

ΓΡΑΦΙΚΕΣ ΔΙΕΠΑΦΕΣ ΧΡΗΣΤΗ Ι

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η εισαγωγή στη σχεδίαση γραφικών διεπαφών χρήστη (**Graphical User Interfaces, GUI**), αξιοποιώντας κλάσεις που προσφέρει η Java για το σκοπό αυτό. Εστιάζει στη χρήση της βιβλιοθήκης **Swing** (πακέτο `javax.swing`) η οποία προσφέρει βελτιωμένες κλάσεις γραφικών συστατικών σε σχέση με την προγενέστερη βιβλιοθήκη **Abstract Windowing Toolkit** (`java.awt`). Ωστόσο, εξακολουθούν να χρησιμοποιούνται κάποια στοιχεία από την τελευταία.

14.1 Παράθυρα

Ένα παράθυρο μπορεί να κατασκευαστεί και να εμφανιστεί εύκολα ως αντικείμενο της κλάσης **JFrame**, όπως εικονίζεται στο Σχ.1. Ωστόσο αυτός δεν είναι ο προτεινόμενος τρόπος για να κατασκευάζετε παράθυρα.

```
1  import javax.swing.*;
2
3  public class FirstAttempt
4  {
5      public static void main (String[] args)
6      {
7          JFrame window = new JFrame("My first window");
8          window.setSize(100,100);
9          window.setDefaultCloseOperation(
10             JFrame.EXIT_ON_CLOSE);
11          window.setVisible(true);
12      }
13 }
```

Σχήμα 1. Εμφάνιση παραθύρου μέσα από μια `main`.

Η μέθοδος `setSize` ορίζει το αρχικό μέγεθος του παραθύρου, η μέθοδος `setDefaultCloseOperation` ορίζει, μέσω της σταθεράς `EXIT_ON_CLOSE`, πως όταν πιέζεται το κουμπί **[X]** του παραθύρου θα πρέπει να τερματίζεται η λειτουργία της εφαρμογής, και η `setVisible` δίνει την εντολή να εμφανιστεί το παράθυρο (ή να κρυφτεί, σε περίπτωση που το όρισμά της είναι `false`).

Ο προτεινόμενος τρόπος για την κατασκευή παραθύρων εικονίζεται στο Σχ.2, όπου παρουσιάζεται η κατασκευή ενός παραθύρου με ένα κουμπί (το οποίο δεν εκτελεί κάποια ενέργεια). Σύμφωνα με τον τρόπο αυτό:

- Ένα παράθυρο πρέπει να κατασκευάζεται ως νέα κλάση η οποία είναι υποκλάση της κλάσης `JFrame`.
- Τα περιεχόμενα του παραθύρου (τα οποία ονομάζονται συστατικά – components), π.χ. κουμπιά, πρέπει να δηλώνονται και να δημιουργούνται ως μεταβλητές υπόστασης της κλάσης του παραθύρου.
- Η συναρμολόγηση του παραθύρου από τα συστατικά του πρέπει να γίνεται μέσα στη μέθοδο δημιουργό της κλάσης του παραθύρου.

```
1  import javax.swing.*;
2
3  public class MyWindow extends JFrame
4  {
5      JButton b1 = new JButton("OK");
6
7      public MyWindow(String title)
8      {
9          super(title);
10         setSize(100,100);
11         setDefaultCloseOperation(EXIT_ON_CLOSE);
12         setVisible(true);
13
14         // Add button to window
15         add(b1);
16     }
17 }
```

Σχήμα 2. Προτεινόμενος τρόπος κατασκευής κλάσης παραθύρου.

Για να εμφανιστεί το παράθυρο του Σχ.2 αρκεί να δημιουργηθεί ένα αντικείμενο της κλάσης `myWindow`, όπως διακρίνεται στο Σχ.3.

```
1  public class SecondAttempt
2  {
3      public static void main (String[] args)
4      {
5          MyWindow w = new MyWindow("My second window");
6
7      }
8  }
```

Σχήμα 3. Εμφάνιση παραθύρου δημιουργώντας ένα αντικείμενο της κλάσης `MyWindow` (Σχ.2).

14.2 Συναρμολόγηση παραθύρου

Για να συναρμολογηθεί ένα παράθυρο από διάφορα συστατικά πρέπει πρώτα να δημιουργήσουμε τα αντικείμενα των συστατικών και κατόπιν να τα προσθέσουμε στο παράθυρο, με τον ίδιο τρόπο που προστέθηκε το κουμπί στο παράθυρο του Σχ.2.

Όταν επιθυμούμε να συναρμολογήσουμε ένα παράθυρο από πολλά συστατικά τότε πρέπει να προσδιορίσουμε τον τρόπο με τον οποίο θα διατάσσονται (layout) στο παράθυρο.

Μελετώνται τρία είδη διατάξεων:

- **Διάταξη ροής**, που υλοποιείται μέσω της κλάσης `FlowLayout`. Σύμφωνα με τη διάταξη αυτή τα συστατικά του παραθύρου διατάσσονται το ένα δίπλα στο άλλο, όσο χωράνε στο πλάτος του παραθύρου, και μετά συνεχίζουν από κάτω (Σχ.4).

Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, π.χ.:

```
setLayout(new FlowLayout());
```

και μετά ένα συστατικό `b` μπορεί να προστεθεί με:

```
add(b);
```



Σχήμα 4. Διάταξη ροής (εικόνα από Oracle Java™ Tutorials).

- **Διάταξη πλέγματος**, που υλοποιείται μέσω της κλάσης `GridLayout`. Σύμφωνα με τη διάταξη αυτή τα συστατικά του παραθύρου διατάσσονται το ένα μετά το άλλο σε ένα πλέγμα (grid) / πίνακα των διαστάσεων που επιθυμούμε. Δεν

προσδιορίζουμε σε ποιο κελί του πίνακα θα τοποθετηθεί ένα αντικείμενο (Σχ.5).

Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, π.χ.:

```
setLayout(new GridLayout(3,2));
```

αν θέλουμε το πλέγμα να έχει 3 γραμμές και 2 στήλες (όπως στο Σχ.5), και μετά ένα συστατικό b μπορεί να προστεθεί με:

```
add(b);
```



| | |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

Σχήμα 5. Διάταξη πλέγματος (εικόνα από Java™ API).

- **Διάταξη περιθωρίου**, που υλοποιείται μέσω της κλάσης BorderLayout. Σύμφωνα με τη διάταξη αυτή τα συστατικά του παραθύρου διατάσσονται σε πέντε δυνατές θέσεις: Βόρεια (**NORTH**), Νότια (**SOUTH**), Ανατολικά (**EAST**), Δυτικά (**WEST**) (Σχ.6).

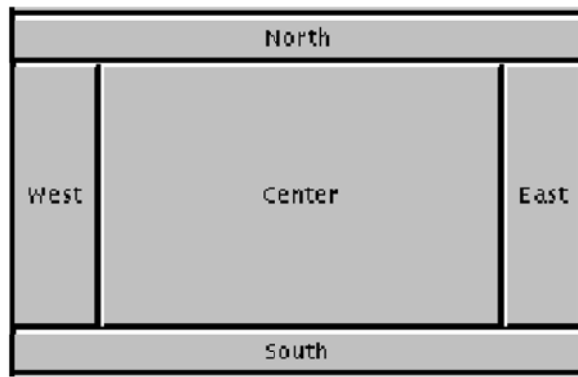
Για την εφαρμογή αυτής της διάταξης πρέπει στη μέθοδο δημιουργό του παραθύρου να γράψουμε, π.χ.:

```
setLayout(new BorderLayout());
```

και μετά ένα συστατικό b μπορεί να προστεθεί με:

```
add(b, BorderLayout.NORTH);
```

αν επιθυμούμε να προστεθεί το b βόρεια.



Σχήμα 6. Διάταξη περιθωρίου (εικόνα από Java™ API).

Στο Σχ. 7 εικονίζεται ένα παράδειγμα γραφικής διεπαφής χρήστη με δύο κουμπιά (Yes και No) και ένα πεδίο κειμένου, με διάταξη πλέγματος τριών γραμμών και μιας στήλης.

```

1  import javax.swing.*;
2  import java.awt.*;
3
4  public class YesNo extends JFrame
5  {
6      JButton b1 = new JButton("YES");
7      JButton b2 = new JButton("NO");
8      JTextField tf = new JTextField();
9
10     public YesNo()
11     {
12         super("Yes/No");
13         setSize(100,100);
14         setDefaultCloseOperation(EXIT_ON_CLOSE);
15         setVisible(true);
16
17         setLayout(new GridLayout(3,1));
18         add(tf);
19         add(b1);
20         add(b2);
21     }
22
23     public static void main (String[] args)
24     {
25         YesNo w = new YesNo();
26     }
27 }

```

Σχήμα 7. Ένα παράθυρο με ένα πεδίο κειμένου και επιλογές Yes/No (τα κουμπιά δεν εκτελούν κάποια ενέργεια).

14.3 Εκτέλεση ενεργειών

Προκειμένου να γίνει ένα GUI λειτουργικό για την αλληλεπίδραση του χρήστη με αυτό, πρέπει οι ενέργειες που εκτελεί ο χρήστης σε αυτό, π.χ., κλικ σε ένα κουμπί, να γίνονται αντιληπτές από το σύστημα. Δηλαδή με άλλα λόγια, το σύστημα να «ακούει» (listens) τις ενέργειες του χρήστη.

Αυτό επιτυγχάνεται επιβάλλοντας στην κλάση του παραθύρου να **υλοποιεί τη διεπαφή ActionListener** (πακέτο `java.awt.event`), ή και άλλες διεπαφές που διαθέτει η Java. Οι διεπαφές αυτές προσφέρουν μεθόδους, οι οποίες ενεργοποιούνται κάθε φορά που εκτελείται μια ενέργεια. Με αυτό τον τρόπο μπορεί το σύστημα να ανταποκριθεί στις ενέργειες του χρήστη, π.χ. όταν κάνει κλικ σε ένα κουμπί, να εκτελεστεί μια ενέργεια.

Ειδικά για την εκτέλεση ενεργειών που σχετίζονται με την υλοποίηση της διεπαφής ActionListener απαιτούνται τρία βήματα:

- Υλοποίηση της διεπαφής, δηλ. **implements ActionListener**.
- Προσθήκη της δυνατότητας σε ένα αντικείμενο να «ακούγεται», δηλ. σε ένα κουμπί b πρέπει να κάνουμε **b.addActionListener(this)**;
- Υλοποίηση της μεθόδου **actionPerformed**, στην οποία πρέπει να εντοπίζεται από ποιο συστατικό προήλθε η ενέργεια και να εκτελούνται οι λειτουργίες που επιθυμούμε ως αποτέλεσμα της ενέργειας του χρήστη. Αυτό γίνεται μέσω της μεθόδου **getSource**, όπως εικονίζεται στις γραμμές 28, 29 του Σχ.8.

Στο Σχ.8 παρουσιάζεται όλη η παραπάνω διαδικασία για το GUI Yes/No. Δηλαδή, στο νέο σχήμα, το GUI Yes/No μπορεί να «ακούσει» τα κλικ στα κουμπιά και να ανταποκριθεί. Ο κώδικας που προστέθηκε εικονίζεται με κόκκινο χρώμα.

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class YesNo extends JFrame implements ActionListener
6  {
7      JButton b1 = new JButton("YES");
8      JButton b2 = new JButton("NO");
9      JTextField tf = new JTextField();
10
11     public YesNo()
12     {
13         super("Yes/No");
```

```

14         setSize(100,100);
15         setDefaultCloseOperation(EXIT_ON_CLOSE);
16         setVisible(true);
17
18         setLayout(new GridLayout(3,1));
19         add(tf);
20         add(b1);
21         b1.addActionListener(this);
22         add(b2);
23         b2.addActionListener(this);
24     }
25
26     public void actionPerformed(ActionEvent e)
27     {
28         if (e.getSource() == b1) tf.setText("Yes");
29         if (e.getSource() == b2) tf.setText("No");
30     }
31
32     public static void main (String[] args)
33     {
34         YesNo w = new YesNo();
35     }
36
37 }

```

Σχήμα 8. Ένα παράθυρο με ένα πεδίο κειμένου και επιλογές Yes/No (τα κουμπιά λειτουργούν).

Η μέθοδος `setText` χρησιμοποιείται για την εκτύπωση της λέξης “Yes” ή “No” στο πεδίο κειμένου. Αντίστοιχα μπορείτε να τη χρησιμοποιήσετε και για άλλα συστατικά που περιέχουν κείμενο, π.χ. `JLabel` (ετικέτες). Για να διαβάσει κανείς το κείμενο που περιέχει ένα συστατικό χρησιμοποιείται η μέθοδος `getText`, π.χ.,

```
String s = tf.getText();
```

Δείτε περισσότερες λεπτομέρειες για άλλες κλάσεις και μεθόδους συστατικών στο API του πακέτου `javax.swing`.

Ασκήσεις

1. Να υλοποιήσετε το παράδειγμα του Σχ.7 και μετά, αφού το εκτελέσετε, να εισάγετε τις τροποποιήσεις που προτείνονται στο Σχ.8.
2. Να τροποποιήσετε το παράδειγμα του Σχ.8 δοκιμάζοντας να χρησιμοποιήσετε διάταξη ροής και διαφορετικές διατάξεις περιθωρίου.

3. Να χρησιμοποιήσετε την κλάση **JLabel** (δείτε το Java API για λεπτομέρειες) για να προσθέσετε δύο ετικέτες στο GUI του Σχ.8. Η μία να είναι επάνω από τα κουμπιά και να γράφει “Press buttons:” και η άλλη επάνω από το πεδίο κειμένου και να γράφει “Output:”.

4. Να κατασκευάσετε μια γραφική διεπαφή χρήστη με ένα αντικείμενο περιοχής κειμένου **JTextArea** και ένα κουμπί από κάτω. Όταν πιέζεται το κουμπί, στην περιοχή κειμένου να προστίθεται μια νέα γραμμή κειμένου που να γράφει “Pressed.”