

Java terzo contatto

Swing Java





Introduzione

- Componenti grafici di Java (Swing)
 - Utilizzo di alcuni dei metodi per creare semplici finestre grafiche
 - Accesso ai JavaDoc per usare meglio gli oggetti già pronti
- Richiami
 - Variabili locali, di istanza e parametri
 - Costruttori
 - Overloading dei metodi e dei costruttori
 - metodo main

Java Swing

JLabel

Etichetta

JButton

Pulsante di prova

JComboBox

pippo
pippo
pluto
paperino

JTextField

testo di prova

JCheckBox

Spunta la selezione

JFrame

ContentPane



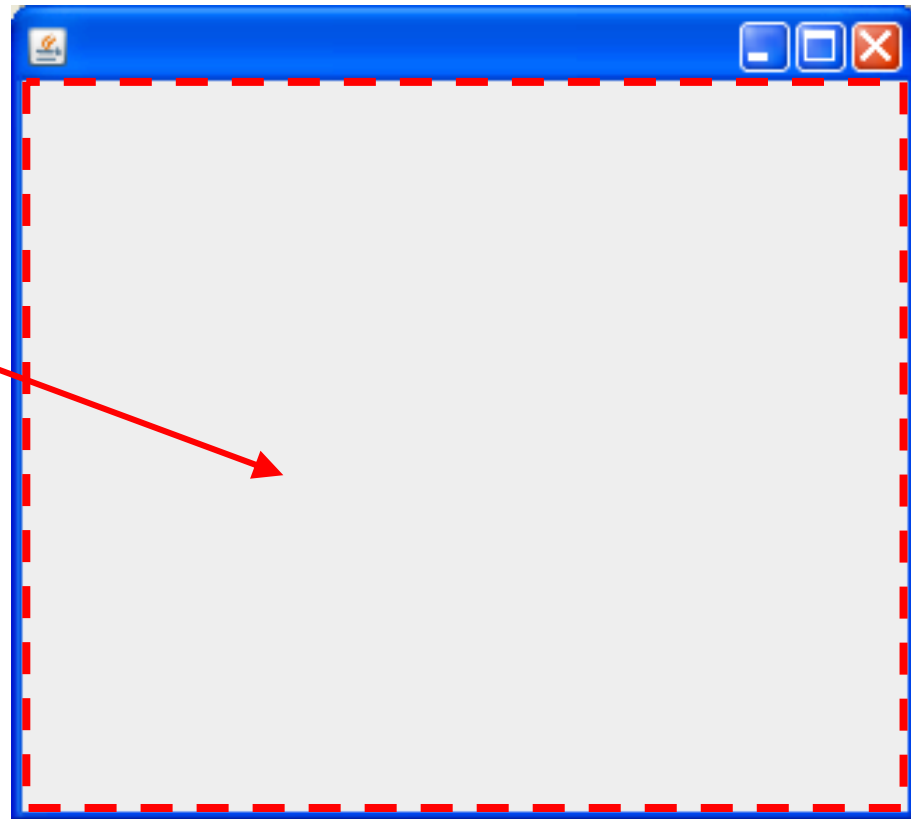
I Lego

- JFrame → getContentPane()
 - Al **Content Pane** è un Panel a cui si possono aggiungere (con il metodo add)
 - JPanel
 - JButton, JTextField, JCheckBox, JLabel, JComboBox
 - ...altri componenti Swing
- JPanel
 - Metodo add per aggiungere componenti
 - JPanel
 - JButton, JTextField, JCheckBox, JLabel, JCombo
 - ...altri componenti Swing

JFrame - `javax.swing.JFrame`

- **JFrame**: una finestra dotata di cornice, barra del titolo con pulsanti tipici e un pannello (content pane)

Content Pane
È il pannello contenuto
nel JFrame





JFrame

- Come si aggiungono gli elementi e si imposta il colore?
 - Si applicano al Content Pane!
- Content Pane
 - È un Container, in sostanza è un pannello semplice (contiene componenti)
 - In genere si crea un proprio JPanel (o JContainer) e lo si imposta come Content Pane
 - I metodi che si possono applicare sono quali tutti quelli del JPanel




JFrame

Costruttori

- `public JFrame()`
 - Crea un nuovo Frame inizialmente invisibile
- `public JFrame(String title)`
 - Crea un nuovo frame, inizialmente invisibile, con un testo specificato.


Metodi

- `public void setVisible(boolean b)`
 - Mostra o nasconde un Frame in base al valore del parametro b



JFrame - metodi

- public Container **getContentPane()**
 - Ritorna il `contentPane` per il `Frame`.
- public void **setContentPane**(Container `contentPane`)
 - Imposta il `contentPane` del `Frame`.
- public void **setDefaultCloseOperation**(int `operation`)
 - Imposta l'operazione che viene eseguita quando si chiude la finestra.
 - `JFrame.DO_NOTHING_ON_CLOSE`
 - `JFrame.HIDE_ON_CLOSE`
 - `JFrame.DISPOSE_ON_CLOSE`
 - `JFrame.EXIT_ON_CLOSE`.



JFrame - metodi

- public void **setSize**(int width, int height)
 - Imposta la dimensione della finestra
- public void **setLocation**(int x, int y)
 - Sposta la finestra nella posizione x,y specificata (relativa all'angolo in alto a sinistra della finestra)



Color - `java.awt.Color`

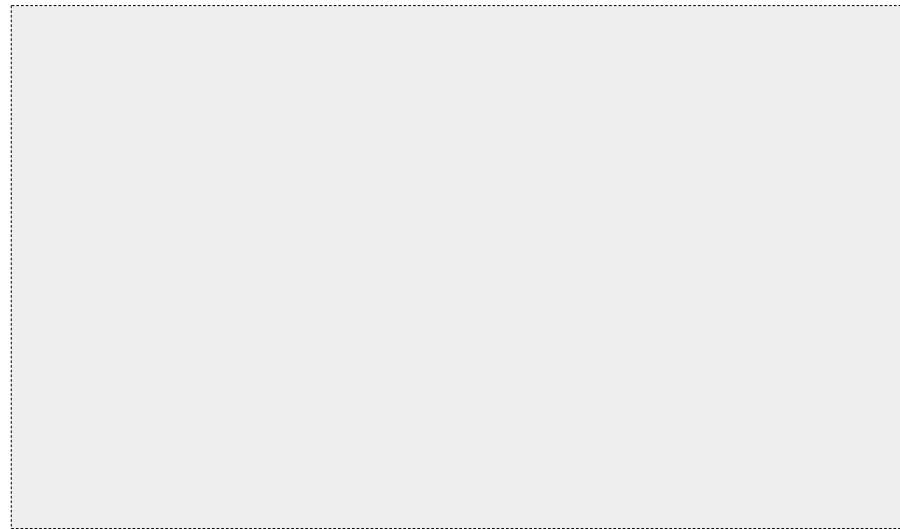
Costruttori

- **public Color**(int r, int g, int b)
 - Crea un colore RGB opaco nel range (0- 255).
- **public Color**(int r, int g, int b, int a)
 - Crea un colore RGB nel range (0- 255) con l'aggiunta della trasparenza (canale Alpha).
- Alcuni colori già pronti, come campi statici di Color
 - Color.BLACK ,Color.BLUE ,Color.CYAN ,
Color.DARK_GRAY, Color. GRAY,
Color.GREEN, Color.LIGHT_GRAY, Color.MAGENTA ,
Color.ORANGE, Color.PINK, Color.RED, Color.WHITE,
Color.YELLOW



JPanel - `javax.swing.JPanel`

- La classe **JPanel** implementa un contenitore senza cornice che si usa per contenere altri componenti






JPanel

Costruttore

- `public JPanel()`
 - Crea un panel vuoto con flow layout.

Metodi

- `public void setBackground(Color bg)`
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- `public void setOpaque(boolean b)`
 - Se true lo sfondo del componente (background) viene disegnato



JPanel - metodi

- `public Component add(Component comp)`
 - Aggiunge un componente (Component di Swing) al pannello e viene posizionato in base al Layout Manager
 - `public void add(Component comp, Object constraints)`
 - Come il metodo precedente, ma permette di specificare alcuni vincoli di alcuni LayoutManager - **overloading**
 - `public void setPreferredSize(Dimension preferredSize)`
 - Imposta la dimensione preferita di un componente
- Ad un pannello posso aggiungere anche altri Pannelli
- Serve per organizzare meglio lo spazio
- Il metodo `setSize(...)` non funziona in quanto la size è gestita dal LayoutManager (vedremo in seguito)



Esempio

```
public static void main(String[] s){
    JFrame f=new JFrame();//nuova finestra
    JPanel p=new JPanel();//nuovo pannello
    f.setContentPane(p);//imposta il content pane
    p.setBackground(Color.GREEN);/*colore dello
    sfondo (content pane)*/
    f.setSize(500,500);/*imposta la dimensione
    della finestra */
    f.setVisible(true);/*visualizza la
    finestra*/
}
```



JLabel - javax.swing.JLabel

Bottoni: la classe **JLabel** implementa un etichetta con sfondo non opaco

Etichetta

Costruttori

■ JLabel()

- Crea una label vuota.

■ JLabel(String text)

- Crea una lavel con il testo specificato

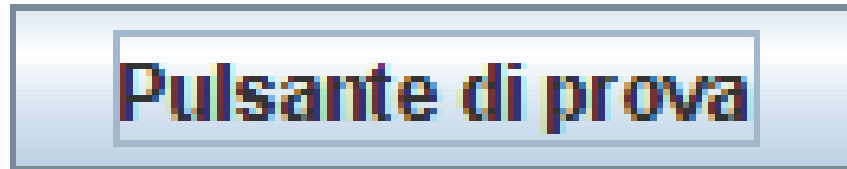


JLabel – metodi comuni

- public void **setPreferredSize**(Dimension preferredSize)
 - Imposta la dimensione preferita di un componente
- public void **setForeground**(Color fg)
 - Imposta il colore del testo.
- public void **setBackground**(Color bg)
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- public void **setText**(String text)
 - Imposta il testo visualizzato
- public void **setOpaque**(boolean isOpaque)
 - Se true lo sfondo del componente (background) viene disegnato

JButton - javax.swing.JButton

Bottoni: la classe **JButton** implementa un pulsante



Costruttori

- **public JButton()**
 - Crea un pulsante senza testo
- **public JButton(String text)**
 - Crea un pulsante con il testo specificato



JButton – metodi comuni

- public void **setPreferredSize**(Dimension preferredSize)
 - Imposta la dimensione preferita di un componente
- public void **setForeground**(Color fg)
 - Imposta il colore del testo.
- public void **setBackground**(Color bg)
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- public void **setText**(String text)
 - Imposta il testo visualizzato
- public void **setOpaque**(boolean isOpaque)
 - Se true lo sfondo del componente (background) viene disegnato



Esempio

```
public static void main(String[] s){
    /*nuova finestra*/
    JFrame f=new JFrame();
    /*nuovo pannello */
    JPanel p=new JPanel();
    /*imposta il content pane*/
    f.setContentPane(p);
    /*crea un pulsante*/
    JButton b=new JButton("pulsante di prova");
    /*aggiunge il pulsante al pannello */
    p.add(b);
    /*imposta la dimensione della finestra */
    f.setSize(500,500);
    /*visualizza la finestra*/
    f.setVisible(true);
}
```



JTextField - `javax.swing.JTextField`

- Casella di Testo: la classe **JTextField** implementa un'area in cui è possibile inserire del testo



testo di prova

Costruttori

- `public JTextField()`
 - Costruisce una Test Field vuota
- `public JTextField(String text)`
 - Costruisce una Test Field con il testo specificato

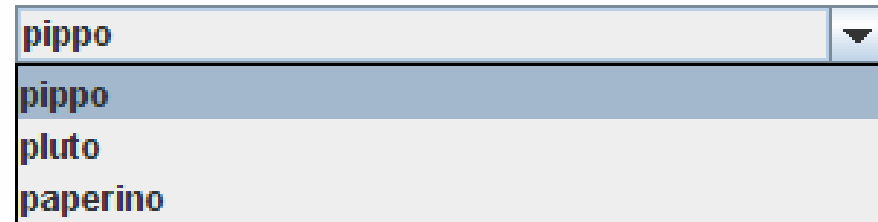


JTextField – metodi comuni

- public void **setPreferredSize**(Dimension preferredSize)
 - Imposta la dimensione preferita di un componente
- public void **setForeground**(Color fg)
 - Imposta il colore del testo.
- public void **setBackground**(Color bg)
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- public void **setText**(String text)
 - Imposta il testo visualizzato
- public void **setOpaque**(boolean isOpaque)
 - Se true lo sfondo del componente (background) viene disegnato

JComboBox - `javax.swing.JComboBox`

■ Casella selezione



Costruttori

■ `JComboBox(Object[] items)`

- Crea una JComboBox che contiene gli elementi specificati nell'array.



JComboBox – metodi comuni

- public void **setPreferredSize**(Dimension preferredSize)
 - Imposta la dimensione preferita di un componente
- public void **setForeground**(Color fg)
 - Imposta il colore del testo.
- public void **setBackground**(Color bg)
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- public void **setOpaque**(boolean isOpaque)
 - Se true lo sfondo del componente (background) viene disegnato



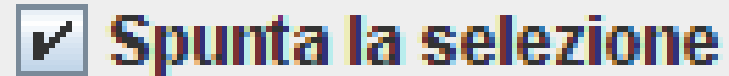
JComboBox

Metodi

- public int **getSelectedIndex()**
 - Ritorna l'indice dell'elemento selezionato o `-1` se non è selezionato nulla
- public void **setSelectedIndex(int anIndex)**
 - Imposta l'indice selezionato
- public Object **getSelectedItem()**
 - Ritorna l'elemento selezionato
- public void **setSelectedItem(Object anObject)**
 - Imposta l'elemento specificato come selezionato

JCheckBox - `javax.swing.JCheckBox`

■ Casella selezione



Costruttori

■ `public JCheckBox()`

- Crea un check box non selezionato

■ `public JCheckBox(String text)`

- Crea un check box non selezionato con il testo specificato



JCheckBox – metodi comuni

- public void **setPreferredSize**(Dimension preferredSize)
 - Imposta la dimensione preferita di un componente
- public void **setForeground**(Color fg)
 - Imposta il colore del testo.
- public void **setBackground**(Color bg)
 - Imposta il colore di sfondo. Il colore è usato solo se il componente è opaco.
- public void **setOpaque**(boolean isOpaque)
 - Se true lo sfondo del componente (background) viene disegnato



JCheckBox

Metodi

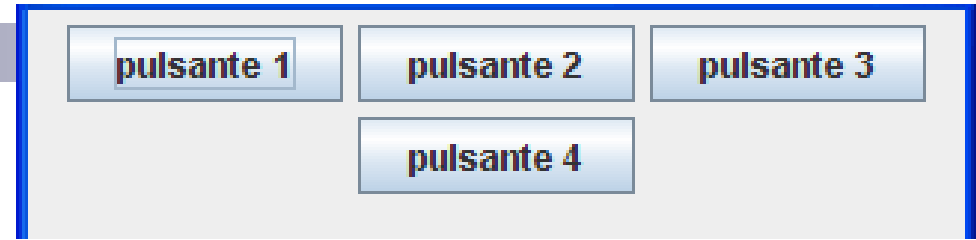
- public boolean **isSelected()**
 - Indica se selezionato
- public void **setSelected(boolean b)**
 - Imposta il check su selezionato



LayoutManager

- Permette di definire il modo in cui i componenti vengono disposti all'interno di un pannello
 - FlowLayout
 - GridLayout
 - BorderLayout

FlowLayout



- Dispone i componenti uno di seguito all'altro (eventualmente andando alla riga successiva)
- FlowLayout()
 - Costruisce un manager con allineamento al centro
- FlowLayout(int align)
 - Costruisce un manager con allineamento specificato dal parametro
- FlowLayout(int align, int hgap, int vgap)
 - Costruisce un manager con allineamento specificato dal parametro e spaziatura orizzontale e verticale specificata

GridBag



- Dispone i componenti a scacchiera (Matrice)
- **public GridLayout()**
 - Crea una griglia su singola colonna
- **public GridLayout(int rows, int cols)**
 - Crea una griglia con numero di righe e colonne specificate
- **public GridLayout(int rows, int cols, int hgap, int vgap)**
 - Crea una griglia con numero di righe e colonne specificate, con una spaziatura (tra le celle) orizzontale e verticale specificata

BorderLayout



- Dispone i componenti in 5 aree (Nord, Sud, Est, Ovest e Centro)
- **public BorderLayout()**
 - Costruisce un border layout spaziatura
- **public BorderLayout(int hgap, int vgap)**
 - Costruisce un border layout spaziatura orizzontale e verticale specificata



BorderLayout

- Per aggiungere elementi occorre specificare in quale area collocarli
- Supponendo che xxx sia una variabile associata ad un Component (es JButton etc)
 - `p.add(xxx, BorderLayout.EAST);`
 - `p.add(xxx, BorderLayout.WEST);`
 - `p.add(xxx), BorderLayout.NORTH);`
 - `p.add(xxx, BorderLayout.SOUTH);`
 - `p.add(xxx, BorderLayout.CENTER);`