάση δεδομένων του προγράμματος και την ονομάσαμε “mikro”.



Εικόνα 21-database

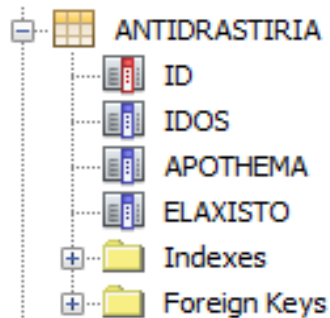
Όπως βλέπουμε στην εικόνα, η βάση δεδομένων μας αποτελείται από 7 πίνακες



(antidrastiria, antx, astheneis, paragelia, stoixeia, stparagelias).

### Αναλυτική περιγραφή των πινάκων :

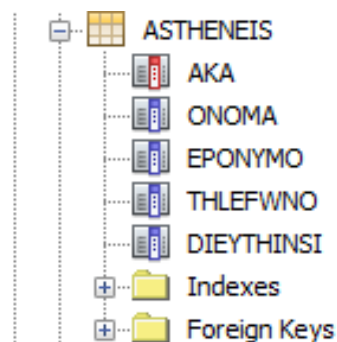
- ANTIDRASTIRIA



Εικόνα 22-πίνακας βάσης 1

Σε αυτό το πίνακα αποθηκεύουμε τα στοιχεία των αντιδραστηρίων. Όνομα(**IDOS**), κωδικός(**ID**) και αποθέματα(**APOTHEMA**) που υπάρχουν. Επίσης και το ελάχιστο αποθεματικό που πρέπει να έχουμε από το κάθε αντιδραστήριο(**ELAXISTO**).

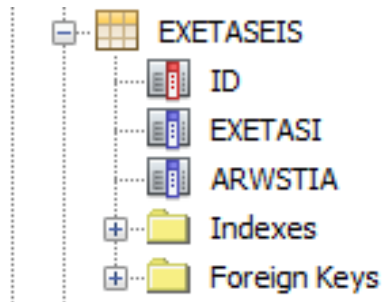
- ASTHENEIS



Εικόνα 23-πίνακας βάσης 2

Σε αυτό το πίνακα αποθηκεύουμε τα στοιχεία των ασθενών. Όνομα(**ONOMA**), επίθετο(**EPONYMO**) τηλέφωνο(**THLEFWNO**) διεύθυνση(**DIEYTHINSI**) και αριθμό κοινωνικών ασφαλίσεων(**AKA**).

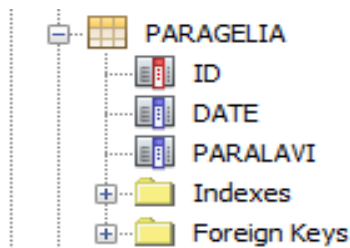
- EXETASEIS



Εικόνα 24-πίνακας βάσης 3

Σε αυτό το πίνακα αποθηκεύουμε τα διάφορα είδη των εξετάσεων. Όνομα(**ΕΞΕΤΑΣΙ**), κωδικό(**ID**), αρρώστια για την οποία γίνεται(**ΑΡΩΣΤΙΑ**).

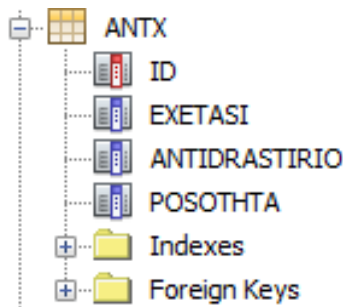
- **PARAGELIA**



Εικόνα 25-πίνακας βάσης 4

Σε αυτό τον πίνακα αποθηκεύουμε τα στοιχεία για τις παραγγελίες που πραγματοποιούνται για την προμήθεια νέων ποσοτήτων αντιδραστηρίων. Έχουν κωδικό(**ID**), ημερομηνία που έγινε(**DATE**) και ημερομηνία που παραλείφθηκε(**PARALAVI**).

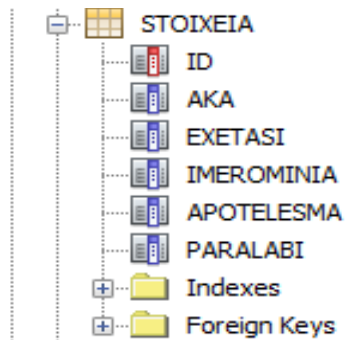
- **ANTX**



Εικόνα 26-πίνακας βάσης 5

Μια εξέταση(**ΕΞΕΤΑΣΙ**) χρησιμοποιεί κάποια αντιδραστήρια(**ΑΝΤΙΔΡΑΣΤΗΡΙΟ**). Θέλουμε να αποθηκεύσουμε την ποσότητα(**ΠΟΣΟΤΗΤΑ**) και το αντιδραστήριο που χρησιμοποιεί.

- **ΣΤΟΙΧΕΙΑ**



Εικόνα 27-πίνακας βάσης 6

Θέλουμε να αποθηκεύουμε την ημερομηνία(**IMEROMINIA**) και τα αποτελέσματα(**APOTELESMA**) της εξέτασης(**EXETASI**) κάθε ασθενή(**AKA**). Επίσης θέλουμε να αποθηκεύουμε αν ο ασθενής έχει πάρει ή όχι τα αποτελέσματα των εξετάσεων(**PARALABI**).

- STPARAGELIAS



Εικόνα 28-πίνακας βάσης 7

Μια παραγγελία(**PCODE**) μπορεί να αφορά πολλά αντιδραστήρια (**ANTIDRASTIRIO**). Θέλουμε να αποθηκεύουμε την ποσότητα(**POSOTHTA**) του καθενός.

### 5.3 Συναρτήσεις προγράμματος



- DoConnect

```
//H sinartisi DoConnect mas sindeei sthn vasi dedomenwn
public void DoConnect( ) {

    try{
        String host = "jdbc:derby://localhost:1527/mikro";
        String uName = "epp";
        String uPass= "epp";
        Connection con=DriverManager.getConnection(host, uName, uPass);

        Statement stmt = con.createStatement( );
        String SQL = "SELECT * FROM ASTHENEIS";
        String SQL1 = "SELECT * FROM STOIXEIA";
        String SQL2 = "SELECT * FROM ANTIDRASTIRIA";
        String SQL3 = "SELECT * FROM PARAGELIA";
        ResultSet rs = stmt.executeQuery(SQL);
        pinakas.setModel(DbUtils.resultSetToTableModel(rs));

        ResultSet rs1 = stmt.executeQuery(SQL1);
        pinakas1.setModel(DbUtils.resultSetToTableModel(rs1));
        pinakas2.setModel(pinakas1.getModel());

        ResultSet rs2 = stmt.executeQuery(SQL2);
        pinakas3.setModel(DbUtils.resultSetToTableModel(rs2));

        ResultSet rs3 = stmt.executeQuery(SQL3);
        pinakas4.setModel(DbUtils.resultSetToTableModel(rs3));

    }

    catch ( SQLException err ) {
        System.out.println(err.getMessage() );
    }

}
```

Εικόνα 29-Doconnect

Η συνάρτηση DoConnect μας συνδέει στη βάση δεδομένων και στη συνέχεια υποβάλει τα αρχικά ερωτήματα σ' αυτή, για να εμφανίσει στους πίνακες του προγράμματος τα δεδομένα της.

- DoImport



```
// Bazei dedomena ston pinaka ASTHENEIS ths vaseis dedomenwn
public void DoImport( ) {

    String o=on.getText();
    String e=ep.getText();
    String t=thl.getText();
    String d=die.getText();
    String a=ak.getText();
    Double al=Double.parseDouble(a);

    String insertStr="";

    try{

        String host = "jdbc:derby://localhost:1527/mikro";
        String uName = "epp";
        String uPass= "epp";
        Connection con=DriverManager.getConnection(host, uName, uPass);

        Statement stmt = con.createStatement( );
        insertStr="insert into EPP.ASTHENEIS (AKA,ONOMA,EPONYMO,THLEFWNO,DIEYTHINSI)
        +al+", "+"'+o+'"+", "+"'+e+'"+", "+"'+t+'"+", "+"'+d+'"+")";

        stmt.executeUpdate(insertStr);

        JOptionPane.showMessageDialog(forma.this, "Η ΕΙΣΑΓΩΓΗ ΚΑΤΑΧΩΡΗΘΗΚΕ (" +a+"

        getContentPane().removeAll();
        initComponents();
    }
}
```

Εικόνα 30-DoImport

Η συνάρτηση **DoImport** εισάγει τα στοιχεία των ασθενών στη βάση δεδομένων, τα οποία καταγράφονται από τη φόρμα που συμπληρώνει ο χρήστης.

- Fillexet

```
//Emfanizei tis exetaseis apo thn vash dedomenwn stis epiloges tu combobox
public void Fillexet(){

    try{

        String host = "jdbc:derby://localhost:1527/mikro";
        String uName = "epp";
        String uPass= "epp";
        Connection con=DriverManager.getConnection(host, uName, uPass);
        Statement stmt = con.createStatement( );
        String SQL = "SELECT * FROM EXETASEIS";
        ResultSet rs = stmt.executeQuery(SQL);

        while(rs.next()){
            String name=rs.getString("EXETASI");
            exet.addItem(name);
        }
    }
}
```

Εικόνα 31-Fillexet

Η **Fillexet** γεμίζει το combobox με τα είδη των εξετάσεων που υπάρχουν.



- parState

```
//Emfanizei tis exetaseis simfwna me to an ekremoun ta apotelesmata i oxi
public void parState() {

    if(par.isSelected()){

        par.setText("Εμφάνιση_όλων");

        try{
            String host = "jdbc:derby://localhost:1527/mikro";
            String uName = "app";
            String uPass= "app";
            Connection con=DriverManager.getConnection(host, uName, uPass);

            Statement stmt = con.createStatement( );
            String SQL = "SELECT * FROM STOIXEIA WHERE PARALABI='EKREMOYN'";
            ResultSet rs = stmt.executeQuery(SQL);
            pinakas2.setModel(DbUtils.resultSetToTableModel(rs));

        }

    }

}
```

Εικόνα 32-parState

Η συνάρτηση **parState** μας εμφανίζει σε έναν πίνακα τις εξετάσεις, σύμφωνα με το αν κρεμούν ή όχι τα αποτελέσματα της.

- Auxisi

```
Connection con=DriverManager.getConnection(host, uName, uPass);
Statement stmt = con.createStatement( );
Statement stmt1 = con.createStatement( );
String SQL = "SELECT * FROM ANTIDRASTIRIA";
String SQL1 = "SELECT * FROM STPARAGELIAS WHERE PCODE="+code;
ResultSet rs = stmt.executeQuery(SQL);
ResultSet rs1 = stmt1.executeQuery(SQL1);

while(rs.next() && rs1.next()){
    timi=rs.getInt("APOTHEMA");
    timi1=Integer.parseInt(rs1.getString("POSOTHTA"));
    int synolo=timi+timi1;

    String inse="";

}
```

Η συνάρτηση **auxisi** χρησιμοποιείται για να ενημερώσει τα αντιδραστήρια, όταν παραλάβουμε μία παραγγελία που έχουμε κάνει.



- Meiwsi

```
Connection con=DriverManager.getConnection(host, uName, uPass);
Statement stmt = con.createStatement( );
Statement stmt1 = con.createStatement( );
String SQL = "SELECT * FROM EPP.ANTX WHERE EXETASI="+""+exeta+"";

ResultSet rs = stmt.executeQuery(SQL);

while(rs.next()){
    timi=rs.getInt("POSOTHTA");
    anti=rs.getString("ANTIDRASTIRIO");

    String SQL1 = "SELECT * FROM EPP.ANTIDRASTIRIA WHERE IDOS="+""+anti+"";
    ResultSet rs1 = stmt1.executeQuery(SQL1);
    while(rs1.next()){
        timi1=rs1.getInt("APOTHEMA");
    }
}
```

Εικόνα 33-meiwsi

Η συνάρτηση **meiwsi** χρησιμοποιείται για να ενημερώσει τα αντιδραστήρια, όταν κλείσουμε μία νέα εξέταση για κάποιο ασθενή. Αν δεν υπάρχουν διαθέσιμα αποθέματα αντιδραστηρίων για την πραγματοποίηση της εξέτασης, η συνάρτηση σταματάει να εκτελείται και ο χρήστης ενημερώνεται με ένα μήνυμα.

Εαν η εξέταση αυτή προκαλέσει μείωση των αποθεμάτων κάτω από το ελάχιστο, εμφανίζεται το μήνυμα <<ΤΑ ΠΑΡΑΚΑΤΩ ΑΝΤΙΔΡΑΣΤΗΡΙΑ ΕΧΟΥΝ ΠΕΣΕΙ ΚΑΤΩ ΑΠΟ ΤΟ ΕΛΑΧΙΣΤΟ:>>. Αυτό δεν επηρεάζει το κλείσιμο της εξέτασης.





Παρακάτω φαίνονται οι συναρτήσεις που δημιουργήθηκαν από τα αντικείμενα που υπάρχουν στο πρόγραμμα.

```
private void exetActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void prosthikiActionPerformed(java.awt.event.ActionEvent evt) {...11 lines }
private void probolhActionPerformed(java.awt.event.ActionEvent evt) {...62 lines }
private void apoth_exetActionPerformed(java.awt.event.ActionEvent evt) {...75 lines }
private void saveActionPerformed(java.awt.event.ActionEvent evt) {...6 lines }
private void akActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void dieActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void thlActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void onActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void kataxwrhshActionPerformed(java.awt.event.ActionEvent evt) {...45 lines }
private void searchActionPerformed(java.awt.event.ActionEvent evt) {...42 lines }
private void idActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }
private void problActionPerformed(java.awt.event.ActionEvent evt) {...47 lines }
private void parActionPerformed(java.awt.event.ActionEvent evt) {...7 lines }
private void paralabipActionPerformed(java.awt.event.ActionEvent evt) {...46 lines }
private void apostolipActionPerformed(java.awt.event.ActionEvent evt) {...63 lines }
```

Εικόνα 34-συναρτήσεις αντικειμένων

Αναλόγως τις δυνατότητες που θέλουμε να έχει το κάθε αντικείμενο, συνδιάζουμε τις συναρτήσεις που προαναφέραμε για να πάρουμε τα επιθυμητά αποτελέσματα.

```
public forma () {
    initComponents ();
    DoConnect ();
    Fillexet ();
    Date x=new Date ();
    hm.setDate (x);
    hm1.setDate (x);
}
```

Εικόνα 35-forma

Η forma είναι η βασική συνάρτηση του προγράμματος και εκτελείται με το που ανοίξουμε το πρόγραμμα.



## 6. Αποτελέσματα

Με την υλοποίηση αυτής της πτυχιακής εργασίας μας δώθηκε η ευκαιρία να αποκομίσουμε γνώσεις αλλά και εμπειρία πάνω στη Java στις βάσεις δεδομένων αλλά και στον συνδιασμό τους. Πραγματοποιήσαμε έρευνα γύρω απο το αντικείμενο της Java, των βάσεων δεδομένων και ότι άλλο αυτό συμπεριλαμβάνει. Εργαστήκαμε πάνω στο Netbeans για να φτιάξουμε το πρόγραμμα μας. Αφιερώσαμε αρκετό χρόνο για να βρούμε λύσεις στο πως θα είναι το προγράμμας πιο λειτουργικό, έτσι ώστε να καλύβει όλες τις ανάγκες του χρήστη, αλλά ταυτόχρονα να είναι και απλό στη χρήση του. Επίσης αποκομίσαμε γνώσεις πάνω στις γλώσσες προγραμματισμού που χρησιμοποιεί το Netbeans, όπως είναι η Java και η sql.

### 6.1 Συμπεράσματα

Το Netbeans είναι ένα ευρέως διαδεδομένο λογισμικό και πάρα πολλά προγράμματα μπορούν να δημιουργηθούν σε αυτό. Οι δυνατότητες του είναι σχεδόν απεριόριστες και χρησιμοποιείται για την κατασκευή προγραμμάτων και άλλων θεματικών περιεχομένων, καθώς εκτός απο τις γλώσσες προγραμματισμού java και sql που χρησιμοποιήσαμε εμείς, υποστηρίζει κι άλλες. Πολύ βασικό πλεονέκτημα του Netbeans είναι η μεταφερσιμότητα λόγω του ότι είναι γραμμένο σε Java και ότι έχει ευκολία στη συγγραφή κώδικα.

Το πρόγραμμα που φτιάξαμε είναι απαραίτητο για ένα μικροβιολογικό εργαστήριο, γιατί με αυτό μπορούμε να αποθηκεύσουμε, να οργανώσουμε και να διαχειριστούμε όλες τις απαραίτητες πληροφορίες του. Επίσης το οτι είναι απλό και εύκολο στη χρήση του μας εξοικονομεί χρόνο και κόπο.

### 6.2 Μελλοντική εργασία και επεκτάσεις

Το πρόγραμμα μας έχει δυνατότητες για μελλοντικές επεκτάσεις καθώς, εάν οι απαιτήσεις ενός μικροβιολογικού εργαστηρίου αυξηθούν μπορούμε εύκολα να το προσαρμόσουμε σε αυτές και να είναι για αρκετό καιρό ακόμα λειτουργικό.



## Βιβλιογραφία

- 1. "Java Developer's Reference", Mike Cohn, Bryan Morgan, Michael Morrison, Michael t. Nygard, Dan Sashi, Tom Trinko, Εκδόσεις SamNet, 1996, ISBN 1-57521-1297.
- 2. "Professional Java Fundamentals", Shy Cohen, Tom Mitchell, Adres Gonzalez, Lary Rodrigues, Kerry Hammil, Εκδόσεις WROX PRESS Ltd, 1996, ISBN 1-861000-38-3.
- 3. "Exploring Java", Patrick Niemeyer & Joshua, Εκδόσεις O' REILLY, 1997, ISBN 1-56592-271-9.
- 4. "Εγχειρίδιο της Java", Laura Lemay, Charles L Perkins, Michael Morrison, Εκδόσεις Μ.Γκιούρδας, 1996, ISBN 960-512-083-6.

### Ηλεκτρονικές διευθύνσεις:

1. <http://www.halcyon.com/mclain/java.htm>
2. <http://www.javacoffeebreak.com/tutorials/index.html>
3. <http://journals.ecs.soton.ac.uk/java/tutorial/>
4. <http://www.theparticle.com/>
5. <http://www.oreilly.com/>
6. <http://www.devx.com/>
7. <http://java.sun.com/>
8. <http://softwaredev.earthweb.com/java>
9. <http://www.dgp.toronto.edu/~mjmcguff/learn/java>
10. <http://www.javaboutique.internet.com/>
11. <http://www.javapplets.com/>
12. <http://www.freewarejava.com/>
13. <http://www.javapowered.com/>
14. <http://www.zdnet.com/devhead/resources/scriptlibrary/applets>
15. <http://www.coffeecup.com/java/>
16. <http://www.apl.jhu.edu/~hall/java/CGI-with-Java.html>
17. <http://www.epm.ornl.gov/java/>
18. <http://www.cs.washington.edu/>
19. <http://www.telecom.ntua.gr/HTML.Tutorials/java>
20. <http://www.cs.aston.ac.uk/~beaumoaj>
21. <http://www.essi.fr/~sander/courses/Java/Examples>