

Εργαστήριο 6 - Άσκηση - Ανάλυση

Εκφώνηση: Δημιουργείτε την εφαρμογή «fastfood» σε Java Swing με χρήση NetBeans χρησιμοποιώντας τις εικόνες που θα βρείτε στο moodle. Στην εφαρμογή ο υπάλληλος του μαγαζιού μπορεί να καταχωρεί την παραγγελία που του ζητάει ο πελάτης, επιλέγοντας τα προϊόντα που βρίσκονται στις λίστες, την ποσότητα από ένα spinner και να βλέπει την παραγγελία του και την συνολική τιμή. Ακολουθείστε τις παρακάτω οδηγίες:

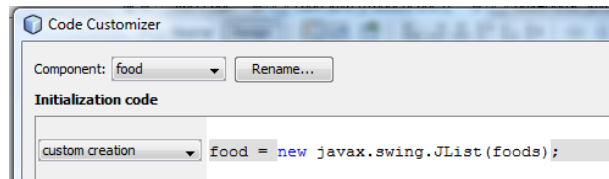
1. Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame, ο οποίος θα έχει
 - a. τίτλο «Ο ΜΑΚΗΣ»,
 - b. θα οριστεί με την έξοδο του να κλείνει και η εφαρμογή, και
 - c. θα βασιστεί σε ένα TabbedPane με δύο Tabs.
 - i. Το πρώτο θα λέγεται «ΦΟΡΜΑ ΠΑΡΑΓΓΕΛΙΑΣ» και
 - ii. το δεύτερο θα λέγεται «ΣΥΝΟΛΙΚΗ ΠΑΡΑΓΓΕΛΙΑ»

Παραγγελία	Ποσότητα	Τιμή Μονάδας
Γύρος	1	6
Πορτοκαλάδα	1	2
Σουτζουκάκι	2	5
Σουβλάκι	3	4
Χωριάτικη	4	3
Πράσινη	5	2
Γμιστά	6	7
ΚόκαΚόλα	2	3
Νερό	3	1
Λεμονάδα	4	2

2. Στο πρώτο TAB:

- a. Χρησιμοποιήστε ένα JSeparator για να χωρίσετε οριζόντια το panel σε δύο μέρη.
- b. Στο πάνω μέρος θα ορίσετε την παραγγελία των φαγητών ενώ στο κάτω των ποτών.
- c. Τα φαγητά θα τα ορίσετε σε πίνακα αλφαριθμητικών και ακεραίων, δηλαδή:

```
String foods[] = {"Γύρος", "Σουτζουκάκι", "Σουβλάκι",  
"Χωριάτικη", "Πράσινη", "Γεμιστά"};  
int foodsprice[] = {6, 5, 4, 3, 2, 7};
```
- d. Για την εμφάνιση των φαγητών θα χρησιμοποιήσετε ένα JList το οποίο τοποθετείται αυτόματα μέσα σε JScrollPane. Το JList θα το ορίσετε με βάση τον πίνακα foods από το Customize Code, όπως φαίνεται στην παρακάτω εικόνα.



Επίσης διαγράψτε από τα properties του JList τις τιμές item 1, item 2, item 3, item 4, item 5 της ιδιότητας model.

- Ορίστε μοντέλο απλής επιλογής (Single SelectionMode).
 - Προγραμματίστε το γεγονός ListSelection στη JList, ώστε να εμφανίζετε η κατάλληλη εικόνα σε ετικέτα ανάλογα με την επιλογή σας από τη λίστα. Χρησιμοποιήστε τη μέθοδο getSelectedIndex() για να πάρετε την θέση της επιλογής σας από τη λίστα.
 - Στο πάνω μέρος προσθέστε ένα JSpinner, για να γίνει η διαχείριση της ποσότητας των φαγητών - αλλάξτε την ιδιότητα μοντέλο σε «Μοντέλο Αριθμών» με ελάχιστο 1.
 - Τέλος, στο πάνω μέρος τοποθετήστε ένα JButton.
- e. Τα ποτά θα τα ορίσετε με ένα πίνακα αλφαριθμητικών και ένα πίνακα ακεραίων, δηλαδή:

```
String drinks[] = { "Πορτοκαλάδα", "ΚόκαΚόλα", "Νερό",  
"Λεμονάδα"};  
int drinksprice[] = {2, 3, 1, 2};
```
 - f. Για την εμφάνιση των ποτών θα χρησιμοποιήσετε ένα JComboBox, το οποίο θα το ορίσετε με βάση τον πίνακα drinks παρόμοια με τον τρόπο που ορίσατε τη λίστα για τον πίνακα foods.
 - g. Προγραμματίστε το γεγονός ActionPerformed στο JComboBox ώστε να εμφανίζετε η κατάλληλη εικόνα σε ετικέτα ανάλογα με την επιλογή σας από το ComboBox.
 - h. Ορίστε ένα JSpinner για τη διαχείριση της ποσότητας των ποτών όπως το ορίσατε για τα φαγητά και προσθέστε ένα JButton.

3. Το δεύτερο TAB:
 - Χωρίστε το με χρήση δύο υποδοχέων (Panel).
 - Στο πάνω τμήμα, τοποθετήστε σε ένα Panel τα κατάλληλα συστατικά για να εμφανίσετε το συνολικό κόστος της παραγγελίας.
 - Στο δεύτερο τμήμα εμφανίστε αναλυτικά την παραγγελία. Εισάγετε ένα Swing Control, το Table, το οποίο θα πρέπει να μπει μέσα σε ScrollPane (εισάγοντας το Table μπαίνει αυτόματα μέσα σε scrollpane).
 - Φροντίστε επιλέγοντας (Table Contents) ο Πίνακας να έχει τους σωστούς τίτλους επικεφαλίδων των στηλών (columns) - οι τύποι που θα δέχεται να είναι String, integer, integer και να μην έχει καμία γραμμή (row).
4. Προγραμματίστε τα γεγονότα actionPerformed των δύο JButton ώστε να ενημερώνονται κατάλληλα οι γραμμές του πίνακα και το συνολικό κόστος της παραγγελίας στο δεύτερο TAB. Για να γεμίσετε τον πίνακα θα πρέπει να χρησιμοποιήσετε κατάλληλα τις παρακάτω εντολές:

```
Object[] row = { Τιμή_1ης_στήλης, Τιμή_2ης_στήλης, Τιμή_3ης_στήλης};  
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();  
model.addRow(row);
```

Θα χρειαστεί να προσθέσετε βιβλιοθήκη για το DefaultTableModel .

Σημείωση: Μέθοδοι που πιθανώς θα σας φανούν χρήσιμες:

- setText(string)
- Αντικείμενο ImageIcon
- setIcon(Αντικείμενο ImageIcon)
- getSelectedIndex()
- String.valueOf
- JSpinner.getValue()
- Integer.valueOf

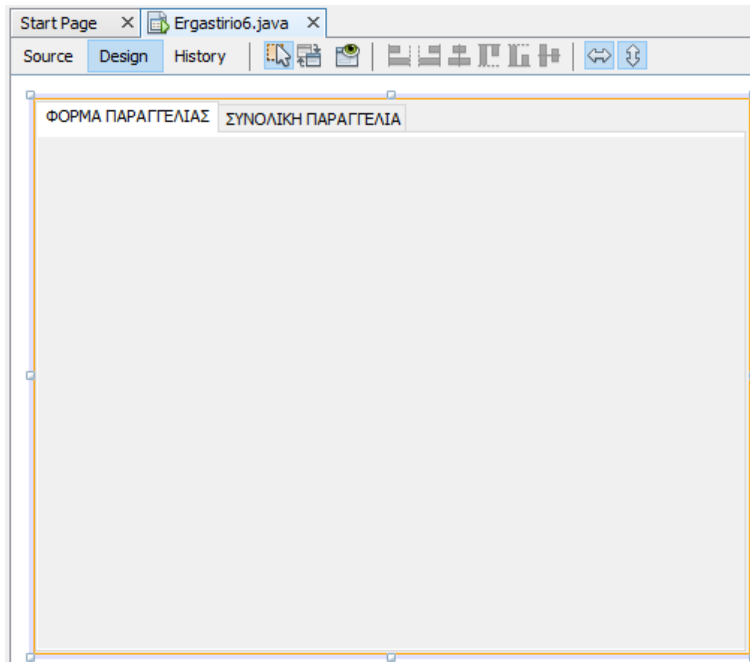
Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi8.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio6.

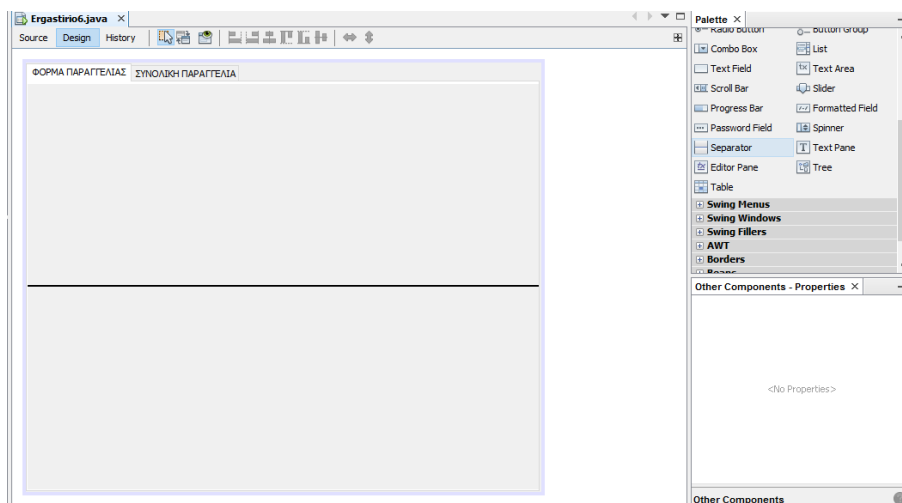
Βήμα 3: Ορισμός Ιδιοτήτων στο JFrame. Ορίζουμε ως τίτλο του JFrame το «Ο ΜΑΚΗΣ».

Βήμα 4: Η εφαρμογή θα έχει δύο καρτέλες. Έτσι, τοποθετώ στο JFrame ένα JTabbedPane. Στο JTabbedPane τοποθετώ δύο JPanel, ένα για κάθε καρτέλα και αλλάζω τον τίτλο τους (Edit Text) όπως φαίνεται στην Εικόνα 1.



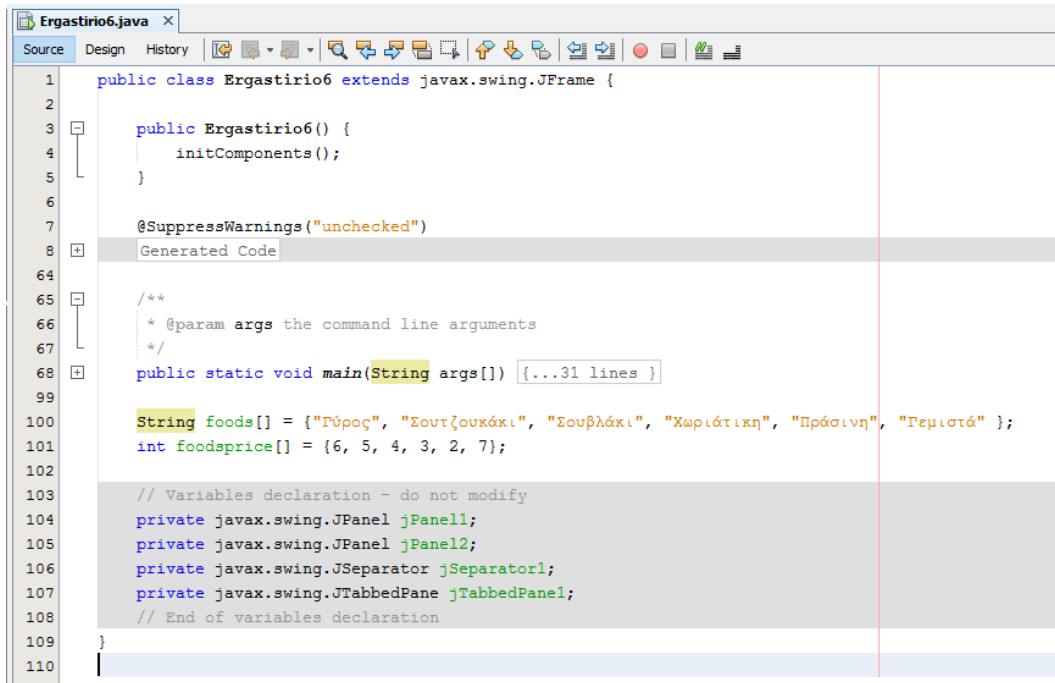
Εικόνα 1: Η εφαρμογή θα έχει δύο καρτέλες.

Βήμα 5: Για να διαχωρίσουμε το πρώτο Panel σε δύο τμήματα χρησιμοποιούμε ένα Separator. Αλλάζουμε κατάλληλα το μέγεθος του και τις ιδιότητες foreground και background. Το αποτέλεσμα το βλέπουμε στην Εικόνα 2.



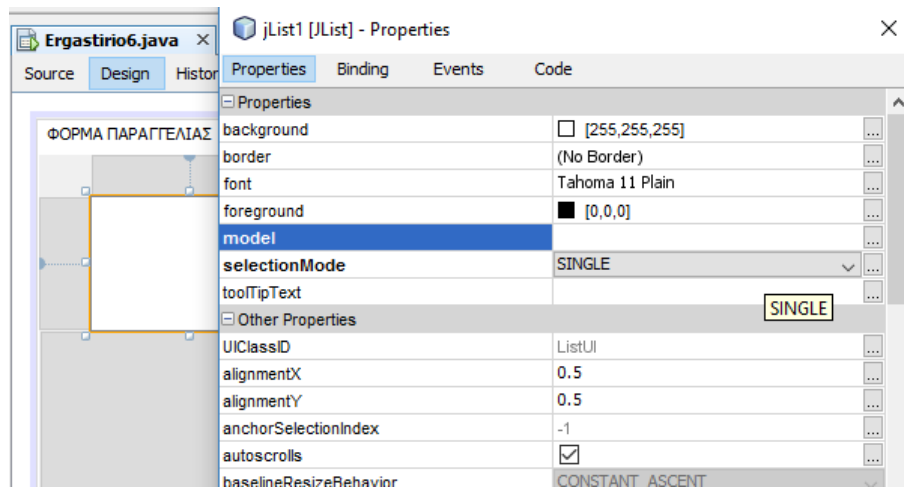
Εικόνα 2: Πρόσθεση του separator

Βήμα 6: Δηλώνουμε δύο arrays ως global μεταβλητές (Εικόνα 3). Οι δύο arrays περιέχουν τα δεδομένα για τα φαγητά, δηλαδή περιγραφή φαγητού και τιμή.

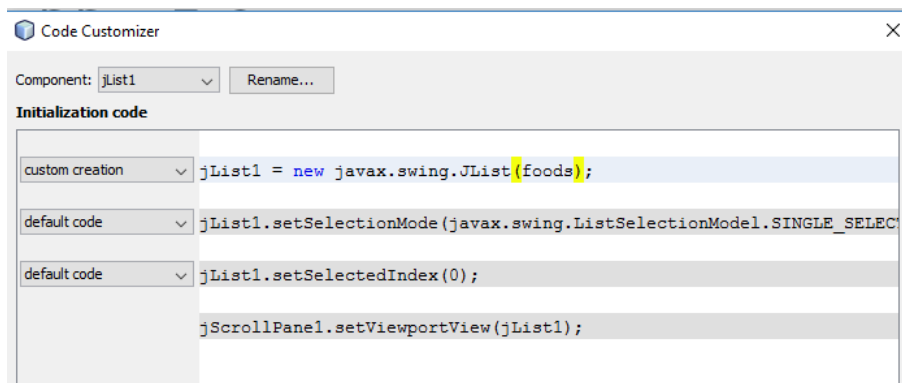


Εικόνα 3: Δήλωση πινάκων δεδομένων

Βήμα 7: Τοποθετούμε ένα JList στην πρώτη καρτέλα για να εμφανίσουμε τα φαγητά. Ορίζουμε κατάλληλα τις τιμές σε τρεις ιδιότητες του JList. Από την ιδιότητα model διαγράφουμε τις τιμές. Στην ιδιότητα selectionMode ορίζουμε την τιμή SINGLE (Εικόνα 4). Θέλουμε να επιλέγουμε μόνο μία τιμή φαγητού κάθε φορά. Τέλος, στην ιδιότητα selectedIndex δίνουμε την τιμή 0, ώστε όταν εκτελείται η εφαρμογή να ξεκινά με επιλεγμένο το πρώτο στοιχείο της λίστας. Για να «φορτώσει» η λίστα τις τιμές από τον πίνακα foods, θα πρέπει από το pop up μενού που ανοίγει όταν κάνουμε δεξί click στο JList να επιλέξουμε το “customize code” και στη συνέχεια να ορίσουμε στην εντολή δημιουργίας της λίστας, ως παράμετρο τον πίνακα foods, όπως φαίνεται στην Εικόνα 5 – θα πρέπει να αλλάξετε το “default code” σε “custom creation”.



Εικόνα 4: Οι ιδιότητες του JList



Εικόνα 5: Code Customizer για τη JList

Αλλάζουμε το variable name του JList σε «food».

Βήμα 8: Σε αυτό το βήμα θα προγραμματίσουμε ένα γεγονός στη λίστα, ώστε όταν επιλέγεται ένα φαγητό να εμφανίζεται η εικόνα του σε μία ετικέτα.

Πρώτα από όλα τοποθετούμαι μία ετικέτα στο καμβά μας. Ορίζουμε από τις ιδιότητες της:

- Horizontal Size – την τιμή 110
- Vertical Size – την τιμή 100
- Border – Bevel border – type Lowered για να φαίνεται η εικόνα σε 3D κάδρο. Πειραματιστείτε.
- Text – διαγράψτε το κείμενο
- Variable name – foodicon.

Στη συνέχεια, ορίζουμε σε ImageIcon αντικείμενα, τις φωτογραφίες των φαγητών που θα χρησιμοποιήσουμε. Μπορείτε να ορίσετε τα αντικείμενα ως global μεταβλητές. Προφανώς θα χρειαστεί και η βιβλιοθήκη για την ImageIcon. Στην Εικόνα 6, στις γραμμές 125-130 μπορείτε να δείτε τη δήλωση των ImageIcon αντικειμένων.

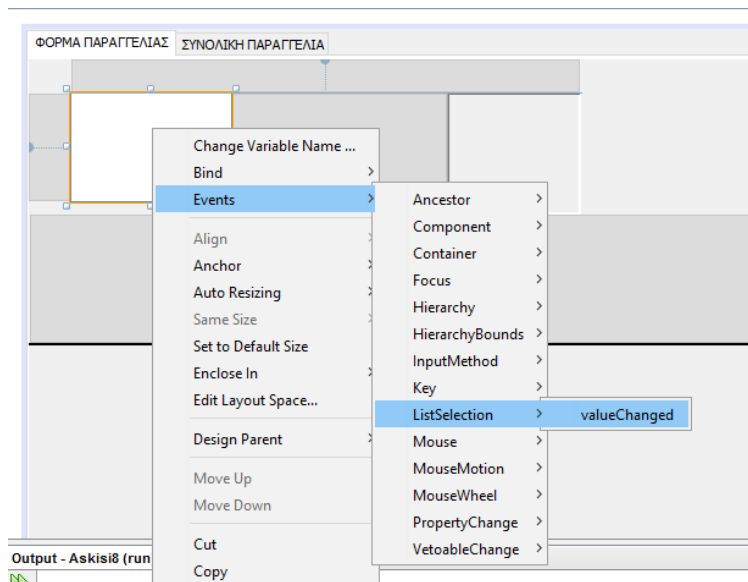
```

121
122 String foods[] = {"Γύρος", "Σουτζουκάκι", "Σουβλάκι", "Χωριάτικη", "Πράσινη", "Γεμιστά" };
123 int foodsprice[] = {6, 5, 4, 3, 2, 7};
124
125 ImageIcon gyros = new ImageIcon("gyros.jpg");
126 ImageIcon soutzoukaki = new ImageIcon("soutsoukaki.jpg");
127 ImageIcon souvlaki = new ImageIcon("souvlaki.jpg");
128 ImageIcon xoriatiki = new ImageIcon("xoriatiki.jpg");
129 ImageIcon prasini = new ImageIcon("prasini.jpg");
130 ImageIcon gemista = new ImageIcon("gemista.jpg");
131

```

Εικόνα 6: Δημιουργία ImageIcon αντικειμένων για τα αρχεία εικόνων των φαγητών

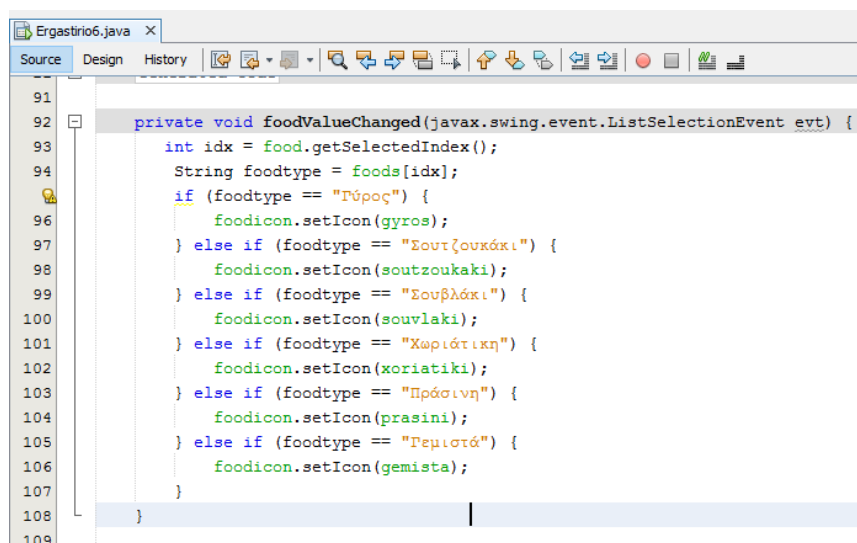
Ολοκληρώνουμε αυτό το βήμα προγραμματίζοντας το event ListSelection του JList, όπως φαίνεται στην Εικόνα 7.



Εικόνα 7: Event ListSection-valueChanged του JList

Σε αυτό το γεγονός θα πρέπει να βρίσκουμε ποιο στοιχείο της λίστας έχει επιλεγεί και με βάση αυτό να εμφανίζουμε την κατάλληλη εικόνα. Για να βρούμε το στοιχείο της λίστας που είναι επιλεγμένο και έχει «πυροδοτήσει» το γεγονός μπορούμε να χρησιμοποιήσουμε τις μεθόδους `getSelectedIndex()` και `getSelectedValue()`.

Με την `getSelectedIndex()` επιστρέφεται η θέση στη λίστα του στοιχείου που έχει επιλεγεί - η μέτρηση ξεκινά από το μηδέν. Στην Εικόνα 8, στη γραμμή 93 χρησιμοποιείται η `getSelectedIndex()` στη λίστα `food`, και αποθηκεύεται η θέση στην ακεραία μεταβλητή `idx`. Επειδή, η λίστα δημιουργήθηκε με βάση τον πίνακα `foods`, τα φαγητά έχουν την ίδια θέση στη λίστα και στον πίνακα. Αυτό εκμεταλλευόμαστε στη γραμμή 94 και αποθηκεύουμε το όνομα του φαγητού στην μεταβλητή `foodtype`.

Εικόνα 8: ListSelection 1^{ος} τρόπος

Με την `getSelectedValue()` επιστρέφεται η επιλεγμένη τιμή (και όχι η θέση της) στη λίστα, όπως ακριβώς παρουσιάζεται στη λίστα, δηλαδή στο παράδειγμά μας το όνομα του φαγητού. Στην Εικόνα 9, παρουσιάζεται η λύση με χρήση της `getSelectedValue()` ενώ στην Εικόνα 10, παρουσιάζεται η λύση με χρήση της `switch-case` σύνταξης αντί της `if` σύνταξης.

```

Ergastirio6.java
Source Design History
Generated Code
private void foodValueChanged(javax.swing.event.ListSelectionEvent evt) {
    String foodtype = food.getSelectedValue();
    if (foodtype == "Γύρος") {
        foodicon.setIcon(gyros);
    } else if (foodtype == "Σουτζουκάκι") {
        foodicon.setIcon(soutzoukaki);
    } else if (foodtype == "Σουβλάκι") {
        foodicon.setIcon(souvlaki);
    } else if (foodtype == "Χωριάτικη") {
        foodicon.setIcon(xoriatiki);
    } else if (foodtype == "Πράσινη") {
        foodicon.setIcon(prasini);
    } else if (foodtype == "Γεμιστά") {
        foodicon.setIcon(gemista);
    }
}

```

Εικόνα 9: ListSelection 2^{ος} τρόπος

```

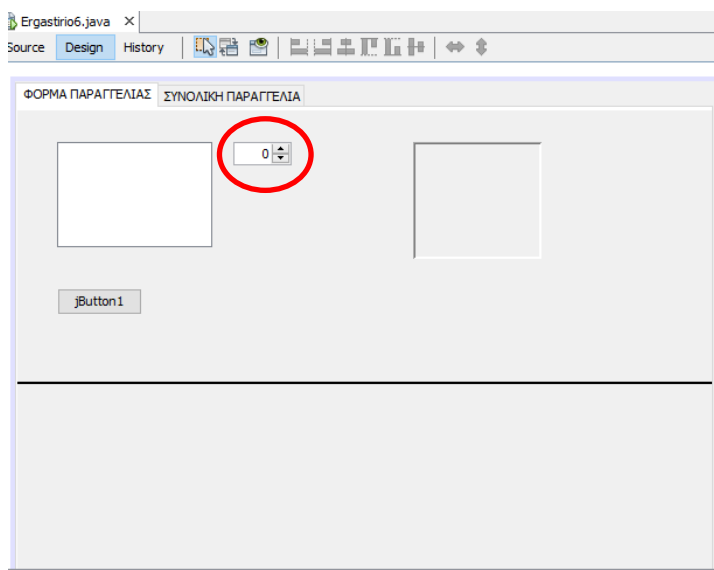
Ergastirio6.java
Source Design History
private void foodValueChanged(javax.swing.event.ListSelectionEvent evt) {
    String foodtype = food.getSelectedValue();
    if (null != foodtype) switch (foodtype) {
        case "Γύρος":
            foodicon.setIcon(gyros);
            break;
        case "Σουτζουκάκι":
            foodicon.setIcon(soutzoukaki);
            break;
        case "Σουβλάκι":
            foodicon.setIcon(souvlaki);
            break;
        case "Χωριάτικη":
            foodicon.setIcon(xoriatiki);
            break;
        case "Πράσινη":
            foodicon.setIcon(prasini);
            break;
        case "Γεμιστά":
            foodicon.setIcon(gemista);
            break;
        default:
            break;
    }
}

```

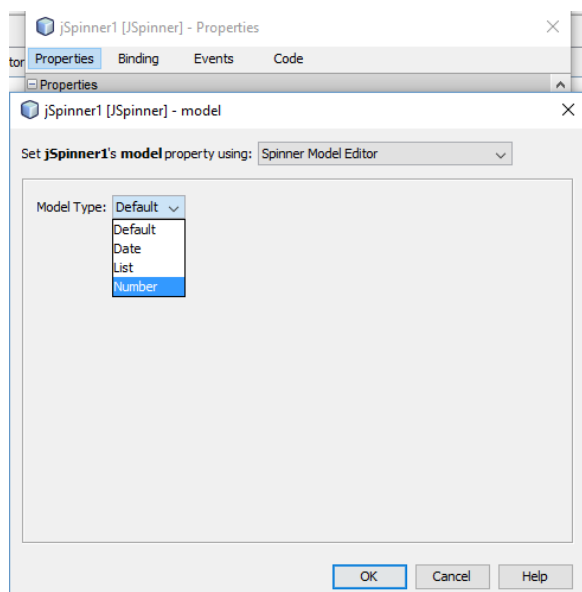
Εικόνα 10: ListSelection 3^{ος} τρόπος

Βήμα 9: Σε αυτό το βήμα θα προσθέσουμε ένα spinner στον καμβά μας για να διαχειριστούμε την ποσότητα παραγγελίας κάθε προϊόντος. Στην Εικόνα 11, παρατηρούμε το spinner δίπλα στη λίστα ενώ έχουμε προσθέσει και ένα JButton. Ορίζουμε τις παρακάτω ιδιότητες για το spinner:

- Horizontal Size: 50
- Ανοίγουμε το model και επιλέγουμε Number (Εικόνα 12). Εκεί ορίζουμε την τιμή 1 στις ιδιότητες *Initial Value* και *Minimum*. Ορίζουμε αυτή την τιμή για να μην μπορούμε να ορίσουμε 0 ή αρνητική τιμή στην ποσότητα.



Εικόνα 11: Το spinner



Εικόνα 12: Το spinner model

Βήμα 10: Ότι σχεδιάσαμε για την παραγγελία των φαγητών στο πάνω μέρος του καμβά θα σχεδιάσουμε σε αυτό το βήμα για την παραγγελία των ποτών στο κάτω μέρος του καμβά. Έτσι, όπως φαίνεται στην Εικόνα 13, γραμμές 169-170 ορίζουμε τους πίνακες drinks και drinksprice. Επίσης, στις γραμμές 178-181 ορίζουμε τέσσερα ImageIcon αντικείμενα για τις εικόνες των ποτών.

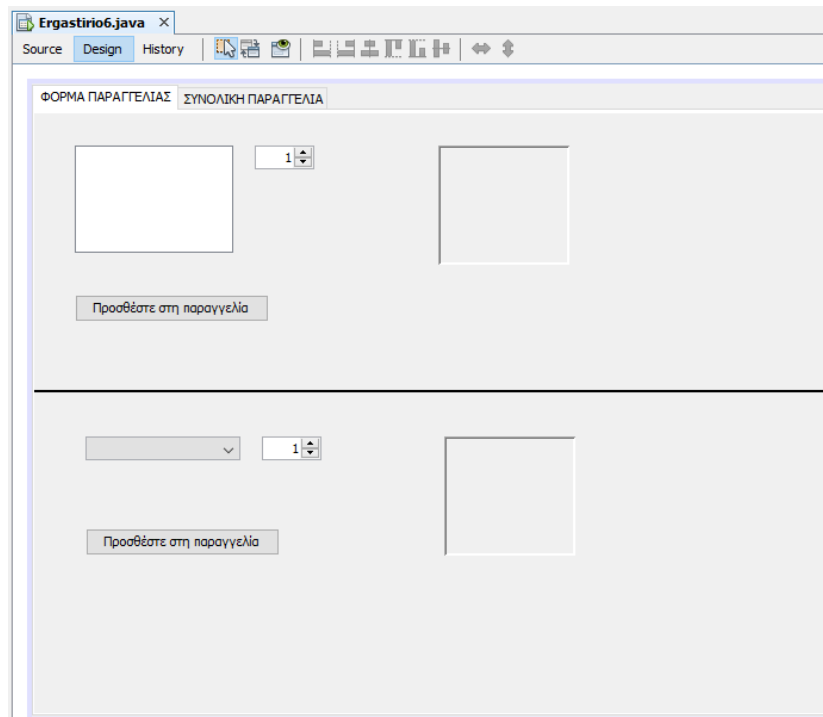
```

Ergastirio6.java x
Source Design History
166
167 String foods[] = {"Γύρος", "Σουτζουκάκι", "Σουβλάκι", "Χωριάτικη", "Πράσινη", "Γεμιστά" };
168 int foodsprice[] = {6, 5, 4, 3, 2, 7};
169 String drinks[] = { "Πορτοκαλάδα", "ΚόκαΚόλα", "Νερό", "Λεμονάδα"};
170 int drinksprice[] = {2, 3, 1, 2};
171
172 ImageIcon gyros = new ImageIcon("gyros.jpg");
173 ImageIcon soutsoukaki = new ImageIcon("soutsoukaki.jpg");
174 ImageIcon souvlaki = new ImageIcon("souvlaki.jpg");
175 ImageIcon xoriatikiki = new ImageIcon("xoriatikiki.jpg");
176 ImageIcon prasini = new ImageIcon("prasini.jpg");
177 ImageIcon gemista = new ImageIcon("gemista.jpg");
178 ImageIcon fanta = new ImageIcon("fanta.jpg");
179 ImageIcon kokakola = new ImageIcon("kokakola.jpg");
180 ImageIcon nero = new ImageIcon("nero.jpg");
181 ImageIcon lemonada = new ImageIcon("lemonada.jpg");
182 // Variables declaration - do not modify

```

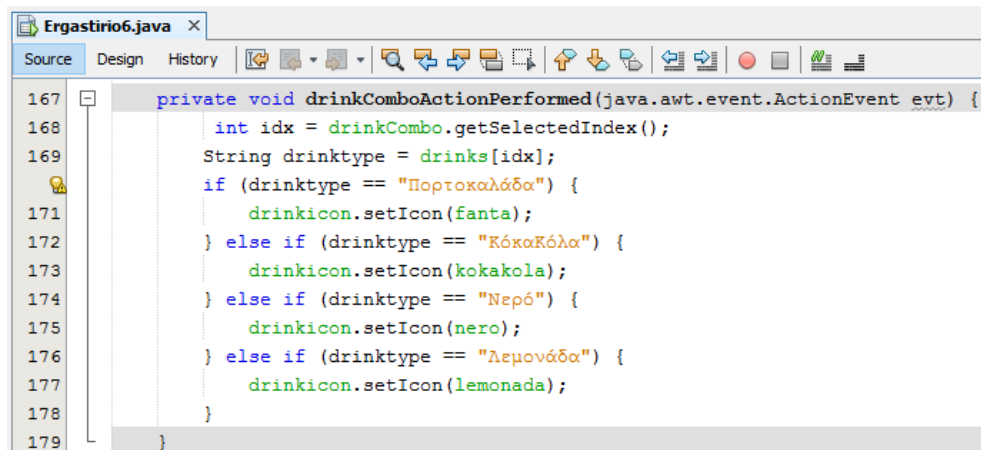
Εικόνα 13: Τα στοιχεία για τα ποτά

Προσθέτουμε ένα combo box, ένα spinner, ένα Label και ένα Button, όπως φαίνεται στην Εικόνα 14 και τα διαμορφώνουμε με παρόμοιο τρόπο με τον τρόπο που διαμορφώσαμε τα συστατικά για την παραγγελία των φαγητών.



Εικόνα 14: Το TAB Φόρμα Παραγγελίας

Δημιουργούμε το γεγονός `ActionPerformed` για το `combobox` και προγραμματίζουμε την εμφάνιση των εικόνων των ποτών στην ετικέτα (Εικόνα 15) με αντίστοιχο τρόπο όπως προγραμματίσαμε τη λίστα των φαγητών για την ίδια ενέργεια.



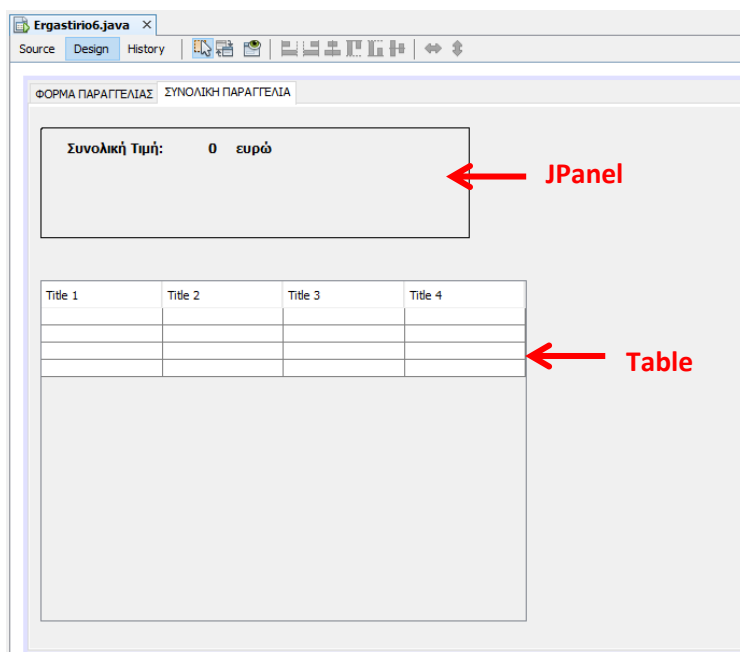
```

167 private void drinkComboActionPerformed(java.awt.event.ActionEvent evt) {
168     int idx = drinkCombo.getSelectedIndex();
169     String drinktype = drinks[idx];
170     if (drinktype == "Πορτοκαλάδα") {
171         drinkicon.setIcon(fanta);
172     } else if (drinktype == "ΚόκαΚόλα") {
173         drinkicon.setIcon(kokakola);
174     } else if (drinktype == "Νερό") {
175         drinkicon.setIcon(nero);
176     } else if (drinktype == "Λεμονάδα") {
177         drinkicon.setIcon(lemonada);
178     }
179 }

```

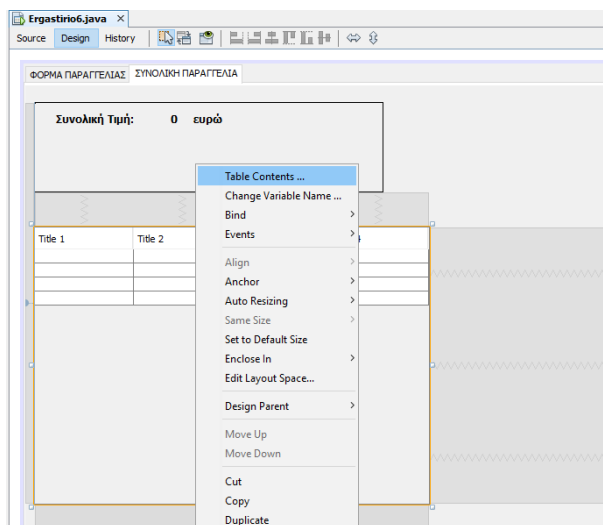
Εικόνα 15: Το γεγονός `ActionPerformed` του `combobox`

Βήμα 11: Θα συνεχίσουμε με τη σχεδίαση της δεύτερης καρτέλας. Θα προσθέσουμε ένα `JPanel` και ένα `Table`, όπως φαίνεται στην Εικόνα 16. Στο `JPanel` θα τοποθετήσουμε τρία `JLabel`. Τα δύο ακριανά θα έχουν παθητικό χαρακτήρα παρουσιάζοντας ένα στατικό κείμενο. Στο μεσαίο `JLabel` θα εμφανίζουμε τη συνολική τιμή της παραγγελίας. Το μεσαίο `JLabel` θα έχει όνομα μεταβλητής *stim1* και επίσης η ιδιότητα *Horizontal Alignment* θα έχει την τιμή *Right* και η ιδιότητα *Horizontal Size* θα έχει την τιμή 40. Και στα τρία `JLabel` τα *font* θα είναι *Tahoma bold 12*.

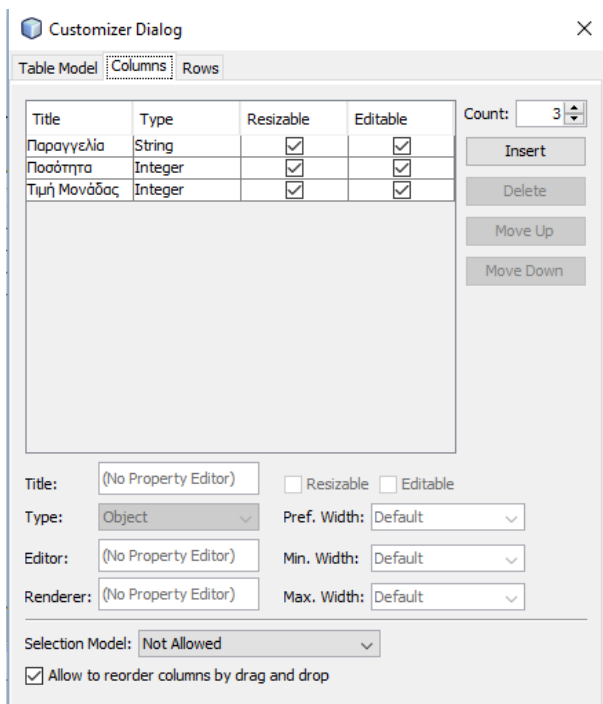


Εικόνα 16: Σχεδίαση της δεύτερης καρτέλας

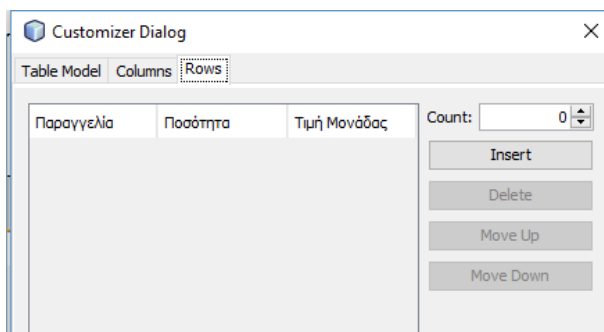
Βήμα 12: Σε αυτό το βήμα θα διαμορφώσουμε το συστατικό table. Με δεξί click Πάνω στο table ανοίγει το pop up menu και επιλέγουμε *Table Contents* (Εικόνα 17). Διαμορφώνουμε τους τίτλους των στηλών του πίνακα και τους τύπους δεδομένων που θα παρουσιαστούν στις στήλες, όπως φαίνεται στην Εικόνα 18. Τέλος, διαμορφώνουμε τις γραμμές του πίνακα. Στο παράδειγμα ορίζουμε μηδέν γραμμές γιατί όλες θα δημιουργηθούν δυναμικά κατά την εκτέλεση του προγράμματος (Εικόνα 19).



Εικόνα 17: Από το pop menu του table επιλέγουμε Table Contents



Εικόνα 18: Διαμόρφωση των στηλών του πίνακα



Εικόνα 19: Διαμόρφωση των γραμμών του πίνακα

Βήμα 13: Σε αυτό το βήμα θα προγραμματίσουμε το γεγονός `ActionPerformed` των δύο `JButton` της πρώτης καρτέλας. Επιλέγοντας ένα `JButton` θα πρέπει να προστεθεί μία γραμμή στην παραγγελία – για φαγητό αν επιλεγεί το `JButton` στο πάνω μέρος της καρτέλας ή για ποτό αν επιλεγεί το άλλο. Στην Εικόνα 20, παρουσιάζεται ο κώδικας του `JButton` για τα φαγητά. Αναλυτικά:

- Στη γραμμή 263, αποθηκεύεται στη μεταβλητή `idx` η θέση επιλογής της λίστας.
- Στη γραμμή 264, αποθηκεύεται στη μεταβλητή `foodtype` το όνομα του φαγητού.
- Στη γραμμή 265, αποθηκεύεται στη μεταβλητή `strposotita` η τιμή του `JSpinner`, που δείχνει την ποσότητα παραγγελίας, αφού πρώτα μετατραπεί σε `String`.
- Στη γραμμή 266, αποθηκεύεται στη μεταβλητή `posotita` η τιμή της `strposotita` αφού μετατραπεί σε `Integer`.
- Στη γραμμή 267, αποθηκεύεται στη μεταβλητή `timi` η τιμή μονάδας του φαγητού που θα προστεθεί στην παραγγελία. Την τιμή θα την πάρουμε από τον πίνακα `foodsprice`.
- Στη γραμμή 268, δημιουργείται το αντικείμενο `row`, το οποίο αποτελείται από τις τιμές των τριών μεταβλητών `foodtype`, `posotita` και `timi` και θα χρησιμοποιηθεί για να γεμίσει μία γραμμή στο συστατικό `JTable` της δεύτερης καρτέλας.
- Στη γραμμή 269, αποθηκεύεται στο αντικείμενο `model` το μοντέλο του `JTable` όπως ορίστηκε στο προηγούμενο βήμα.
- Στη γραμμή 270, προστίθεται μία γραμμή στον `JTable`.
- Στη γραμμή 271 και 272, υπολογίζεται το συνολικό ποσό και εμφανίζεται στο `JLabel` `stimi`.

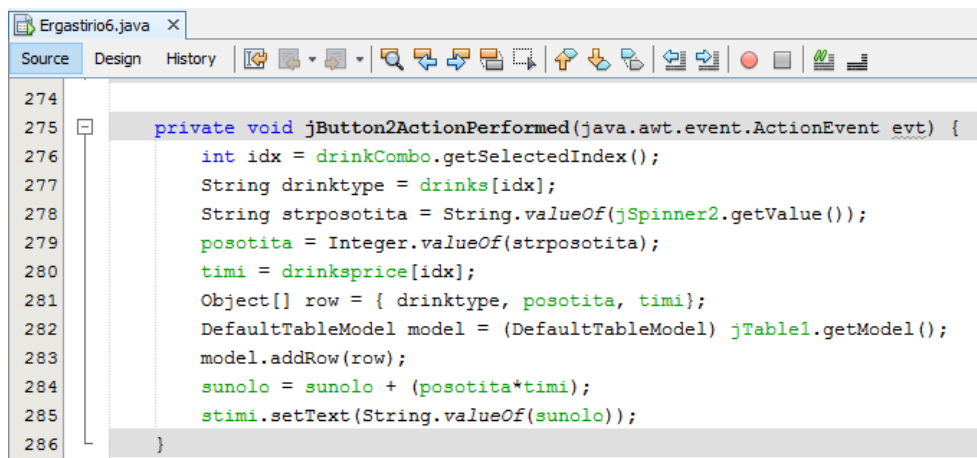
```

Ergastrio6.java x
Source Design History
262 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
263     int idx = food.getSelectedIndex();
264     String foodtype = foods[idx];
265     String strposotita = String.valueOf(jSpinner1.getValue());
266     posotita = Integer.valueOf(strposotita);
267     timi = foodsprice[idx];
268     Object[] row = { foodtype, posotita, timi};
269     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
270     model.addRow(row);
271     sunolo = sunolo + (posotita*timi);
272     stimi.setText(String.valueOf(sunolo));
273 }

```

Εικόνα 20: Το γεγονός `ActionPerformed` του `JButton` των φαγητών

Στην Εικόνα 21, παρουσιάζεται ο κώδικας του JButton για τα ποτά. Ο κώδικας είναι αντίστοιχος με τον κώδικα για το JButton για τα φαγητά. Προφανώς, θα πρέπει να δηλωθούν οι *sunolo*, *posotita* και *timi* ως global μεταβλητές τύπου ακεραίου.



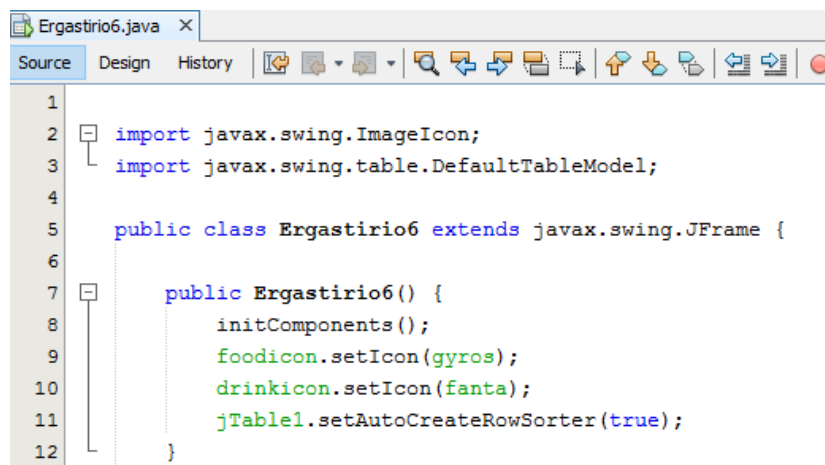
```

274
275 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
276     int idx = drinkCombo.getSelectedIndex();
277     String drinktype = drinks[idx];
278     String strposotita = String.valueOf(jSpinner2.getValue());
279     posotita = Integer.valueOf(strposotita);
280     timi = drinksprice[idx];
281     Object[] row = { drinktype, posotita, timi};
282     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
283     model.addRow(row);
284     sunolo = sunolo + (posotita*timi);
285     stimi.setText(String.valueOf(sunolo));
286 }

```

Εικόνα 21: Το γεγονός actionPerformed του JButton των ποτών

Βήμα 14: Τελευταίο βήμα είναι η διαμόρφωση της αρχικής εμφάνισης της εφαρμογής. Έτσι, καλό θα ήταν ξεκινώντας η εφαρμογή να προτείνει ως επιλεγμένο το πρώτο προϊόν στη λίστα και στο comboBox, εμφανίζοντας και την αντίστοιχη εικόνα. Αυτό γίνεται με τον κώδικα στις γραμμές 9 και 10, ο οποίος εκτελείται αμέσως μετά τη μέθοδο initComponents στο δομητή της τάξης της εφαρμογής. Επίσης, ορίζεται ως επιλεγμένη η θέση 0 της λίστας και του comboBox, όπου ορίζουμε και για τα δύο την ιδιότητα selectedIndex να έχει την τιμή 0. Τέλος, με τον κώδικα στην γραμμή 11 δίνεται η δυνατότητα στο χρήστη να ταξινομεί τα περιεχόμενα του συστατικού JTable επιλέγοντας δυναμικά (όταν εκτελείται η εφαρμογή) τον τίτλο της στήλης που θέλει να είναι το κριτήριο ταξινόμησης.



```

1
2 import javax.swing.ImageIcon;
3 import javax.swing.table.DefaultTableModel;
4
5 public class Ergastirio6 extends javax.swing.JFrame {
6
7     public Ergastirio6() {
8         initComponents();
9         foodicon.setIcon(gyros);
10        drinkicon.setIcon(fanta);
11        jTable1.setAutoCreateRowSorter(true);
12    }

```

Εικόνα 22: Διαμόρφωση αρχικής εμφάνισης της εφαρμογής