

## Chapitre 2 : Programmation graphique

RT2

---

# Plan

---

- ▶ Introduction : AWT & Swing
- ▶ Fenêtre et cadre de fenêtre : JFrame, Jpanel

# Pourquoi une interface Utilisateur?

---

- ▶ Une interface Utilisateur permet de :
  - Accepter les entrées de l'utilisateur
  - Traiter les entrées
  - Produire des sorties
- ▶ Une interface graphique doit être :
  - **Convivialité** : intuitives les utiliser sans apprendre un langage de codage ou des commandes informatiques.
  - **Efficacité** : effectuer des tâches plus rapidement et plus facilement.
  - **Clarté** : fournir aux utilisateurs un retour visuel pour indiquer si leurs actions sont réussies ou non.
  - **Esthétique** : créer une expérience utilisateur agréable.
  - **Accessibilité** : accessibles aux utilisateurs handicapés et limités.

# Types d'interface utilisateur

- Interface utilisateur basée sur des caractères : C'est une interface où l'entrée de l'utilisateur et la sortie du programme sont sous forme de texte
- Une interface graphique : Elle permet aux utilisateurs d'interagir avec un programme à l'aide d'éléments graphiques appelés contrôles ou widgets en utilisant un clavier ou une souris.

```
Please enter person details
Name:John Jacobs
DOB:01/12/1971
Enter S to Save and E to exit without saving:S
```



The image shows a graphical user interface window titled "Person Details". It contains two input fields: "Name:" with the text "John Jacobs" and "DOB:" with the text "01/12/1971". To the right of the Name field is a "Save" button, and below it is a "Close" button. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

# Bibliothèques JAVA : AWT & SWING

---

- ▶ Pour programmer des IHM on peut soit :
  - ▶ Utiliser un concepteur GUI visuel : Drag & drop (SwisiDad, **WindowBuilder Pro**, etc).
  - ▶ Écrire manuellement tout le code de l'objet graphique.
- ▶ Pour programmer des IHM, on utilise principalement les packages `javax.swing` et `java.awt`

# Bibliothèques JAVA : AWT & SWING

## AWT : Abstract Window Toolkit

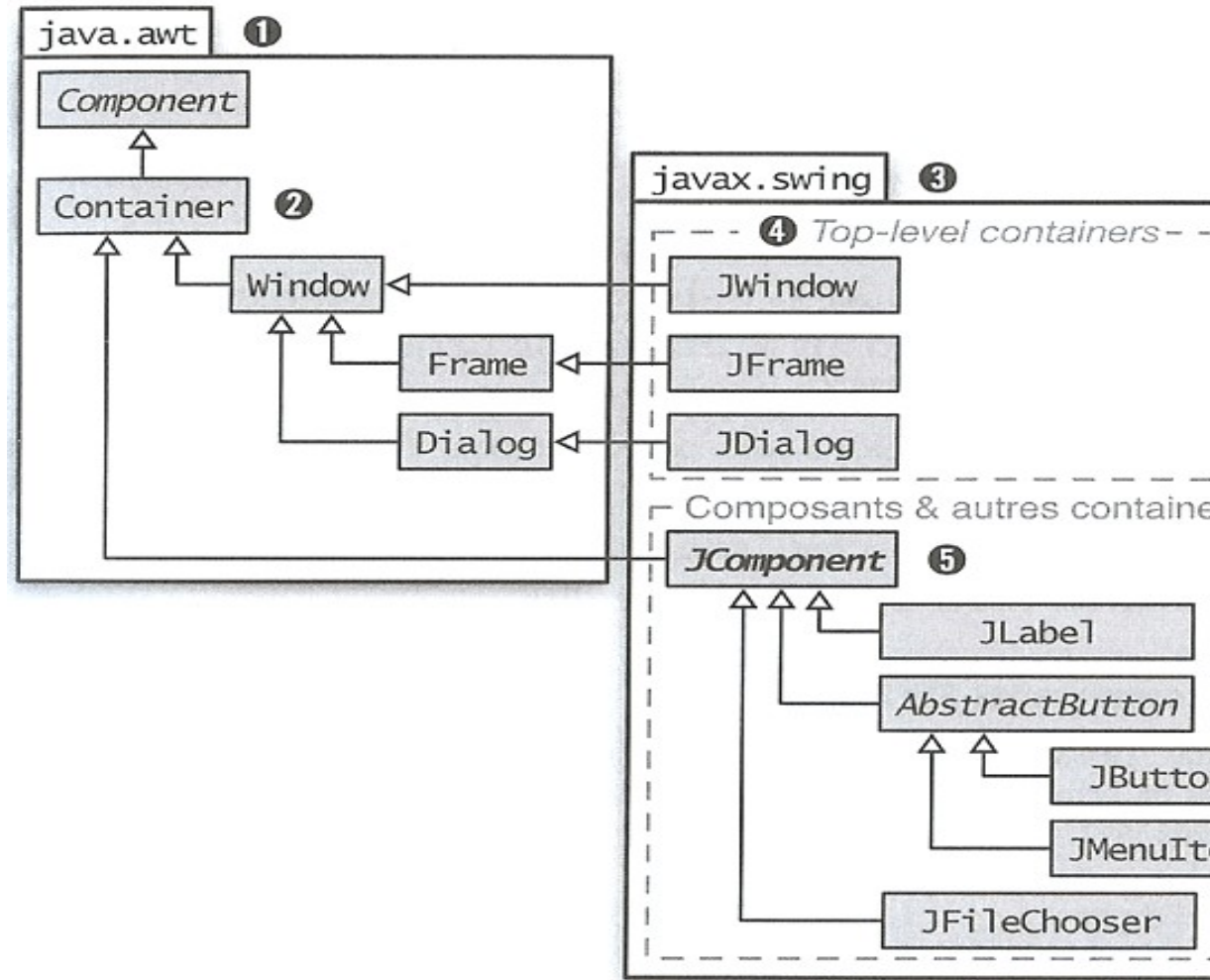
- Package apparu dans la version java 1.0
- Boîte à outils **portable** : le code peut être écrit sous Windows, puis exécuté sur un autre système Linux ou Macintosh.
- Les composants awt sont **HeavyWeight** : ils sont fortement liés au système d'exploitation.

## Swing

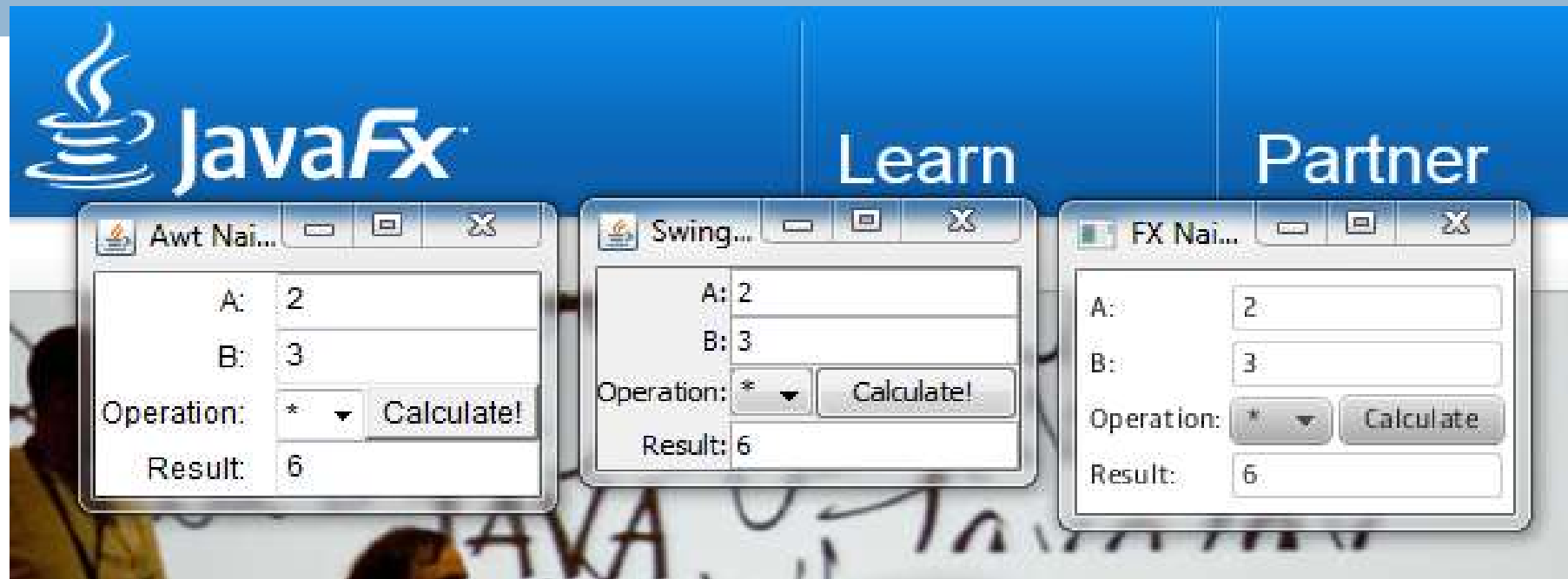
- Package apparu dans la version de java 1.2 (Java2)
- Les composants graphiques de Swing sont implémentés directement dans Java  
→ Indépendance de la plateforme utilisée, un bouton Swing a toujours la **même apparence**.
- Les composants Swing sont dessinés dans **un conteneur** et ils sont dit **LightWeight**.
- Il existe également d'autres différences, comme le nombre de composants utilisables, la gestion des bordures...




# Bibliothèques JAVA : AWT & SWING

- ▶ AWT et Swing sont **complémentaires**.
  - Il est préférable d'utiliser **Swing**.
  - On utilise certains éléments d'AWT comme la classe **Color** et la gestion des **événements**, ainsi que quelques autres éléments.



# AWT / Swing / JAVAFX (JAVA8)

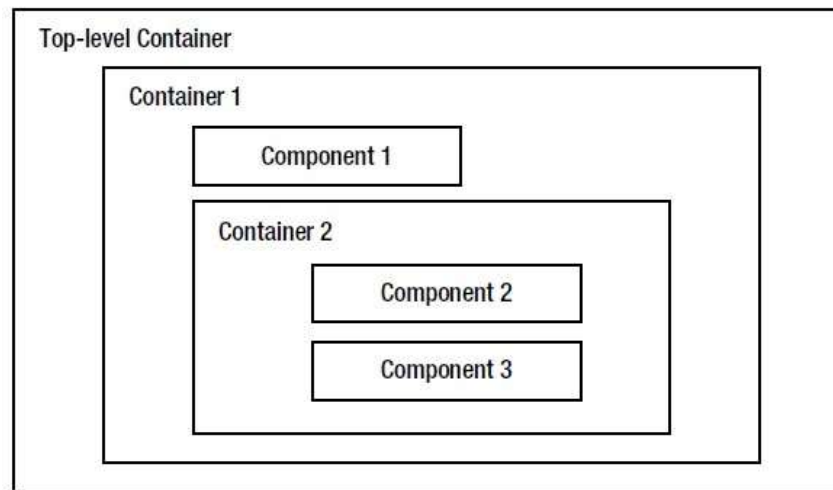


Process		CPU	Private Bytes	Working Set	Virtual Size
Swing	 java.exe	0.09	106,004 K	56,012 K	558,808 K
AWT	 java.exe	0.08	75,276 K	36,120 K	528,528 K
FX	 java.exe	0.32	95,560 K	76,688 K	529,088 K



# La hiérarchie des conteneurs dans SWING

- ▶ Un **conteneur** est un composant qui peut contenir d'autres composants à l'intérieur.
- ▶ Un **conteneur de niveau supérieur** : c'est un conteneur au niveau le plus élevé (Ex : JFrame, un JDialog, un JWindow )
- ▶ Un JPanel est un exemple de **conteneur simple**.
- ▶ Un JButton, un JTextField, etc. sont des exemples de **composants**.



# La hiérarchie des conteneurs dans SWING

---

- ▶ Dans une application Swing, chaque **composant** doit être contenu dans un **conteneur**.
- ▶ Le conteneur est considéré comme le parent du composant et le composant est considéré comme l'enfant du conteneur.
- ▶ Pour afficher un composant à l'écran, un conteneur de niveau supérieur doit se trouver à la racine de la hiérarchie des conteneurs.
- ▶ Chaque application Swing doit avoir **au moins un conteneur** de niveau supérieur.

# L'objet JFrame

- Pour créer une fenêtre de type JFrame :

```
import javax.swing.JFrame;
public class Test {
    public static void main(String[] args) {
        JFrame fenetre = new JFrame();
    }
}
```

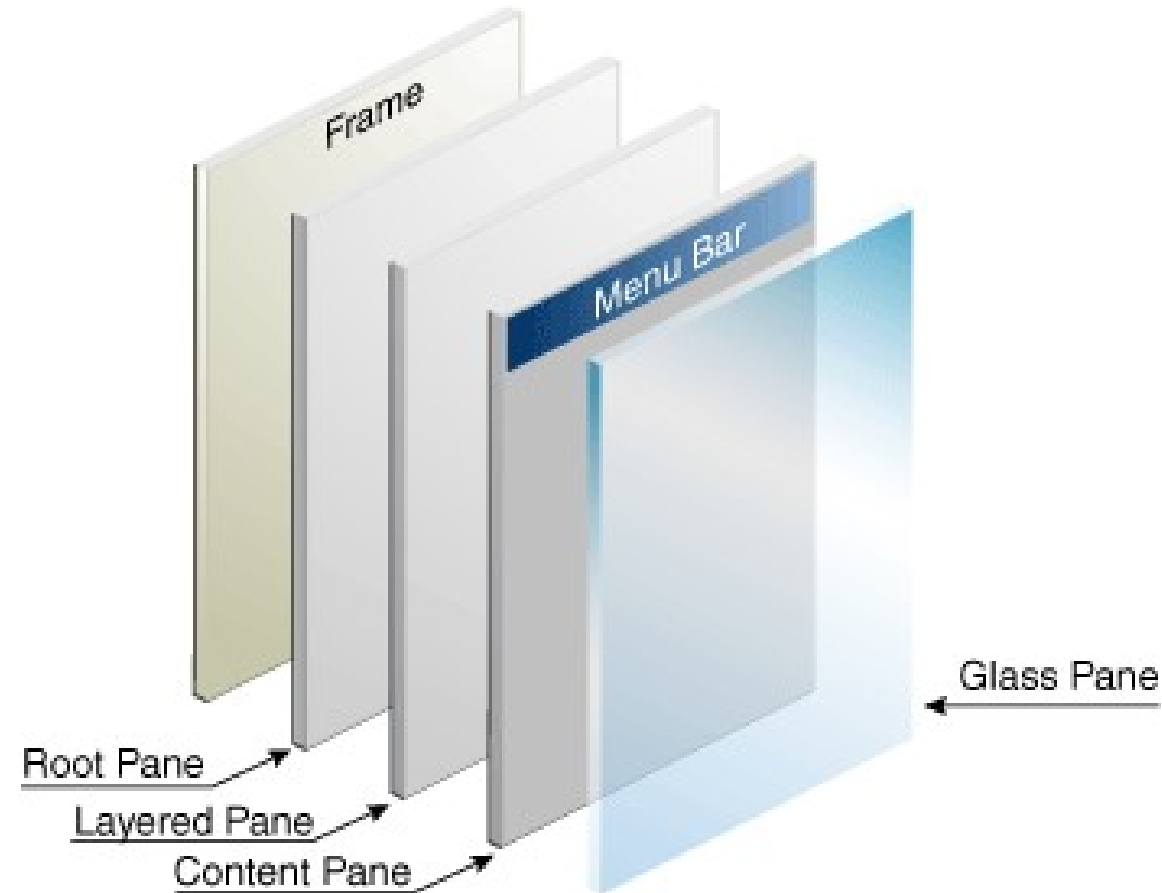
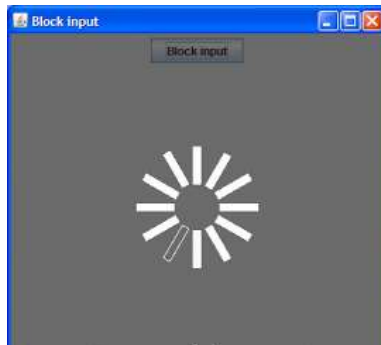
→ Lorsqu' on exécute ce code, on n'obtient rien, car par défaut, JFrame est **invisible**.

- Pour le rendre visible, on doit ajouter :

```
fenetre.setVisible(true);
```

# Structure du JFrame

- **LayeredPane** : contient des composants (ContentPane, JMenuBar, popup, palette) selon un ordre Z spécifique.
- **ContentPane** : Dans lequel on place des composants graphiques visibles.
- **GlassPane** : Permet d'intercepter des événements ou dessiner sur une zone qui contient déjà un ou plusieurs composants..



# Structure du JFrame

---

- ▶ La couche la plus importante est celle de **ContentPane**.
- ▶ Pour la manipuler, on fait appel aux méthodes **setContentPane()** et **getContentPane()** de la classe **JFrame**.
- ▶ Pour insérer des composants graphiques, on peut utiliser un autre composant **JPanel**.
- ▶ Il existe d'autres types de fenêtre :
  - ▶ **JWindow** : un **JFrame** sans bordure et non draggable (déplaçable).
  - ▶ **JDialog** : une fenêtre non redimensionnable

# L'objet JPanel

- ▶ JPanel est un composant de type conteneur qui permet d'accueillir d'autres objets de même type ou des objets de type composant (boutons, cases à cocher...).
- ▶ La démarche à suivre :
  1. Importer la classe **javax.swing.JPanel** dans le classe héritée de JFrame.
  2. Instancier un **JPanel** puis lui spécifier une couleur de fond pour mieux le distinguer.
  3. Avertir le JFrame que ça sera le JPanel qui constituera son **ContentPane**.

# L'objet JPanel

```
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Fenetre extends JFrame {
    public Fenetre(){
        this.setTitle("Ma première fenêtre Java");
        this.setSize(400, 100);
        this.setLocationRelativeTo(null);

        //Instanciation d'un objet JPanel
        JPanel pan = new JPanel();
        //Définition de sa couleur de fond
        pan.setBackground(Color.ORANGE);
        //On prévient le JFrame que le JPanel sera son ContentPane
        this.setContentPane(pan);      //this.getContentPane().add(pan) ;
        this.setVisible(true);
    }
}
```

