**IUT d’Orsay - DUT Informatique - Jean-Claude MARTIN – JEAN-CLAUDE.MARTIN@U-PSUD.FR**

# Programmation Orientée Objet (POO)

# COURS 7 : Interfaces graphiques : composants et conteneurs

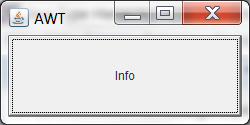
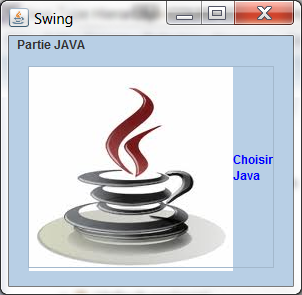
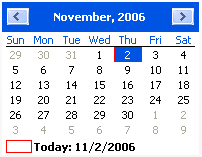
**Interfaces graphiques en Java**

* **Tout est objet : conteneurs / composants, évenements / écouteurs, gestionnaires de mise en page**

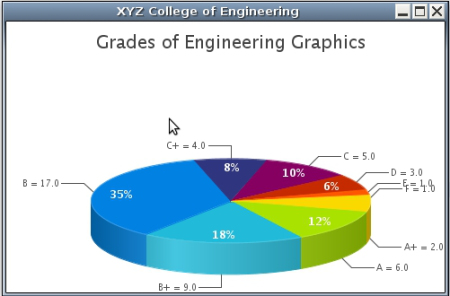
**IMPORTANT : comme tout programme orienté objet, créer et gérer une interface graphique consiste à :**

**1) …………………………………………………….**

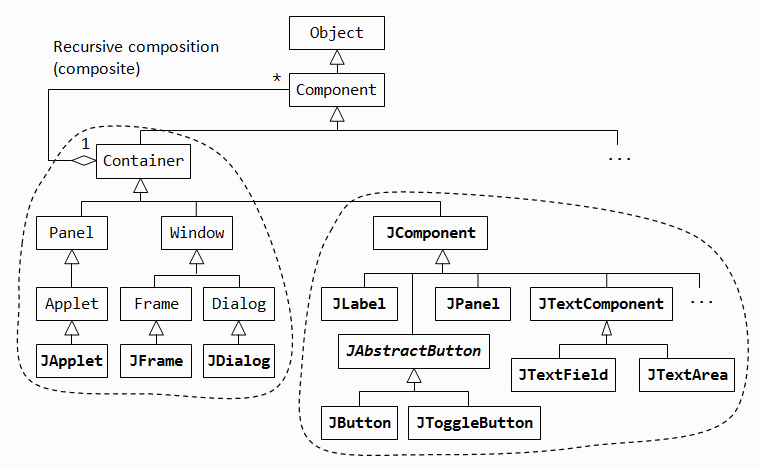
**2) ……………………………………………………...**

**  **

* **Librairies graphiques en Java**
  + **AWT : Abstract Window Kit: utilise via une couche d’abstraction des composants dits “lourds” dépendants et dessinés par le système d’exploitation natif   
    => pas très jolis, pas très sophistiqués et en nombre limité**
  + **Swing : composants légers, indépendants du système d’exploitation (donc pas natifs) et dessinés par Java (certaines parties du dessin des fenêtres restent à la charge du système)**
  + **Standard Widget Toolkit: librairie native fournie par Eclipse**
  + **Java / FX depuis 2014 : peut-être combiné avec Swing ; métaphore du théâtre, feuilles de style, diagrammes, gestion des touch … ex : PieChart3D :**

****

**Hiérarchie de quelques classes de composants et conteneurs Swing**

****

* **En gras : classes Swing, Non gras : classes AWT**

**Un premier exemple de conteneur et composant**

**Afficher une fenêtre**

* **Créer une fenêtre**
* **Lui demander de s’afficher**
* **Lui donner sa taille**
* **Lui dire de quitter le programme quand on clique sur la p’tite croix**
* **Javax = java extended**

**import javax.swing.JFrame;**

**// Creer une fenetre et appeler des methodes sur cette fenetre**

**public class FenetreSimple {**

**public static void main(String[] args) {**

**// Creer une fenetre**

**…**

**// Lui donner sa taille**

**…**

**// Lui dire de terminer le programme quand on clique sur la petite croix**

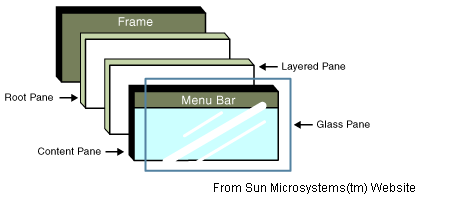
**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Afficher la fenetre**

**…**

**// ou fenetre.setBounds (100, 100, 100, 400);**

**Les conteneurs d’une fenêtre**

****

**Etapes dans la construction d’une interface**

* **Créer un CONTENEUR (par exemple UNE JFRAME)**
* **Créer un COMPOSANT (par exemple UN JBUTTON)**
* **Ajouter le COMPOSANT (JBUTTON) au CONTENEUR (LA JFRAME)**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**// Créer une fenêtre et y ajouter un bouton**

**public class FenetreBouton {**

**public static void main(String[] args) {**

**// Créer une fenêtre**

**JFrame fenetre = new JFrame ("Le soleil brille");**

**fenetre.setSize (300, 300);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Créer un bouton**

**…**

**// Ajouter le bouton à la fenêtre**

**…**

**// Afficher la fenêtre**

**fenetre.setVisible(true);**

* **REMARQUE :** 
  + **JDK < 5.0 : on n’ajoute pas directement dans la fenêtre : on ajoute sur le carreau (pane = vitre) : les 2 fonctionnent :** 
    - **fenetre.getContentPane().add(bouton);**
    - **fenetre.add(bouton);**
* **LIMITES DU PROGRAMME**
  + **Exécution : IL NE SE PASSE RIEN QUAND ON CLIQUE SUR LE BOUTON**
  + **Mise en page : PAS JOLI**
  + **On ne peut afficher qu’un bouton**

**=> Il faut créer des objets spécifiques pour s’occuper de la mise en page et de la gestion des événements**

**Méthodes de la classe JFrame**

* + **setResizable (false) : la taille n’est plus modifiable par l’utilisateur**
  + **toFront() / toBack()**

**Conteneurs**

**DEFINITION: un conteneur peut contenir**

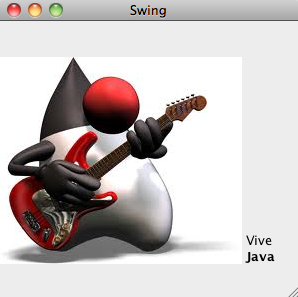
**des COMPOSANTS et des CONTENEURS**

**SCHEMA DE LA MEMOIRE DE FenetreBouton**

**Composants**

* **Widget = Window + gadget = composant d’une interface graphique**
* **Tous les composants Swing héritent de la classe JComponent**
* **Il faut éviter de mélanger dans vos programmes des composants AWT (Button) et des composants Swing (JButton)**
* **Vous devrez cependant utiliser certaines classes d’AWT car la hiérarchie de classes graphique fait intervenir des classes AWT et Swing**

**JLabel**

****

**import javax.swing.\*;**

**public class ExempleJLabel {**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Swing");**

**fenetre.setSize (300, 300);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Creer un JLabel**

**// Ajouter une icone au label**

**// Changer le texte du label**

**// Changer l'alignement (par defaut : centre verticalement)**

**l.setVerticalTextPosition (SwingConstants.*BOTTOM*);**

**// Ajouter le label dans un des conteneurs de la fenetre**

**// Afficher la fenetre**

**fenetre.setVisible(true);**

**JButton**

****

**import java.awt.\*;**

**import javax.swing.\*;**

**public class ExempleJButton {**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Swing");**

**fenetre.setSize (300, 300);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Creer un JButton**

**……………………………………………………………………………**

**// Dire de changer l'icone quand le bouton est clique**

**b.setPressedIcon(new ImageIcon ("java-design.jpeg"));**

**// Changer le texte du label**

**b.setText("<html>Choisir <p><b>Java</b> </html>");**

**// Associer un raccourci clavier pour cliquer sur le bouton : Alt + n**

**b.setMnemonic('n');**

**// Changer le curseur quand la souris passe sur le bouton**

**Cursor c = Cursor.*getPredefinedCursor* (Cursor.*CROSSHAIR\_CURSOR*);**

**b.setCursor (c);**

**// Afficher une bulle d'aide quand la souris passe sur le bouton**

**b.setToolTipText ("<html>Cliquez pour en savoir plus sur <p><b>Java</b></html>");**

**// Specifier les couleurs du bouton**

**b.setForeground(Color.*blue*); // ou new Color (int r, int g, int b);**

**b.setBackground(new Color (200, 100, 0));**

**b.setOpaque(true); // Pour que la couleur s'affiche par dessus celle du conteneur**

**// Mettre une bordure**

**//b.setBorder(BorderFactory.createLoweredBevelBorder());**

**b.setBorder(BorderFactory.*createTitledBorder*("Partie JAVA"));**

**// Ajouter le bouton dans un des conteneurs de la fenetre**

**fenetre.getContentPane().add(b);**

**// Afficher la fenetre**

**fenetre.setVisible(true);**

**// Afficher la taille actuelle du bouton dans la console**

**Dimension d = b.getSize();**

**System.*out*.println (d);**

**// Afficher la taille preferee du bouton dans la console**

**Dimension dp = b.getPreferredSize();**

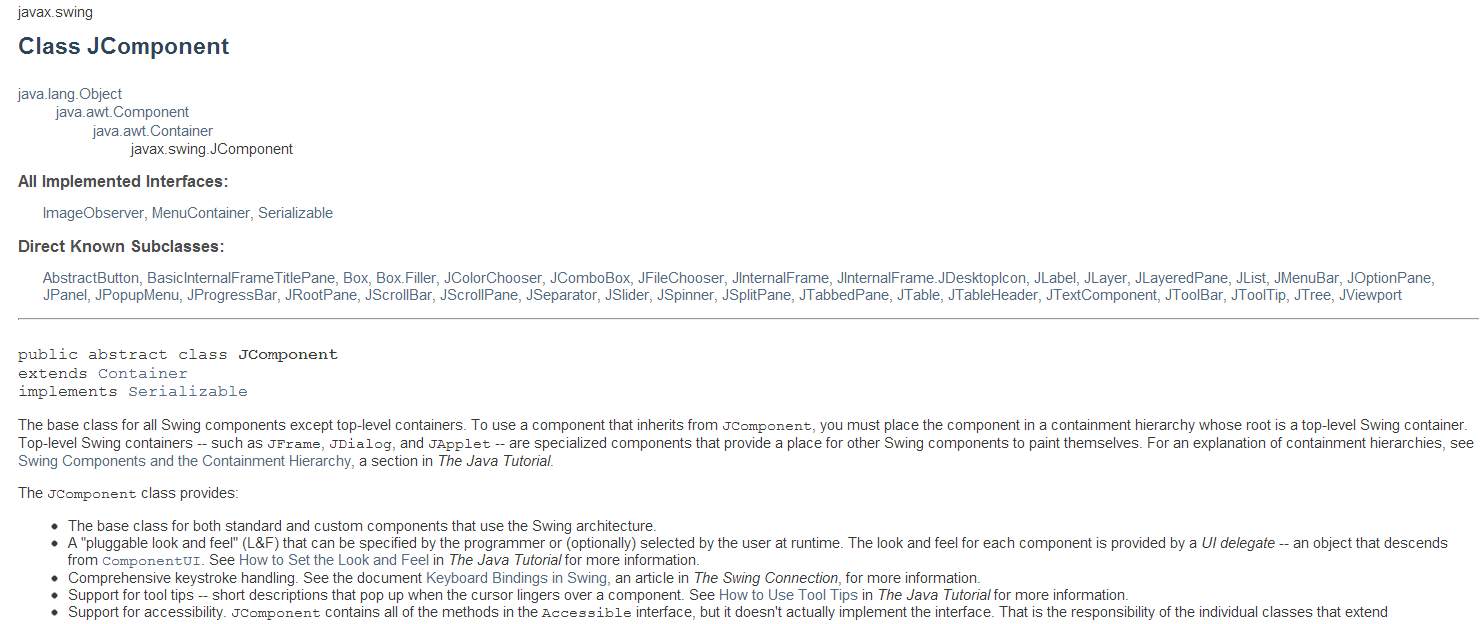
**System.*out*.println (dp);**

**}**

**}**

**Capacités communes aux composants Swing**

* **Définis dans la classe abstraite JComponent**
  + **Visible / Invisible : setVisible (boolean)**
  + **Activation / désactivation : setEnabled(boolean)**
  + **Taille actuelle :** 
    - **Dimension d = b.getSize(null);**
    - **setSize() : à éviter pour les composants : utiliser plutôt des layouts**
  + **Taille préférée : Dimension getPreferedSize()**
  + **Curseur : Utiliser une usine / factory à curseur :** 
    - * **Cursor c = Cursor.getPredefinedCursor (Cursor.CROSSHAIR\_CURSOR);**
  + **Bulles d’aides (tooltip)**
    - **b.setToolTipText (“<html>Cliquez pour en savoir plus sur <p><b>Java</b></html>”);**
  + **Couleur du composant : b.setForeground(Color.blue);**
  + **Bordures : b.setBorder(BorderFactory.createTitledBorder("Partie JAVA"));**

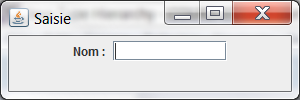
****

**Le conteneur JPanel**

**Une JFrame ne peut pas contenir plus d’un composant dans son content pane.**

**Un JPanel est un conteneur qui peut contenir plusieurs composants (ou plusieurs JPanels).**

**Un JPanel a besoin d’être mis dans une JFrame pour être affiché.**

****

**import javax.swing.\*;**

**public class FenetreSaisie {**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Saisie");**

**// Lui donner sa taille**

**fenetre.setSize (300, 100);**

**// Lui dire de terminer le programme quand on clique sur la petite croix**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Créer un JPanel**

**JPanel p = ...**

**// Ajouter un label et un champ de texte**

**JLabel l = ...**

**JTextField t = new JTextField (20);**

**// Ajouter les composants au panel**

**...**

**...**

**// Ajouter le panel à la fenêtre**

**fenetre.getContentPane().add(p);**

**// Afficher la fenetre**

**fenetre.setVisible(true);**

**}**

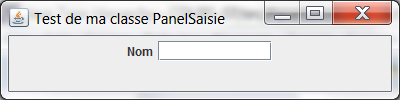
**}**

**Les exemples précédents n’étaient pas en POO ☹ (tout est programmé dans le main() !**

**Il est préférable d’étendre une classe comme JPanel : cela permet de**

**1) Créer plusieurs instances de votre classe graphique**

**2) Etendre votre classe graphique pour rajouter d’autres composants et comportements**

****

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JPanel;**

**import javax.swing.JTextField;**

**// Etendre la classe Panel**

**public class PanelSaisie extends ............................... {**

**// Déclarer les composants du panel en tant qu'attributs pour qu'ils soient accessibles par toutes les méthodes**

**JLabel l ;**

**JTextField t ;**

**PanelSaisie (String s) {**

**// Créer les composants**

**l = new JLabel (s);**

**t = new JTextField (10);**

**// Ajouter les composants dans CE paneau : ceci est possible car ...**

**add(l);**

**add(t);**

**}**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Test de ma classe PanelSaisie");**

**fenetre.setSize (400, 100);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Creer une instance de ma classe**

**PanelSaisie p = new PanelSaisie ("Nom");**

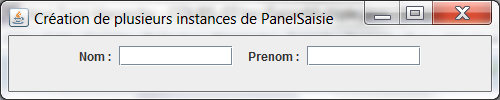
**// Ajouter mon instance dans un des conteneurs de la fen?tre**

**fenetre.getContentPane().add(p);**

**// Afficher la fenetre**

**fenetre.setVisible(true);**

1. **L’héritage permet de créer plusieurs instances de votre classe graphique**

****

**import javax.swing.JFrame;**

**import javax.swing.JPanel;**

**public class SaisieMultiple {**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Création de plusieurs instances de PanelSaisie");**

**fenetre.setSize (500, 100);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Creer des instances PanelSaisie**

**PanelSaisie pn = new PanelSaisie ("Nom : ");**

**PanelSaisie pp = new PanelSaisie ("Prenom : ");**

**// Les regrouper dans un 3ème JPanel**

**JPanel panelGlobal = new JPanel();**

**panelGlobal.add(pn);**

**panelGlobal.add(pp);**

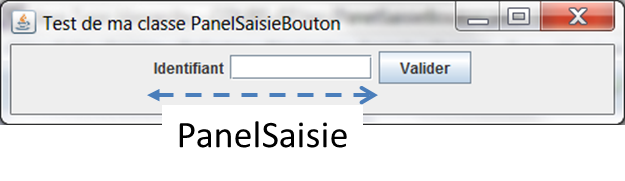
**// Ajouter mon instance dans un des conteneurs de la fen?tre**

**fenetre.getContentPane().add(panelGlobal);**

**// Afficher la fenetre**

**fenetre.setVisible(true); }**

1. **L’héritage permet d’ajouter des composants (et des comportements) dans une sous-classe**

****

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JPanel;**

**import javax.swing.JTextField;**

**public class PanelSaisieBouton extends ................... {**

**// Rajouter un composant**

**JButton b ;**

**PanelSaisieBouton(String s) {**

**super (s);**

**// Creer le bouton**

**b = new JButton ("Valider");**

**// Ajouter le bouton dans CE panel : ceci est possible car PanelSaisieBouton hérite de PanelSaisie**

**// qui hérite elle-même de Panel**

**add (b);**

**}**

**public static void main(String[] args) {**

**// Creer une fenetre**

**JFrame fenetre = new JFrame ("Test de ma classe PanelSaisieBouton");**

**fenetre.setSize (500, 100);**

**fenetre.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);**

**// Creer une instance de ma classe**

**PanelSaisieBouton p = new PanelSaisieBouton ("Identifiant");**

**// Ajouter mon instance dans un des conteneurs de la fen?tre**

**fenetre.getContentPane().add(p);**

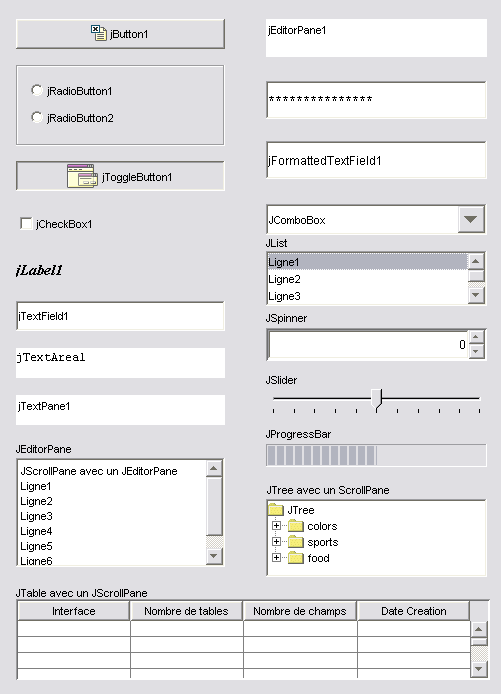
**// Afficher la fenetre**

**fenetre.setVisible(true);**

**}**

**}**

**Autres composants**

****