

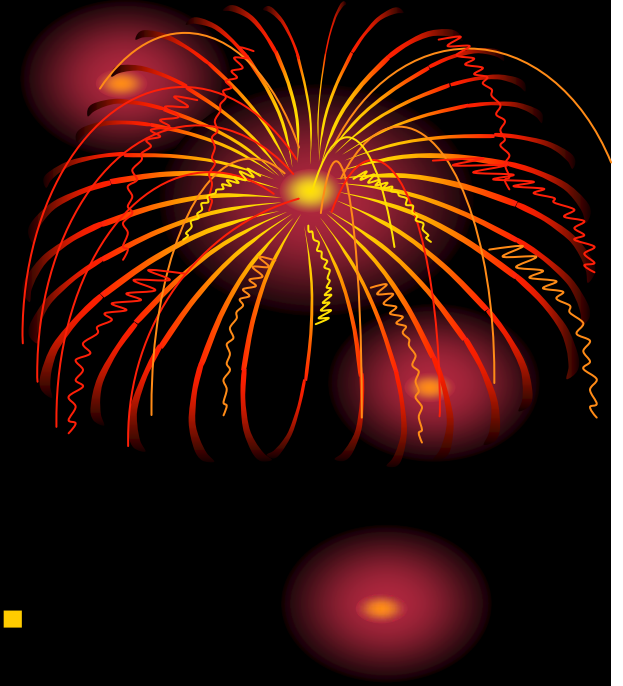


Graphical User Interfaces. Swing

Jose Jesus García Rueda

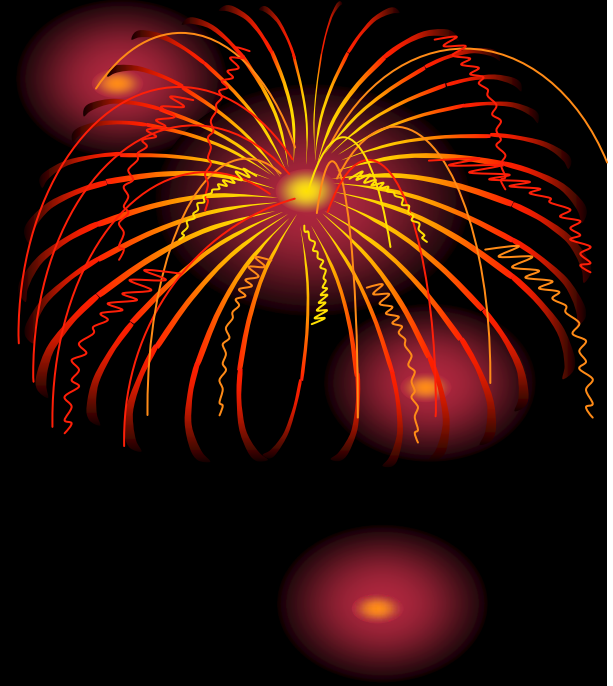
Introduction

- **What are the GUIs?**
- **Well known examples...**

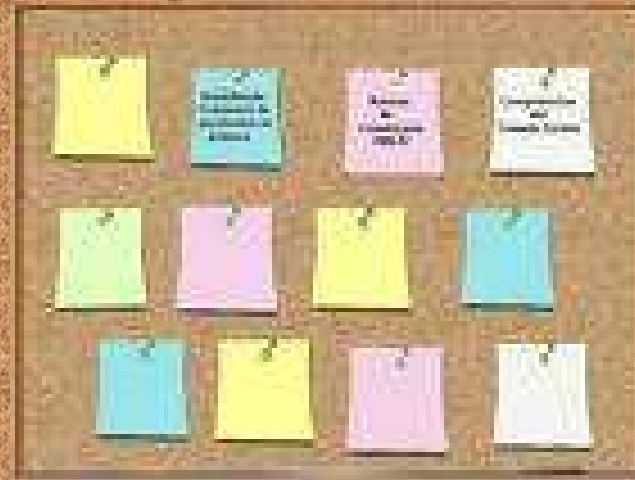


Basic concepts

- **Graphical application.**
- **Containers.**
- **Actions.**
- **Events.**
- **Graphical elements:**
 - **Menu bar.**
 - **Title bar.**
 - **Minimize and maximize buttons.**
 - **Closing button.**
 - **Scroll.**
 - **Window frame.**
 - **Icons.**
 - **Buttons.**
 - **Text areas...**



The wall metaphore



The wall metaphor

Frame

contentPane

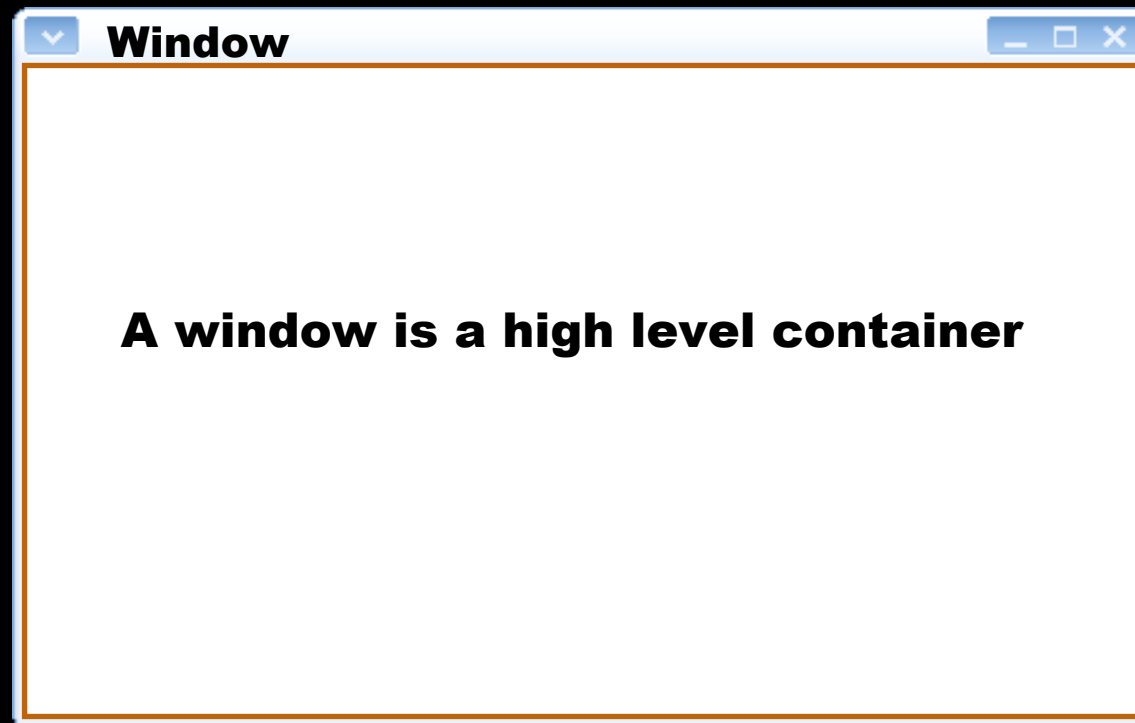
Panel

softpedia

¡Hola!
Tu opinión cuenta también.
Denuncia los abusos e injusticias.
Que se sientan incómodos.
Seamos valientes porlavoces.
Lucha por otra realidad.
Denuncia ahora!

Creating the wall

- **How to create a window in Java?**



How to create a window in Java?

```
import javax.swing.*;
```

```
public class Example extends JFrame {
```

The classes needed to build GUIs are included in the Swing package

```
    methods starts everything*/  
    public static void main (String argv[]) {
```

```
        Example window= new Example();
```

```
        window.setSize (400, 400);
```

```
        window.setVisible(true);
```

```
    }
```

```
}
```

A window in Java is just a class extending JFrame, the generic window.

The window must be made visible explicitly

Covering the wall with cork



- **Every high level container in Swing (windows, for instance) will be “covered” with a “contentPane”.**
- **The rest of the graphical components will be place on it.**
 - **Including other containers.**



Adding it to the code



```
import javax.swing.* ;
```

```
public class Example extends JFrame {
```

```
/* This methods starts everything*/
```

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

```
    Example window = new Example();
```

```
    window.getContentPane().add(...);
```

```
    window.setSize (400, 400);
```

```
    window.setVisible(true);
```

```
}
```

```
}
```

What elements can I “attach to the cork”?



- **In the contentPane you can put elements from the Swing package:**
 - **Labels: JLabel**
 - **Buttons: JButton**
 - **Text boxes: JTextField, JTextArea**
 - **Checkboxes: JCheckBox**
 - **Option buttons: JRadioButton**
 - **Lists: JList**
 - **Scroll bars: JScrollBar**
- **All the Swing components extend Jcomponent.**

And how can I attach them?



```
 JButton button;  
 JLabel label;
```

```
 public Example() {
```

```
     label = new JLabel("A label");  
     button = new JButton("A button");  
     button.setSize(100, 70);  
     getContentPane().add(button);  
     getContentPane().add(label);
```

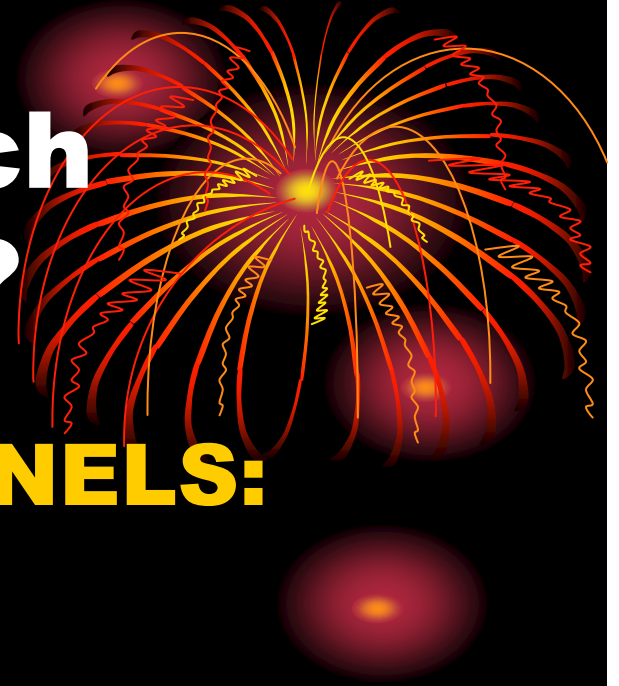
```
 }
```

LITTLE PAUSE



- **(A good time to take a look at the Java API, in order to get to know where to find information on the different graphical components and how to use them ;-)**

And how can I attach “corks to the cork”?



- **We'll use CONTENT PANELS:
Jpanel**
- **They are medium level containers:**
 - **They simplify the window organization.**
- **A panel may contain other panels.**

Panels hierarchy

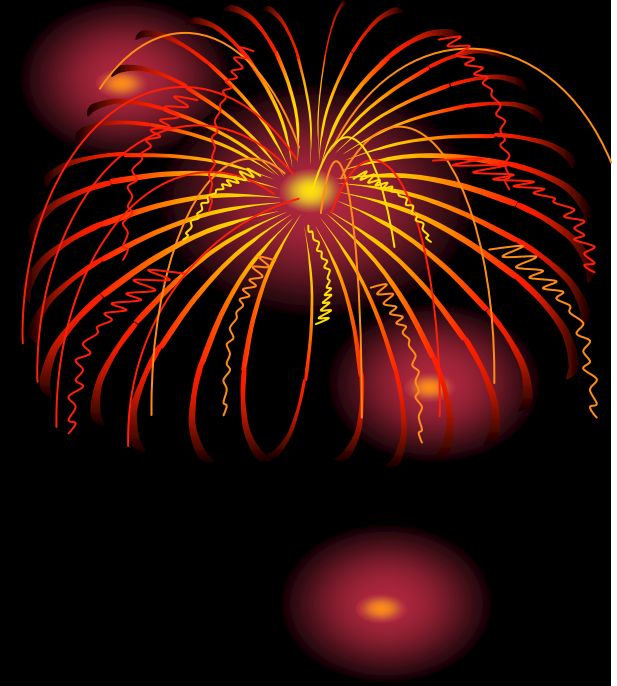


B *I* U Preview Submit

softpedia



Example of panel



```
 JButton button;  
 JLabel label;  
 JPanel panel;
```

```
 public Example() {
```

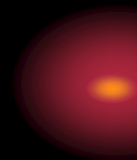
```
     panel = new JPanel();  
     getContentPane().add(panel);
```

```
     label = new JLabel("A label");  
     button = new JButton("A button");  
     button.setSize(100, 70);  
     panel.add(button);  
     panel.add(label);
```

```
 }
```

How can I put together all those components?



- **You can use either coordinates...**
 - `label.setBounds(100, 70, 50, 50);`
 - **...or LAYOUTS:**
 - These are like templates to organize graphical components.
 - They are associated to panels.
 - We'll see three types here.
 - **To use coordinates you have to neutralize the layout first:**
 - `panel.setLayout(null)`
 - **You need to import `java.awt.*` in order to use layouts!**
- 

Example using FlowLayout

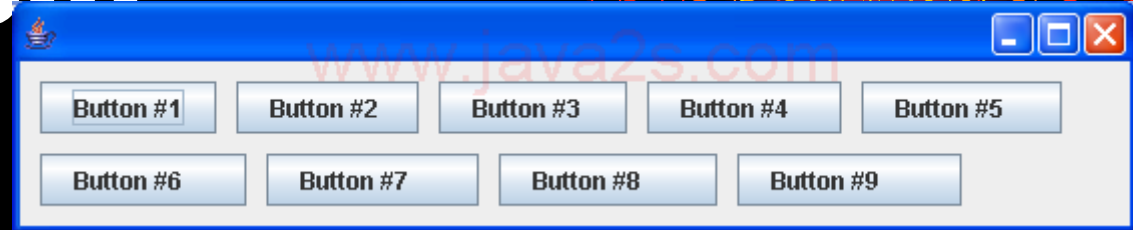
```
 JButton button;  
 JLabel label;  
 JButton otherButton;  
 JPanel panel;
```

```
public Example() {
```

```
    panel = new JPanel();  
    getContentPane().add(panel);
```

```
    label = new JLabel("A label");  
    button = new JButton("A button");  
    otherButton = new JButton("Other button");  
    panel.add(button);  
    panel.add(label);  
    panel.add(otherButton);
```

```
}
```



It places the elements in a row, one after the other.

**FlowLayout is the default!
(In panels)**

Example using GridLayout

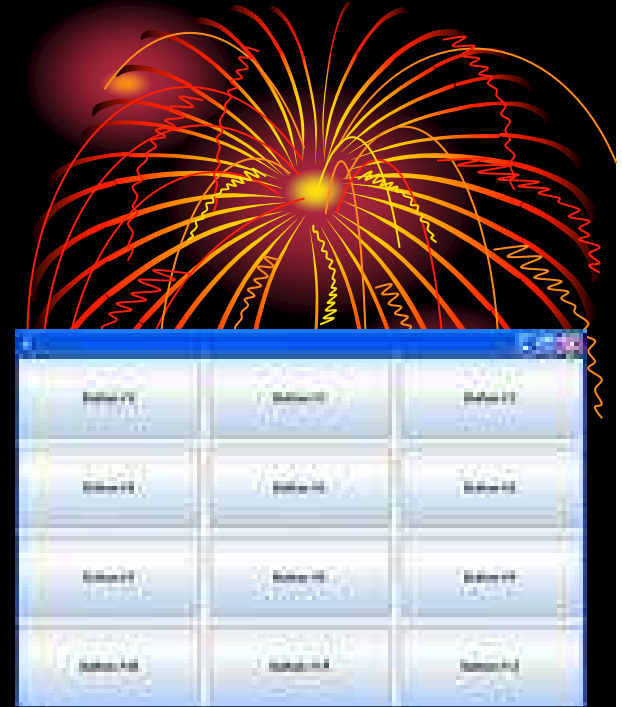
```
public Example() {
```

```
    panel = new JPanel();  
    panel.setLayout(new GridLayout(2, 2));  
    getContentPane().add(panel);
```

```
    label = new JLabel("A label");  
    button = new JButton("A button");  
    otherButton = new JButton("Other button");  
    panel.add(button);  
    panel.add(label);  
    panel.add(otherButton);
```

```
}
```

It places the elements
in a grid.



Example with BorderLayout

It is the default in high level containers.

```
public Example() {
```

```
    panel = new JPanel();  
    panel.setLayout(new BorderLayout());  
    getContentPane().add(panel);
```

```
    label = new JLabel("A label");  
    button = new JButton("A button");  
    otherButton = new JButton("Other button");  
    panel.add(button, BorderLayout.SOUTH);  
    panel.add(label, BorderLayout.WEST);  
    panel.add(otrobutton, BorderLayout.NORTH);
```

```
}
```

It divides the container in five sections:
North, south, east, west and center.

