

TD2 – NF28 : modèle MVC et composants Java

Présentation

Objectif

L'objectif de ce TD est d'implémenter le modèle MVC en utilisant le concept de `PropertyChangeListener`, d'utiliser un composant de haut niveau possédant son propre modèle et d'implémenter l'usage d'un `JFileChooser`.

Le principe de base reste qu'une vue est nécessairement adaptée à son modèle et que le modèle ne connaît pas les éventuels objets qui peuvent dépendre de son état (que ceux-ci soient des vues ou des objets quelconques).

Principe de réalisation

On considère un dictionnaire comme étant une liste de mots associés à une définition. Sous forme de fichier texte, c'est une suite de lignes de la forme `<mot>=<définition>`. L'interface doit présenter la liste des mots d'un dictionnaire et afficher la définition du mot sélectionné.

Exemple

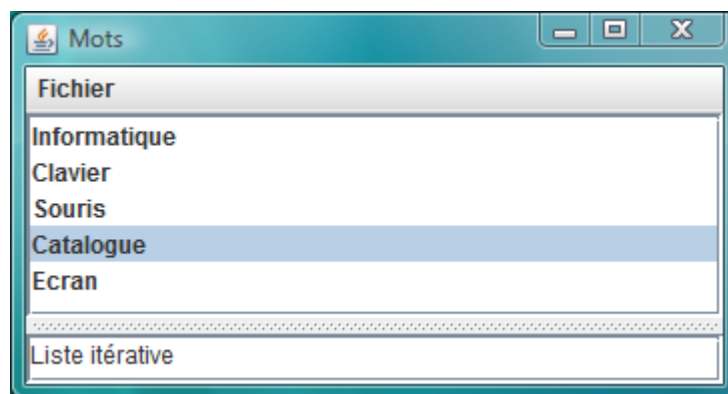


Figure 1 : présentation d'une liste

Proposition de phases de réalisation

Phase 1

Créer une interface graphique permettant de présenter la liste des mots (dans un objet de type `JList`) et une définition. Une entrée de menu permet de choisir un fichier contenant un dictionnaire. On aura intérêt à créer une classe spécifique pour le panneau contenant la définition d'un mot puisqu'il sera spécialisé en `PropertyChangeListener`. Ajouter un menu pour le choix d'un fichier dictionnaire (écrit à la main).

Phase 2

Créer une classe servant de modèle à l'application. Elle devra contenir un champ pour le modèle du `JList` (`DefaultListModel`), et un champ de type `PropertyChangeSupport` pour l'ajout de listener et l'envoi de notifications. Elle contiendra une méthode permettant de lire un fichier dictionnaire et de compléter au fur et à mesure le modèle du `JList`.

Test

Pour tester l'application, on pourra créer une classe contenant une méthode main ne servant qu'à lancer l'ouverture de la fenêtre. Pour faire apparaître la fenêtre, il suffit de la créer. Son constructeur doit contenir l'ensemble des paramètres permettant de la rendre visible et de quitter l'application en fermant la fenêtre.

```
public class MainTDnf28 {
    public static void main(String[] args) {
        td2();
    }
    public static void td2() {
        ListView listView = new ListView("Mots");
    }
}
```

Phase 3

Ajouter les listener dans la classe fenêtre pour répondre au choix d'un fichier dictionnaire et à la sélection dans la liste.

Transformer le panneau d'affichage de définition en `PropertyChangeListener`.

Mettre en place le principe de notification à ce listener.

Prolongement

Prévoir une possibilité d'ajout et de suppression de mots dans le dictionnaire.

Adapter l'application pour pouvoir lire des fichiers contenant des lignes du type :

```
<mot>=<définition>@<traduction du mot en anglais>
```

et d'afficher dans le panneau du bas la définition et la traduction dans deux champs différents.