

GRAU D'ENGINYERIA INFORMÀTICA

PROGRAMACIÓ II

Bloc 3:

Programació Orientada a Events (3)

Laura Igual

Departament de Matemàtica Aplicada i Anàlisi

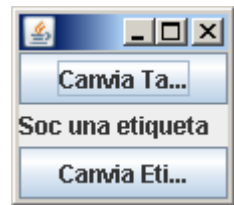
Facultat de Matemàtiques

Universitat de Barcelona

Exemple 4

- Com gestionem events `ActionEvent` per a dos botons quan cada botó necessita fer una cosa diferent?

Exemple 4: Com gestionem events ActionEvent per a dos botons quan cada botó necessita fer una cosa diferent?



Exemple 4: Opició A

- Implementar dos mètodes actionPerformed()

```
class LaMevaGUI implements ActionListener {  
    // codi aquí  
    public void actionPerformed(ActionEvent ev) {  
        frame.setSize(300, 300);  
    }  
    public void actionPerformed(ActionEvent ev) {  
        etiqueta.setText("Etiqueta canviada");  
    }  
} // fi de la classe LaMevaGUI
```

Però això és impossible

Exemple 4: Opició B

- Registrar el mateix listener amb dos botons

```
class LaMevaGUI implements ActionListener {
```

```
    // declarem variables aquí
```

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

```
        //construir gui
```

```
        boto1 = new JButton();
```

```
        boto2 = new JButton();
```

```
        boto1.addActionListener(this);
```

```
        boto2.addActionListener(this);
```

```
        // més codi aquí
```

```
    }
```

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

```
        if(ev.getSource() == boto1 ) {
```

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

```
        }else{
```

```
            etiqueta.setText("Etiqueta canviada");
```

```
        }
```

```
    }
```

```
} // fi de la classe LaMevaGUI
```

Registre'm el mateix listener

Mirem quin event és

Opció correcta, però pot arribar a ser complicada de gestionar

Exemple 4: Opició C

- Crear dues classes ActionListener separades

```
class LaMevaGUI {  
    private JFrame frame;  
    private JLabel etiqueta;  
    public static void main (String[] args){  
        LaMevaGUI2 gui = new LaMevaGUI2();  
        gui.go();  
    }  
    public void go(){  
        frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JButton boto1 = new JButton("Canvia Tamany");  
        JButton boto2 = new JButton("Canvia Etiqueta");  
        etiqueta = new JLabel("Soc una etiqueta");  
        frame.getContentPane().add(BorderLayout.NORTH, boto1);  
        frame.getContentPane().add(BorderLayout.SOUTH, boto2);  
        frame.getContentPane().add(BorderLayout.CENTER, etiqueta);  
        boto1.addActionListener(new Boto1Listener());  
        boto2.addActionListener(new Boto2Listener());  
        frame.setSize(100, 100);  
        frame.setVisible(true);  
    }  
} // fi de la classe
```

```
public class Boto1Listener implements  
ActionListener {  
    public void actionPerformed(ActionEvent ev){  
        frame.setSize(300, 300);  
    }  
} // fi de la classe
```

```
public class Boto2Listener implements  
ActionListener {  
    public void actionPerformed(ActionEvent ev){  
        etiqueta.setText("Etiqueta canviada");  
    }  
} // fi de la classe
```

Problema! Aquesta classe no té
referència a la variable etiqueta i
frame

Exemple 4: Opició D

- Crear dues classes ActionListener separades

Arreglant el problema d'abans

```
public class LaMevaGUI2 {  
    private JFrame frame;  
    private JLabel etiqueta;  
    public static void main (String[] args){  
        LaMevaGUI2 gui = new LaMevaGUI2();  
        gui.go();  
    }  
    public void go(){  
        frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JButton boto1 = new JButton("Canvia Tamany");  
        JButton boto2 = new JButton("Canvia Etiqueta");  
        boto1.addActionListener(new Boto1Listener(this));  
        boto2.addActionListener(new Boto2Listener(this));  
        etiqueta = new JLabel("Soc una etiqueta");  
        frame.getContentPane().add(BorderLayout.NORTH, boto1);  
        frame.getContentPane().add(BorderLayout.SOUTH, boto2);  
        frame.getContentPane().add(BorderLayout.CENTER, etiqueta);  
        frame.setSize(100, 100);  
        frame.setVisible(true);  
    }  
}
```

```
    public JLabel getEtiqueta(){  
        return etiqueta;  
    }  
    public void setEtiqueta(JLabel etiqueta){  
        this.etiqueta = etiqueta;  
    }  
    public JFrame getFrame(){  
        return frame;  
    }  
    public void setFrame(JFrame frame){  
        this.frame = frame;  
    }  
} // fi de la classe
```

Exemple 4: Opició D

- Crear dues classes ActionListener separades

```
public class Boto1Listener implements ActionListener {  
    LaMevaGUI2 gui;  
  
    public Boto1Listener(LaMevaGUI2 gui){  
        this.gui = gui;  
    }  
  
    public void actionPerformed(ActionEvent ev) {  
        gui.getFrame().setSize(300, 300);  
    }  
} // fi de la classe
```

Atenció! Aquestes classes han de rebre l'objecte LaMevaGUI2 per poder accedir als atributs de la mateixa.

Potser fa **difícil** entendre el codi.

```
public class Boto2Listener implements ActionListener {  
    LaMevaGUI2 gui;  
  
    public Boto2Listener(LaMevaGUI2 gui){  
        this.gui = gui;  
    }  
  
    public void actionPerformed(ActionEvent ev) {  
        gui.getEtiqueta().setText("Etiqueta canviada");  
    }  
} // fi de la classe
```


Exemple 4: Opició E

- Crear dues classes ActionListener **internes amb nom**.

```

import java.awt.BorderLayout;
import javax.swing.*;
import java.awt.event.*;
public class LaMevaGUI {
    JFrame frame;
    JLabel etiqueta;
    public static void main(String[] args){
        LaMevaGUI gui = new LaMevaGUI();
        gui.go();
    }
    public void go() {
        frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton botoTamany = new JButton("Canvia Tamany");
        botoTamany.addActionListener(new
BotoTamanyListener());
        JButton botoEtiqueta = new JButton("Canvia Etiqueta");
        botoEtiqueta.addActionListener(new
BotoEtiquetaListener());
        etiqueta = new JLabel("Soc una etiqueta");
        frame.getContentPane().add(BorderLayout.NORTH,
botoTamany);
        frame.getContentPane().add(BorderLayout.SOUTH,
botoEtiqueta);
        frame.getContentPane().add(BorderLayout.CENTER,
etiqueta);
        frame.setSize(100, 100);
        frame.setVisible(true);
    }
} // Fi mètode go

```

Classes internes

```

class BotoTamanyListener implements ActionListener
{
    public void actionPerformed(ActionEvent ev) {
        frame.setSize(300, 300);
    }
}
class BotoEtiquetaListener implements ActionListener
{
    public void actionPerformed(ActionEvent ev) {
        etiqueta.setText("Etiqueta canviada");
    }
}
} // Fi de la classe LaMevaGUI

```



Exemple 4: Opició E

Exemple 4: Opició F

- Crear dues classes ActionListener **internes anònimes**.
- Aquesta és l'opció més comuna quan tenim moltes components en la GUI.

```

import java.awt.BorderLayout;
import javax.swing.*;
import java.awt.event.*;

public class LaMevaGUI4 {
    JFrame frame;
    JLabel etiqueta;
    public static void main(String[] args){
        LaMevaGUI4 gui = new LaMevaGUI4();
        gui.go();
    }
    public void go() {
        frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JButton botoTamany = new JButton("Canvia Tamany");
        botoTamany.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ev) {
                frame.setSize(300, 300);
            }
        });
        JButton botoEtiqueta = new JButton("Canvia Etiqueta");
        botoEtiqueta.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ev) {
                etiqueta.setText("Etiqueta canviada");
            }
        });
    }
}

```

```

etiqueta = new JLabel("Soc una etiqueta");

frame.getContentPane().add(BorderLayout.NORTH,
    botoTamany);
frame.getContentPane().add(BorderLayout.SOUTH,
    botoEtiqueta);

frame.getContentPane().add(BorderLayout.CENT
ER, etiqueta);

frame.setSize(100, 100);
frame.setVisible(true);
} // Fi mètode go
} // Fi de la classe LaMevaGUI

```

Classes internes anònimes

Fa **més fàcil** entendre el codi.



Exemple 4: Opició F

Comentaris sobre les classes internes

- S'ha d'implementar tots els mètodes de la interfície
- Si el codi utilitzat per a implementar la manipulació d'events té unes poques línies es sol utilitzar una **classe interna anònima**.
- No hi ha garantia de quin Listener és notificat primer.
- No escriure codi contant amb un ordre específic.

COMPONENTS

Components

Les components estan classificades com

1. Contenidor d'alt nivell

Cada contenidor d'alt nivell té un JRootPane que és l'arrel de la jerarquia de contenidors.

2. Contenidor intermedi

3. Contenidor específic

4. Control bàsic

5. Displays no editables

6. Displays interactius

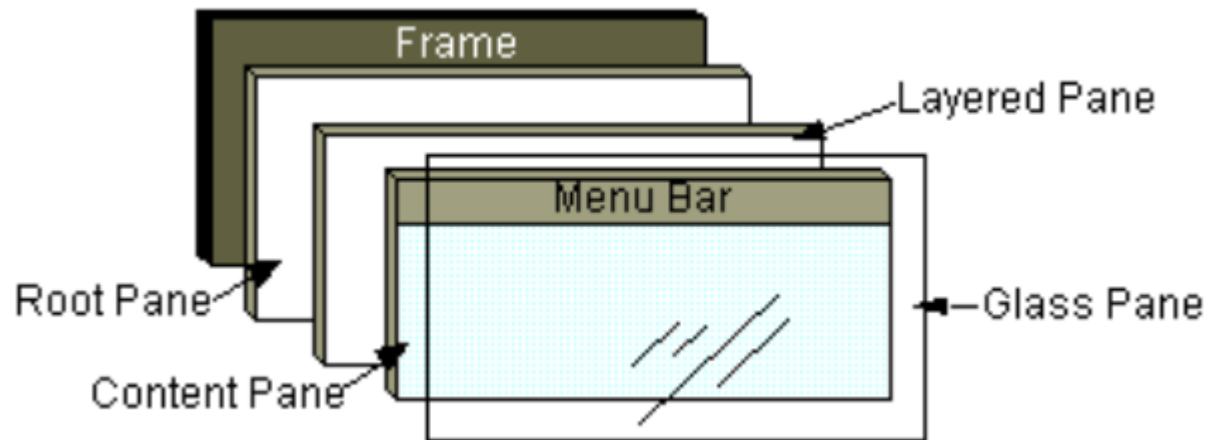
Contenidors d'alt nivell

- JFrame - Frames o Marcos (finestres)
 - Finestra principal
- JDialog – Diàlegs
 - Una finestra independent que ens servirà per a proporcionar informació temporal a la finestra principal de l'aplicació. La majoria serveixen per mostrar un missatge d'error o warnings als usuaris, però poden mostrar imatges, arbres de directoris, etc.
- JApplet – Applets
 - Una component d'una aplicació que s'executa en el context d'un altre programa, per exemple un navegador web.

(<http://download.oracle.com/javase/tutorial/deployment/TOC.html>)

JRootPane

- El JRootPane es crea quan s'instancia un dels contenidors d'alt nivell de Swing.

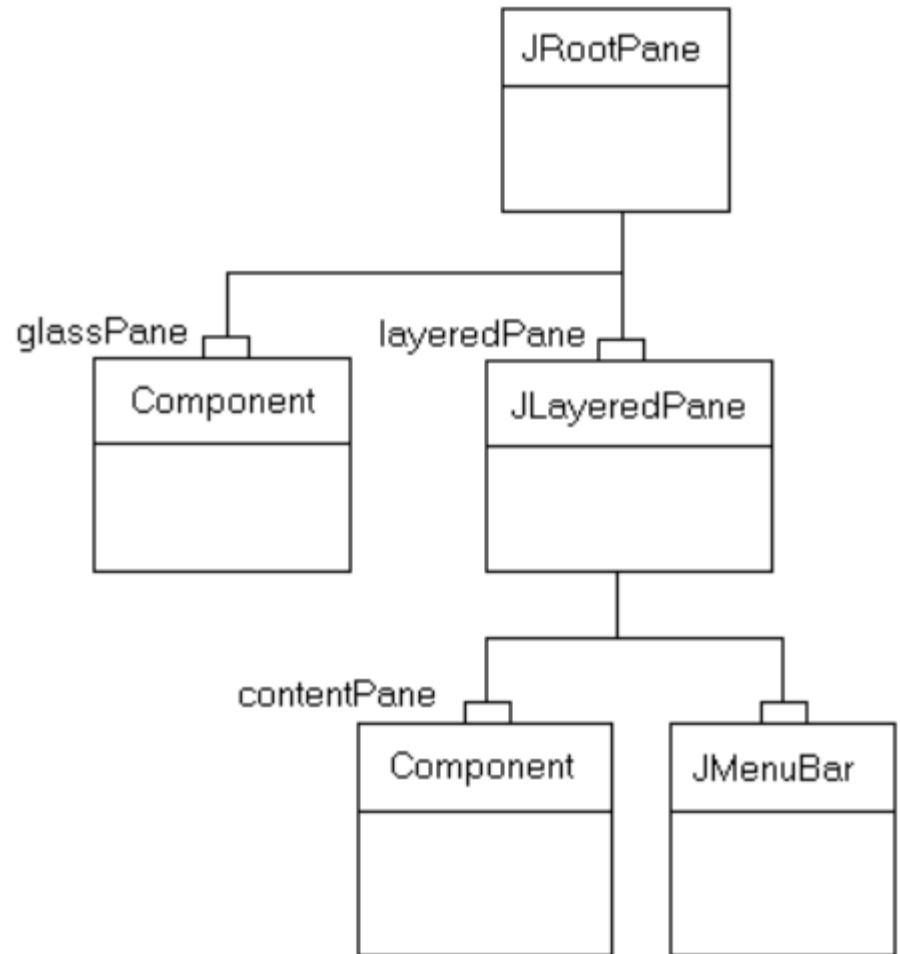


<http://download.oracle.com/javase/tutorial/uiswing/components/rootpane.html>

Jerarquia de JRootPane

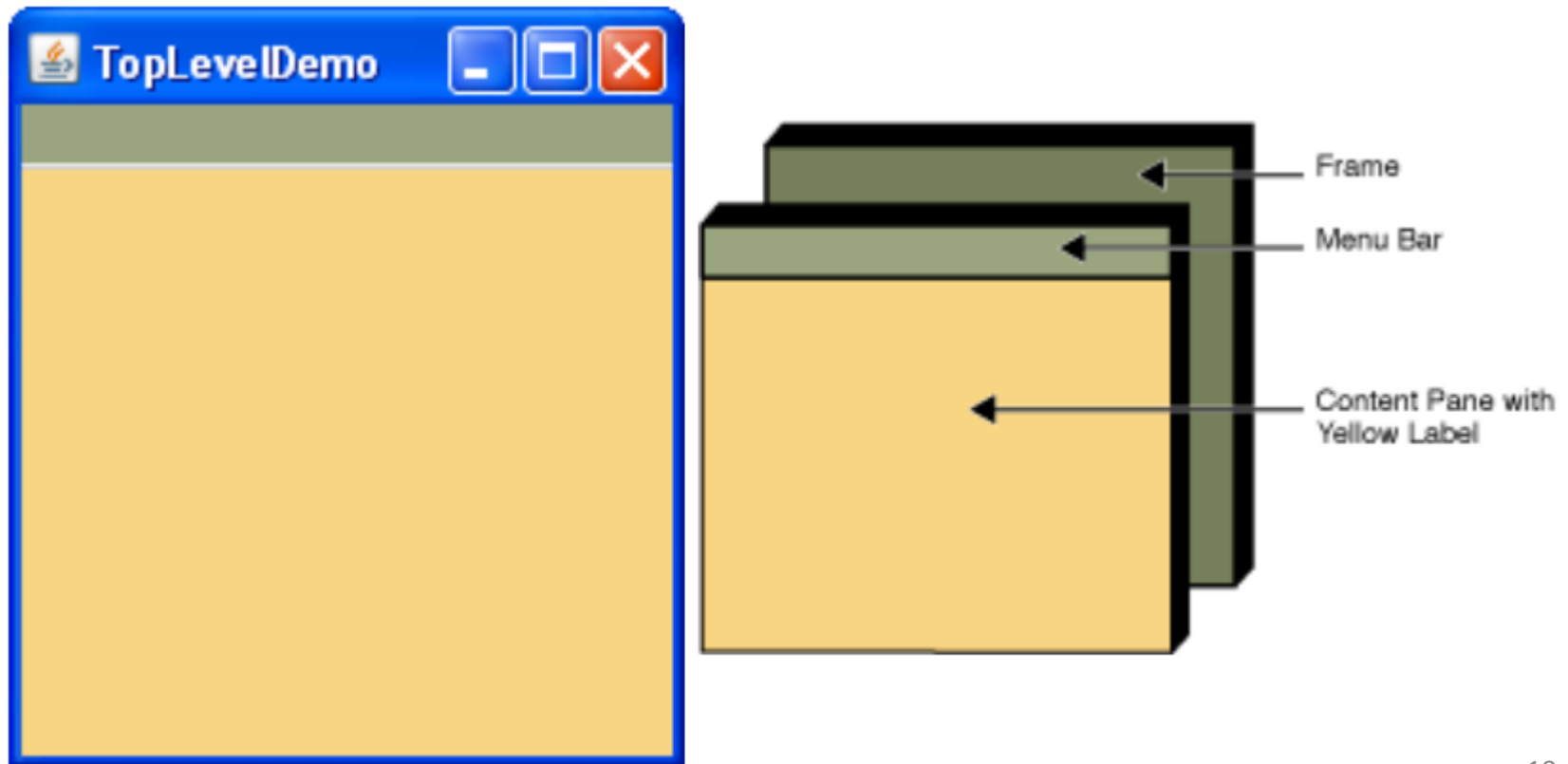
- Totes els contenidors d'alt nivell deleguen les seves operacions a un JRootPane
- Per afegir components al JRootPane, s'afegeix l'objecte al contentPane del JRootPane de la següent manera:

`rootPane.getContentPane().add(child)`



JRootPane

- El JRootPane es crea quan s'instancia un dels contenidors d'alt nivell de Swing.



Contenidors intermedis

- Normalment, s'utilitzen per a agrupar components, perquè les components estan relacionades o només perquè agrupar-les fa que la distribució sigui més senzilla.
- Un panell pot fer servir qualsevol controlador de distribució (layout manager), ho veurem després.
 - Panell (panel)
 - Panell lliscants (scroll pane)
 - Panell dividit (split pane)
 - Panell amb solapes (tabbed pane)
 - Barra d'eines (tool bar)

Control bàsic

- Botons (Buttons),
 - Caixes combo (Combo boxes),
 - Barra lliscant (Sliders),...
-
- Elements d'interfície que els usuaris poden manipular per prémer un botó, seleccionar una opció o fixar un valor.

Displays no editables

- Etiquetes (labels)
- Barres de progrés (progress bars)
- Pistes d'eines (tool tips)

Per afegir una pista d'eina a un botó:

```
b1.setToolTipText("Clica aquest botó per desactivar el botó del mig.");
```

Displays interactius

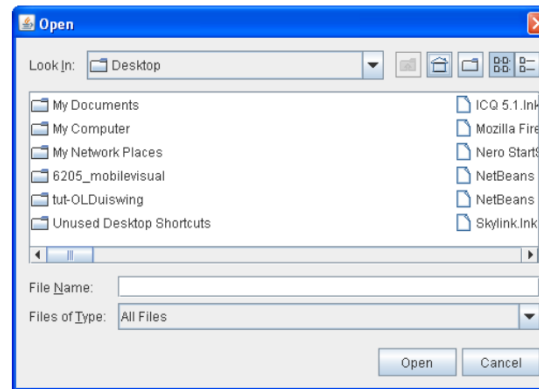
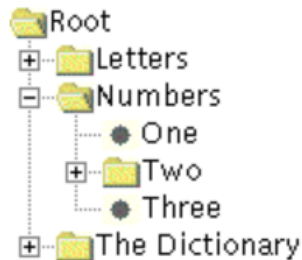
- Selector de colors (JColorChooser)
- Taula (JTable)
- Text (JTextComponent)
- Selector de fitxers (JFileChooser)
- Arbres (JTree)

The Header contains Column labels

First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>

Each Cell displays a data item

Each Column displays one type of data



LAYOUT MANAGER

Layout Manager

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

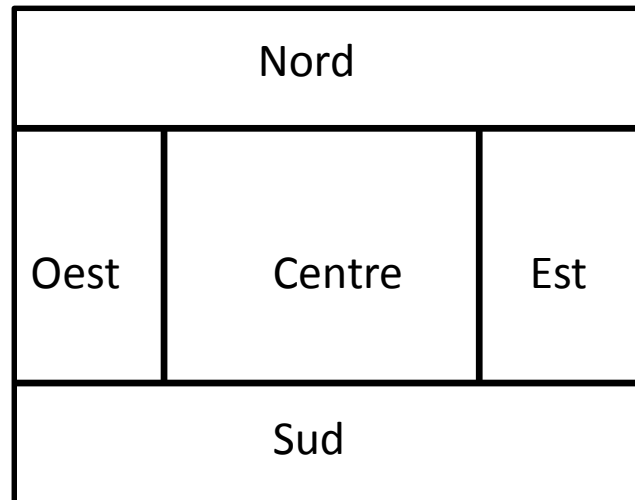
Exemple:

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**

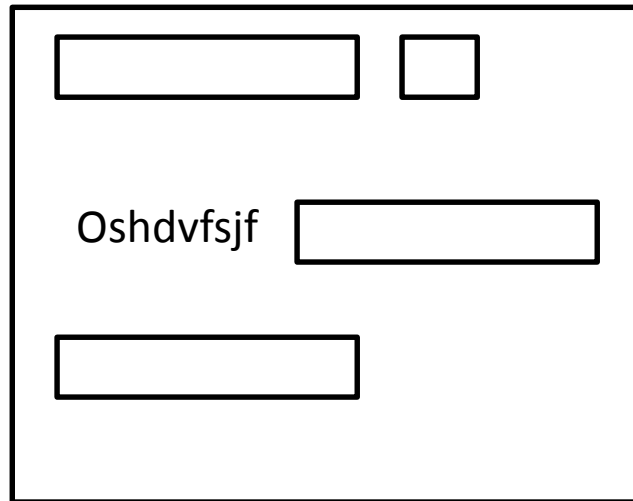
Exemple: BorderLayout



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**

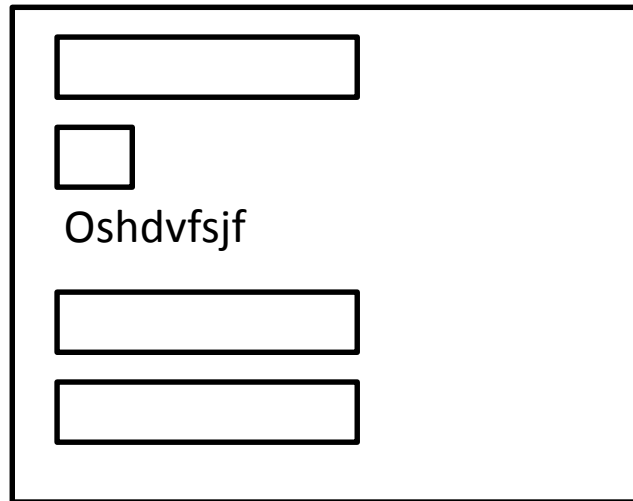
Example: FlowLayout



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



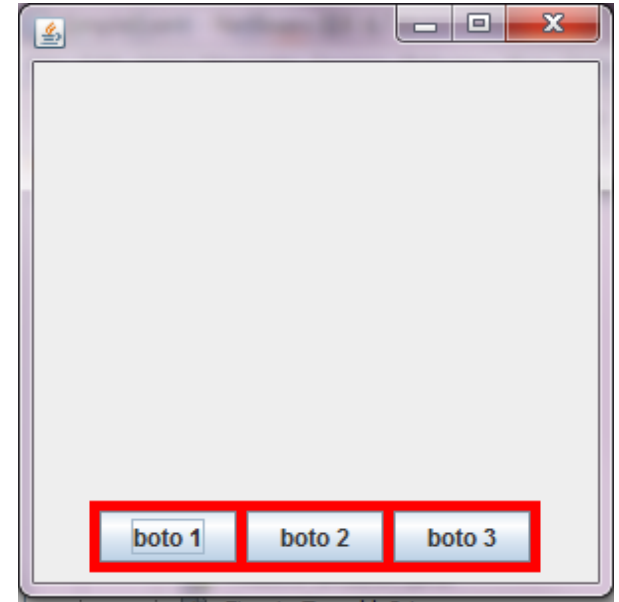
Canvi de Layout

- Per fer un canvi de Layout manager:

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


Exemple

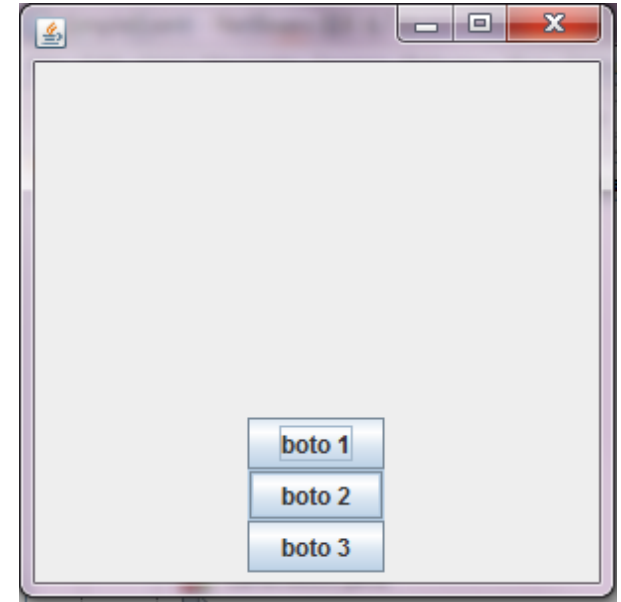
```
public class ProvaLayout {  
    public static void main (String[] args) {  
        ProvaLayout gui = new ProvaLayout();  
        gui.go();  
    }  
    public void go(){  
        JFrame frame = new JFrame();  
  
        JPanel panelA = new JPanel();  
        JPanel panelB = new JPanel(); 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



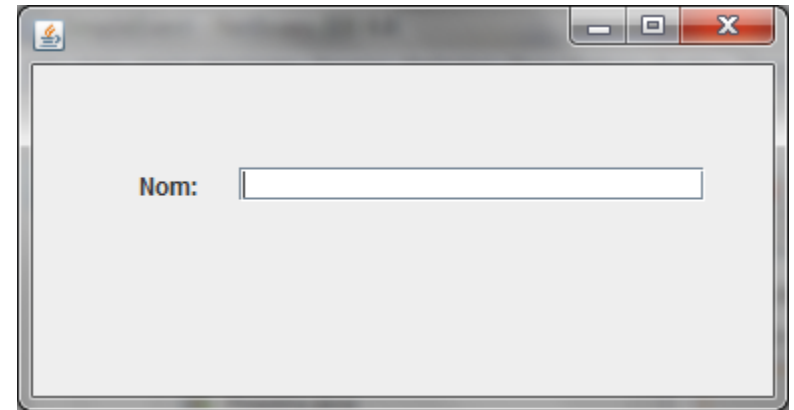
MÉS SOBRE SWING COMPONENTS: EXAMPLES

JTextField

```
JTextField camp = new JTextField(20);  
JTextField camp = new JTextField("El meu missatge");
```

← 20 columnes

- Mètodes importants:
- `String getText()` i `setText(String str)`
Permeten establir o obtenir el text del component
- `addActionListener(ActionListener al)`,
- `removeActionListener(ActionListener al)`
Permet registrar o borrar l'objecte que gestionarà l'event
- `selectAll()`, `select(int start, int end)`
Selecciona tot o part del text
- `requestFocus()` (de la classe `Component`)
Permet fer des del programa que un component obtinga el Focus.



JTextArea

- Pot contenir més d'una línia de text
- Per afegir barres lliscants s'ha d'afegir un JScrollPane

- Constructor:

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

10 files 20 columnes

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

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

Estableix la política d'ajustament de línies de l'àrea de text.

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

```
O ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER,  
  ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
```

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

```
text.setText("Canviem el text");
```

```
text.append("botó apretat");
```

```
text.selectAll();
```

```
text.requestFocus();
```

L'ample d'una columna és igual a l'ample d'un caràcter, en la font particular que s'està utilitzant.

JTextArea

- Afegir-li una scrollbar vertical:

```
JScrollPane scroller = new JScrollPane (list);  
scroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);  
scroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);  
panel.add(scroller);
```

Exemple 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.

```
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;
```

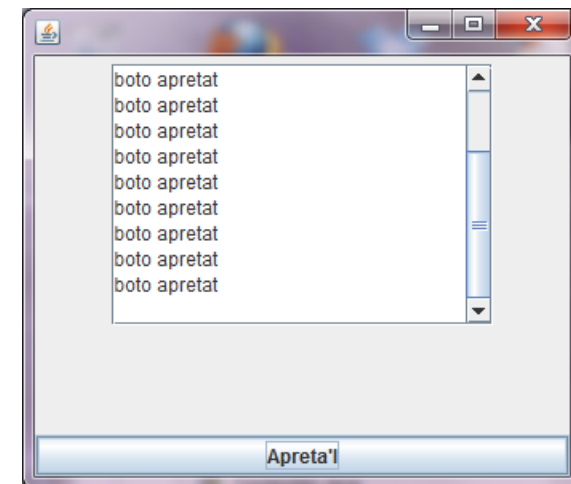
```
import javax.swing.JTextArea;
```

```
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)



Exemple 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){
        ExempleJTextArea e = new ExempleJTextArea();
        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

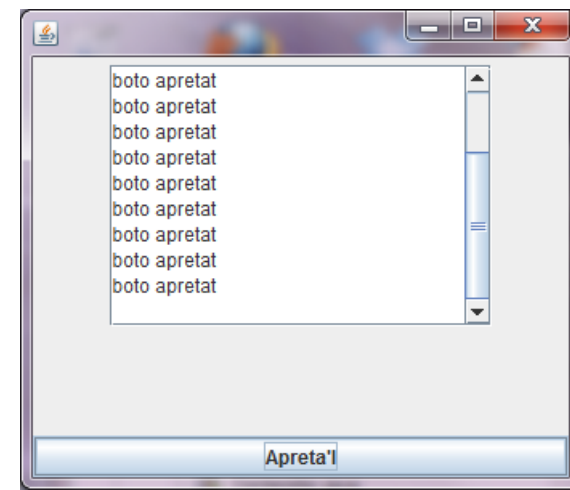
```

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;

```



Exemple 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 fer!