

Programmazione in Java e gestione della grafica

Lezione 19

Parliamo di ...

- Componenti della

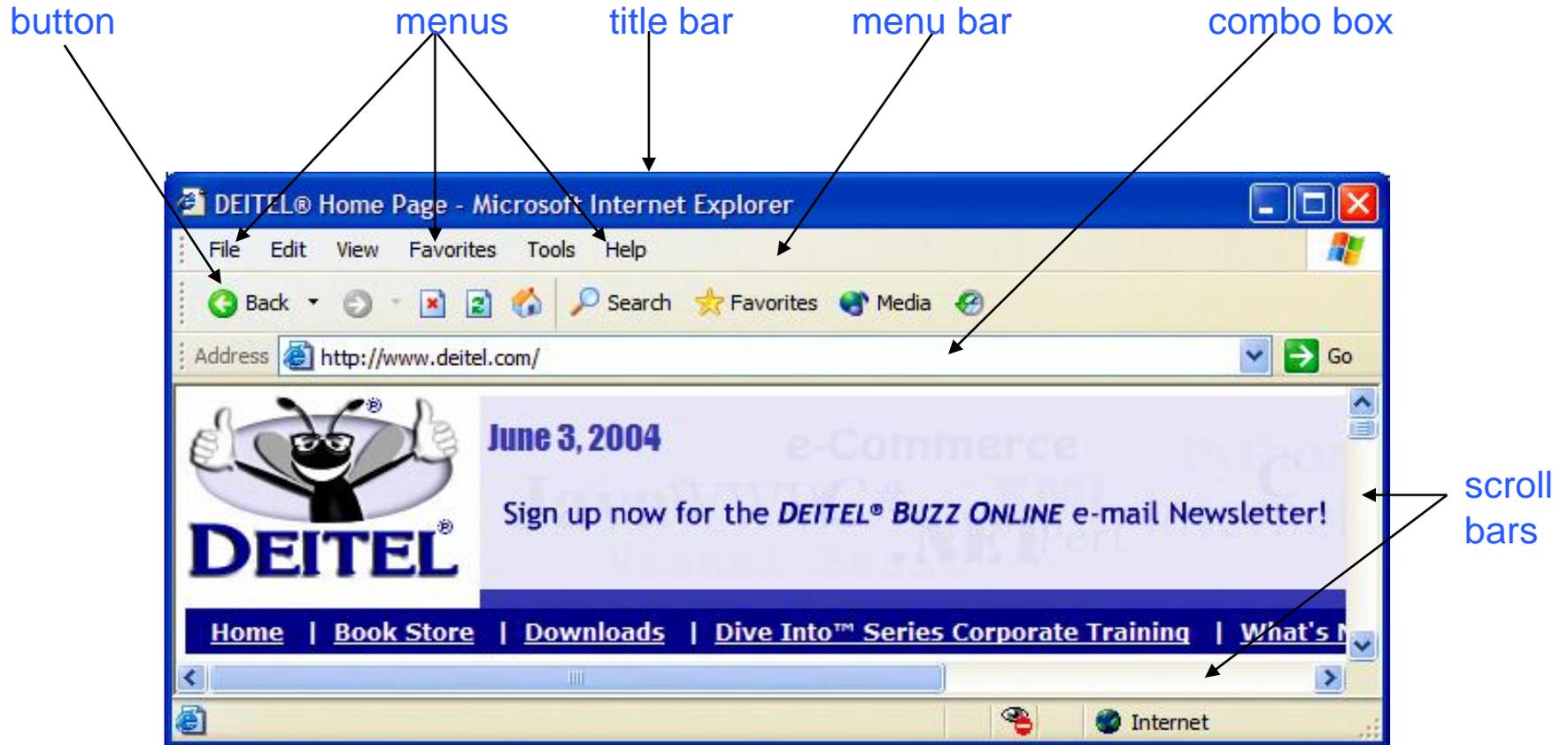
GUI (Graphics User Interface)

OBIETTIVI

- **Imparare** i principi su cui sono basate le GUI
- **Costruire** delle GUI e gestire gli eventi generate dall'interazione dell'utente con le GUI.
- **Conoscere** i packages che contengono component GUI, le classi per la gestione degli eventi e le interface.
- **Creare e manipolare** *buttons, labels, lists, text fields, panels.*
- **Gestire** gli eventi del mouse e della tastiera
- **Usare** *layout managers* per organizzare le component GUI

GUI (Graphical user interface)

- Presenta un meccanismo *user-friendly* per interagire con un'applicazione
- Spesso contiene:
 - barra del titolo
 - Barra dei menu
 - Pulsanti (buttons)
 - Finestre (combo boxes)
- E' costruita a partire da componenti GUI



Internet Explorer con GUI components.

Input/Output con JOptionPane

- **Dialog boxes**
 - Usati dalle applicazioni per interagire con l'utente
 - Si trovano nella classe Java JOptionPane
 - Esistono di tipo **input** e di tipo **message**

Addition.java

Tipo di Message dialog	Icona	Descrizione
<code>ERROR_MESSAGE</code>		Un dialogo che indica la presenza di un errore
<code>INFORMATION_MESSAGE</code>		Un dialogo con un messaggio informativo
<code>WARNING_MESSAGE</code>		Un dialogo che avverte di un potenziale problema
<code>QUESTION_MESSAGE</code>		Un dialogo che pone una domanda: di solito richiede una risposta come cliccare un pulsante YES o NO.
<code>PLAIN_MESSAGE</code>	Nessuna icona	Un dialogo che contiene un messaggio ma non icone.

Constanti `static` di **message dialogs**

Componenti Swing della GUI

- Dichiarate nel package `javax.swing`
- Molte sono pure component Java (scritte completamente in Java)
- Fanno parte della *Java Foundation Classes (JFC)* cioè le libreria di Java per lo sviluppo multiplatforma.

Component	Description
JLabel	Displays uneditable text or icons.
TextField	Enables user to enter data from the keyboard. Can also be used to display editable or uneditable text.
Button	Triggers an event when clicked with the mouse.
CheckBox	Specifies an option that can be selected or not selected.
ComboBox	Provides a drop-down list of items from which the user can make a selection by clicking an item or possibly by typing into the box.
List	Provides a list of items from which the user can make a selection by clicking on any item in the list. Multiple elements can be selected.
Panel	Provides an area in which components can be placed and organized. Can also be used as a drawing area for graphics.

Componenti GUI di base

Swing vs. AWT

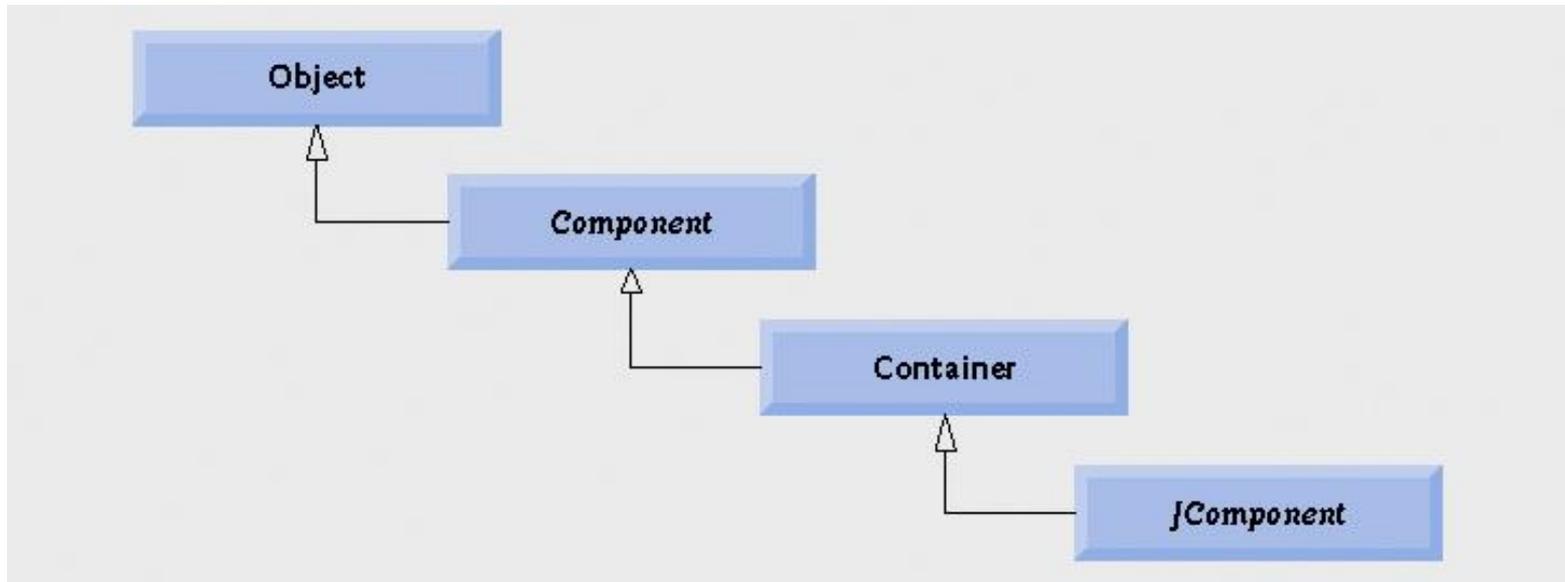
- Abstract Window Toolkit (AWT)
 - Precedente a Swing
 - Dichiarata nel package `java . awt`
 - Cambiando piattaforma ha diversi **look-and-feel** (aspetto e modo di interazione con l'utente). Riprende il look-and-feel del Sistema operativo su cui è eseguito

Obiettivo portabilità

- Le componenti di Swing sono implementate in Java e sono più portabili di quelle originali AWT (basate sulle component GUI della piattaforma sottostante)

Componenti leggere vs. pesanti

- Componenti **leggere**
 - Non legate direttamente alle component GUI ma supportate dalla piattaforma sottostante
- Componenti **pesanti**
 - Legate direttamente alla piattaforma locale
 - Componenti AWT
 - Componenti Swing



Superclassi di molte component Swing.

Visualizzare testo e immagini in una finestra

- Class JFrame
 - La maggior parte delle finestre sono una istanza o una sottoclasse di JFrame
 - Componenti di una finestra sono:
 - Barra del titolo
 - Pulsanti per minimizzare /massimizzare/chiudere l'applicazione.
 - Per visualizzare del testo informativo uso delle label create con la classe JLabel

Alcuni metodi di JFrame

- `setDefaultCloseOperation`
 - Prescrive come reagisce l'applicazione quando l'utente clicca sul pulsante "close"
- `setSize`
 - Specifica altezza e larghezza della finestra
- `setVisible`
 - Determina se mostrare la finestra (`true`) o (`false`)

TextPad - C:\Users\Dora\Documents\Didattica\AA12-13\Java\Lezioni\Esercizi_19\LabelTest.java *

File Edit Search View Tools Macros Configure Window Help

Find incrementally

Document Sel... TextFieldFrame.java TextFieldTest.java Addition.java LabelFrame.java LabelTest.java * x

```
// LabelTest.java
// Testing LabelFrame.
import javax.swing.JFrame;

public class LabelTest
{
    public static void main( String args[] )
    {
        LabelFrame labelFrame = new LabelFrame(); // create LabelFrame
        labelFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        labelFrame.setSize( 275, 180 ); // set frame size
        labelFrame.setVisible( true ); // display frame
    } // end main
} // end class LabelTest
```

Ex... D... Cl...

Tool Output

Tool completed successfully

Search Results Tool Output

9 | 1 | Read | Ovr | Block | Sync | Rec | Caps

LabelFrame.java

Constant	Description
<i>Horizontal-position constants</i>	
<code>SwingConstants.LEFT</code>	Place text on the left.
<code>SwingConstants.CENTER</code>	Place text in the center.
<code>SwingConstants.RIGHT</code>	Place text on the right.
<i>Vertical-position constants</i>	
<code>SwingConstants.TOP</code>	Place text at the top.
<code>SwingConstants.CENTER</code>	Place text in the center.
<code>SwingConstants.BOTTOM</code>	Place text at the bottom.

Componenti GUI di base

Interagire con una GUI

- Le GUI sono basate sugli **eventi**
 - L'interazione di un utente crea un **evento**
 - Eventi comuni sono il click di un pulsante, scrivere in un campo, selezionare un item da un menu, chiudere una finestra, muovere il mouse
 - Un evento causa la chiamata di un metodo chiamato “event handler” o **gestore dell'evento**

Modello eventi GUI con interfaccia Listener

- “Delegation event model”
 - **Event source** è la component con cui interagisce l'utente
 - **Event object** viene creato e contiene informazioni sull'evento che si è verificato
 - **Event listener** viene notificato quando si verifica l'evento

Campi di testo e gestione degli eventi nei campi di testo

- Class `JTextComponent`
 - Superclass di `JTextField`
 - Superclass di `JPasswordField`
 - Aggiunge un carattere echo per nascondere il testo di input nella component
 - Permette all'utente di inserire del testo nella component quando ha il "focus"
 - Una component ha il focus quando si clicca su di essa.

Passi richiesti per usare un gestore degli eventi per una component GUI

- Per scrivere una applicazione che risponde agli eventi bisogna:
 - Creare una classe per la gestione degli eventi
 - Implementare l'interfaccia `event-listener` adatta per quell'evento
 - Registrare il gestore degli eventi
 - (Si usa spesso definire una **classe interna**)

Classi innestate/Classi interne

- Top-level classes
 - Non dichiarate dentro altre classi
- Nested classes
 - dichiarate dentro altre classi
 - Classi innestate non-`static` si chiamano classi interne (inner classes)
 - Si usano spesso nella gestione degli eventi

- Una classe interna ha accesso diretto a tutte le variabili e metodi della sia classe top-level anche se questi sono dichiarati `private`.
- Prima di creare un oggetto della classe interna deve esistere un oggetto della classe top-level

Usare una classe interna per implementare un gestore di eventi

- `JTextField` e `JPasswordField`
 - Pressando enter all'interno di uno di questi campi produco `ActionEvent`
 - Un `ActionEvent` è poi processato da un oggetto che implementa interfaccia `ActionListener`

Registrare il gestore degli eventi per ogni campo di testo

- Registrare un gestore degli eventi
 - Chiama metodo Call method `addActionListener` per registrare un oggetto `ActionListener`
 - `ActionListener` “ascolta” gli eventi per l’oggetto.

TextFieldFrame.java
TextFieldTest.java