

GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 3:

Programació Orientada a Events (4)

Laura Igual

Departament de Matemàtiques i Informàtica

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona



UNIVERSITAT DE
BARCELONA

LAYOUT MANAGER

Layout Manager

- Controla les components que estan contingudes en una altra component associada al Layout Manager

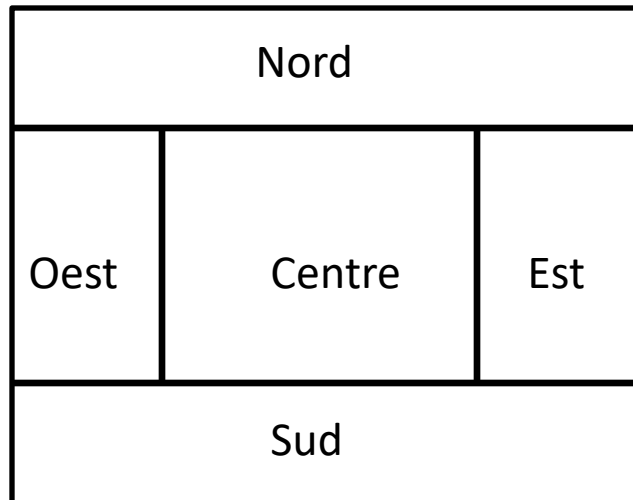
Exemple:

Si un frame conté un panell que conté un botó, el layout manager del panell controlarà el tamany i posició del botó, mentre que el layout manager del frame controlarà el tamany i posició del panell. El botó no necessitarà un layout manager.

BorderLayout

- Divideix el fons del component en 5 regions.
- Es pot afegir només una component per regió.
- Normalment, no s'aconsegueixen els tamanys preferits pels components.
- És el layout manager per defecte per a un **frame**

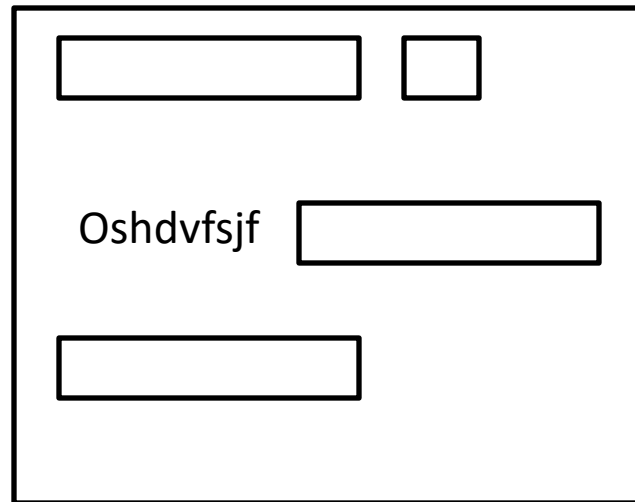
Exemple: BorderLayout



FlowLayout

- Actua com un processador de text, amb components en lloc de paraules
- Cada component és del tamany predefinit i es van afegint d'esquerra a dreta en l'ordre que són afegits
- És el layout manager per defecte per a un **panel**

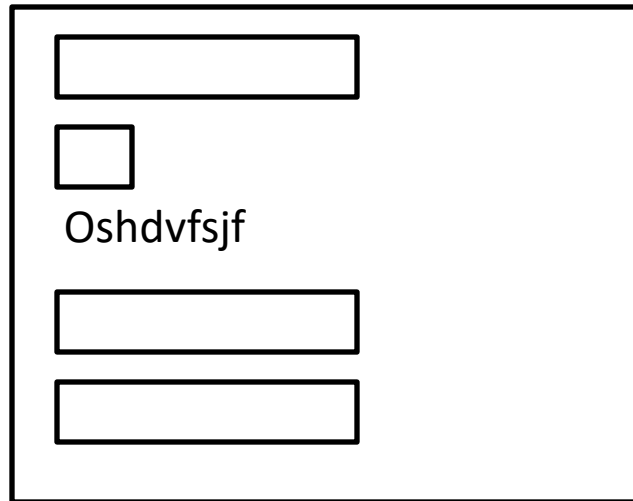
Example: FlowLayout



BoxLayout

- És com FlowLayout en que cada component és del tamany predefinit i en que es van col·locant en l'ordre que són afegits.
- Però, BoxLayout pot apilar els components verticalment (o horitzontalment).
- En lloc de tindre un wrapping automàtic , es pot forçar a començar una nova línia.

Example: BoxLayout



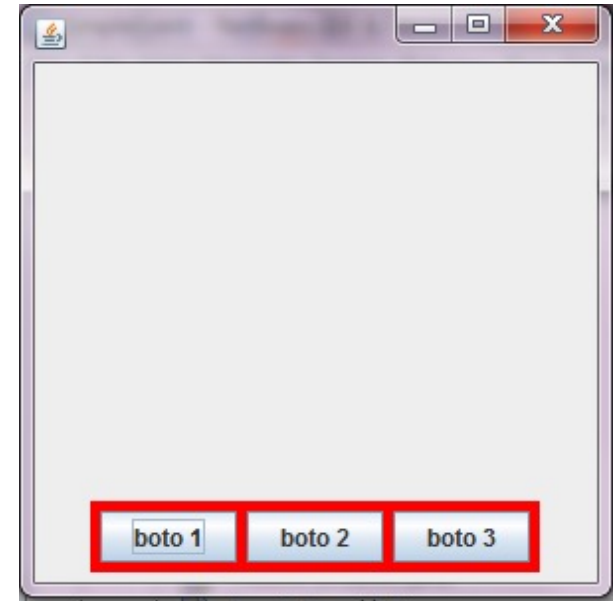
Canvi de Layout

- Per fer un canvi de Layout manager:

```
rootPane.getContentPane().setLayout(new BorderLayout());
```

Exemple

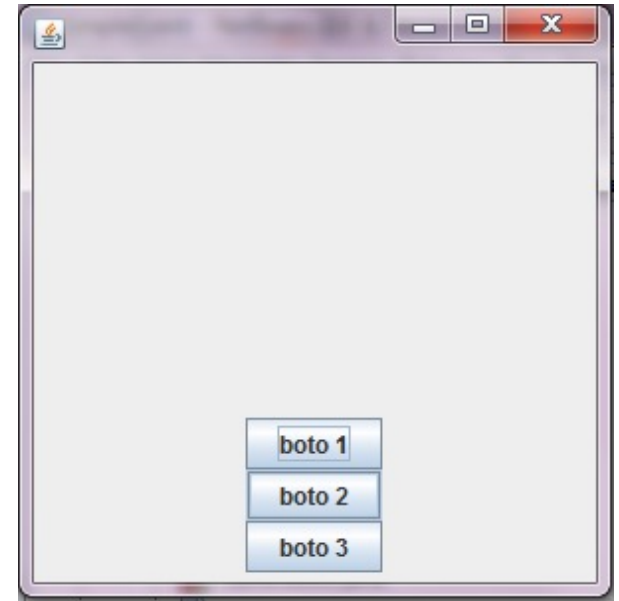
```
public class ProvaLayout {  
    public static void main (String[] args) {  
        ProvaLayout gui = new ProvaLayout();  
        gui.go();  
    }  
    public void go(){  
        JFrame frame = new JFrame();  
  
        JPanel panelA = new JPanel();  
        JPanel panelB = new JPanel();  
  
        panelB.setBackground(Color.red);  
  
        panelB.add(new JButton("boto 1"));  
        panelB.add(new JButton("boto 2"));  
        panelB.add(new JButton("boto 3"));  
        panelA.add(panelB);  
  
        frame.getContentPane().add(BorderLayout.SOUTH,panelA);  
  
        frame.setSize(300,300);  
        frame.setVisible(true);  
    }  
}
```



Posició seleccionada a la distribució del frame.

Example

```
public class ProvaLayout {  
    public static void main (String[] args) {  
        ProvaLayout gui = new ProvaLayout();  
        gui.go();  
    }  
    public void go(){  
        JFrame frame = new JFrame();  
  
        JPanel panelA = new JPanel();  
        JPanel panelB = new JPanel();  
  
        panelB.setBackground(Color.red);  
        panelB.setLayout(new BoxLayout(panelB, BoxLayout.Y_AXIS));  
        panelB.add(new JButton("boto 1"));  
        panelB.add(new JButton("boto 2"));  
        panelB.add(new JButton("boto 3"));  
        panelA.add(panelB);  
  
        frame.getContentPane().add(BorderLayout.SOUTH,panelA);  
  
        frame.setSize(300,300);  
        frame.setVisible(true);  
    }  
}
```



Altres layouts

- **GridLayout** simplement fa un grapat de components iguals en grandària i els mostra en el nombre sol·licitat de files i columnes.
- **GroupLayout** treballa amb els dissenys horitzontals i verticals per separat. La disposició es defineix per a cada dimensió de forma independent. En conseqüència, però, cada component necessita ser definida dues vegades en el disseny.
- **SpringLayout** és un controlador de distribució flexible. Se li permet especificar les relacions precises entre les components sota el seu control. Per exemple, és possible definir que la vora esquerra d'un component és a una certa distància (que es pot calcular dinàmicament) des de la vora dreta d'un segon component.

Veure més en:

- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>
- <https://docs.oracle.com/javase/tutorial/uiswing/layout/using.html>

LOOK AND FEEL

Look and feel

- Swing està dissenyat per a que es pugui canviar el *look and feel* d'una aplicació GUI
 - **look** fa referència a l'**aparença** de les components de la GUI
 - **feel** fa referència al **funcionament** de les components de la GUI
- Els programes Java poden adoptar l'aparença de la plataforma sobre la que s'executen, o aparences específiques.

Look and feel

- *CrossPlatformLookAndFeel* or *Metal*
Standard Java look and feel. Forma part de l'API (javax.swing.plaf.metal) i s'utilitza per defecte.
- *SystemLookAndFeel*: look and feel natiu per la plataforma.
Dins del Java SDK.

Look and feel

- Java proporciona:

- Metal

- Nimbus

- Motif

- Windows

- Windows classic



Van a totes les
plataformes

Look and feel

```
import napkin.NapkinLookAndFeel;
```

```
public class ExempleLookAndFeel{
```

```
    public static void main(String [] args){
```

```
        ExempleLookAndFeel ex = new ExempleLookAndFeel();
```

```
        ex.goLookAndFeelNapkin();
```

```
    }
```

```
    public void goLookAndFeelNapkin(){
```

```
        try {
```

```
            UIManager.setLookAndFeel(new NapkinLookAndFeel(););
```

```
        } catch (Exception unused) {
```

```
            // Ignorar l'excepció, ja que no podem fer res. Utilitzarem el default.
```

```
        }
```

```
        JFrame frame = new JFrame();
```

```
        JButton button = new JButton("Boto");
```

```
        frame.getContentPane().add(BorderLayout.SOUTH, button);
```

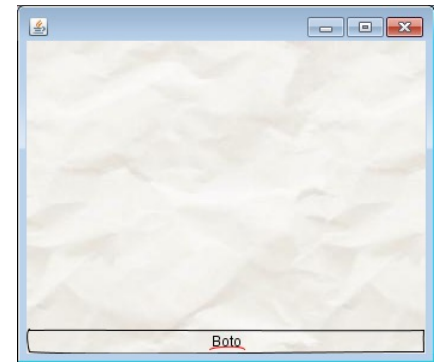
```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setSize(350,300);
```

```
        frame.setVisible(true);
```

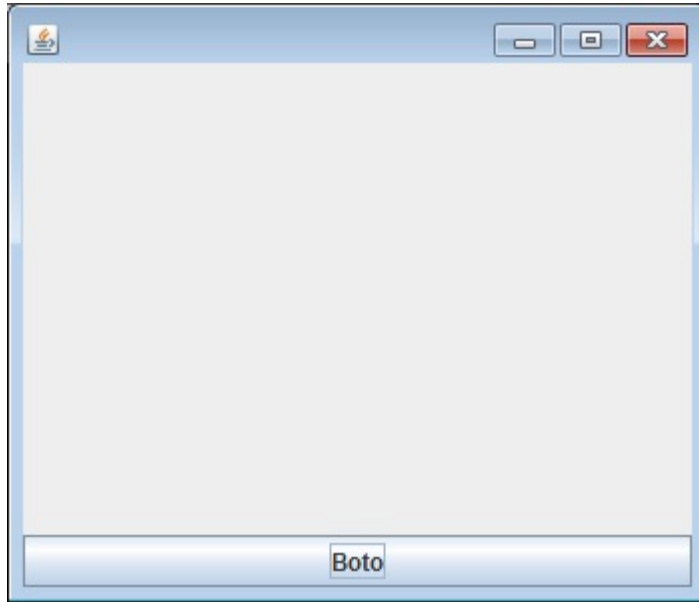
```
    }
```

```
}
```

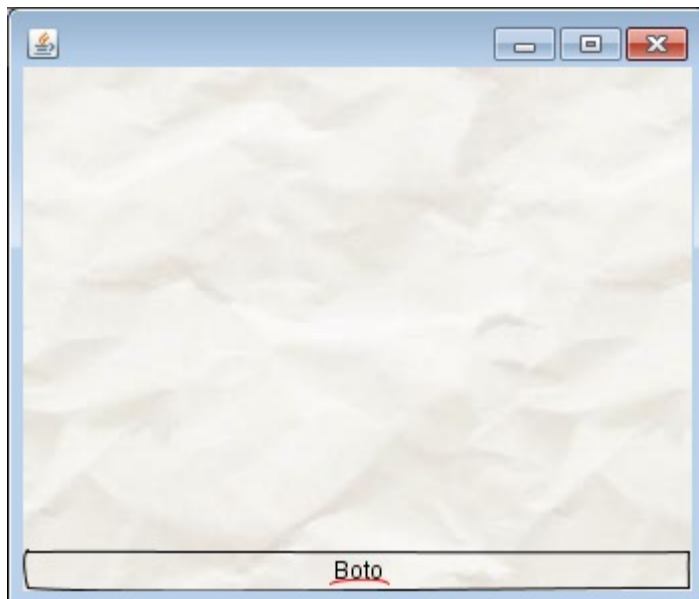


Creem una instància
d'aquesta classe
LookAndFeel

Look and feel

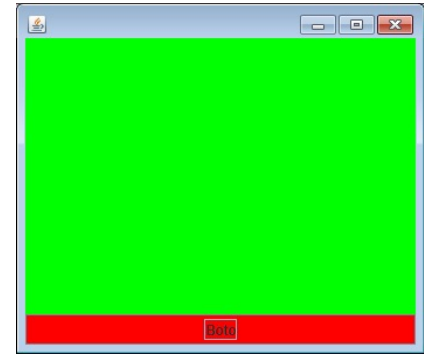


← Default Look And Fell



← NapkinLookAndFeel

Exemple Colors



```
public class ExempleLookFeelProves {  
  
    public static void main(String [] args){  
        ExempleLookFeelProves exemple = new ExempleLookFeelProves();  
        exemple.go();  
    }  
  
    public void go(){  
        JFrame frame = new JFrame();  
        JButton button = new JButton("Boto");  
        button.setBackground(Color.red);  
        frame.getContentPane().add(BorderLayout.SOUTH, button);  
  
        frame.getContentPane().setBackground(Color.green);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(350,300);  
        frame.setVisible(true);  
    }  
}
```

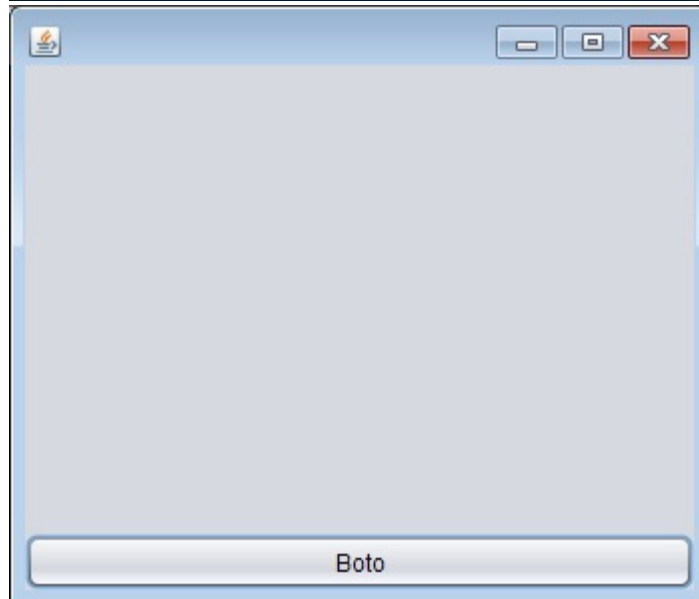
Canvia el color
de fons del
JButton

Canvia el color
de fons del
JFrame

Look and feel



Color Look And Fell



nimbus Look And Feel

Look and feel

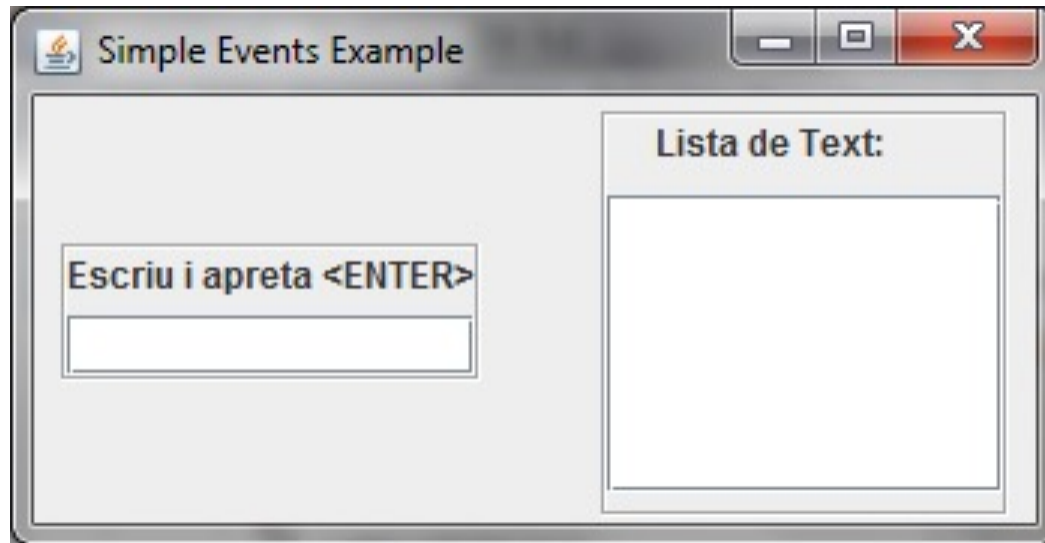
- Per fixar el look-and-feel podem utilitzar:

```
UIManager.LookAndFeelInfo plafinfo[] = UIManager.getInstalledLookAndFeels();
boolean nimbusfound = false;
int nimbusindex = 0;
for (int look = 0; look < plafinfo.length; look++) {
    if (plafinfo[look].getClassName().toLowerCase().contains("nimbus")) {
        nimbusfound = true;
        nimbusindex = look;
    }
}
try {
    if (nimbusfound) {
        UIManager.setLookAndFeel(plafinfo[nimbusindex].getClassName());
    } else {
        UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
    }
} catch (Exception e) { }
```

EXERCICIS

Exercici: Panel d'escriptura

Implementem aquesta interfície on podem escriure text al camp de l'esquerra i quan premem enter apareixerà a la lista de text.



Exercici: implementació A

```
// importa els símbols de AWT i Swing
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class SimpleEventsA{
    // ample i alt del frame
    static final int WIDTH=350;
    static final int HEIGHT=180;

    // Declara JTextField per a entrar text
    JTextField textField;
    // Declara JTextArea per rebre línies de text
    JTextArea textList;
    // Declara JScrollPane per a JTextArea
    JScrollPane pane;

    public static void main(String args[]) {
        SimpleEventsA gui = new SimpleEventsA();
        gui.go();
    } // Fi mètode main
```

← Per definir el tamany de la finestra.

Atributs

// Mètode go: aquí es fa quasi tot el treball

```
public void go(){
```

```
    /******* Crea un contenidor per a textfield *****/
```

```
    // Instancia un JPanel
```

```
    JPanel textPanel = new JPanel();
```

```
    // li posa un borde (per defecte no en te)
```

```
    textPanel.setBorder(BorderFactory.createEtchedBorder());
```

```
    // Fixa el layout del textPanel a BorderLayout
```

```
    textPanel.setLayout(new BorderLayout());
```

← JPanel

```
    // Crea una etiqueta i la afegeix al panell
```

```
    JLabel textTitle = new JLabel("Escriu i apreta <ENTER>");
```

```
    textPanel.add(textTitle, BorderLayout.NORTH);
```

← Afegim JLabel

```
    // Instancia un JTextField i l'afegeix al panell
```

```
    textField = new JTextField();
```

```
    textPanel.add(textField, BorderLayout.SOUTH);
```

← Afegim
JTextField

```
    // Afegeix un strut al textPanel com a marge inferior
```

```
    textPanel.add(Box.createVerticalStrut(6));
```

← Afegim Marge

```
/** ***** Crea un contenidor pel textArea ***** */
```

```
// Instancia un JPanel
```

```
JPanel listPanel = new JPanel();
```

```
// afegeix borde
```

```
listPanel.setBorder (BorderFactory.createEtchedBorder());
```

```
// Set el layout del textPanel
```

```
listPanel.setLayout(new BorderLayout(listPanel,BorderLayout.Y_AXIS));
```

```
// Crea una etiqueta i afegeix al panel
```

```
JLabel title = new JLabel("Llista de Text:");
```

```
listPanel.add(title);
```

```
// Afegeix un strut al BorderLayout
```

```
listPanel.add(Box.createVerticalStrut(10));
```

```
// Instancia una JTextArea sense text inicial
```

```
// 6 files, 10 columnes, i vertical scrollbars
```

```
textList = new JTextArea("", 6, 10);
```

```
// la fem read-only (només de lectura)
```

```
textList.setEditable(false);
```

```
// Afegeix textList a listPanel
pane = new JScrollPane(textList);
listPanel.add(pane);
// Afegeix un strut a listPanel com a margen inferior
listPanel.add(Box.createVerticalStrut(6));
```

```
// Afegeix un listener a textField quan se pulsa ENTER copia el text de
//textField a l'area de text. Les components estan interrelacionades
textField.addActionListener(new CampText());
```

← Registrem
escoltador

```
// Afegeix els 2 panels al frame, separats per strut
JFrame frame = new JFrame("Simple Events Example");
frame.setLayout (new FlowLayout());
frame.add(textPanel);
frame.add(Box.createHorizontalStrut(30));
frame.add(listPanel);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(WIDTH, HEIGHT);
frame.setVisible(true);
```

```
}//Fi mètode go
```

```
class CampText implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        // Afegeix el text de textField a textList
        textList.append(textField.getText());
        textList.append("\n");
        // Reset el textField
        textField.setText("");
    }
}
```

← Classe interna que
implementa la interfície
ActionListener

```
}// Fi de la classe SimpleEvent
```

Exercici: implementació B

```
// importa els símbols de AWT and Swing
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SimpleEventsB extends JFrame {

    // ample i alt del frame
    static final int WIDTH=350;
    static final int HEIGHT=180;
    // Declara JTextField per a entrar text
    JTextField textField;
    // Declara JTextArea per rebre línies de textField
    JTextArea textList;
    // Declara JScrollPane per a JTextArea
    JScrollPane pane;
```

```
public static void main(String args[]) {  
    SimpleEventsB frame = new SimpleEventsB("Simple Events Example");  
    // Standard adapter usat en quasi totes les  
    // aplicacions per a tancar la finestra  
    frame.addWindowListener(new WindowAdapter() {  
        @Override  
        public void windowClosing(WindowEvent e) {  
            System.exit(0);  
        }  
    });  
    // fixa el tamany de frame i el mostra  
    frame.setSize(WIDTH, HEIGHT);  
    frame.setVisible(true);  
} // Fi mètode main
```

← Mètode
main


```
// Constructor: aquí es fa quasi tot el treball
public SimpleEventsB(String lab) {
    // crida al constructor de JFrame: posa etiqueta
    super(lab);

    /******* Crea un contenidor per a textField *****/
    // Instancia un JPanel
    JPanel textPanel = new JPanel();
    // li posa un borde (per defecte no en te)
    textPanel.setBorder(BorderFactory.createEtchedBorder());
    // Fixa el layout del textPanel a BorderLayout
    textPanel.setLayout(new BorderLayout());
    // Crea una etiqueta i la afegeix al panel
    JLabel textTitle =new JLabel("Escriu i apreta <ENTER>");
    textPanel.add(textTitle, BorderLayout.NORTH);
    // Instancia un JTextField i afegeix a textPanel
    textField = new JTextField();
    textPanel.add(textField, BorderLayout.SOUTH);
    // Afegeix un strut al textPanel com a marge inferior
    textPanel.add(Box.createVerticalStrut(6));
```

```
/****** Crea un contenidor pel textArea *****/  
// Instancia un JPanel  
JPanel listPanel = new JPanel();  
// afegeix borde  
listPanel.setBorder (BorderFactory.createEtchedBorder());  
// Set el layout del textPanel  
listPanel.setLayout(new BorderLayout(listPanel,BorderLayout.Y_AXIS));  
// Crea una etiqueta i afegeix al panel  
JLabel title = new JLabel("Lista de Text:");  
listPanel.add(title);  
// Afegeix un strut al BorderLayout  
listPanel.add(Box.createVerticalStrut(10));  
// Instancia una JTextArea sense text inicial  
// 6 files, 10 columnes, i vertical scrollbars  
textList = new JTextArea("", 6, 10);  
// la fem read-only (només de lectura)  
textList.setEditable(false);  
// Afegeix textList a listPanel  
pane = new JScrollPane(textList);  
listPanel.add(pane);  
// Afegeix un strut a listPanel com a margin inferior  
listPanel.add(Box.createVerticalStrut(6));
```

```
// Afegeix un listener a textField quan se pulsa ENTER copia el text de  
//textField a l'area de text. Les components estan interrelacionades
```

```
textField.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        // Afegeix el text de textField a textList  
        textList.append(textField.getText());  
        textList.append("\n");  
        // Reset el textField  
        textField.setText("");  
    }  
});
```

← Classe interna
anònima que
implementa la
interfície
ActionListener

```
// Afegeix els 2 panels al frame, separats per strut  
Container c = getContentPane();  
c.setLayout (new FlowLayout());  
c.add(textPanel);  
c.add(Box.createHorizontalStrut(30));  
c.add(listPanel);  
} // Fi mètode constructor
```

```
} // Fi de la classe SimpleEvent
```

Referències

- Llibre “**Head First Java**”, Kathy Sierra & Bert Bates.
- “**Creating a GUI with JFC/Swing**” (The Swing Tutorial)
<http://java.sun.com/docs/books/tutorial/uiswing/>
- Using Swing Components:
<http://download.oracle.com/javase/tutorial/uiswing/components/>