



Java Foundation Classes Swing και GUI

Αθ. Ανδρούτσος



Εφαρμογές - GUI

Java Swing

- Μία εμπορική εφαρμογή είναι αποτελείται από μία ΒΔ, την λογική της εφαρμογής (middleware) και τις οθόνες (frontend)
- Όσο αφορά τις οθόνες, όταν περιλαμβάνουν γραφικά στοιχεία ονομάζονται **Graphical User Interfaces (GUIs)**
- Οθόνες μόνο με κονσόλα χωρίς γραφική διεπαφή, δεν συνηθίζεται σε εμπορικές εφαρμογές γιατί δεν είναι φιλικές προς τον χρήστη



Γραφικές Διεπαφές

Java Swing

- Υπάρχουν δύο τύποι -και ως εκ τούτου δύο τεχνολογίες- γραφικών διεπαφών.
- **Desktop GUI**
 - Αφορά εφαρμογές που τρέχουν σε προσωπικά pc ή σε μικρά τοπικά δίκτυα. Οι βασικές Java-based frontend τεχνολογίες είναι οι Swing και JavaFX
- **Web-based GUI**
 - Αφορά εφαρμογές over web, όπου οι βασικές frontend τεχνολογίες είναι HTML/CSS/JavaScript / JSP/JSF/Thymeleaf / Vue/React/Angular



Φόρμες

Java Swing

- Ειδικά στις εμπορικές εφαρμογές εκτός από το εικαστικό μέρος μας ενδιαφέρει ιδιαίτερα η διαχείριση (CRUD) δεδομένων από τους χρήστες μέσα από **φόρμες εισαγωγής στοιχείων**
- Τα στοιχεία αυτά είναι που δίνουν τη δυνατότητα αλληλεπίδρασης με τον χρήστη



GUI και GUI Components

Java Swing

- Οι οθόνες **περιέχουν φόρμες που με τη σειρά τους** περιλαμβάνουν controls ή widgets που είναι αντικείμενα UI (UI Elements) για την αλληλεπίδραση με τον χρήστη, όπως:
 - Ετικέτες (Labels)
 - Πεδία εισαγωγής κειμένου (Text Boxes)
 - Πεδία πολλαπλών επιλογών (Combo Boxes)
 - Κουμπιά εντολών (Buttons)
 - Και άλλα ...

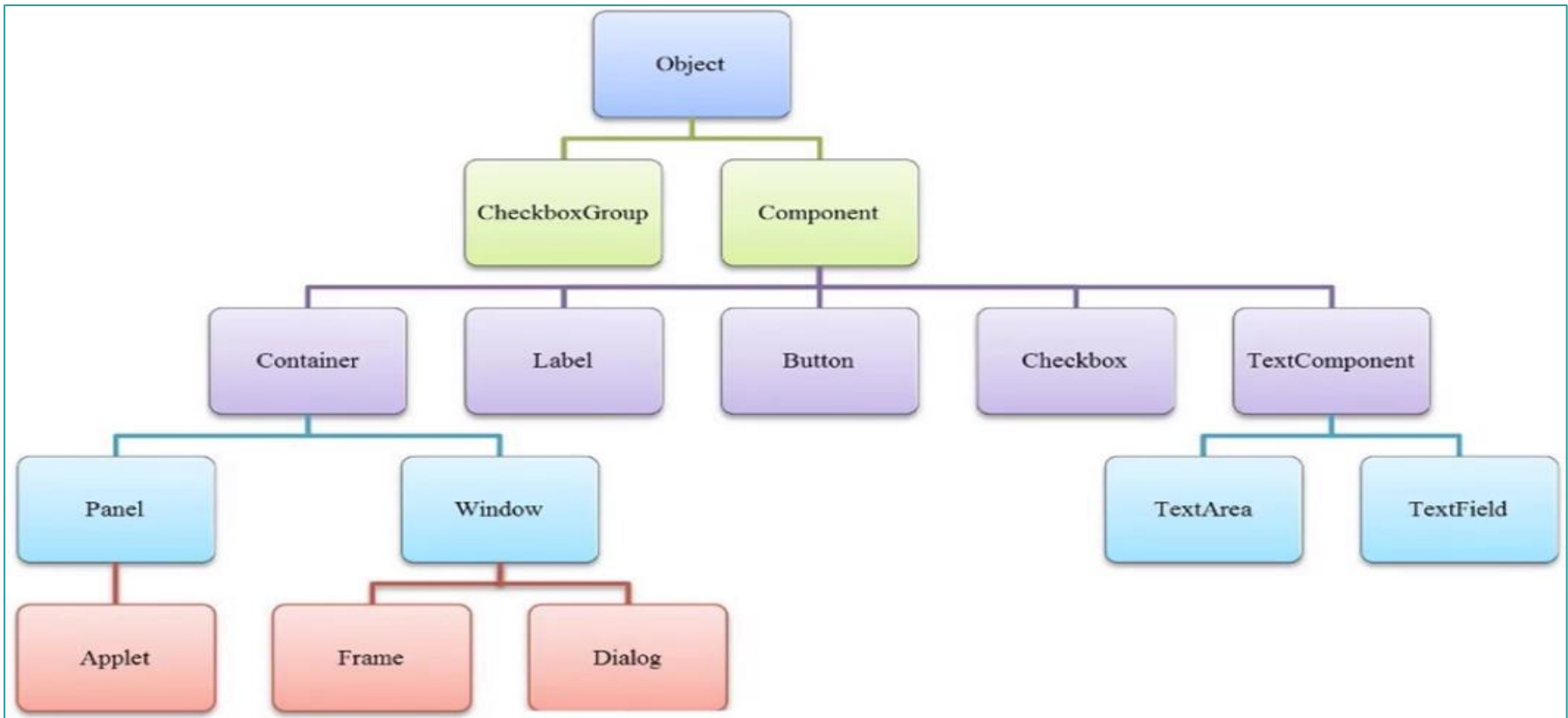


Java GUI Library

- Τα **Java Foundation Classes (JFC)** είναι GUI components για desktop εφαρμογές Java και είναι platform-dependent (δηλαδή εμφανίζονται όπως ορίζει το Λειτουργικό Σύστημα). Όλα τα JFCs είναι JavaBeans με ιδιότητες, getters και setters
- Τα JFCs περιλαμβάνονται στις παρακάτω βιβλιοθήκες
 - AWT (Abstract Window Toolkit)
 - Swing
 - Java 2D
- **Swing** – Αντικατέστησε το AWT. Παρέχει platform-independent GUI components



AWT



- Όλα τα UI Elements όπως Button, Label, Checkbox, TextField κλπ. είναι UI Elements ενώ υπάρχουν και containers όπως Frame που αντιπροσωπεύει όλο το window και Panel που είναι container μέσα στο Frame

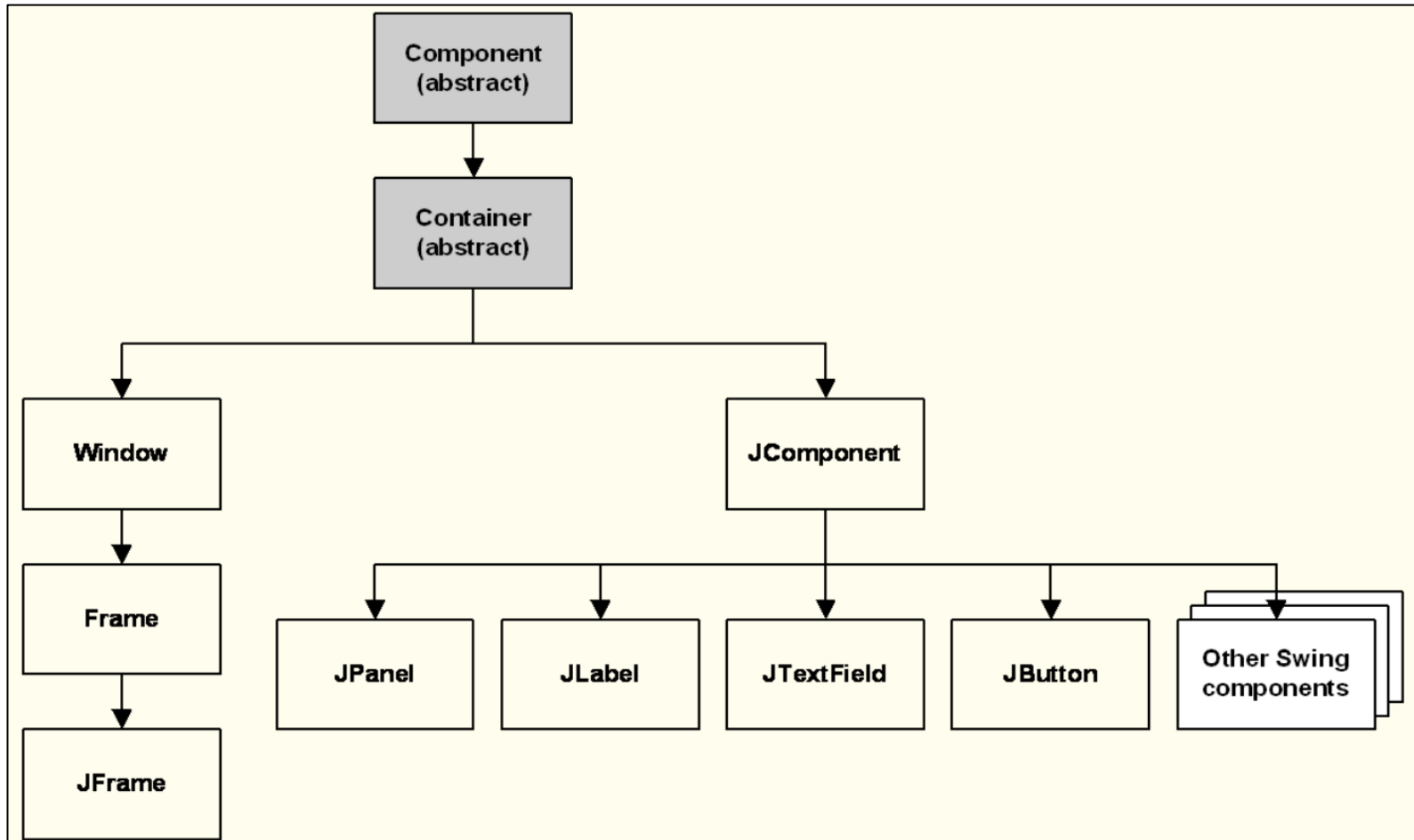


Swing και AWT

- Το Swing έχει δημιουργηθεί on-top of AWT και είναι platform-independent. Οι κλάσεις και τα Interfaces που παρέχονται από το swing βρίσκονται μέσα στο `javax.swing package`
- Οι κλάσεις και τα Interfaces του AWT όπως events και Listeners βρίσκονται στο `java.awt package`



Swing Hierarchy





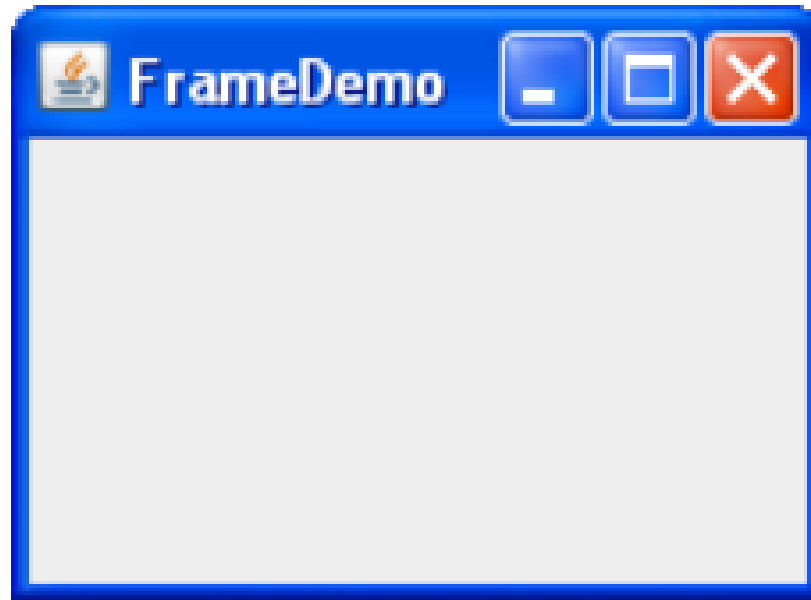
Δομή Παραθύρων

Java Swing

- Κάθε παράθυρο στο Swing είναι ένα ***JFrame***
- Ένα JFrame μπορεί να περιέχει ένα ή περισσότερα JPanel που αποτελεί χώρο για ομαδοποιήσεις των controls
- Τα controls είναι αντικείμενα για αλληλεπίδραση με τον χρήστη, όπως JLabel, JTextField, JButton, κλπ



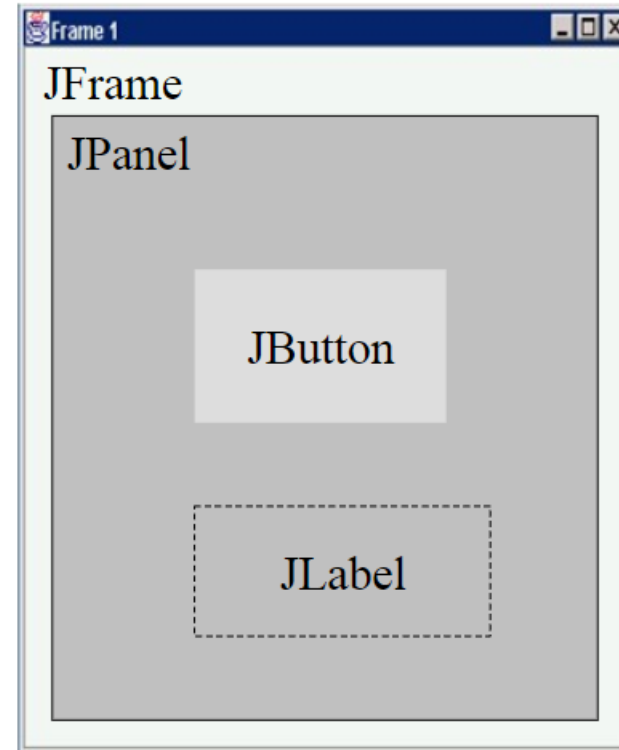
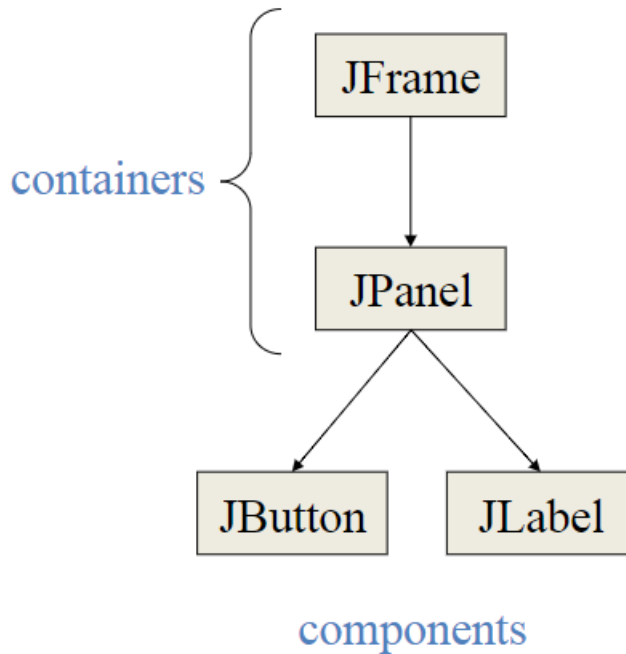
JFrame



- Το **JFrame** είναι ένα απλό παράθυρο με ένα περίγραμμα, τίτλο και buttons που κάνουν close ή iconify το παράθυρο



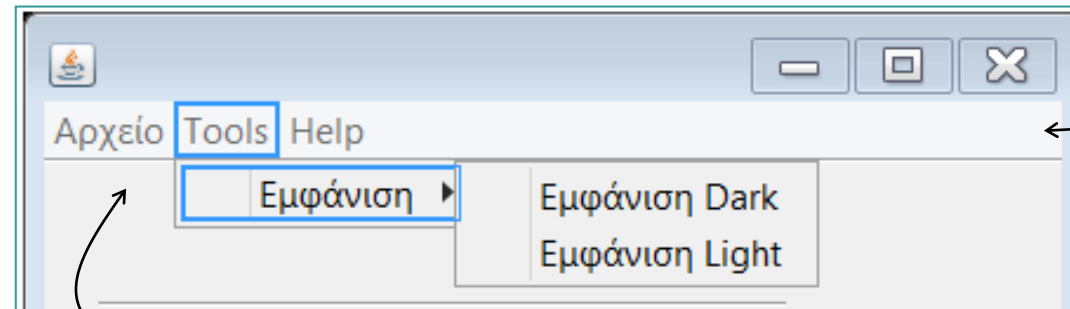
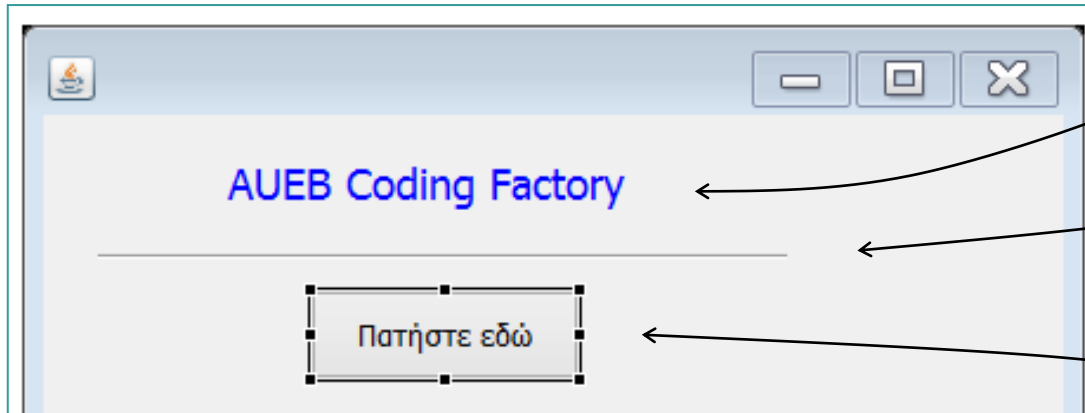
Δομή JFrame



- Το **JFrame** είναι το παράθυρο
- Το **JPanel** είναι το content pane που περιέχει αντικείμενα **JComponent** (**JButton**, **JLabel**, ..)



Controls (1)



- JLabel
- JSeparator
- JButton
- JMenuBar
- JMenu
- JMenuItem

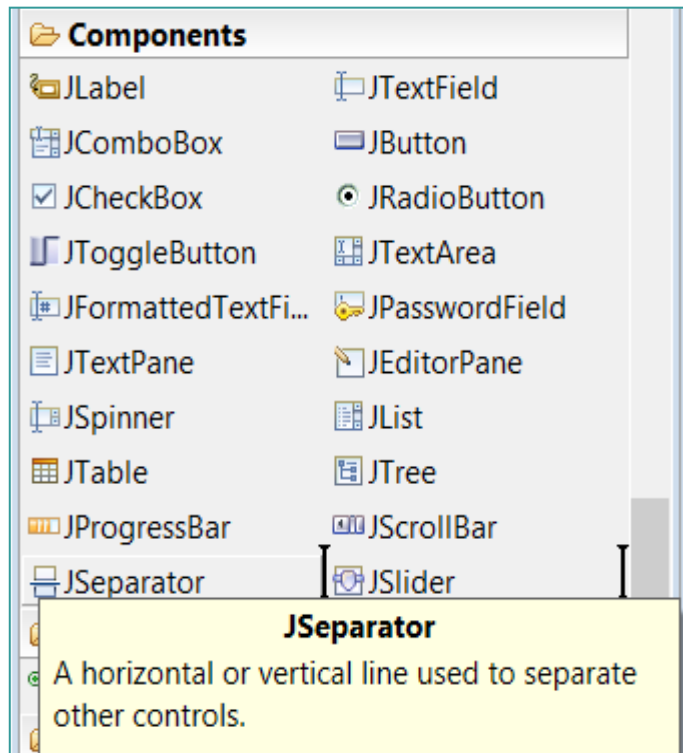


Controls (2)

- JTextBox
- JComboBox
- JCheckBox
- JPanel
- JLabel's



Components



- Διάφορα Components
- Μπορείτε να δείτε την περιγραφή στο Eclipse αν τοποθετήσετε το ποντίκι πάνω στο Component



Δημιουργία Component

Java Swing

- **Create** - Δημιουργία του component με new
- **Configure** – Set τις ιδιότητές του component
- **Add** - Προσθήκη του component στο JPanel



Εφαρμογές με Γραφική Διεπαφή

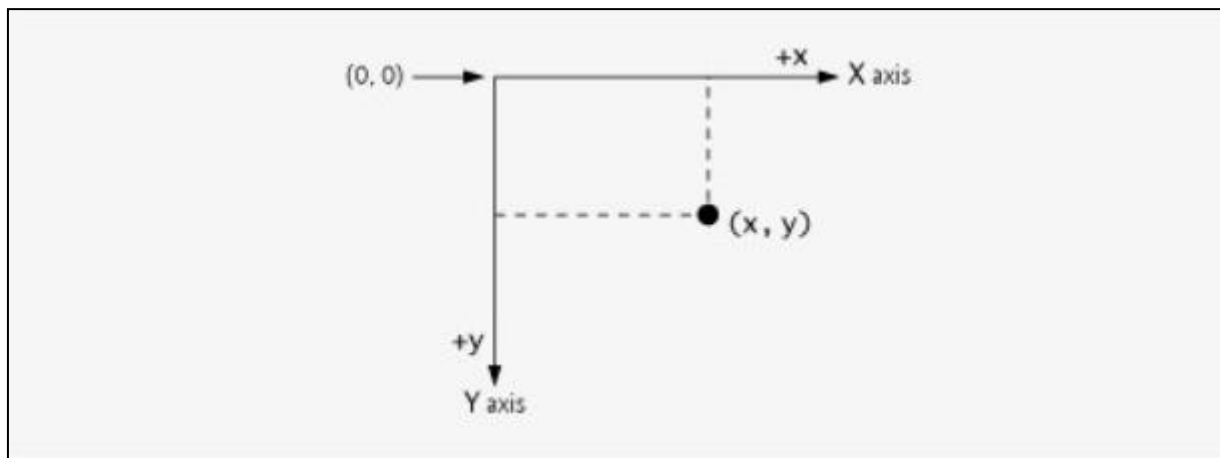
Java Swing

- **Δημιουργία παραθύρου.** Οι εφαρμογές με γραφική διεπαφή τρέχουν μέσα σε παράθυρα που πρέπει να δημιουργηθούν
- Η Java παρέχει την κλάση **JFrame** που αναπαριστά ένα παράθυρο



Java Coordinate System

- Τα μεγέθη του συστήματος συντεταγμένων της Java μετρώνται σε pixels
- Η `.setSize(width, height)` καθορίζει τις διαστάσεις του Παραθύρου σε pixels
- Η `.setBounds(x, y, width, height)` τοποθετεί την αριστερή γωνία ενός component στο σημείο x, y και θέτει πλάτος = width και ύψος = height





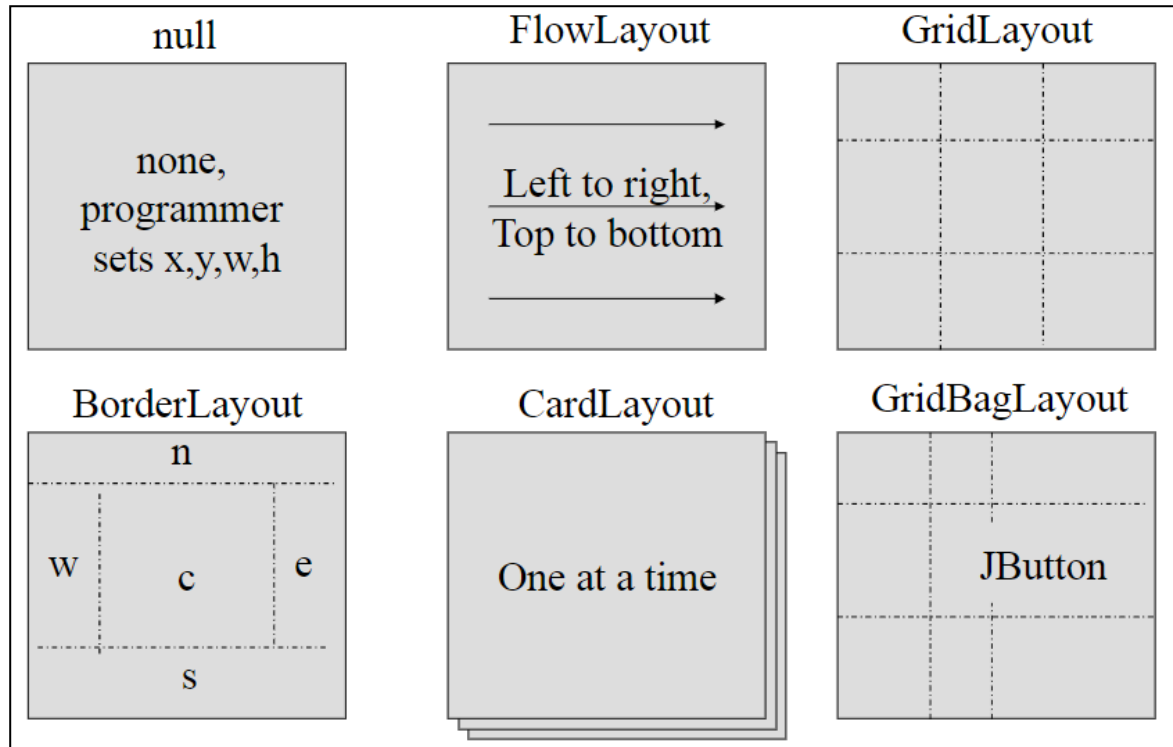
Controls - Components

Java Swing

- Το JFrame περιλαμβάνει ένα root Pane δηλαδή ένα JPanel όπου μπορούμε να κάνουμε add components
- Η μέθοδος **setContentPane()** θέτει το root JPanel του JFrame
- Κάθε JPanel καθώς και το root ContentPane περιλαμβάνουν ένα Layout Manager ώστε τα διαχειρίζεται την τοποθέτηση των components με τρόπο που να μπορούν να είναι resizable και στη σωστή σχετική θέση όταν μεταβάλλεται το μέγεθος της οθόνης.
- Παρέχεται επίσης και ο null ή **Absolute Layout manager** που επιτρέπει την ρητή τοποθέτηση από τον χρήστη σε οποιοδήποτε σημείο του content pane των controls



Layout Managers



- Οι **Layout Managers** ρυθμίζουν την εμφάνιση των αντικειμένων στο JPanel. Υπάρχουν διάφοροι Layout Managers
- Το **FlowLayout** είναι το **default layout** του **JPanel** (left to right flow, by default όλα τα components τοποθετούνται στο κέντρο 'του container)



Παράδειγμα – Hello World

Java Swing

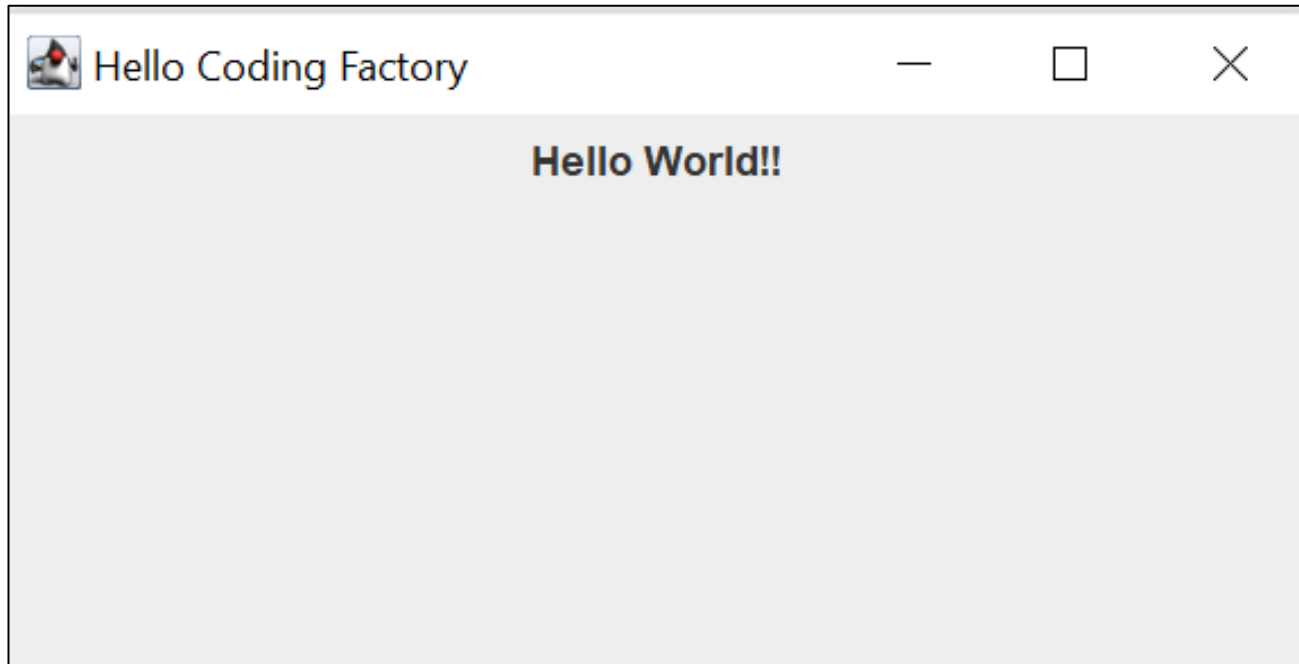
```
1 package gr.aueb.cf.jframes;
2
3 import javax.swing.*;
4
5 public class HelloSwing {
6     public static void main(String[] args) {
7         JFrame helloFrame = new JFrame("Hello Coding Factory");
8
9         helloFrame.setSize(400, 200);
10        helloFrame.setLocationRelativeTo(null);
11        helloFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12
13        JPanel contentPane = new JPanel();
14        contentPane.add(new JLabel("Hello World!!"));
15        helloFrame.setContentPane(contentPane);
16
17        helloFrame.setVisible(true);
18    }
19 }
```

Αφού δημιουργήσουμε ένα **JFrame** με τίτλο στη συνέχεια ρυθμίζουμε το **defaultClose Operation**, ώστε όταν πατάμε το close button του παραθύρου να κλείνει η εφαρμογή, το **size** σε pixels, το **Location** του παραθύρου ως προς την οθόνη (null) centered και τέλος πρέπει να κάνουμε **visible** (true) το παράθυρο αλλιώς το default είναι false



Αποτέλεσμα

Java Swing

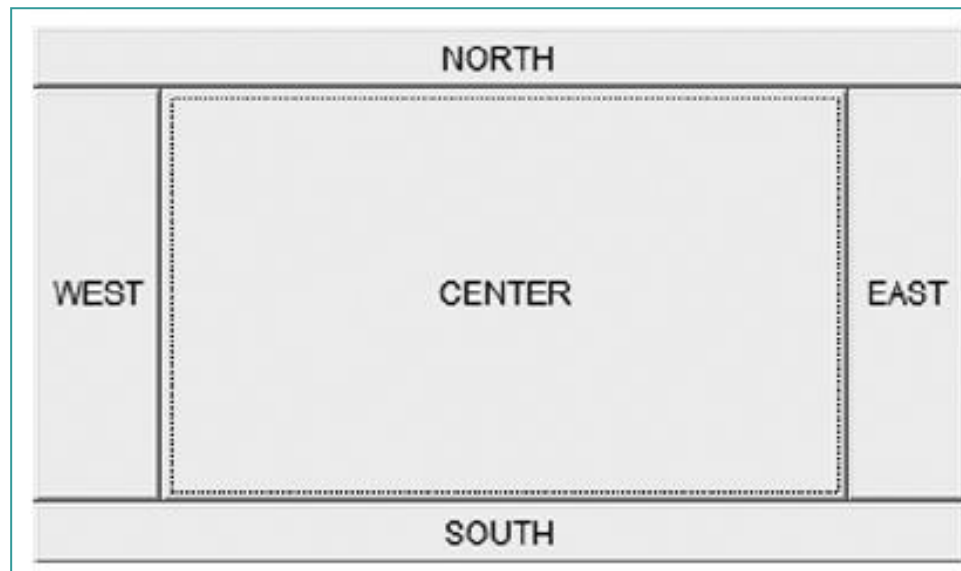


- Η default περιοχή του BorderLayout είναι το CENTER



Border Layout

- Το Border Layout χωρίζει την περιοχή σε 5 μέρη όπως παρακάτω
- Το Border Layout δημιουργεί ένα container, που τοποθετεί και κάνει resize τα components που περιέχει σε πέντε (5) περιοχές: north, south, east, west, και center. Κάθε περιοχή μπορεί να περιέχει μόνο ένα component και προσδιορίζεται από μία σταθερά: **NORTH, SOUTH, EAST, WEST, και CENTER.**





Event-driven Programming

Java Swing

- Τα προγράμματα με Γραφική Διεπαφή (GUI) ακολουθούν το μοντέλο **προγραμματισμού βασισμένου σε γεγονότα**
- **Γεγονός (Event)** είναι κάθε πράξη του χρήστη (π.χ. mouse clicked) ή του Συστήματος / Εφαρμογής (π.χ. Window Opened) που όταν συμβεί τυγχάνει διαχείρισης από ένα ειδικό κομμάτι κώδικα που λέγεται **Event-Handler** ή **Event Listener**



ActionEvent - ActionListener

Java Swing

- Οι Event Listeners είναι συνδεδεμένοι (registered) με ένα control, π.χ. **JFrame**, **JButton**, κλπ
- Αν –για παράδειγμα- έχουμε ένα **JButton** και ο χρήστης πατήσει κλικ, τότε δημιουργείται ένα event που ονομάζεται **ActionEvent** (`java.awt.event.ActionEvent`)
- Το **ActionEvent** χειρίζεται η μέθοδος **actionPerformed(ActionEvent evt)**; του Functional Interface **ActionListener** (`java.awt.event.ActionListener`)



Events και Πολλαπλά Threads

- Για να μπορεί μία εφαρμογή GUI να επεξεργάζεται τα διάφορα events που μπορεί να δημιουργούνται από διάφορα threads με συνέπεια, η Java παρέχει τον μηχανισμό **EDT (Event Dispatch Thread)** και το `awt.java.EventQueue` ή το `javax.swing.SwingUtilities`
- Το **EDT** είναι ένα μοναδικό thread που διαχειρίζεται μία μοναδική ουρά γεγονότων (event queue) όπου όλα τα events που δημιουργούνται από οποιοδήποτε άλλο thread εισάγονται στην ουρά και εξυπηρετούνται σε βάση FIFO (First In First Out) και επομένως δεν υπάρχουν race conditions



Swing Threads (1)

- Στο Swing υπάρχουν τρεις τύποι thread
 - Το initial thread, δηλαδή το thread της μεθόδου `main()`
 - Το Event Dispatch Thread (EDT) , όπου (πρέπει να) εκτελείται όλο το event handling μιας και σχεδόν όλες οι component methods του Swing δεν είναι thread-safe
 - Worker threads (ή background threads) όπου εκτελούνται time-consuming threads ώστε το EDT να μην έχει καθυστερήσεις μιας και κατά τη διάρκεια εκτέλεσης ενός task του EDT τα άλλα tasks περιμένουν στην ουρά (EDT Queue)



Swing Threads (2)

- Το initial thread απλά δημιουργεί το EDT με την utility κλάση `SwingUtilities` ή **`EventQueue`** και τις static μεθόδους **`invokeLater`** (schedule the task και returns) και **`invokeAndWait`** (περιμένει να τελειώσει το task και returns)
- Και οι δύο μέθοδοι παίρνουν ως παράμετρο ένα `Runnable` και υλοποιούν την μέθοδο `run()`



HelloSwing με EDT (1)

```
1 package gr.aueb.cf.jframes2;
2
3 import javax.swing.*;
4
5 public class MFrame extends JFrame {
6     public MFrame() {
7         this.setSize(500, 250);
8         this.setLocationRelativeTo(null);
9         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
10        this.setTitle("Hello AUEB Coding Factory");
11    }
12 }
```

- Ορίζουμε το *MFrame* ως ανεξάρτητη **JFrame** κλάση κάνοντας extends το **JFrame**



HelloSwing με EDT (2)

```
1 package gr.aueb.cf.jframes2;
2
3 import java.awt.*;
4
5 public class Main {
6
7     public static void main(String[] args) {
8
9         EventQueue.invokeLater(new Runnable() {
10             @Override
11             public void run() {
12                 new MFrame().setVisible(true);
13             }
14         });
15     }
16 }
```

- Σε άλλη κλάση που έχει μία **main()** δημιουργούμε ένα **Runnable** ως ανώνυμη κλάση που κάνει Override την **run** – το **Runnable** είναι functional Interface-
- Η **run()** απλά δημιουργεί με **new** ένα αντικείμενο **Mframe** και το κάνει visible



Κλάση JOptionPane

Java Swing

- Εμφάνιση pop-up μηνυμάτων
 - `JOptionPane.showMessageDialog()`
 - `JOptionPane.showInputDialog()`
 - `JOptionPane.showConfirmDialog()`
 - `JOptionPane.showOptionDialog()`



ActionListener & JOptionPane

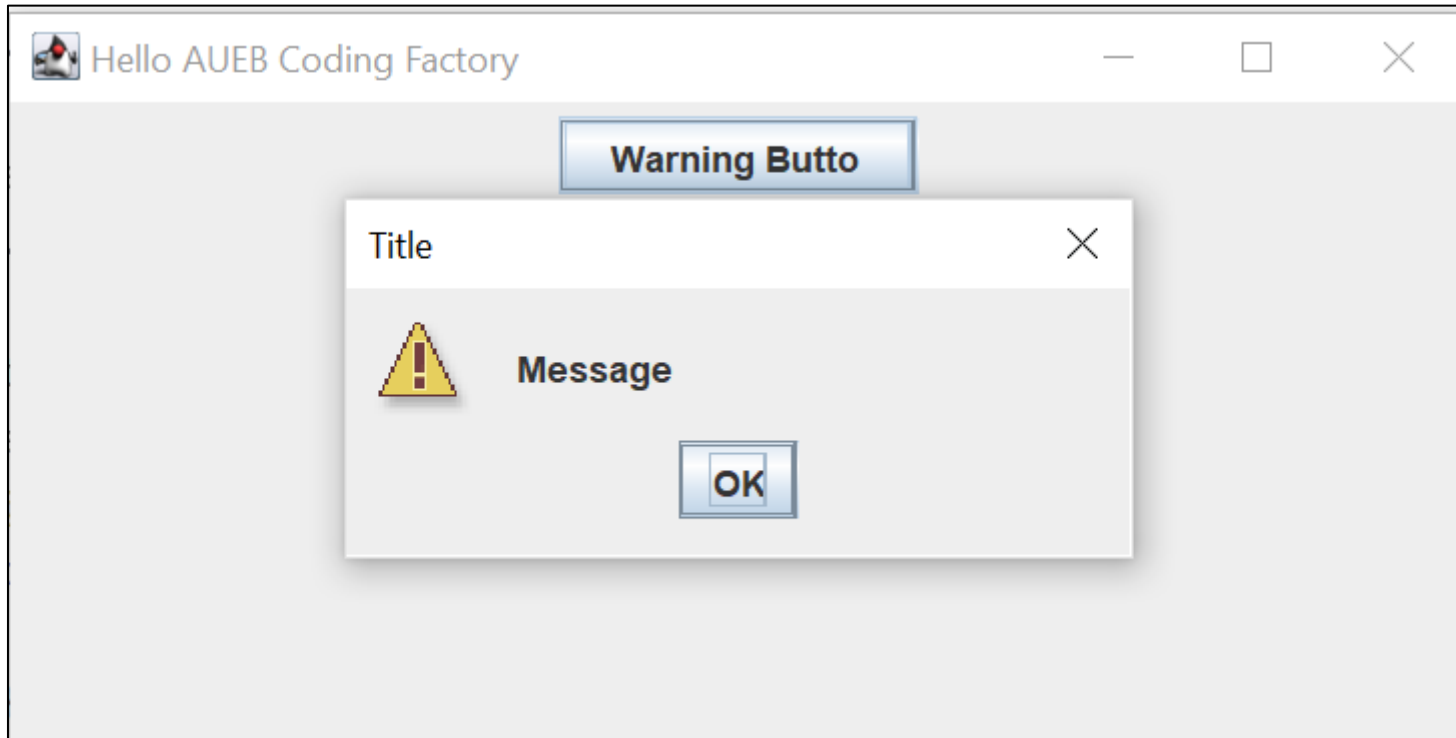
```
3 import javax.swing.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 public class MFrame extends JFrame {
8     public MFrame() {
9         this.setSize(500, 250);
10        this.setLocationRelativeTo(null);
11        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
12        this.setTitle("Hello AUEB Coding Factory");
13
14        JPanel contentPane = new JPanel();
15        this.setContentPane(contentPane);
16
17        JButton btn = new JButton("Warning Button ");
18        btn.addActionListener(new ActionListener() {
19            @Override
20            public void actionPerformed(ActionEvent e) {
21                JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.WARNING_MESSAGE);
22            }
23        });
24
25        contentPane.add(btn);
26    }
27 }
```

- Με *showMessageDialog* εκτυπώνουμε ένα popup message



Αποτέλεσμα

Java Swing





Confirm Dialog

```
17 JButton btn = new JButton("Warning Button ");
18 btn.addActionListener(new ActionListener() {
19     @Override
20     public void actionPerformed(ActionEvent e) {
21         JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.WARNING_MESSAGE);
22     }
23 });
24
25 JButton btn2 = new JButton("Confirm Button ");
26 btn2.addActionListener(new ActionListener() {
27     @Override
28     public void actionPerformed(ActionEvent e) {
29         int answer = JOptionPane.showConfirmDialog(null, "Message", "Title", JOptionPane.YES_NO_OPTION);
30         if (answer == JOptionPane.YES_OPTION) {
31             JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.INFORMATION_MESSAGE);
32         } else {
33             JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.ERROR_MESSAGE);
34         }
35     }
36 });
37
38 contentPane.add(btn);
39 contentPane.add(btn2);
40 }
```

- Επιστρέφει 0 (YES_OPTION) ή 1 (NO_OPTION)



Lambdas

- Ο ActionListener είναι Functional Interface και μπορούμε συνεπώς να χρησιμοποιήσουμε Lambdas

```
JButton btn = new JButton("Warning Button ");
btn.addActionListener(e -> JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.WARNING_MESSAGE));

JButton btn2 = new JButton("Confirm Button ");
btn2.addActionListener(e -> {
    int answer = JOptionPane.showConfirmDialog(null, "Message", "Title", JOptionPane.YES_NO_OPTION);
    if (answer == JOptionPane.YES_OPTION) {
        JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(null, "Message", "Title", JOptionPane.ERROR_MESSAGE);
    }
});

contentPane.add(btn);
contentPane.add(btn2);
```



Eclipse / WindowBuilder

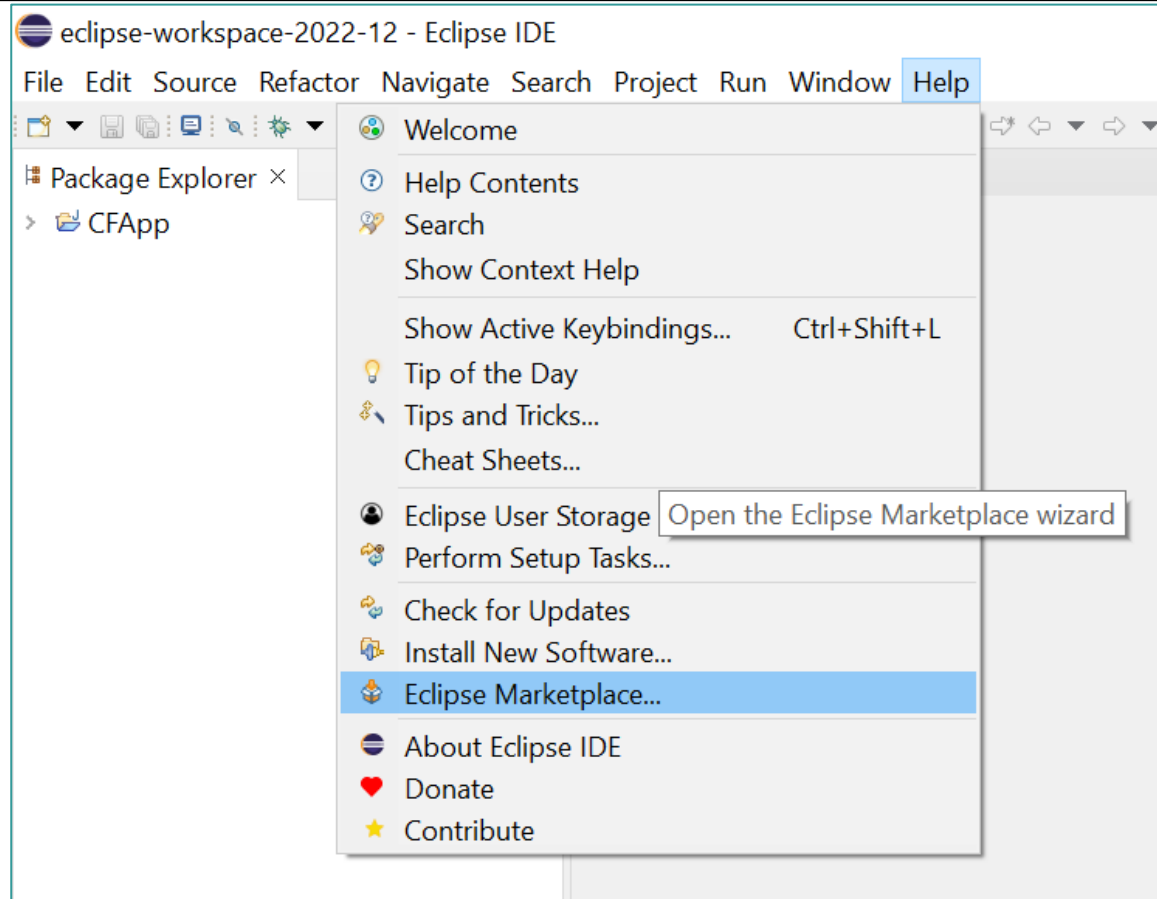
Java Swing

- Συνήθως δημιουργούμε γραφικές διεπαφές με συστήματα **WYSIWYG (What You See Is What You Get)** δηλαδή συστήματα που μας επιτρέπουν να σχεδιάζουμε Γραφικές Διεπαφές όπως θα σχεδιάζαμε σε ένα σχεδιαστικό πρόγραμμα γραφικών (π.χ. Ζωγραφική των Windows)
- Τα συστήματα αυτά μας επιτρέπουν να σχεδιάζουμε την γραφική διεπαφή και καθώς σχεδιάζουμε δημιουργούν αυτόματα κώδικα που βασίζεται στον σχεδιασμό μας
- Ένα τέτοιο σύστημα είναι το **WindowBuilder** του **Eclipse** το οποίο θα χρησιμοποιήσουμε αφού πρώτα εγκαταστήσουμε από το μενού Help/EclipseMarketplace του Eclipse



WindowBuilder στο Eclipse

Java Swing



- **Help / Eclipse-Marketplace.** Στο Find εισάγουμε **Window Builder**. Κλικ στο Install



Install Window Builder (1)

Java Swing

The screenshot shows the Eclipse Marketplace window. At the top, it says "Eclipse Marketplace" with a logo. Below that, it says "Eclipse Marketplace" and "Select solutions to install. Press Install Now to proceed with installation. Press the 'more info' link to learn more about a solution." There is a search bar with "Window Builder" entered. Below the search bar, there are tabs for "Recent", "Popular", "Favorites", "Installed", and "Giving IoT an Edge". The search results show "WindowBuilder Current" by Eclipse Foundation, EPL. It describes WindowBuilder as a tool for creating Java GUI applications. There are 899 stars and 907K installs (14.850 last month). An "Install" button is visible.

Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press **Install Now** to proceed with installation.
Press the "more info" link to learn more about a solution.

Search Recent Popular Favorites Installed Giving IoT an Edge

Find: Window Builder All Markets All Categories Go

WindowBuilder Current

WindowBuilder is composed of SWT Designer and Swing Designer and makes it very easy to create Java GUI applications without spending a lot of time writing code.... [more info](#)

by Eclipse Foundation, EPL

[SWT](#) [swing](#) [wysiwyg](#) [graphical](#) [WindowBuilder](#)

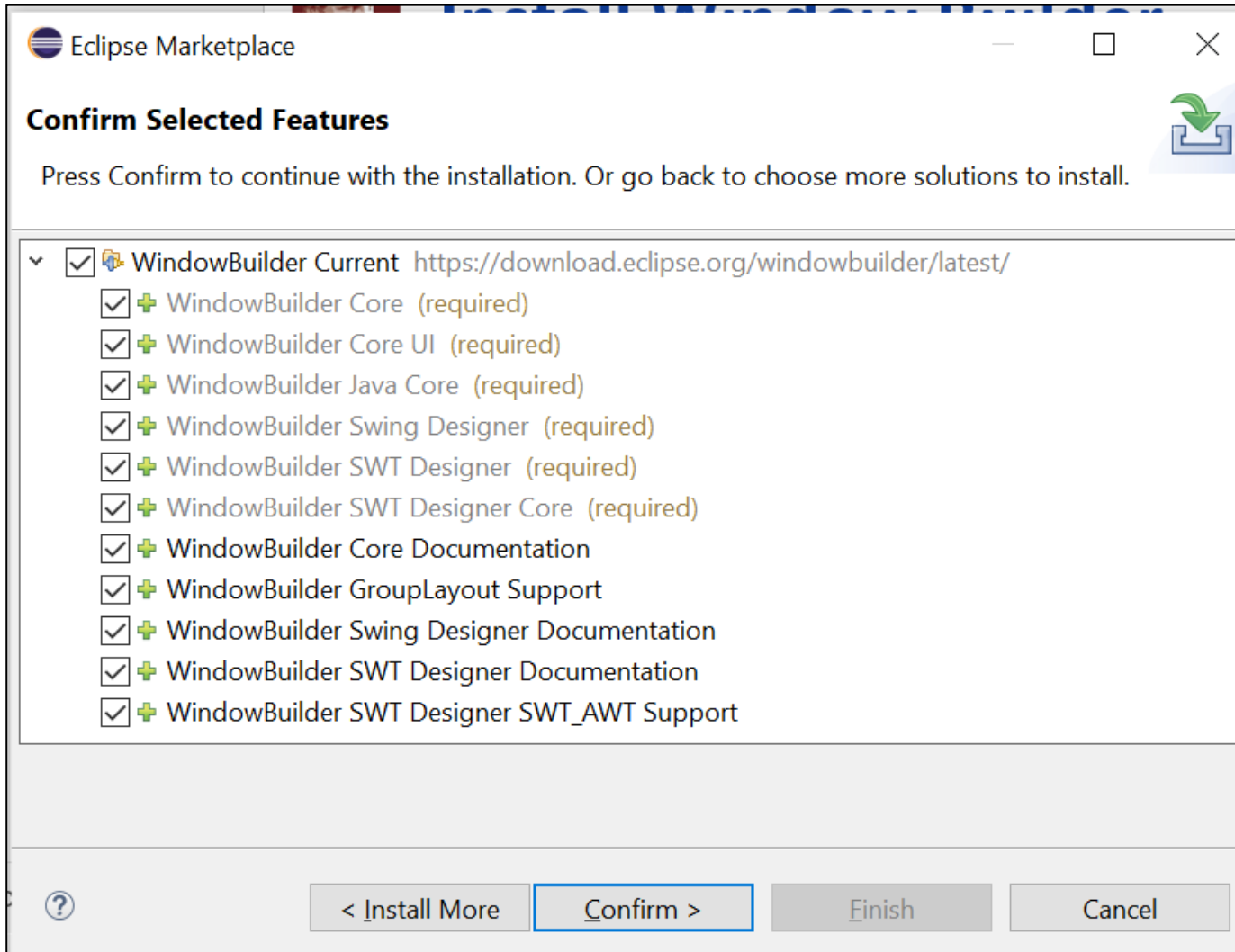
★ 899 Installs: **907K** (14.850 last month) **Install**

- Window Builder Install



Install Window Builder (2)

Java Swing



- Confirm



Install Window Builder (3)

Java Swing

- Accept
τα
terms
του
License
Agreement

Eclipse Marketplace

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

- > Eclipse Foundation Software User Agreement
- > Eclipse Foundation Software User Agreement

License text:

Eclipse Foundation Software User Agreement

November 22, 2017

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED

☒ I accept the terms of the license agreements

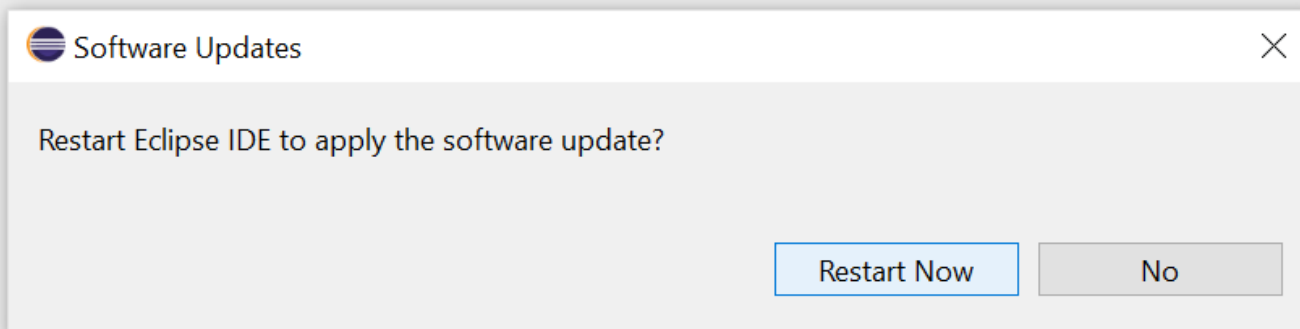
☐ I do not accept the terms of the license agreements

< Back Next > **Finish** Cancel



Install Window Builder (4)

Java Swing



- Restart Now



SwingPro

Java Swing

New Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location: [Browse...](#)

JRE

☐ Use an execution environment JRE:

☒ Use a project specific JRE:

☐ Use default JRE 'jdk11.0.10_9' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

Module

☐ Create module-info.java file

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

- Νέο project με όνομα SwingPro



Swingapp package

Java Swing

New Java Package

Java Package
Create a new Java package.

Creates folders corresponding to packages.

Source folder: [Browse...](#)

Name:

☐ [Create package-info.java](#)

☐ [Generate comments \(configure templates and default value \[here\]\(#\)\)](#)

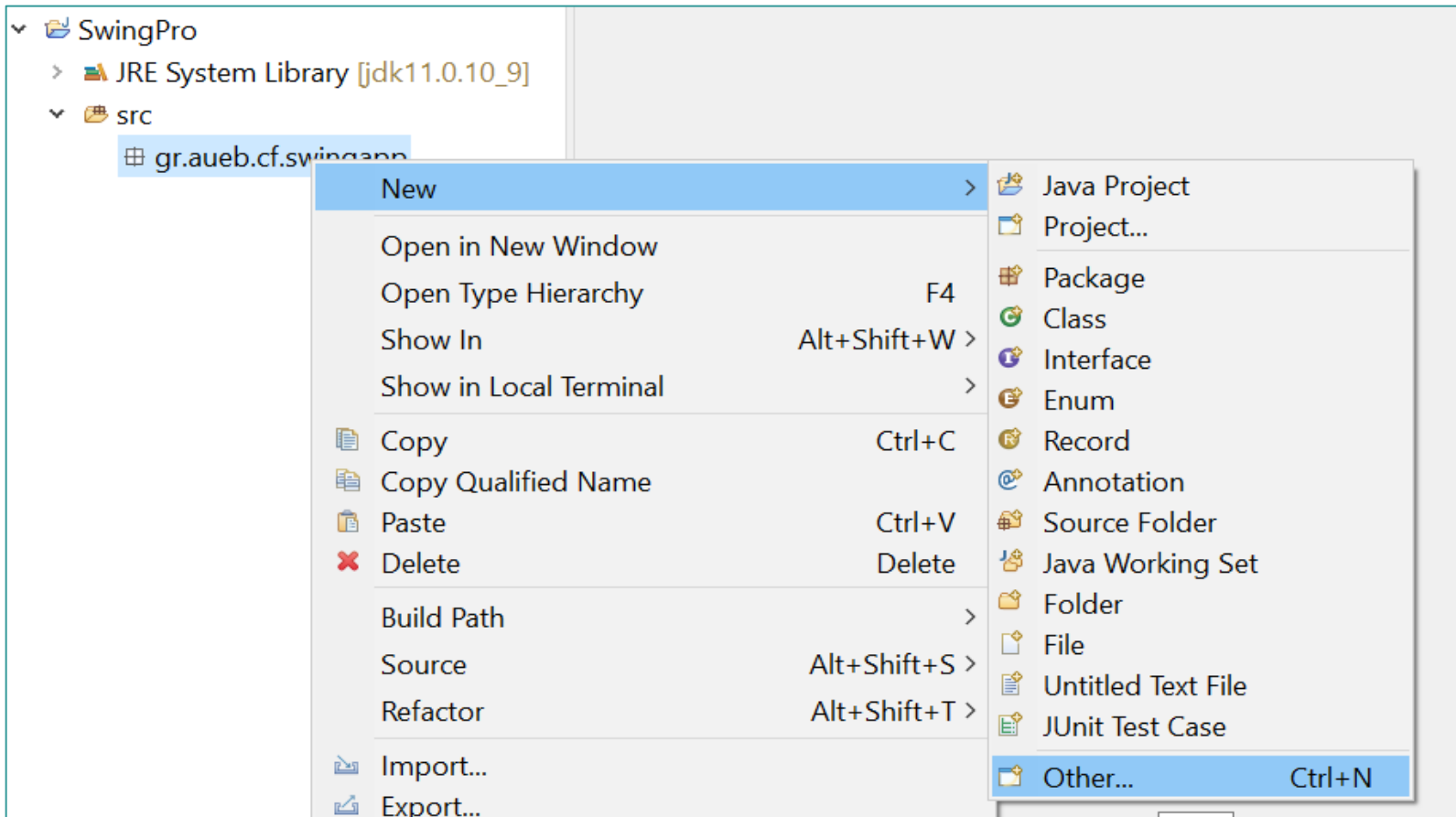
[?](#) [Finish](#) [Cancel](#)

- Néó package `gr.aueb.cf.swingapp`



Παράδειγμα με τον window builder του Eclipse

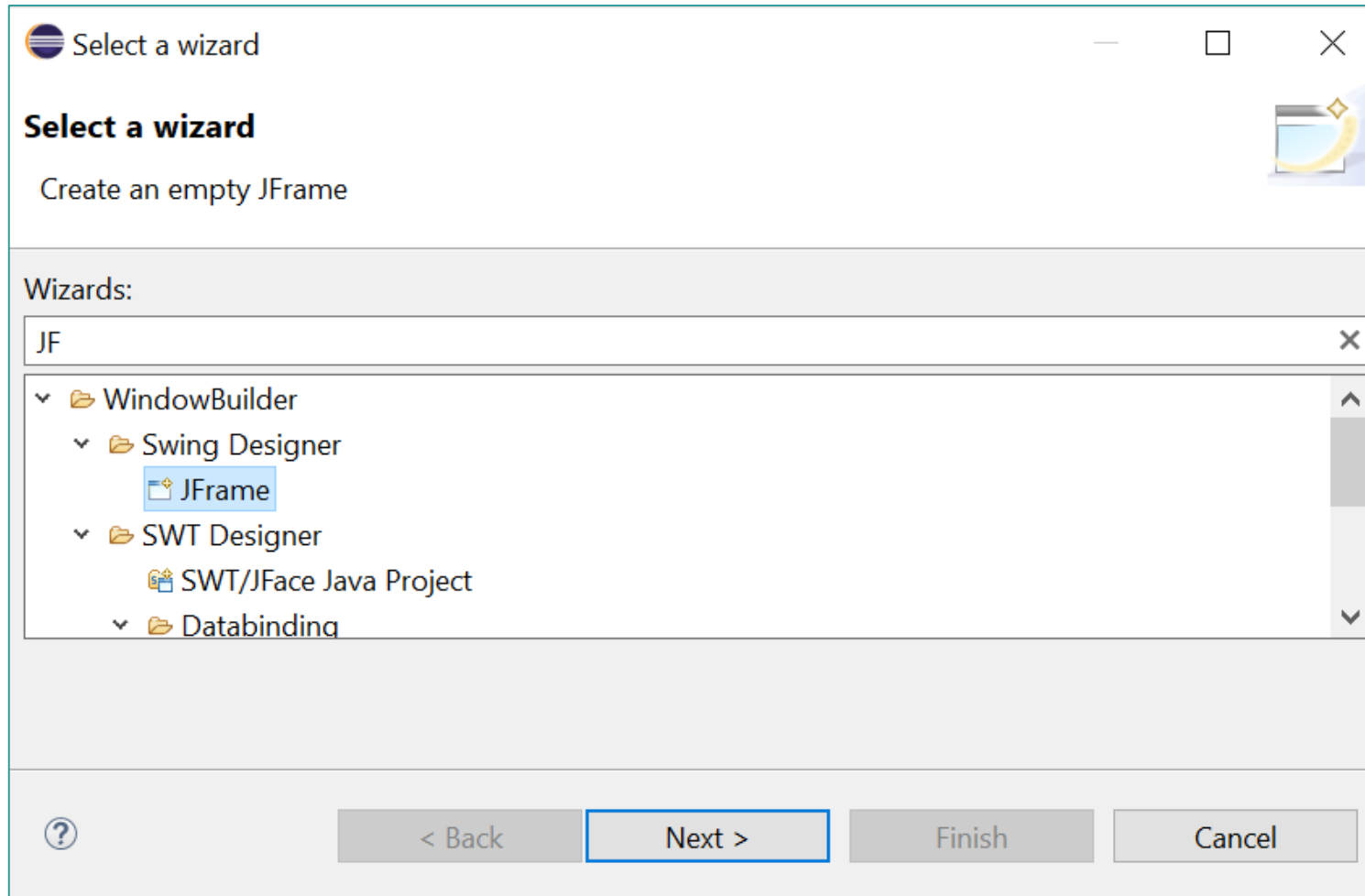
Java Swing



- Δεξί κλικ πάνω στο package / new και αντί για Class, πάμε στο **Other**. Αναζητούμε **JFrame**, το επιλέγουμε και πατάμε Next



JFrame






Δημιουργία JFrame (2)

Java Swing

- Δίνουμε το όνομα της Κλάσης

 New JFrame

Create JFrame

Create an empty JFrame.


Source folder:

Package:

Name:

Superclass:

☒ Use advanced template for generate JFrame





Source / Design

Java Swing

```
CodingFactoryFrame.java ×
1 package gr.aueb.cf.swingapp;
2
3 import java.awt.EventQueue;
4
5
6
7
8
9 public class CodingFactoryFrame extends JFrame {
10     private static final long serialVersionUID = 1L;
11     private JPanel contentPane;
12
13     public static void main(String[] args) {
14         EventQueue.invokeLater(new Runnable() {
15             public void run() {
16                 try {
17                     CodingFactoryFrame frame = new CodingFactoryFrame();
18                     frame.setVisible(true);
19                 } catch (Exception e) {
20                     e.printStackTrace();
21                 }
22             }
23         });
24     }
25
26     public CodingFactoryFrame() {
27         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28         setBounds(100, 100, 450, 300);
29         contentPane = new JPanel();
30         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
31
32         setContentPane(contentPane);
33     }
34
35 }
```

- Δημιουργούνται δύο μέρη:
 - **Source** – που περιέχει τον κώδικα
 - **Design** – που περιέχει το γραφικό μέρος
- Όταν σχεδιάζουμε στο Design δημιουργείται αυτόματα κώδικας στο Source



Design View

Έχουμε τα δύο components: **JFrame** και **JPanel** (contentPane) και ανάλογα με τι επιλέγουμε εμφανίζονται οι ιδιότητες

Επίσης, ανάλογα με το τι επιλέγουμε τονίζεται το περίγραμμα του αντικειμένου

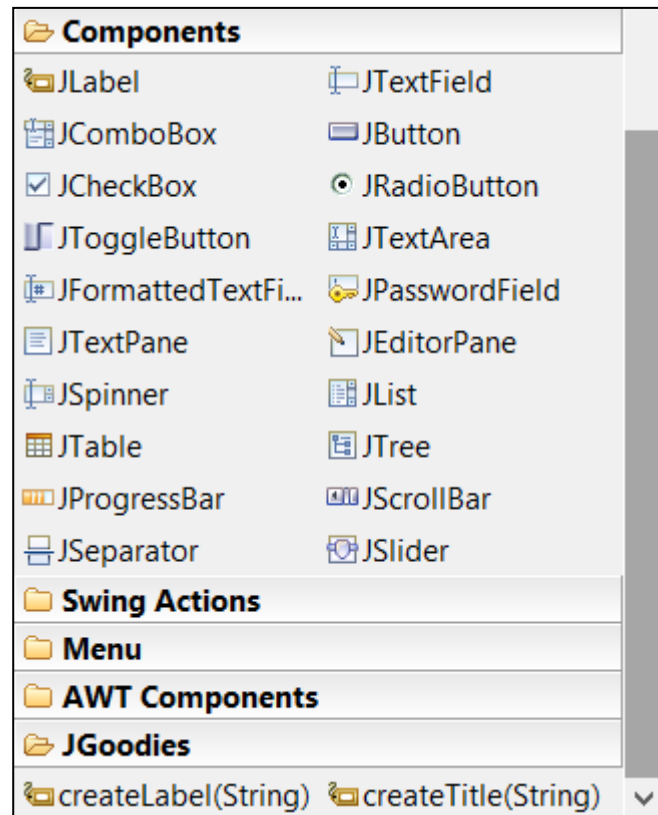
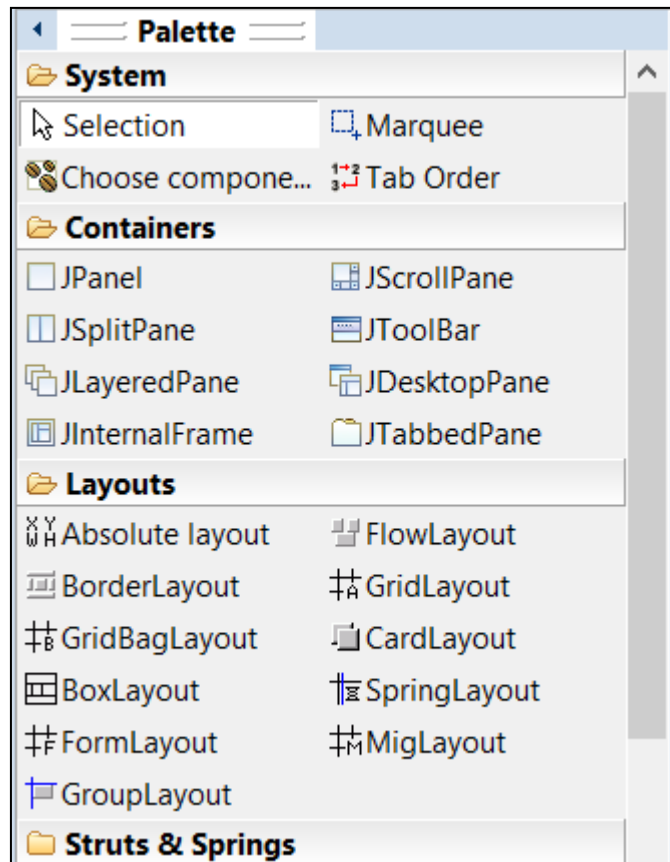
- Επίσης, ανάλογα με το τι επιλέγουμε τονίζεται το περίγραμμα του αντικειμένου
- Αριστερά, είναι η εργαλειοθήκη από όπου μπορούμε να επιλέξουμε controls και να το εισάγουμε με κλικ στο contentPane

<terminated> BuilderDemo [Java Application] C:\Program Files\Java\jdk1.8.0_40\bin\javaw.exe (19 Ιαν 2019 - 12:44:17 μ.μ.)



Η παλέτα του Design View

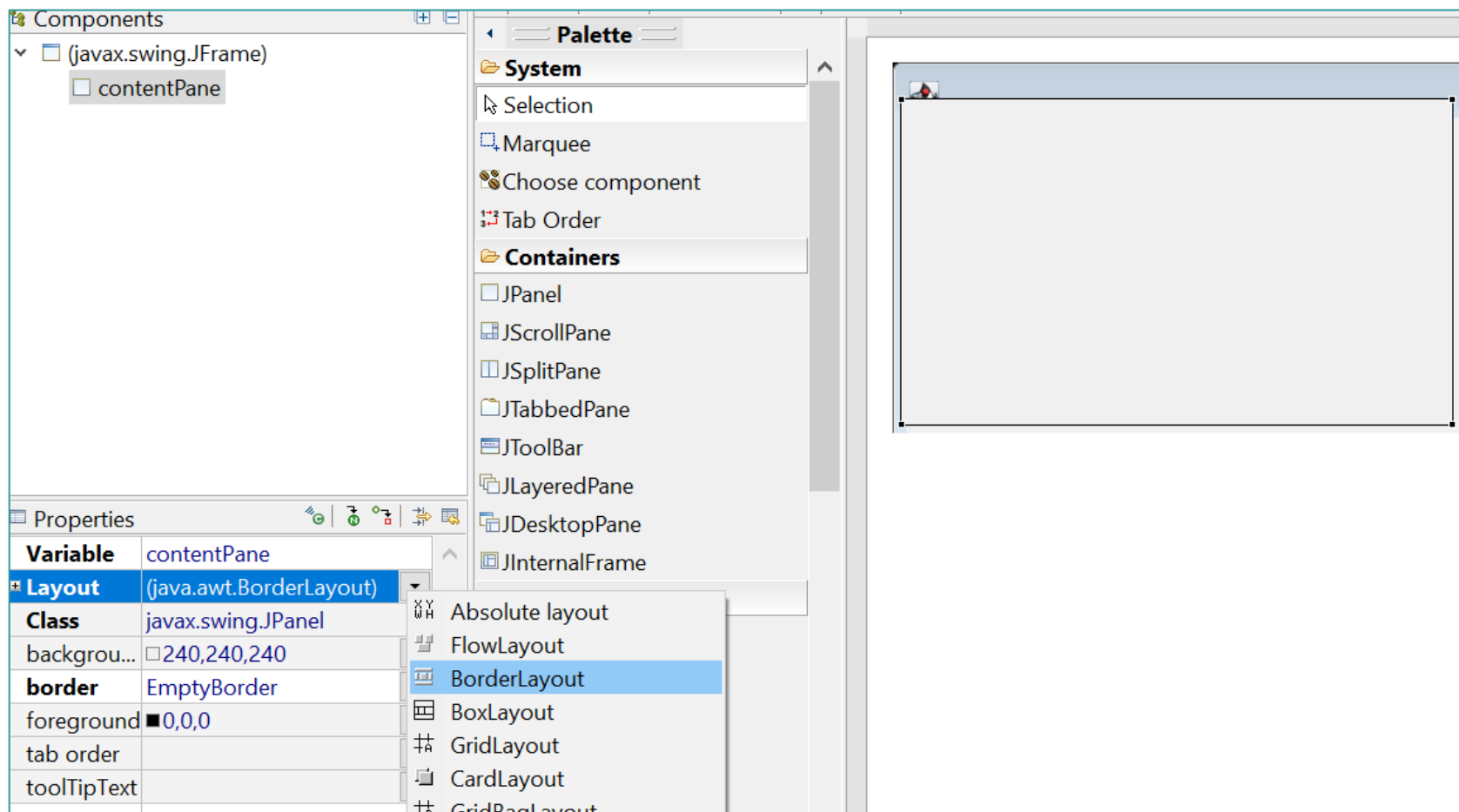
Java Swing



- Η Εργαλειοθήκη περιέχει components / controls / gadgets χωρισμένα σε διάφορες κατηγορίες
- Η κατηγορία που περιέχει τα controls που εισάγουμε στο JPanel είναι η Components



contentPane

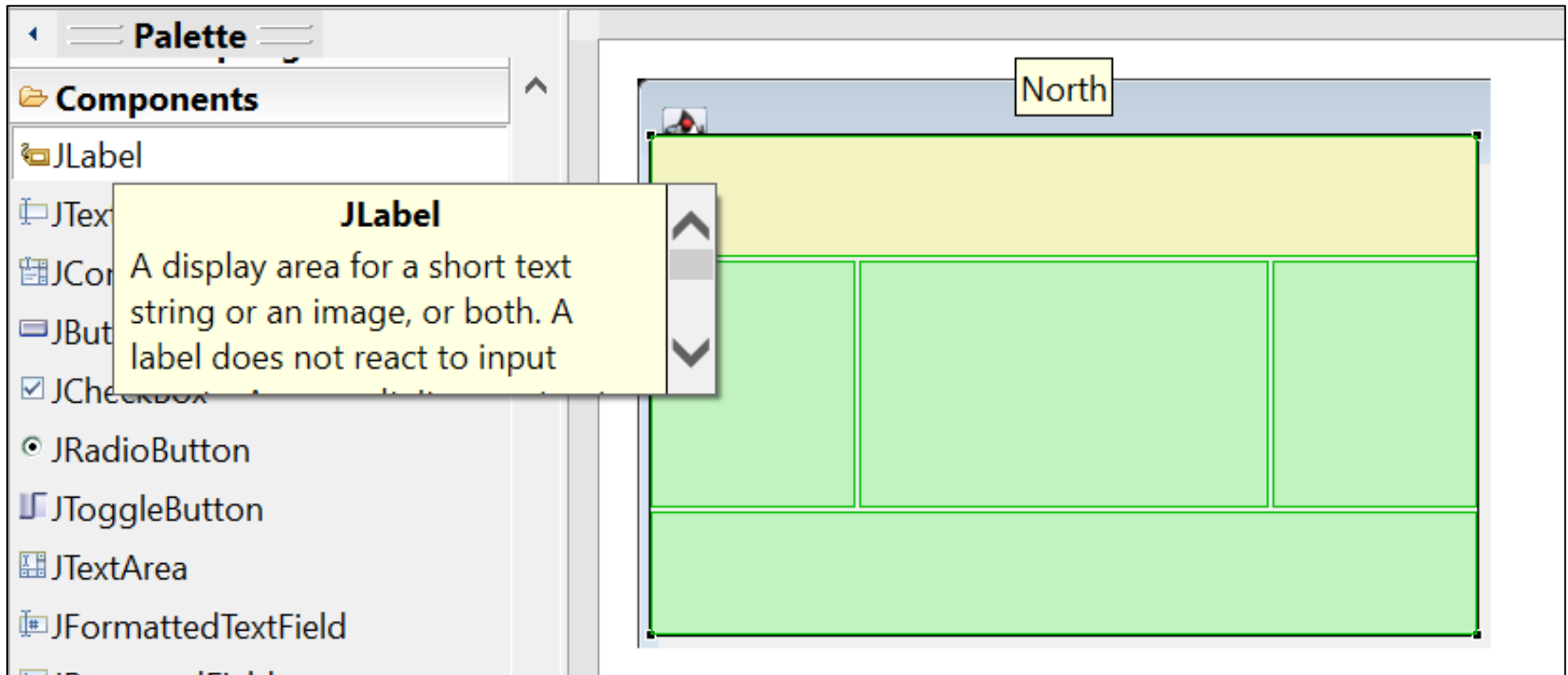


- Επιλέγουμε το contentPane και τονίζεται το Περίγραμμα ενώ οι ιδιότητες περιλαμβάνουν τα Layout, κλπ. ιδιότητες Επιλέγουμε **Border Layout**



Εισαγωγή JLabel στο North (1)

Java Swing



Κάνουμε κλικ στο JLabel και κλικ μέσα στην περιοχή North (δεν σύρουμε το ποντίκι)



Εισαγωγή JLabel στο North (2)

Java Swing

The screenshot shows the Swing Builder interface with the following components:

- Structure:** A tree view showing the component hierarchy. Under 'contentPane', there is a 'JLabel' component named 'lbnNewLabel'.
- Properties:** A table of properties for the selected 'JLabel' component.
- Palette:** A list of available Swing components and actions.
- Canvas:** A visual representation of the GUI, showing a window titled 'New label' with a text area.

Variable	lbnNewLabel
Constructor	(Constructor properties)
Constraints	North
Class	javax.swing.JLabel
background	240,240,240
displayedMn...	
enabled	<input checked="" type="checkbox"/>
font	Tahoma 13
foreground	0,0,0
horizontalAli...	LEADING
icon	LEFT
labelFor	CENTER
text	RIGHT
toolTipText	LEADING
	TRAILING

- Η εισαγωγή έχει γίνει στην περιοχή North
- Στα Components έχει προστεθεί επίσης το JLabel
- Το όνομα μεταβλητής μέσα στον κώδικα είναι lbnNewLabel

- Εισήχθη το JLabel. Όπως είναι επιλεγμένο πάμε στις ιδιότητες να κάνουμε ρυθμίσεις
- Το default `horizontalAlignment` είναι LEADING δηλαδή αριστερή στοίχιση



Εισαγωγή JLabel στο North (3)

Components

- (javax.swing.JFrame)
 - contentPane
 - IbICf - "AUEB Coding Factory"

Properties

Variable	contentPane
Layout	(java.awt.BorderLayout)
Class	javax.swing.JLabel
background	240,240,240
border	EmptyBorder
foreground	0,0,0
tab order	
toolTipText	

Palette

Components

- JLabel
- JTextField
- JComboBox
- JButton
- ☒ JCheckBox
- ☐ JRadioButton
- JToggleButton
- JTextArea
- JFormattedTextField
- JPasswordField
- JTextPane
- JEditorPane
- JSpinner
- JList
- JTable
- JTree
- JProgressBar
- JScrollBar
- JSeparator

AUEB Coding Factory

- Στις ιδιότητες αλλάζουμε: font, foreground, horizontalAlignment, και text



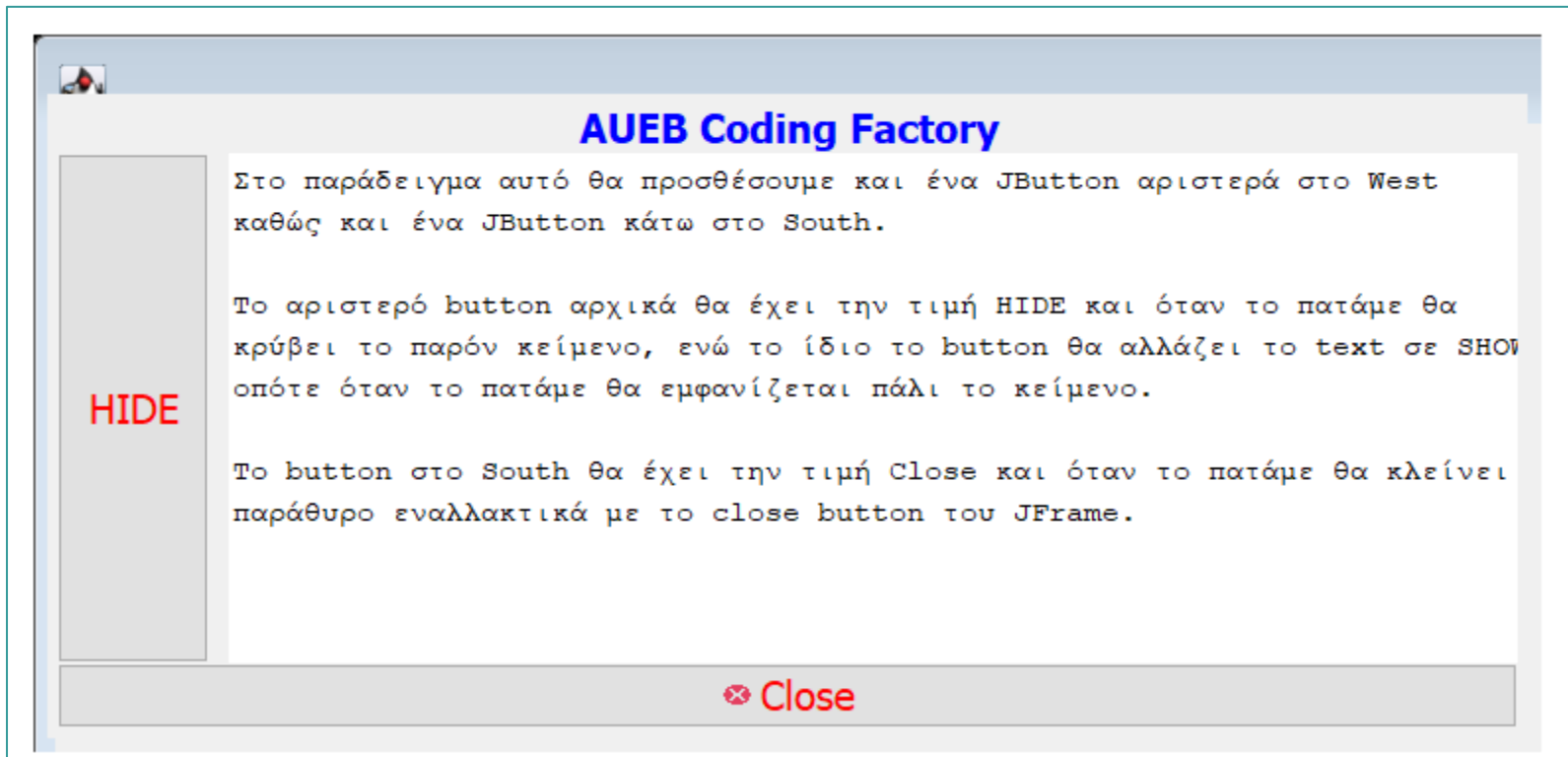
Εισαγωγή JTextArea

The screenshot shows the Java Swing IDE interface. On the left, the 'Components' palette lists various Swing components, with 'JTextArea' selected. Below it, the 'Properties' window shows the configuration for the selected 'textAreaCf' component, including its class 'javax.swing.JTextArea' and the 'editable' property set to 'false'. On the right, a preview window displays a 'JTextArea' control with the title 'AUEB Coding Factory'. The text inside the text area is in Greek, describing a coding factory example and its buttons.

- Εισάγουμε στο CENTER του BorderLayout ένα control JTextArea και εισάγουμε στην ιδιότητα text το κείμενο που φαίνεται στην εικόνα. Το **toolTipText** (εμφανίζεται όταν βάζουμε το ποντίκι πάνω στο κείμενο) έγινε "Περιγραφή Coding Factory"
- Το editable έγινε false ώστε να μην μπορεί ο χρήστης να αλλάξει το κείμενο δεδομένου ότι το JTextArea δεν είναι JLabel που δεν είναι editable



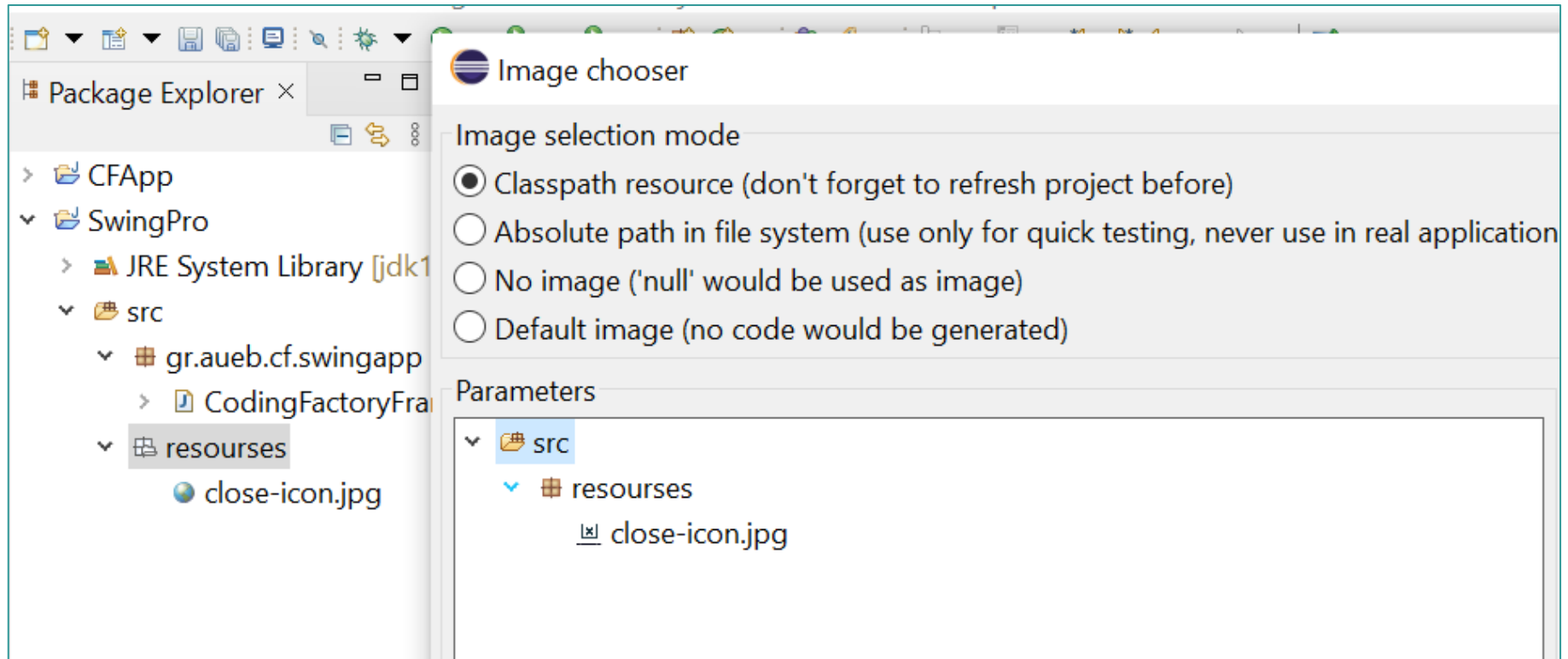
Εισαγωγή JButton (1)



- Εισάγουμε δύο JButton στο West και στο South αντίστοιχα και στο μεν πρώτο αλλάζουμε το text σε **HIDE** και το Foreground σε red, στο άλλο αλλάζουμε το text σε **Close** και προσθέτουμε στο icon ένα εικονίδιο (**16x16 pixels**) που βρήκαμε με αναζήτηση στο Google



Εικόνες και άλλα resources



- Εικόνες και άλλα **resources** μπορούμε να τα εισάγουμε σε ένα φάκελο resources για να είναι μέρος του project και να μπορούν να γίνουν distribute



Εισαγωγή JButton (2)

- Στο `contentPane` αλλάζουμε επίσης το `hgap` σε 10 ώστε να υπάρχει οριζόντιο διάστημα μεταξύ `JButton` αριστερά και κειμένου
- Επίσης, αλλάξαμε τα ονόματα των μεταβλητών των `buttons` σε `btnHideShow` και `btnClose`



Δημιουργία ActionListener

Java Swing

AUEB Coding Factory

Στο παράδειγμα αυτό θα προσθέσουμε και ένα JButton αριστερά στο West καθώς και ένα JButton κάτω στο South.

Το αριστερό button αρχικά θα έχει την τιμή HIDE και όταν το πατάμε θα κρύβει το παρόν κείμενο, ενώ το ίδιο το button θα αλλάζει το text σε SHOW οπότε όταν το πατάμε θα εμφανίζεται πάλι το κείμενο.

Close και όταν το πατάμε θα κλείνει τον JFrame.

HIDE

Context Menu:

- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete
- Test/Preview...
- Refresh (F5)
- Add event handler**
 - action** → **actionPerformed**
 - ancestor

Στο button HIDE δημιουργούμε **ActionListener** με δεξί κλικ πάνω στο button και μετά από το μενού Add event handler / action επιλέγουμε actionPerformed



Υλοποίηση Listeners (1)

Java Swing

```
60 btnHideShow.addActionListener(new ActionListener() {  
61     public void actionPerformed(ActionEvent e) {  
62         if (btnHideShow.getText() == "HIDE") {  
63             textAreaCf.setVisible(false);  
64             btnHideShow.setText("SHOW");  
65             btnHideShow.setForeground(Color.GREEN);  
66         } else {  
67             textAreaCf.setVisible(true);  
68             btnHideShow.setText("HIDE");  
69             btnHideShow.setForeground(Color.RED);  
70         }  
71     }  
72 });
```

- Αν το κείμενο `.getText()` του `btnHideButton` είναι "HIDE", τότε κάνουμε το `textArea` non-visible, το χρώμα του `btnHideShow` πράσινο και το κείμενο του `btnHideShow` "SHOW"
- Αλλιώς, αν το κείμενο του `btnHideShow` δεν είναι "HIDE" (αλλά "SHOW") τότε κάνουμε ανάλογες αλλαγές



Υλοποίηση Listeners (2)

Java Swing

```
79 btnClose.addActionListener(new ActionListener() {  
80     public void actionPerformed(ActionEvent e) {  
81         System.exit(0);  
82     }  
83 });
```

- Το ίδιο κάνουμε και στο Close Button, εισάγουμε ActionListener ώστε στη συνέχεια να εισάγουμε εντολές μέσα στη μέθοδο actionPerformed() που θα κλείνουν την εφαρμογή (System.exit(0)) όταν ο χρήστης πατάει κλικ πάνω στο Close button



Μέγεθος και Στοιχίση

Java Swing

```
23 public static void main(String[] args) {
24     EventQueue.invokeLater(new Runnable() {
25         public void run() {
26             try {
27                 CodingFactoryFrame frame = new CodingFactoryFrame();
28                 frame.setVisible(true);
29             } catch (Exception e) {
30                 e.printStackTrace();
31             }
32         }
33     });
34 }
35
36 public CodingFactoryFrame() {
37     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38     setBounds(0, 0, 850, 400);
39     setLocationRelativeTo(null);
40     contentPane = new JPanel();
41     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
42 }
```

- Αλλάζουμε τα bounds για να είναι visible το textArea της εφαρμογής καθώς και θέτουμε setLocationRelativeTo(null) για να στοιχίσουμε στο κέντρο της οθόνης



Το τελικό αποτέλεσμα

Java Swing

