Τεκμηρίωση Προγράμματος

Εργαστήριο Μηχανικής Λογισμικού

Το παρακάτω έγγραφο εμπεριέχει την τεκμηρίωση του τελικού προγράμματος για το εργαστήριο Μηχανικής Λογισμικού.

Αγιοπετρίτης Χριστόδουλος,Δαλέζιος Ιωάννης,Σάβας Νικόλαος.

**ΠΕΡΙΕΧΟΜΕΝΑ**

**Γενικά 2**

**Κώδικας**

1. EnoikiazomenaDwmatia 5

*Συνοπτικά 8*

1. SearchDB 10

*Συνοπτικά 12*

1. Details 13

*Συνοπτικά 14*

1. ClientInfo 15

*Συνοπτικά 17*

**Βάση Δεδομένων 18**

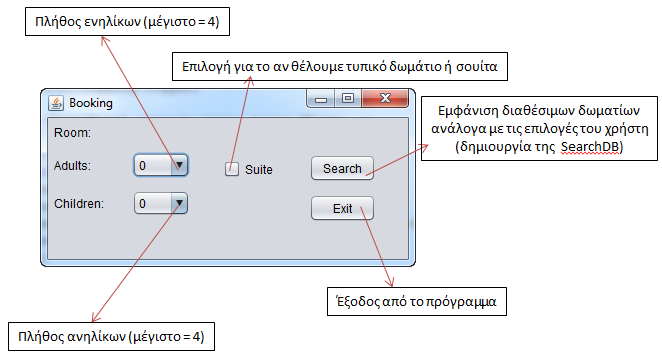
**Παρατηρήσεις – Γνωστά προβήματα 19**

**ΓΕΝΙΚΑ**

Το πρόγραμμα του ξενοδοχείου αποτελείται από τέσσερις κλάσεις, τις EnoikiazomenaDwmatia, SearchDB, ClientInfo και Details, κάθε μια από τις οποίες έχει το δικό της interface. Η βασική ιδέα είναι ο πελάτης να επιλέγει τι είδος δωματίου επιθυμεί και για πόσα άτομα και το πρόγραμμα να του εμφανίζει ποια είναι τα διαθέσιμα δωμάτια, ανάλογα με τις επιλογές του (τα δωμάτια και οι πληροφορίες για το καθένα είναι αποθηκευμένα σε μια βάση δεδομένων). Στη συνέχεια ο χρήστης θα μπορεί να κάνει κράτηση σε όποιο από τα διαθέσιμα δωμάτια επιθυμεί.

Η λειτουργία της κάθε κλάσης περιγράφεται παρακάτω:

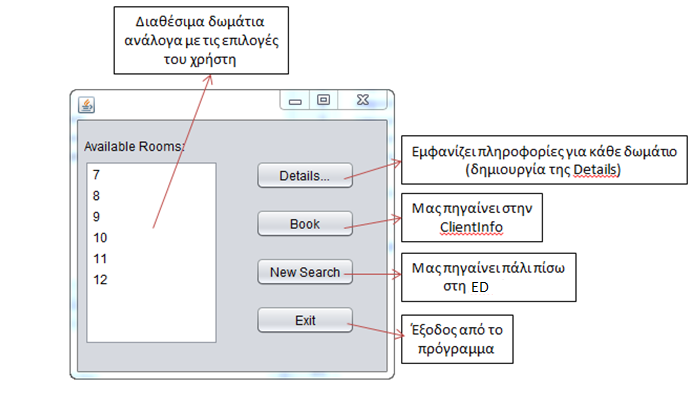
**EnoikiazomenaDwmatia:**



Παρατηρήσεις:

1. Σε περίπτωση που ο χρήστης αφήσει τις προεπιλεγμένες τιμές τότε το πρόγραμμα θα εμφανίσει όλα τα διαθέσιμα «τυπικά» δωμάτια. Αν επιλέξει «σουίτα» τότε θα του εμφανίσει τις διαθέσιμες σουίτες.
2. Σε περίπτωση που επιλέξει αριθμό ατόμων τότε το πρόγραμμα θα εμφανίσει όλα τα διαθέσιμα «τυπικά» δωμάτια τα οποία προορίζονται για άτομα περισσότερα ή ίσα του αριθμού ενηλίκων μαζί με τον αριθμό των ανηλίκων (για παράδειγμα αν Adults =2, Children=1 τότε το αποτέλεσμα θα είναι όλα τα δωμάτια που προορίζονται για άτομα>=3).

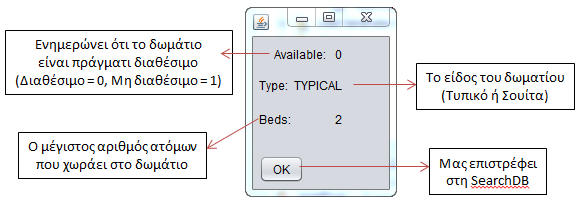
**SearchDB:**



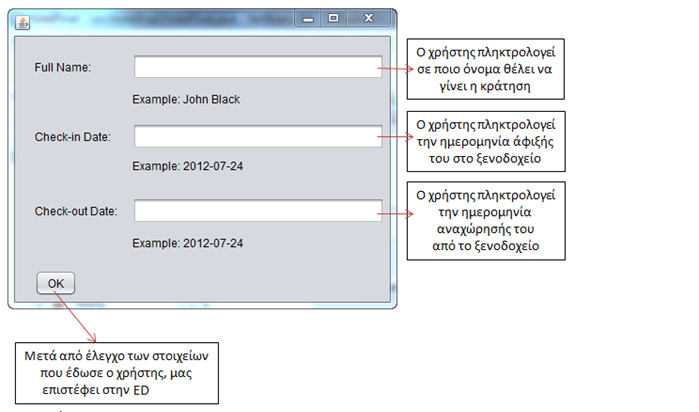
Παρατήρηση:

Τα κουμπιά Details και Book λειτουργούν μόνο αν ο χρήστης έχει επιλέξει ένα δωμάτιο από τη λίστα με τα διαθέσιμα δωμάτια αλλιώς εμφανίζεται το μήνυμα «Please select a room».

**Details:**



**ClientInfo:**



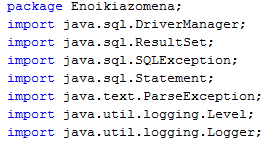
Παρατηρήσεις:

1. Στο πεδίο Full Name γίνεται έλεγχος για το αν ο χρήστης έδωσε όνομα.
2. Στα πεδία Check-in Date και Check-out Date γίνεται έλεγχος για το αν ο χρήστης έδωσε ημερομηνίες και για το αν οι ημερομηνίες είναι σωστές (είναι σύμφωνες με τα παραδείγματα).
3. Σε περίπτωση λάθους ο χρήστης πρέπει να ξαναεισάγει όνομα και ημερομηνίες.

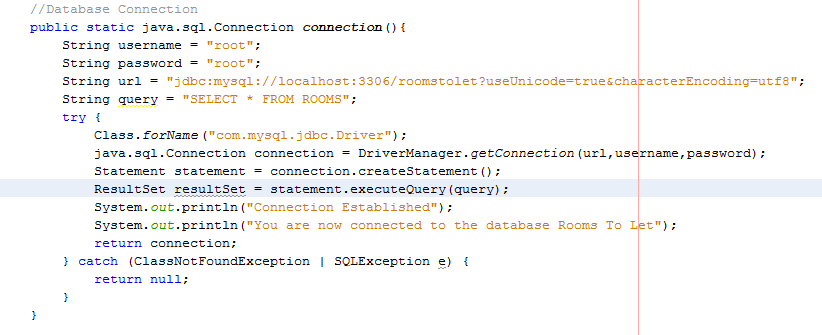
**ΚΩΔΙΚΑΣ**

* **EnoikiazomenaDwmatia**

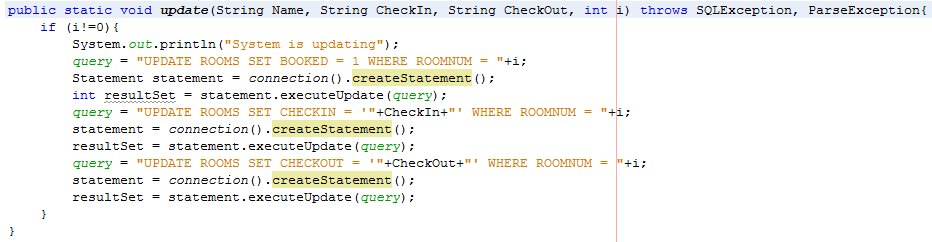
Η κλάση αυτή είναι η main του προγράμματος. Στην αρχή εισάγουμε τα απαραίτητα πακέτα που χρησιμοποιεί η κλάση.



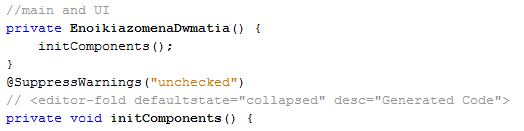
H μέθοδος connection() κάθε φορά που καλείτε κάνει σύνδεση με την βάση δεδομένων και μας επιστρέφει τη σύνδεση αυτή. Το query είναι δοκιμαστικό για να δοκιμάσουμε αν πράγματι γίνεται η σύνδεση.



Η μέθοδος update() κάθε φορά που καλείτε αλλάζει τα δεδομένα της βάσης δεδομένων. Παίρνει ως ορίσματα το όνομα και τις ημερομηνίες που θα εισάγει ο χρήστης στην κλάση ClientInfo. Το i είναι ο δείκτης που δείχνει για ποιο δωμάτιο θα γίνει αλλαγή στις ημερομηνίες του και στη διαθεσιμότητά του. Αν ο δείκτης είναι 0 σημαίνει ότι ο χρήστης δεν έκλεισε δωμάτιο και άρα δε χρειάζεται να γίνει καμιά αλλαγή στη βάση δεδομένων.



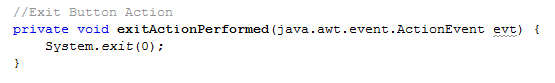
Οι δύο παρακάτω μέθοδοι δημιουργούν το interface της κλάσης.



Παρακάτω ακολουθεί ο κώδικας του κουμπιού Search. Αν ο χρήστης πατήσει το Search τότε θα δημιουργηθεί η κλάση SearchDB με το όνομα DB και σε αυτήν θα μεταφερθούν οι πληροφορίες της βάσης δεδομένων ανάλογα με τις επιλογές του χρήστη (τύπος δωματίου και αριθμός ατόμων). Τα δύο query πιάνουν όλες τις πιθανές επιλογές του χρήστη. Το πρώτο query = "SELECT \* FROM ROOMS WHERE ROOMTYPE = "+"'"+ roomt + "'"+"&&"+"BOOKED = 0 order by ROOMNUM"; τρέχει αν ο χρήστης δεν έχει κάνει καμία επιλογή ή έχει τσεκάρει το SUITE. Και ουσιαστικά εμφανίζει όλα τα άδεια δωμάτια που είναι ή TYPICAL ή SUITE. Το δεύτερο query = "SELECT \* FROM ROOMS WHERE ROOMTYPE = "+"'"+ roomt + "'" +"&&"+ "BEDSNUM >= " +all+"&& BOOKED = 0 order by ROOMNUM"; είναι όλες οι υπόλοιπες περιπτώσεις αν ο χρήστης έχει επιλέξει αριθμό ενηλίκων και αριθμό ανηλίκων τότε θα εμφανιστούν όλα τα δωμάτια (TYPICAL ή SUITE) που χωράνε αριθμό ατόμων >= αριθμό ενηλίκων + αριθμό ανηλίκων. Και στις δύο περιπτώσεις τα δομάτια ταξινομούνται ανάλογα με τον αριθμό τους.



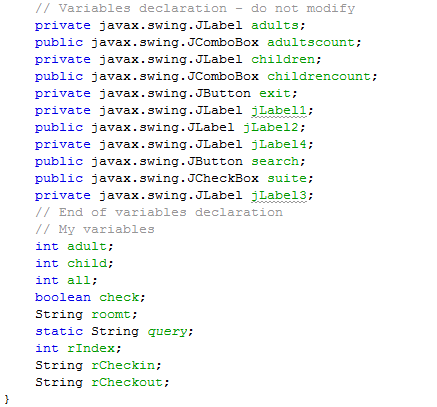
Ο κώδικας του κουμπιού Exit ακολουθεί παρακάτω. Όταν ο χρήστης πατήσει το κουμπί Exit τερματίζεται το πρόγραμμα.



Η μέθοδος main είναι η αρχή του προγράμματος. Εμφανίζει το interface της κλάσης EnoikiazomenaDwmatia στην οθόνη και περιέχει τη διαχείριση εξαιρέσεων που μπορεί να προκύψουν.



Τέλος έχουμε τις μεταβλητές που χρησιμοποιεί το πρόγραμμα.



**ΣΥΝΟΠΤΙΚΑ**

package Enoikiazomena;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.ParseException;

import java.util.logging.Level;

import java.util.logging.Logger;

public class EnoikiazomenaDwmatia extends javax.swing.JFrame {

//Database Connection

public static java.sql.Connection connection(){

String username = "root";

String password = "root";

String url = "jdbc:mysql://localhost:3306/roomstolet?useUnicode=true&characterEncoding=utf8";

String query = "SELECT \* FROM ROOMS";

try {

Class.forName("com.mysql.jdbc.Driver");

java.sql.Connection connection = DriverManager.getConnection(url,username,password);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(query);

System.out.println("Connection Established");

System.out.println("You are now connected to the database Rooms To Let");

return connection;

} catch (ClassNotFoundException | SQLException e) {

return null;

}

}

public static void update(String Name, String CheckIn, String CheckOut, int i) throws SQLException, ParseException{

if (i!=0){

System.out.println("System is updating");

query = "UPDATE ROOMS SET BOOKED = 1 WHERE ROOMNUM = "+i;

Statement statement = connection().createStatement();

int resultSet = statement.executeUpdate(query);

query = "UPDATE ROOMS SET CHECKIN = '"+CheckIn+"' WHERE ROOMNUM = "+i;

statement = connection().createStatement();

resultSet = statement.executeUpdate(query);

query = "UPDATE ROOMS SET CHECKOUT = '"+CheckOut+"' WHERE ROOMNUM = "+i;

statement = connection().createStatement();

resultSet = statement.executeUpdate(query);

}

}

//main and UI

private EnoikiazomenaDwmatia() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

adults = new javax.swing.JLabel();

children = new javax.swing.JLabel();

adultscount = new javax.swing.JComboBox();

childrencount = new javax.swing.JComboBox();

suite = new javax.swing.JCheckBox();

search = new javax.swing.JButton();

exit = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Booking");

adults.setText("Adults:");

adults.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

children.setText("Children:");

children.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

adultscount.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "0", "1", "2", "3", "4" }));

adultscount.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

childrencount.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "0", "1", "2", "3", "4" }));

childrencount.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

suite.setText("Suite");

suite.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

search.setText("Search");

search.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

search.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

try {

searchActionPerformed(evt);

} catch (SQLException ex) {

Logger.getLogger(EnoikiazomenaDwmatia.class.getName()).log(Level.SEVERE, null, ex);

} catch (ParseException ex) {

Logger.getLogger(EnoikiazomenaDwmatia.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

exit.setText("Exit");

exit.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

exit.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

exitActionPerformed(evt);

}

});

jLabel4.setText("Room:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(children)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 28, Short.MAX\_VALUE)

.addComponent(childrencount, javax.swing.GroupLayout.PREFERRED\_SIZE, 58, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(adults)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(adultscount, javax.swing.GroupLayout.PREFERRED\_SIZE, 58, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(33, 33, 33)

.addComponent(suite)

.addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(exit, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(search))

.addGap(32, 32, 32))))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(adults)

.addComponent(adultscount, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(suite))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(children)

.addComponent(childrencount, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 30, Short.MAX\_VALUE)

.addComponent(search)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(exit)

.addGap(37, 37, 37))))

);

pack();

}// </editor-fold>

//Search Button Action

private void searchActionPerformed(java.awt.event.ActionEvent evt) throws SQLException, ParseException {

SearchDB DB = new SearchDB();

String a=(String)adultscount.getSelectedItem();

adult = Integer.parseInt(a);

String b=(String)childrencount.getSelectedItem();

child = Integer.parseInt(b);

all = adult+child;

roomt = "TYPICAL";

if (check = suite.isSelected()){

roomt = "SUITE";

}else{

roomt = "TYPICAL";

}

if (all==0){

query = "SELECT \* FROM ROOMS WHERE ROOMTYPE = "+"'"+ roomt + "'"+"&&"+"BOOKED = 0 order by ROOMNUM";

Statement statement = connection().createStatement();

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next())

{

String roomNum = resultSet.getString("ROOMNUM");

String booked = resultSet.getString("BOOKED");

String roomType = resultSet.getString("ROOMTYPE");

String bedsNum = resultSet.getString("BEDSNUM");

DB.getElements(roomNum);

DB.getDetails(booked, roomType, bedsNum);

}

}else{

query = "SELECT \* FROM ROOMS WHERE ROOMTYPE = "+"'"+ roomt + "'" +"&&"+ "BEDSNUM >= " +all+"&& BOOKED = 0 order by ROOMNUM";

Statement statement = connection().createStatement();

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next())

{

String roomNum = resultSet.getString("ROOMNUM");

String booked = resultSet.getString("BOOKED");

String roomType = resultSet.getString("ROOMTYPE");

String bedsNum = resultSet.getString("BEDSNUM");

DB.getElements(roomNum);

DB.getDetails(booked, roomType, bedsNum);

}

}

DB.setVisible(true);

}

//Exit Button Action

private void exitActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);

}

public static void main(String args[]) {

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(EnoikiazomenaDwmatia.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(EnoikiazomenaDwmatia.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(EnoikiazomenaDwmatia.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(EnoikiazomenaDwmatia.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//Connection.connection();

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new EnoikiazomenaDwmatia().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JLabel adults;

public javax.swing.JComboBox adultscount;

private javax.swing.JLabel children;

public javax.swing.JComboBox childrencount;

private javax.swing.JButton exit;

private javax.swing.JLabel jLabel1;

public javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel4;

public javax.swing.JButton search;

public javax.swing.JCheckBox suite;

private javax.swing.JLabel jLabel3;

// End of variables declaration

// My variables

int adult;

int child;

int all;

boolean check;

String roomt;

static String query;

int rIndex;

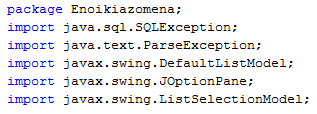
String rCheckin;

String rCheckout;

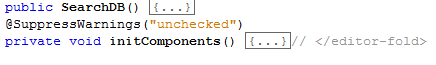
}

* **SearchDB**

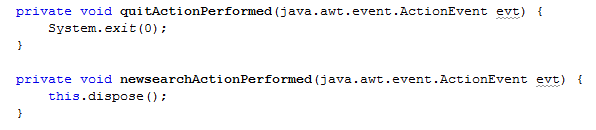
Στην αρχή εισάγουμε τα απαραίτητα πακέτα που χρησιμοποιεί η κλάση αυτή.



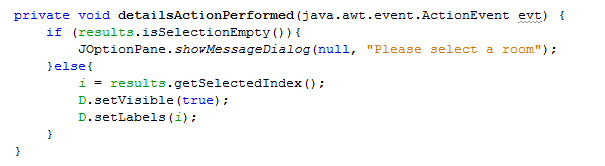
Οι δύο παρακάτω μέθοδοι δημιουργούν το interface της κλάσης.



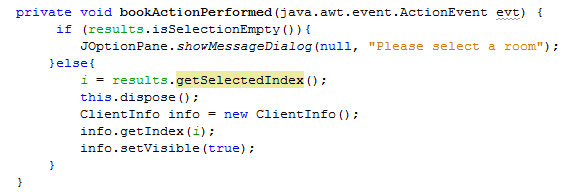
Παρακάτω είναι οι κώδικες των κουμπιών Exit και New Search αντίστοιχα. Το κουμπί Exit τερματίζει το πρόγραμμα ενώ το κουμπί New Search μας μεταφέρει πάλι πίσω στη EnoikiazomenaDwmatia «ξεκάνοντας» την κλάση DB που έχει δημιουργήσει η EnoikiazomenaDwmatia.



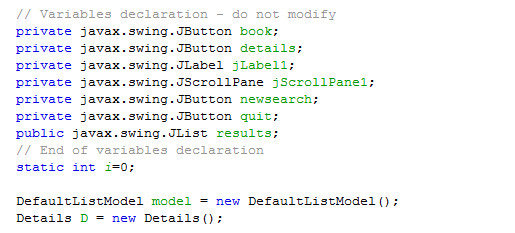
Παρακάτω ακολουθεί ο κώδικας του κουμπιού Details. Πατώντας το κουμπί αυτό εμφανίζεται η κλάση Details. Σε μέθοδο της κλάσης Details στέλνεται ο δείκτης του δωματίου που έχει επιλέξει ο χρήστης. Αν ο χρήστης δεν έχει επιλέξει κάποιο δωμάτιο τότε εμφανίζεται μήνυμα ότι πρέπει να επιλεγεί κάποιο δωμάτιο.



Παρακάτω ακολουθεί ο κώδικας του κουμπιού Book. Πατώντας το κουμπί αυτό δημιουργείτε η κλάση Book. Σε μέθοδο της κλάσης Book στέλνεται ο δείκτης του δωματίου που έχει επιλέξει ο χρήστης. Αν ο χρήστης δεν έχει επιλέξει κάποιο δωμάτιο τότε εμφανίζεται μήνυμα ότι πρέπει να επιλεγεί κάποιο δωμάτιο.



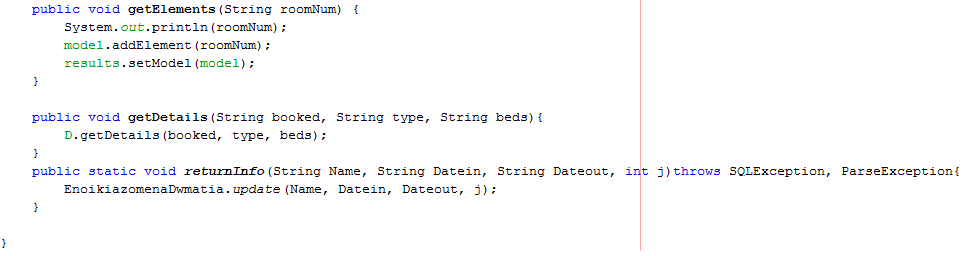
Παρακάτω ακολουθούν οι μεταβλητές της κλάσης και δημιουργείτε η κλάση Details με το όνομα D. *Σημείωση: η κλάση δημιουργείτε δεν εμφανίζεται στο χρήστη.*



Η μέθοδος getElements() εισάγει στη λίστα τα διαθέσιμα δωμάτια ανάλογα με τις επιλογές του χρήστη στην κλάση EnoikiazomenaDwmatia. Αντλεί τα δεδομένα από την κλάση EnoikiazomenaDwmatia (η οποία παίρνει τις πληροφορίες από τη βάση δεδομένων).

Η μέθοδος getDetails στέλνει σε μέθοδο της κλάσης Details (D) πληροφορίες σχετικά με το δωμάτιο που έχει επιλέξει ο χρήστης (αν είναι ελεύθερο, τι είδος είναι και πόσα άτομα χωράει). Οι πληροφορίες αντλούνται από την κλάση EnoikiazomenaDwmatia (η οποία παίρνει τις πληροφορίες από τη βάση δεδομένων).

Η μέθοδος returnInfo κάθε φορά που καλείται δίνει τις πληροφορίες που έχει δώσει ο χρήστης στην κλάση ClientInfo σε όποια μέθοδο την καλέσει. Στην ουσία γεφυρώνει την κλάση ClientInfo με την κλάση EnoikiazomenaDwmatia. Οι πληροφορίες που στέλνονται είναι το όνομα που έχει δώσει ο χρήστης, η ημερομηνία άφιξής του, η ημερομηνία αποχώρησης του και ο δείκτης του δωματίου που έχει επιλέξει.



**ΣΥΝΟΠΤΙΚΑ**

package Enoikiazomena;

import java.sql.SQLException;

import java.text.ParseException;

import javax.swing.DefaultListModel;

import javax.swing.JOptionPane;

import javax.swing.ListSelectionModel;

public class SearchDB extends javax.swing.JFrame {

public SearchDB() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

results = new javax.swing.JList();

jLabel1 = new javax.swing.JLabel();

details = new javax.swing.JButton();

book = new javax.swing.JButton();

newsearch = new javax.swing.JButton();

quit = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jScrollPane1.setViewportView(results);

results.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

jLabel1.setText("Available Rooms:");

details.setText("Details...");

details.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

detailsActionPerformed(evt);

}

});

book.setText("Book");

book.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

bookActionPerformed(evt);

}

});

newsearch.setText("New Search");

newsearch.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

newsearchActionPerformed(evt);

}

});

quit.setText("Exit");

quit.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

quitActionPerformed(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()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 128, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(book, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(details, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(newsearch, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(quit, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 258, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(30, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(16, 16, 16)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(details)

.addGap(18, 18, 18)

.addComponent(book)

.addGap(18, 18, 18)

.addComponent(newsearch)

.addGap(18, 18, 18)

.addComponent(quit))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 176, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(25, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void quitActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);

}

private void newsearchActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

}

private void detailsActionPerformed(java.awt.event.ActionEvent evt) {

if (results.isSelectionEmpty()){

JOptionPane.showMessageDialog(null, "Please select a room");

}else{

i = results.getSelectedIndex();

D.setVisible(true);

D.setLabels(i);

}

}

private void bookActionPerformed(java.awt.event.ActionEvent evt) {

if (results.isSelectionEmpty()){

JOptionPane.showMessageDialog(null, "Please select a room");

}else{

i = results.getSelectedIndex();

this.dispose();

ClientInfo info = new ClientInfo();

info.getIndex(i);

info.setVisible(true);

}

}

// Variables declaration - do not modify

private javax.swing.JButton book;

private javax.swing.JButton details;

private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton newsearch;

private javax.swing.JButton quit;

public javax.swing.JList results;

// End of variables declaration

static int i=0;

DefaultListModel model = new DefaultListModel();

Details D = new Details();

public void getElements(String roomNum) {

System.out.println(roomNum);

model.addElement(roomNum);

results.setModel(model);

}

public void getDetails(String booked, String type, String beds){

D.getDetails(booked, type, beds);

}

public static void returnInfo(String Name, String Datein, String Dateout, int j)throws SQLException, ParseException{

EnoikiazomenaDwmatia.update(Name, Datein, Dateout, j);

}

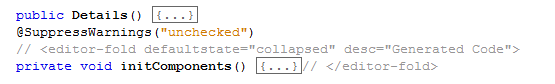
}

* **Details**

Στην αρχή εισάγουμε τα απαραίτητα πακέτα που χρησιμοποιεί η κλάση αυτή.

C:\Users\IMBAS\Desktop\details import.png

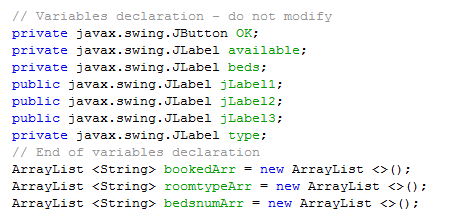
Οι δύο παρακάτω μέθοδοι δημιουργούν το interface της κλάσης.



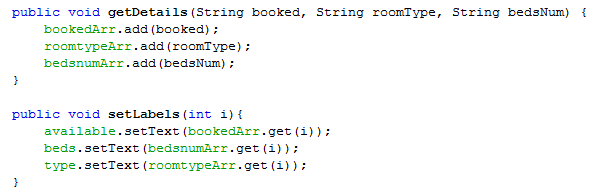
Παρακάτω ακολουθεί ο κώδικας του κουμπιού OK. Το κουμπί αυτό «ξεκάνει» την κλάση Details και μας στέλνει πίσω στη SearchDB .



Παρακάτω ακολουθούν οι μεταβλητές της κλάσης.



Η μέθοδος getDetails παίρνει από την SearchDB πληροφορίες σχετικά με το δωμάτιο που έχει επιλέξει ο χρήστης, ενώ η setLabels εμφανίζει τις πληροφορίες αυτές στα κατάλληλα Labels της κλάσης Details



**ΣΥΝΟΠΤΙΚΑ**

package Enoikiazomena;

import java.util.ArrayList;

public class Details extends javax.swing.JFrame {

public Details() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

available = new javax.swing.JLabel();

type = new javax.swing.JLabel();

beds = new javax.swing.JLabel();

OK = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("Available:");

jLabel2.setText("Beds:");

jLabel3.setText("Type:");

available.setText("jLabel4");

type.setText("jLabel4");

beds.setText("jLabel4");

OK.setText("OK");

OK.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

OKActionPerformed(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()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(beds))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(type))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(available)))

.addComponent(OK))

.addContainerGap(28, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(available))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(type))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(beds))

.addGap(28, 28, 28)

.addComponent(OK)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void OKActionPerformed(java.awt.event.ActionEvent evt) {

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JButton OK;

private javax.swing.JLabel available;

private javax.swing.JLabel beds;

public javax.swing.JLabel jLabel1;

public javax.swing.JLabel jLabel2;

public javax.swing.JLabel jLabel3;

private javax.swing.JLabel type;

// End of variables declaration

ArrayList <String> bookedArr = new ArrayList <>();

ArrayList <String> roomtypeArr = new ArrayList <>();

ArrayList <String> bedsnumArr = new ArrayList <>();

public void getDetails(String booked, String roomType, String bedsNum) {

bookedArr.add(booked);

roomtypeArr.add(roomType);

bedsnumArr.add(bedsNum);

}

public void setLabels(int i){

available.setText(bookedArr.get(i));

beds.setText(bedsnumArr.get(i));

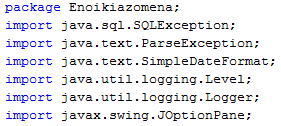
type.setText(roomtypeArr.get(i));

}

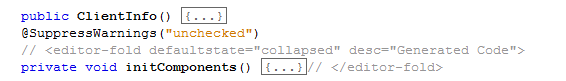
}

* **ClientInfo**

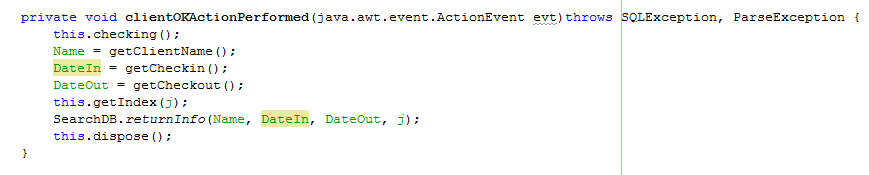
Στην αρχή εισάγουμε τα απαραίτητα πακέτα που χρησιμοποιεί η κλάση αυτή.



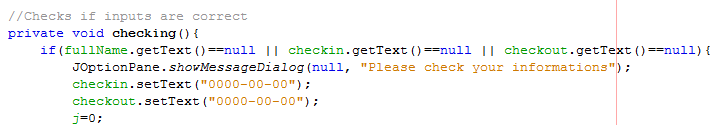
Οι δύο παρακάτω μέθοδοι δημιουργούν το interface της κλάσης.



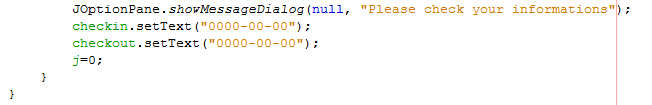
Παρακάτω ακολουθεί ο κώδικας του κουμπιού OK. Το κουμπί αυτό στέλνει της πληροφορίες που έχει εισάγει ο χρήστης (όνομα και ημερομηνίες) και το δείκτη του δωματίου που έχει επιλέξει στη μέθοδο returnInfo της κλάσης SearchDB. Από εκεί αυτές οι πληροφορίες καταλήγουν στη μέθοδο update της EnoikiazomenaDwmatia. Οι πληροφορίες στέλνονται αφού περάσουν από τη μέθοδο checking η οποία τις ελέγχει.



Η μέθοδος checking ελέγχει τις πληροφορίες που έδωσε ο χρήστης. Όσο αφορά το όνομα ελέγχεται αν ο χρήστης έδωσε όνομα και όσο αφορά της ημερομηνίες ελέγχετε επιπλέον αν ο χρήστης τις έδωσε σωστά (σύμφωνα με το παράδειγμα). Σε περίπτωση λάθους εμφανίζεται μήνυμα και ο χρήστης πρέπει να ξαναεισάγει όνομα και ημερομηνίες. Να σημειωθεί ότι σε περίπτωση λάθους για να μη σταλούν λανθασμένες πληροφορίες στη βάση δεδομένων δίνονται στις ημερομηνίες προεπιλεγμένες τιμές και ο δείκτης παίρνει την τιμή 0.

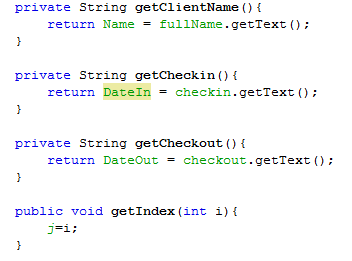




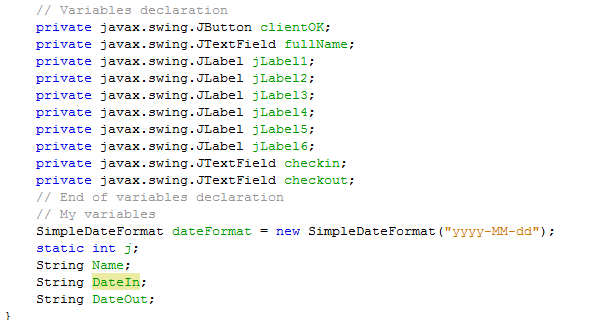


Οι πρώτες τρεις μέθοδοι, getClientName, getCheckin και getCheckout, παίρνουν τις πληροφορίες που έχει εισάγει ο χρήστης.

Η μέθοδος getIndex παίρνει από τη μέθοδο bookActionPerformed της SearchDB τον δείκτη του δωματίου που έχει επιλέξει ο χρήστης.



Παρακάτω ακολουθούν οι μεταβλητές της κλάσης.



**ΣΥΝΟΠΤΙΚΑ**

package Enoikiazomena;

import java.sql.SQLException;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class ClientInfo extends javax.swing.JFrame {

public ClientInfo() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

fullName = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

checkin = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

checkout = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

clientOK = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("Full Name:");

jLabel2.setText("Check-in Date:");

jLabel3.setText("Check-out Date:");

jLabel4.setText("Example: John Black");

jLabel5.setText("Example: 2012-07-24");

jLabel6.setText("Example: 2012-07-24");

clientOK.setText("OK");

clientOK.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

try {

clientOKActionPerformed(evt);

} catch ( SQLException | ParseException ex) {

Logger.getLogger(ClientInfo.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

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(21, 21, 21)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(clientOK)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(16, 16, 16)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel6)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(checkout, javax.swing.GroupLayout.DEFAULT\_SIZE, 268, Short.MAX\_VALUE)

.addComponent(jLabel5)

.addComponent(checkin)

.addComponent(jLabel4)

.addComponent(fullName)))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(fullName, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(checkin, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(9, 9, 9)

.addComponent(jLabel5)

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(checkout, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21, Short.MAX\_VALUE)

.addComponent(clientOK)

.addContainerGap())

);

pack();

}// </editor-fold>

private void clientOKActionPerformed(java.awt.event.ActionEvent evt)throws SQLException, ParseException {

this.checking();

Name = getClientName();

DateIn = getCheckin();

DateOut = getCheckout();

this.getIndex(j);

SearchDB.returnInfo(Name, DateIn, DateOut, j);

this.dispose();

}

//Checks if inputs are correct

private void checking(){

if(fullName.getText()==null || checkin.getText()==null || checkout.getText()==null){

JOptionPane.showMessageDialog(null, "Please check your informations");

checkin.setText("0000-00-00");

checkout.setText("0000-00-00");

j=0;

}else if (checkin.getText().trim().length() != dateFormat.toPattern().length() || checkout.getText().trim().length() != dateFormat.toPattern().length()){

JOptionPane.showMessageDialog(null, "Please check your informations");

checkin.setText("0000-00-00");

checkout.setText("0000-00-00");

j=0;

}

}

private String getClientName(){

return Name = fullName.getText();

}

private String getCheckin(){

return DateIn = checkin.getText();

}

private String getCheckout(){

return DateOut = checkout.getText();

}

public void getIndex(int i){

j=i;

}

// Variables declaration

private javax.swing.JButton clientOK;

private javax.swing.JTextField fullName;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JTextField checkin;

private javax.swing.JTextField checkout;

// End of variables declaration

// My variables

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

static int j;

String Name;

String DateIn;

String DateOut;

}

**ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ**

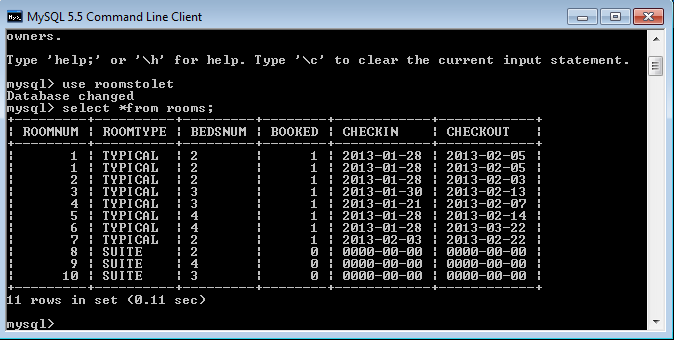
Η βάση δεδομένων που περιέχει όλα τα δωμάτια και πληροφορίες για κάθε ένα από αυτά ονομάζεται HOTEL. Διαθέτει έναν πίνακα ROOMS, ο οποίος έχει τα εξής πεδία:

* ROOMNUM (int)
* ROOMTYPE (varchar)
* BEDSNUM (int)
* BOOKED (boolean)
* CHECKIN (date)
* CHECKOUT (date)
* Το πεδίο ROOMNUM περιέχει τους αριθμούς των δωματίων. Κάθε δωμάτιο έχει ξεχωριστό αριθμό και δεν πρέπει να υπάρχει ο ίδιος αριθμός δύο φορές.
* Το πεδίο ROOMTYPE περιέχει τον τύπο του δωματίου. Θεωρούμε ότι υπάρχουν δύο τύποι δωματίων: TYPICAL και SUITE.
* Το πεδίο BEDSNUM περιέχει τον μέγιστο αριθμό ατόμων που χωράει το κάθε δωμάτιο.
* Το πεδίο BOOKED λαμβάνει την τιμή 0 αν το δωμάτιο είναι διαθέσιμο ή την τιμή 1 αν το δωμάτιο δεν είναι διαθέσιμο.
* Το πεδίο CHECKIN είναι η ημερομηνία άφιξης του πελάτη. Σε περίπτωση που το δωμάτιο δεν είναι κλεισμένο τότε λαμβάνει την τιμή: 0000-00-00.
* Το πεδίο CHECKOUT είναι η ημερομηνία αποχώρησης του πελάτη. Σε περίπτωση που το δωμάτιο δεν είναι κλεισμένο τότε λαμβάνει την τιμή: 0000-00-00.

Ως username και password της βάσης δεδομένων έχουν οριστεί τα:

* Username = root
* Password = root

Η βάση διαθέτει ήδη μερικές τιμές για δοκιμή.



**ΠΑΡΑΤΗΡΗΣΕΙΣ**

* Οι ημερομηνίες πρέπει να δίνονται ως εξής: yyyy/mm/dd -> 2012/06-25. Αυτό συμβαίνει διότι η mySQL αναγνωρίζει αυτόν τον τύπο ημερομηνιών.
* Η διαθεσιμότητα του δωματίου συμβολίζεται με 0 ή 1. 0=το δωμάτιο είναι διαθέσιμο, 1=το δωμάτιο δεν είναι διαθέσιμο.
* Θεωρούμε ότι όταν το δωμάτιο είναι διαθέσιμο έχει Check-in και Ckeck-out ίσες με 0000-00-00.
* Ο δείκτης είναι απαραίτητος διότι δείχνει σε ποιο δωμάτιο πρέπει να γίνουν αλλαγές. Η αρχική του τιμή είναι 0 ώστε αν ο χρήστης τερματίσει την εφαρμογή ή εισάγει λάθος στοιχεία να μην αλλάξει η βάση δεδομένων και πάρει λανθασμένα στοιχεία.
* Για τη συγγραφή του προγράμματος χρησιμοποιήθηκε το πρόγραμμα Netbeans IDE 7.1 και Java SE Runtime Environment 7. Στις κλάσεις δεν επεξηγούνται τα κομμάτια του κώδικα που έχουν σχέση με το interface και τις εξαιρέσεις. Πολλά από αυτά τα κομμάτια όπως επίσης και αρκετές μεταβλητές είναι auto-generated από το Netbeans.
* Για τη δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε MySQL Server 5.5.

Σημείωση: Το τελικό πρόγραμμα γράφτηκε σε Netbeans IDE 7.1, έτρεξε σε υπολογιστές που διαθέτουν Java SE Runtime Environment 7 και την βάση δεδομένων roomstolet. .Η βάση δεδομένων είναι το αρχείο «roomstoletfinal.sql» το οποίο πρέπει να εισάγουμε στη mySQL.

**ΓΝΩΣΤΑ ΠΡΟΒΛΗΜΑΤΑ**

* Το πρόγραμμα δεν κάνει update τη βάση δεδομένων. Όταν ο χρήστης πατήσει OK στην κλάση ClientInfo (info) και έχει εισάγει σωστές τιμές τα αποτελέσματα στέλνονται στη βάση δεδομένων αλλά δεν αλλάζει το πεδίο που θα έπρεπε να αλλάξει. Δοκιμές:

1. Room Number = 17

Full Name = test

Check-in Date = 2012-07-12

Check-out Date = 2012-07-16

Αποτέλεσμα: Κανένα error αλλά η τιμές BOOKED, CHECKIN, CHECKOUT του RO0MNUM = 17 παρέμειναν οι ίδιες (BOOKED=0, CHECKIN=CHECKOUT=0000-00-00).

1. Room Number = 9

Full Name = test

Check-in Date = 2012-06-20

Check-out Date = 2012-06-25

Αποτέλεσμα: Κανένα error αλλά η τιμές BOOKED, CHECKIN, CHECKOUT του RO0MNUM = 9 παρέμειναν οι ίδιες (BOOKED=0, CHECKIN=CHECKOUT=0000-00-00).