

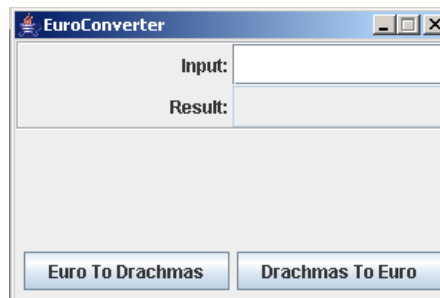
Εργαστηριακή άσκηση #12

Θέμα: Εφαρμογή με Γραφικό Περιβάλλον Επικοινωνίας

Η εργαστηριακή αυτή άσκηση αποσκοπεί στην εξοικείωση με τον προγραμματισμό εφαρμογών (applications) Γραφικού Περιβάλλοντος Επικοινωνίας (ΓΠΕ) και τις τεχνικές χειρισμού γεγονότων. Μέχρι στιγμής τα προγράμματα τα οποία είχαμε δημιουργήσει, είτε δεν υποστήριζαν καθόλου διαλογικότητα με τον χρήστη, ή απλά επέτρεπαν την εισαγωγή επιλογών από το πληκτρολόγιο μέσω του ρεύματος δεδομένων εισόδου System.in. Το έργο το οποίο θα κατασκευάσουμε στο πλαίσιο αυτής της εργαστηριακής άσκησης αναλαμβάνει την υλοποίηση τόσο της γραφικής διαπροσωπείας, μέσω της οποίας ο χρήστης αποκτά πρόσβαση στη λειτουργικότητα της εφαρμογής, όσο και του κώδικα χειρισμού των γεγονότων που συμβαίνουν κατά τη διάρκεια της αλληλεπίδρασης του χρήστη με τα συστατικά της διαπροσωπείας.

Θα κατασκευάσουμε την κλάση EuroConverter, η οποία θα σχεδιάζει το ΓΠΕ και θα υλοποιεί τον κώδικα για την εισαγωγή αριθμητικών δεδομένων κινητής υποδιαστολής (double) και τη μετατροπή τους από Ευρώ σε δραχμές και αντίστροφα.

Το ΓΠΕ της εφαρμογής θα αποτελείται από ένα αυτόνομο παράθυρο (Frame), στο πλαίσιο του οποίου θα ενταχθούν ως συστατικά επικοινωνίας: μια ετικέτα (Label) και ένα πεδίο κειμένου (TextField) για την εισαγωγή των δεδομένων προς μετατροπή, μια ετικέτα και ένα πεδίο κειμένου για την εμφάνιση των αποτελεσμάτων, μια ετικέτα για την εμφάνιση μηνύματος λάθους σε περίπτωση εισαγωγής λανθασμένου τύπου δεδομένων και δυο πλήκτρα (Button) για την επιλογή του είδους μετατροπής. Ο επιθυμητός τρόπος διάταξης των συστατικών στο παράθυρο της εφαρμογής εμφανίζεται στην εικόνα που ακολουθεί:



1. Δημιουργήστε το έργο EuroConverter με το BlueJ.
2. Δημιουργήστε την κλάση EuroConverter, η οποία θα επεκτείνει την κλάση JFrame και θα έχει τα παρακάτω πεδία:

private JTextField inputField	Πεδίο κειμένου για την εισαγωγή των δεδομένων
private JTextField outputField	Πεδίο κειμένου για την παρουσίαση των αποτελεσμάτων
private JLabel errorMessage	Ετικέτα εμφάνισης μηνύματος λανθασμένης εισαγωγής
private Container cp	Περιοχή περιεχομένων για την ένταξη των συστατικών

3. Να υλοποιηθεί ο κατασκευαστής EuroConverter(), όπως παρακάτω:

```
public EuroConverter()
{
    super("EuroConverter");
    cp=getContentPane();
    makeConverterGUI();
}
```

Η εντολή super προκαλεί κλήση του κατασκευαστή της υπερκλάσης JFrame, ο οποίος όταν δέχεται ως παράμετρο ένα String κατασκευάζει ένα, αρχικά μη ορατό, αυτόνομο παράθυρο με τίτλο αυτόν της παραμέτρου. Η μέθοδος getContentPane() επιστρέφει, για το συγκεκριμένο παράθυρο, τον υποδοχέα της περιοχής περιεχομένων, στον οποίο θα ενταχθούν τα επιθυμητά γραφικά συστατικά. Τέλος, ο κατασκευαστής περιέχει μια κλήση στη μέθοδο υλοποίησης του ΓΠΕ makeConverterGUI().

4. Να υλοποιηθεί η μέθοδος main της εφαρμογής ώστε:
 - a. να δημιουργεί ένα EuroConverter παράθυρο (frame),
 - b. να ορίζει αρχικές διαστάσεις (300x200)
 - c. χαρακτηριστικά του παραθύρου (ορατότητα, σταθερό μέγεθος, λειτουργία κατά το κλείσιμο του παραθύρου).

ΣΗΜ. Συμβουλευτείτε την τεκμηρίωση και χρησιμοποιήστε τις μεθόδους setDefaultCloseOperation, setSize, setVisible και setResizable.

5. Να υλοποιήσετε τη μέθοδο `makeConverterGUI()`, με βάση τις εξής παρατηρήσεις:

- Ο υποδοχέας της περιοχής περιεχομένων του παραθύρου της εφαρμογής, `cp`, θα περιέχει τρεις υποδοχείς τύπου `JPanel`: τον `inoutPanel`, στον οποίο θα ενταχθούν τα συστατικά `inputField` και `outputField` με τις αντίστοιχες ετικέτες πληροφορίας, τον `errorPanel`, που θα περιλαμβάνει το συστατικό `errorMessage`, και τον `conversionPanel`, για τα πλήκτρα μετατροπής από Ευρώ σε δραχμές και αντίστροφα.
- Ο ορισμός κάθε υποδοχέα περιλαμβάνει τη δημιουργία σχετικού αντικειμένου, τον ορισμό των χαρακτηριστικών του (π.χ. διαστάσεις), τον ορισμό του τύπου διαχειριστή διάταξης (`layout manager`) που θα χρησιμοποιήσει για τη διάταξη των συστατικών. Στο απόσπασμα κώδικα που ακολουθεί φαίνεται η δημιουργία και διαμόρφωση του `inoutPanel`. Για τους υπόλοιπους υποδοχείς επιλέξτε διαχειριστή διάταξης `BorderLayout`.

ΣΗΜ. Για τα σημεία όπου χρησιμοποιούνται αντικείμενα ή μέθοδοι κλάσεων με τα οποία δεν είστε εξοικειωμένοι, συμβουλευτείτε την τεκμηρίωση της γλώσσας που παρέχεται από την ιστοσελίδα του μαθήματος.

```
public void makeConverterGUI()
{
    ...
    JPanel inoutPanel=new JPanel(new GridLayout(2,2));
    inoutPanel.setPreferredSize(new Dimension(0,60));
    inoutPanel.setBorder(BorderFactory.createEtchedBorder());
    ...
}
```

- Η εισαγωγή κάθε συστατικού περιλαμβάνει τη δημιουργία του συστατικού, τον ορισμό των ειδικών χαρακτηριστικών του (π.χ. χρώμα γραμματοσειράς, ευθυγράμμιση κ.λπ), το συσχετισμό του με κατάλληλο ακροατή γεγονότων (`Action Listener`) – στις περιπτώσεις όπου χρειάζεται χειρισμός των δράσεων που προκαλούνται από ενέργειες αλληλεπίδρασης του χρήστη – και, τέλος, την ένταξή του στον κατάλληλο υποδοχέα περιεχομένων. Στη συγκεκριμένη άσκηση, μόνο τα πλήκτρα μετατροπής καλούν ακροατή γεγονότων. Ακολουθεί ο κώδικας για το πεδίο εισόδου δεδομένων και το πλήκτρο μετατροπής από Ευρώ σε δραχμές.

```
public void makeConverterGUI()
{
    ...
    inputField=new JTextField("",25);
    inputField.setHorizontalAlignment(JTextField.RIGHT);
    inoutPanel.add(inputField);

    ...

    JButton b=new JButton("Euro To Drachmas");
    b.addActionListener(new ConversionListener());
    conversionPanel.add(b);
    ...
}
```

- Η δημιουργία των υποδοχέων και η ένταξη των αντίστοιχων συστατικών σε αυτά, συμπληρώνεται με την προσθήκη όλων των υποδοχέων στον υποδοχέα της περιοχής περιεχομένων του παραθύρου, οπότε και ολοκληρώνεται η διαδικασία υλοποίησης του ΓΠΕ της εφαρμογής.

6. Να δημιουργήσετε την εσωτερική κλάση `ConversionListener`, η οποία θα υλοποιεί τη διαπροσωπεία `ActionListener`. Η κλάση `ConversionListener` περιέχει τη μέθοδο `actionPerformed(ActionEvent e)`, στην οποία περιγράφεται ο τρόπος απόκρισης της εφαρμογής σε πιθανά πατήματα των πλήκτρων μετατροπής από το χρήστη.

```
public void actionPerformed(ActionEvent e)
{
    double numResult=0.0;
    String strResult="";

    NumberFormat nf = NumberFormat.getNumberInstance();
    nf.setMaximumFractionDigits(2);

    String operation=e.getActionCommand();

    try
    {
        errorMessage.setText("");
        if (operation.equals("Euro To Drachmas"))
        {
```

```
        numResult = 340.75 * Double.parseDouble(inputField.getText());
    }
    else
    {
        numResult = Double.parseDouble(inputField.getText()) / 340.75;
    }

    outputField.setText(nf.format(numResult));
}
catch (NumberFormatException ex)
{
    errorMessage.setText("Please insert numeric data!");
}
}
```

7. Εκτελέστε τη μέθοδο `main()` και πειραματιστείτε με τη λειτουργικότητα της εφαρμογής. Παρατηρήστε το εξής 'πρόβλημα': Όταν εισάγετε αριθμούς με δεκαδικά ψηφία στο πεδίο `inputField`, είναι απαραίτητη η χρήση της τελείας ως σημείου υποδιαστολής. Σε περίπτωση που χρησιμοποιηθεί κόμμα, η είσοδος θεωρείται ως συμβολοσειρά και η εφαρμογή τυπώνει μήνυμα λάθους. Η ίδια σύμβαση δεν ακολουθείται όμως και κατά την εκτύπωση αποτελεσμάτων με δεκαδικά ψηφία! Στο πεδίο αποτελεσμάτων ως σημείο υποδιαστολής εμφανίζεται το κόμμα, ενώ η τελεία χρησιμοποιείται στο διαχωρισμό των χιλιάδων, εκατομμυρίων, κ.ο.κ. Για ποιο λόγο νομίζετε ότι συμβαίνει αυτό; Κάντε τις απαραίτητες μετατροπές στο πρόγραμμά σας ώστε να ακολουθείται κοινή σύμβαση αναπαράστασης κατά την είσοδο και έξοδο των δεδομένων.