

Fenêtres, panneaux, boutons, labels, listeners

Exercice 1 - Exemple de JFrame

- 1 Écrire une classe `SimpleFrame` permettant d'afficher une fenêtre. Nous utiliserons pour cela la classe `javax.swing.JFrame`.
On lui donnera simplement :
 - un titre (voir le constructeur ou `setTitle()`)
 - une taille minimale (`setSize()`)Puis on l'affichera via la commande `show()`.
- 2 Changer le comportement par défaut de la fenêtre lors de sa fermeture pour que l'application soit quitter (`JFrame.setDefaultCloseOperation(EXIT_ON_CLOSE)`).
- 3 Dans un second temps, ajouter à cette fenêtre un label (`JLabel`), représentant une étiquette (un texte court). En fait, pratiquement, on ajoute (`add`) le label au contenu de la fenêtre. Pour récupérer le conteneur permettant d'ajouter des composants à une fenêtre nous utiliserons la méthode `getContentPane()`.
- 4 Puis, ajouter un bouton "OK" (`JButton`) en plus du label.
Que se passe-t-il ? Comment faire pour que le label ne disparaisse pas ?

Exercice 2 - À la découverte de nouveaux composants

Créer une nouvelle `JFrame` nommée "Découverte" et ajouter dans son `contentPane` les composants suivant :

- un bouton (`JButton`)
 - un champ de texte (`JTextField`)
 - une zone de texte (`JTextArea`)
- 1 Ajouter une bulle d'aide contextuelle (`setToolTipText()`) sur le bouton.
 - 2 Rendre le bouton inopérent (`setEnabled()`)
 - 3 Changer la couleur du texte et du fond du bouton (`setForeground()` et `setBackground()`).
 - 4 Ajouter un texte dans le champ de texte et dans la zone de texte. (`setText()`)
 - 5 Rendre le champ de texte non éditable (`setEditable()`)
 - 6 Ajouter une bordure au champs de texte (avec la méthode `setBorder()`) en fournissant un objet de type `java.swing.border.Border` créé via une méthode de la classe `java.swing.BorderFactory`, par exemple la méthode `createLineBorder(Color color, int thickness)`.

Exercice 3 - Associer des actions aux boutons

Créer une fenêtre contenant un bouton de label "Test" qui, lorsqu'il reçoit un clic, affiche "Test clic" dans la fenêtre de contrôle (console ou xterm).

- 1 Créer la fenêtre et le bouton.
- 2 Créer une classe (elle peut être anonyme) implantant l'interface `ActionListener` et effectuant l'affichage dans la méthode `actionPerformed()`.
- 3 Relier le listener au bouton grâce à la méthode `addActionListener()` du bouton.
Utiliser la même technique avec trois boutons, "Rouge", "Vert" et "Bleu", qui permettent de

changer la couleur de fond de l'application. Produire un code qui permette de rajouter facilement une nouvelle couleur.

Exercice 4 - À la maison ...

Chaque TD est suivi d'un devoir à la maison vous permettant de remanipuler ce qui à été vu en TD.

On souhaite créer un panneau contenant quatre cases à cocher (`JCheckBox`) étiquetées "zoro", "invanhoé", "spiderman" et "moi". On veut faire en sorte qu'il n'y ait qu'un seul checkbox sélectionné à la fois. Il y a deux algorithmes différents :

- 1 Faire en sorte que lorsque l'on sélectionne un checkbox les autres soient automatiquement désélectionnés.
- 2 Faire en sorte de désélectionner l'ancien checkbox sélectionné avant de sélectionner le nouveau.

Il est possible d'obtenir tous les fils d'un `Container` (un `JPanel` est un `Container`) à l'aide des méthodes `getComponentCount()` et `getComponent()`.

Dans cette exercice, il est demandé de ne pas utiliser la classe `ButtonGroup` mais de programmer 'à la main' le comportement.