

SE1: Introduction aux IHM

Objectifs

- *S'initier aux concepts d'IHM en Java en réalisant ses premières interfaces graphiques.*

Nous voulons un programme avec interface graphique qui affiche une valeur entière et la modifie de +1 ou -1 par le biais des boutons appropriés.

1 – Rappels (très brefs !) de Java

Ecrivez, histoire de vous dérouiller un peu, une classe Compteur qui consiste en une valeur entière pouvant être incrémentée, décrétementée ou remise à zéro. Testez cette classe (sans trop vous étendre dessus).

2 – Votre toute première IHM

Nous allons maintenant faire une première interface graphique. Pour l'instant, elle sera déconnectée de la classe Compteur (nous verrons par la suite comment faire la connexion).

Ecrivez une classe IHMCompteur qui affiche une fenêtre titrée “Compteur”, avec un label qui dit “bonjour”. Changez la valeur du label par “Compteur : 0”. Donnez à la fenêtre une taille fixe, puis enlevez-la. Que constatez-vous ? Trouvez le moyen de garder la taille fixe.

3 – Placement de composants

Maintenant que nous avons affiché un composant, nous allons en afficher plusieurs et les disposer les uns par rapport aux autres.

Ajoutez un bouton à la fenêtre, intitulé “Plus Un”, au-dessous du label. Utilisez *GridLayout* pour placer les deux composants l'un par rapport à l'autre. Recommencez avec *BorderLayout* en essayant plusieurs positions possibles (nord-centre, nord-sud, centre-sud...).

4 – Utilisation de panel

Ajoutez deux autres boutons, “Moins Un” et “Zero”, au même niveau que le premier bouton. Utilisez d'abord *BorderLayout* pour disposer les trois boutons au même niveau. Que constatez-vous ? Essayez avec *GridLayout*.

On se rend vite compte que ce n'est pas très propre de disposer tous les composants sur la même Frame. Ce n'est pas non plus très cohérent, étant donné qu'il y a, d'un côté, le label, et de l'autre les boutons. Il va falloir utiliser les panneaux.

Mettez les 3 boutons dans un panneau, que vous disposerez en-dessous du label dans la fenêtre principale. Pour disposer les boutons à l'intérieur du panneau, utiliser *GridLayout*, puis aucun *Layout*. Que constatez-vous ?

5 – Evènements associés aux composants

Maintenant que nous avons tous nos composants, nous allons pouvoir les lier au code de la classe *Compteur*.

Créez une classe *IncDecListener*, dont l'action est associée à un objet de type *Compteur*, ainsi qu'à un bouton et un label. L'action consiste à modifier et afficher la valeur du compteur (sous la forme "Compteur : {nouvelle valeur}") quand on appuie sur un bouton. Si c'est le bouton "Plus Un", la valeur du compteur est incrémentée. Si c'est le bouton "Moins Un", elle est décrémentée. Si c'est le bouton "Zero", elle est remise à zéro. Veillez à bien utiliser les méthodes de la classe *Compteur*.

Utilisez cette classe pour rendre les trois boutons utilisables, et testez votre application, désormais complète... ou presque.

6 – Evènements associés à la fenêtre

Essayez maintenant de fermer la fenêtre. Que se passe-t-il ?

Utilisez l'interface *WindowListener* afin de remédier au problème. Quel est l'inconvénient de cette méthode ? Essayez de trouver une solution vraiment adaptée.

