



Programmazione 3 e Laboratorio di Programmazione 3

Interfacce grafiche

Proff. Angelo Ciaramella – Emanuel Di Nardo

GUI

- **Graphical User interface (GUI)**
 - Abstract Window Toolkit (AWT)
 - Swing
 - JavaFX

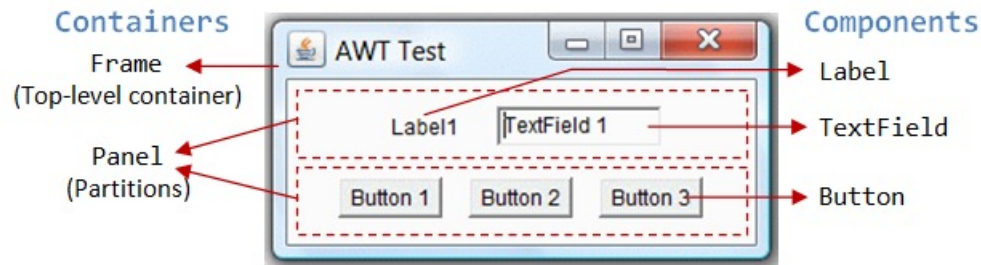


AWT

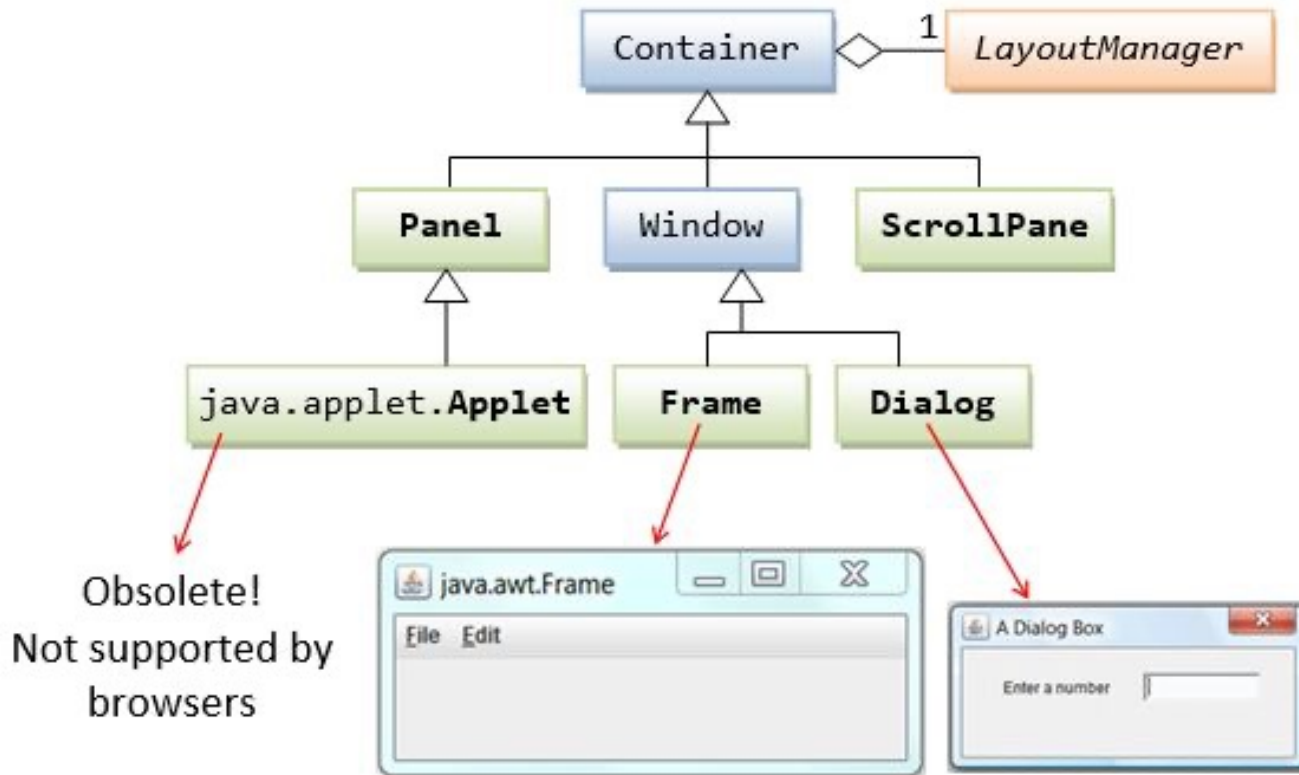
- **AWT** è la prima libreria di Java che permette di creare interfacce grafiche
 - Platform-dependant
 - Event-based
 - Richiede molte risorse



- Esistono due macro gruppi di elementi grafici:
 - Containers:
 - Contengono i componenti (Frame, Panel, ...)
 - Components:
 - Elementi grafici singoli (Button, Label, ...)



AWT - Gerarchia delle classi

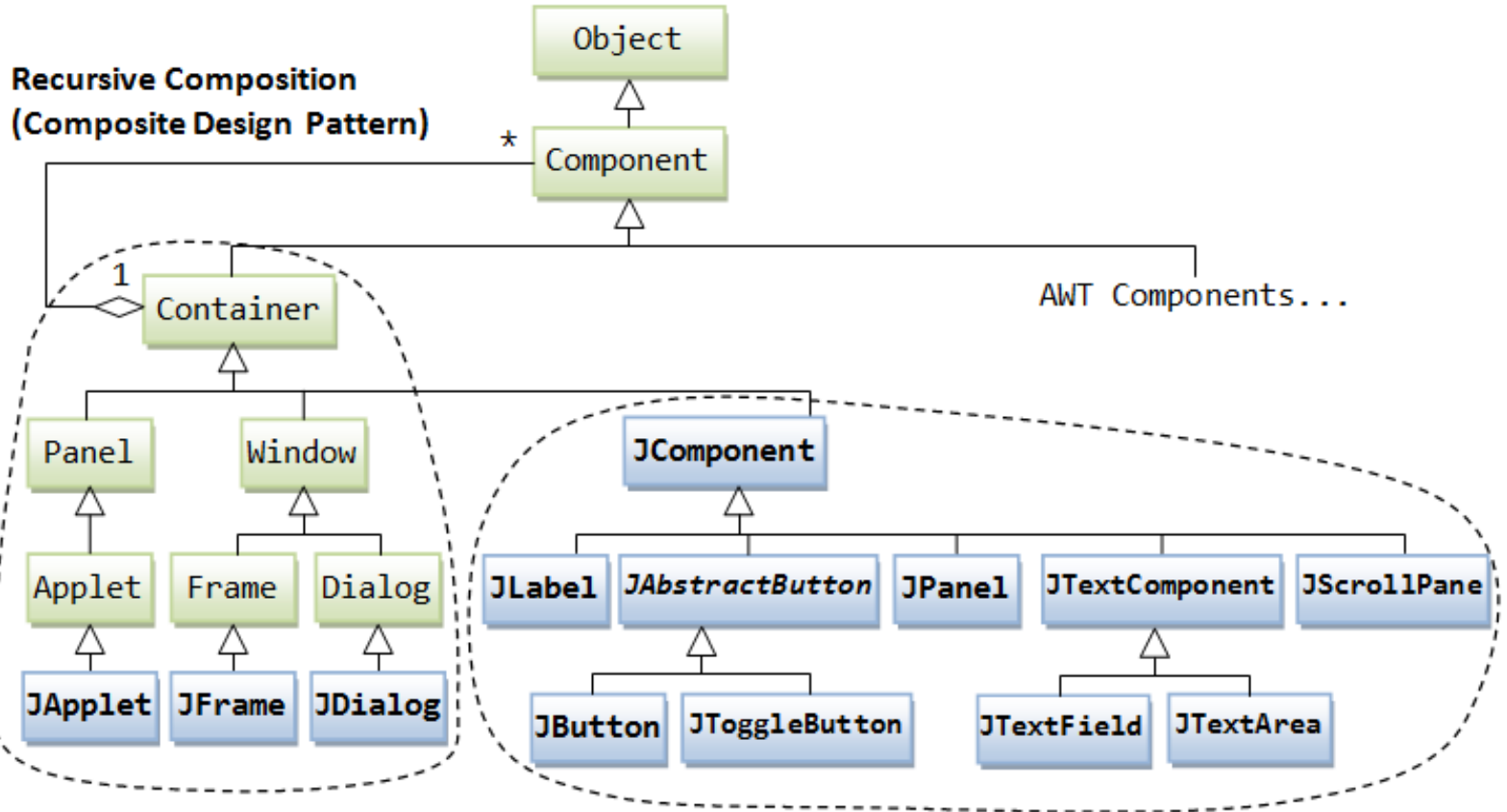


Swing

- **Swing** è il successore di AWT, anche se ne riprende alcuni elementi
 - **Platform-independant**
 - **Event-based** (riprende gli eventi di awt)
 - **Lightweight** (leggero da utilizzare ed elaborare)
 - **Non thread-safe**
- Basta aggiungere agli elementi base di AWT la lettera J...
 - Button -> JButton
 - Panel -> JPanel
 - Frame -> JFrame
 - Label -> JLabel



Swing - Gerarchia delle classi



JFrame

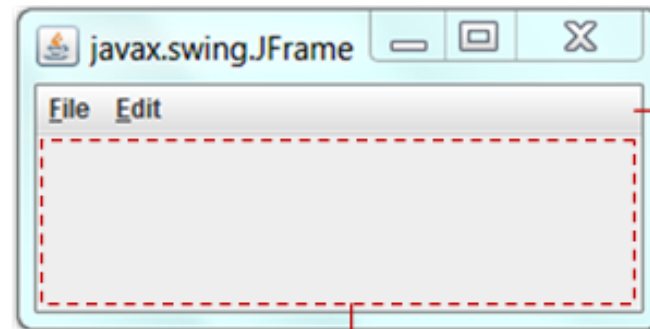
- Tutto avviene nei **JFrame**:
 - Se molto complessi si consiglia di creare una classe specifica per ogni frame.
- Un JFrame è composto da:

- **Menu Bar** (optional)

- **Content Pane**

- Tutti i componenti vengono
NECESSARIAMENTE
aggiunti al
ContentPane

javax.swing.JFrame



Menu Bar
(Optional)

Content Pane

```
Container cp = aJFrame.getContentPane();  
aJFrame.setContentPane(aPanel);
```



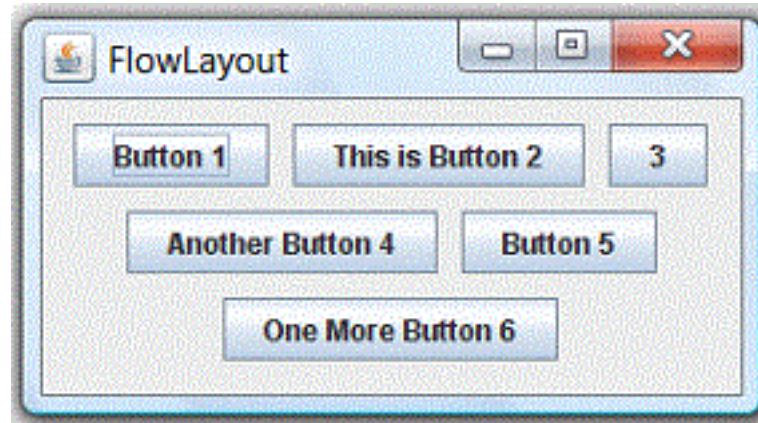
LayoutManager

- L'organizzazione degli elementi è definita da LayoutManager:
 - Permette di sistemare gli elementi grafici secondo un determinato 'ordine'
- Esistono molti layout, alcuni esempi:
 - FlowLayout
 - GridLayout
 - BorderLayout (*default*)
 - ...



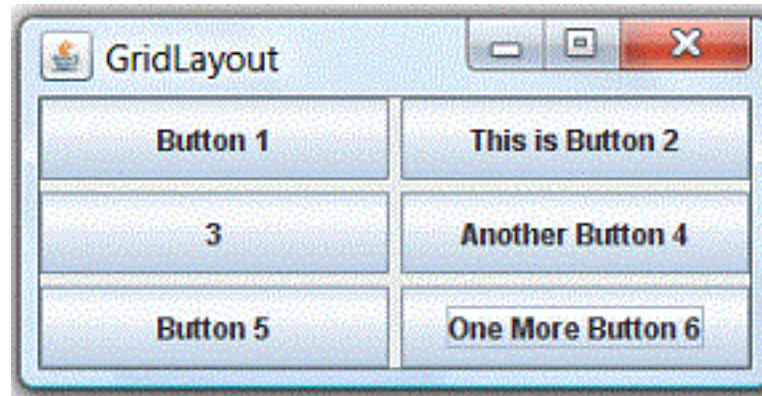
FlowLayout

- Gli elementi sono inseriti da sinistra a destra in base all'ordine in cui vengono aggiunti
 - Al termine dello spazio su una riga, ne viene automaticamente aggiunta una seconda



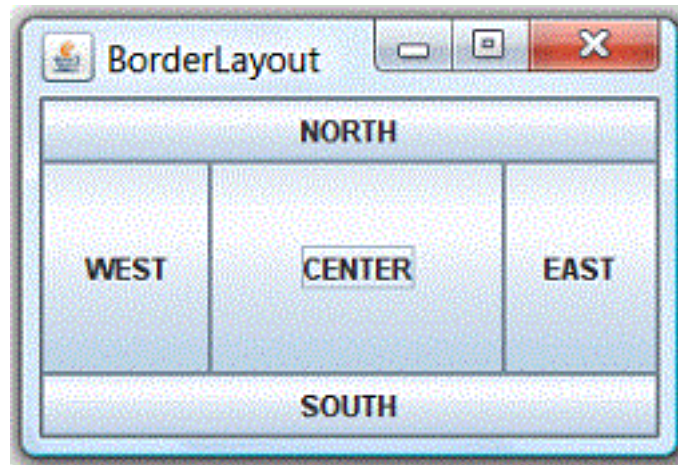
GridLayout

- Gli elementi sono inseriti come in una griglia (matrice):
 - ordinata da sinistra a destra e dall'alto verso il basso
 - al momento della creazione vanno specificate il numero di righe e colonne



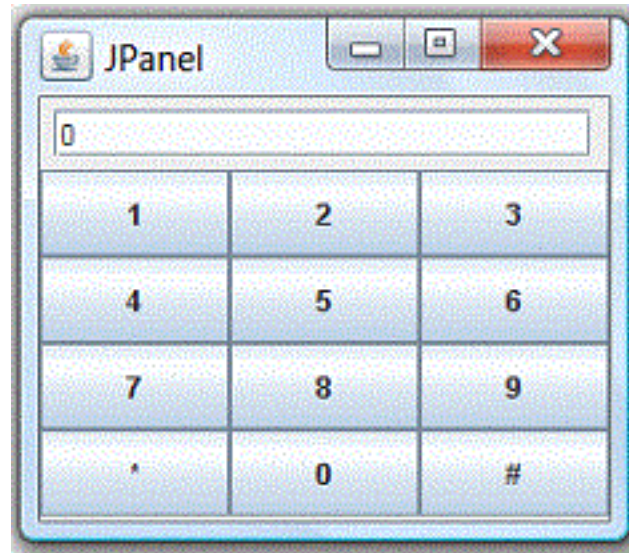
BorderLayout

- Gli elementi sono inseriti tramite coordinate in 5 zone:
 - EAST, WEST, SOUTH, NORTH, and CENTER
 - gli elementi devono essere tutti riempiti



Pannelli

- Per creare schemi più sofisticati è possibile utilizzare **JPanel**
- Ogni JPanel può utilizzare un diverso **LayoutManager**
 - Permette di creare GUI complesse



Pannelli

```
public class MyPanel extends JPanel
{
    @Override
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;

        // codice per il disegno
    }
}
```

Metodo della
superclasse per
cancellare i vecchi
contenuti dei pannelli

Codice di riferimento

RectanglePanelTest.java; RectanglePanel.java



Scelte

Creazione di ogni
singolo pulsante

```
JRadioButton smallButton = new JRadioButton("Small");  
    . . .  
    ButtonGroup sizeGroup = new ButtonGroup();  
    sizeGroup.add(smallButton);
```

Gruppo di pulsanti

```
JPanel panel = new JPanel();  
    panel.setBorder(new TitleBorder(new  
                                EtchedBorder(),"Size"));
```

Titolo di bordo



Scelte

```
JComboBox facenameCombo = new JComboBox();  
facenameCombo.addItem("Serif");
```

Casella combinata

```
JMenuBar menuBar = new JMenuBar();  
setJMenuBar();  
  
JMenu fileMenu = new JMenu("File");  
menuBar.add(fileMenu);
```

Barra di menù

Codice di riferimento

```
ChoiceTest.java; ChoiceFrame.java
```

Codice di riferimento

```
MenuTest.java; MenuFrame.java; RectanglePanel.java
```



Slider

```
public JSlider(int min, int max, int value)
```

Creiamo un cursore orizzontale

Codice di riferimento

```
SliderTest.java; SliderFrame.java
```

ActionListener

- Permette di rispondere agli eventi generati dall'interazione tra l'utente e la GUI.
- E' un'interfaccia (va implementata)
 - Il metodo da implementare è:

```
public void actionPerformed(ActionEvent event) ;
```

- Utilizzato da **qualunque** componente che preveda l'interazione con l'utente



Elaborare testo in ingresso

```
JLabel xLabel = new JLabel("x =");
```

Costruzione campo testo

```
moveButton = new JButton("Move", new ImageIcon("fig.gif"));
```

Definizione di pulsante

```
class CalculateListener implements ActionListener{
    public void actionPerformed(ActionEvent event){
        double number = Double.parseDouble(numField.getText());
        textArea.append("x : " + number + " - - ->" + " log(x): " +
            Math.log(number) + "\n");
    }
}
```

Gestione evento

```
JTextArea textArea = new JTextArea(10,30);
```

Visualizzazione linee di testo

Codice di riferimento

```
TextAreaTest.java
```



Esercizio - base

- Creiamo una calcolatrice come nell'immagine



Esercizio - avanzato

- Forniamo un'interfaccia grafica per **BankAccount** in modo da creare un ATM (Bancomat) che permetta di:
 - Visualizzare il saldo
 - Ritirare soldi
 - Depositare soldi



IDE Designer

- La maggior parte degli IDE fornisce uno strumento per creare interfacce grafiche in Swing con un semplice drag and drop:
 - IntelliJ – GUI Designer
(<https://www.jetbrains.com/help/idea/components-of-the-gui-designer.html>)
 - Eclipse – WindowsBuilder
(<https://www.eclipse.org/windowbuilder/>)
 - NetBeans – GUI Builder
(<https://netbeans.apache.org/kb/docs/java/quickstart-gui.html>)

