Μηχανική Λογισμικού

Εργασία Εξαμήνου

Ονόματα σπουδαστών Α.Μ.Σ. Νικολάου Χρήστος 40819 Ρακιτζής Γιώργος 40194 Μανώλης Δημήτρης 40338



2014

ΤΕΙ ΠΕΙΡΑΙΑ ΗΥΣ

24/1/2014

1. **Εισαγωγή,**

Σκοπός της εργασίας μας είναι να υλοποιήσουμε τον κώδικα για ένα λογισμικό για online κρατήσεις ενοικιαζόμενων δωματίων καθώς και η ανάλυση των λειτουργιών και των απαιτήσεων με αντίστοιχα διαγράμματα.

Έχουμε κάνει σύνδεση με βάση δεδομένων mySQL και λειτουργίες που χρησιμοποιούν GUI.

Οι λειτουργίες που υλοποιήσαμε για τον πελάτη είναι η δυνατότητα αναζήτησης με κριτήριο την ημερομηνία, δυνατότητα login για να κάνει κράτηση . Ο υπάλληλος έχει τις ίδιες δυνατότητες αλλά επιπλέον έχει τη δυνατότητα να προσθέσει και να αφαιρέσει κάποιο δωμάτιο.

Τέλος ο διαχειριστής μπορεί να βλέπει στατιστικά κάνοντας login. Τα στατιστικά που μπορεί να δει είναι: κρατήσεις ανά μήνα, συνολικές κρατήσεις, έσοδα του μήνα, συνολικά έσοδα και τους χρήστες.

Για την σύνδεση της εφαρμογής με τη βάση δεδομένων είναι απαραίτητο το Username=askisi10 και το Password=askisi10.

Σκοπός του παρακάτω κειμένου είναι η περιγραφή των λειτουργιών του λογισμικού για online κρατήσεις.

Παρακάτω καλύπτονται όλες οι απαιτήσεις που πρέπει να ξέρει ο σχεδιαστής ώστε να το λογισμικό που θα δημιουργηθεί να είναι το καλύτερο αποτέλεσμα.

1. **Γενική Περιγραφή**
   1. Λειτουργίες εφαρμογής

Οι λειτουργίες που προσφέρει η εφαρμογή μας είναι οι εξής :

* Εγγραφή, διαγραφή και επεξεργασία στοιχείων κρατήσεων.
* Προσθήκη και διαγραφή δωματίων.
* Αποθήκευση, επεξεργασία και διαγραφή πελατών.
* Διαχείριση των κρατήσεων.
* Εμφάνιση στατιστικών στοιχείων για κρατήσεις, έσοδα και χρήστες.
  1. Κατηγορίες χρηστών

Οι κατηγορίες χρηστών της εφαρμογής είναι τρεις :

* Πελάτης
* Υπάλληλος
* Διαχειριστής
  1. Περιβάλλον λειτουργίας της εφαρμογής

Η εφαρμογή θα τρέχει σε λειτουργικό σύστημα Windows XP σε υπολογιστές με Pentium 2GHz και 4 GB RAM.

* 1. Περιορισμοί στη σχεδίαση και την υλοποίηση

Για την ανάπτυξη του λογισμικού χρησιμοποιήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού Java και για την αποθήκευση των δεδομένων μια βάση δεδομένων mySQL.

* 1. Τεκμηρίωση για τον χρήστη

Όταν ολοκληρωθεί η ανάπτυξη του λογισμικού θα διεξαχθούν σεμινάρια στους υπάλληλους του ξενοδοχείου για τη σωστή λειτουργία του λογισμικού καθώς θα δοθούν και εγχειρίδια χρήσης. Επίσης

* 1. Προϋποθέσεις - Εξαρτήσεις

Το λογισμικό θα πρέπει να είναι διαθέσιμο στους χρήστες καθώς και να υπάρχει σύνδεση στο Internet με σύνδεση ADSL 2MBps, αλλά στο ξενοδοχείο θα μπορεί να λειτούργει και στο εσωτερικό δίκτυο για την εκχώρηση κράτησης από τους υπαλλήλους με LAN 1GBps.

1. **Απαιτήσεις για τις εξωτερικές διεπαφές**
   1. Διεπαφή χρήστη

Η διεπαφή του χρήστη με το λογισμικό γίνεται μέσω του GUI. Ο χρήστης θα πρέπει να συμπληρώσει κάποια πεδία που θα στέλνονται με φόρμα δεδομένων για την εισαγωγή του και θα υπάρχουν πεδία για να επεξεργάζεται δεδομένα που αφορούν την κράτηση. Τέλος ο διαχειριστής θα μπορεί να βλέπει τα στατιστικά στοιχεία.

* 1. Διεπαφή λογισμικού

Το λογισμικό συνεργάζεται με μια βάση δεδομένων mySQL για την αποθήκευση των στοιχείων.

1. **Λειτουργίες συστήματος**
   1. Επεξεργασία στοιχείων πελατών
      1. Περιγραφή

Ο πελάτης θα μπορεί να κάνει αναζήτηση την ημερομηνία που θέλει για να δει αν υπάρχουν διαθέσιμα και να κάνει κράτηση αν βρει ελεύθερο δωμάτιο.

* + 1. Ανάλυση σε λειτουργικές απαιτήσεις

|  |  |
| --- | --- |
| Αναζήτηση πελάτη  Αποτέλεσμα αναζήτησης  Αποτέλεσμα αναζήτησης αποτυχία | Το σύστημα επιτρέπει την αναζήτηση των στοιχείων του ξενοδοχείου.  Το σύστημα εμφανίζει στην οθόνη τα στοιχεία των διαθέσιμων δωματίων.  Αν η αναζήτηση είναι αποτυχημένη το σύστημα εμφανίζει αντίστοιχο μήνυμα. |
| Καταχώρηση νέου πελάτη  Αποθήκευση στοιχείων πελάτη δωματίου και ημερομηνίας  Λάθος εισαγωγή στοιχείων | Το σύστημα ανοίγει μια κενή καρτέλα πελάτη.  Το σύστημα αποθηκεύει τα στοιχεία του νέου πελάτη στην βάση  Αν υπάρχουν πιθανά λάθη στην εισαγωγή των στοιχείων (πχ αριθμοί στο πεδίο του ονόματος) το σύστημα τον ειδοποιεί. |

* 1. Επεξεργασία στοιχείων κρατήσεων
     1. Περιγραφή

Ο υπάλληλος θα έχει τη δυνατότητα να δημιουργήσει νέες, να διαγράψει ή να επεξεργαστεί τα στοιχεία των κρατήσεων.

* + 1. Ανάλυση σε λειτουργικές απαιτήσεις

|  |  |
| --- | --- |
| Καταχώρηση νέας κράτησης  Αποθήκευση στοιχείων πελάτη δωματίου και ημερομηνίας  Λάθος εισαγωγή στοιχείων | Το σύστημα ανοίγει μια κενή καρτέλα κράτησης.  Το σύστημα αποθηκεύει τα στοιχεία της νέας κράτησης στην βάση δεδομένων  Αν ο υπάλληλος έχει πιθανά λάθη στην εισαγωγή των στοιχείων το σύστημα τον ειδοποιεί. |
| Επεξεργασία στοιχείων  Εισαγωγή λανθασμένου στοιχείου | Ο υπάλληλος ζητάει να επεξεργαστεί τα στοιχεία μιας κράτησης. Το σύστημα αποθηκεύει τις αλλαγές στην βάση δεδομένων  Αν κάποιο από τα στοιχεία είναι λανθασμένο το σύστημα ειδοποιεί τον υπάλληλο με κατάλληλο μήνυμα. |

* 1. Συλλογή στατιστικών στοιχείων
     1. Περιγραφή

Ο διαχειριστής έχει τη δυνατότητα να δει στατιστικά στοιχεία όπως το σύνολο των κρατήσεων, τις κρατήσεις του μήνα, τα συνολικά έσοδα, τα έσοδα του μήνα και τους χρήστες.

* + 1. Ανάλυση σε λειτουργικές απαιτήσεις

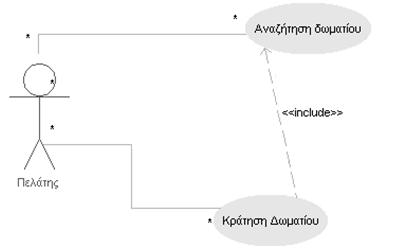
|  |  |
| --- | --- |
| Επιλογή εμφάνισης στατιστικών στοιχείων  Αποτέλεσμα αναζήτησης | Ο Διαχειριστής έχει την επιλογή για εμφάνιση διάφορων στατιστικών στοιχείων. Αυτά είναι: σύνολο των κρατήσεων, τις κρατήσεις του μήνα, τα συνολικά έσοδα, τα έσοδα του μήνα και τους χρήστες.  Το σύστημα εμφανίζει στην οθόνη τα ζητούμενα στατιστικά στοιχεία. |

1. **Μη λειτουργικές απαιτήσεις**
   1. Φυσική ασφάλεια

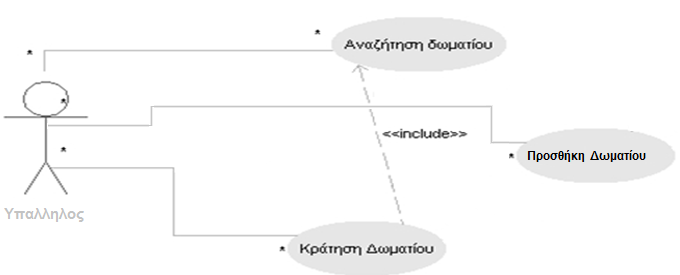
Για την αποστολή των στοιχείων του πελάτη γίνεται χρήση πρωτοκόλλου κρυπτογράφησης (SSL).

* 1. Ασφάλεια πληροφορίας

Ο υπάλληλος και ο διαχειριστής για να χρησιμοποιήσουν το λογισμικό θα πρέπει να εισέλθουν τον κωδικό χρήστη.

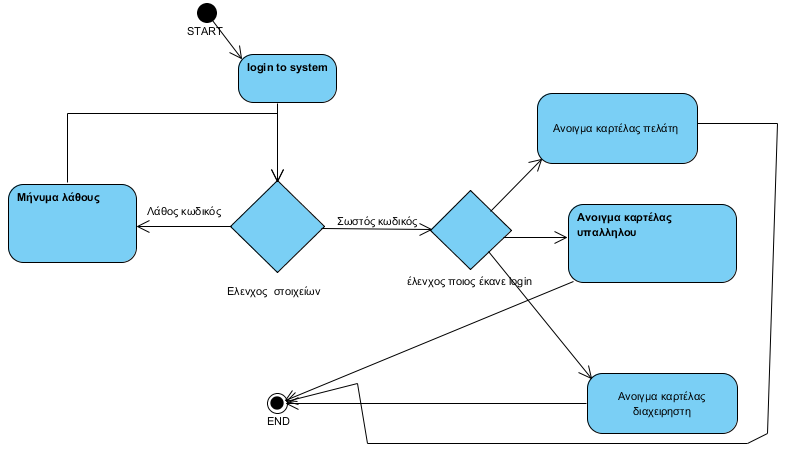
1. **Διαγράμματα περιπτώσεων χρήσης** 



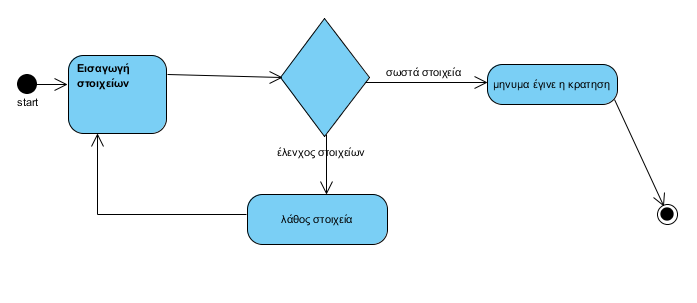


1. **Διαγράμματα UML**

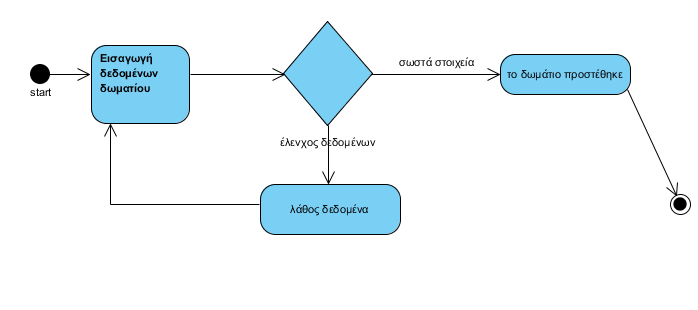
Uml login

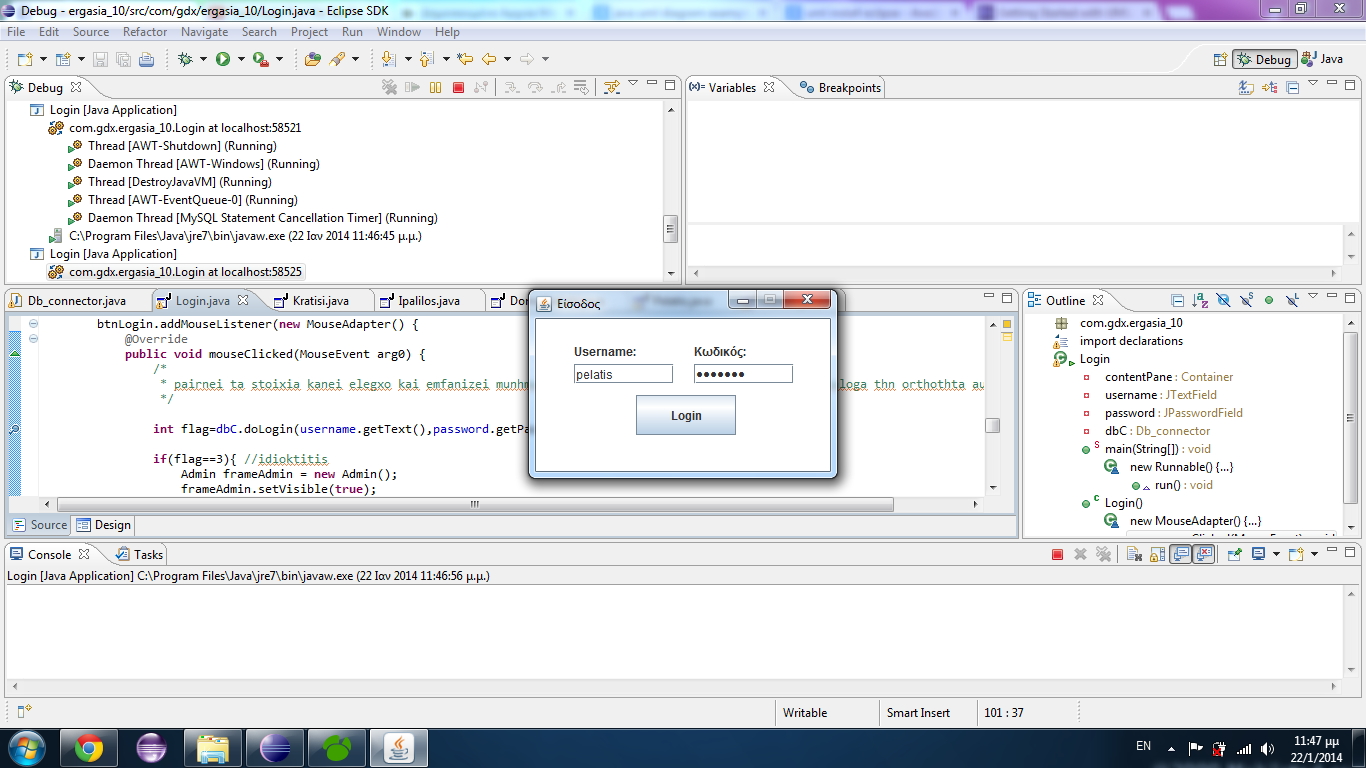


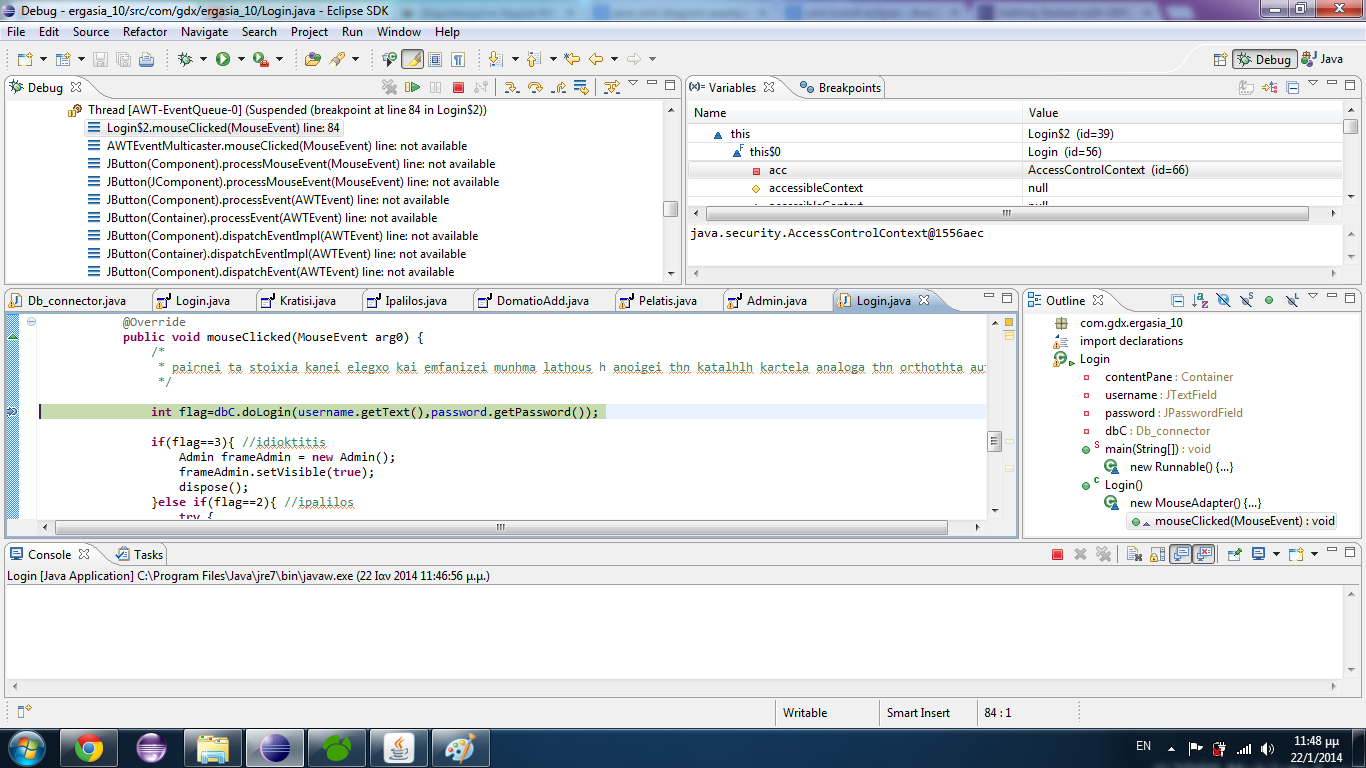
Uml kratisi

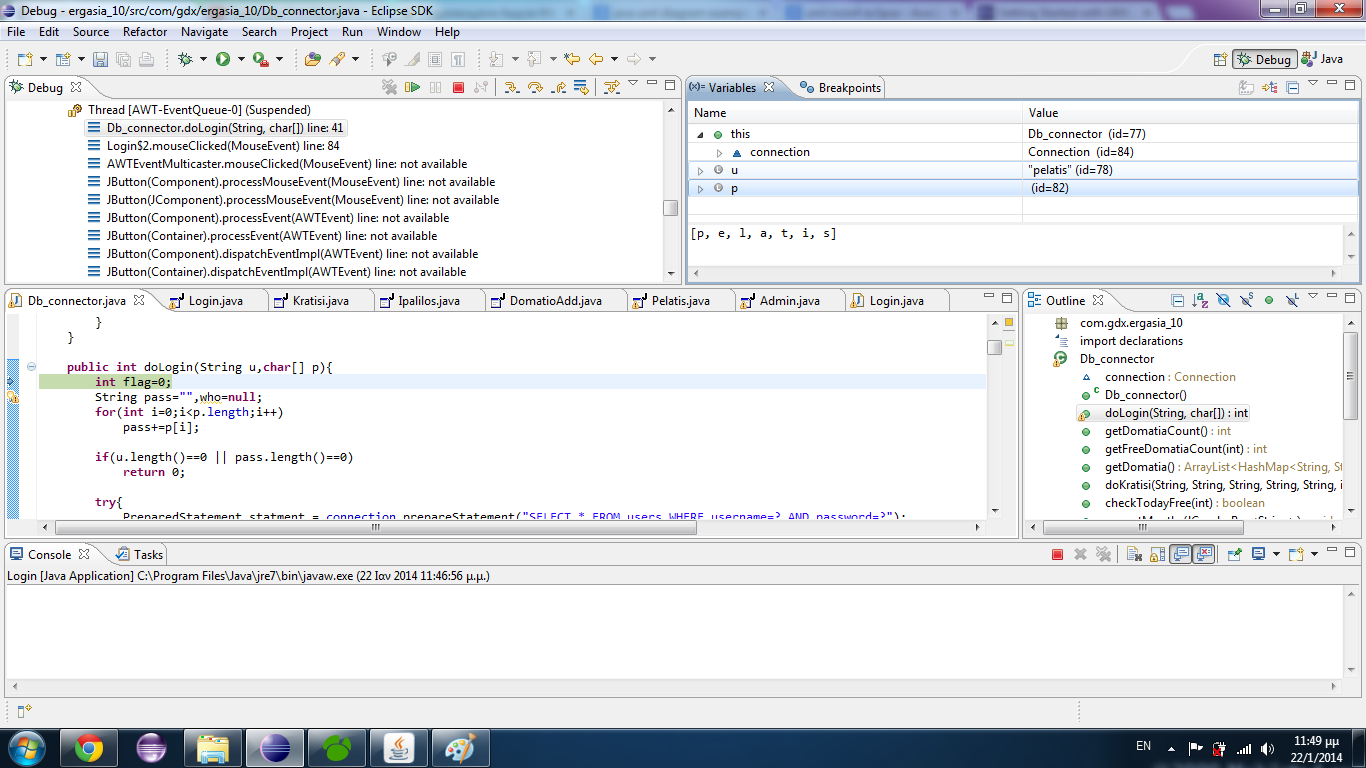


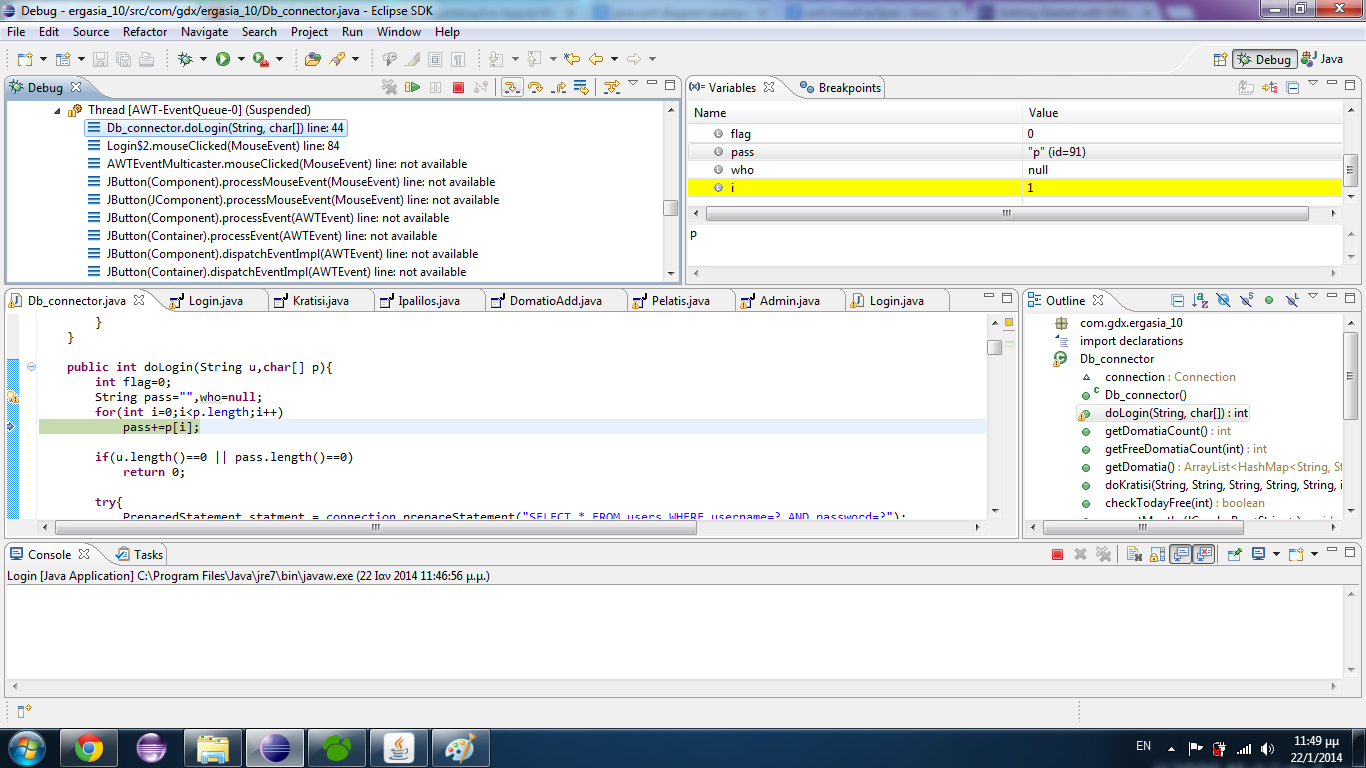
Uml domatioadd



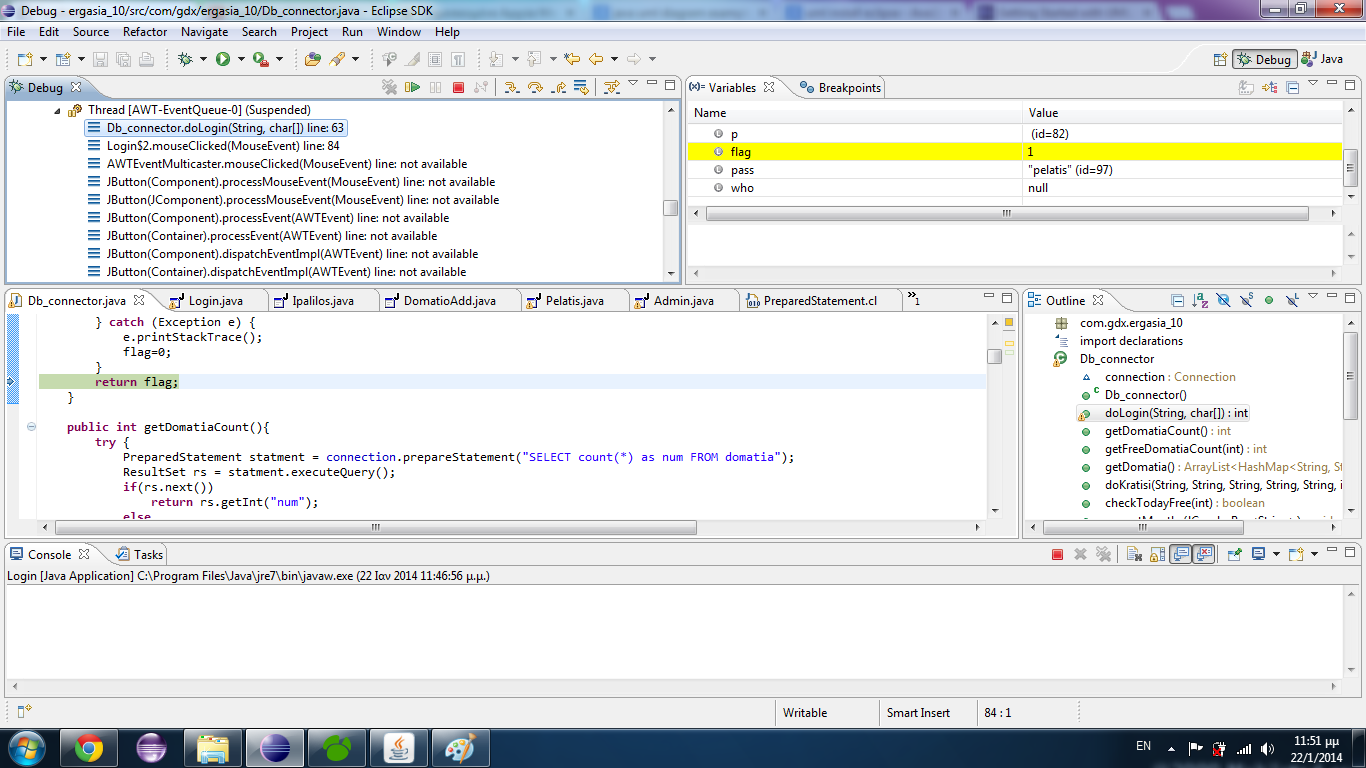
1. **Debugging**

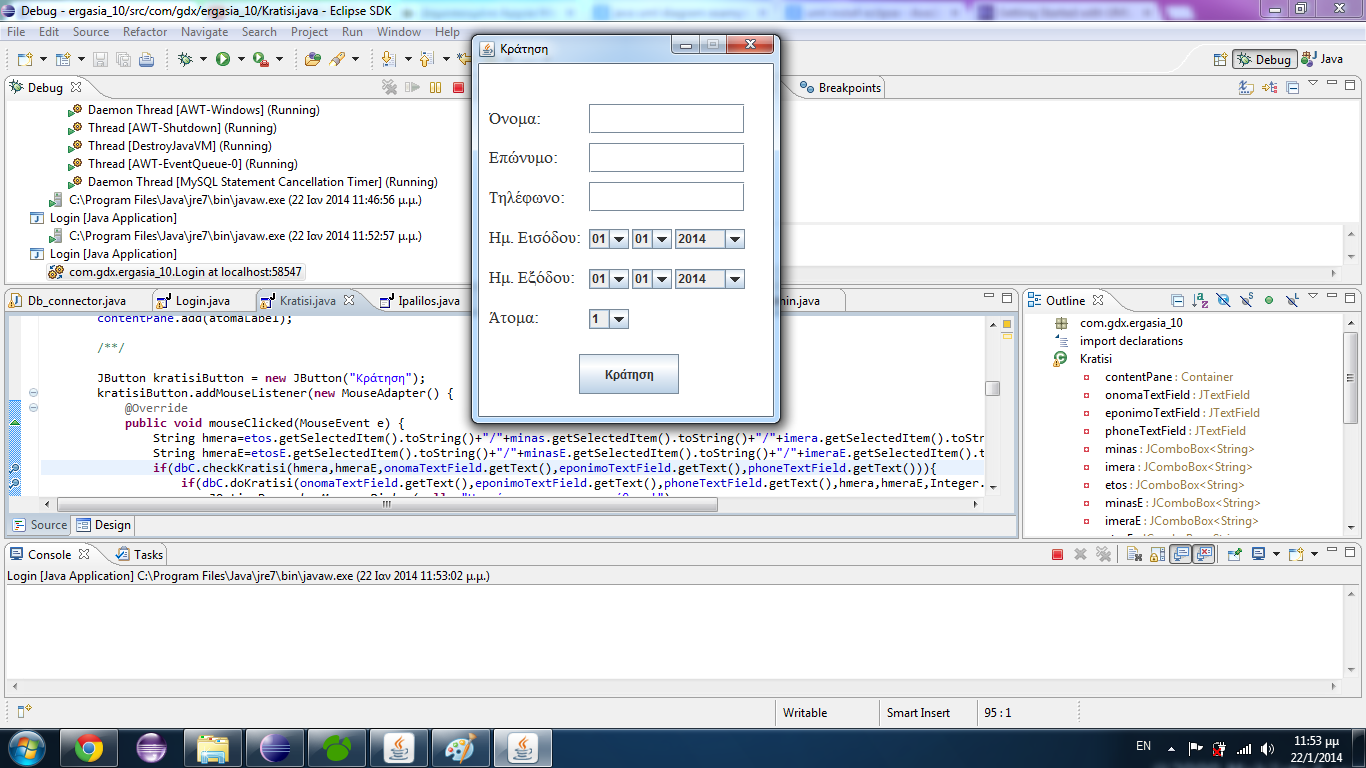
Παρατηρούμε το breakpoint στην function doLogin();  
Ο κώδικας με το που φτάνει εκεί σταματάει να τρέχει συνεχόμενα και περιμένει την δική μας κίνηση έτσι ώστε να ελέγξουμε την πορεία αυτού αλλά και των μεταβλητών.

Στην παρακάτω εικόνα βλέπουμε στο δεξί παάθυρο τις μεταβλητές u και p με τις τιμές τους.



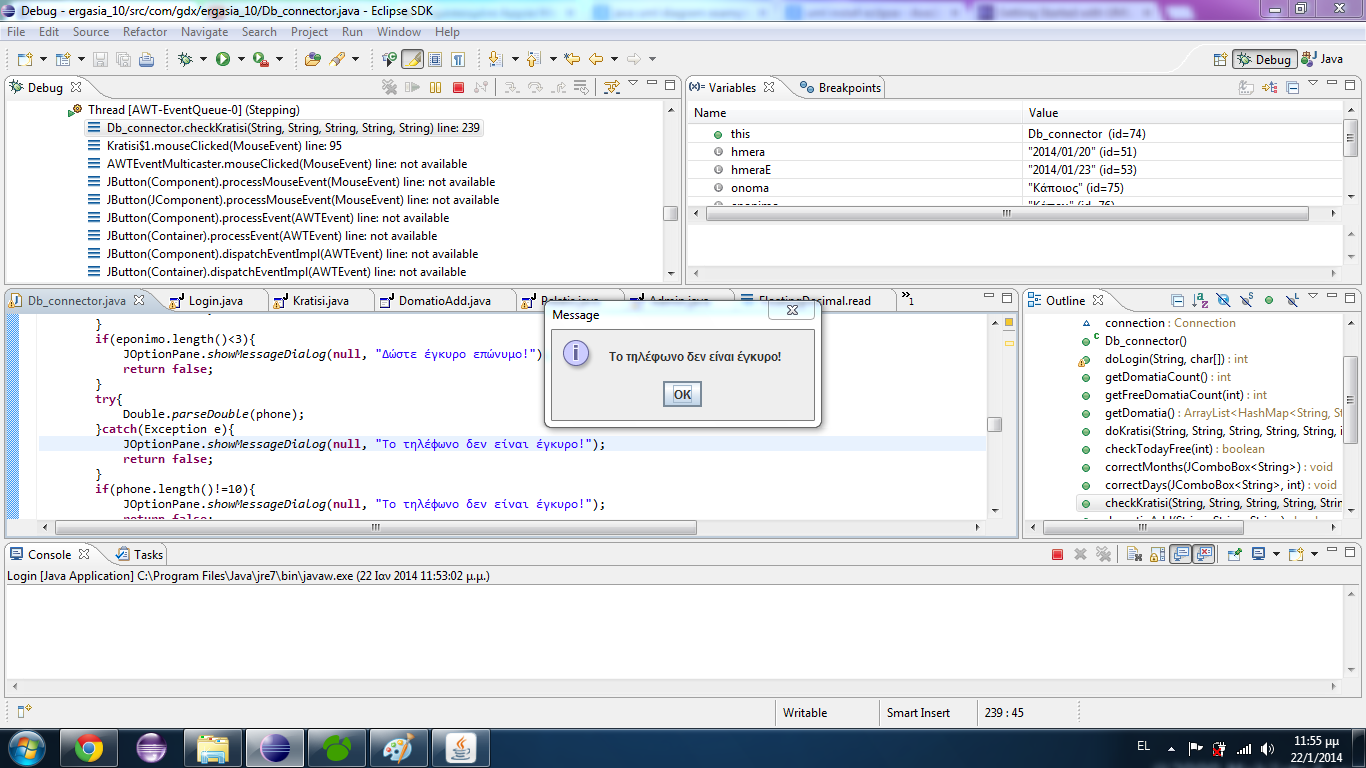
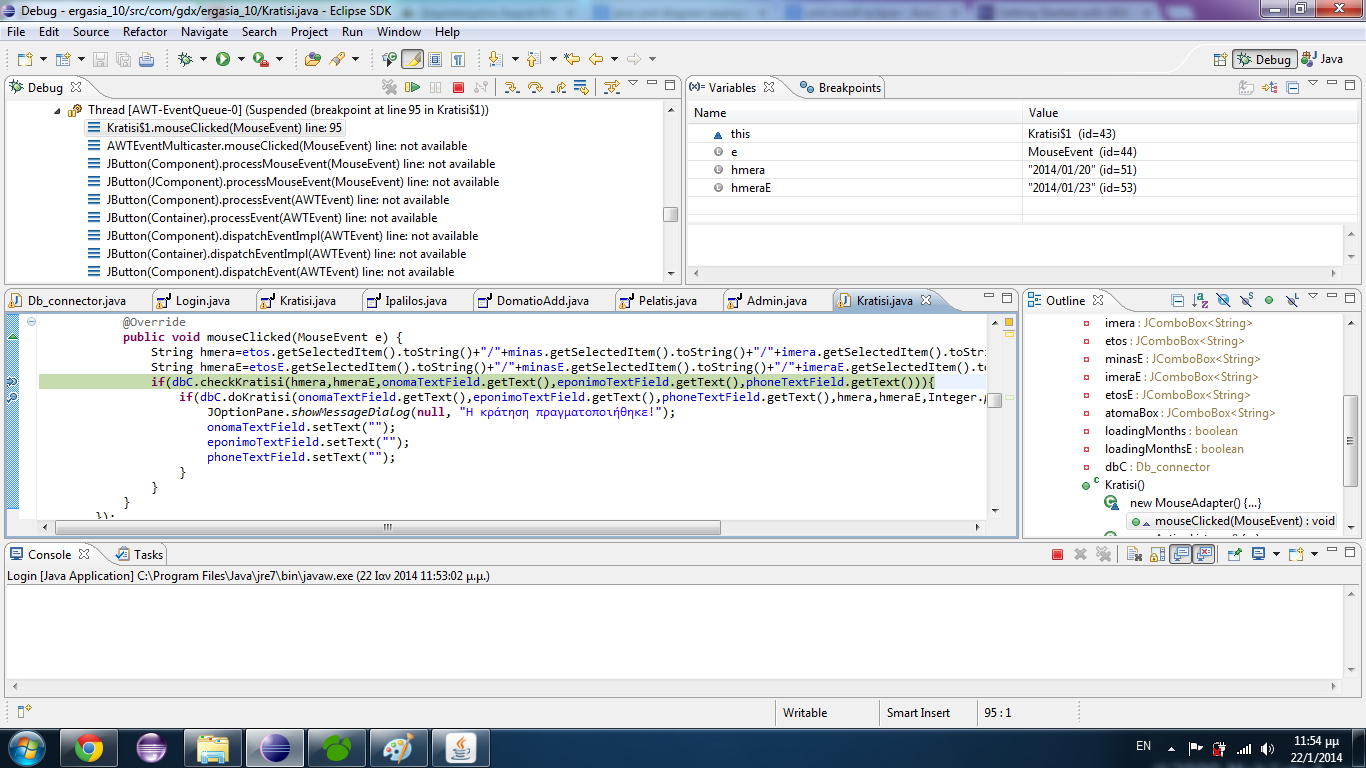
Παρατηρούμε την πορεία που ακολουθεί ο κώδικας μέσα στην for και πως ακριβώς παίρνει τιμή η μεταβλητή String pass από την μεταβλητή char[] p με την βοήθεια της for.



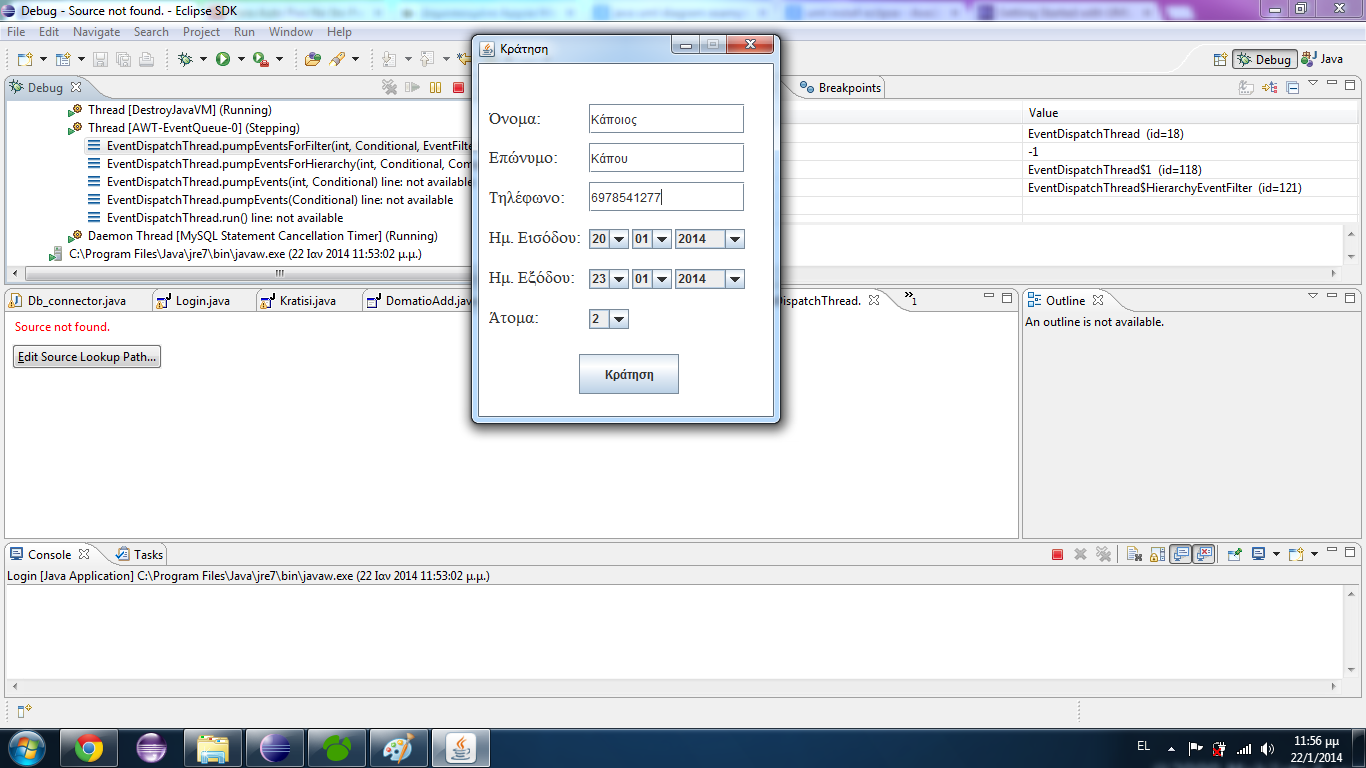
Στο τέλος της function βλέπουμε την τιμή της flag όταν αυτή έχει φτάσει στο σημείο να κάνει return.

Στην λειτουργία της κράτησης και συγκεκριμένα μέσα στην κλάση Kratisi.class

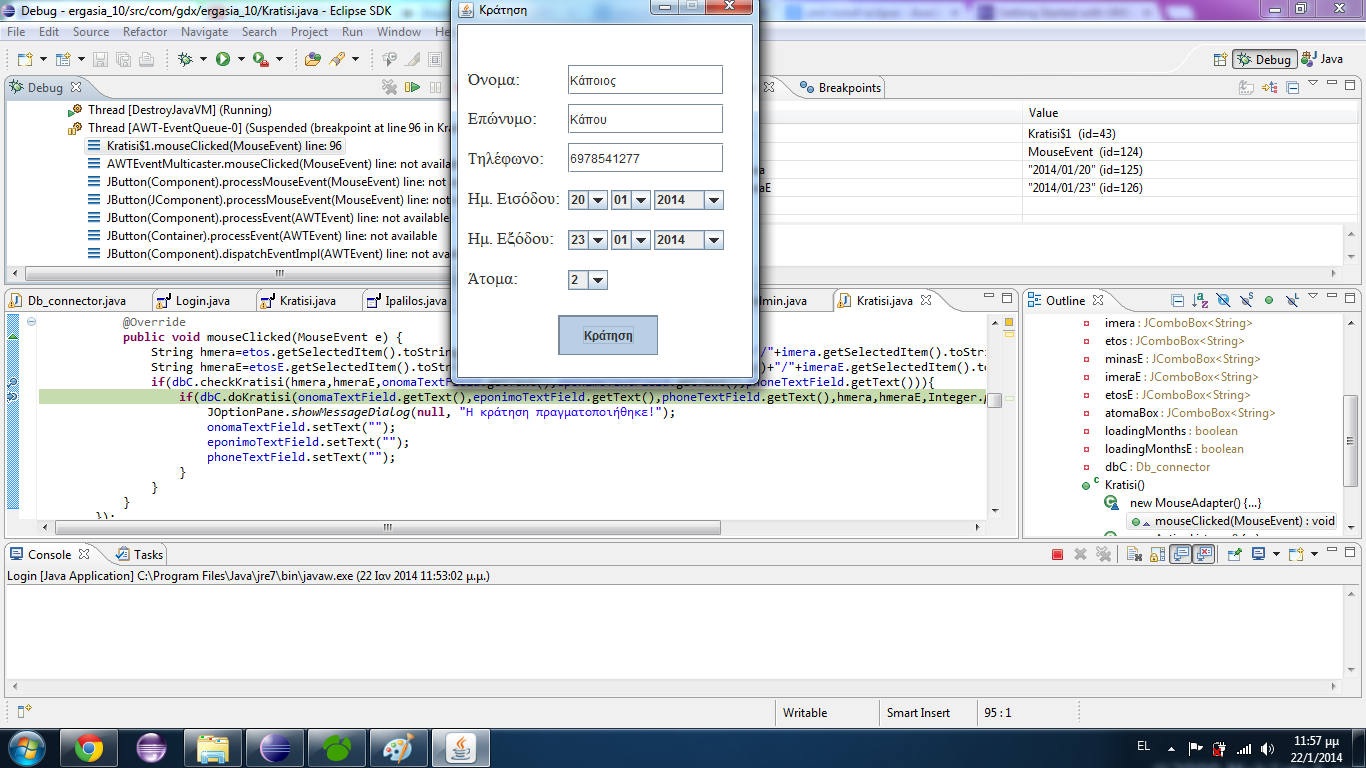
έχουμε βάλει breakpoints στις functions checkKratisi() και doKratisi() για να παρακολουθήσουμε την πορεία τους.



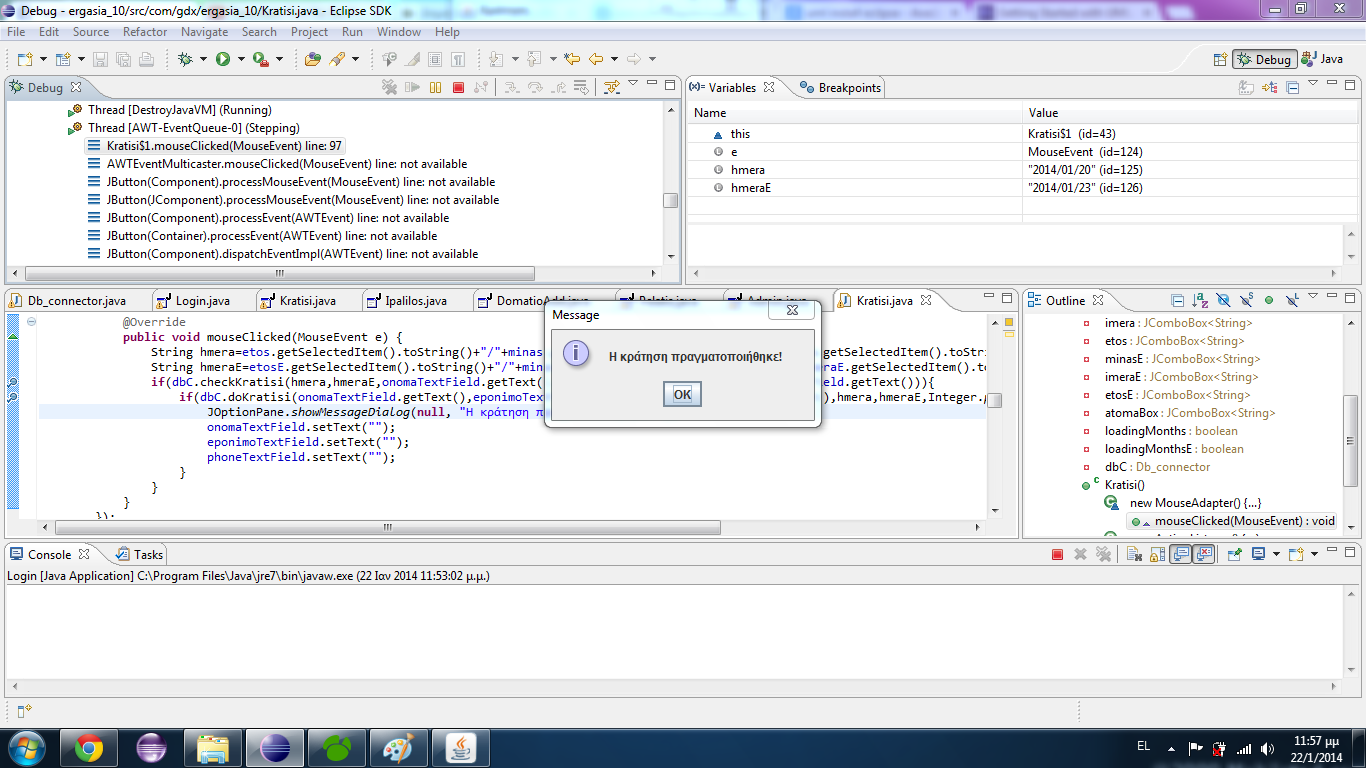
Όταν ο κώδικας φτάνει στον έλεγχο του τηλεφώνου διαπιστώνει λάθος στην μετατροπή του string σε Double(τρόπος για να δούμε αν όντως είναι αριθμός αυτός που έχει δωθεί) εμφανίζει μύνημα λάθους και περιμένει από τον χρήστη να πατήσει ok για να συνεχίσει.

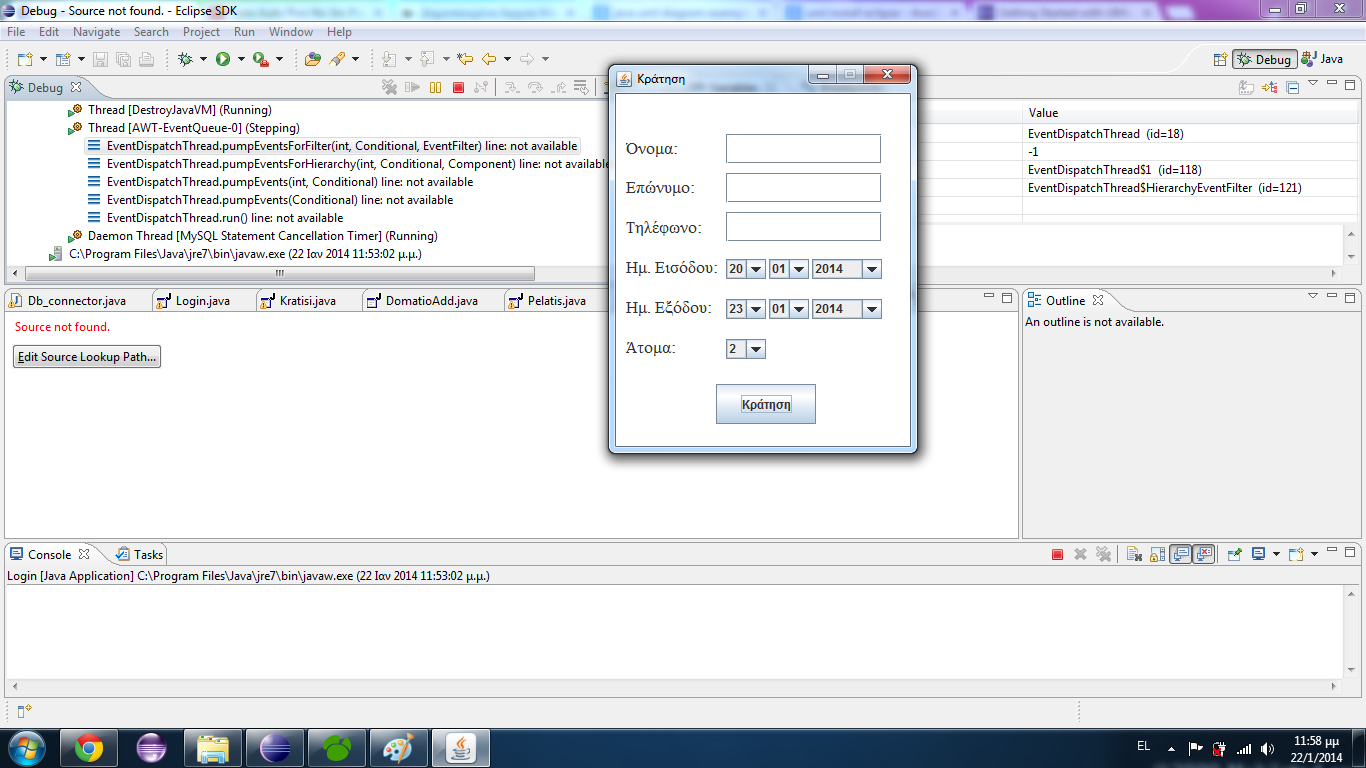


Ξαναδοκιμάζουμε με σωστό αριθμό



Και παρατηρούμε ότι η checkKratisi() επιστρέφει true και ο κώδικας προχωράει παρακάτω.



Γίνεται η κράτηση από την συνάρτηση doKratisi() και εμφανίζεται μήνυμα επιτυχίας! Και έπειτα σβήνονται τα πεδία της καρτέλας.

1. **Unit testing**
   1. Κώδικας

package com.gdx.ergasia\_10;

import static org.junit.Assert.\*;

import javax.swing.JComboBox;

import org.junit.Test;

public class TestClass {

Db\_connector dbC=new Db\_connector();

@Test

public void testGetDomatiaCount() {

assertEquals(8,dbC.getDomatiaCount());

}

@Test

public void testCheckTodayFree() {

assertFalse(dbC.checkTodayFree(1));

}

@Test

public void testCheckKratisi() {

assertTrue(dbC.checkKratisi("2014/02/14","2014/02/15","Giwrgos","Rakitzis","6973047764"));

}

@Test(timeout=2000)

public void openKratisi() {

Kratisi frame = new Kratisi();

frame.setVisible(true);

}

@Test(timeout=2000)

public void openIpalilos() {

Ipalilos frame = new Ipalilos();

frame.setVisible(true);

}

@Test(timeout=2000)

public void openPelatis() {

Pelatis frame = new Pelatis();

frame.setVisible(true);

}

@Test(timeout=2000)

public void openLogin() {

Login frame = new Login();

frame.setVisible(true);

}

@Test(timeout=2000)

public void openAdmin() {

Admin frame = new Admin();

frame.setVisible(true);

}

@Test(timeout=2000)

public void openDomatioAdd() {

DomatioAdd frame = new DomatioAdd();

frame.setVisible(true);

}

@Test(timeout=2000)

public void testSearchTime() {

dbC.searchDomatia("2014/01/28","2014/01/30");

}

@Test(timeout=10000)

public void testDoKratisiTime() {

dbC.doKratisi("Giwrgos","Rakitzis","6989524125","2014/02/12","2014/02/14",4);

}

@Test

public void testDoKratisi() {

assertFalse(dbC.doKratisi("Dimitris","Manolis","69895241aa","2014/02/12","2014/02/14",4));

assertFalse(dbC.doKratisi("Xristos","Nikolaou","6989524111","2014/a2/12","2014/02/14",4));

}

@Test

public void testDomatioAdd() {

assertFalse(dbC.domatioAdd("500","4","200"));

assertFalse(dbC.domatioAdd("5s0","s","200"));

assertFalse(dbC.domatioAdd("500","4","200"));

assertFalse(dbC.domatioAdd("500","4","20s0"));

}

@Test(timeout=2000)

public void searchClick() {

Ipalilos ip=new Ipalilos();

ip.searchClick();

}

@Test

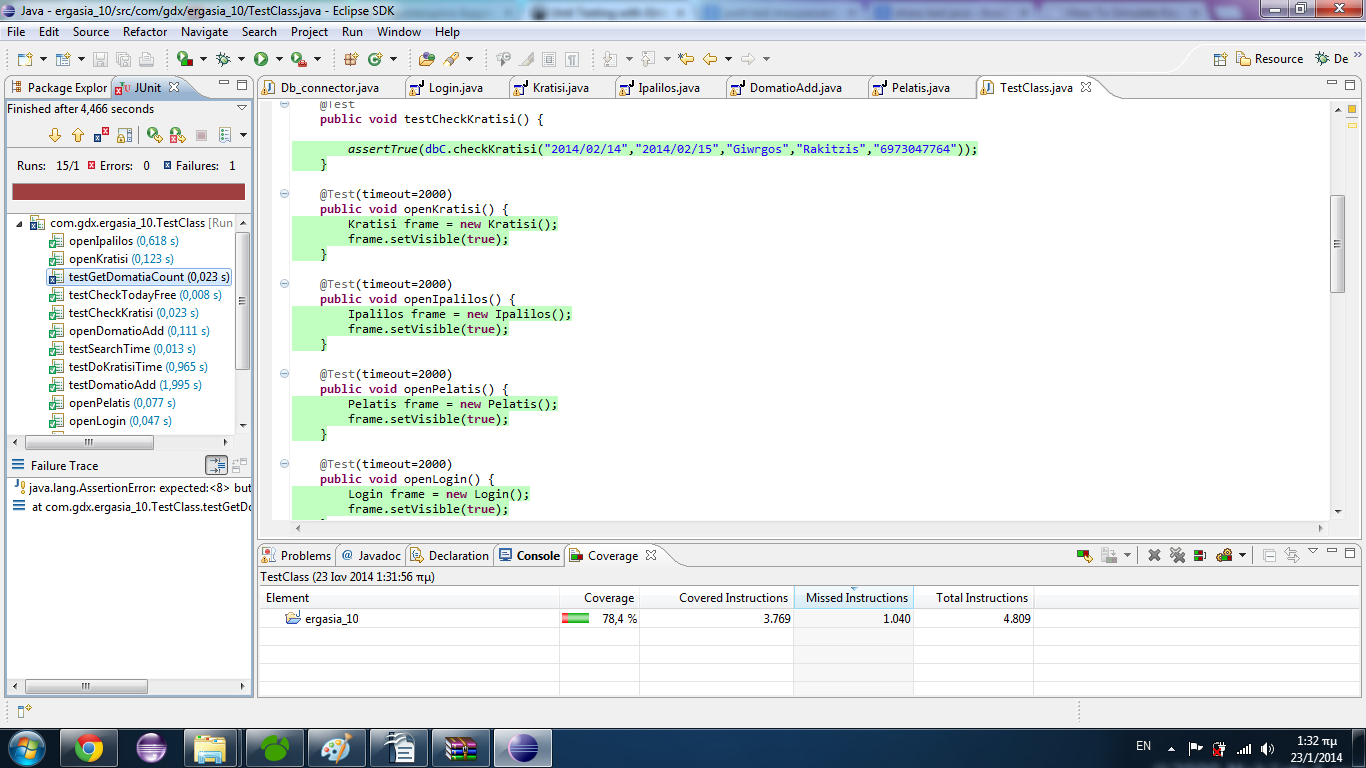
public void testLogin() {

char[] p={'a','b'};

assertEquals(0,dbC.doLogin(" ",p));

}

}

* 1. Screenshot