



ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR

*Membre de*   
HONORIS UNITED UNIVERSITIES

# POO 3 (JAVA ET JAVA AVANCÉ)

**Prof. Nisrine DAD**

**4° Ingénierie Informatique et Réseaux - Semestre I**

Ecole Marocaine des Sciences d'Ingénieur

Année Universitaire : 2022/2023

## VII. LES INTERFACES GRAPHIQUES AVEC SWING

## VII. LES INTERFACES GRAPHIQUES AVEC SWING

### Interface graphique

- **GUI** Graphical User Interface.
- **IHM** Interface Homme Machine.
- Une interface graphique assure la communication entre le programme et l'utilisateur.
- L'interface est constituée :
  - d'une fenêtre principale et d'éventuellement de fenêtres secondaires
  - de fenêtres de dialogue
- Une fenêtre est composée de composants graphiques : zones d'édition, boutons, listes déroulantes, cases à cocher, menus.
- L'utilisateur interagit avec le logiciel grâce à une souris et/ou un touchpad et/ou le clavier, ecran tactile etc . . .

## VII. LES INTERFACES GRAPHIQUES AVEC SWING

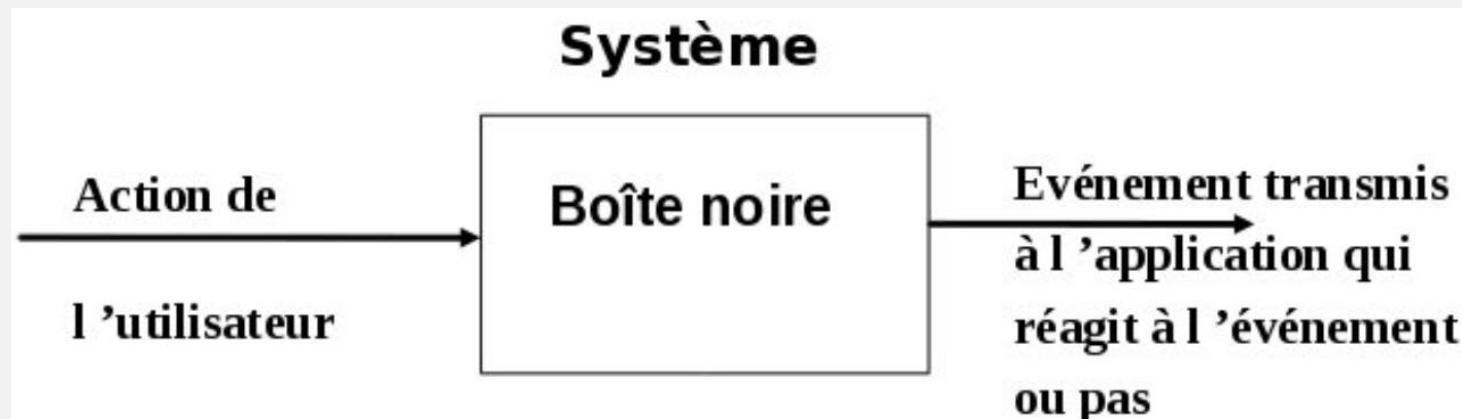
### Programmation événementielle

- **Interactivité** : l'utilisateur agit sur l'interface graphique.
- A chaque action est associé un événement.
- A chaque événement est associé un morceau de code.
- Dès que vous lancez un programme avec une interface graphique :
  - une boucle infinie est générée.
  - le programme attend les actions de l'utilisateur (enfoncer une touche de clavier, clic souris etc . . .)
- Le programme traite l'action si elle a été prévue.

## VII. LES INTERFACES GRAPHIQUES AVEC SWING

### Programmation événementielle

- **Événement** : objet construit par le système en réponse à une action de l'utilisateur et qui contient toutes les informations concernant cette action.
  - **Clic souris** : les coordonnées de la souris au moment du clic, l'heure système.
  - **Clavier** : la touche utilisée, etc . . .



## VII. LES INTERFACES GRAPHIQUES AVEC SWING

### MVC

- La partie données qu'on appelle le **modèle**.
- La partie visualisation (Interface graphique) qu'on appelle **la vue**.
- La partie gestion des évènements qu'on appelle le **contrôleur**.

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Les API d'interfaces en Java

- Java 1.1 (1996) : package **awt**.
- Java 1.2 : package **Swing** plus complet et light-weight.

AWT	Swing
Les composants AWT dépendent de la plate-forme.	Les composants Java swing ne dépendent pas de la plateforme. Ils sont purement scriptés en Java.
Lourd.	Légers, les composants sont conçus pour être au TOP des composants AWT et effectuer leurs tâches. Ces composants sont généralement légers car ils n'ont pas besoin d'objets OS natifs pour implémenter leurs fonctionnalités.
Les composants AWT en Java ne suivent pas l'architecture MVC (Model View Controller)	Les composants Swing en Java suivent le modèle MVC (Model View Controller).

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Les API d'interfaces en Java

AWT	Swing
Les composants AWT sont moins nombreux que les composants Swing.	Les composants Java Swing sont bien plus nombreux.
Les composants AWT ont besoin du package <b>java.awt</b> .	Les composants Swing ont besoin du package <b>javax.swing</b> .
Les fonctionnalités avancées décrites par les composants Swing ne sont pas présentes dans les composants AWT.	Les composants Swing décrivent plusieurs fonctionnalités avancées telles que <b>JTable</b> , <b>JtabbedPane</b> , etc. Ces fonctionnalités avancées sont spécifiques uniquement aux composants Swing.
De nombreuses fonctions de composants AWT doivent être implémentées par le programmeur.	Les composants Swing offrent des fonctions intégrées.

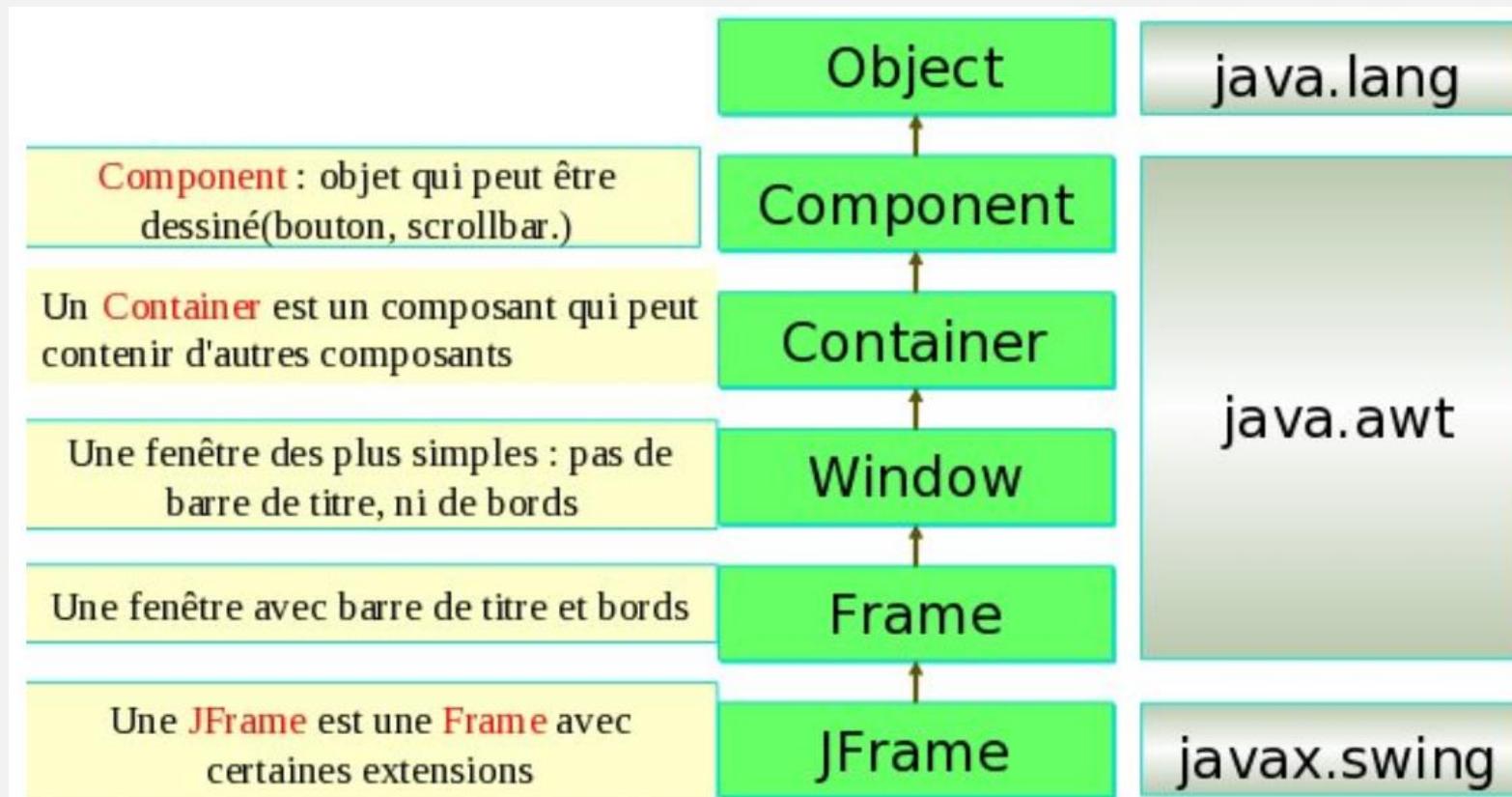
## VII. LES INTERFACES GRAPHIQUES AVEC SWING

### Les fenêtres Jframe

- Pour une application graphique standard développée sur la machine locale, on utilise la classe **JFrame**.
- Pour une applet développée pour le web, c'est la classe **JApplet**.
- Dans la mesure du possible, éviter le mélange des composants swing et awt.
- Les classes **Swing** commencent par la lettre **J**

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Les fenêtres JFrame

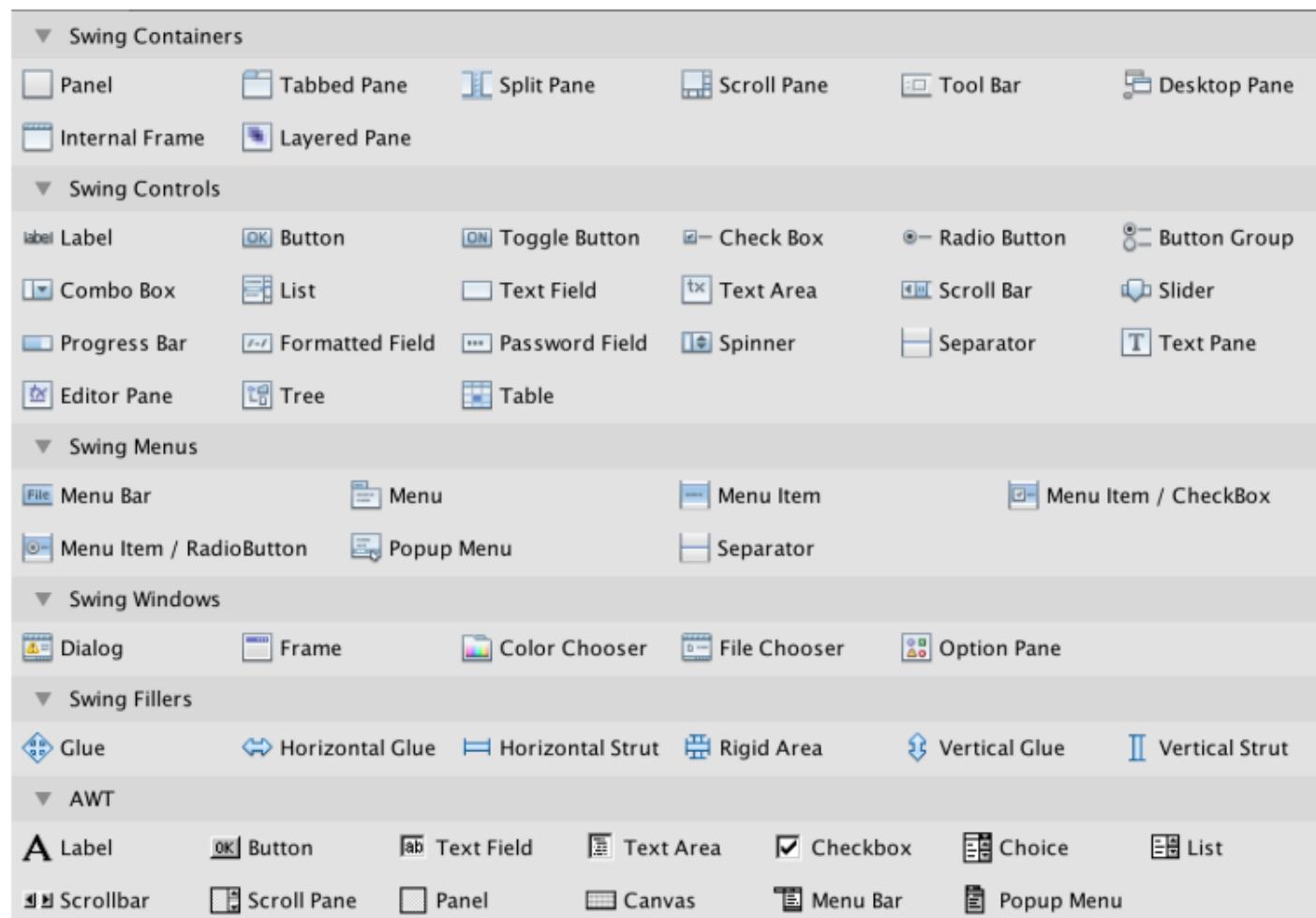


# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Composants graphiques

### Les widgets

- JLabel
- JButton
- JToggleButton
- JCheckbox
- JRadioButton
- ButtonGroup
- JComboBox
- JList
- JTextField
- JTextArea
- JScrollPane
- JMenuBar
- JPopupMenu

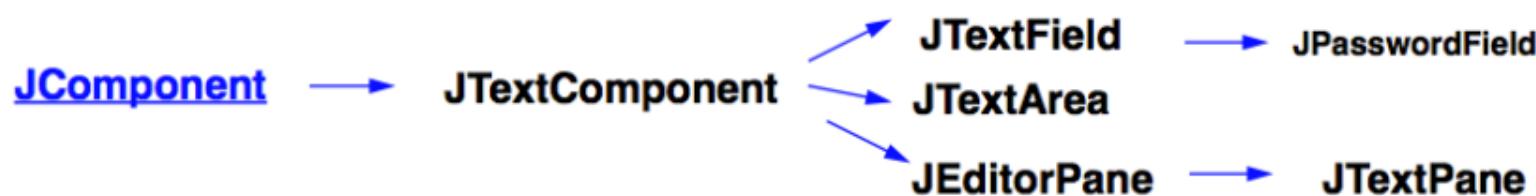


### Les containers

- JWindow
  - JFrame
  - JDialog
    - JFileDialog
- JPanel
  - Applet
- JTabbedPane
- JScrollPane

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Composants texte



**JTextField**



**JPasswordField**

Enter the password:

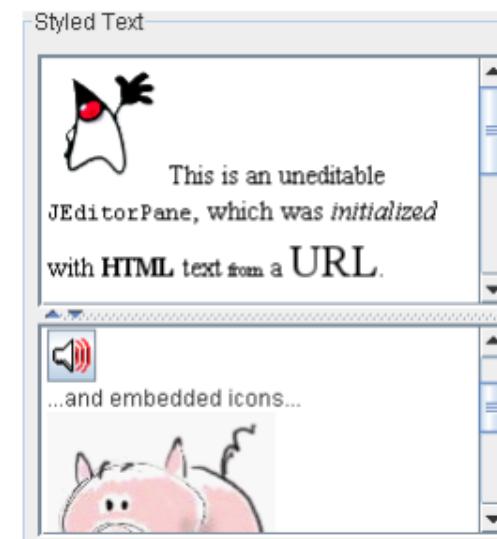
**JTextArea** :

texte simple multilignes

Ascenseurs :

**JScrollPane**

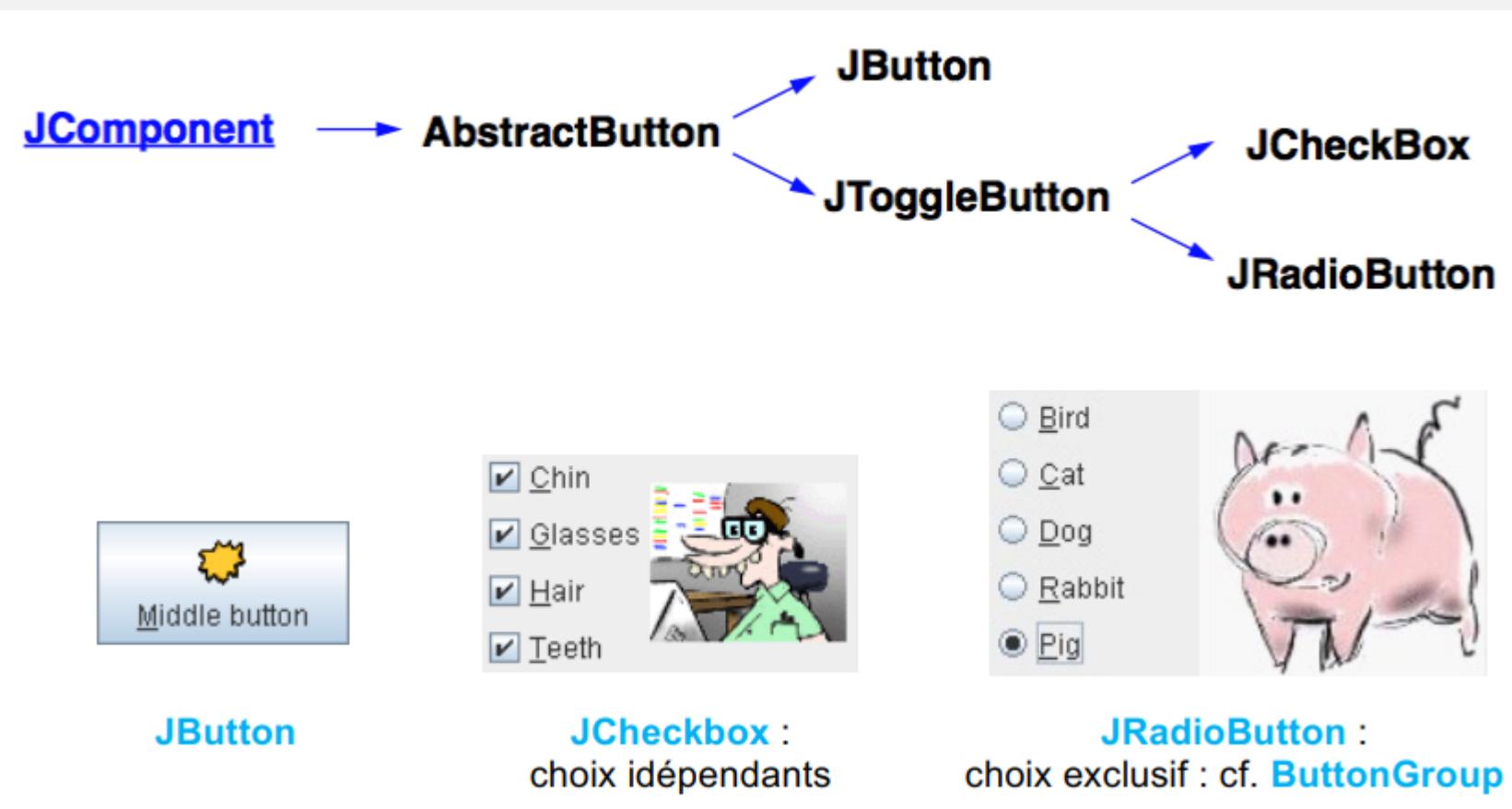
This is an editable JTextArea. A text area is a "plain" text component, which means that although it can display text in any font, all of the text is in the same font.



**JEditorPane** : texte avec styles compatible HTML et RTF

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Composants bouton



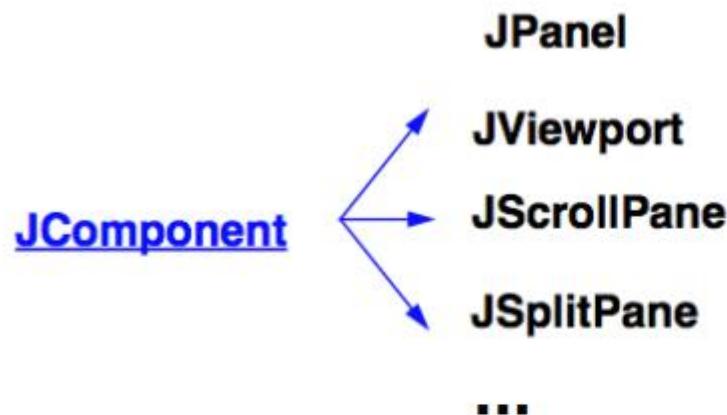
# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Composants interacteur



# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Conteneurs



**JScrollPane:**  
avec ascenseurs intégrés



**JPanel:** conteneur générique



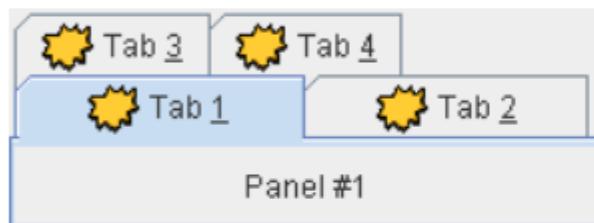
**JSplitPane:**  
avec « diviseur » intégré

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Conteneurs



**JToolBar:** barre d'outils  
(sous la barre de menus)



**JTabbedPane:**  
onglets



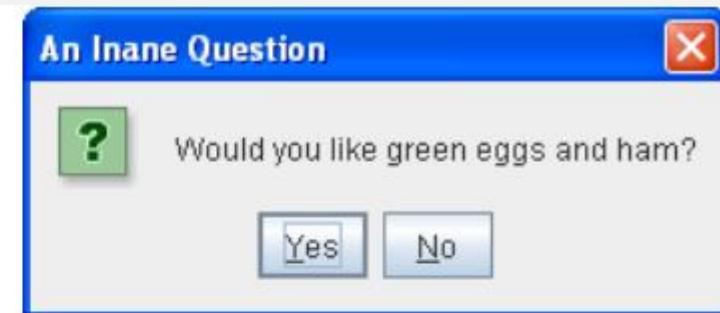
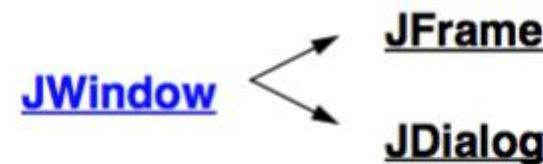
**JTree**

Host	User	Password	Last Modified
Biocca Games	Freddy	I#asf6Awwzb	Mar 16, 2006
zabble	ichabod	Tazbl!34\$fZ	Mar 6, 2006
Sun Developer	fraz@hotmail.co...	AasVW541fbZ	Feb 22, 2006
Heirloom Seeds	shams@gmail....	bkz[ADF78!	Jul 29, 2005
Pacific Zoo Shop	seal@hotmail.c...	vbAf124%z	Feb 22, 2006

**JTable**

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Fenêtres



**JFrame** : fenêtre **principale** de l'application

**JDIALOG** : fenêtre **secondaire**

- dépendante de la **JFrame** (en théorie pas d'iconification séparée, toujours au dessus)

## VII. LES INTERFACES GRAPHIQUES AVEC SWING

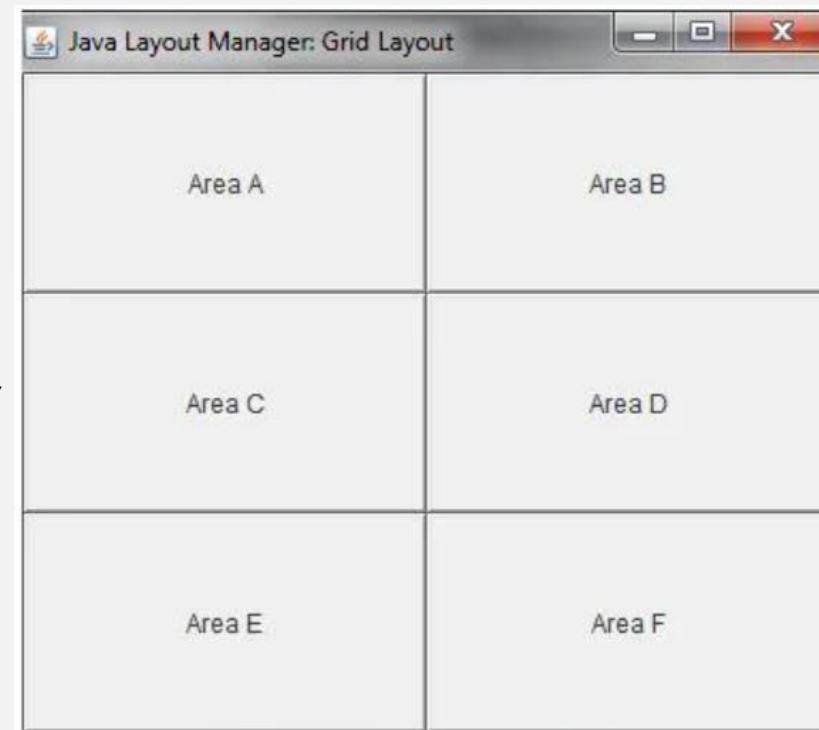
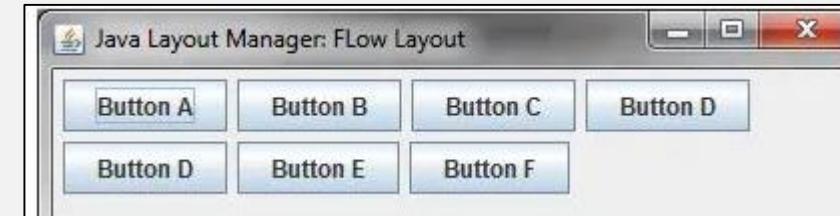
### Les layout

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Les layout

- Les **LayoutManager** pour organiser votre fenêtre.
- **awt.CardLayout**
- **awt.FlowLayout**
- **awt.BorderLayout**
- **awt.GridLayout**
- **awt.GridBagLayout**

```
fj.setLayout(new FlowLayout(FlowLayout.LEFT));
```



```
GridLayout gl = new GridLayout(3, 2);  
setLayout(gl);
```

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

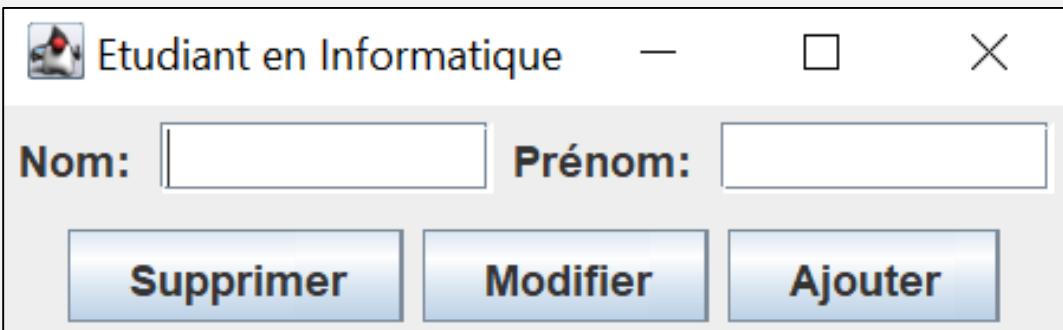
## Les layout

```
pnl.setLayout(new BorderLayout());  
pnl.add(jbtn1, BorderLayout.NORTH);  
pnl.add(jbtn2, BorderLayout.SOUTH);  
pnl.add(jbtn3, BorderLayout.WEST);  
pnl.add(jbtn4, BorderLayout.EAST);  
pnl.add(jbtn5, BorderLayout.CENTER);
```



# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Exemple Création d'une fenêtre

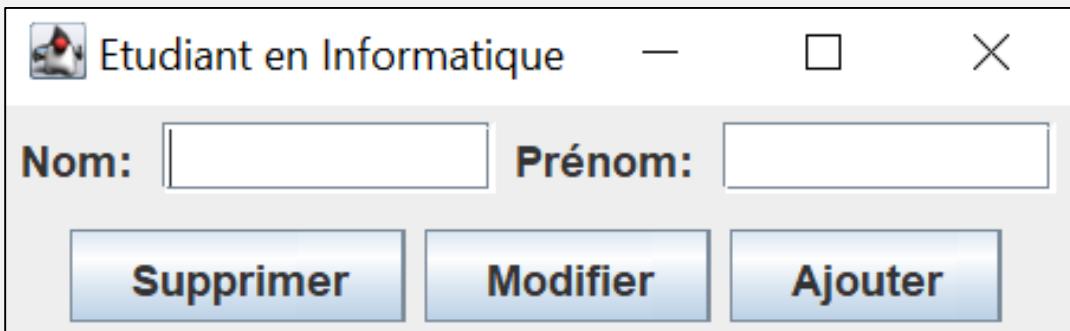


```
import java.awt.*;
import javax.swing.*;

public class FenetreEtudiant extends JFrame{
    // Les composants swing de la fenêtre
    private JLabel nomLabel, prenomLabel;
    private JTextField nomTextField,
    prenomTextField;
    private JButton supprimerButton,
    modifierButton, ajouterButton;
```

## VII. LES INTERFACES GRAPHIQUES

### Exemple Création d'une fenêtre



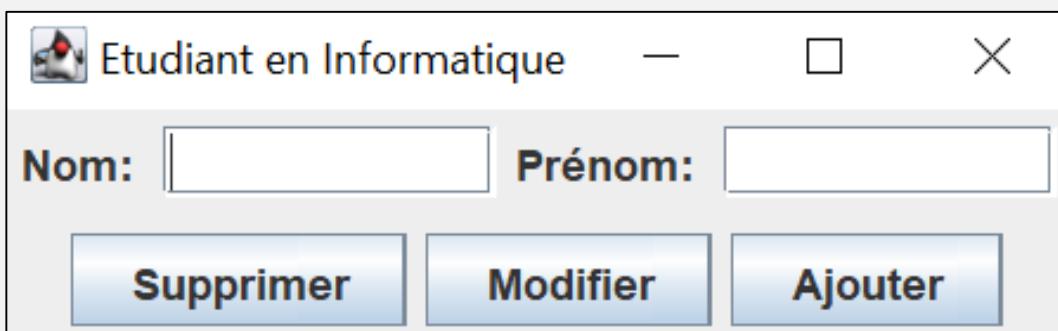
```
public void initComponents() {  
    nomLabel=new JLabel("Nom: ");  
    prenomLabel=new JLabel("Prénom: ");  
    nomTextField=new JTextField(10);  
    prenomTextField=new JTextField(10);  
    supprimerButton=new JButton("Supprimer");  
    modifierButton=new JButton("Modifier");  
    ajouterButton=new JButton("Ajouter");  
  
    JPanel nomPrenomPanel=new JPanel();  
    nomPrenomPanel.add(nomLabel);  
    nomPrenomPanel.add(nomTextField);  
    nomPrenomPanel.add(prenomLabel);  
    nomPrenomPanel.add(prenomTextField);  
  
    JPanel boutonsPanel=new JPanel();  
    boutonsPanel.add(supprimerButton);  
    boutonsPanel.add(modifierButton);  
    boutonsPanel.add(ajouterButton);  
  
    this.add(nomPrenomPanel, BorderLayout.NORTH);  
    this.add(boutonsPanel, BorderLayout.SOUTH);  
}
```

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Exemple Création d'une fenêtre

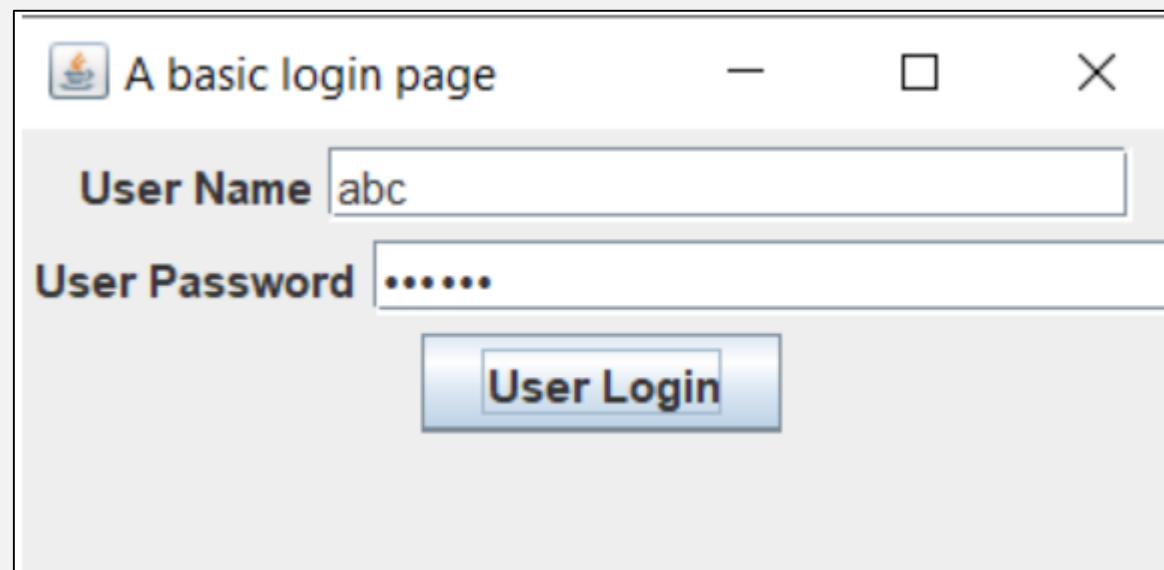
```
public FenetreEtudiant() {  
    this.initComponents();  
    this.setTitle("Etudiant en Informatique");  
    this.setSize(300,500);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        FenetreEtudiant f1=new FenetreEtudiant();  
        f1.pack();  
        //Adapter la taille de la fenêtre à ses composants  
        f1.setVisible(true);  
    }  
}
```

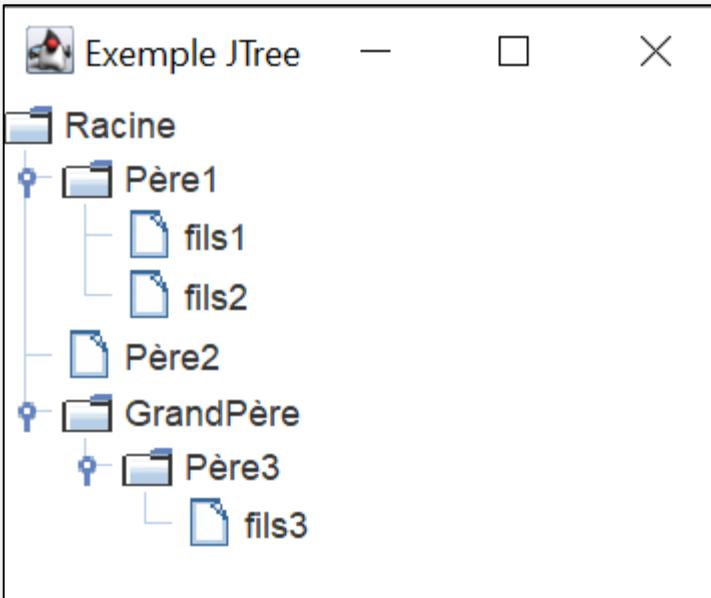


# VII. LES INTERFACES GRAPHIQUES AVEC SWING

## Exercice



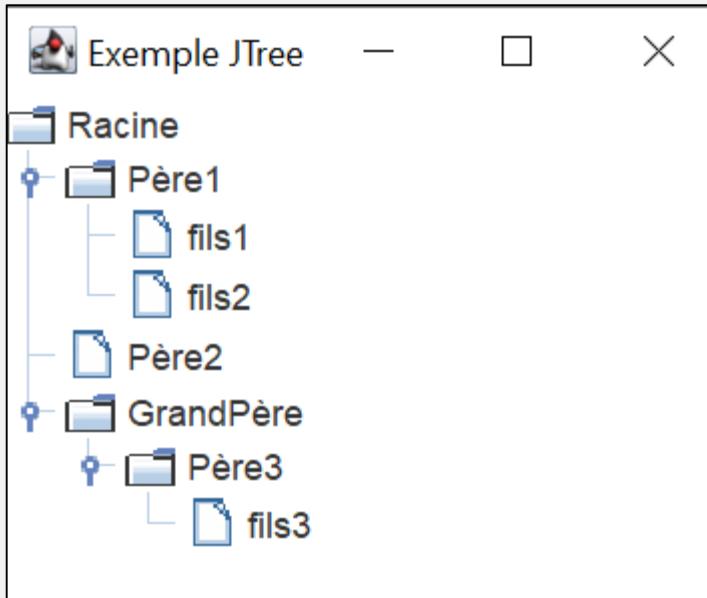
## Exemple JTree



```
public void initComponents() {  
    // Créer la racine  
    DefaultMutableTreeNode racine=new  
    DefaultMutableTreeNode("Racine");  
    // Créer les noeuds fils  
    DefaultMutableTreeNode p1=new  
    DefaultMutableTreeNode("Père1");  
    DefaultMutableTreeNode p2=new  
    DefaultMutableTreeNode("Père2");  
    DefaultMutableTreeNode gp=new  
    DefaultMutableTreeNode("GrandPère");  
    // Créer les noeuds fils  
    DefaultMutableTreeNode f1=new  
    DefaultMutableTreeNode("fils1");  
    DefaultMutableTreeNode f2=new  
    DefaultMutableTreeNode("fils2");  
    DefaultMutableTreeNode f3=new  
    DefaultMutableTreeNode("Père3");  
    DefaultMutableTreeNode f4=new  
    DefaultMutableTreeNode("fils3");  
    DefaultMutableTreeNode f5=new  
    DefaultMutableTreeNode("fils4");  
}
```

# VII. LES INTERFACES GRAPHIQUES AVEC SWING

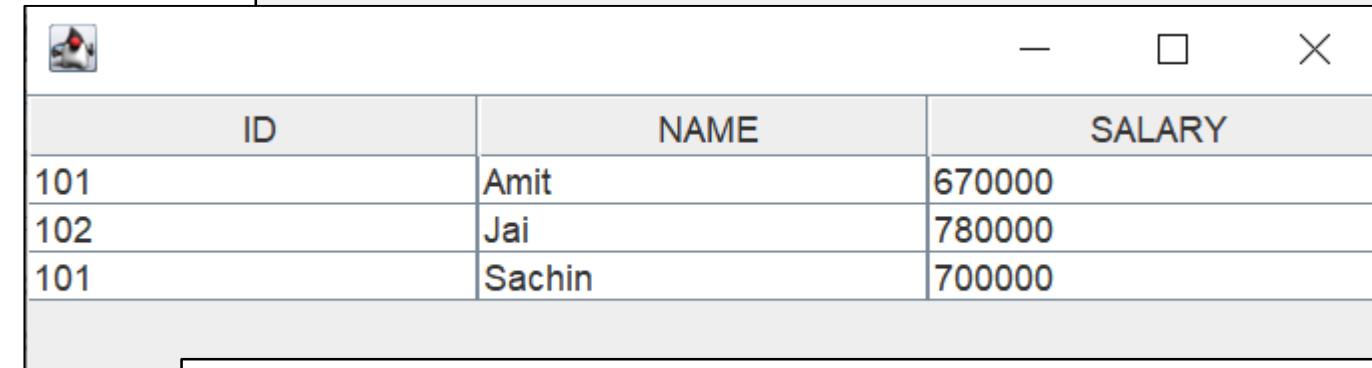
## Exemple JTree



```
public void initComponents() {  
    //...  
    // Ajouter les noeuds fils à leurs parents  
    racine.add(p1);  
    racine.add(p2);  
    racine.add(gp);  
    p1.add(f1);  
    p1.add(f2);  
    gp.add(f3);  
    f3.add(f4);  
    f3.add(f4);  
  
    tree=new JTree(racine);  
    this.add(tree);  
}
```

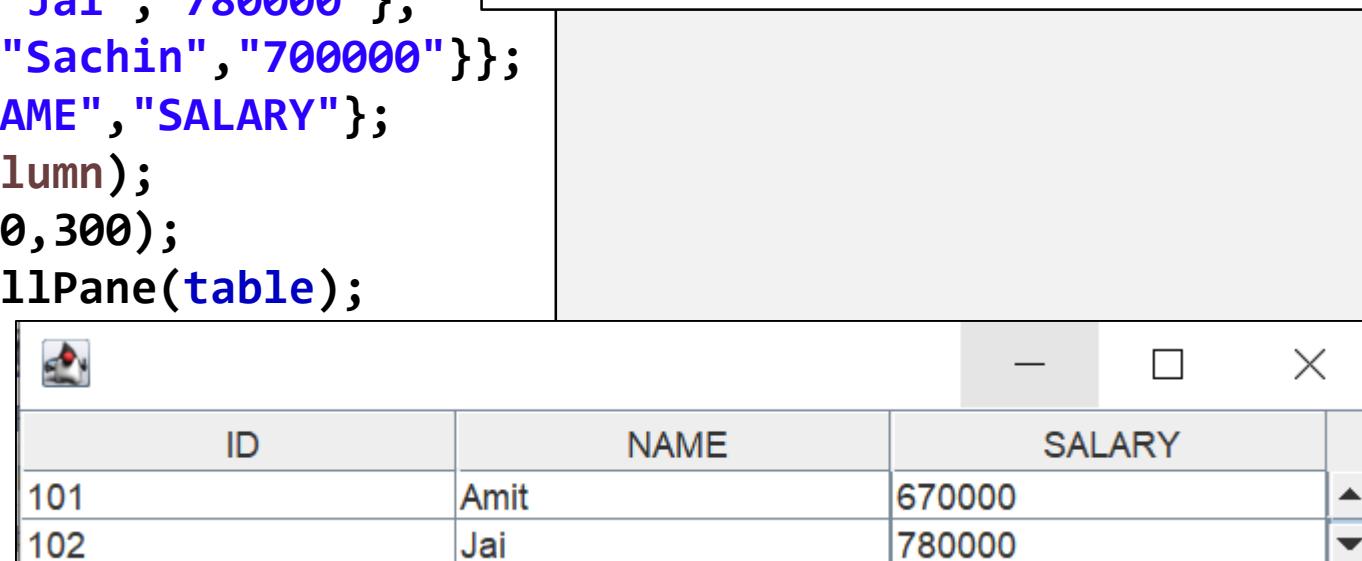
## Exemple JTable

```
import javax.swing.*;  
public class JTableExample extends JFrame{  
    private JTable table;  
  
    public JTableExample() {  
        initComponents();  
        this.setSize(300,400);  
        this.pack();  
        this.setVisible(true);  
    }  
    public void initComponents() {  
        String data[][]={{ "101","Amit","670000"},  
                        {"102","Jai","780000"},  
                        {"101","Sachin","700000"}};  
        String column[]={ "ID", "NAME", "SALARY"};  
        table=new JTable(data,column);  
        table.setBounds(30,40,200,300);  
        JScrollPane sp=new JScrollPane(table);  
        this.add(sp);  
    }  
}
```



ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

```
public class Test {  
    public static void main(String[] args) {  
        JTableExample f1=new JTableExample();  
    }  
}
```



ID	NAME	SALARY
101	Amit	670000
102	Jai	780000