

myPhotoshop

Adrien Toubiana Nils Layet Tania Situm



April 18, 2016

Copyright

This document is for internal use only at EPITA <<http://www.epita.fr>>.

Copyright © 2015/2016 Assistants <yaka@yaka.epita.fr>.

Copying is allowed only under these conditions:

- You must have downloaded your copy from the Assistants' Intranet <<https://www.acu.epita.fr>>.
- You must make sure you have the latest version of this document.
- It is your responsibility to make sure that this document stays out of reach of students or individuals outside your year (your "promotion").



① Swing it, baby

② Provided code

③ Java concepts



① Swing it, baby

- GUI
- Look and Feel

② Provided code

③ Java concepts



Components

- **Every item** in your window is a component.
- Examples: buttons, menus, images ...
- Your components are in a panel, managed by a layout manager



Frames

- A frame is a window
- No component in the window
- Only contains panels
- In Swing, you have the `JFrame` class



Panels

- A panel contains components and a layout manager
- Included in a frame
- In Swing, you have the `JPanel` class



Layouts

- A layout manager handles the positioning of the different components in a panel
- Different layout managers exist, depending on what you want to do
- Next slides present the main layout managers and examples on how they work

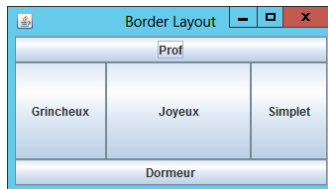


BorderLayout

- Use borders to put your components where you want
- Up to five areas
 - top
 - bottom
 - left
 - right
 - center



BorderLayout – Example



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

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

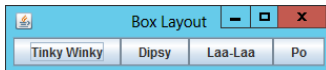
```
5 panel.add(new JButton("Prof"), BorderLayout.NORTH);  
panel.add(new JButton("Simplet"), BorderLayout.EAST);  
panel.add(new JButton("Dormeur"), BorderLayout.SOUTH);  
panel.add(new JButton("Grincheux"), BorderLayout.WEST);  
panel.add(new JButton("Joyeux"), BorderLayout.CENTER);
```

BoxLayout

- Puts components in a single row or column
- Respects the components' requested maximum size
- It is possible to align components



BoxLayout – Example



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

   panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));

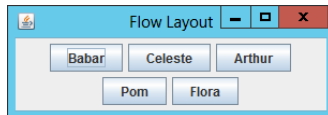
   panel.add(new JButton("Tinky_Winky"));
6  panel.add(new JButton("Dipsy"));
   panel.add(new JButton("Laa-Laa"));
   panel.add(new JButton("Po"));
```

FlowLayout

- Default layout manager for every `JPanel`
- Lays out components in a single row, creating another one if the panel is not wide enough
- Easy to use



FlowLayout – Example



```
JPanel panel = new JPanel();

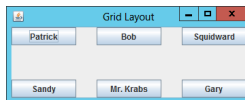
panel.add(new JButton("Babar"));
panel.add(new JButton("Celeste"));
5 panel.add(new JButton("Arthur"));
panel.add(new JButton("Pom"));
panel.add(new JButton("Flora"));
```

GridLayout

- Aligns components in a grid of cells
- All the cells are equal
- Easier to use



GridLayout – Example



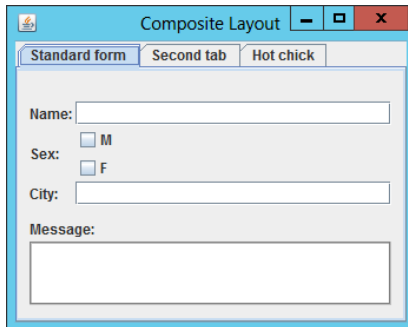
```
JPanel panel = new JPanel();
```

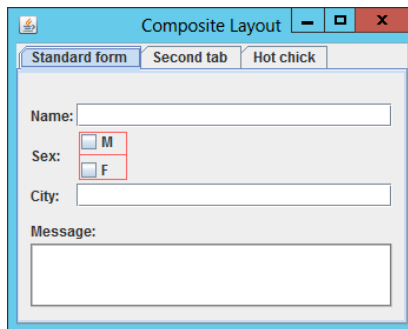
```
3  GridLayout layout = new GridLayout(2, 3);  
    panel.setLayout(layout);
```

```
    layout.setHgap(30);  
    layout.setVgap(50);
```

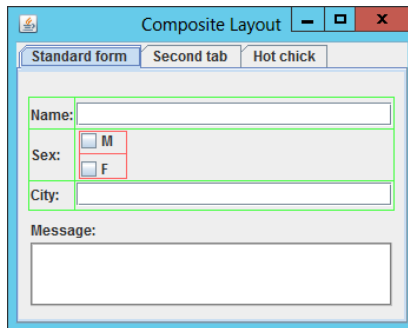
```
8  
    panel.add(new JButton("Patrick"));  
    panel.add(new JButton("Bob"));  
    panel.add(new JButton("Squidward"));  
    panel.add(new JButton("Sandy"));  
13  panel.add(new JButton("Mr. Krabs"));  
    panel.add(new JButton("Gary"));
```

Composite layout

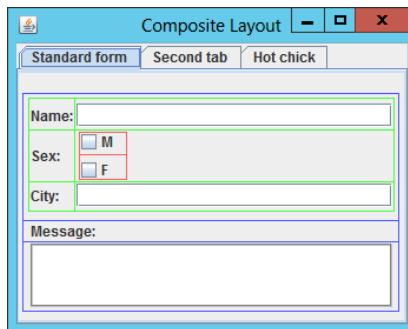




- BorderLayout



- BorderLayout
- GridLayout



- BorderLayout
- GridLayout
- BorderLayout

When should you use any layout manager?

- Different scenarios
- Try to think of ***what you want to do*** and not *what your friends use*
- Our solution is just a piece of advice: you can use what you want



Look and Feel

Look and Feel

- Appearance of the application
- Look is represented by fonts, colors, shapes, layouts ...
- Components included in your application
- Swing implements Look and Feel



① Swing it, baby

② Provided code

■ Code explanations

③ Java concepts



Code explanations

BufferedImage

- Included in `java.awt`
- Included in `ImagePanel`
- Use it for image handling
- Not serializable



ImagePanel

- Extends `JPanel`
- Contains a `BufferedImage`
- Serializable panel
- Useful for saving your project
- Only an helping tool
- It does not necessarily suit your needs



Filter

Methods to implement

- `getName()`: Returns the filter's name.
- `perform(BufferedImage img)`: Modifies the image, executing your filter.

In general

- Every filter **must** implement this interface.
- Filters can also implement other interfaces (if needed).



① Swing it, baby

② Provided code

③ Java concepts

- Serialization
- Threads



Serialization

Serialization

In general

Serialization saves your objects, so that you can load them and stream them.

In MyPhotoshop

- Extension (*.myPSD)
- Use your own implementation for history



Threads



Threads

- Usually you should inherit from **Thread** or implement **Runnable**.
- In the project, you are **strongly advised** to extend **SwingWorker** to implement asynchronous filters.
- Override the methods you want.



