

Cours
Programmation Orientée Objet 2
Pour
ING 2

Chap 04:
Interfaces Graphiques

MEKAHLIA Fatma Zohra LAKRID
Maître de Conférences Classe B

Laboratoire de Modélisation, Vérification et Evaluation des Performances des systèmes complexes (MOVEP)
Bureau 123

PLAN

- Généralités sur les interfaces graphiques.
- Composants des interfaces graphiques.
- Les packages AWT et Swing.
- Classes de base.
- Création et affichage d'une fenêtre.
- Placer des composants dans une fenêtre .
- Gestion des événements.
- Le modèle MVC.

GÉNÉRALITÉS SUR LES INTERFACES GRAPHIQUES

- Généralement les programmes informatiques nécessitent:
 - l’affichage de questions posées à l’utilisateur,
 - l’entrée de données par l’utilisateur au moment de l’exécution,
 - l’affichage d’une partie des résultats obtenus par le traitement informatique.

- Cet échange d’informations peut s’effectuer avec une interface utilisateur (**User Interface**) :
 - en mode texte,
 - ou console,
 - ou encore en **mode graphique**.

2

GÉNÉRALITÉS SUR LES INTERFACES GRAPHIQUES

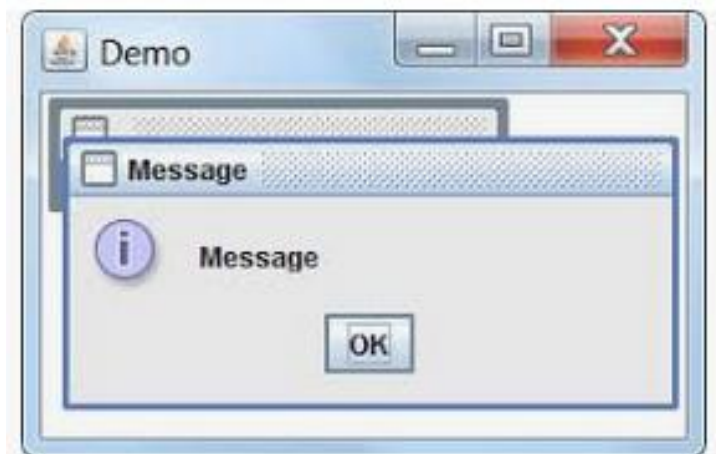
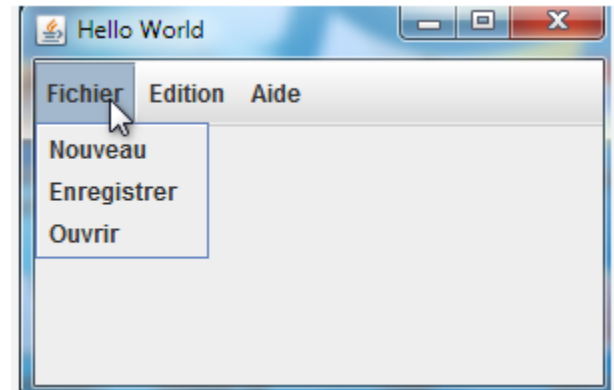
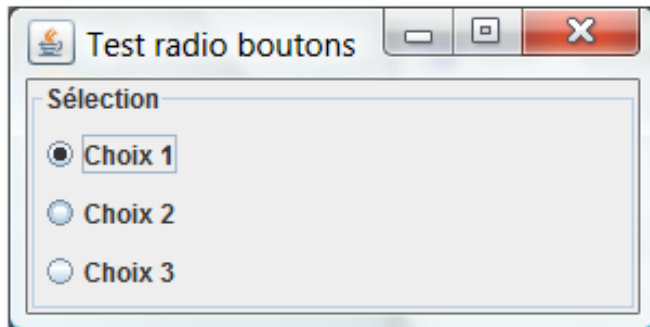
- Une interface graphique est souvent appelé **GUI** (Graphical User Interface), est un ensemble de **composants graphiques permettant à un utilisateur de communiquer avec un logiciel.**



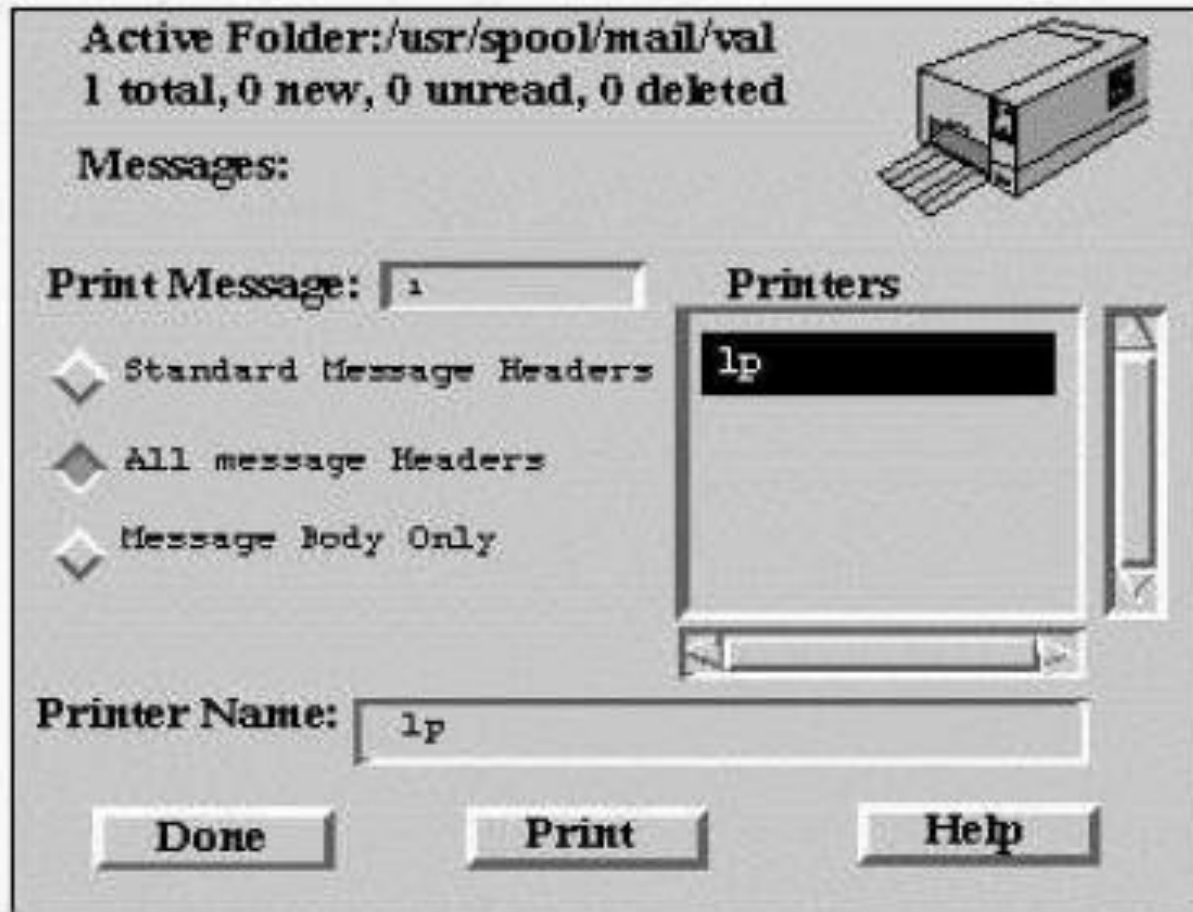
The screenshot shows a window titled "Gestion des salles ENSAJ" with standard Windows window controls (minimize, maximize, close). The interface contains the following elements:

- Two radio buttons for room type: "Salle de cours" (selected) and "Salle de réunion".
- Three text input fields on the left: "Numéro de la salle", "Numéro de l'étage", and "Capacité".
- Two text input fields on the right: "Nom" and "Créneau".
- A "Date" label next to a text input field.
- A "Type" label above two radio buttons: "Amphi" and "Salle".
- A "Valider" button at the bottom right.

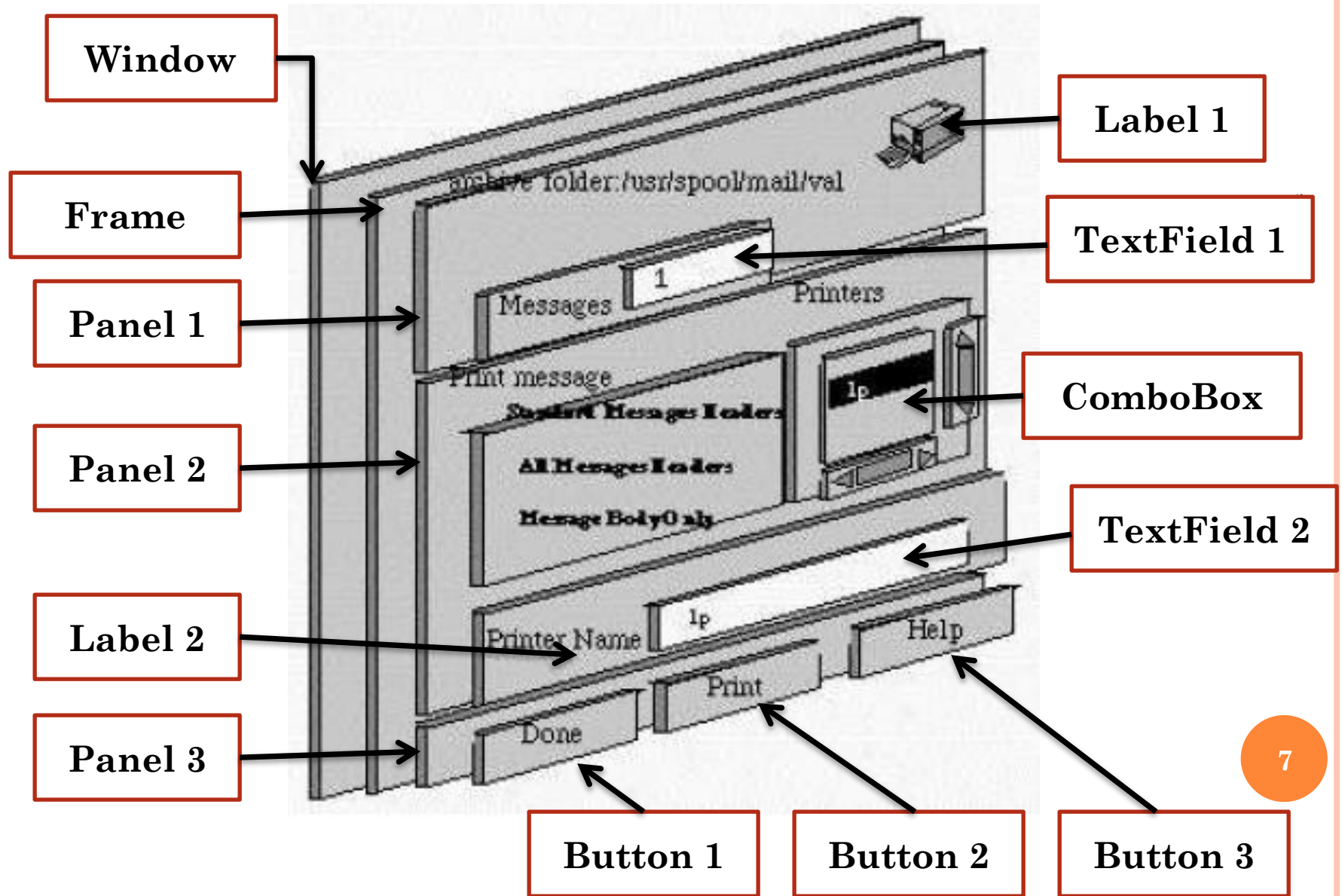
GÉNÉRALITÉS SUR LES INTERFACES GRAPHIQUES



COMPOSANTS DES INTERFACES GRAPHIQUES



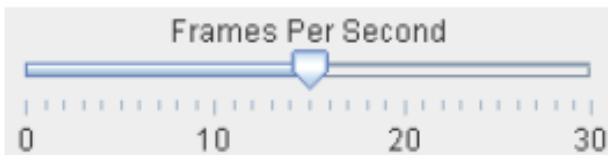
COMPOSANTS DES INTERFACES GRAPHIQUES



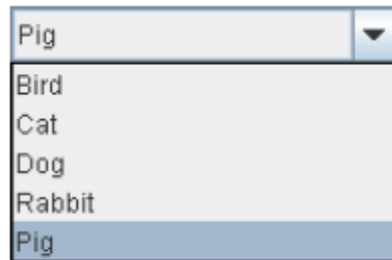
COMPOSANTS DES INTERFACES GRAPHIQUES



ProgressBar



Slider



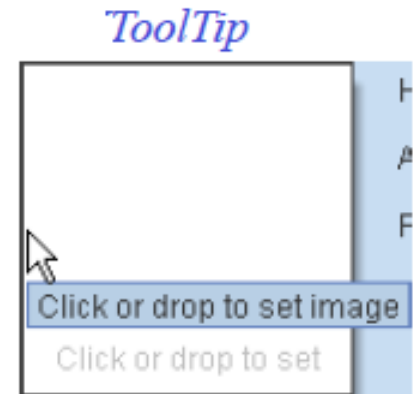
ComboBox



List



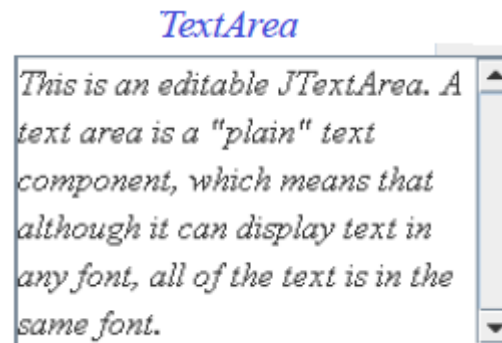
Dialog



ToolTip



Tree



TextArea

More component in:

<https://docs.oracle.com/javase/8/docs/api/java/awt/Component.html>



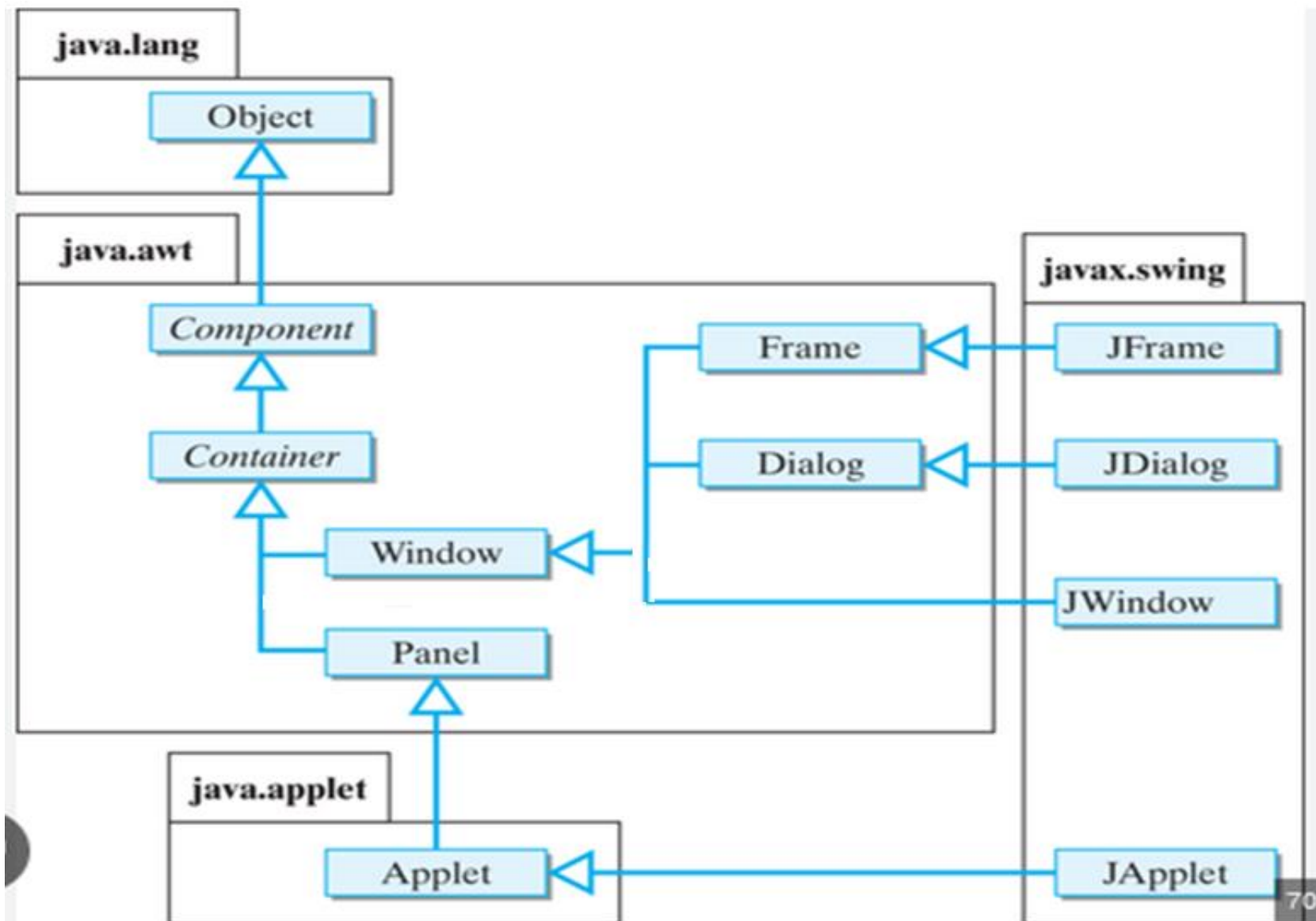
CheckBox

LES PACKAGES AWT ET SWING

- Généralement, les éléments graphiques sont définis dans deux grandes familles de composant graphique:
 - **AWT (abstract window toolkit, *JDK 1.1*)**.
 - **Swing (JDK 1.2)**.
- **Swing** est construit au-dessus de **AWT**:
 - même gestion des événements.
 - les classes de Swing héritent des classes de AWT.

23

LES PACKAGES AWT ET SWING



LES PACKAGES AWT ET SWING

- Swing et AWT font partie de JFC (Java Foundation Classes) qui offre des facilités pour construire des interfaces graphiques.
- Vous pouvez identifier les composants de swing à partir de leur nom qui débute par la lettre **J** (JDialog, JFrame etc.). Ces composants se trouvent dans le paquetage: **javax.swing**
- **x** vient du mot "eXtension« .
- Par contre les composants awt se trouvent dans le package **java.awt**

Les packages AWT

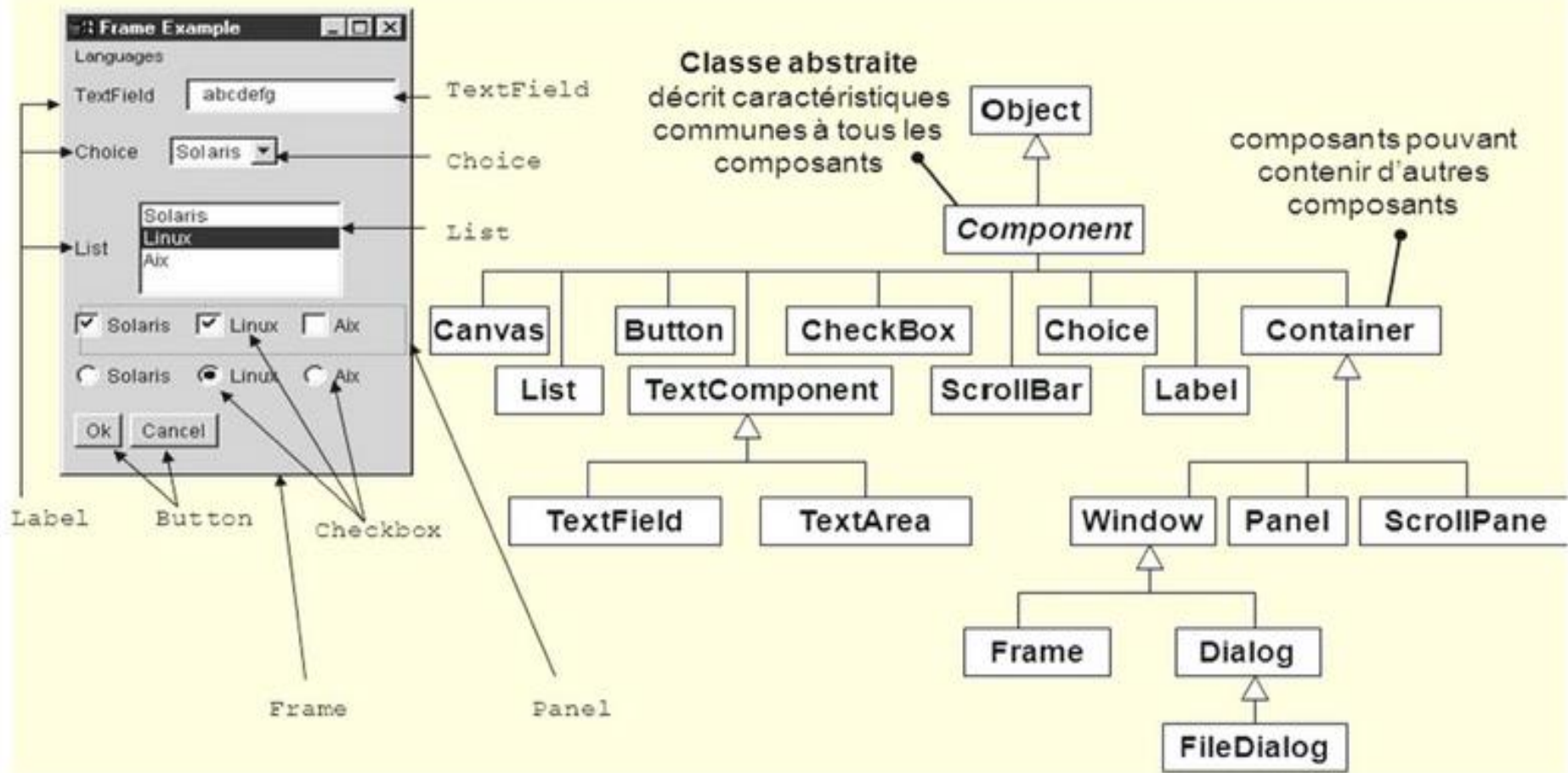
LES PACKAGES AWT

- Les classes du toolkit AWT permettent d'écrire des interfaces graphiques indépendantes du système d'exploitation sur lesquelles elles vont fonctionner.
- Cette librairie utilise le système graphique de la plateforme d'exécution (Windows, MacOS ou autre) pour afficher les objets graphiques.
- Le toolkit contient des classes décrivant les composants graphiques, les polices, les couleurs et les images.

LES PACKAGES AWT

- L'apparence des fenêtres et boutons diffère d'un système d'exploitation à l'autre, car chaque composant AWT est dessiné et contrôlé par un composant **tiers natif** spécifique au système d'exploitation.

LES PACKAGES AWT



Les packages SWING

LES PACKAGES SWING

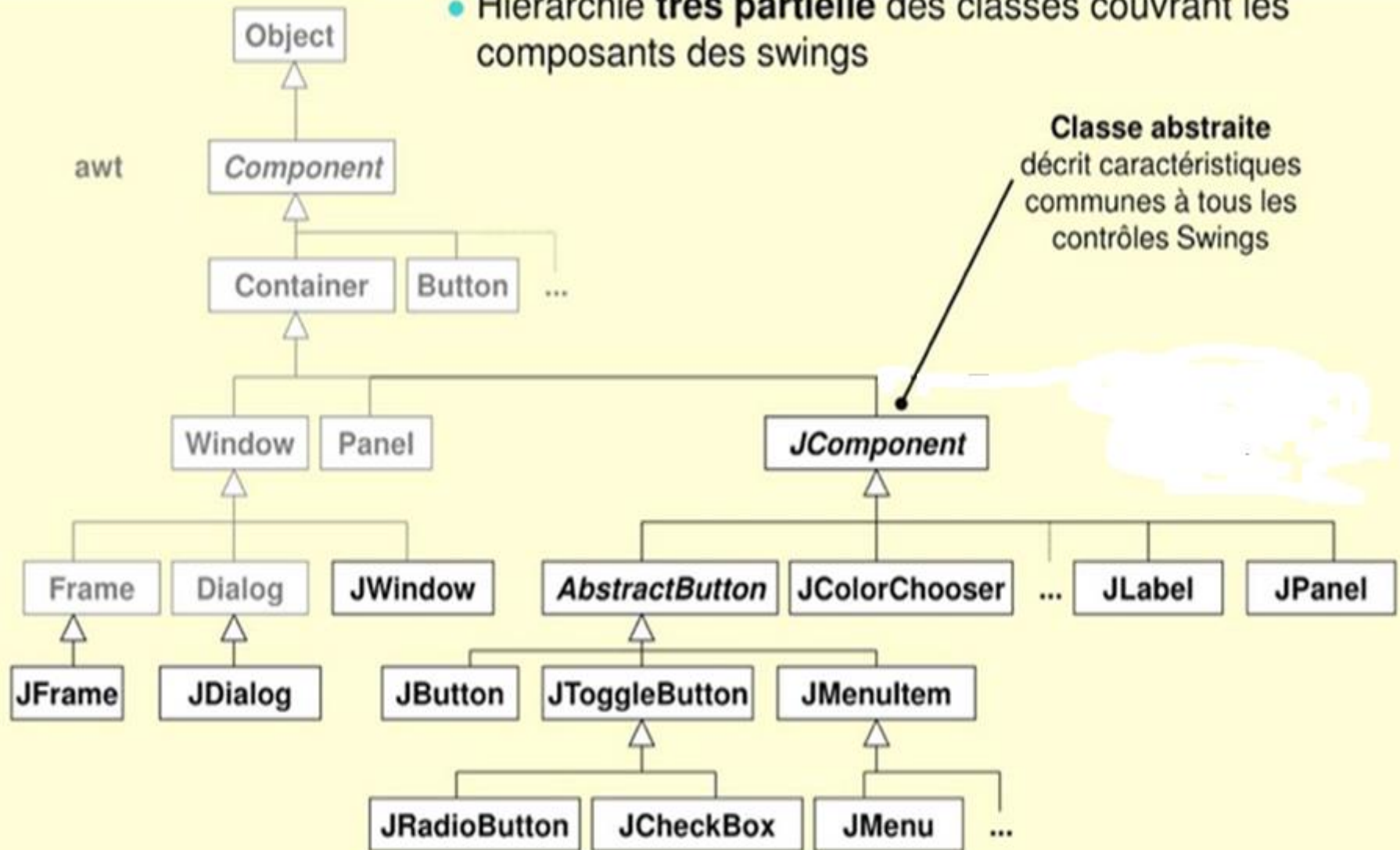
- Il existe un autre package d'interfaces graphiques et qui hérite de AWT appelée **Swing** qui est entièrement autonome, qui ne dépend pas du système d'exploitation.
- SWING ne requièrent pas d'allocation de ressources natives de la part du système d'exploitation, mais utilise les ressources de leurs ancêtres.

LES PACKAGES SWING

- Swing est une API dont le but est similaire à celle de l'AWT mais dont le mode de fonctionnement et d'utilisation est totalement différent.
- Swing a été intégrée au JDK depuis sa version 1.2. Cette bibliothèque existe séparément pour le JDK 1.1.
- La plupart des programmes utilisant Swing nécessite aussi l'importation de deux importants paquets de AWT :
 1. **`import java.awt.*;`**
 2. **`import java.awt.event.*;`**

LES PACKAGES SWING

- Hiérarchie **très partielle** des classes couvrant les composants des swings



LES PACKAGES SWING

- Les composants Swing forment une nouvelle hiérarchie parallèle à celle de l'AWT.
- Presque tous ces composants sont écrits en pure Java : ils ne possèdent **aucune partie native** **sauf** ceux qui assurent l'interface avec le système d'exploitation : JApplet, JDialog, JFrame, et JWindow.

AWT vs SWING

AWT vs SWING

- AWT utilise des composants natifs du système alors que Swing utilise des composants dessinés par Java.
- AWT est Lourd. Alors que les composants SWING sont généralement légers car ils n'ont pas besoin d'objets OS natifs pour implémenter leurs fonctionnalités.

AWT vs SWING

- Les composants AWT sont moins nombreux que les composants Swing. Alors que , les composants Java Swing sont bien plus nombreux.
- Les composants AWT ne suivent pas l'architecture MVC (Model View Controller). Tandis que, les composants Swing suivent le modèle MVC (Model View Controller).
- AWT signifie Abstract Windows Toolkit. Alors que, les composants Swing en Java sont également appelés JFC(Java Foundation Classes).

AWT vs SWING

- Les composants AWT ont besoin du package **java.awt**. Alors que, les composants Swing ont besoin du package **javax.swing**.
- Les composants AWT nécessitent et occupent un espace mémoire plus important. Tandis que, Les composants Swing n'occupent pas autant d'espace mémoire que les composants AWT.

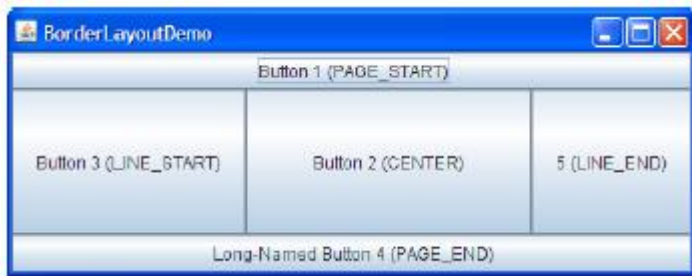
POUR INFORMATION ...

- il existe deux autres bibliothèques de mise en oeuvre d'IHM:
- 1. Avec l'apparition de Java 8 **JavaFX** est désormais l'API d'interface graphique principale du Java. Dans sa conception, elle est plus moderne que Swing et permet de produire des interfaces graphiques pouvant facilement être utilisées sur différents types d'écrans (écran standard de PC, smartphone & tablettes et applications Web).

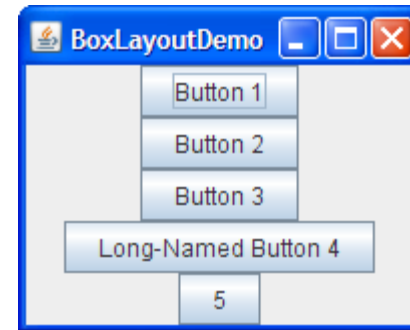
Architecture d'une application Swing

ARCHITECTURE D'UNE APPLICATION SWING

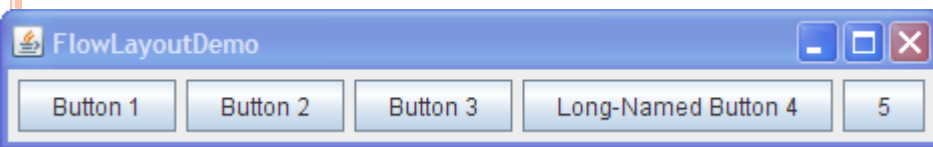
- Une application est un ensemble de fenêtres qui contient des items bien placées.
- Un conteneur (container) en top-level c'est le composant racine par exemple **JFrame**.
- JFrame contient d'autres composants qui peuvent être :
 - Composants atomiques (simples), par ex: un **bouton**.
 - Des composants intermédiaires qui permettent de diviser la fenêtre comme le **Jpanel** (des panneaux).
- Le placement des composants dans un conteneur correspond à une stratégie de placement, par exemple soit délégué à un *LayoutManager* (qui est interface).



BorderLayout : présentation avec bordures.



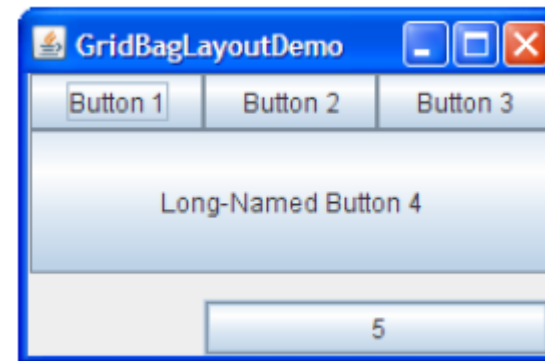
BoxLayout : en ligne ou en colonne



FlowLayout : présentation en file



GridLayout : en grille

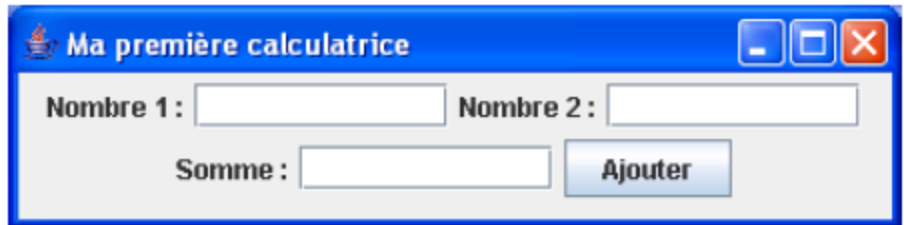


GridBagLayout : en grille composite qui est plus sophistiqué.

Cas d'études

CAS D'ÉTUDES

- Nous voulons créer une calculatrice simple capable d'ajouter deux nombres et d'afficher le résultat.



JFame: Ma première calculatrice

JPanel: contenuFenêtre

JLabel:
Nombre 1

JLabel:
Nombre 2

JLabel:
Somme

JButton:
lancer

JTextFie
Id :
entrée1

JTextFie
Id :
entrée2

JTextFie
Id :
résultat

- **import javax.swing.*;**
- **import java.awt.FlowLayout;**
- **public class CalculatriceSimple {**
- **public static void main(String[] args) {**
- // Crée la fenetre**
- **JFrame fenetre= new JFrame("Ma première calculatrice");**
- // Créer un panneau**
- **JPanel contenuFenêtre = new JPanel();**
- // Affecter un gestionnaire de disposition à ce panneau**
- **FlowLayout disposition = new FlowLayout();**
- **contenuFenêtre.setLayout(disposition);**
- // Créer les contrôles en mémoire**
- **JLabel label1 = new JLabel("Nombre 1 :");**
- **JTextField entrée1 = new JTextField(10);**
- **JLabel label2 = new JLabel("Nombre 2 :");**
- **JTextField entrée2 = new JTextField(10);**
- **JLabel label3 = new JLabel("Somme :");**
- **JTextField résultat = new JTextField(10);**
- **JButton lancer = new JButton("Ajouter");**

// Ajoute les contrôles au panneau

- contenuFenêtre.add(label1);
- contenuFenêtre.add(entrée1);
- contenuFenêtre.add(label2);
- contenuFenêtre.add(entrée2);
- contenuFenêtre.add(label3);
- contenuFenêtre.add(résultat);
- contenuFenêtre.add(lancer);

// ajouter le panneau dans la fenetre

- fenetre.setContentPane(contenuFenêtre);

// Positionner les dimensions et rend la fenêtre visible

- fenetre.setSize(400,100);
- fenetre.setVisible(**true**);} }

Classes de base

