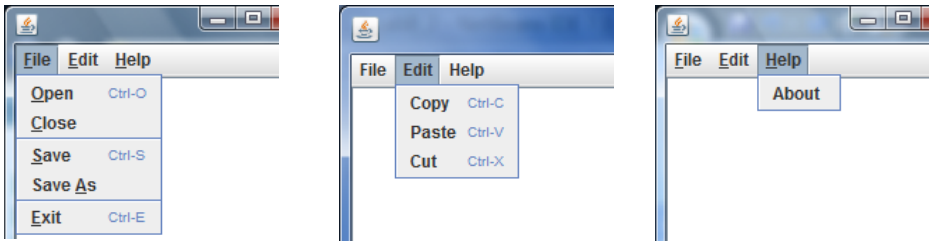
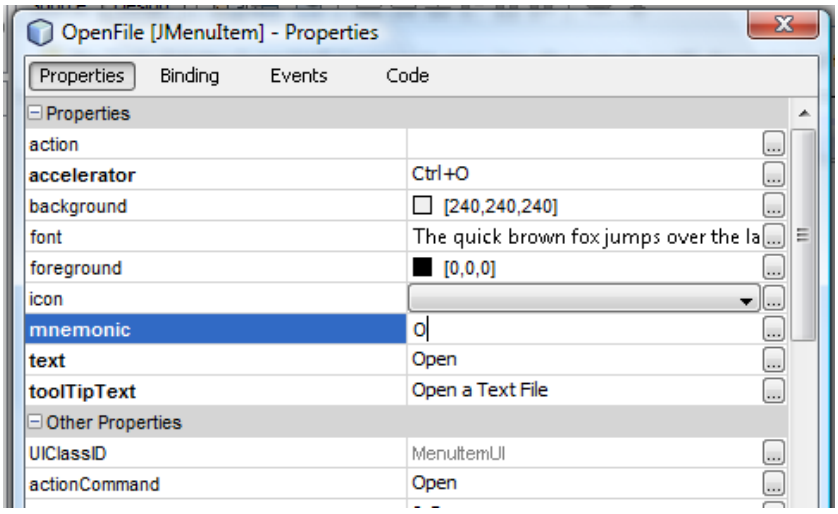
Εργαστήριο 8 - Άσκηση - Ανάλυση

Εκφώνηση: Δημιουργείστε την εφαρμογή «Simple Menu Editor» σε Java Swing με χρήση NetBeans. Στην εφαρμογή αυτή ο χρήστης θα μπορεί να φορτώνει, αποθηκεύει απλό αρχείο κειμένου και να εμφανίζει στοιχεία σχετικά με την εφαρμογή. Η εφαρμογή θα έχει την παρακάτω εμφάνιση σε επίπεδο μενού:

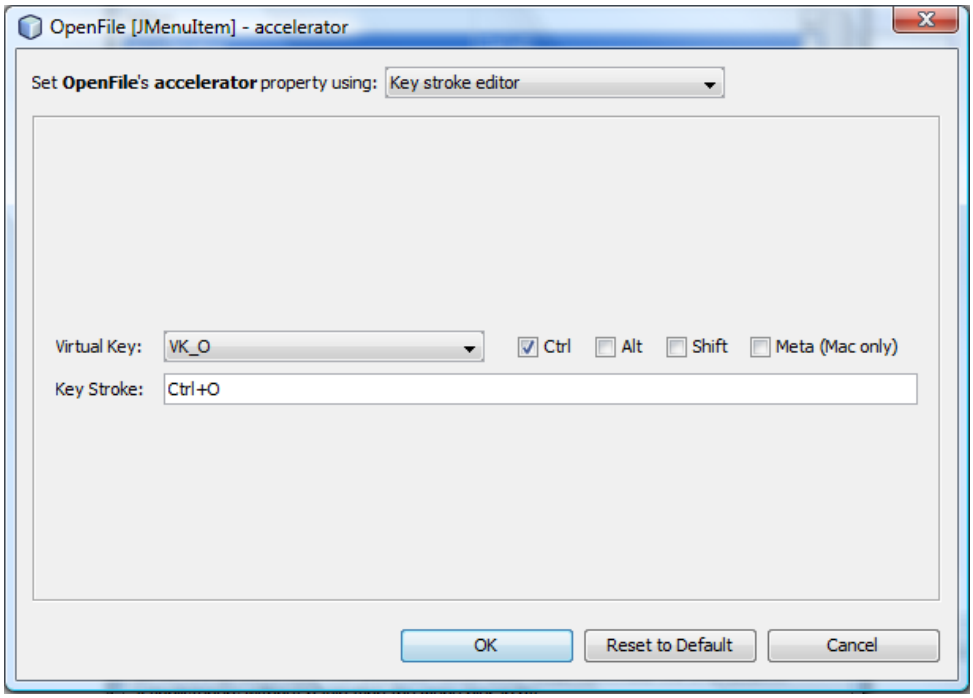


Ακολουθείστε τις παρακάτω οδηγίες:

1. Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame.
2. Πρώτα τοποθετήστε στο JFrame ένα συστατικό «Menu Bar», ένα «Text Area» το οποίο θα καταλαμβάνει σχεδόν όλο το JFrame και ένα JLabel στο κάτω μέρος του JFrame, το οποίο θα το χρησιμοποιήσουμε για να εμφανίζουμε τα μηνύματα της εφαρμογής.
3. Μετά θα πρέπει στο «Menu Bar» να κρέμονται τρία συστατικά «Menu», τα «File», «Edit», «Help».
4. Σε κάθε συστατικό «Menu» ορίστε τα κατάλληλα συστατικά «Menu Item» για να αποκτήσει η εφαρμογή σας όλο το σετ των μενού όπως φαίνεται στην παραπάνω εικόνα.
5. Από τις ιδιότητες (properties) των συστατικών μενού mnemonic, toolTipText και accelerator ορίστε τις κατάλληλες τιμές για να εξασφαλίζετε περισσότερη λειτουργικότητα στην εφαρμογή σας όπως φαίνεται παραπάνω. Δηλαδή, για το συστατικό «Open» του menu «File» φροντίστε τα παρακάτω:



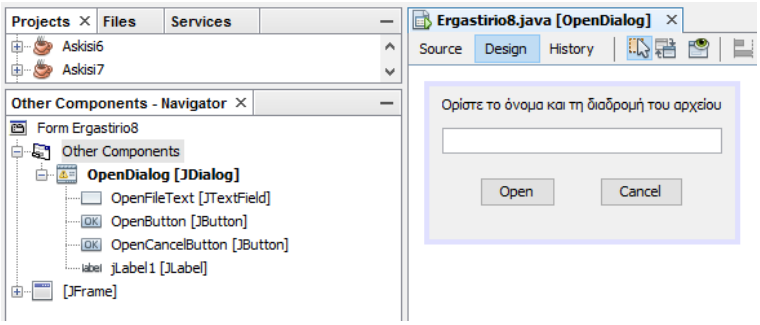
Εικόνα 1: Ορισμός mnemonic για το Open



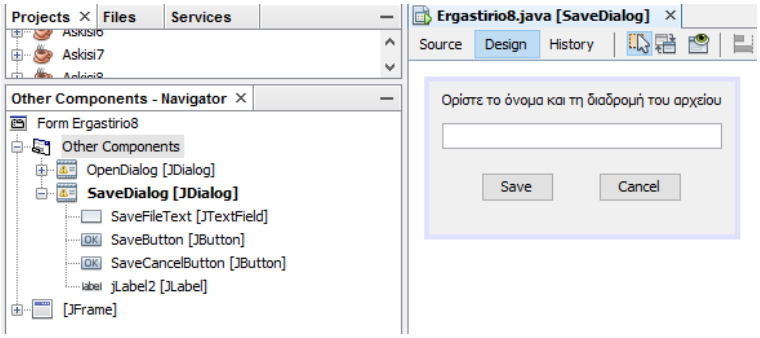
Εικόνα 2: Ορισμός accelerator για το Open

6. Δημιουργήστε τρία Swing Windows Dialog. Ένα για το άνοιγμα των αρχείων, ένα για την αποθήκευση και ένα για να εμφανίσετε τα στοιχεία σας. Σε αυτό θα βάλετε τα κατάλληλα συστατικά όπως φαίνεται στις παρακάτω τρεις εικόνες.

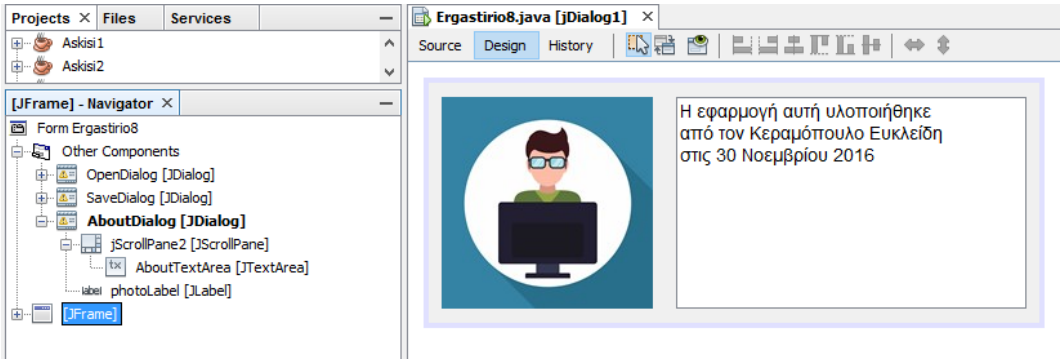
   Για τη λειτουργία Open



Για τη λειτουργία Save



Για τη λειτουργία About



7. Προγραμματίστε τις επιλογές του «menu» «File» ως εξής:
   - Με την «Open» να ανοίγει ένα text αρχείο.
   - Με την «Close» να αδειάζει το συστατικό Text Area.
   - Με την «Save» να αποθηκεύει το Text Area στο τρέχον αρχείο. Αν δεν υπάρχει τρέχον αρχείο η «Save» να λειτουργεί ως «Save As».
   - Με την «Save As» να ζητείτε όνομα και διαδρομή αρχείου για να αποθηκευθεί το περιεχόμενο του Text Area.
   - Με την «Exit» να κλείνει η εφαρμογή.
8. Προγραμματίστε τις επιλογές του «menu» «Edit».
9. Προγραμματίστε την επιλογή «About» του «menu» «Help» να εμφανίζει το AboutDialog.

Σημείωση:

- Ένα «Dialog» εμφανίζετε με χρήση της μεθόδου: DialogName.setVisible(true);
- Το μέγεθος του ορίζεται με χρήση της μεθόδου: DialogName.setSize(250, 130);
- Και εξαφανίζεται με χρήση της μεθόδου: DialogName.setVisible(false);

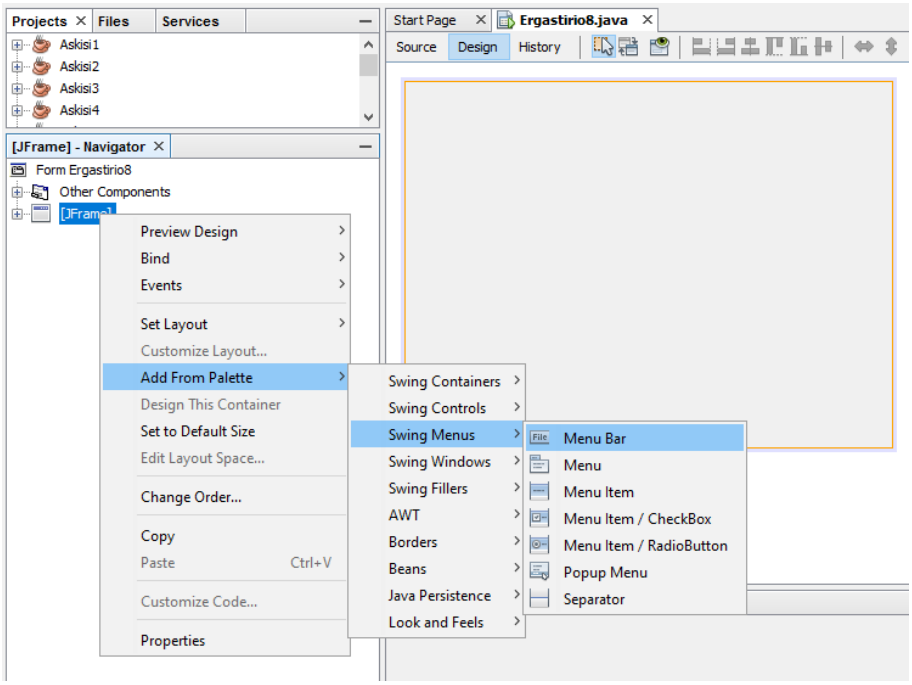
Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi10.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio8.

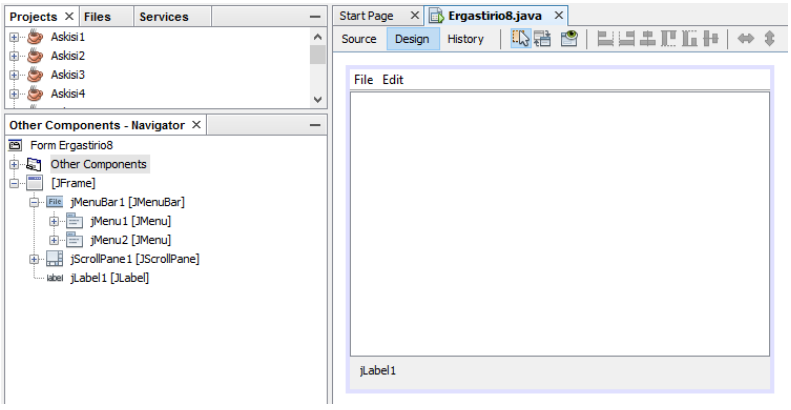
Βήμα 3: Ορίζουμε ως τίτλο του JFrame το «Simple Menu Editor».

Βήμα 4: Προσθέτουμε στο JFrame ένα Menu Bar, όπως φαίνεται στην Εικόνα 3.



Εικόνα 3: Πρόσθεση Menu Bar στο JFrame

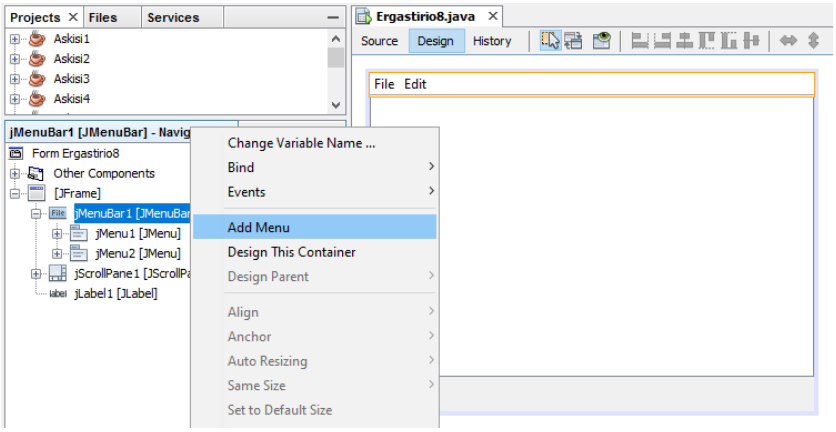
Βήμα 5: Επίσης, προσθέτουμε στο JFrame ένα JTextArea και ένα JLabel. Το JFrame διαμορφώνεται όπως φαίνεται στην Εικόνα 4.



Εικόνα 4: Ο Καμβάς με menu, TextArea και ένα Label για εμφάνιση μηνυμάτων

Διαγραφούμε το κείμενο του JLabel, ορίζουμε ως όνομα μεταβλητής του το «message» και ορίζουμε HorizontalSize 400 και VerticalSize 20.

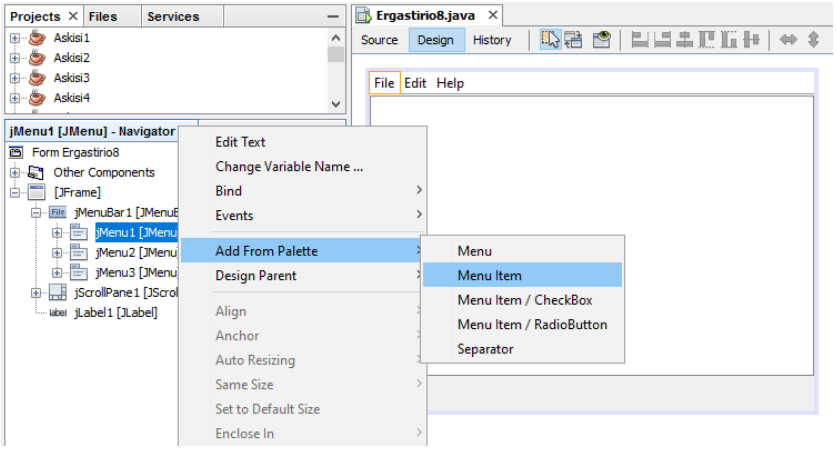
Βήμα 6: Παρατηρούμε ότι τα μενού File και Edit προστίθενται αυτόματα με τη δημιουργία του MenuBar. Για να προσθέσουμε στο MenuBar και το Help επιλέγουμε το «Add Menu» από το pop-up menu που ανοίγει με δεξί click πάνω στο jMenuBar1 που βρίσκεται στο Navigator. Στη συνέχεια, αφού προστεθεί το νέο στοιχείο (jMenu3) στο MenuBar αλλάζουμε την ιδιότητα του Text (με Edit Text) να έχει την τιμή Help.



Εικόνα 5: Πρόσθεση στοιχείου στο MenuBar

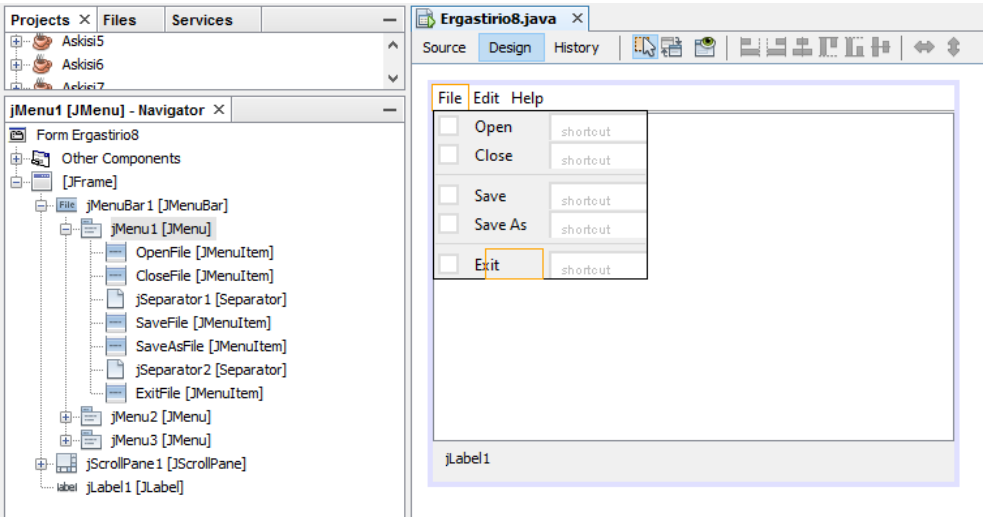
Βήμα 7: Για να προσθέσουμε σε κάθε menu στο menuBar μία εντολή, επιλέγουμε την κατάλληλη επιλογή από το μενού «Add From Palette -> Menu Item» (Εικόνα 6). Οι επιλογές που έχουμε είναι πέντε. Αναλυτικά:

- Menu: προσθέτει υπομενού.
- MenuItem: προσθέτει JButton.
- MenuItem/CheckBox: προσθέτει JCheckBox.
- MenuItem/RadioButton: προσθέτει JRadioButton.
- Separator: προσθέτει μία διαχωριστική γραμμή στο μενού.

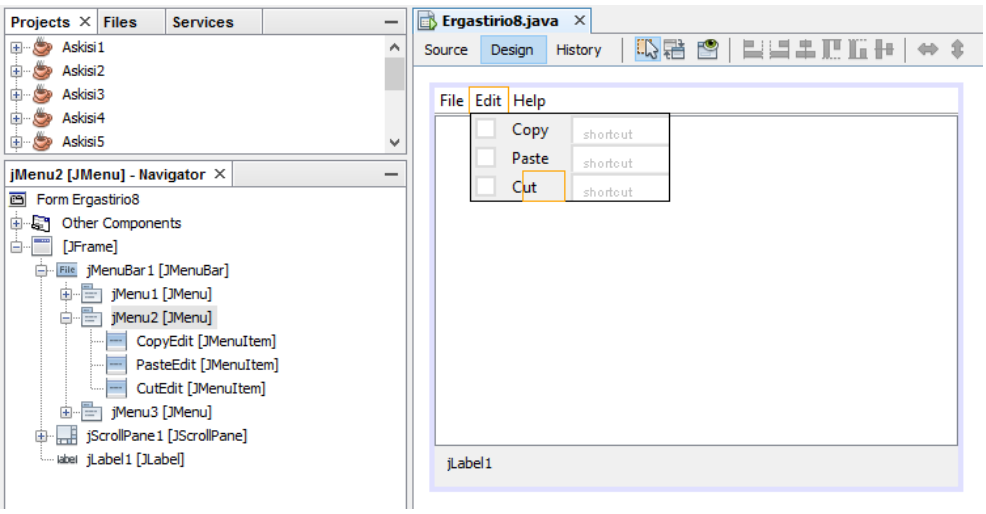


Εικόνα 6: Πρόσθεση στοιχείων στα menu του menuBar

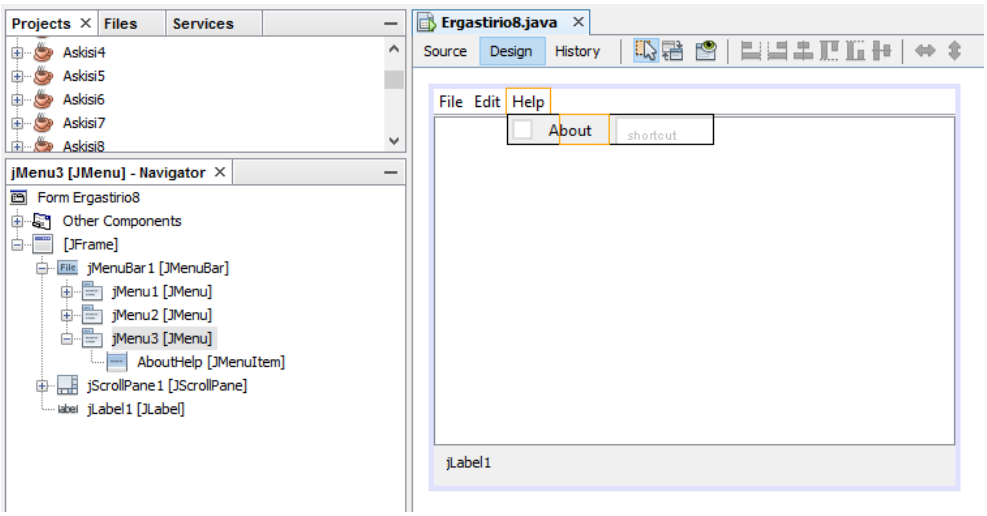
Έτσι, στην Εικόνα 7 παρουσιάζονται οι τιμές των μεταβλητών και του κειμένου των συστατικών του menu File, στην Εικόνα 8 παρουσιάζονται οι τιμές των μεταβλητών και του κειμένου των συστατικών του menu Edit και στην Εικόνα 9 παρουσιάζονται οι τιμές των μεταβλητών και του κειμένου των συστατικών του menu Help.



Εικόνα 7: Το μενού File



Εικόνα 8: Το μενού Edit

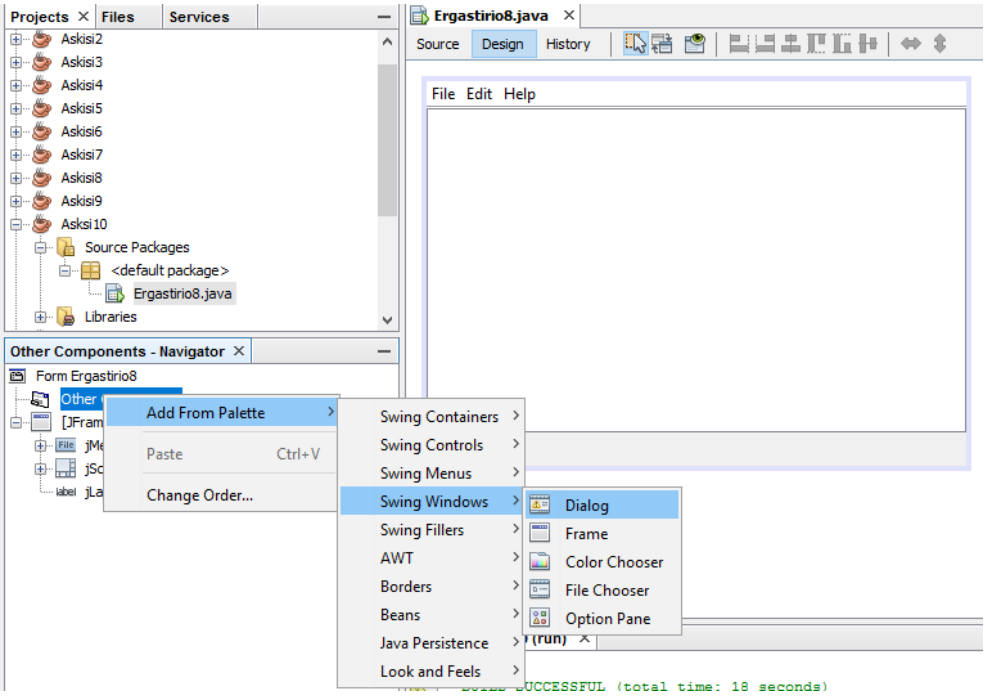


Εικόνα 9: Το μενού Help

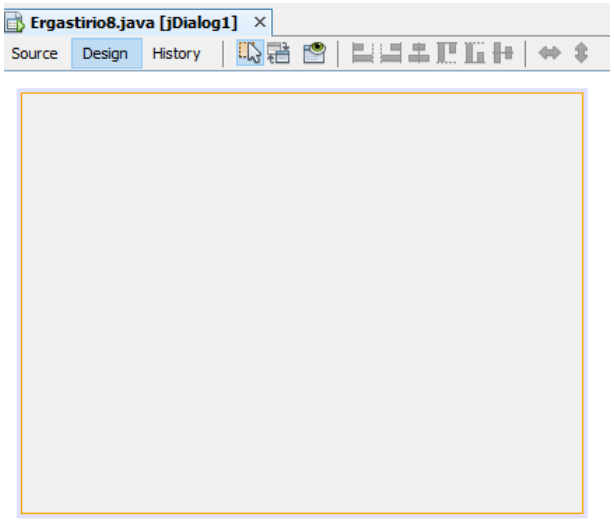
Βήμα 8: Σε αυτό το βήμα θα ορίσουμε το κείμενο (edit text), το όνομα μεταβλητής (Change variable name), το mnemonic, το accelerator και το tooltiptext όλων των συστατικών των μενού. Αναλυτικά :

- File (jMenu1): mnemonic – F
- Open (OpenFile): mnemonic – O, accelerator – CTRL+O, tooltip – «Open a Text File».
- Close (CloseFile): mnemonic – C, accelerator – κανένα, tooltip – «Close the file».
- Save (SaveFile): mnemonic – S, accelerator – CTRL+S, tooltip – «Save the File».
- Save As (SaveAsFile): mnemonic – A, accelerator – κανένα, tooltip – «Save file with a new name» και displayedMnemonicIndex – 5 για να υπογραμμίσει το $2^{ο}$ A στο «Save As».
- Exit (ExitFile): mnemonic – E, accelerator – CTRL+E, tooltip – «Close the application».
- Edit (jMenu2): mnemonic – E
- Copy (CopyEdit): mnemonic – C, accelerator – CTRL+C, tooltip – «Copy to clipboard».
- Paste (PasteEdit): mnemonic – P, accelerator – CTRL+P, tooltip – «Paste text from clipboard».
- Cut (CutEdit): mnemonic – U, accelerator – CTRL+X, tooltip – «Cut to clipboard».
- Help (jMenu3): mnemonic – H
- About (AboutHelp): mnemonic – A, accelerator – κανένα, tooltip – «Info about the developer».

Βήμα 9: Σε αυτό το βήμα θα δημιουργήσουμε τρία dialog window τα οποία αρχικά θα είναι αόρατα. Για να είναι αόρατα θα πρέπει να τα τοποθετήσουμε στα «Other Components». Έτσι, για να δημιουργήσουμε ένα dialog window κάνουμε δεξί click στα «Other Components» και επιλέγουμε «Add From Palette -> Swing Window -> Dialog», όπως φαίνεται στην Εικόνα 10. Για να διαμορφώσουμε ένα dialog window κάνουμε διπλό click πάνω στο dialog window, όπως «κρέμεται» κάτω από το other components. Έτσι, στον καμβά παρουσιάζεται μόνο το dialog window (Εικόνα 11).

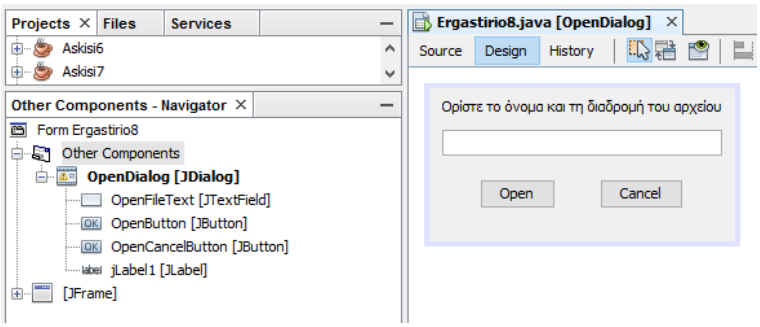


Εικόνα 10: Δημιουργία Dialog Window



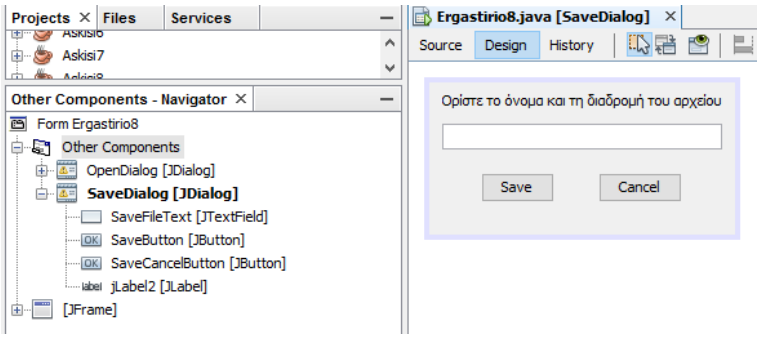
Εικόνα 11: Dialog Window

Στη συνέχεια το διαμορφώνουμε ακολουθώντας τις οδηγίες της εκφώνησης. Έτσι, το πρώτο dialog window το ονομάζουμε (τη μεταβλητή του) OpenDialog (Εικόνα 12) με τίτλο παραθύρου «Open File». Τοποθετούμε σε αυτό ένα JLabel με κείμενο «Ορίστε το όνομα και τη διαδρομή του αρχείου», ένα JTextField (με όνομα μεταβλητής OpenFileText) και δύο JButton. Στο πρώτο JButton ορίζουμε ως όνομα μεταβλητής το «OpenButton» και κείμενο το «Open», ενώ στο δεύτερο ορίζουμε ως όνομα μεταβλητής το «OpenCancelButton» και κείμενο το «Cancel».



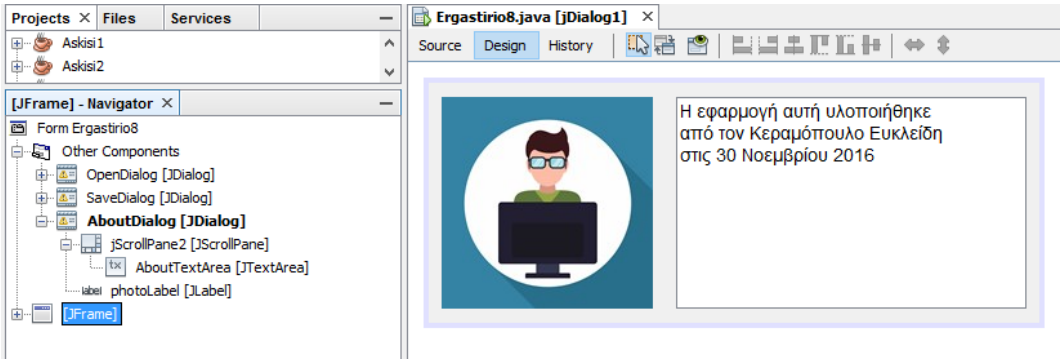
Εικόνα 12: Το OpenDialog window

Παρόμοια, το δεύτερο dialog window το ονομάζουμε SaveDialog (Εικόνα 13) με τίτλο παραθύρου «Save File». Τοποθετούμε σε αυτό ένα JLabel με κείμενο «Ορίστε το όνομα και τη διαδρομή του αρχείου», ένα JTextField (με όνομα μεταβλητής SaveFileText) και δύο JButton. Στο πρώτο JButton ορίζουμε ως όνομα μεταβλητής το «SaveButton» και κείμενο το «Save», ενώ στο δεύτερο ορίζουμε ως όνομα μεταβλητής το «SaveCancelButton» και κείμενο το «Cancel».



Εικόνα 13: Το SaveDialog window

Στο τρίτο dialog window (Εικόνα 14) ονομάζουμε τη μεταβλητή AboutDialog με τίτλο παραθύρου «Info Details» και τοποθετούμε ένα JLabel και ένα JTextArea. Στο JLabel, με όνομα μεταβλητής «photoLabel» ορίζουμε κατάλληλη εικόνα με χρήση της ιδιότητας «icon». Στο JTextArea, με όνομα μεταβλητής «AboutTextArea», ορίζουμε το κατάλληλο κείμενο.

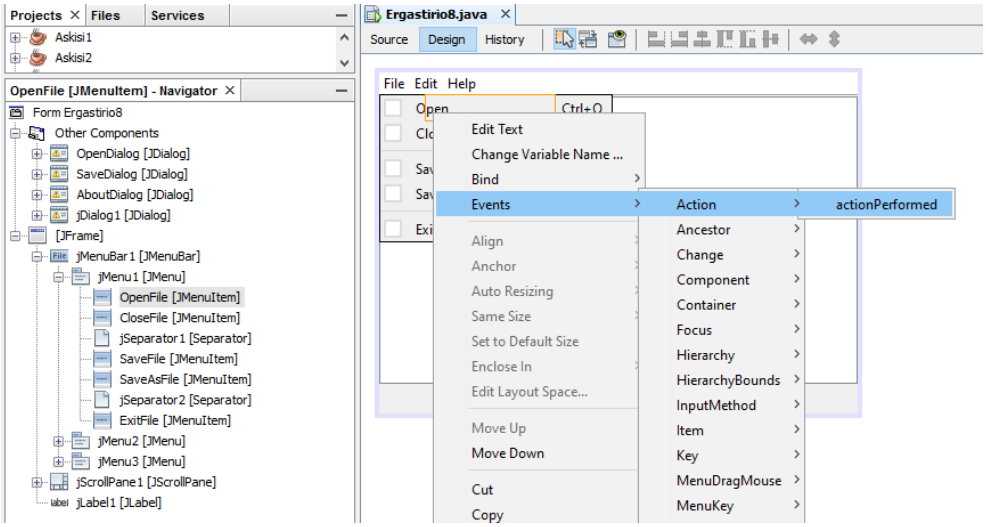


Εικόνα 14: Το AboutDialog window

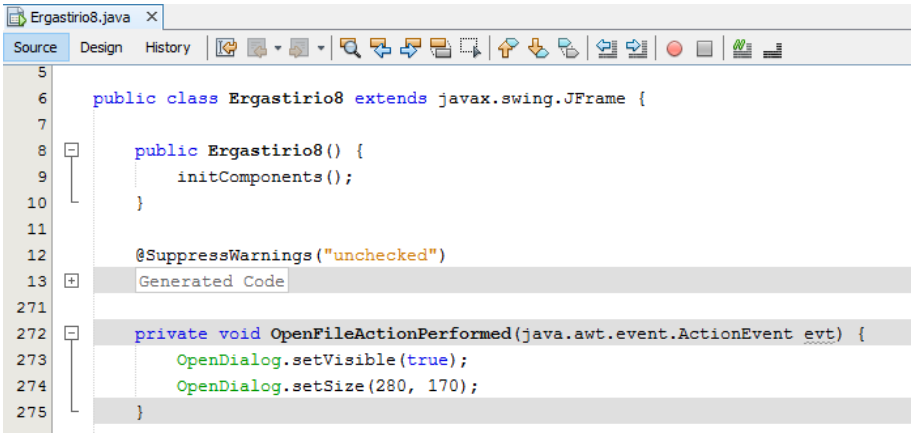
Στη συνέχεια θα προγραμματίσουμε τα γεγονότα του μενού και των συστατικών των dialog window.

Βήμα 10: Θα ξεκινήσουμε με τον προγραμματισμό του ανοίγματος ενός αρχείου. Αυτό θα γίνει σε δύο φάσεις. Πρώτα από το μενού File->Open θα ανοίξουμε το OpenDialog και στη συνέχεια από το OpenDialog θα ανοίξουμε το αρχείο και θα εμφανίσουμε το περιεχόμενο του στο JTextArea.

Για να υλοποιήσουμε την πρώτη φάση, θα προγραμματίσουμε το γεγονός ActionPerformed του OpenFile. Στην Εικόνα 15, φαίνεται πως θα δημιουργήσουμε το γεγονός ActionPerformed στο OpenFile. Στην Εικόνα 16, φαίνεται ο κώδικας που θα χρησιμοποιήσουμε για να εμφανίζουμε το OpenDialog. Η εντολή της γραμμής 273, εμφανίζει το OpenDialog. Το OpenDialog είναι αόρατο, αφού έχει τοποθετηθεί στο «Other Components». Η εντολή της γραμμής 274, ορίζει το μέγεθος του OpenDialog όταν εμφανιστεί.



Εικόνα 15: Δημιουργία γεγονότος ActionPerformed για το OpenFile



Εικόνα 16: Ο κώδικας του OpenFile

Στη δεύτερη φάση θα προγραμματίσουμε τα συστατικά του OpenDialog. Έτσι, όταν ο χρήστης θα επιλέγει το κουμπί Open θα ανοίγει στο JTextArea το περιεχόμενο του αρχείου κειμένου, που το όνομα του αρχείου έχει δηλωθεί στο JTextField OpenFileText του OpenDialog. Ενώ, όταν ο χρήστης θα επιλέγει το κουμπί Cancel θα γίνεται αόρατο το OpenDialog χωρίς να συμβεί κάτι άλλο.

Στην Εικόνα 17, παρουσιάζεται ο κώδικας των ActionPerformed γεγονότων των δύο JButton του OpenDialog. Αναλυτικά για το OpenButton:

Γραμμή 278: Γίνεται η δήλωση ενός αντικειμένου FileReader για την ανάγνωση του αρχείου. Θα χρειαστεί και η εισαγωγή της βιβλιοθήκης java.io.FileReader.

Γραμμή 279: Στην global αλφαριθμητική μεταβλητή filename (δηλώθηκε στη γραμμή 333) αποθηκεύουμε το περιεχόμενο του TextField OpenFileText, στο οποίο ο χρήστης δηλώνει το αρχείο (και τη διαδρομή του φακέλου που βρίσκεται) που θέλει να ανοίξει.

Γραμμή 280: Γίνεται έλεγχος αν το OpenFileText ήταν κενό όταν πατήθηκε το κουμπί OpenButton.

Γραμμή 281-282: Σε αυτή την περίπτωση εμφανίζεται κατάλληλο μήνυμα στην ετικέτα message.

Γραμμή 283: Γίνεται αόρατο το OpenDialog.

Γραμμή 284: Επιστρέφει από τη μέθοδο.

Γραμμή 285: Αν το OpenFileText δεν είναι κενό.

Γραμμή 286: Ανοίγει η try-catch

Γραμμή 287: Δημιουργείται το αντικείμενο fr με βάση το όνομα αρχείου που δηλώθηκε.

Γραμμή 288: Εμφανίζει στο JTextArea το περιεχόμενο του αρχείου.

Γραμμή 289: Κλείνει το αντικείμενο fr.

Γραμμή 290: Γίνεται αόρατο το OpenDialog.

Γραμμή 291: Ορίζεται ως τίτλος που JFrame το όνομα του αρχείου.

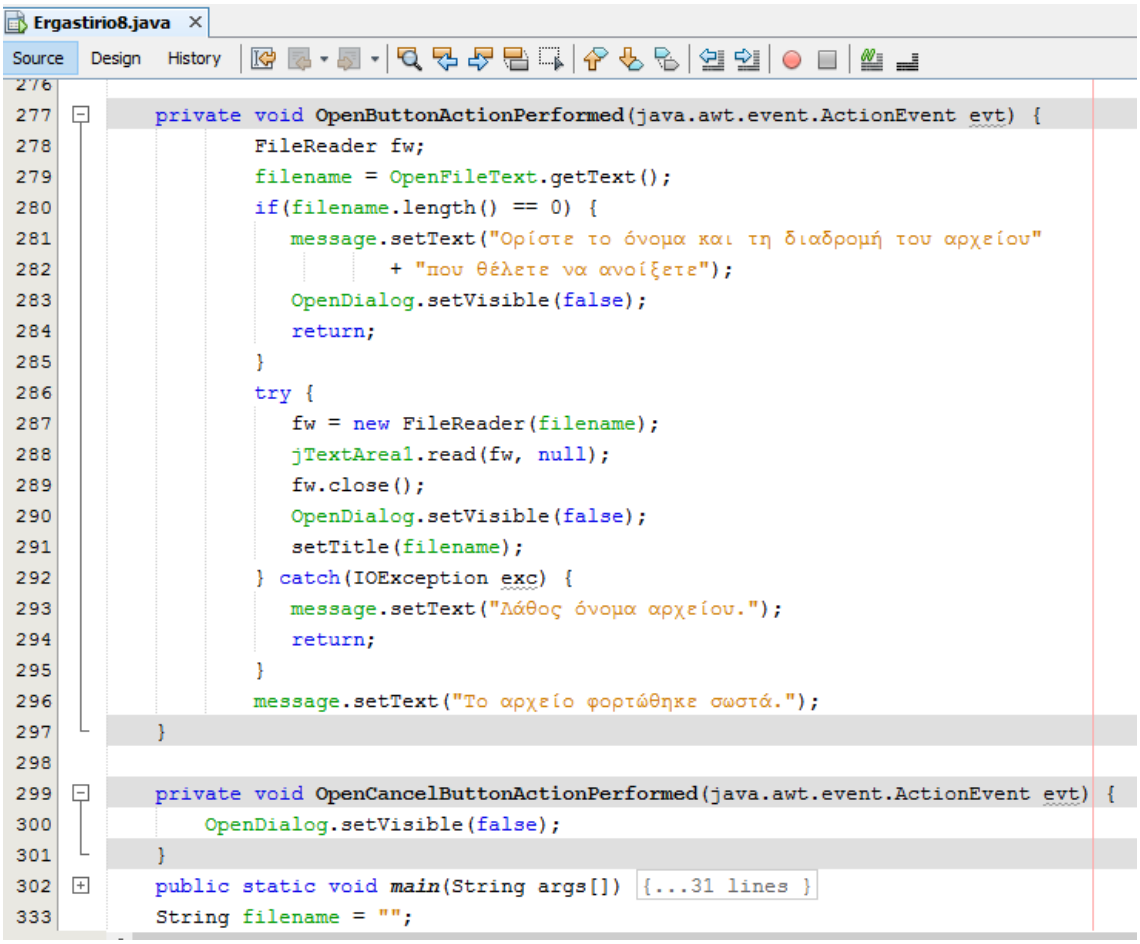
Γραμμή 292: Η catch ορίζεται με βάση το IOException. Θα χρειαστεί και η εισαγωγή της βιβλιοθήκης java.io.IOException.

Γραμμή 293: Εμφανίζεται κατάλληλο μήνυμα στην ετικέτα message.

Γραμμή 294: Επιστρέφει από τη μέθοδο.

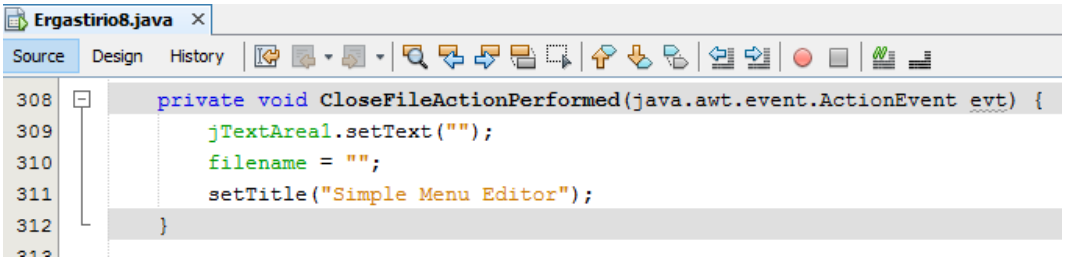
Γραμμή 296: Εμφανίζεται κατάλληλο μήνυμα στην ετικέτα message αν όλα πάνε καλά.

Στη γραμμή 300 είναι ο κώδικας του OpenCancelButton όπου γίνεται αόρατο το OpenDialog.



Εικόνα 17: Ο κώδικας των JButton του OpenDIalog

Βήμα 11: Στην Εικόνα 18, φαίνεται ο κώδικας του File->Close. Με την συγκεκριμένη επιλογή αδειάζει το περιεχόμενο του TextArea (γραμμή 309), γίνεται αρχικοποίηση στη μεταβλητή που αποθηκεύεται (γραμμή 310) και αλλάζει ο τίτλος του JFrame στον αρχικό (γραμμή 311).



Εικόνα 18: Ο κώδικας του CloseFile

Βήμα 12: Σε αυτό το βήμα θα υλοποιήσουμε το «Save As» και μετά το «Save», όπου θα χρησιμοποιήσουμε το «Save As».

Το «Save As» θα γίνει παρόμοια με το «Open» που αναλύθηκε στο βήμα 10. Έτσι, πρώτα δημιουργούμε το γεγονός ActionPerformed του SaveAsFile. Στην Εικόνα 19, στις γραμμές 326-327 φαίνεται ο κώδικας που θα χρησιμοποιήσουμε για να εμφανίσουμε το SaveDialog.

Στις γραμμές 331-349, φαίνεται ο κώδικας του γεγονότος ActionPerformed του κουμπιού Save του SaveDialog. Ο κώδικας έχει την ίδια λογική με τον κώδικα που υλοποιήσαμε για το OpenButton του OpenDialog, με τις απαραίτητες τροποποιήσεις. Οι διαφορές σε σχέση με τον κώδικα του OpenButton είναι οι παρακάτω:

Γραμμή 331: Η δήλωση του αντικειμένου για την αποθήκευση του αρχείου θα γίνει με βάση την τάξη FileWriter. Θα χρειαστεί και η εισαγωγή της βιβλιοθήκης java.io.FileWriter.

Γραμμή 332: Χρησιμοποιείται το SaveFileText, δηλαδή το TextField του SaveDialog.

Γραμμή 336: Γίνεται αόρατο το SaveDialog.

Γραμμή 340: Δημιουργείται το αντικείμενο fw με βάση το όνομα αρχείου που δηλώθηκε.

Γραμμή 341: Αποθηκεύει το περιεχόμενο του JTextArea στο αρχείο.

Γραμμή 343: Γίνεται αόρατο το SaveDialog.

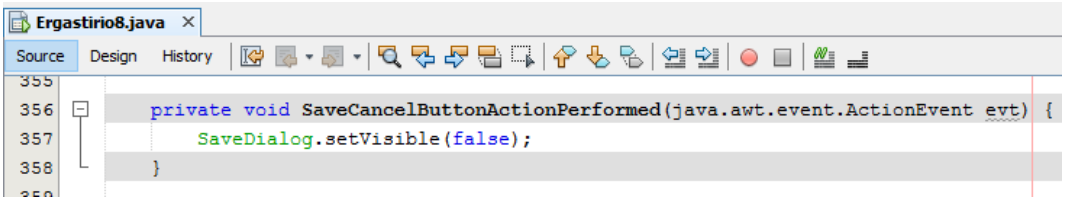
Ergastirio8.java

Source | Design | History

```
325     private void SaveAsFileActionPerformed(java.awt.event.ActionEvent evt) {
326         SaveDialog.setVisible(true);
327         SaveDialog.setSize(280, 170);
328     }
329
330     private void SaveButtonActionPerformed(java.awt.event.ActionEvent evt) {
331             FileWriter fw;
332             filename = SaveFileText.getText();
333             if(filename.length() == 0) {
334                message.setText("Ορίστε το όνομα και τη διαδρομή του αρχείου"
335                        + "που θέλετε να αποθηκεύσετε!");
336                SaveDialog.setVisible(false);
337                return;
338             }
339             try {
340                fw = new FileWriter(filename);
341                jTextArea1.write(fw);
342                fw.close();
343                SaveDialog.setVisible(false);
344                setTitle(filename);
345             } catch(IOException exc) {
346             message.setText("Λάθος όνομα αρχείου.");
347             return;
348             }
349             message.setText("Το αρχείο αποθηκεύτηκε σωστά.");
350     }
```

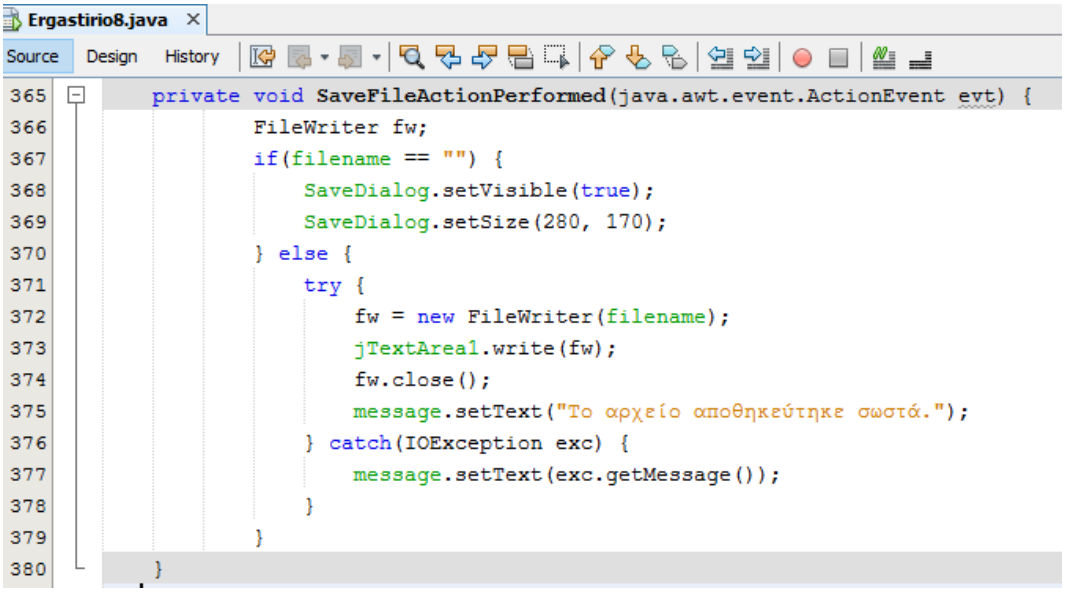
Εικόνα 19: Ο κώδικας του SaveAsFile και του SaveButton

Τέλος, ο κώδικας του γεγονότος ActionPerformed του κουμπιού Cancel του SaveDialog θα αποτελείται από μία εντολή που θα κάνει το SaveDialog αόρατο (Εικόνα 20).



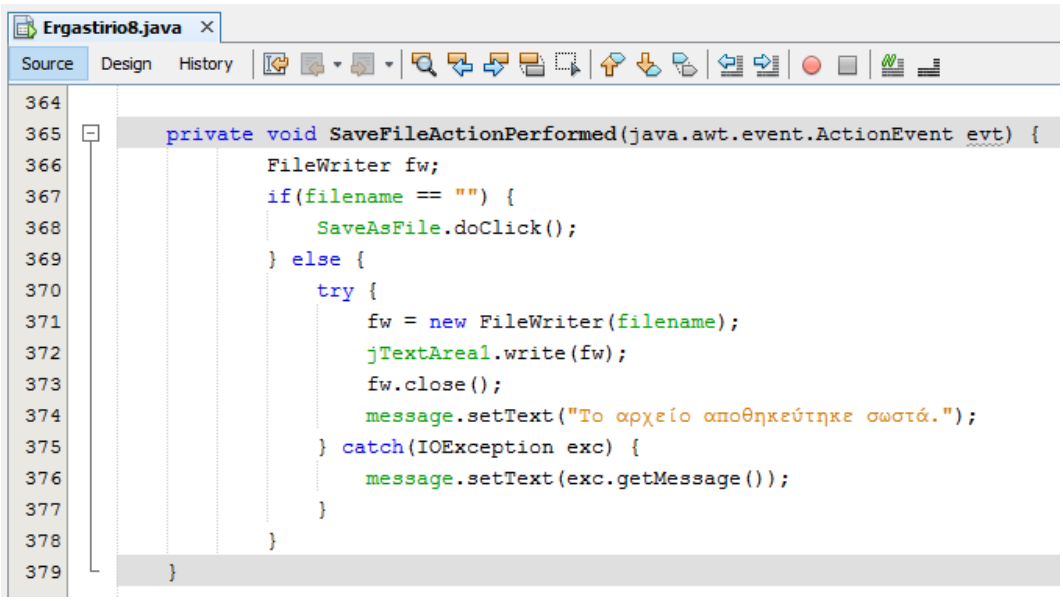
Εικόνα 20: Ο κώδικας του SaveCancelButton

Βήμα 13: Στη λειτουργία του «Save» (από το μενού) θεωρείται ότι το όνομα του αρχείου είναι γνωστό. Αυτή είναι η βασική διαφορά σε σχέση με το «Save As». Στην Εικόνα 21, φαίνεται ο κώδικας του γεγονότος ActionPerformed του SaveFile του μενού. Ο κώδικας είναι υποσύνολο του SaveButton του SaveDialog. Η λογική είναι ότι αν δεν έχει τιμή η μεταβλητή που αποθηκεύει το όνομα αρχείου, θα ανοίξει το SaveDialog (γραμμές 367-370) αλλιώς γίνεται η αποθήκευση (γραμμές 370-379).



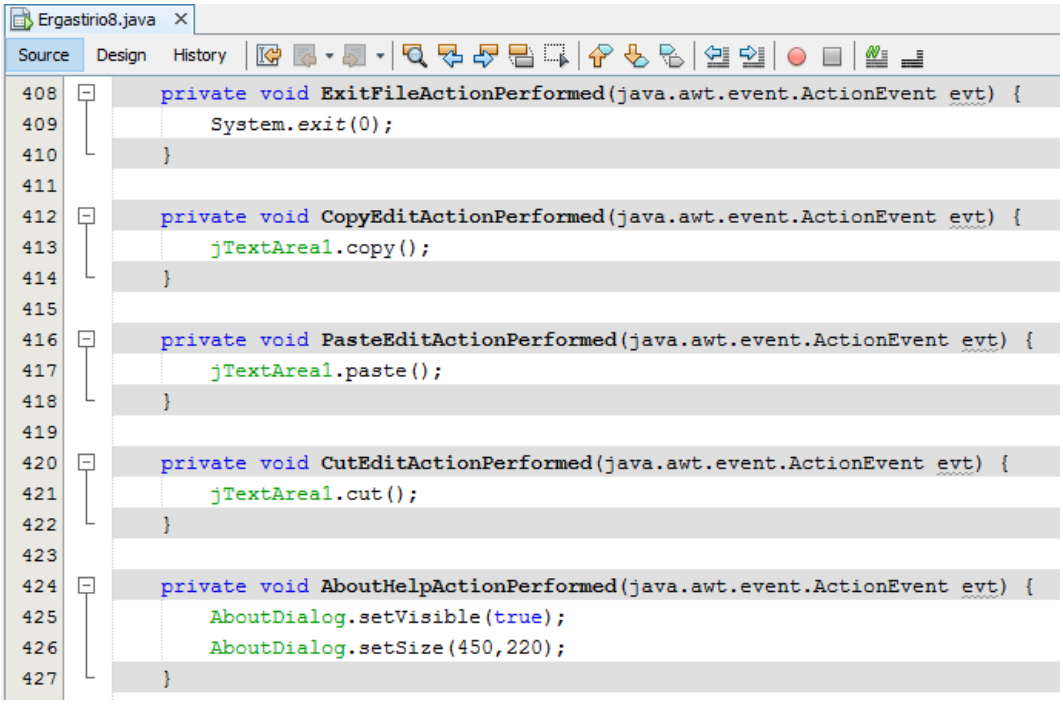
Εικόνα 21: Ο κώδικας του SaveFile

Εναλλακτικά, οι γραμμές 368-379 της Εικόνα 21 μπορούν να αντικατασταθούν με τη γραμμή 368 της Εικόνα 22. Στη $2^{η}$ λύση γίνεται κλήση του «Save As» του μενού. Ουσιαστικά με τη μέθοδο SaveAsFile.doClick() εκτελείται ο κώδικας της SaveAsFileActionPerformed της Εικόνα 19.



Εικόνα 22: Ο κώδικας του SaveFile (2η λύση) – μία μικρή αλλαγή

Βήμα 14: Σε αυτό το βήμα θα μελετήσουμε τις υπόλοιπες επιλογές των μενού (Εικόνα 23).



Εικόνα 23: Ο κώδικας των υπόλοιπων ενεργειών

Έτσι, η τελευταία επιλογή του υπομενού File, η Exit, ουσιαστικά κλείνει το πρόγραμμα. Ο κώδικας της φαίνεται στις γραμμές 408-410.

Το υπομενού Edit αποτελείται από τις επιλογές Copy, Paste και Cut. Ο κώδικας τους φαίνεται στις γραμμές 412 έως 422. Για να λειτουργήσουν οι Copy και Cut θα πρέπει πρώτα να επιλεγεί

το κείμενο που θέλει να αντιγράψει ή να αποκόψει ο χρήστης και μετά από το μενού να επιλεγούν οι συγκεκριμένες επιλογές. Επίσης, ο χρήστης αφού εκτελέσει την Copy ή την Cut θα τοποθετήσει τον δείκτη του ποντικιού στο επιθυμητό σημείο στο κείμενο και μετά θα επιλέξει από το μενού την επιλογή Paste.

Τέλος, η επιλογή About του υπομενού Help εμφανίζει το AboutDialog Dialog window (γραμμές 424-427).