SIMPLON.CO

Formation POE Java
Journée #2: Part 2 – Programmation événementielle & interfaces graphiques (GUI) Java Swing

Athanasia Katsouraki

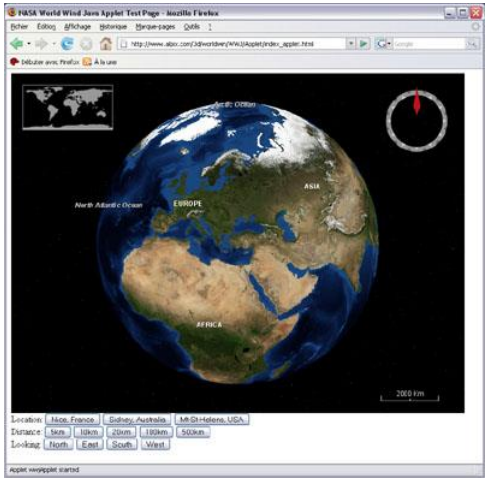
17.10.2017

Sortes de programmes avec Java

On peut faire :

1. des applications, sous forme de fenêtre ou de console ;
2. des applets, qui sont des programmes Java incorporés à des pages web ;
3. des applications pour appareils mobiles, avec J2ME ;
4. et bien d'autres ! J2EE, JMF, J3D pour la 3D...

Exemple d'application Java sous forme de fenêtre



Exemple d'applet Java

Exemple d'application mobile Java

Qu’est-ce qu’une interface graphique?

Une interface graphique (en anglais ***GUI*** pour *graphical user interface*) ou un **environnement graphique**:

- ✓ un dispositif de dialogue homme-machine, dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, de sorte que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage (ex. une souris)

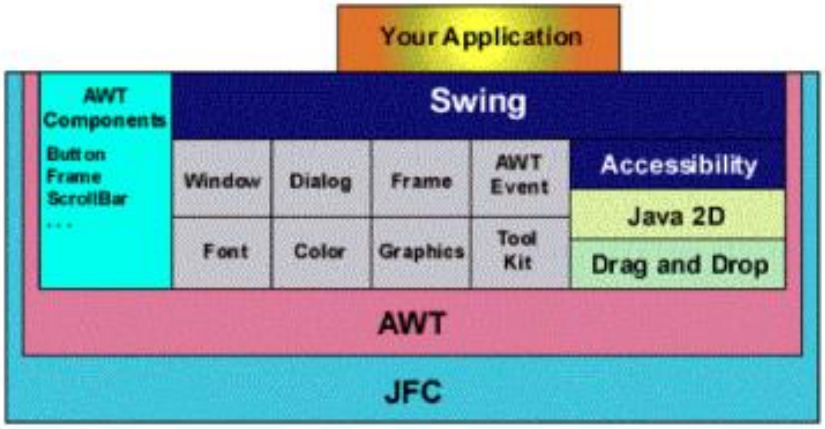
Toolkits graphiques Java

Il y en a trois !

- AWT **Components**, obsolète
- Swing supporté par Oracle (autrefois Sun)
- SWT libre, initié par IBM / Eclipse
- tous (+ ou -) multi-plateformes

Swing repose sur AWT

- mais Swing != AWT !
- **JButton != Button !**



JFC comprend :
-les composants **Swing**: bibliothèque graphique pour le langage de programmation Java
-Java **2D**: Utilisation de classes Graphics 2D amenant des manipulations complexes de la couleur, la manipulation simple des transformation affines (rotation), traitement des textures
-**L'accessibilité** : la manipulation simple des ordinateurs pour les personnes handicapés moteurs
-Le « **drag and drop** » : glisser-déposer entre application quelconque (pas forcément Java) sur une plate-forme.

AWT = Abstract Window Toolkit (AWT) : outils de fenêtrage dépendant de la plate-forme de Java , graphiques, toolkit widget d'interface utilisateur.
JFC = Java Foundation Classes (JFC) : set de composants et de services qui simplifient le développement et le déploiement d'applicationscommerciales-bureautiques et Internet / Intranet.

Swing - Packages

The Swing API has 18 public packages:

javax.accessibility
javax.swing
javax.swing.border
javax.swing.colorchooser
javax.swing.event
javax.swing.filechooser
javax.swing.plaf
javax.swing.plaf.basic
javax.swing.plaf.metal
javax.swing.plaf.multi
javax.swing.plaf.synth
javax.swing.table
javax.swing.text
javax.swing.text.html
javax.swing.text.html.parser
javax.swing.text.rtf
javax.swing.tree
javax.swing.undo

Swing contient outils GUI avancée. Widgets de base : boutons, étiquettes, les arbres et les tables. Swing est écrit en Java.

Swing fait partie de JFC: Java Foundation Classes. C'est une collection de paquets pour créer des applications bureautiques complètes.

JFC comprend AWT, Swing, Accessibilité, Java 2D et Drag and Drop.

Swing a été publié en 1997 avec JDK 1.2. C'est une boîte à outils mature.

First Window…

SimpleEx.java

```
package com.zetcode;

import java.awt.EventQueue;
import javax.swing.JFrame;

public class SimpleEx extends JFrame {

    public SimpleEx() {

        initUI();
    }

    private void initUI() {

        setTitle("Simple example");
        setSize(300, 200);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {

        EventQueue.invokeLater(() -> {
            SimpleEx ex = new SimpleEx();
            ex.setVisible(true);
        });
    }
}
```

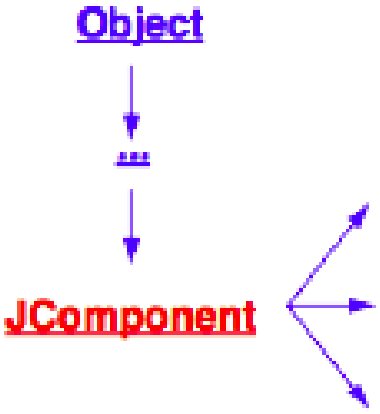
classes

La classe SimpleEx hérite du JFrame. JFrame est un conteneur. Le but fondamental du conteneur est de contenir les composants de l'application.

set the title/size /location of the window using the setTitle() method.

Close the window

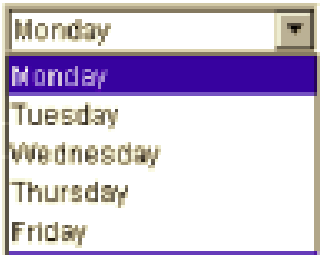
Interacteurs



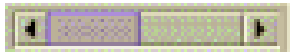
- AbstractButton
- JLabel
- JSeparator
- JList
- JComboBox
- JScrollBar
- JSlider
- JToolTip
- JTextComponent
- *Conteneurs ...*

JLabel

JList



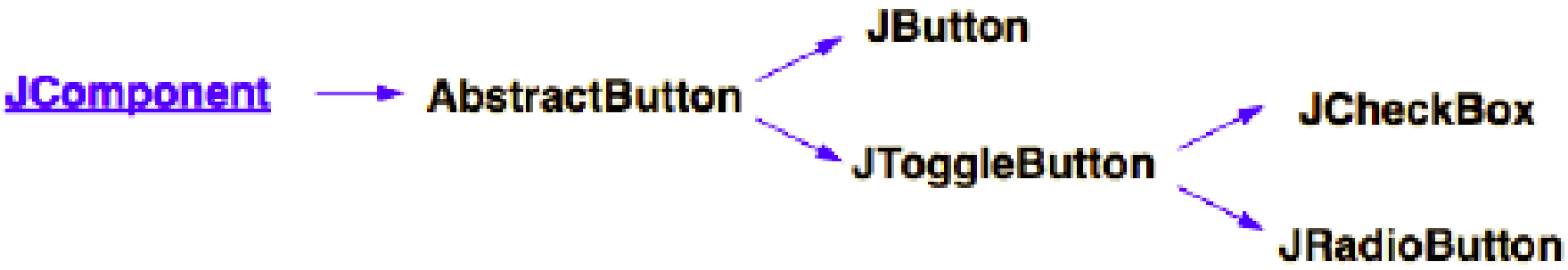
JComboBox

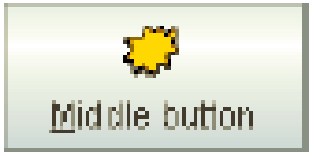


JScrollBar

JSlider

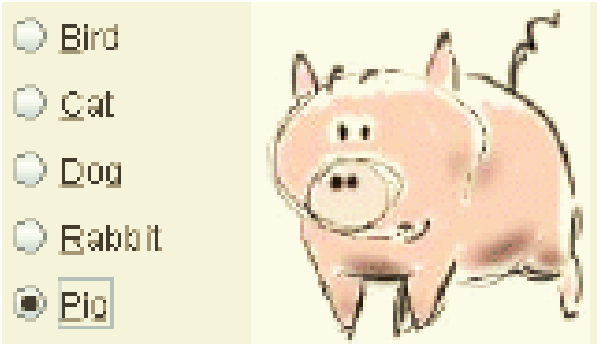
Boutons





JButton

JCheckbox :
choix idépendants

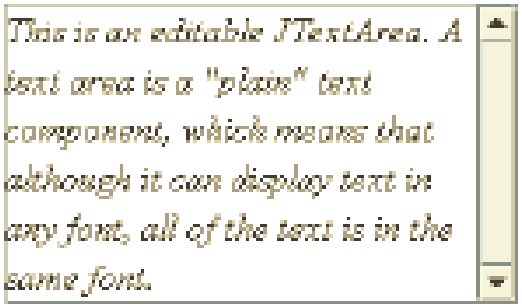


JRadioButton :
choix exclusif : cf. ButtonGroup

Texte

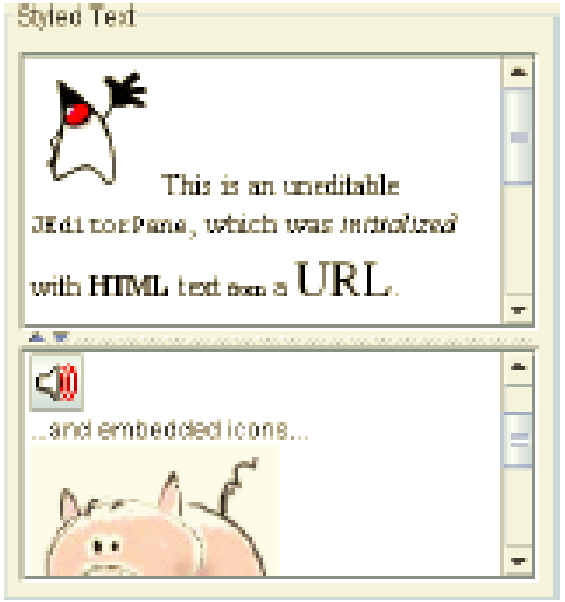
JTextField

JPasswordField



JTextArea :
texte simple multilignes

Ascenseur :
cf. JScrollPane



JEditorPane : texte avec styles
compatible HTML et RTF

Conteneurs

JComponent

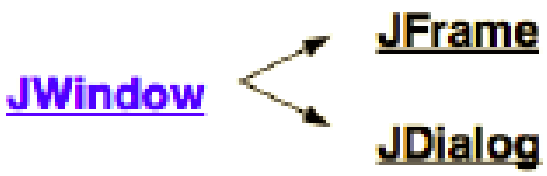
- JPanel
- JViewport
- JScrollPane
- JSplitPane
- JTabbedPane
- ...

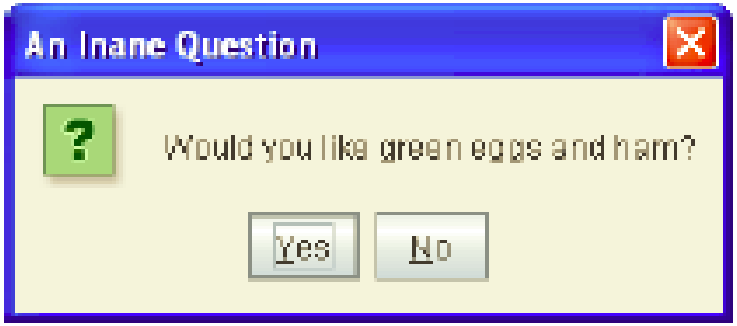
JPanel: conteneur générique

JScrollPane:
avec ascenseurs intégrés

JSplitPane:
avec « diviseur » intégré

Fenêtres



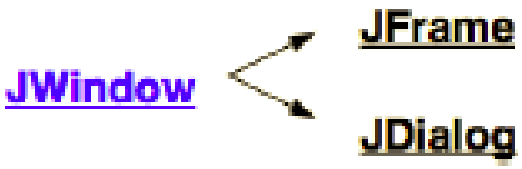


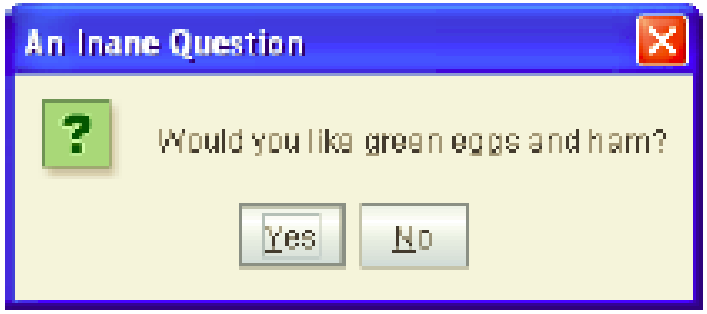
JFrame : fenêtre principale de l'application

JDialog : fenêtre secondaire

- normalement **dépendante** de la JFrame :
  - pas d'iconification séparée, toujours au dessus de la JFrame
- généralement **temporaire** et **modale** (« transiente »):
  - bloque l'interaction, impose à l'utilisateur de répondre

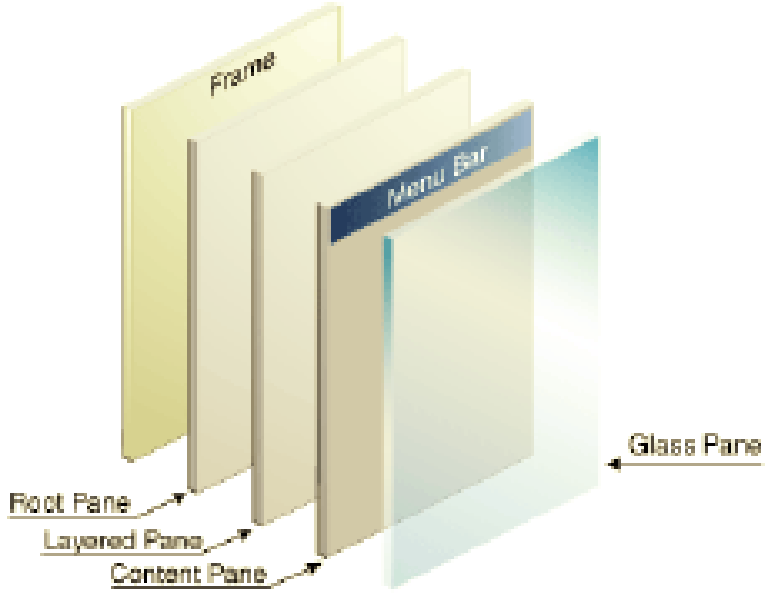
Fenêtres



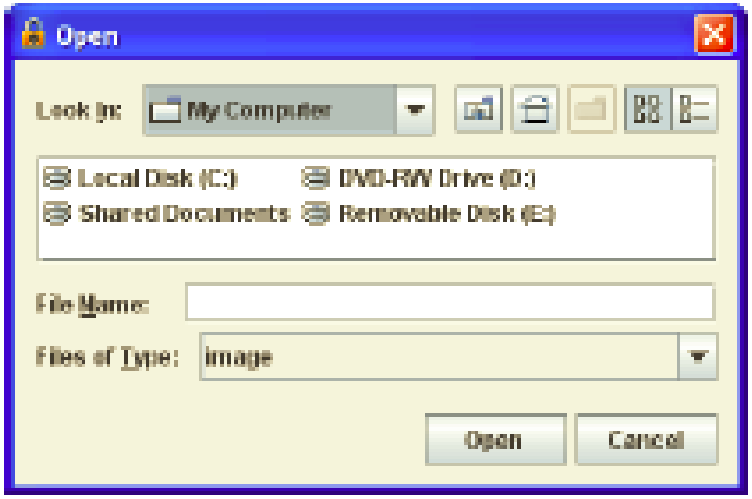


JRootPane implicitement créé

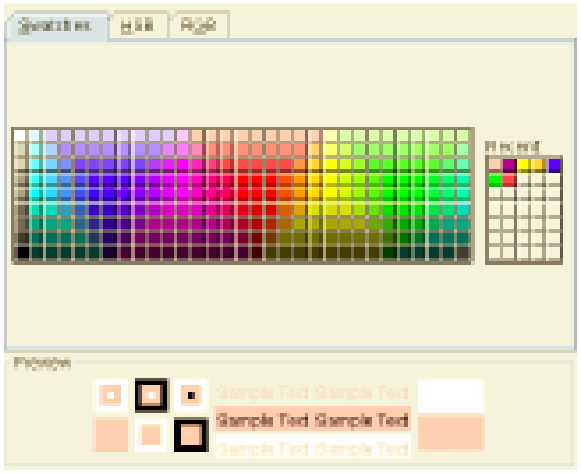
- par JApplet, JDialog, JFrame
- et JInternalFrame



Boîtes de dialogue



JFileChooser



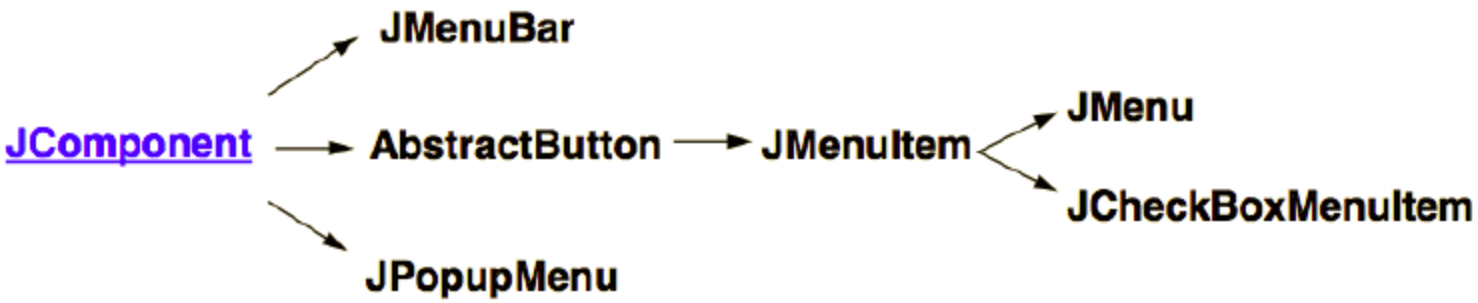
JColorChooser

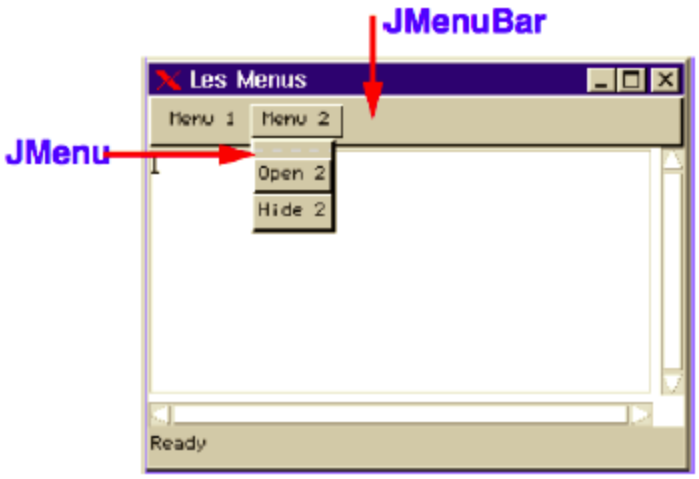
JOptionPane (multiples variantes)

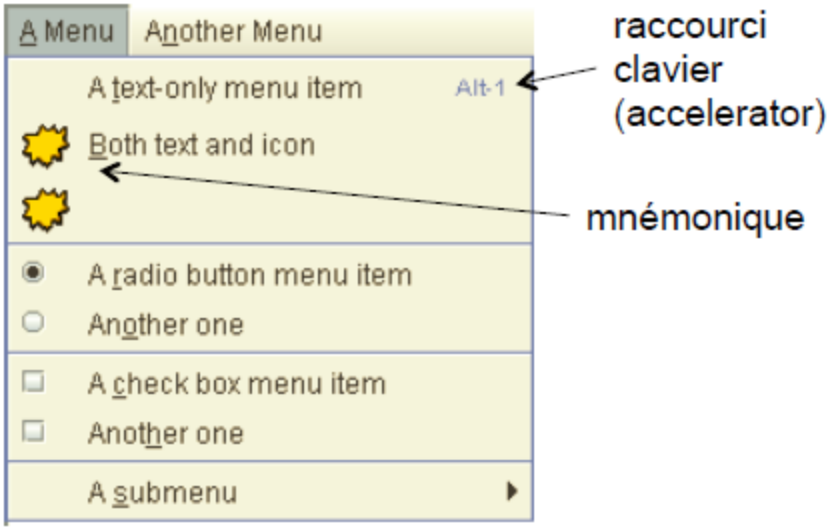
Particularité

- peuvent être créés :
  - comme composants internes
  - ou comme boîtes de dialogue

Menus


JMenuBar
JComponent
AbstractButton
JMenuItem
JMenu
JCheckBoxMenuItem
JPopupMenu


JMenuBar
JMenu
Les Menus
Menu 1
Menu 2
Open 2
Hide 2
Ready


A Menu
Another Menu
A text-only menu item
Alt-1
Both text and icon
A radio button menu item
Another one
A check box menu item
Another one
A submenu
raccourci clavier (accelerator)
mnémonique

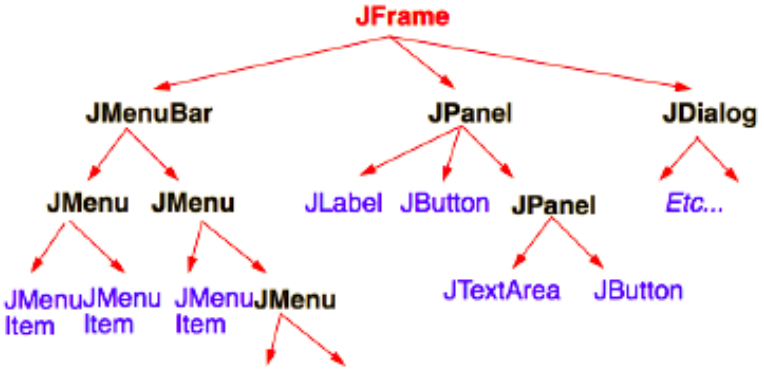
Arbre d'instanciation

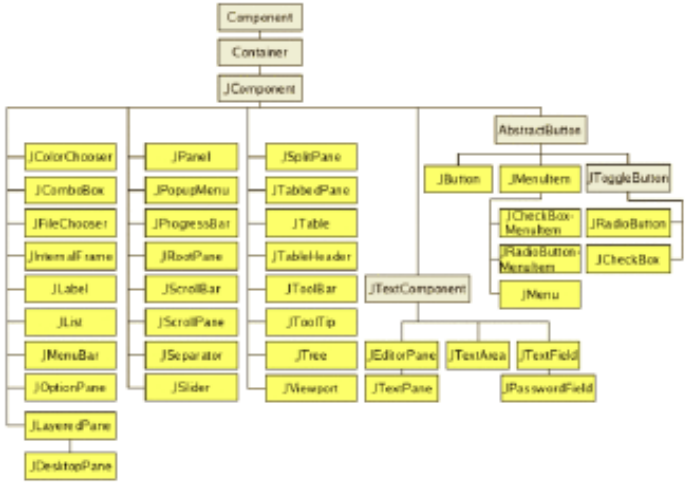
Arbre d'instanciation

- arbre de **filiation** des **instances** de composants graphiques

Attention : ne pas confondre avec l'arbre d'héritage !

- **arbre d'instanciation** = arbre de filiation des instances
- **arbre d'héritage** = hiérarchie des classes

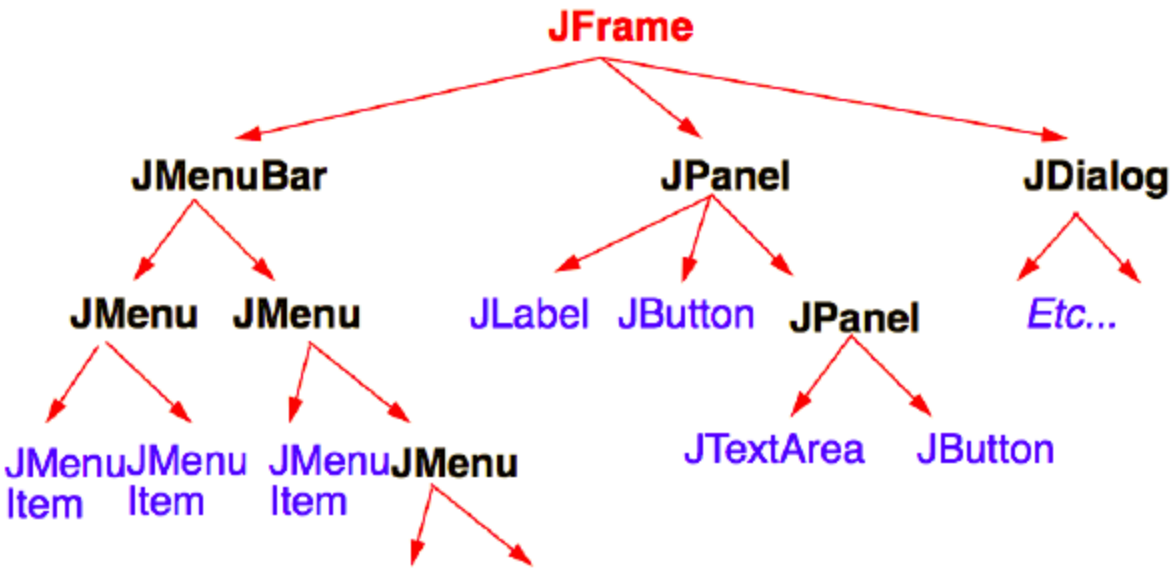


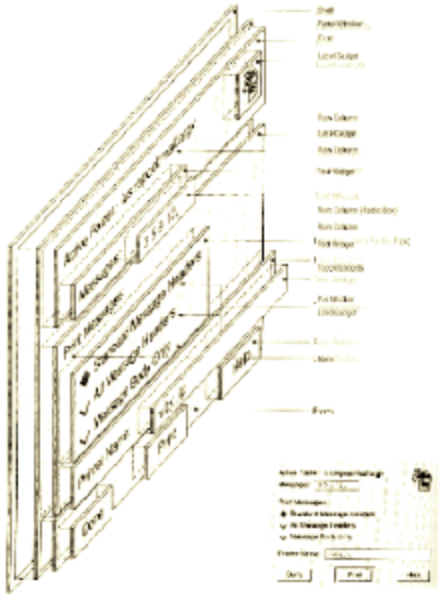


Arbre d'instanciation

Chaque objet graphique « contient » ses enfants

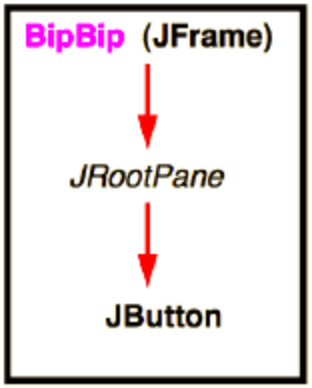
- **superposition** : enfants affichés au dessus des parents
- **clipping** : enfants « découpés » : ne dépassent pas des parents

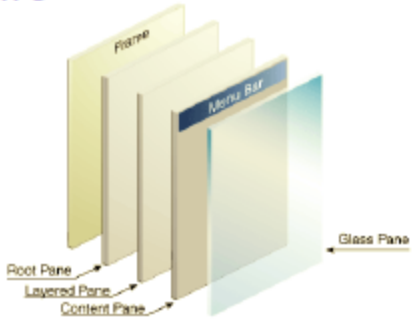




Exemple

```
// donne accès aux composants Swing
import javax.swing.*;

public class BipBip extends JFrame {        // fenêtre principale
    JButton button = null;

    public static void main(String argv[ ]) {
        BipBip toplevel = new BipBip();       // en gris : optionnel
    }

    public BipBip() {
        button = new JButton ("Please Click Me !");
        getContentPane().add(button);      // en gris : nécessaire avant version 5

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Bib Bip");
        pack();                          //  calcule la disposition spatiale
        setVisible(true);                // fait apparaître l'interface
    }
}
```





Notes et rappels

- package javax.swing
- une seule classe public par fichier, le fichier doit avoir le même nom

- button est une variable d'instance
- toplevel est une variable locale

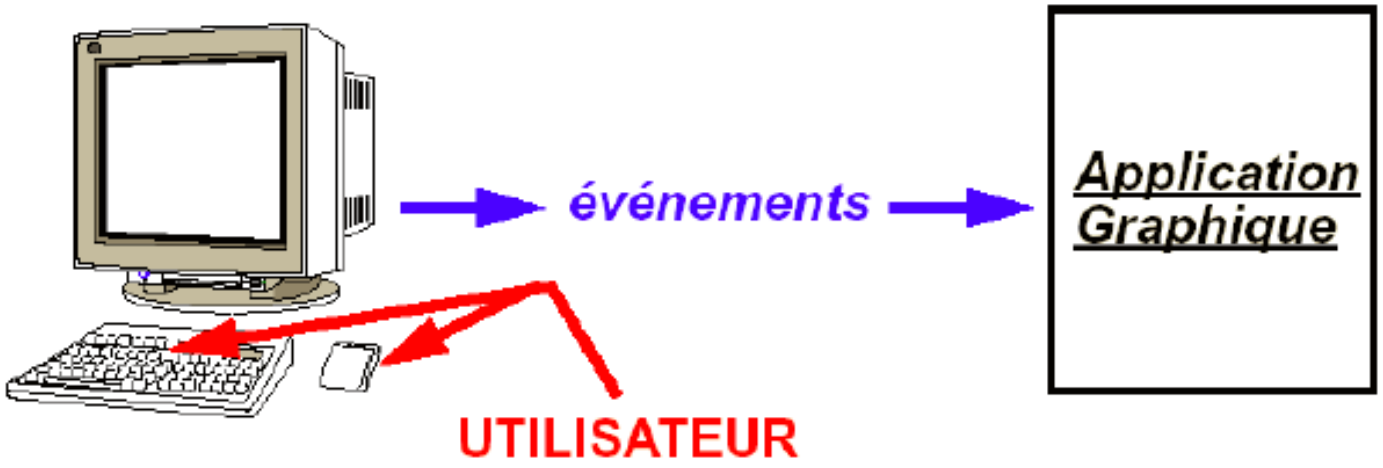
- main() est une méthode de classe (cf. static)
- les méthodes d'instance ont automatiquement accès aux variables d'instance
  elles ont un paramètre caché this qui pointe sur l'instance

- getContentPane() nécesaire avant la version 5 à cause du JRootPane
  JWindow.add() a été redéfini dans les versions ultérieure de Java

Evenements

Evénements

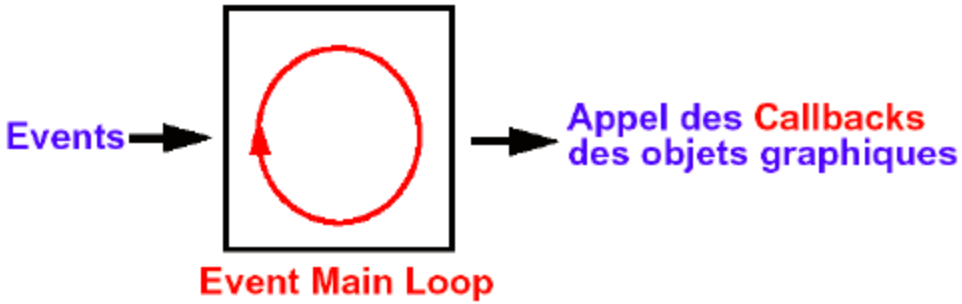
- envoyés à l'application ciblée
- à chaque action élémentaire de l'utilisateur



Boucle de gestion des evenem.

Boucle infinie qui

- récupère les événements
- appelle les **fonctions de callback** des composants graphiques



Lancée automatiquement

- à la fin de la fonction main() dans le cas de Java

Evenements

Evénements AWT et Swing

- **objets** correspondant à des catégories d'évenements
- les principaux héritent de java.awt.event.AWTEvent

Evénements de "bas niveau"

- MouseEvent — appuyer, relacher, bouger la souris ...
- KeyEvent — appuyer, relacher une touche clavier...
- WindowEvent — fermeture des fenêtres ....
- FocusEvent — focus clavier (= où vont les caractères tapés au clavier)
- etc.

Evénements de "haut niveau"

- ActionEvent — **activer** un bouton, un champ textuel ...
  abstraction des événements de bas niveau
- TextEvent — modification du texte entré
- etc.

Evenements

Méthodes communes aux AWTEvent

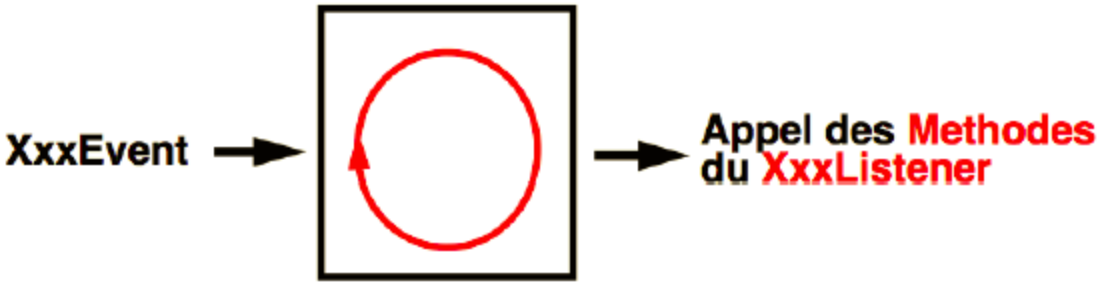
- getSource() **objet** producteur (Object)
- getID() **type** d'événement (int)

Exemple: méthodes de MouseEvent

- getX(), getY()
- getClickCount()
- getModifiers()
- getWhen()
- etc.

Event Listeners

- à chaque classe d'événement correspond une classe d'**EventListener** (en général)



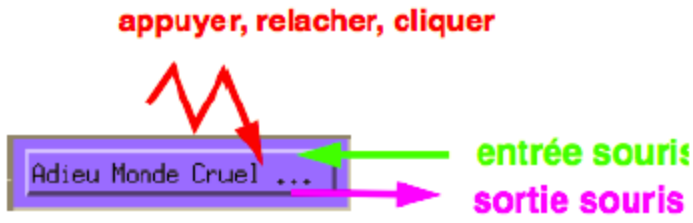
Exemple : ActionEvent

- Evénement : ActionEvent
- Listener : ActionListener
- Méthode : actionPerformed(**ActionEvent**)

Exemple : MouseEvent

- Evénement : **MouseEvent**

- Listener : **MouseListener**
- Méthodes :
  - mouseClicked(MouseEvent)
  - mouseEntered(MouseEvent)
  - mouseExited(MouseEvent)
  - mousePressed(MouseEvent)
  - mouseReleased(MouseEvent)



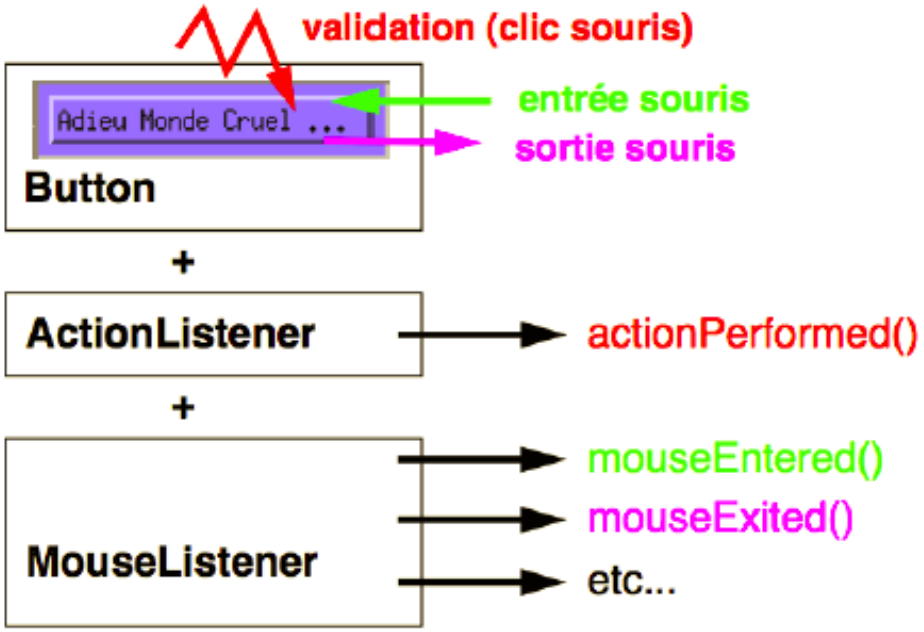
- Listener : **MouseMotionListener**
- Méthodes :
  - mouseDragged(MouseEvent)
  - mouseMoved(MouseEvent)

Remarque

- **toutes** les méthodes doivent être **implémentées**
- car les **Listeners** sont des **interfaces** (au sens du langage Java)

Associer des Listeners aux composants graphiques

- un **composant** peut avoir plusieurs **listeners**
- un même **listener** peut être associé à plusieurs **composants**



Exemple

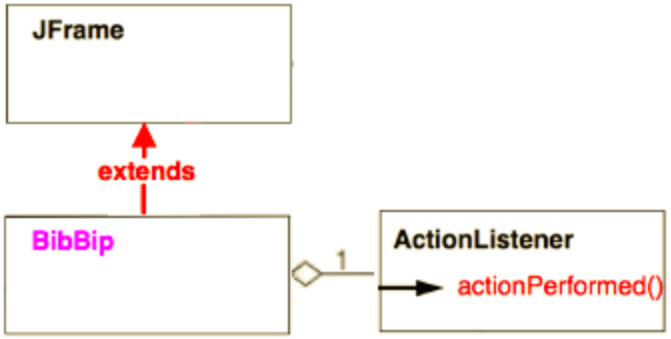
```
import javax.swing.*;
import java.awt.event.*;

BipBip extends JFrame {
    JButton doIt, close;
    JLabel label = new JLabel();

    public static void main(String argv[ ])  {
        new BipBip();
    }

    public BipBip()  {
        add(doIt = new JButton ("Do It"));
        add(close = new JButton ("Close"));

        doIt.addActionListener(new DoItListener( ));
        close.addActionListener(new CloseListener( ));

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }
```

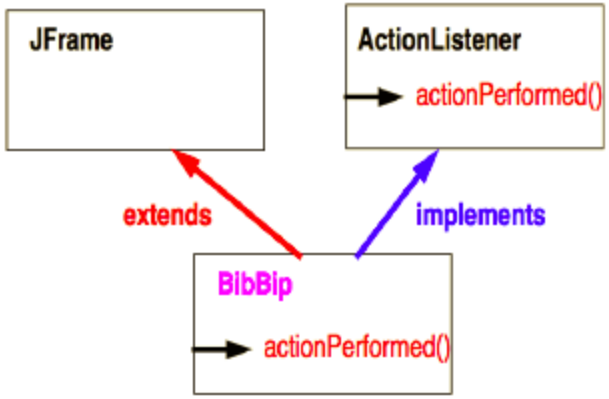
```
class DoItListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        label.setText("Done!");
    }
}

class CloseListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}

}   // fin de la classe BibBip
```

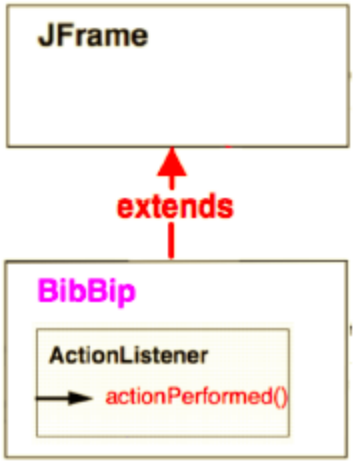
Remarque

- actionPerformed() à accès à label car DoItListener est une classe imbriquée de BibBip

Synthese


JFrame
extends
BibBip
1
ActionListener
actionPerformed()


JFrame
ActionListener
actionPerformed()
extends
implements
BibBip
actionPerformed()


JFrame
extends
BibBip
ActionListener
actionPerformed()