

Exemple d'application JAVA Swing

Swing est une boîte à outils de widgets graphiques pour Java. Il fait partie de Java Foundation Classes (JFC) d'Oracle, une API fournissant une interface graphique aux programmes Java. Swing a été développé pour fournir un ensemble de composants d'interface graphique plus sophistiqué que la version précédente de Abstract Window Toolkit (AWT). JAVA fournit un riche ensemble de bibliothèques pour créer une interface utilisateur graphique indépendante de la plate-forme.

1. Introduction

L'API Swing est un ensemble de composants d'interface graphique extensibles destinés à faciliter la vie des développeurs en créant des applications frontales / graphiques basées sur JAVA. Il repose sur l'API AWT et remplace l'API AWT car il possède presque tous les contrôles correspondant aux contrôles AWT. Le composant Swing suit une architecture modèle-vue-contrôleur pour répondre aux critères suivants.

- Une seule API doit être suffisante pour prendre en charge plusieurs aspects.
- L'API doit être guidée par le modèle afin que l'API de niveau supérieur ne soit pas obligée d'avoir les données.
- L'API consiste à utiliser le modèle Java Bean afin que les outils de génération et l'EDI puissent fournir de meilleurs services aux développeurs afin qu'ils puissent l'utiliser.

2. JAVA Swing

Contrairement à AWT, Java Swing fournit des composants légers et indépendants de la plate-forme.

Le package `javax.swing` fournit des classes pour l'API java swing telles que `JButton`, `JTextField`, `JTextArea`, `JRadioButton`, `JCheckbox`, `JMenu`, `JColorChooser`, etc.

2.1 Architecture MVC

L'architecture de l'API Swing suit l'architecture MVC à base lâche de la manière suivante.

- Un modèle représente les données du composant.
- La vue représente une représentation visuelle des données du composant.
- Le contrôleur prend l'entrée de l'utilisateur dans la vue et reflète les modifications apportées aux données de Component.
- Les composants Swing ont Model en tant qu'élément séparé et les composants View et Controller sont superposés dans les éléments de l'interface utilisateur. En utilisant cette manière, Swing a une architecture de look-and-feel enfichable.

Chaque interface utilisateur prend en compte les trois principaux aspects suivants:

- **Éléments d'interface utilisateur:** Il s'agit des éléments visuels de base que l'utilisateur voit et interagit éventuellement. SWT fournit une vaste liste d'éléments communs et largement utilisés, allant du plus simple au plus complexe.
- **Layouts:** Ils définissent la manière dont les éléments de l'interface utilisateur doivent être organisés à l'écran et donnent une apparence finale à l'interface graphique (GUI).
- **Comportement:** il s'agit d'événements qui se produisent lorsque l'utilisateur interagit avec des éléments de l'interface utilisateur.

2.2 Caractéristiques de la balançoire

- **Poids léger:** les composants Swing sont indépendants de l'API du système d'exploitation natif. En effet, les contrôles des API Swing sont rendus principalement à l'aide de code JAVA pur au lieu d'appels du système d'exploitation sous-jacent.
- **Contrôles enrichis -** Swing fournit un riche ensemble de contrôles avancés, tels que les contrôles Tree, TabbedPane, Slider, ColorPicker et Table.
- **Hautement personnalisable -** Les commandes Swing peuvent être personnalisées très facilement, l'apparence visuelle étant indépendante de la représentation interne.
- **Apparence enfichable:** l'interface graphique basée sur SWING peut être modifiée au moment de l'exécution en fonction des valeurs disponibles.

2.3 Configuration

Editeurs Java populaires:

Pour écrire vos programmes Java, vous aurez besoin d'un éditeur de texte. Il existe des IDE encore plus sophistiqués disponibles sur le marché. Mais pour l'instant, vous pouvez envisager l'un des éléments suivants:

- **Bloc-notes:** sur un ordinateur Windows, vous pouvez utiliser n'importe quel éditeur de texte tel que le Bloc-notes TextPad.
- **NetBeans:** est un IDE Java open source et gratuit pouvant être téléchargé à partir de <http://www.netbeans.org/index.html>.
- **Eclipse:** est également un IDE java développé par la communauté open source eclipse et pouvant être téléchargé à partir de <http://www.eclipse.org>.

2.4 Classe et description

- **Composant:** un composant est la classe de base abstraite pour les commandes d'interface utilisateur autres que les menus de SWING. Le composant représente un objet avec une représentation graphique.
- **Conteneur:** un conteneur est un composant pouvant contenir d'autres composants SWING.
- **JComponent:** un JComponent est une classe de base pour tous les composants d'interface utilisateur swing. Pour utiliser un composant swing héritant de JComponent, il doit figurer dans une hiérarchie de confinement dont la racine est un conteneur Swing de niveau supérieur.

2.4 Éléments d'interface utilisateur SWING

- JLabel Un objet JLabel est un composant permettant de placer du texte dans un conteneur.
- JButton Cette classe crée un bouton étiqueté.
- JColorChooser Un JColorChooser fournit un panneau de commandes conçu pour permettre à un utilisateur de manipuler et de sélectionner une couleur.
- Boîte de contrôle JCheck Un JCheckBox est un composant graphique pouvant être activé (true) ou désactivé (false).
- JRadioButton La classe JRadioButton est un composant graphique pouvant être activé (true) ou désactivé (false). dans un groupe.
- JList Un composant JList présente à l'utilisateur une liste déroulante d'éléments de texte.
- JComboBox Un composant JComboBox offre à l'utilisateur un menu de choix.
- JTextField Un objet JTextField est un composant texte qui permet de modifier une seule ligne de texte.
- JPasswordField Un objet JPasswordField est un composant texte spécialisé dans la saisie d'un mot de passe.
- JTextArea Un objet JTextArea est un composant texte qui permet la modification de plusieurs lignes de texte.
- ImageIcon Un contrôle ImageIcon est une implémentation de l'interface Icon qui peint des icônes à partir d'images.