

Programmation Graphique - SWING

©peio Loubière



18 novembre 2009

- 1 Sommaire
- 2 Programmation Graphique : swing
 - swing
- 3 MVC
- 4 Plus

SWING

Présentation

- API similaire à AWT, sous paquetage de javax
- possède ses propres composants graphiques ayant des fonctionnalités étendues (tooltip, images en icône, onglets, tableaux...)
- codés entièrement en Java, correspondance avec AWT (Jframe, JPanel, JDialog...)
- une apparence modifiable, même apparence suivant chaque SE
- gestion d'évènements identique à AWT, évènements particuliers dans le paquetage : `javax.swing.event`

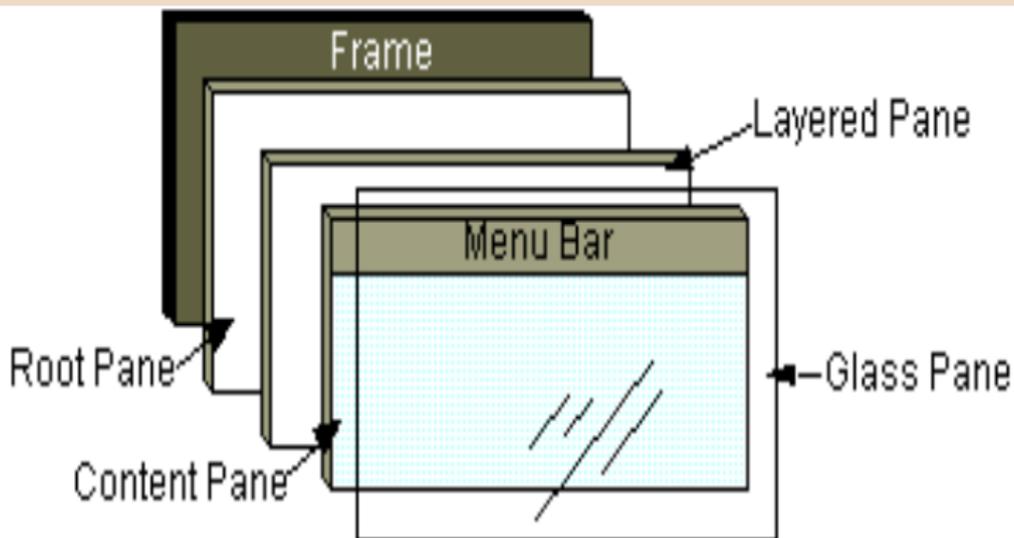
SWING

Différences

- la classe JFrame est l'équivalent de la classe Frame
- la plupart des composants AWT ont un équivalent Swing
- utilise le double-buffering (meilleur affichage)
- Il n'existe pas d'équivalent de Canvas, on utilise le JPanel
- gestion différente du contenu d'un panel : JRootPane

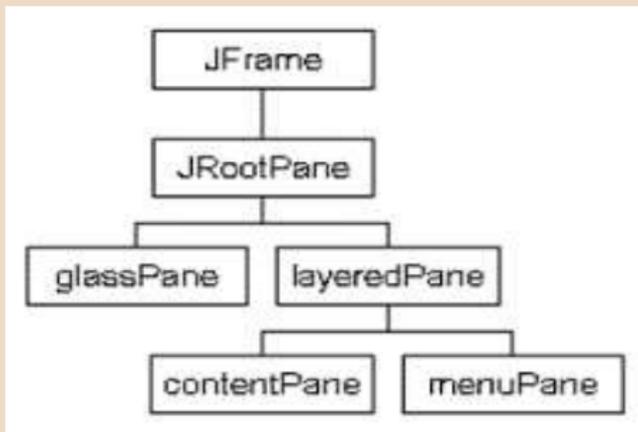
SWING

Détail du JPanel



SWING

Détail du JPanel



SWING

Détail du JPanel

- **JRootPane** : panel racine, se compose du layeredPane et du glassPane
- **LayeredPane** : se compose du contentPane et du menuBar
- **ContentPane** : un JPanel par défaut dont le gestionnaire de présentation est un BorderLayout, c'est celui qui contient les composants ou dans lequel on dessine par défaut. Ce panel peut être remplacé par n'importe quel composant grâce à la méthode `setContentPane()`
- **MenuBar** : un objet de type `JMenuBar` contient, éventuellement, un ensemble de menus
- **GlassPane** : JPanel transparent qui se situe au dessus du layeredPane. Le glassPane peut être n'importe quel composant : pour le modifier, utiliser la méthode `setGlassPane()`

Fermeture

Comportement

- simplification du comportement de la croix dans les fenêtres
- méthode `setDefaultCloseOperation()`
- Paramètre :
 - `WindowConstants.DISPOSE_ON_CLOSE` : détruit la fenêtre
 - `WindowConstants.DO_NOTHING_ON_CLOSE` : désactive la croix
 - `WindowConstants.HIDE_ON_CLOSE` : cache la fenêtre

Panneaux spéciaux

- **JScrollPane** : permet d'obtenir des ascenseurs
- **JSplitPane** : permet de diviser le panel en 2
- **JTabbedPane** : permet d'avoir des panels sous forme d'onglet

Boutons

Nouveautés

- **JButton** : peut contenir une icône au format gif
- **JToggleButton** : bouton qui reste enfoncé sur le clic
- **ButtonGroup** : garantit qu'un seul bouton du groupe peut être sélectionné
- **JRadioButton** : bouton radio, s'utilise avec un groupe

Plus de composants

- **JFileChooser** : choisir des fichiers
- **JColorChooser** : choisir des couleurs
- **JEditorPane** : panel qui permet un affichage avec de la mise en forme, notamment HTML
- **JPasswordField** : champ de mot de passe
- **JProgressBar** : barre de progression
- ...

Architecture

Présentation

Swing est basé sur une architecture MVC :
Modèle-Vue-Contrôleur

- MVC : résout la plupart des problèmes concernant une interface graphique
- décrit un cadre de développement
- sépare les données, les traitements et leur représentation graphique
- la vue et le contrôleur sont souvent regroupés dans les composants swing

Architecture

Le modèle

- offre des méthodes de gestion et de manipulation des données

Architecture

La vue

- interface à proprement parler
- présente les données récupérées par le modèle
- n'effectue aucun traitement sinon de modification de vue
- réceptionne les interactions avec l'utilisateur, les envoie au contrôleur

Architecture

Le contrôleur

- se charge du traitement des événements
- dialogue avec le modèle si modification de données
- dialogue avec la vue si modification de visualisation

Architecture

Application à Swing

- chaque élément graphique, vue, est couplé d'un modèle qui gère les données (ex : ButtonModel pour JButton, ListModel pour JList...)
- les classes Adapter ou interfaces Listener s'occupent de contrôler les évènements
- le contrôleur et la vue étant souvent associés

Plus

- SWT (Standard Widget Toolkit) librairie graphique plus légère que Swing (entre AWT et Swing) développée par Eclipse
- JFace librairie graphique s'appuyant sur SWT, plus riche
- Framework de développement d'application MVC
 - Eclipse (évidement !)
 - struts (Apache)
 - JSF (Java Server Faces)