

GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 3:

Programació Orientada a Events (4)

Sergio Sayago (basat en material de Laura Igual)

Departament de Matemàtiques i Informàtica

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

Índex Bloc 3:

Programació Orientada a Events

- Mecanismes d'interacció
 - Interacció mitjançant flux seqüencial
 - Interacció mitjançant programació orientada a events
- Programació d'Interfícies Gràfiques d'Usuari
- Model de gestió d'events: Exemple d'implementació d'una finestra.
- Events i Listeners
- Components i Contenedors
- Classes adapter i classes internes: Exemple d'implementació d'una finestra que es tanca.
- **Mes sobre swing components: Exemples**
- Layout manager
- Look and feel
- Panells i gràfics
- Animacions

Exercici 1

- (recuperem els exercicis proposats al final del tema 2.3)
- Implementeu una interfície gràfica d'usuari amb un botó i una àrea de text. Cada vegada que es prem el botó s'ha d'escriure el text "botó apretat" a l'àrea de text.

Exercici 1

- Implementeu una interfície gràfica d'usuari que contingui un botó i una àrea de text. Cada vegada que es prem el botó s'ha d'escriure el text “botó apretat” a l'àrea de text.
- Necessitem els següents components:
 - **JFrame**: serà el marc
 - **JPanel**: el necessitem per situar JTextArea al marc
 - **JButton**: el botó
 - **JScrollPane**: el necessitem per fer JTextArea “scrollable”
 - **JTextArea**: l'àrea de text

Exercici 1

- Implementeu una interfície gràfica d'usuari que contingui un boto i una àrea de text. Cada vegada que es prem el boto s'ha d'escriure el text "boto apretat" a l'àrea de text.
- Comentari / observació:
 - No podem afegir directament un JTextArea a un JFrame. Hem de treballar amb capes (**panells**)
 - **new JScrollPane(JTextArea)** -> això fa que el component JTextArea sigui "scrollable", però necessitem primer tenir el JTextArea
 - **new JPanel(JScrollPane)** -> afegim l'àrea de text, i scroll, al panell

Exercici 1

- Implementeu una interfície gràfica d'usuari que contingui un botó i una àrea de text. Cada vegada que es prem el botó s'ha d'escriure el text “botó apretat” a l'àrea de text.
- **JScrollPane:**
 - Necessitem indicar la seva “policy” de desplaçament (vertical / horitzontal)
 - Això ho farem amb els mètodes **setVerticalScrollBarPolicy** / **setHorizontalScrollBarPolicy**
 - I les constants que trobarem a `ScrollPaneConstants`
 - (tot això està a la documentació de JAVA)

```

public class ExempleJTextArea implements ActionListener{
    JTextArea text;
    public static void main (String [] args){
        ExempleJTextArea gui = new ExempleJTextArea();
        gui.go();
    }
    public void go(){
        JFrame frame = new JFrame();
        JPanel panel = new JPanel();
        JButton boto = new JButton("Apreta'l");
        boto.addActionListener(this);
        text = new JTextArea(10,20);
        text.setLineWrap(true);
        JScrollPane scroller = new JScrollPane(text);
        scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        panel.add(scroller);
        frame.getContentPane().add(BorderLayout.CENTER, panel);
        frame.getContentPane().add(BorderLayout.SOUTH, boto);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(350,300);
        frame.setVisible(true);
    }
    public void actionPerformed(ActionEvent ev){
        text.append("boto apretat \n");
    } // Fi classe ExempleJTextArea

```

```

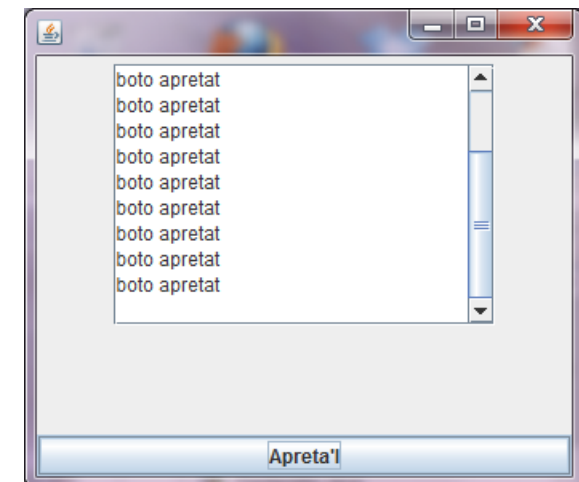
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.ScrollPaneConstants;

```

ajusta el text al
tamany de
l'area de text

Barra
lliscant

Concatena el text
(no borra el que hi
ha escrit)



Exercici 1: una altra opció

- També es pot crear una classe **BotoListener** que implementi la interfície ActionListener.
- En aquest exemple, aquesta classe haurà de ser interna.


```

public class ExempleJTextArea2{
    JTextArea text;
    public static void main(String [] args){
        ExampleJTextArea e = new ExampleJTextArea();
        e.go();
    }
    public void go(){
        JFrame frame = new JFrame();
        JPanel panel = new JPanel();
        //JPanel panel = new JPanel(new BorderLayout()); //PREFERRED!
        JButton boto = new JButton("Apreta'l");
        boto.addActionListener(new BotoListener());
        text = new JTextArea(10,20);
        JScrollPane scroller = new JScrollPane(text);
        text.setLineWrap(true);
        scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        panel.add(scroller);
        frame.getContentPane().add(BorderLayout.CENTER, panel);
        frame.getContentPane().add(BorderLayout.SOUTH, boto);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(350,300);
        frame.setVisible(true);}

    class BotoListener implements ActionListener{
        public void actionPerformed(ActionEvent ev){
            text.append("botó apretat \n");
        }
    }
} // Fi classe ExempleJTextArea

```

Classe interna

```

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.ScrollPaneConstants;

```



Exercici 2

- Implementeu una interfície gràfica d'usuari que contingui un botó i una àrea de text. Quan es prem el botó s'ha d'escriure el text contingut en ell a l'àrea de text.
- Per practicar, podeu fer que l'escoltador del botó sigui una classe interna anònima

```
public class Exemple2 implements ActionListener{
```

```
    JTextArea text;
```

```
    JButton boto;
```

La declaració del botó
s'ha de fer aquí

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

```
        ExempleJTextArea gui = new ExempleJTextArea();
```

```
        gui.go();
```

```
    }
```

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

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

```
        JPanel panel = new JPanel();
```

```
        boto = new JButton("Apreta'l");
```

```
        boto.addActionListener(this);
```

```
        text = new JTextArea(10,20);
```

```
        text.setLineWrap(true);
```

```
        JScrollPane scroller = new JScrollPane(text);
```

```
        scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
```

```
        scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
```

```
        panel.add(scroller);
```

```
        frame.getContentPane().add(BorderLayout.CENTER, panel);
```

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

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

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

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

```
    public void actionPerformed(ActionEvent ev){
```

```
        text.append(boto.getText());
```

```
    } } // Fi classe ExempleJTextArea
```

```
import java.awt.BorderLayout;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;
```

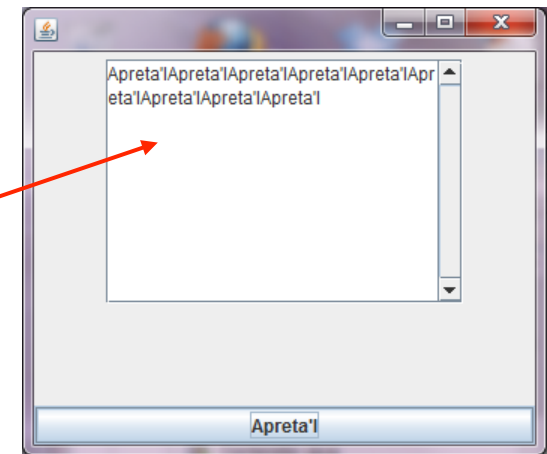
```
import javax.swing.JFrame;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JScrollPane;
```

```
import javax.swing.JTextArea;
```

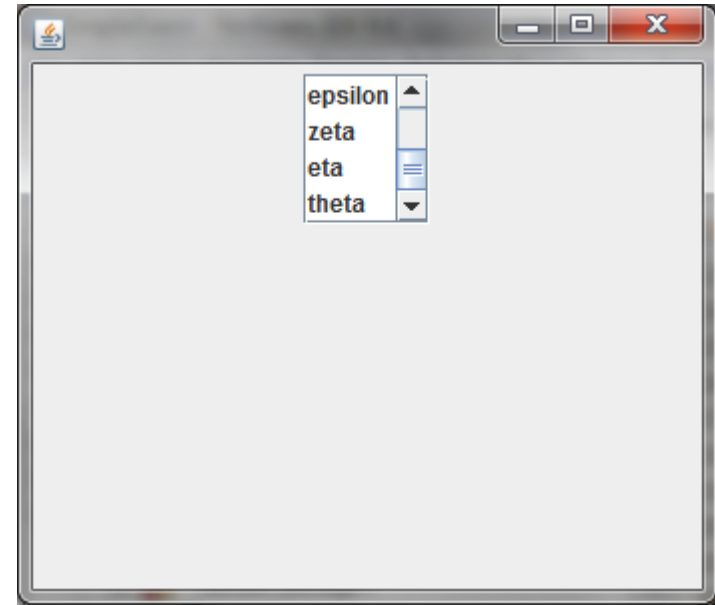
```
import javax.swing.ScrollPaneConstants;
```



MES COMPONENTS

Exercici 3

- Implementeu una interfície gràfica d'usuari que conté una llista amb els Strings: "alpha", "beta", "gamma", "delta", "epsilon", "zeta", "eta", "theta" i que quan seleccionem algun dels elements d'aquesta llista s'imprimeix per pantalla.



alpha
beta
alpha
gamma
delta
alpha

JList

- La creació d'un JList es fa en passos:

- **Constructor:**

`String [] entradesLlista = {"alpha", "beta", "gamma"}; // necessitem tenir la llista per crear l'objecte. Aquesta llista`

`JList list = new JList(entradesLlista);`

- **Afegir-li una scrollbar vertical:**

`JScrollPane scroller = new JScrollPane (list);`

`scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);`

`scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);`

- **Especificar el model o tipus de selecció**

`list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION); // Java ens permet altres modes de selecció`

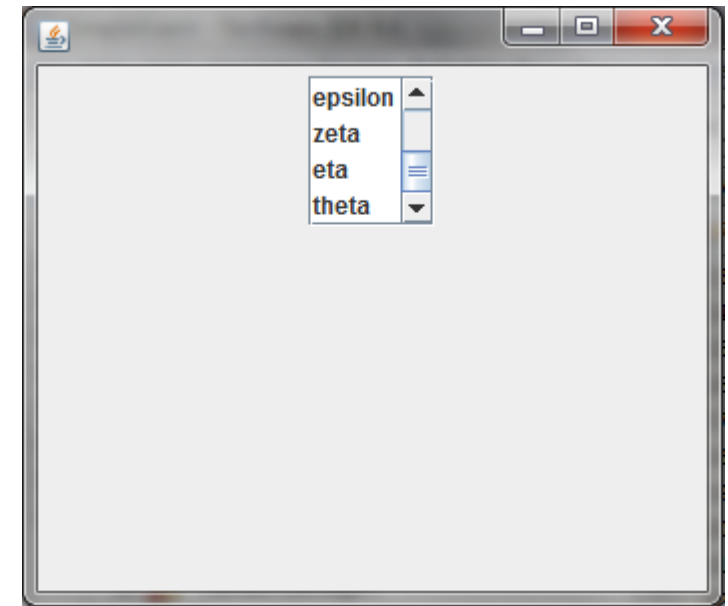
JList

- Per capturar els canvis (per exemple, si l'usuari ha seleccionat un element de la llista), necessitem un escoltador
- Això significa implementar la interfície **ListSelectionListener**
- A la documentació, veureu que aquesta interfície té únicament un mètode, **valueChanged (ListSelectionEvent lse)**

```

public class ExempleJList implements ListSelectionListener{
    String [] entradesLlista = {"alpha", "beta", "gamma", "delta", "epsilon", "zeta","eta", "theta"};
    JList list;
    public static void main (String [] args){
        ExempleJList gui = new ExempleJList ();
        gui.go(); }
    public void go(){
        list = new JList(entradesLlista );
        JScrollPane scr = new JScrollPane(list);
        JPanel panel = new JPanel();
        JFrame frame = new JFrame();
        scr.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        scr.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        panel.add(scr);
        list.setVisibleRowCount(4);
        list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        list.addListSelectionListener(this);
        frame.getContentPane().add(BorderLayout.CENTER, panel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(350,300);
        frame.setVisible(true);
    }
    public void valueChanged(ListSelectionEvent lse){
        if(lse.getValueIsAdjusting()){
            String selection = (String) list.getSelectedValue();
            System.out.println(selection); } }
} // Final classe Exemple

```



Índex Bloc 3:

Programació Orientada a Events

- Mecanismes d'interacció
 - Interacció mitjançant flux seqüencial
 - Interacció mitjançant programació orientada a events
- Programació d'Interfícies Gràfiques d'Usuari
- Model de gestió d'events: Exemple d'implementació d'una finestra.
- Events i Listeners
- Components i Contenedors
- Classes adapter i classes internes: Exemple d'implementació d'una finestra que es tanca.
- Mes sobre swing components: Exemples
- **Layout manager**
- Look and feel
- Panells i gràfics
- Animacions

Layout Manager

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

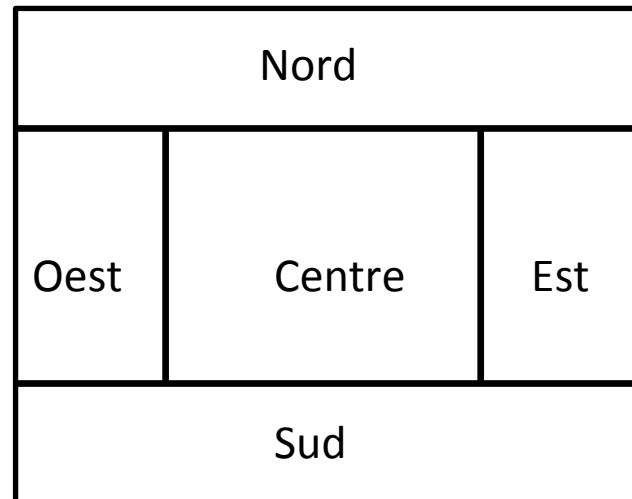
Exemple:

Si un frame conté un panell que conté un botó, el layout manager del panell controlarà la mida i posició del botó, mentre que el layout manager del frame controlarà la mida 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 les mides preferides pels components.
- És el layout manager per defecte per a un **frame**

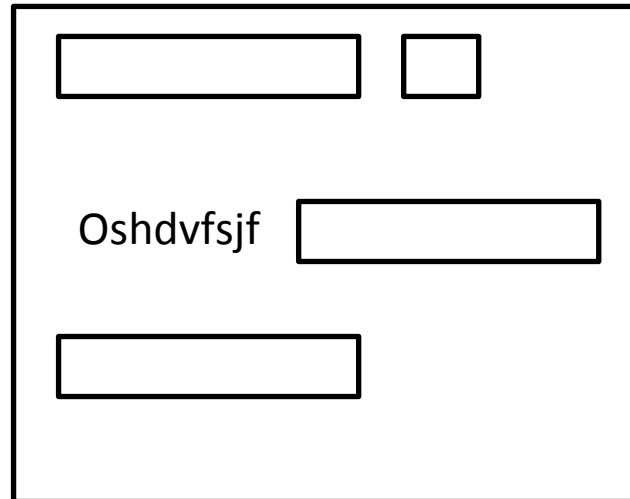
Exemple: BorderLayout



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**

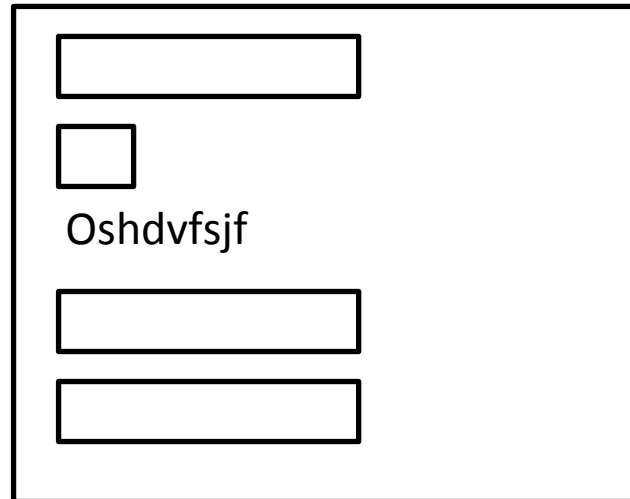
Example: FlowLayout



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



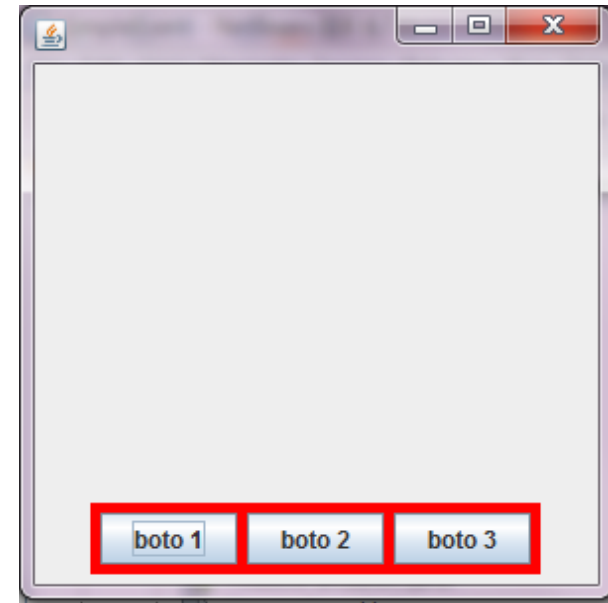
Canvi de Layout

- Per fer un canvi de Layout manager - exemple:

```
JFrame frame = new JFrame();  
frame.getContentPane().setLayout(new  
BoxLayout(frame.getContentPane(), BoxLayout.Y_AXIS)
```

Exemple

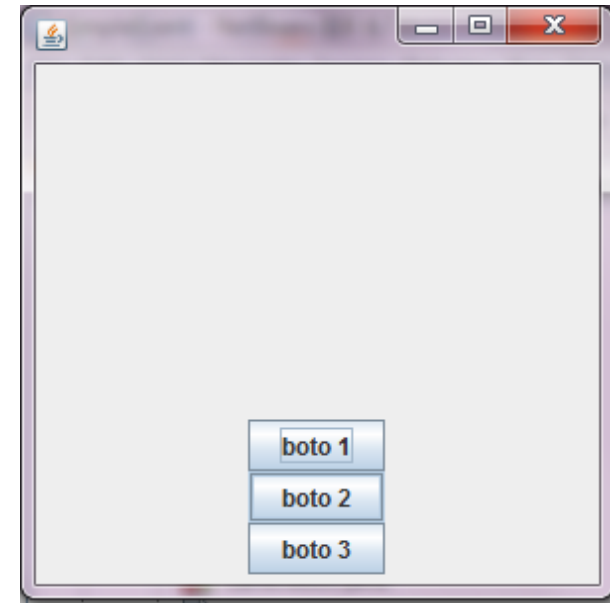
```
public class ProvaLayout {  
    public static void main (String[] args) {  
        ProvaLayout gui = new ProvaLayout();  
        gui.go();  
    }  
    public void go(){  
        JFrame frame = new JFrame();  
  
        JPanel panelA = new JPanel();  
        JPanel panelB = new JPanel();  
  
        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.

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.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 definit dues vegades en el disseny.
- **SpringLayout** és un controlador de distribució flexible dissenyat per ser utilitzat pels constructors de GUI. 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 dret 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>

Índex Bloc 3:

Programació Orientada a Events

- Mecanismes d'interacció
 - Interacció mitjançant flux seqüencial
 - Interacció mitjançant programació orientada a events
- Programació d'Interfícies Gràfiques d'Usuari
- Model de gestió d'events: Exemple d'implementació d'una finestra.
- Events i Listeners
- Components i Contenedors
- Classes adapter i classes internes: Exemple d'implementació d'una finestra que es tanca.
- Mes sobre swing components: Exemples
- Layout manager
- **Look and feel**
- Panells i gràfics
- Animacions

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.
- *UIManager*

Look and feel

- Java proporciona:

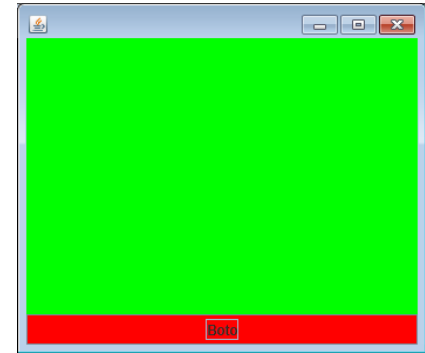
- Metal
- Nimbus
- Motif
- Windows
- Windows classic



Van a totes les
plataformes

Exemple Colors

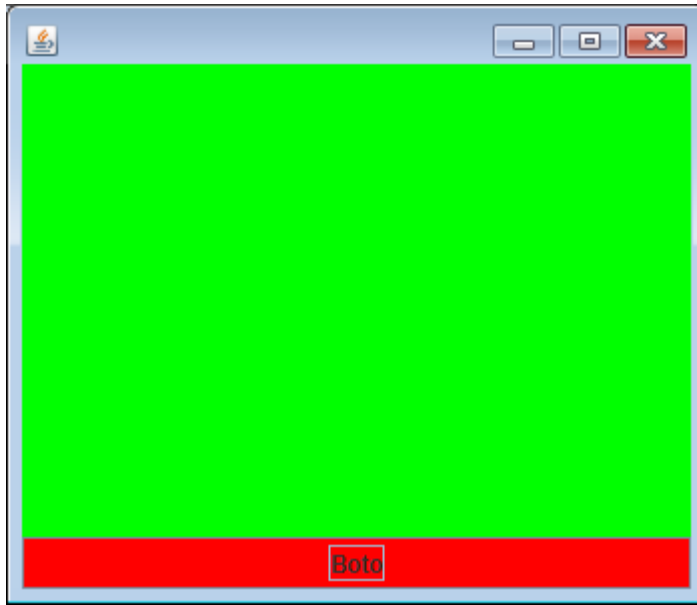
```
public class ExempleLookFeelProves {  
  
    public static void main(String [] args){  
        ExempleLookFeelProves exemple = new ExempleLookFeelProves();  
        exemple.go();  
    }  
  
    public void go(){  
        JFrame frame = new JFrame();  
        JButton button = new JButton("Boto");  
        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);  
    }  
}
```



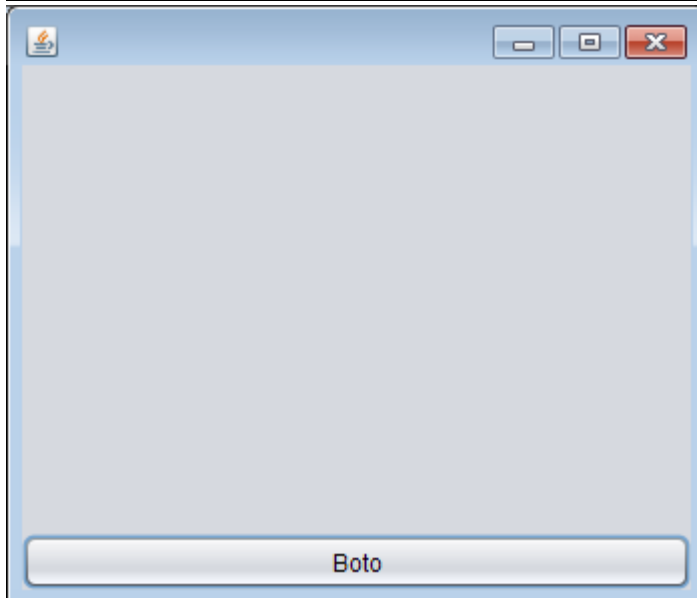
Canvia el color
de fons del
JButton

Canvia el color
de fons del
JFrame

Look and feel



Color Look And Feel

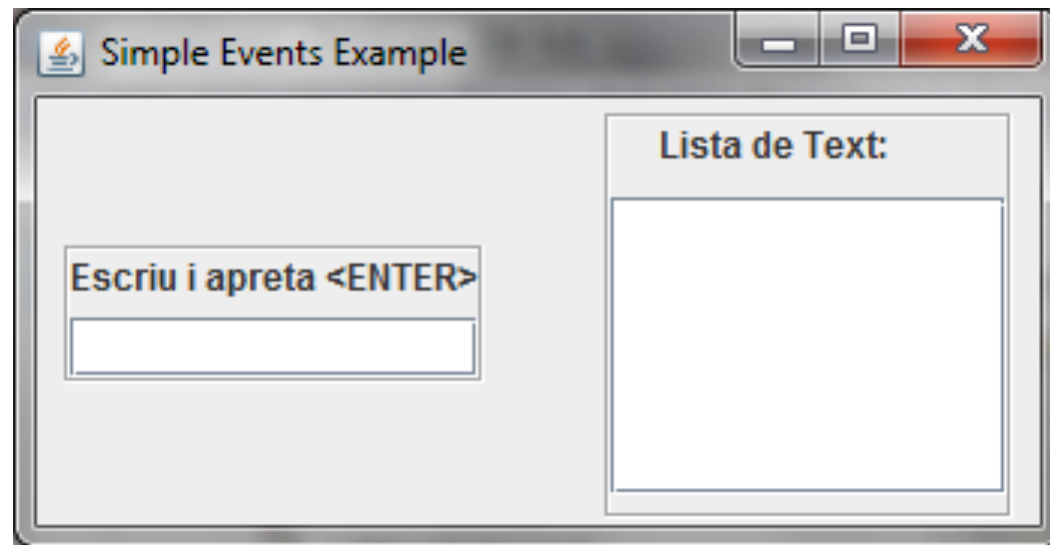


nimbus Look And Feel

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.



Panell d'escriptura: versió 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 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 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
```

Panell d'escriptura: versió 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 text
    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 BoxLayout(listPanel,BoxLayout.Y_AXIS));  
// Crea una etiqueta i afegeix al panel  
JLabel title = new JLabel("Lista de Text:");  
listPanel.add(title);  
// Afegeix un strut al BoxLayout  
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 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 del bloc 3

- Llibre “**Head First Java**”, Kathy Sierra & Bert Bates.
- *Apunts*: “**Aprenda Java como si estuviera en Primero**” (Universidad de Navarra):
<http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf>
- “**Creating a GUI with JFC/Swing**” (The Swing Tutorial)
<http://java.sun.com/docs/books/tutorial/uiswing/>
- The Swing Connection
<http://java.sun.com/javase/technologies/desktop/articles.jsp>
- Components:
<http://download.oracle.com/javase/tutorial/uiswing/components/>