Java Swing

Daniel Porumbel
dp.cnam@gmail.com

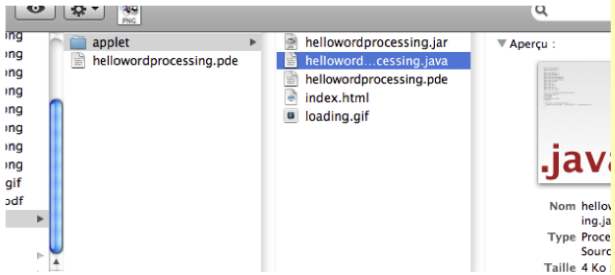
nombreaux slides dus à
Pierre Cubaud

Rappel

Processing est du Java

1) exporter le code

2) un dossier est créé avec un .java




```
import processing.core.*;
import processing.xml.*;

import java.applet.*;
import java.awt.Dimension;
import java.awt.Frame;
import java.awt.event.MouseEvent;
import java.awt.event.KeyEvent;
import java.awt.event.FocusEvent;
import java.awt.Image;
import java.io.*;
import java.net.*;
import java.text.*;
import java.util.*;
import java.util.zip.*;
import java.util.regex.*;

public class hellowordprocessing extends PApplet {
  public void setup() {
size(300,100);
textSize(50);
text("Hello Word!",0,50);
  noLoop();
}
  static public void main(String args[]) {
    PApplet.main(new String[] { "--bgcolor=#FFFFFF", "h
  }
}
```

Rappel

"Hello World" en Java

1) écrire dans un fichier texte appelé **HelloWorld.java** :

```
// ceci est mon premier programme Java
// P. Cubaud oct14

class HelloWorld {
 public static void main(String[] arg)  {
  System.out.println("Hello NFP135 !");
 }
}
```

même nom et
respect de la casse

2) lancer le compilateur javac : produit le .class
3) lancer l'interpréteur java avec le .class (sans l'extension)

```
fibonacci2:MANIPS JAVA pcubaud$
fibonacci2:MANIPS JAVA pcubaud$ javac HelloWorld.java
fibonacci2:MANIPS JAVA pcubaud$ java HelloWorld
Hello NFP135 !
fibonacci2:MANIPS JAVA pcubaud$
```

Reprise d'un ancien code : inversion du contenu d'un tableau (exo5 de la séance de révision)

exo5

```
int[] t = {1,2,4,4,5,6,8,7};

println(t);

int i=0;
int j=t.length-1;
int p;
while (i<j){
  p = t[i];
  t[i] = t[j];
  t[j] = p;
  i++;
  j--;
}

println(t);
```

le code exo5.pde

```
public class exo5 {
  static public void main(String args[]) {
    int[] t = {1,2,4,4,5,6,8,7};
    System.out.println(t);
    int i=0;
    int j=t.length-1;
    int p;
    while (i<j){
      p = t[i];
      t[i] = t[j];
      t[j] = p;
      i++;
      j--;
    }
    System.out.println(t);
  }
}
```

le code exo5.java

Exécution :

```
fibonacci2:programmes cours 11 pcubaud$ javac exo5.java
fibonacci2:programmes cours 11 pcubaud$ java exo5
[I@19908ca1
[I@19908ca1
fibonacci2:programmes cours 11 pcubaud$
```

Que se passe-t'il ?

Comment y remédier ?

Les paquetages (package)

- un paquetage est une collection de classes utiles
pour d'autres programmes

- bibliothèque (library) : notion importante très ancienne
qui remonte à Babbage !

- support dans quasi tous les langages de programmation)
mais pas vraiment standardisé dans les langages avant Ada

- interopérabilité entre langages encore difficile => wrappers

- les librairies traduisent / amplifient le succès d'un langage

Un exemple parmi plein : le paquetage random

```
import java.util.Random;

public class RandomDemo {
    public static void main(String[] args) {
    Random rndNumbers = new Random();
        int rndNumber = 0;

        for (int nbr = 1; nbr < 9; nbr++) {
            rndNumber = rndNumbers.nextInt();
            System.out.println("Number: " + rndNumber);
        }
    }
}
```

ou
import java.util.*

sa doc en ligne :

http://docs.oracle.com/javase/7/docs/api/java/util/Random.html

Rque : créer ses propres packages => dernier cours

```
import javax.swing.*;

public class HelloWorldSwing {
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```

houla!!
multi-tâche
(à voir
plus tard)

version plus simple (mais naïve)

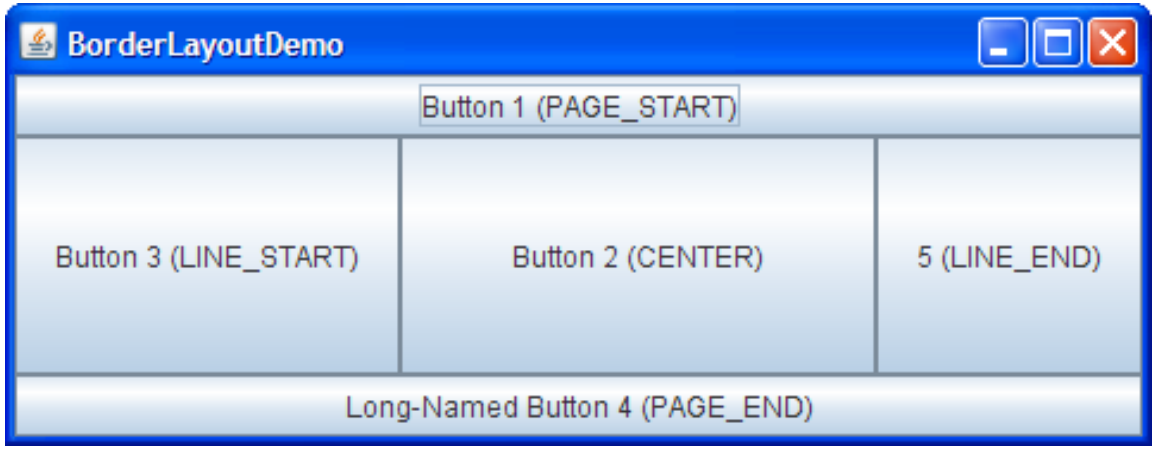
```
import javax.swing.*;

public class HelloWorldSwing2 {
    public static void main(String[] args) {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.add(label);

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }
}
```

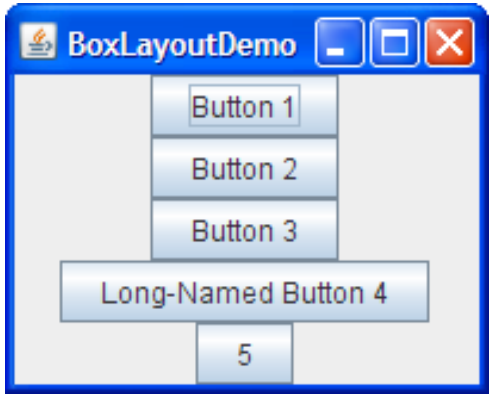
le label est rangé par défaut dans le contentPane de la frame (depuis java 1.6)

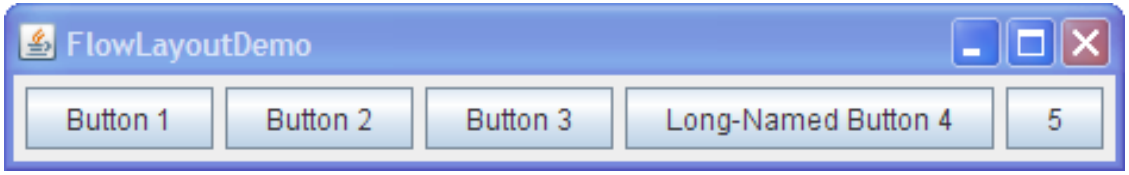
(+ autres versions avec héritage, à voir plus tard)

Organisation des écrans (layout)

Pour éviter de calculer des coordonnées exactes
pour les composants, on utilise des organisations "automatiques"

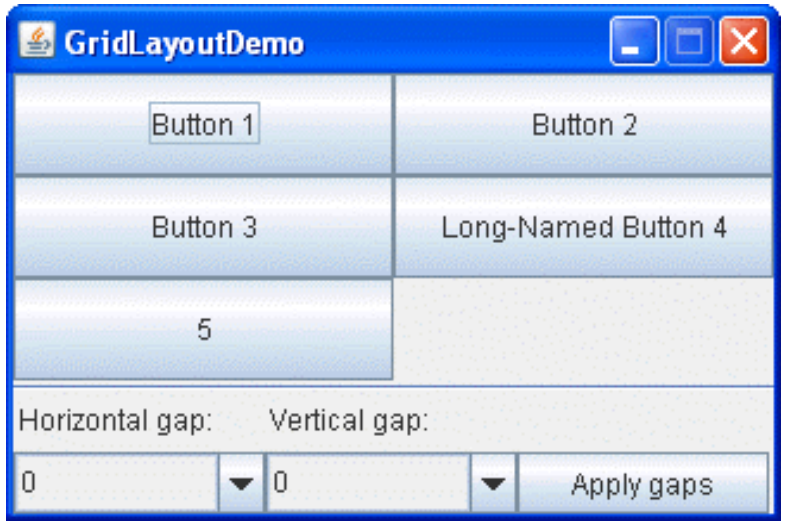


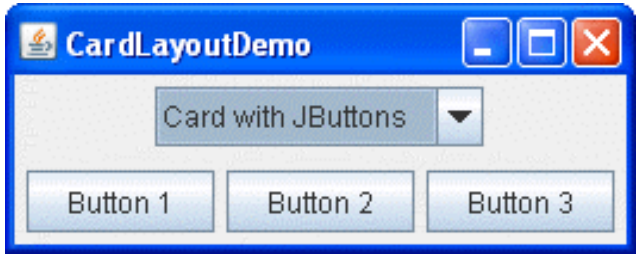




http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

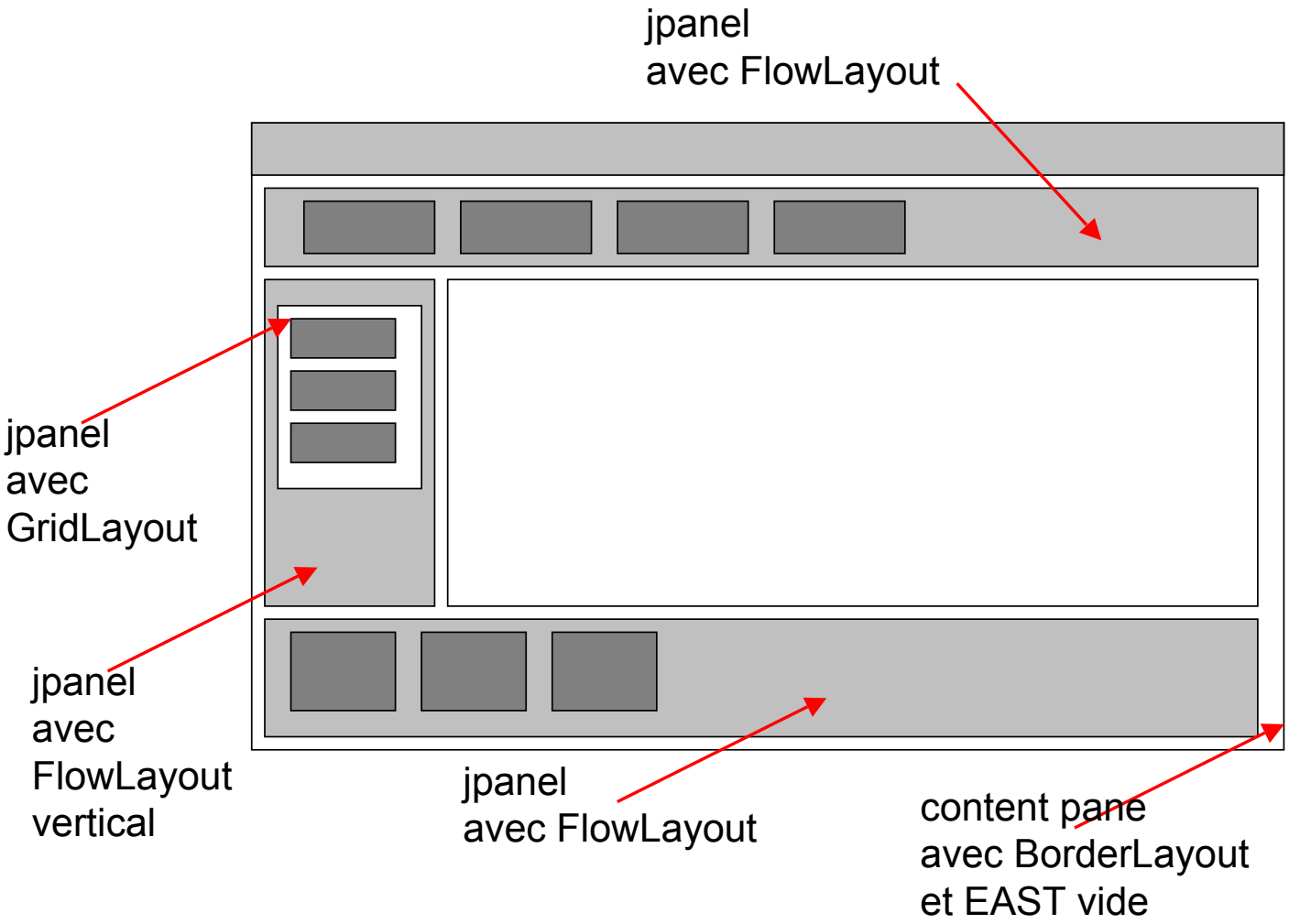
autres exemples





http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

Imbrication des containers (avec des layouts adaptés)



Gestion des évènements utilisateur

- chaque action de l'utilisateur se traduit par un *event*. Un composant peut en être notifié (*listener*) et en propager

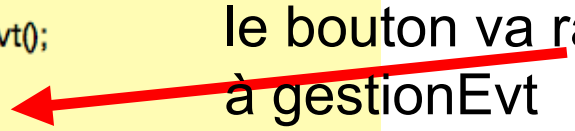
- à chaque composant, on associe une *action* pour rendre ce composant utile à quelque chose

- il faut gérer la correspondance entre les events et les actions (très vieux problème des interfaces graphiques)

- avec Swing, il y a de très (trop ?) nombreuses manières de procéder, selon la complexité des interactions dans l'application

Exemple avec des boutons

```
public class DemoEvt {
   public static void main(String[] args) {
      JFrame frame = new JFrame("demo clic bouton");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      Container  cpane = frame.getContentPane();
      cpane.setLayout(new BoxLayout(cpane, BoxLayout.Y_AXIS));

      JButton b1 = new JButton("titi");
      b1.setAlignmentX(Component.CENTER_ALIGNMENT);
      cpane.add(b1);
      JButton b2 = new JButton("toto");
      b2.setAlignmentX(Component.CENTER_ALIGNMENT);
      cpane.add(b2);

     GestionEvt gestionEvt = new GestionEvt();
     b1.setActionCommand("clic de titi");
     b1.addActionListener(gestionEvt);
     b2.setActionCommand("clic de toto");
     b2.addActionListener(gestionEvt);

       //Display the window.
       frame.pack();
       frame.setVisible(true);
   }
}
```



Avec au début du code

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

class GestionEvt implements ActionListener {
 public void actionPerformed(ActionEvent e){
  System.out.println(e.getActionCommand());
 }
}
```

ici l'action

Execution :

```
fibonacci2:programmes cours 11 pcubaud$ javac DemoEvt.java
fibonacci2:programmes cours 11 pcubaud$ java DemoEvt
clic de titi
clic de toto
clic de titi
clic de titi
clic de titi
clic de toto
```

Lecture Terminal

```
class Lecture{
    public static void main(String[] args){
        java.util.Scanner scnr =
            new java.util.Scanner( System.in );
        int unEntier;
        unEntier = scnr.nextInt();
        System.out.println(unEntier);
    }
}
```