

Action et modèle

Exercice 1 - Action ... Couper

Le but de cet exercice est de manipuler une série d'actions. Les actions sont des objets implémentant l'interface `Action` (qui hérite de `ActionListener`) et permettant de regrouper au sein d'un même objet, le traitement, l'icône, les raccourcis clavier, les bulles d'infos, etc.

Il existe de plus la classe `AbstractAction` qui prédéfinit un mécanisme de listener permettant d'associer une action à plusieurs composants swing (boutons, menus).

- 1 Créer une liste (`JList`) contenant l'ensemble des arguments de la ligne de commande, un menu et une toolbar.
- 2 Créer une action (`New`) permettant de rajouter une chaîne de caractères à la liste en demandant à l'utilisateur de saisir la chaîne. L'action pourra être déclenchée soit par l'intermédiaire d'un menu soit en cliquant sur l'icône de la toolbar.
Utiliser pour cela la classe `JOptionPane` qui permet d'afficher une fenêtre de demande à l'utilisateur et une classe héritant de la classe `AbstractAction` correspondant à l'action (`New`).
- 3 Créer une action (`Delete`) qui permet de détruire la chaîne sélectionnée dans la `JList`.
- 4 Faire en sorte que l'action (`Delete`) soit grisé si aucune chaîne n'est sélectionnée.

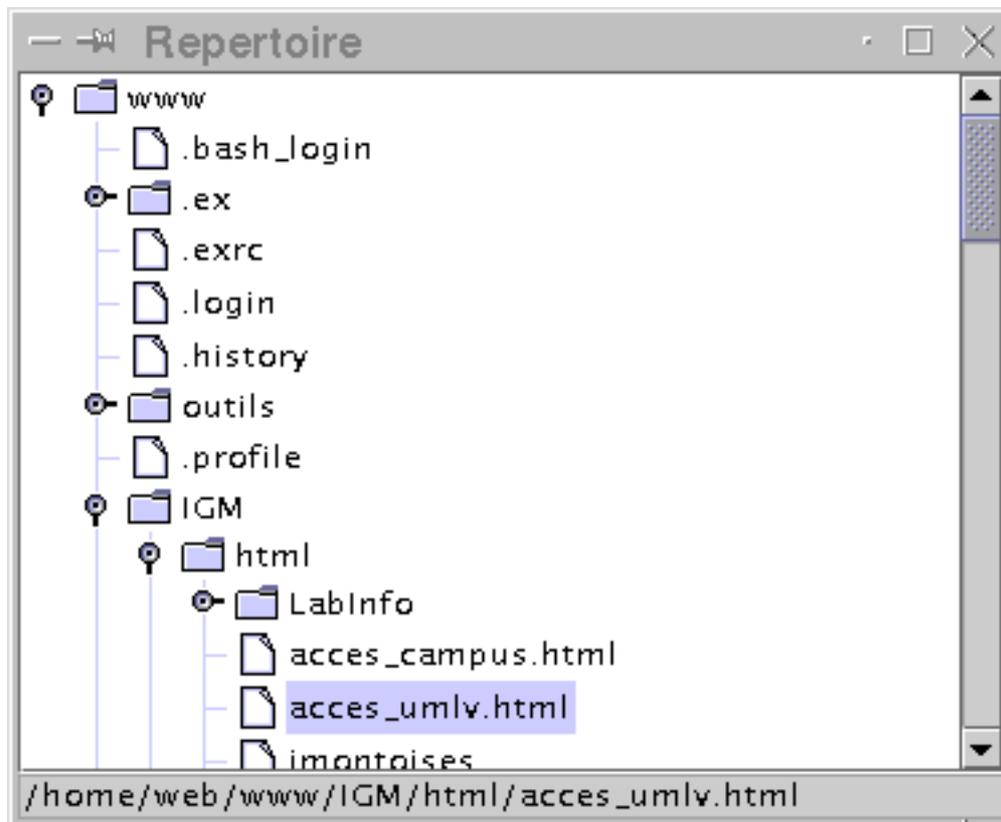
Exercice 2 - Arbre et répertoire

On désire écrire un programme de visualisation de l'arborescence de fichiers d'un répertoire donné (par exemple, le `home.dir`).

Pour cela, on choisit de définir la classe `FileNode` représentant les noeuds d'une arborescence correspondant à des répertoires ou à des fichiers. Cette classe devra donc implanter l'interface `TreeNode`, afin d'être utilisée dans un composant `JTree`.

De plus, il n'est pas souhaitable que l'ensemble de l'arborescence soit chargée au départ. On préférera, autant que possible, un chargement dynamique, à la demande (*lazy*), lors du déplacement dans la hiérarchie et de l'ouverture des répertoires.

Par ailleurs, dans un champ de texte en bas de la fenêtre, on veut visualiser le chemin absolu (dans le système de fichier considéré) du fichier ou du répertoire sélectionné.



On pourra également faire afficher une fenêtre d'alerte lorsque la lecture d'un répertoire pose un problème (protection en lecture ou erreur quelconque d'entrée/sortie).

Parmi les améliorations souhaitables, on peut citer :

- 1 Le tri par ordre alphabétique des noms de répertoires et de fichiers. On peut également envisager une barre de menu permettant de spécifier les critères de tri des fichiers apparaissant dans l'arbre, ou encore des filtres sur les noms des fichiers (ou répertoires) devant apparaître.
- 2 La séparation en deux de la fenêtre (`JSplitPane`) avec l'arborescence des répertoires d'un côté et les fichiers du répertoire sélectionné de l'autre.
- 3 L'association d'icônes ou de "rendus" différents en fonction du type de fichier.
- 4 La possibilité d'ouvrir, de renommer ou de supprimer les fichiers sélectionnés, grâce à l'ouverture d'un menu contextuel, par exemple.
- 5 Etc.