ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ «ΠΛΗΡΟΦΟΡΙΚΗ»
ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ ΠΛΗ24 «ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ»

Netbeans Binding

Α. ΘΑΝΟΣ

Περιεχόμενα Διάλεξης

- Τι είναι το Binding
- Βασικά components
- Παραδείγματα Binding
- Παράδειγμα GUI

Τι είναι το Binding

Τι είναι το Binding

- Είναι ο τρόπος σύνδεσης και συγχρονισμού ενός Entity Bean ή μιας λίστας από Entity Beans με ένα GUI Component.

Τι είναι το Binding

- Επιτυγχάνεται με χρήση wizards (γίνεται και χειροκίνητα αλλά δεν ενδείκνυται)
- Εξασφαλίζει διαφανή συγχρονισμό μεταξύ GUI και Entity Bean:
  - όποια αλλαγή γίνεται στο bean παρουσιάζεται αυτόματα στο GUI
  - όποια αλλαγή γίνεται στο GUI αυτόματα αποτυπώνεται στα properties του bean
- Επιτυγχάνεται με την χρήση της κλάσης PropertyChangeSupport (γίνεται αυτόματα κατά τη δήλωση του binding στο wizard, χωρίς τη συγγραφή κώδικα από το χρήστη) και πυροδότηση ειδικών event κατά την κλήση των setter μεθόδων στα entity beans.

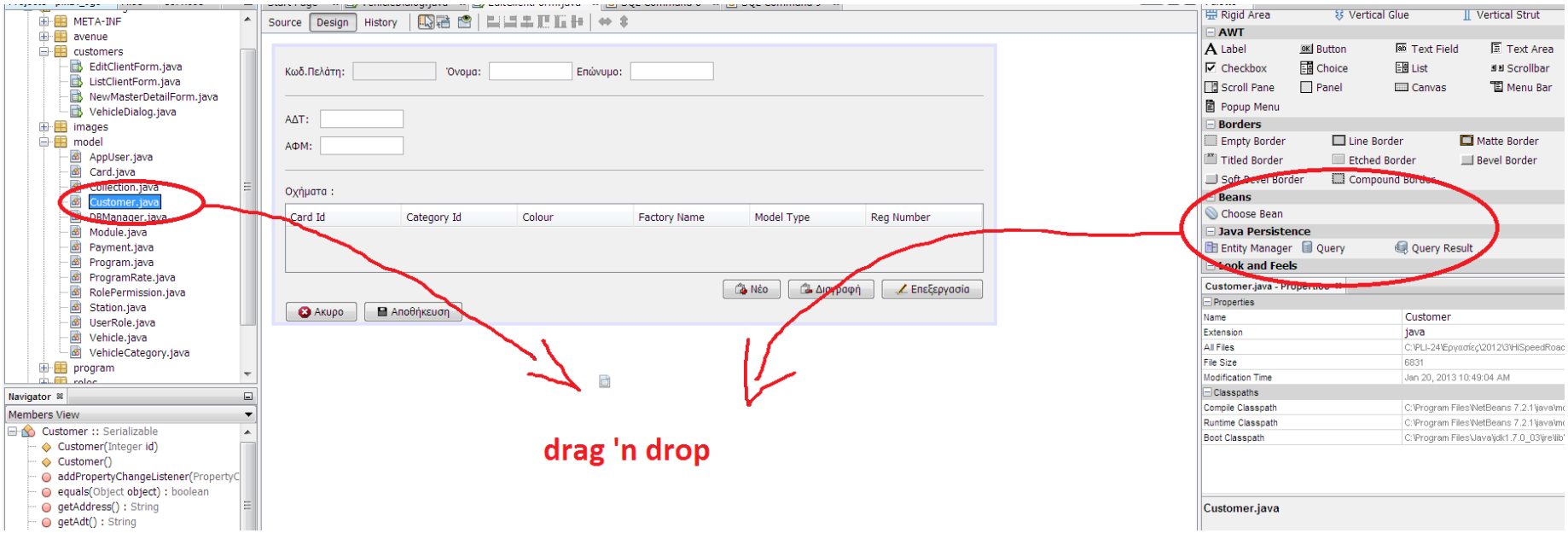
Βασικά Components

Βασικά Components

- Persistence tools
  - Entity Manager
  - Query
  - Result (List)
- Entity Beans
- Swing Components

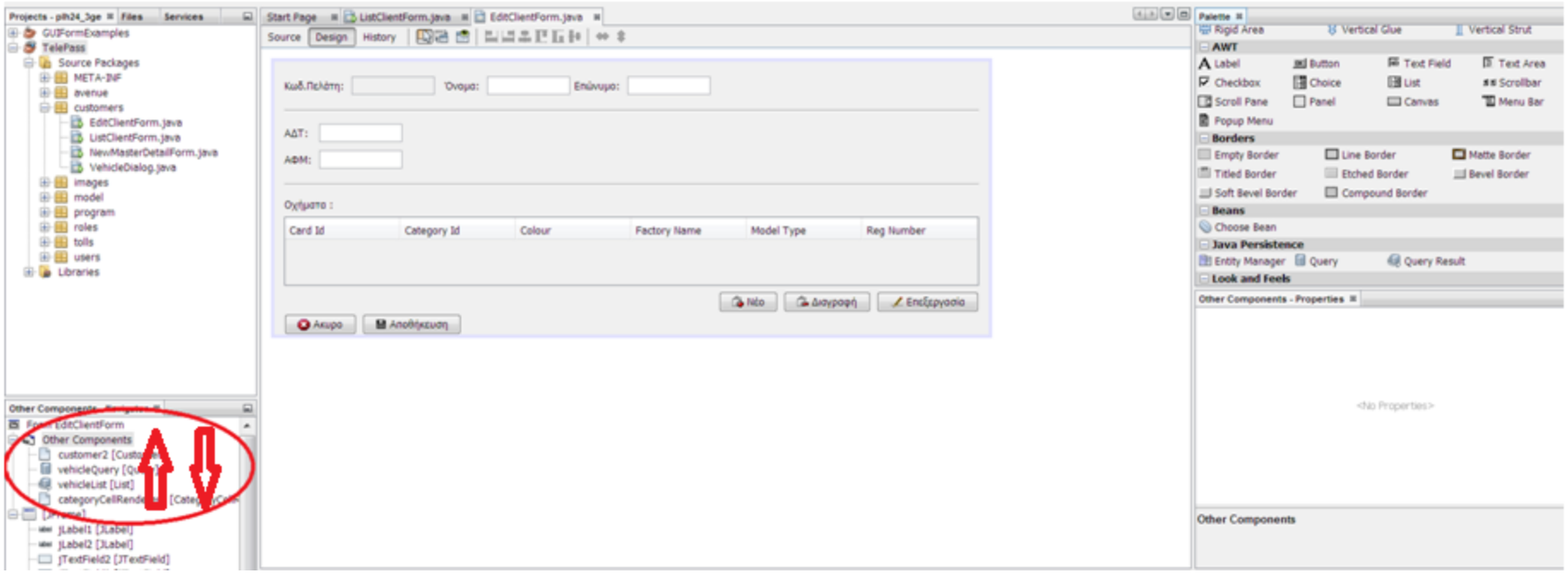
Entities και components

- Για να μπορούν να χρησιμοποιηθούν τα entities ή τα components στα wizards θα πρέπει να γίνει εισαγωγή τους στο design περιβάλλον με drag ‘n drop



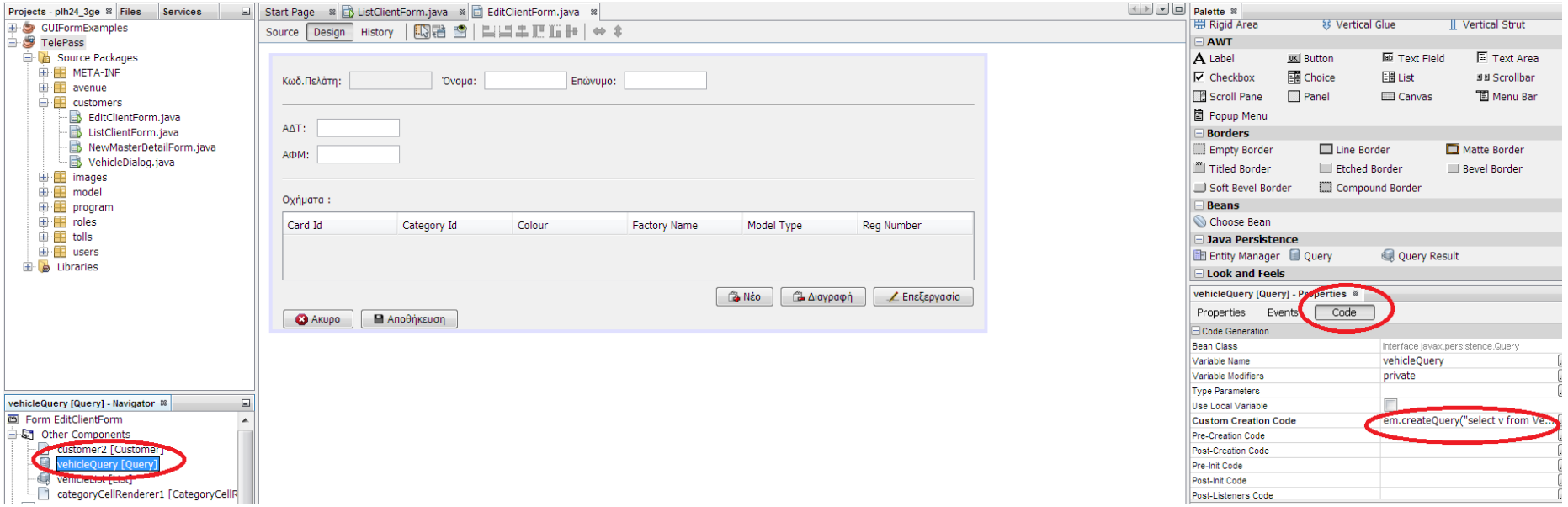
Design Navigator

- Σε design view όλα τα components που μπορούν να χρησιμοποιηθούν σε wizards παρουσιάζονται στο Design Navigator.
- Μπορούμε να αλλάξουμε τη σειρά που εμφανίζονται στον navigator, επιρρεαζοντας τη σειρά που εκτελούνται στο πρόγραμμα



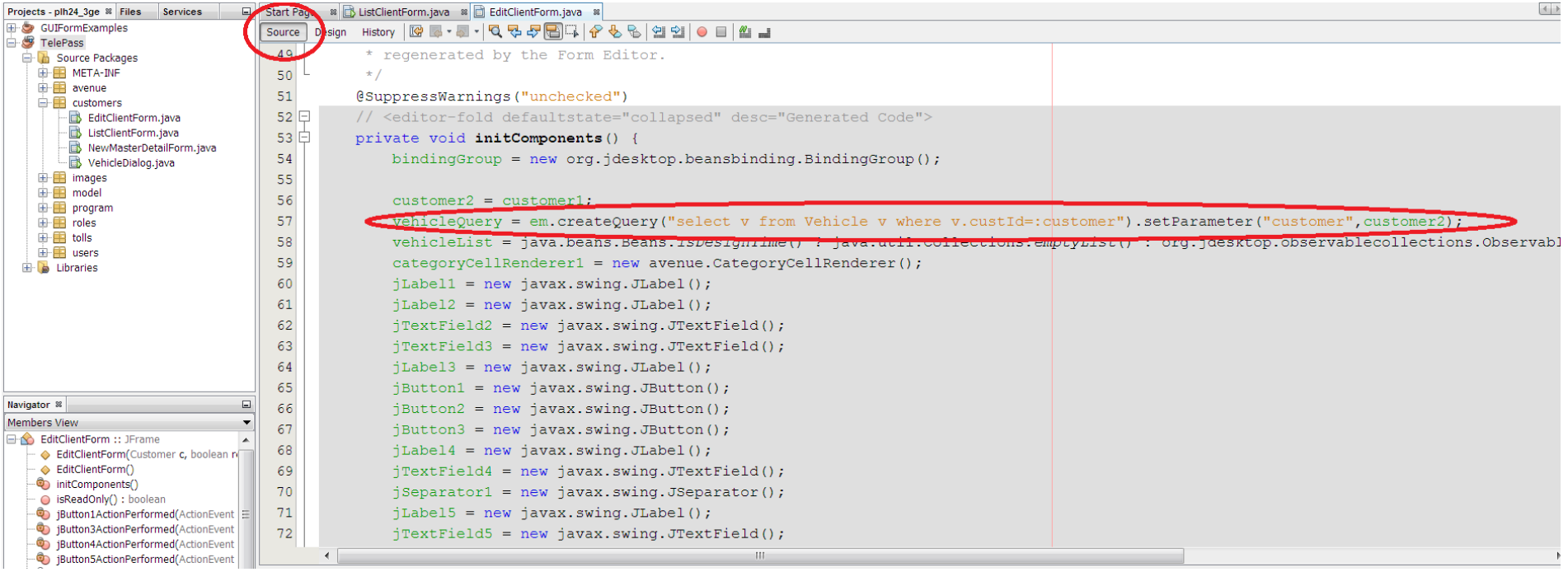
Query Component

- Μπορούμε να χρησιμοποιήσουμε δικό μας κώδικα, αντί να δημιουργήσει το wizard δικό του κώδικα



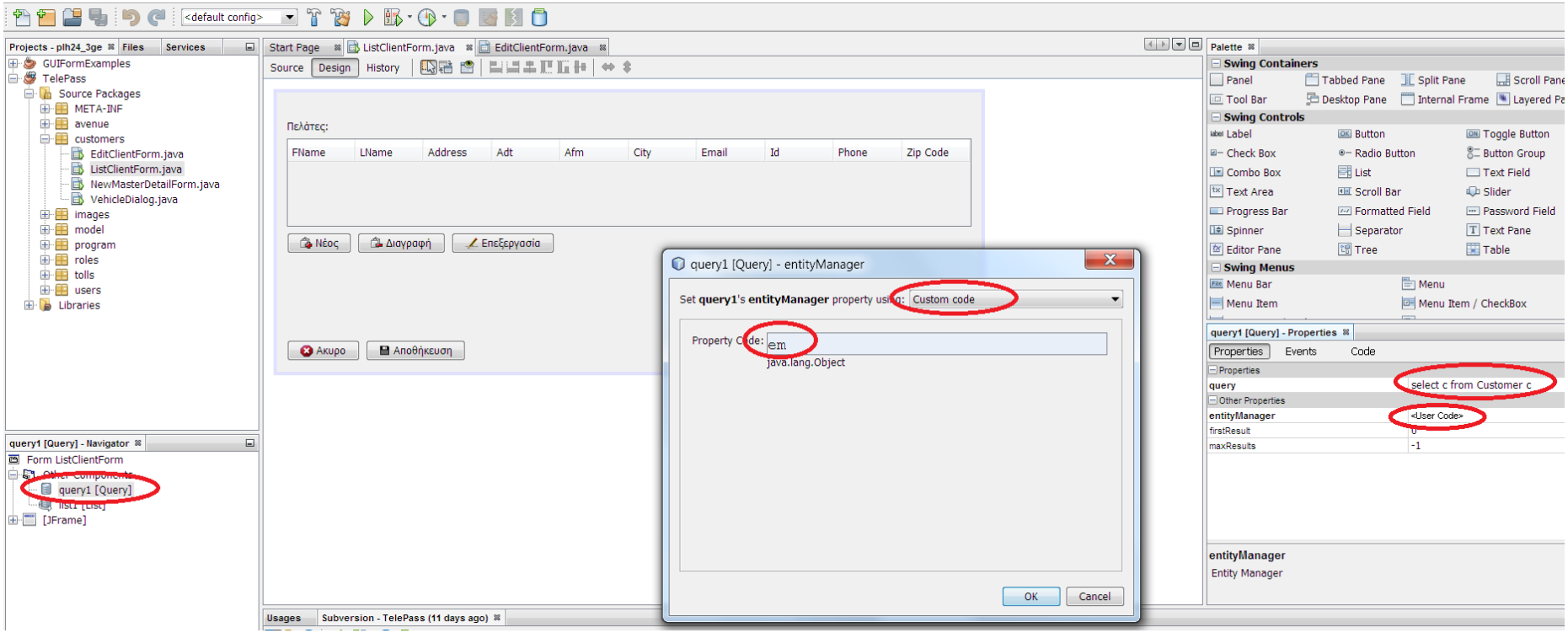
Query Component (Custom Code)

- Στο προηγούμενο παράδειγμα χρησιμοποιήσαμε custom code για το query component. Αυτό σημαίνει ότι στο πρόγραμμα θα γραφεί ακριβώς ο κώδικας που θα δώσουμε, χωρίς να χαθεί η δυνατότητα να μπορούμε να χρησιμοποιούμε το component στα wizards



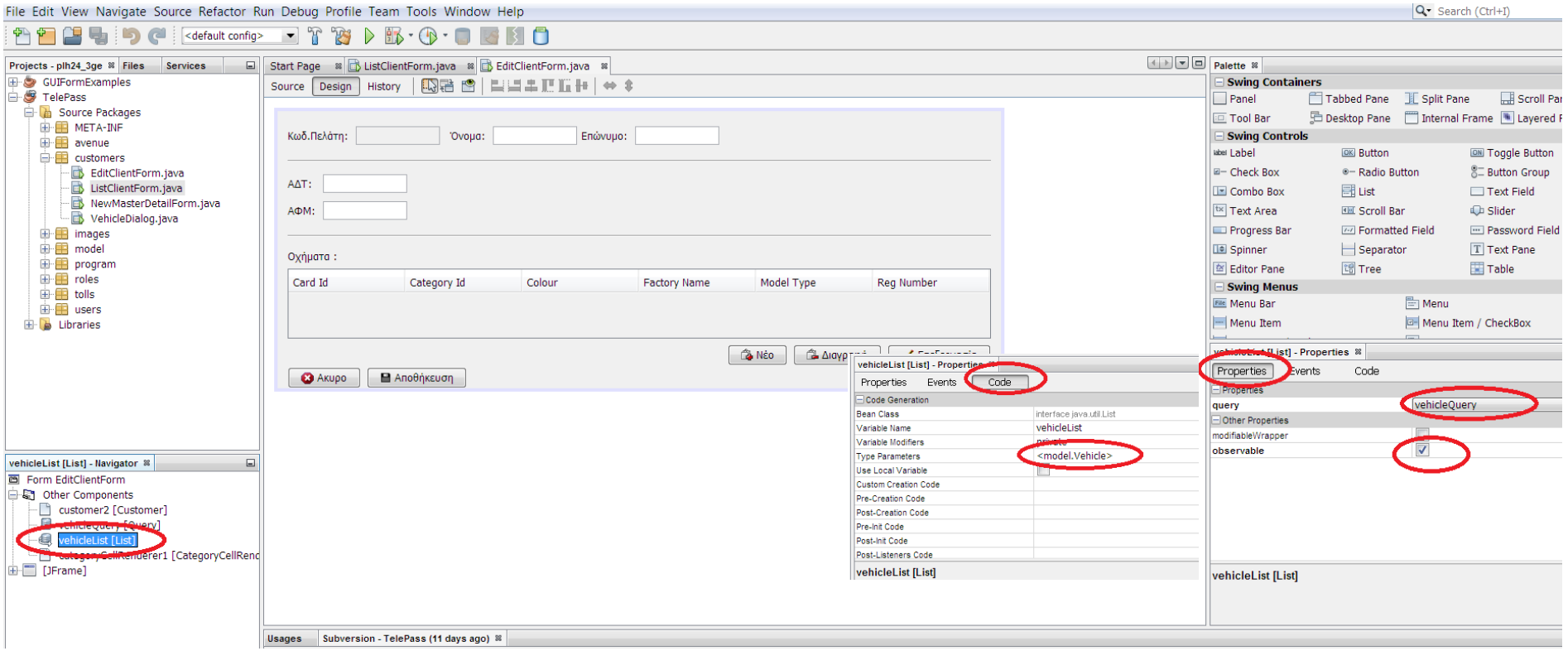
Query Component (2)

- Μπορούμε να χρησιμοποιήσουμε τα properties για να δηλώσουμε το query και τον entity manager
- Σε περίπτωση που ο entity manager δεν είναι component στον design navigator (και συνεπώς δεν μπορούμε να τον χρησιμοποιήσουμε στο wizard) τότε χρησιμοποιούμε custom code



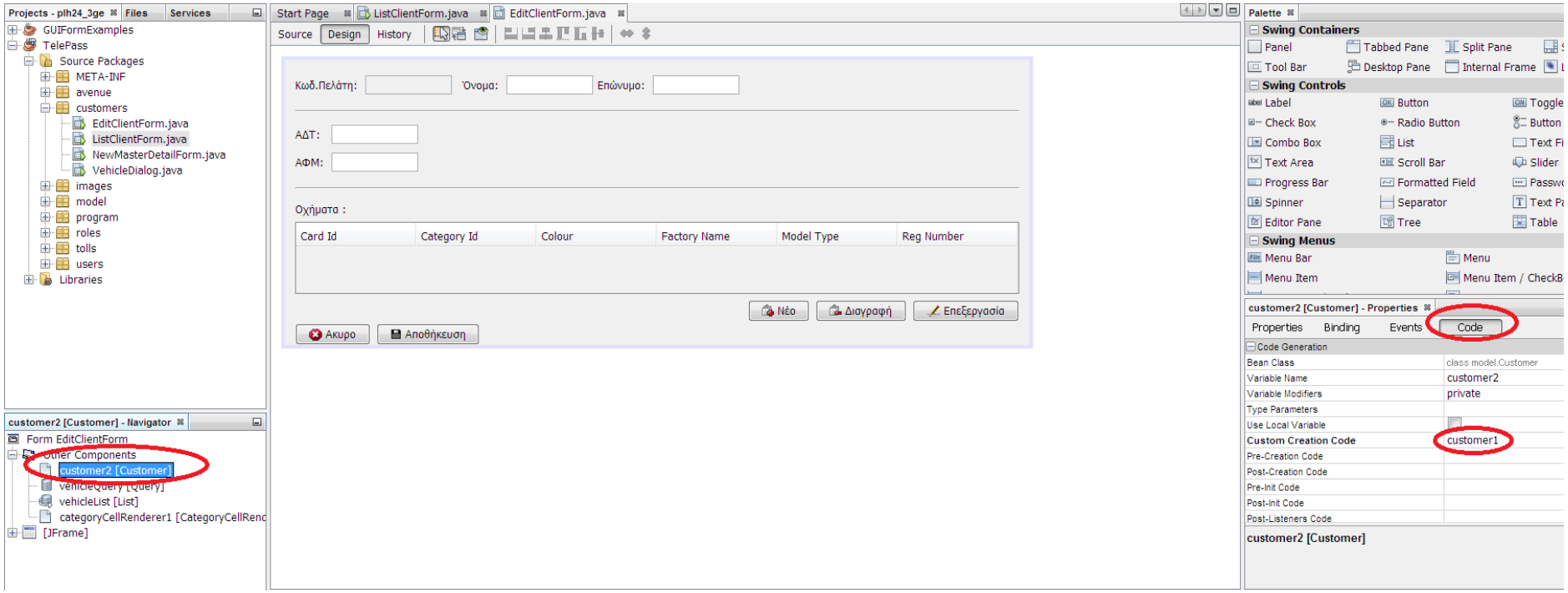
List Component

- Ορίζουμε το query component από το οποίο θα λάβει αποτελέσματα, καθώς και τον τύπο της λίστας
- Ορίζουμε επίσης ότι η λίστα θα είναι observable, που σημαίνει ότι θα είναι συγχρονισμένη με το GUI με το οποίο θα τη συνδέσουμε (bind)



Entity Component

- Εάν δεν θέλουμε να γίνει δημιουργία νέου αντικειμένου (new) αλλά να δείχνει σε ένα άλλο entity, τότε χρησιμοποιούμε custom code. Την τεχνική αυτή χρησιμοποιούμε όταν θέλουμε να δείχνει σε μεταβλητή που περνάμε στη φόρμα. (αν δεν το κάνουμε αυτό δεν μπορούμε να έχουμε ένα entity που να το περνάμε σαν μεταβλητή και να το χρησιμοποιούμε στα wizards).



«Πέρασμα» entity σε φόρμα

- Για να περάσουμε μια entity σαν παράμετρο σε μια φόρμα δημιουργούμε ένα constructor

```
public EditClientForm(Customer c, boolean readOnly) {
        em = DBManager.em;
        customer1=c;
        this.readOnly = readOnly;
        if ( !(em.getTransaction().isActive()) ){
            em.getTransaction().begin();
        }
        initComponents();  // εδώ θα γίνει το πέρασμα της
                           // customer1 στην customer2
}
```

Παραδείγματα Binding

Παραδείγματα Binding

- Text field
- Table
- Combo Box

Text Field Binding

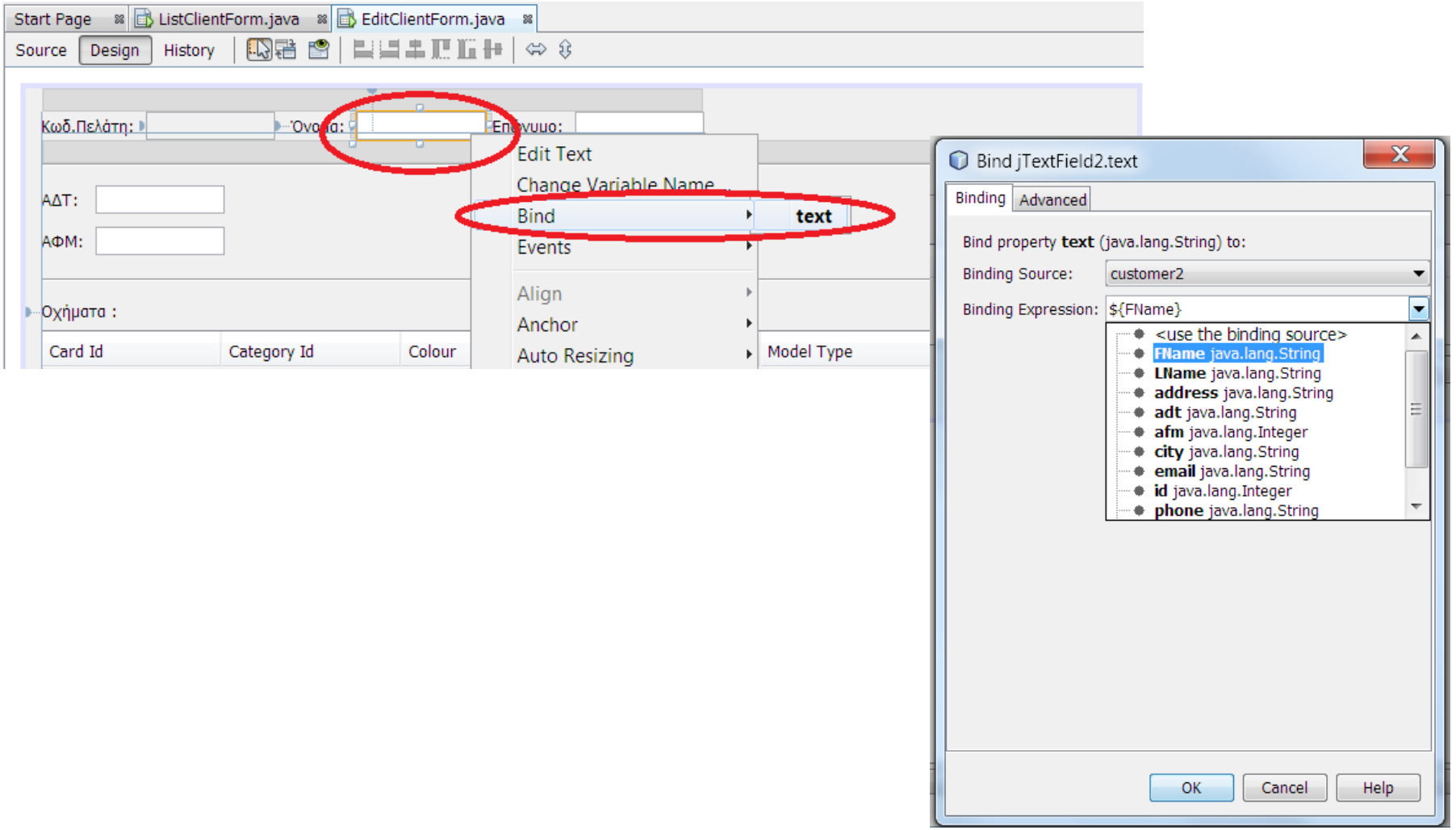


Table binding (1)

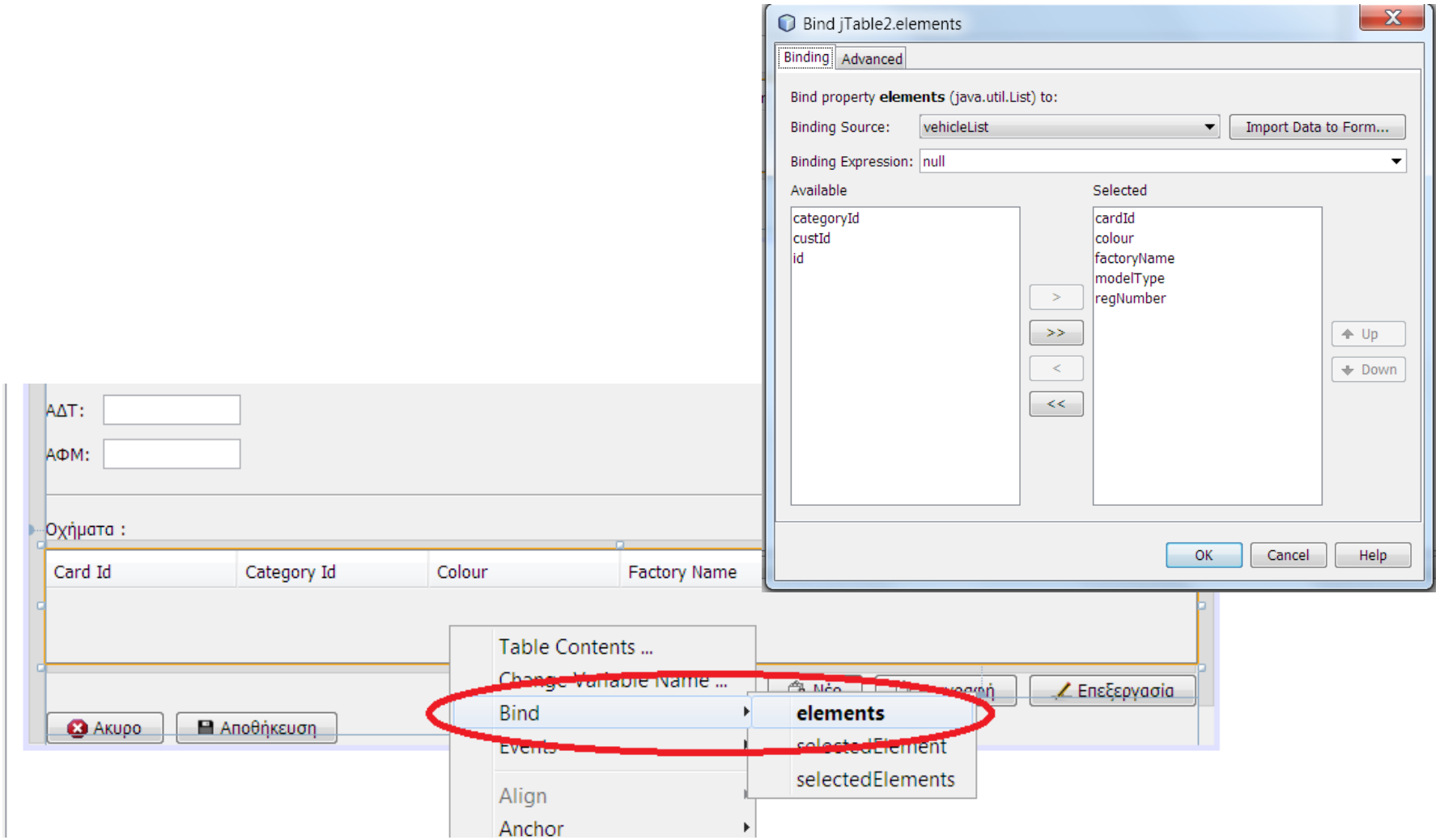
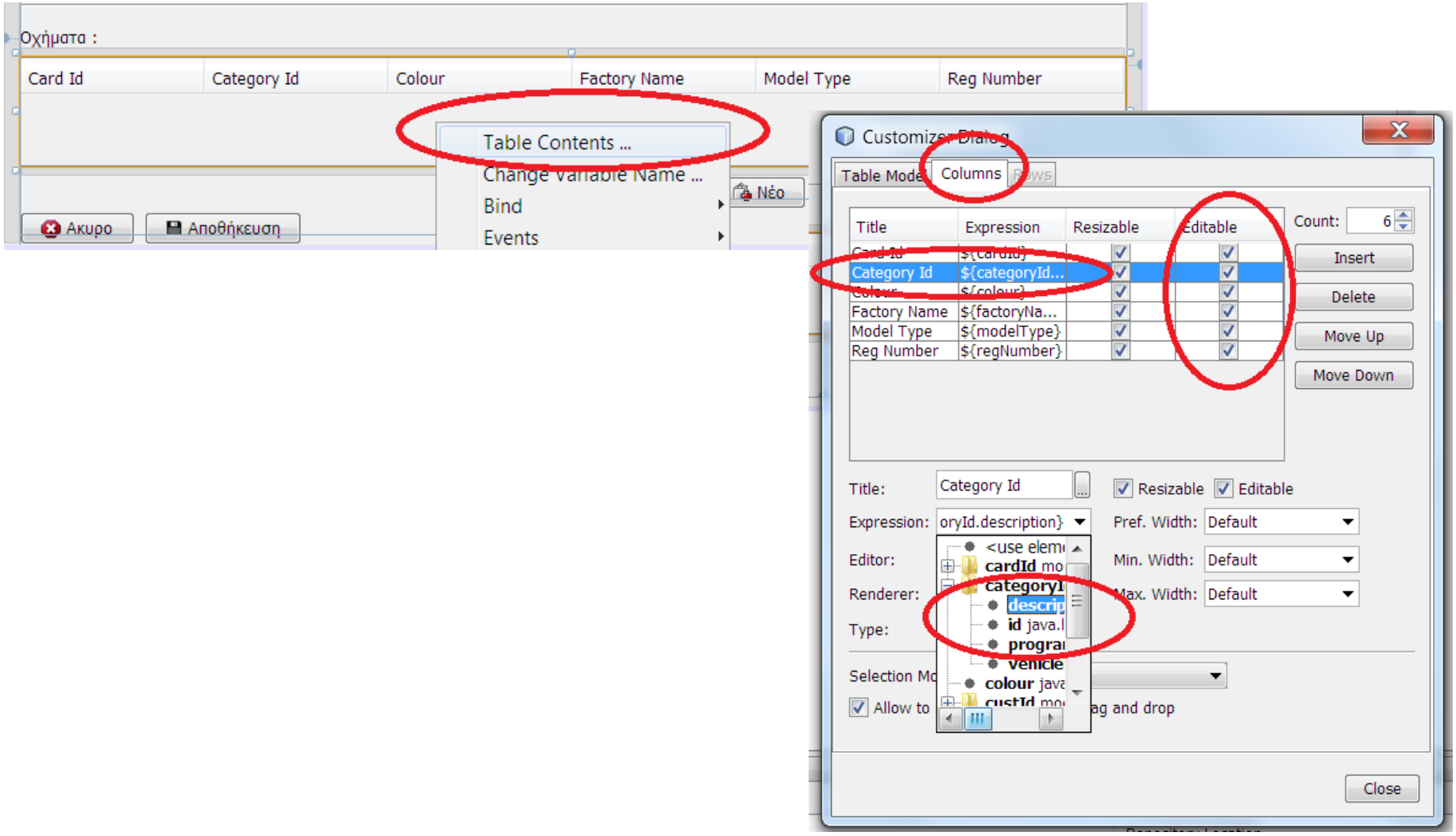
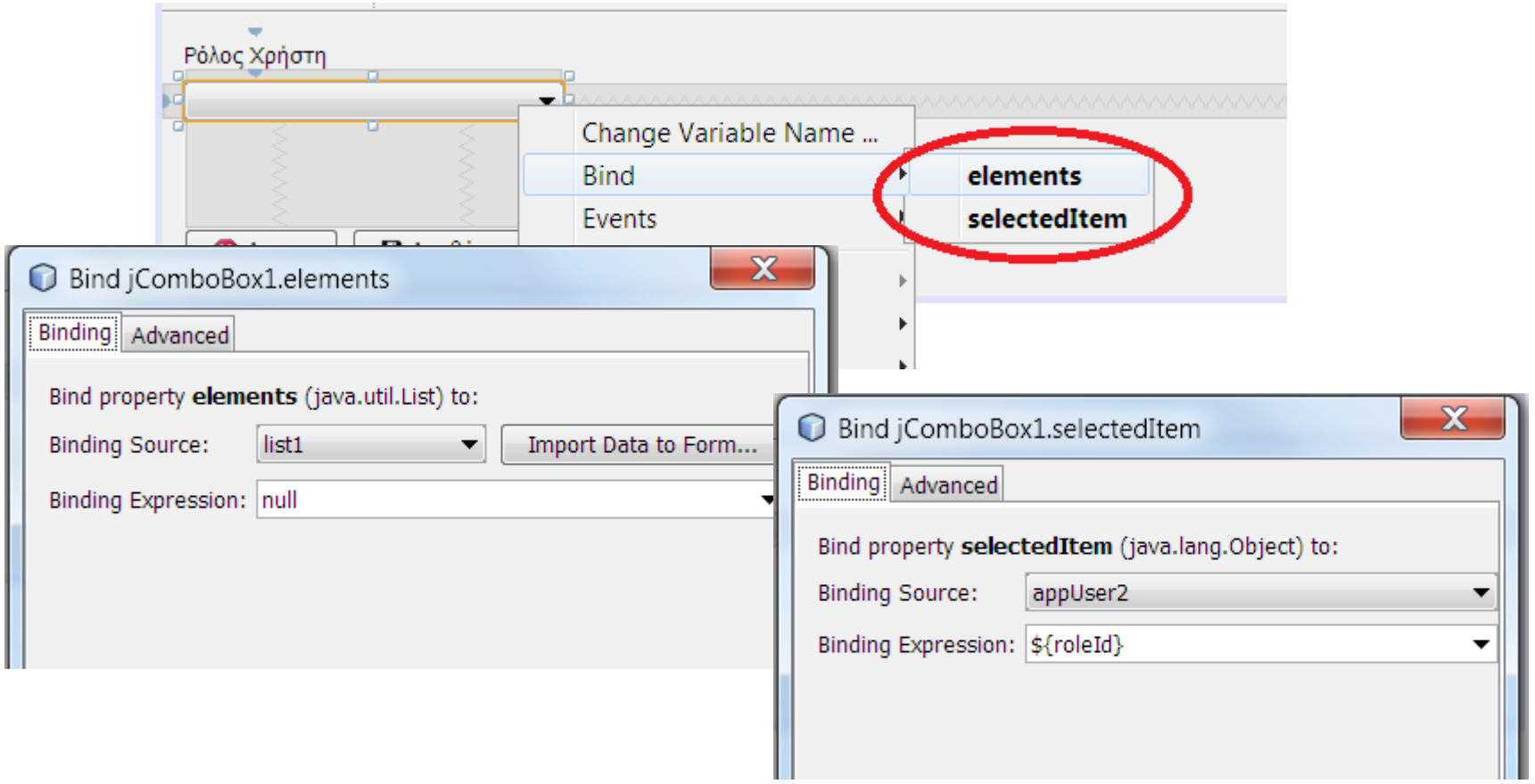


Table Binding (2)



ComboBox Binding

- Το elements δείχνει από ποια λίστα θα γεμίσει τις επιλογές
- Το selectedItem δείχνει σε ποιο component θα συνδεθεί η επιλογή που θα κάνουμε στη πτυσσόμενη λίστα



ComboBox Binding Renderer

- Για να δείξουμε ένα συγκεκριμένο πεδίο από τα αντικείμενα της λίστας θα πρέπει να δημιουργήσουμε έναν Renderer (μια κλάση που κληρονομεί την DefaultListCellRenderer). Αρκεί να αντικαταστήσουμε στον παρακάτω κώδικα τον τύπο των αντικειμένων της λίστας και το πεδίο που θέλουμε να δείξουμε.

```
/**
 *
 * @author Aggelos
 */
public class PeripheryListRenderer extends DefaultListCellRenderer{

        @Override
        public Component getListCellRendererComponent(
                JList list, Object value, int index, boolean isSelected, boolean cellHasFocus)
            super.getListCellRendererComponent(list, value, index, isSelected, cellHasFocus);
            if (value instanceof TblElectoralPeriphery) {
                TblElectoralPeriphery mec = (TblElectoralPeriphery)value;
                setText(mec.getFldName());
            }
            return this;
        }

}
```

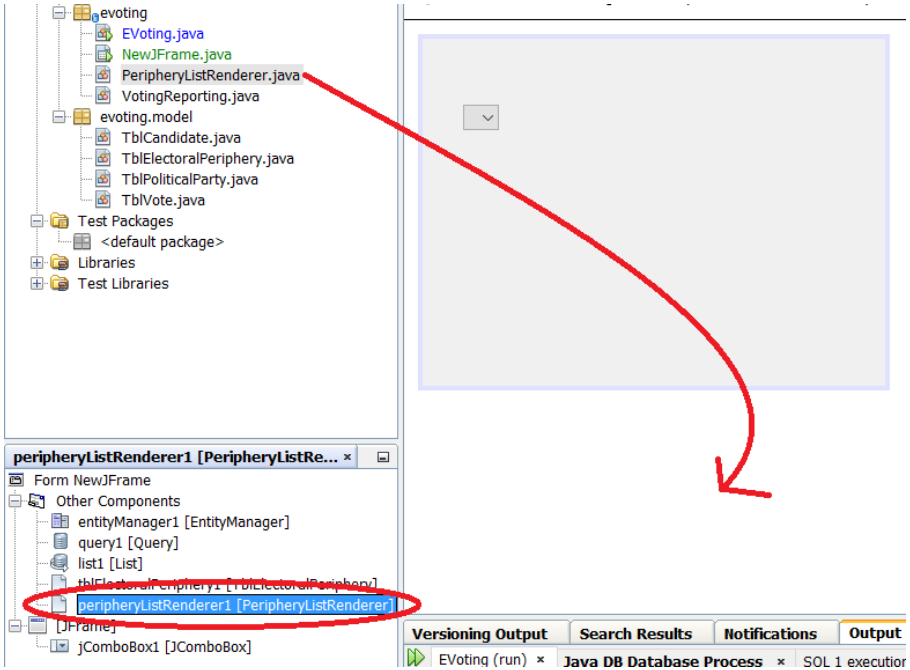
Ονομα του Renderer

Τύπος των αντικειμένων της λίστας που είναι συνδεδεμένο το ComboBox

Πεδίο του αντικειμένου που θέλουμε να φαίνεται στο ComboBox

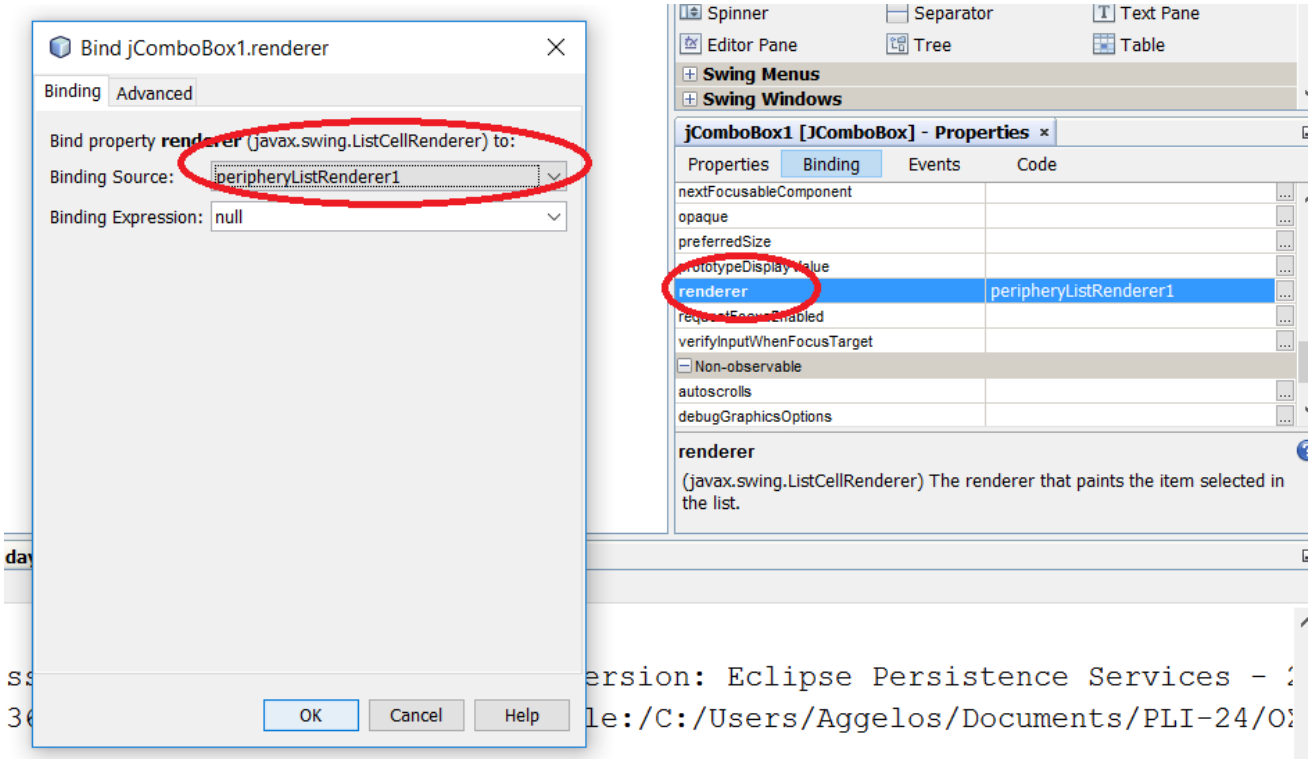
ComboBox set Renderer (1/2)

- Κάνουμε drag'n'drop τον renderer στη φόρμα



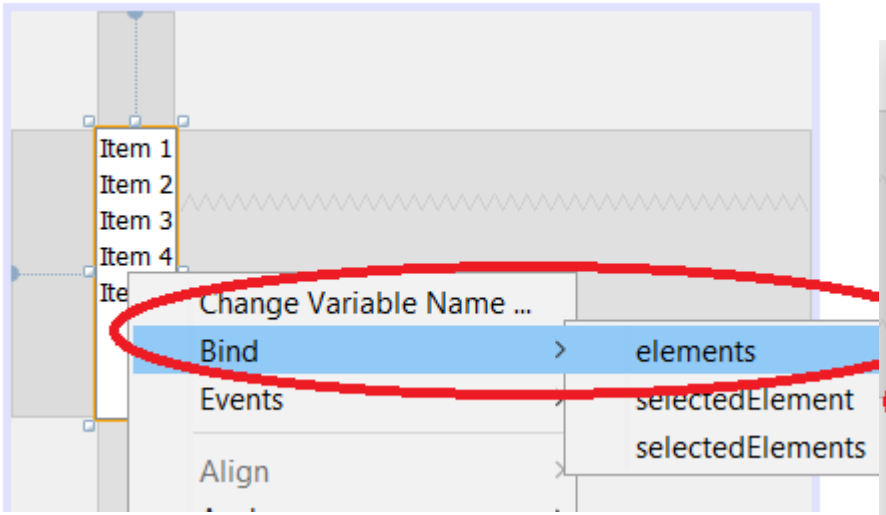
ComboBox set Renderer (2/2)

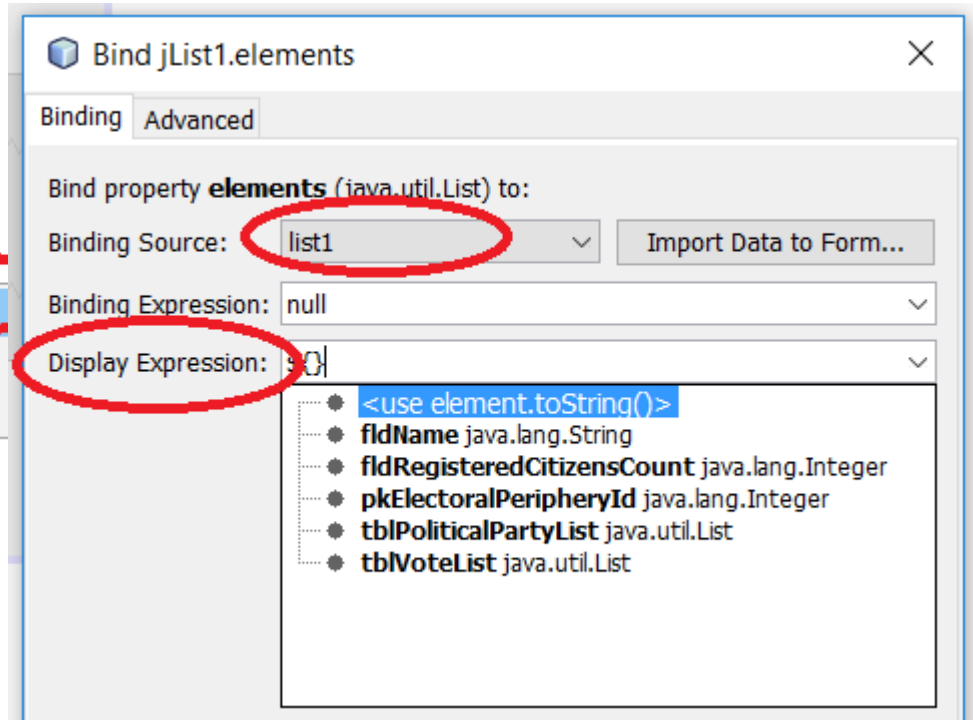
- Ορίζουμε ότι το ComboBox χρησιμοποιεί renderer



JList Binding 1/2

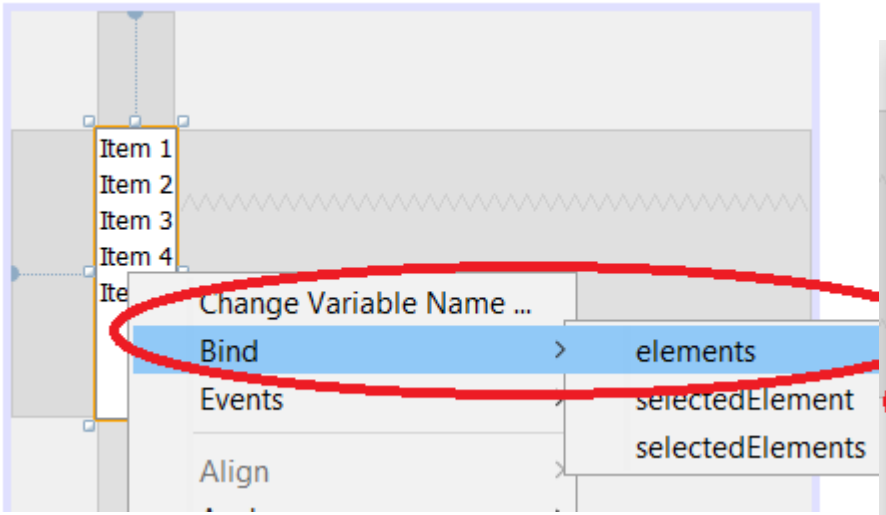
- Το elements δείχνει από ποια λίστα θα γεμίσει τις επιλογές
- Το selectedItem δείχνει σε ποιο component θα συνδεθεί η επιλογή που θα κάνουμε από τη λίστα
- Το selectedItem**s** δείχνει σε ποιο component θα συνδεθεί η λίστα των πολλαπλών επιλογών που μπορεί να κάνουμε από λίστα

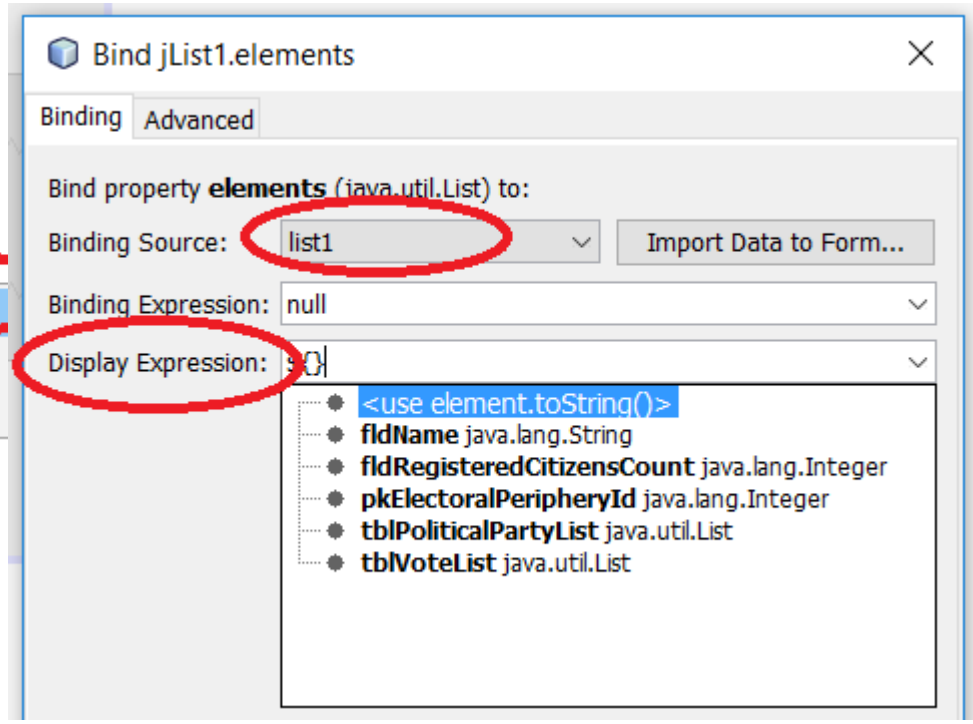




JList Binding 2/2

- Το display Expression επιλέγει ποιο πεδίο από τα αντικείμενα της λίστας θα εμφανίζεται (Δεν χρειάζεται να ορίσουμε renderer δηλαδή, μας δίνει τη δυνατότητα ο wizard να ορίσουμε το πεδίο)





Jlist/ComboBox Selected Item

- Για να ορίσουμε σε ποιο αντικείμενο θα καταχωρηθεί η εκάστοτε επιλογή μας από μια πτυσσόμενη (JComboBox) ή απλή λίστα (JList), θα πρέπει να το έχουμε κάνει drag'n'drop στη φόρμα