

Groupe de boutons, Layout

Exercice 1 - Groupe de boutons

On souhaite créer un panneau contenant quatre cases à cocher (JRadioButton) étiquetées "zorro", "invanhoé", "spiderman" et "moi". On veut faire en sorte qu'il n'y ait qu'un seul bouton radio sélectionné à la fois. Pour cela, nous utiliserons la classe ButtonGroup qui permet de créer un groupe logique de boutons.

Exercice 2 - Exemple de layout

Écrire une classe LayoutExample permettant de visualiser, dans des frames internes de classe JInternalFrame, les mises en pages gérées par les classes :

- FlowLayout, utilisée par défaut dans les JPanel ;
- GridLayout, utilisée pour répartir les composants dans une grille régulière ;
- BorderLayout, utilisée par défaut dans le contentPane des JFrame, il permet d'indiquer une contrainte de placement (NORTH, SOUTH, EAST, WEST et CENTER) ;
- BoxLayout, utilisée pour aligner les composants dans une boîte suivant un axe choisi (X_AXIS pour l'axe horizontal et Y_AXIS pour l'axe vertical).

Une JInternalFrame est une fenêtre logée à l'intérieur de la frame principale (JFrame). Il n'est malheureusement pas possible d'ajouter directement une fenêtre interne à une JFrame. Il faut utiliser un JDesktopPane comme parent de n'importe quelle JInternalFrame. Ce desktop jouera donc le rôle de contentPane auprès de la JFrame. De plus, comme les JFrame, pour ajouter un composant à une JInternalFrame, il faut le faire par l'intermédiaire de son contentPane.

```
JFrame frame=new JFrame();
JDesktopPane desktop=new JDesktopPane();
frame.setContentPane(desktop);

JInternalFrame iframe=new JInternalFrame();
iframe.setVisible(true);
desktop.add(iframe);

Container c=iframe.getContentPane();
c.add(new JButton("hello"));
```

Essayer de redimensionner chacune des JInternalFrame pour observer le comportement des composants que leur layout gère. Insérer dans un second temps, des appels à la méthode pack() sur chacune de ces JInternalFrame avant de les ajouter au JDesktopPane et observer l'effet de ces appels.

Exercice 3 - Le gridbag layout

The screenshot shows a Java Swing window titled "NotesForm". It contains a form with the following elements:

- Nom:** A text field containing "Machprou".
- Prénom:** A text field containing "Duglu".
- Niveau:** A dropdown menu showing "S4".
- Option:** A dropdown menu showing "MI". A list of options is open, showing "SM", "MI" (highlighted), and "SV".
- Groupe:** A dropdown menu showing "3".
- Redoublant:** A checked checkbox.
- Buttons:** "OK" and "Annuler" buttons.

Pour cela, on veut utiliser les classes `GridBagLayout` et `GridBagConstraints`

- 1 Faire hériter votre classe de `JFrame`.
- 2 Créer un `GridBagLayout` afin de le positionner comme gestionnaire de layout du `ContentPane` de la `JFrame`.
- 3 Créer les différents composants nécessaires pour votre formulaire (`JCheckBox`, `JTextField`, `JButton`, `JComboBox`, etc.).
- 4 Découpez les différentes parties de votre formulaire et ajoutez dans le `ContentPane` les différents composants en spécifiant, pour chacun d'entre eux, les paramètres distincts du layout à l'aide d'un objet `GridBagConstraints`.

Exercice 4 - À la maison ...

Écrire un programme qui propose, dans une partie d'une fenêtre, la visualisation de plusieurs composants (par exemple des labels) et dans l'autre partie de la fenêtre un ensemble de choix (sous la forme de radio boutons) permettant dynamiquement de changer le layout d'organisation des composants dans la première partie.