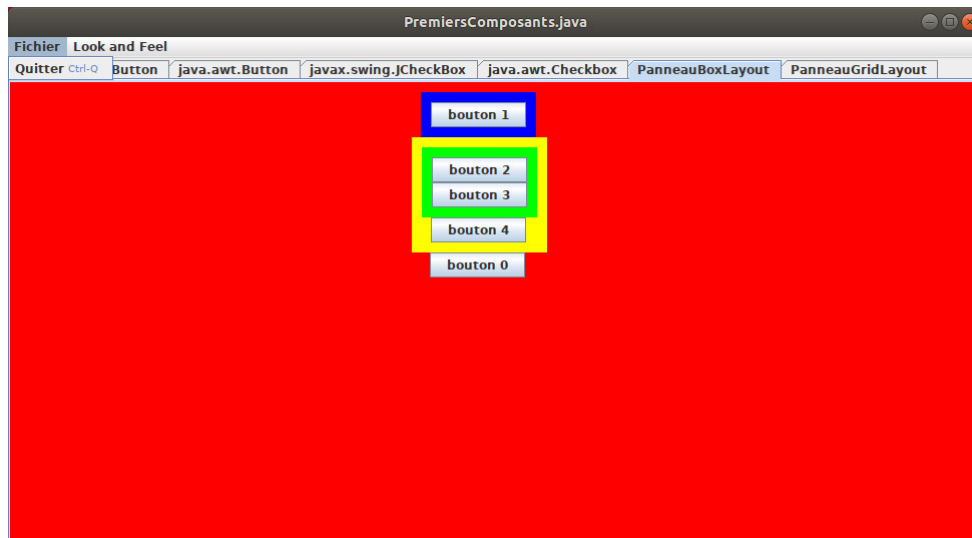


Interfaces Graphiques

TP 1 – Premiers composants¹



L’objectif de ce TP est de mettre en évidence les différences d’apparence entre composants graphiques Swing et AWT, de visualiser la profondeur de hiérarchie de composants et les stratégies de différents gestionnaires de disposition. Vous pouvez vous aider [des exemples d’utilisation de composants swing](#) disponibles sur le site d’Oracle, ainsi que de [la documentation des classes Java](#). Pour ce TP, quelques liens sont donnés vers la documentation de classes spécifiques.

1. Créez une `JFrame`. Faites en sorte que la fermeture de cette `JFrame` termine l’application (`setDefaultCloseOperation`).
2. Insérez dans la `ContentPane` de cette `JFrame` une `JTabbedPane`. Spécifiez pour elle une taille préférée, et faites en sorte que la `JFrame` adapte sa taille à celle de ce composant interne.
3. Ajoutez à la `JTabbedPane` des onglets contenant des instances des `Button`, `JButton`, `Checkbox`, `JCheckBox` – un onglet par instance – chacune étiquetée par un message (*e.g.* “cliquez-moi”). Faites en sorte que chaque onglet ait pour titre le nom de classe de son instance associée (`getClass()`, `getName()`).
4. Vous pouvez pour cette question consulter [la leçon du tutorial sur les menus](#). Créez une barre de menu (`JMenuBar`) contenant un menu nommé “Fichier” (`JMenu`), lui-même contenant une option “Quitter” (`JMenuItem`). Adjoindre à l’option ‘Quitter’ un raccourci clavier CTRL-Q (`setAccelerator()`). Créer un `ActionListener` (de classe anonyme) à l’écoute du `JMenuItem`, et dont la méthode `actionPerformed` terminera l’application.

1. Les sujets ont été rédigés par J-B. Yunès et V.Padovani, avec des contributions de F. Baille et S. Devismes.

5. Définir une usine statique de panneaux colorés à disposition `BoxLayout` verticale, c'est-à-dire une méthode

```
static JPanel panneauBoxLayout (Color couleur)
```

prenant en argument une couleur de classe `Color`, et renvoyant un panneau de classe `JPanel` de disposition `BoxLayout` verticale, de couleur de fond `couleur`, muni d'une marge de 10 pixels (c.f. la méthode `setBorder`, dont l'argument peut être produit par `BorderFactory`).

6. A l'aide de la fonction précédente et de la classe `JButton`, construire un assemblage de panneaux ayant la même structure que la copie d'écran ci-dessus :
- Panneau 0 (rouge)
 - Panneau 1 (bleu)
 - Bouton 1
 - Panneau 2 (jaune)
 - Panneau 3 (vert)
 - Bouton 2
 - Bouton 3
 - Bouton 4
 - Bouton 0

Afficher le composant construit dans le panneau à onglet de l'application, avec le titre "PanneauBoxLayout".

7. Ecrire une seconde usine statique

```
static JPanel panneauGridLayout(Color couleur, int lignes, int colonnes)
```

renvoyant des panneaux munis de la disposition `GridLayout` avec le nombre de lignes et colonnes spécifié, de couleur de fond `couleur`, toujours muni d'une marge de 10 pixels. Construire à l'aide de cette usine une matrice 5×2 boutons étiquetés par 0... 9.

8. On désire rajouter à la fenêtre principale un menu permettant de sélectionner un "Look and Feel" et de l'appliquer à tous les composants de l'application - c.f. la leçon [Modifying the Look and Feel](#) du tutorial de swing. Ecrire une méthode `static void ajouterSelectionLF(JFrame frame)` effectuant les actions suivantes :
- (a) Interroger le système sur les "Pluggable Look-and-feel" installés (c.f. `UIManager`),
 - (b) Créer un `JMenu` formé d'un `JMenuItem` par "Pluggable Look-and-feel" disponible.
 - (c) Créer un `ActionListener` (de classe anonyme) à l'écoute de chaque élément du menu. La sélection d'un élément du menu devra changer le Look and Feel de tous les composants de la `JFrame` (`SwingUtilities`).
 - (d) Intégrer ce menu à la `JFrame`.