UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI **MATEMATICA E INFORMATICA**

Interfacce Grafiche e Programmazione ad Eventi

Carmine Dodaro

Anno Accademico 2019/2020

Controlli grafici di interfaccia

Cosa sono

Sono quegli elementi che possono essere aggiunti ai container per mostrare delle informazioni agli utenti. Ad esempio, pulsanti, aree di testo, etichette, ecc.

Il controllo JLabel

Cos'è

Permette di visualizzare testi e immagini all'interno dei container.

```
JFrame f = new JFrame("Titolo");
JLabel testo = new JLabel("Questo è un testo");
ImageIcon logo = new ImageIcon("logo.jpg");
JLabel immagine = new JLabel(logo);
f.add(testo,BorderLayout.EAST);
f.add(immagine, BorderLayout.CENTER);
f.setSize(800, 800);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

JLabel

Per scalare il logo?

Scalare un'immagine

```
JFrame f = new JFrame("Titolo");
JLabel testo = new JLabel("Questo è un testo");
ImageIcon logo = new ImageIcon("logo.jpg");
Image im = logo.getImage();
Image logoScalato = im.getScaledInstance(400, 120, Image
    .SCALE_SMOOTH);
logo = new ImageIcon(logoScalato);
JLabel immagine = new JLabel(logo);
f.add(testo,BorderLayout.EAST);
f.add(immagine, BorderLayout.CENTER);
f.setSize(800, 800);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

JLabel


Titolo
UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
Questo è un testo

Il controllo JButton

Cos'è

Il controllo JButton permette di visualizzare un pulsante che può essere visualizzato come un testo oppure come un'immagine.

```
JButton button = new JButton("Clicca qui");
```

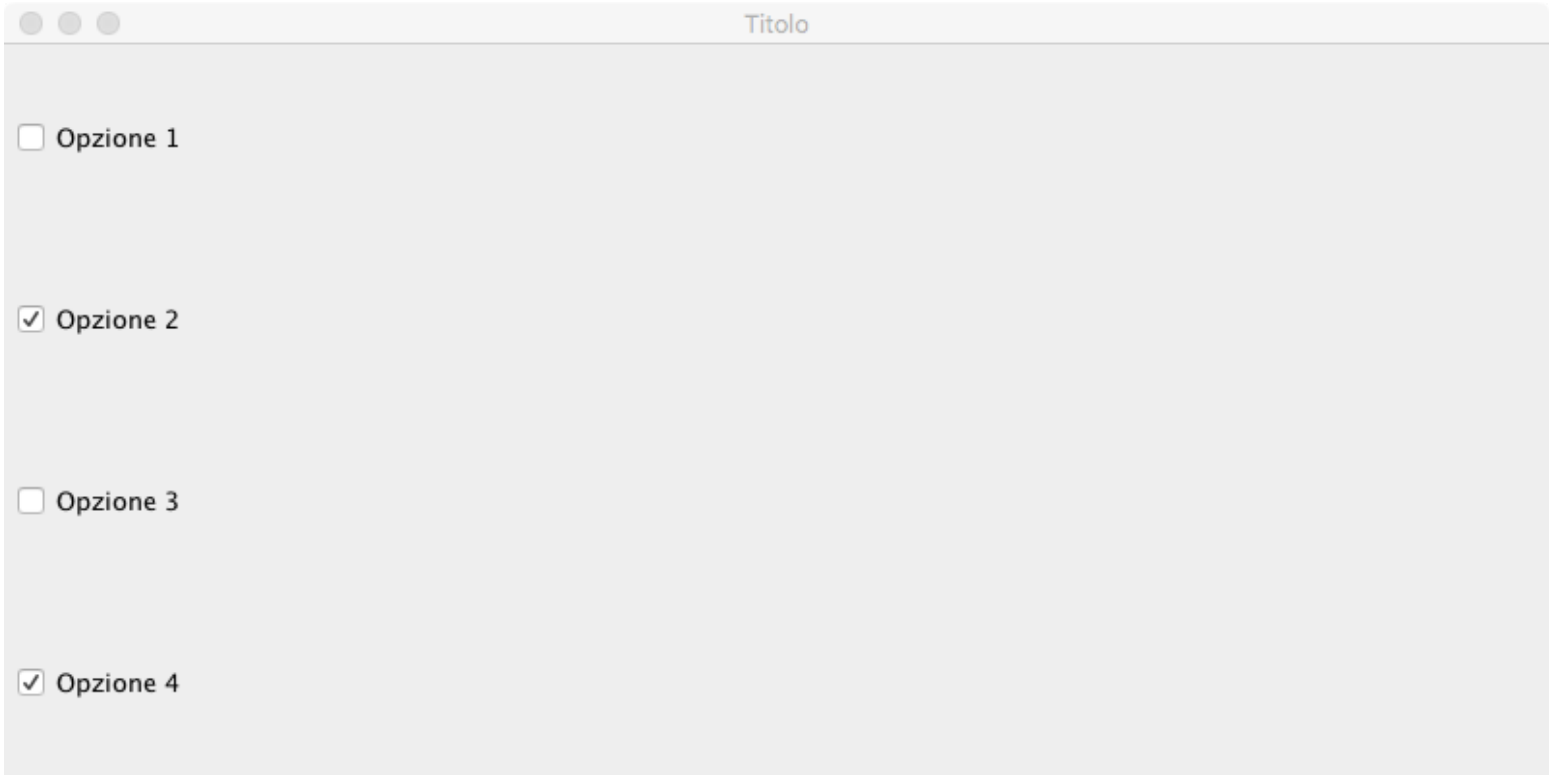
Il controllo JCheckBox

Cos'è

Il controllo JCheckBox permette di visualizzare delle caselle di controllo con delle spunte che permettono di attivare/disattivare una scelta.

```
JFrame f = new JFrame("Titolo");
f.setLayout(new GridLayout(4, 1));
JCheckBox opzione1 = new JCheckBox("Opzione 1");
JCheckBox opzione2 = new JCheckBox("Opzione 2",true);
JCheckBox opzione3 = new JCheckBox("Opzione 3");
JCheckBox opzione4 = new JCheckBox("Opzione 4",true);
f.add(opzione1);
f.add(opzione2);
f.add(opzione3);
f.add(opzione4);
f.setSize(800, 400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

JCheckBox



Il controllo JRadioButton

Cos'è

Il controllo JRadioButton permette di visualizzare dei pulsanti di opzione in cui si può attivare una scelta per volta.

```
JFrame f = new JFrame("Titolo");
f.setLayout(new GridLayout(4, 1));
ButtonGroup b = new ButtonGroup();
JRadioButton opzione1 = new JRadioButton("Opzione 1");
JRadioButton opzione2 = new JRadioButton("Opzione 2");
b.add(opzione1);
b.add(opzione2);
f.add(opzione1);
f.add(opzione2);
f.setSize(800, 400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Nota: ButtonGroup si può utilizzare anche con JCheckBox.

I controlli JTextField, JPasswordField e JTextArea

Cos'è

Il controllo JTextField permette di inserire un campo di testo su una singola riga.
Il controllo JPasswordField è simile, ma ogni carattere inserito è sostituito da un altro carattere in modo da nasconderne il contenuto.
Il controllo JTextArea permette di visualizzare un'area rettangolare formato da più righe dove inserire del testo.

```
JTextField t = new JTextField();
JPasswordField p = new JPasswordField();
JTextArea a = new JTextArea();
```

Il controllo JComboBox

Cos'è

Il controllo JComboBox permette di visualizzare un menu a tendina dove selezionare alcune opzioni tra quelle disponibili.

```
JFrame f = new JFrame("Titolo");
String items[] = {"Scelta1", "Scelta2", "Scelta3"};
JComboBox<String> jComboBox = new JComboBox<String>(
    items);
f.add(jComboBox);
f.setSize(400,400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Il controllo JList

Cos'è

Il controllo JList permette di visualizzare un rettangolo in cui sono presenti una lista di elementi selezionabili.

```
JFrame f = new JFrame("Titolo");
String items[] = {"Scelta1", "Scelta2", "Scelta3"};
JList<String> lista = new JList<String>(items);
f.add(lista);
f.setSize(400,400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Il controllo JColorChooser

Cos'è

Il controllo JColorChooser permette di scegliere un colore.

```
JFrame f = new JFrame("Titolo");
f.setSize(400,400);
Color res = JColorChooser.showDialog(f, "Scegli un
    colore", Color.RED);
JPanel p = new JPanel();
p.setBackground(res);
f.add(p);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Il controllo JFileChooser

Cos'è

Il controllo JFileChooser permette di scegliere un file o una cartella all'interno del computer. Può essere di tipo Open per aprire un file o Save per salvare un file.

```
JFrame f = new JFrame("Titolo");
f.setSize(400,400);
JFileChooser fc = new JFileChooser();
int res = fc.showOpenDialog(f);
if(res == JFileChooser.APPROVE_OPTION) {
  File fileScelto = fc.getSelectedFile();
  System.out.println(fileScelto.getAbsolutePath());
}
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Il controllo JTable

Cos'è

Il controllo JTable permette di visualizzare una tabella.

```
JFrame f = new JFrame("Titolo");
f.setSize(400,400);
String[] header = {"Nome", "Cognome", "Indirizzo"};
String[][] dati = {
  {"Mario","Rossi", "Via Roma"},
  {"Francesco", "Verdi", "Via Napoli"},
  {"Simona", "Bianchi", "Via Torino"}
};
JTable tabella = new JTable(dati,header);
tabella.setAutoCreateRowSorter(true);
JScrollPane scrollPane = new JScrollPane(tabella);
f.add(scrollPane);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Al posto degli array si possono anche usare Vector.

Il controllo JProgressBar

Cos'è

Il controllo JProgressBar permette di visualizzare una barra orizzontale o verticale che indica il progresso nel compiere una determinata azione.

```
JFrame f = new JFrame("Titolo");
f.setSize(400,400);
JProgressBar progress = new JProgressBar();
f.add(progress);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
int percentualeCaricamento = 0;
for(int i = 0; i < 100000 * 10; i++) {
  if(i % 100000 == 0) {
    percentualeCaricamento += 10;
    progress.setValue(percentualeCaricamento);
  }
  System.out.println(i);
}
```

Il controllo JSeparator

Cos'è

Il controllo JSeparator permette di visualizzare una linea che funziona da separatore di elementi orizzontale o verticale.

```
JSeparator separator = new JSeparator();
```

Menu

Aggiungere la barra del menu

Un menu è il componente grafico che permette di inserire una serie di comandi nella barra del menu.

```
JFrame f = new JFrame("Titolo");
f.setSize(400,400);
JMenuBar menubar = new JMenuBar();
JMenu file = new JMenu("File");
JMenuItem nuovo = new JMenuItem("Nuovo");
JMenuItem apri = new JMenuItem("Apri");
file.add(nuovo);
file.add(apri);
menubar.add(file);
JMenu modifica = new JMenu("Modifica");
menubar.add(modifica);
JMenu aiuto = new JMenu("Aiuto");
menubar.add(aiuto);
f.setJMenuBar(menubar);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

Menu contestuali

Aggiungere un menu contestuale

Un menu contestuale è il componente grafico che si attiva su un componente al verificarsi di un qualche evento (ad esempio il click con il tasto destro).

```
JFrame f = new JFrame("Programma");
f.setSize(800,1200);
JPopupMenu popupMenu = new JPopupMenu();
JMenuItem nuovo = new JMenuItem("nuovo");
JMenuItem salva = new JMenuItem("salva");
JMenuItem opzione = new JMenuItem("opzione");
JMenuItem opzione2 = new JMenuItem("altra opzione");
popupMenu.add(nuovo);
popupMenu.add(salva);
popupMenu.addSeparator();
popupMenu.add(opzione);
popupMenu.add(opzione2);
JPanel panel = new JPanel();
panel.setComponentPopupMenu(popupMenu);
panel.setBackground(Color.RED);
f.add(panel);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
```

JPopupMenu