Computer Science Introductory Course MSc - Software Engineering

Lecture 6: GUI design with SWING

Pablo Oliveira <pablo@sifflez.org>

ENST

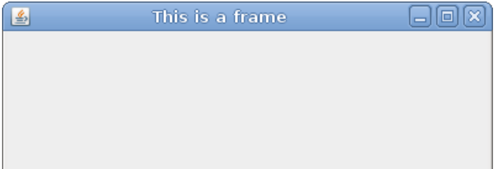
Outline

1. JFrame
2. Components
   - Adding components
   - Components Hierarchy
3. Layouts
4. Event listeners
   - Clicking on a button
   - Listeners
5. Drawing
6. Making your own components

Top-level container : JFrame

```
import javax.swing.*;
import java.awt.*;

class Test {
  public static void main(String[] args) {
    JFrame frame = new JFrame("This is a frame");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setPreferredSize(new Dimension(400, 200));
    frame.pack();
    frame.setVisible(true);
  }
}
```

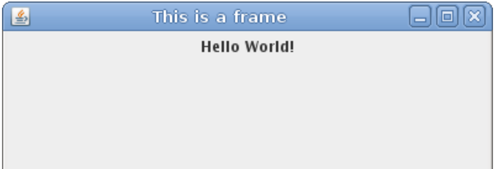


Adding components to our frame

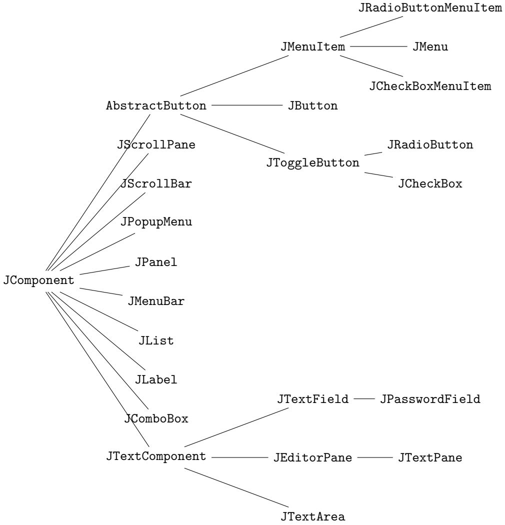
```
public static void main(String[] args) {
  JFrame frame = new JFrame("This is a frame");
  frame.setPreferredSize(new Dimension(400, 200));
  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

  JPanel panel = new JPanel();
  frame.setContentPane(panel);
  panel.add(new JLabel("Hello World!"));

  frame.pack();
  rframe.setVisible();
}
```

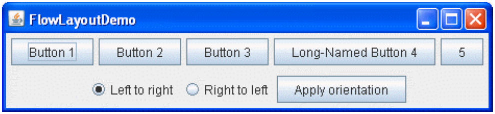


Components Hierarchy

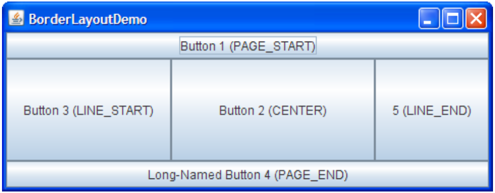


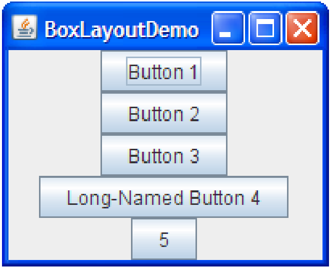
JPanel and layouts

- `JPanel` are containers that group and arrange other components.
- We add a component to a `JPanel` with the `.add(component)` method.
- Components inside a `JPanel` are placed according to its layout.
- Layouts implement the API interface `LayoutManager`.
- We choose a JPanel's layout in its constructor `new JPanel(new FlowLayout())`.



Some examples of layouts



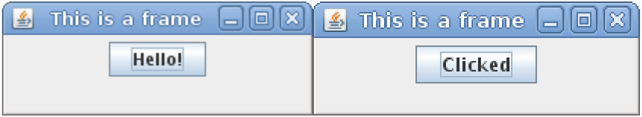


More complex layouts

- There are more complex layouts available, see : `http://java.sun.com/docs/books/tutorial/uiswing/layout/`
- Using hierarchies of layouts, you can place your components very precisely.

Clicking on a button

```
JButton button = new JButton("Hello!");
panel.add(button);
```



How to react to an action from the user ?

Listener interface

```
import java.awt.event.*;
class ButtonListener implements ActionListener {
  JButton button;
  public ButtonListener(JButton button){
    this.button = button;
  }
  public void actionPerformed(ActionEvent e) {
    button.setLabel("Clicked");
  }
}

button.addActionListener(new ButtonListener(button));
```

Anonymous classes and listeners

```
button.addActionListener(
  new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      button.setLabel("Clicked");
    }
  });
```

More on listeners

- More details on listeners at :
  `http ://java.sun.com/docs/books/tutorial/uiswing/events/`

Drawing

- Override the JComponent's method `void paintComponent(Graphics g)`.
- This method is called each time the component must be redrawn.
- The `Graphics` object lets you draw inside the Component.
- For a JFrame you can override the paint method.
- See example !

Making your own components

- As any other java class, `JComponent` can be extended.
- This can be useful in many cases :
  - Factorizing a component and its listeners in the same class.
  - Changing the look of a component.
  - Adding functionalities to a component.