

# Introduction à la programmation et à la modélisation en Java

Introduction aux interfaces graphiques via Swing

C.L.I. : Commande Line Interface. Le programme a la main et sollicite l'utilisateur.  
G.U.I. : Graphical User Interface. L'utilisateur a la main et pilote de le programme. Ce dernier réagit aux évènements.

Notion : Programmation Evènementielle.

API (Application Programming Interface) :

- AWT : Abstract Window Toolkit

- Swing

- JavaFX

Documentation de l'API :

- [Jframe](#) (Attention JDK11)

- [Tutoriel swing complet](#)

## Partie 1 : Fenêtre

Première fenêtre :

- Déplacer

- Redimensionner

- Réduire

- Fin : CTRL+C ou Fermer la console (Notion cachée : Thread)

Source : AppGUI00

Seconde fenêtre personnalisée : (Notion cachée : Héritage, Mot-clé : extends)

- Fermeture « propre » : EXIT\_ON\_CLOSE (Notion cachée : Constante de Classe, Mots-clés : final, static)

- Positionnement et dimensionnement : setBounds (Attention à l'orientation de l'écran)

- Dimension : getSize

Source : AppGUI01

## Partie 2 : Évènement

Notion : Écouteur (listener)

Exemple : L'utilisateur clic dans la fenêtre principale

- L'évènement (clic) a une source : la fenêtre principale.
- L'évènement (clic) a une catégorie : souris.
- Pour détecter les évènements survenant dans la fenêtre principale, il faut attacher un écouteur (d'évènements de souris) à cette fenêtre principale. (Notion cachée : Interface)

Source : AppGUI02

Remarque : L'implémentation de l'écouteur est longue pour peu de résultat.

Solution : Utiliser un adaptateur.

Source : AppGUI03

Mécanisme général :

L'utilisateur déclenche un évènement : Mouse, Key, Xxx (catégorie de l'évènement)

Cet évènement survient relativement à une source : Fenêtre, Bouton, etc.

Un écouteur ou adaptateur doit être attaché à la source pour traiter l'évènement : MouseListener, MouseAdapter, KeyListener, XxxAdapter.

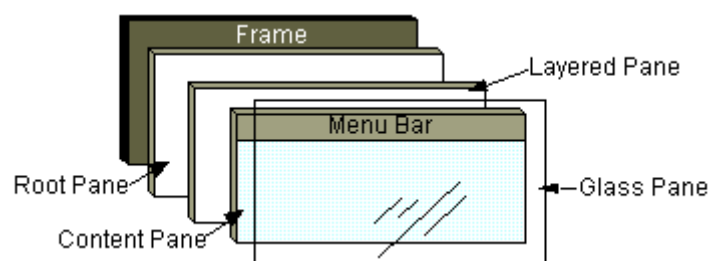
Remarque : Les évènements « souris » et « clavier » sont dits « de bas niveau » (liés directement au matériel).

## Partie 3 : Composants

Objet : JButton

La classe JFrame est composée de beaucoup d'éléments, dont un conteneur dans lequel insérer les composants

Il faut également mettre en forme le bouton dans le cadre principal.



Notion : Gestionnaire de mise en forme ou gestionnaire de disposition (Layout Manager)

Ce gestionnaire de mise en forme est associé à l'objet contenant les composants.

Notion : Évènement sémantique (« haut niveau »)

Le composant « Bouton » peut lui aussi (comme la souris) déclencher un évènement de la catégorie « Action ».

Source : AppGUI04

Méthodes supplémentaires :

getSource : un écouteur peut écouter plusieurs composants et donc il est possible d'avoir une seule méthode gérant plusieurs évènements. Cette méthode permet d'identifier le composant à l'origine de l'évènement.

getActionCommand : comme getSource mais uniquement pour les évènements « Action » et le nom du bouton (texte affiché) est utilisé plutôt que la référence de l'objet.

Problème « classique » : la communication entre les composants. L'écouteur d'un bouton peut avoir besoin d'informations sur la fenêtre ; il suffit alors d'ajouter la fenêtre dans le constructeur du bouton.

Notion : Dynamique. Il est possible de créer, détruire, etc. des composants au fur et à mesure de l'utilisation du programme.

#### **Partie 4 : Dessin**

Technique : ne pas dessiner directement dans le cadre principale (Jframe) mais plutôt dans un panneau (JPane) contenu dans le cadre principal (au même titre que tous les composants). Explication longue omise.

Source : AppGUI05

Technique : redéfinir la méthode paintComponent. Explication longue omise.  
Notion cachée : contexte graphique (Graphics).

Source : AppGUI06

Méthode : repaint() pour forcer la mise à jour d'un dessin.

Pour aller plus loin (1) : AppGUI07. L'action sur le bouton déclenche un dessin. Étudiez attentivement la communication entre les objets.

Pour aller plus loin (2) : AppGUI08. Comme dans la version précédente mais cette fois le dessin est volatile ; si la fenêtre est redessinée (redimensionnée par exemple) le dessin disparaît.

Pour aller plus loin (3) : [E. Lecolinet] Programmation événementielle et interfaces graphiques Java Swing.pdf