Swing

m2iformation.fr

Sommaire

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Introduction à Swing

Qu'est ce que swing

- Swing est une bibliothèque graphique pour la création d'interfaces utilisateur dans les applications Java.
- Elle permet aux développeurs de créer des interfaces graphiques conviviales, interactives et esthétiques pour leurs applications.
- Swing fait partie de la plateforme Java Standard Edition (Java SE) et offre un ensemble complet de composants graphiques personnalisables et extensibles, permettant de créer des fenêtres, des boutons, des champs de texte, des tableaux, des listes, des menus, et bien plus encore.

Les caractéristiques clés de Swing

1 - Indépendance de la plateforme : Les interfaces créées avec Swing ont une apparence et un comportement cohérents sur différentes plateformes, car elles ne dépendent pas des éléments graphiques natifs du système d'exploitation.

2 - Composants riches : Swing offre une vaste gamme de composants graphiques, allant des éléments de base comme les boutons et les champs de texte aux composants plus avancés comme les arbres et les onglets.

Les caractéristiques clés de Swing

3 - Personnalisation : Les composants Swing sont hautement personnalisables. Les développeurs peuvent contrôler l'apparence, la taille, la couleur, la police et d'autres propriétés des composants pour correspondre aux besoins de conception de leur application.

4 - Événements et interactions utilisateur : Swing intègre un mécanisme d'écouteurs d'événements qui permet aux développeurs de répondre aux actions de l'utilisateur, telles que les clics de souris et les frappes de clavier.

Les caractéristiques clés de Swing

5 - Extensibilité : En plus des composants fournis par défaut, les développeurs peuvent créer leurs propres composants personnalisés en étendant les classes de base Swing.

6 - Gestion de la mise en page : Swing offre plusieurs gestionnaires de mise en page (layouts) pour organiser les composants dans une fenêtre de manière flexible et cohérente.

Les caractéristiques clés de Swing

7 - Look and Feel : Swing propose différents "look and feels" (apparences) qui permettent de changer l'apparence générale de l'interface sans changer le code de l'application.

8 - Séparation de la logique métier et de l'interface : Swing permet de séparer la logique de l'interface utilisateur de la logique métier, ce qui favorise une meilleure organisation du code.

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

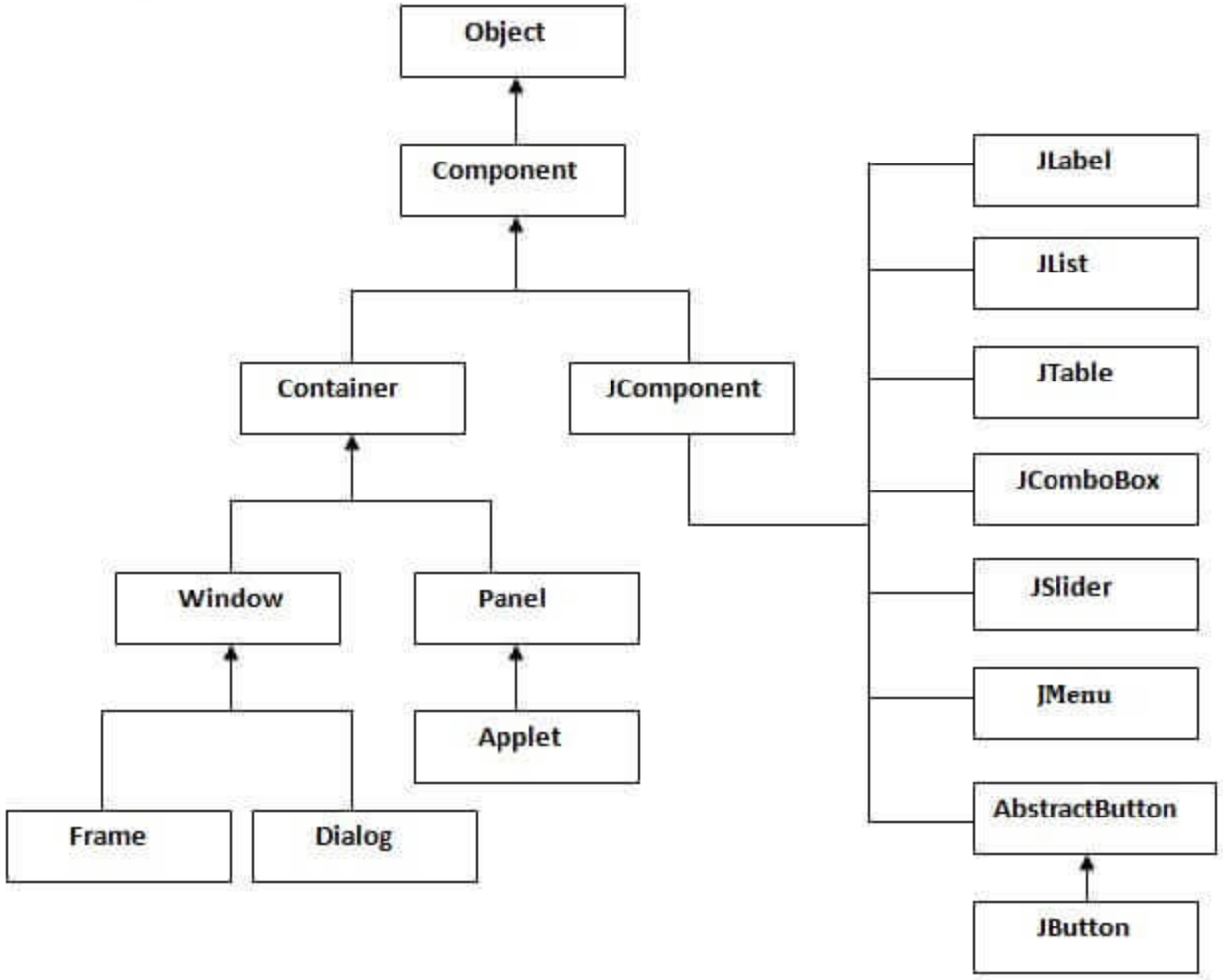
Composants Swing de base

(Container et Component)

Définition Composant Swing

Dans Swing, les termes "containers" et "components" font référence à deux types fondamentaux d'éléments qui sont utilisés pour créer des interfaces graphiques.

Définition Composant Swing



Définition Composant Swing

- **Containers** : Les containers sont des composants qui sont utilisés pour contenir d'autres composants. Ils sont responsables de l'organisation et de la disposition des composants à l'intérieur d'eux. Les containers peuvent eux-mêmes être contenus à l'intérieur d'autres containers, formant ainsi une hiérarchie de mise en page. Les containers Swing sont des classes telles que **JFrame**, **JPanel**, **JScrollPane**, **JTabbedPane**, etc.
- *Les containers gèrent la manière dont les composants sont disposés, alignés et dimensionnés les uns par rapport aux autres.*

Définition Composant Swing

- **Components** : Les components (ou composants) sont les éléments graphiques individuels qui sont affichés à l'écran et avec lesquels les utilisateurs interagissent. Ils sont les éléments de base qui composent une interface utilisateur Swing. Les composants peuvent être des boutons, des zones de texte, des étiquettes, des cases à cocher, des boutons radio, des listes déroulantes, des images, etc.
- *Chaque type de composant a des fonctionnalités et des comportements spécifiques. Les composants sont des classes comme* ***JButton***, ***JTextField***, ***JLabel***, ***JCheckBox***, ***JComboBox***, *etc.*

Les Containers

- Les conteneurs sont des composants spéciaux qui servent de réceptacles pour d'autres composants.
- Ils sont utilisés pour organiser, grouper et disposer d'autres composants de manière structurée sur l'interface utilisateur.
- Les conteneurs sont responsables de la gestion de la disposition (layout) et de la présentation des composants qu'ils contiennent.
- Les conteneurs peuvent également être imbriqués pour créer des structures complexes d'interfaces utilisateur.

Les Components

- Les composants sont les éléments graphiques de base qui constituent l'interface utilisateur.
- Ils sont les éléments visibles tels que les boutons, les zones de texte, les images, les cases à cocher, les listes, etc.
- Les composants sont placés à l'intérieur des conteneurs pour former l'interface utilisateur finale.
- Chaque composant possède ses propres propriétés, méthodes et événements qui lui sont spécifiques.
- Les composants peuvent être interagissants (comme les boutons) ou statiques (comme les étiquettes).

Exemples containers

Il existe plusieurs types de conteneurs dans Swing, chacun ayant ses propres caractéristiques et utilités :

1 - JFrame : Un conteneur de niveau supérieur qui représente une fenêtre. Il peut contenir d'autres composants et est souvent utilisé comme fenêtre principale d'une application.

2 - JPanel : Un conteneur générique qui peut contenir d'autres composants. Les panneaux sont souvent utilisés pour organiser des groupes de composants dans des zones spécifiques de l'interface.

3 - JDialog : Un conteneur similaire à JFrame, mais généralement utilisé comme boîte de dialogue modale ou non modale.

Exemples containers

4 - JScrollPane : Un conteneur qui permet de faire défiler son contenu lorsque l'espace disponible est insuffisant pour afficher tous les composants.

5 - JTabbedPane : Un conteneur qui affiche plusieurs onglets, chacun contenant un contenu différent. Il est utile pour afficher différentes vues ou panneaux dans une même zone.

6 - JSplitPane : Un conteneur qui divise la zone en deux parties redimensionnables, généralement utilisé pour diviser une interface en deux zones ajustables.

7 - JMenuBar, JMenu et JMenuItem : Des conteneurs spécifiques pour la création de menus et de sous-menus dans la barre de menus

Exemples components

Voici quelques-uns des composants les plus couramment utilisés dans Swing :

1- JLabel : Affiche du texte ou une image sans possibilité d'interaction de l'utilisateur.

2 - JButton : Un bouton cliquable qui peut déclencher une action lorsque l'utilisateur clique dessus.

3 - JTextField : Une zone de texte simple où l'utilisateur peut entrer du texte.

4 - JTextArea : Une zone de texte multiligne qui peut afficher et accepter du texte sur plusieurs lignes.

Exemples components

5 - JRadioButton : Un bouton d'option qui fait partie d'un groupe et permet à l'utilisateur de sélectionner une option parmi plusieurs.
6 - JComboBox : Une liste déroulante qui permet à l'utilisateur de sélectionner une valeur parmi plusieurs options.
7 - JList : Une liste qui affiche plusieurs éléments, généralement utilisée pour afficher des listes simples ou multiples.

Exemples components

8 - JTable : Un composant pour afficher des données sous forme de tableau, généralement utilisé pour afficher des données tabulaires.
9 - JScrollPane : Un composant qui permet de faire défiler le contenu d'un autre composant lorsque l'espace disponible est insuffisant.
10 - JPanel : Un conteneur générique qui peut contenir d'autres composants et est souvent utilisé pour organiser des groupes de composants.

Exemple swing

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingExample {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
    private static void createAndShowGUI() {
        // Crée un JFrame (fenêtre principale)
        JFrame frame = new JFrame("Exemple Swing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        // Crée un JPanel (conteneur pour les composants)
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());

        // Crée un JButton (composant bouton)
        JButton button = new JButton("Cliquez moi !");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(frame, "Le bouton a été cliqué !");
            }
        });

        // Ajoute le bouton au panel
        panel.add(button);
        // Ajoute le panel à la frame
        frame.add(panel);
        // Rend la frame visible
        frame.setVisible(true);
    }
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Mise en page et organisation des composants

Mise en page et organisation des composants

La mise en page et l'organisation des composants dans Swing sont des aspects essentiels pour créer des interfaces utilisateur bien structurées et attrayantes. Pour cela, Swing propose plusieurs gestionnaires de mise en page (layout managers) qui déterminent comment les composants seront disposés les uns par rapport aux autres dans un conteneur.

Gestionnaires de mise en page (Layout Managers) :

1 - BorderLayout :

Le BorderLayout divise le conteneur en cinq régions : NORTH, SOUTH, EAST, WEST et CENTER. Chaque région ne peut contenir qu'un seul composant, ce qui est utile pour les interfaces avec des zones bien définies.

BorderLayout

```
JFrame frame = new JFrame("BorderLayout Example");
frame.setLayout(new BorderLayout());

frame.add(new JButton("NORTH"), BorderLayout.NORTH);
frame.add(new JButton("SOUTH"), BorderLayout.SOUTH);
frame.add(new JButton("EAST"), BorderLayout.EAST);
frame.add(new JButton("WEST"), BorderLayout.WEST);
frame.add(new JButton("CENTER"), BorderLayout.CENTER);

frame.setSize(400, 300);
frame.setVisible(true);
```

Gestionnaires de mise en page (Layout Managers) :

2 - FlowLayout :

Le FlowLayout organise les composants dans une seule ligne ou colonne. Les composants suivent le flux naturel de la langue, ce qui est utile pour les interfaces simples.

FlowLayout

```
JFrame frame = new JFrame("FlowLayout Example");
frame.setLayout(new FlowLayout());

frame.add(new JButton("Button 1"));
frame.add(new JButton("Button 2"));
frame.add(new JButton("Button 3"));

frame.setSize(300, 200);
frame.setVisible(true);
```

Gestionnaires de mise en page (Layout Managers) :

3 - GridLayout :

Le GridLayout organise les composants dans une grille de lignes et de colonnes. Tous les composants ont la même taille.

GridLayout

```
JFrame frame = new JFrame("GridLayout Example");
frame.setLayout(new GridLayout(3, 3));

for (int i = 1; i <= 9; i++) {
    frame.add(new JButton("Button " + i));
}

frame.setSize(300, 300);
frame.setVisible(true);
```

Gestionnaires de mise en page (Layout Managers) :

4 - BoxLayout :

Le BoxLayout organise les composants dans une seule ligne ou colonne, mais offre également des options pour aligner les composants.

BoxLayout

```
JFrame frame = new JFrame("BoxLayout Example");
frame.setLayout(new BoxLayout(frame.getContentPane(), BoxLayout.Y_AXIS));

frame.add(new JButton("Button 1"));
frame.add(new JButton("Button 2"));
frame.add(new JButton("Button 3"));

frame.setSize(300, 200);
frame.setVisible(true);
```

Gestionnaires de mise en page (Layout Managers) :

5 - GridBagLayout :

Le GridBagLayout est le plus flexible et puissant. Il permet de créer des mises en page complexes avec des composants de différentes tailles et d'aligner les composants de manière précise.

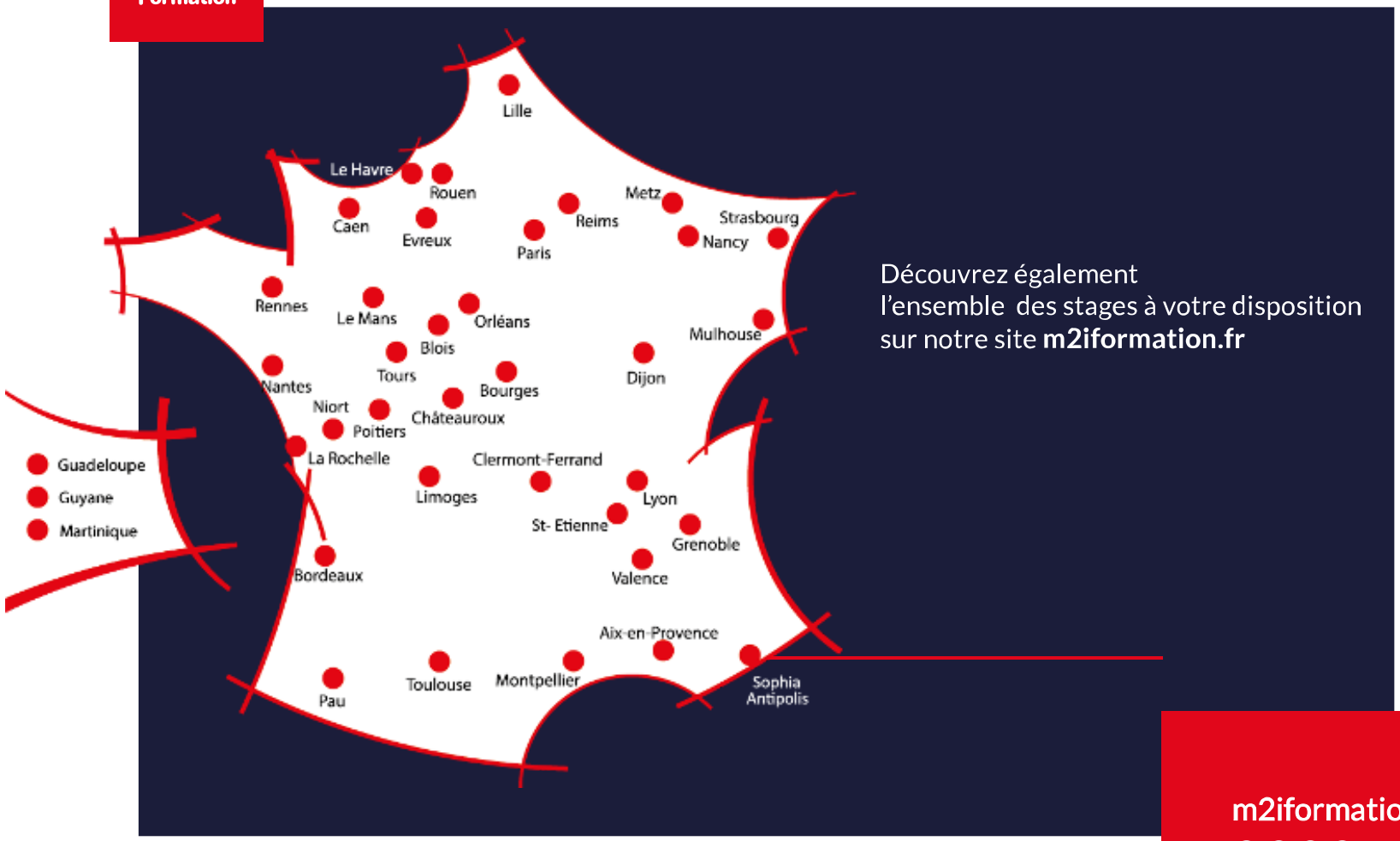
GridBagLayout

```
JFrame frame = new JFrame("GridBagLayout Example");
frame.setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
frame.add(new JButton("Button 1"), gbc);

gbc.gridx = 1;
gbc.gridy = 0;
frame.add(new JButton("Button 2"), gbc);

// ... Ajoutez d'autres composants en modifiant les contraintes

frame.setSize(400, 300);
frame.setVisible(true);
```

UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

Merci pour votre attention

Des questions ?



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr