Gestion de menus dans des applications fenêtrées
Programmes d'exemples avec JAVA/SWING

Cours IHM

C. Bertelle - Université du Havre

1 Construction du menu

Ecrire un programme Java permettant d'ouvrir une fenêtre qui contient un menu dont la structure arborescente est la suivante :

- Fichier
  - Ouvrir
  - Enregistrer
    - avec nom courant
    - avec nouveau nom
  - Quitter
- Edition
  - Couper
  - Copier
  - Coller
  - Options
    - mode insert
    - mode refrappe
- Aide
  - Tutorial
  - Doc API

2 Menu Réactif

Compléter le programme Java précédent de manière à placer dans la fenêtre d'affichage un libellé "Votre choix : ", de type `JLabel` et un champ de saisie de texte `JTextField`. Ainsi lorsque l'utilisateur choisit un item du menu, celui-ci doit s'afficher dans le champ de texte. De plus le choix de l'item "Quitter" doit fermer la fenêtre d'affichage.

3 Ajout d'une boîte de dialogue de sélection de fichiers

On complète le programme précédent de façon à ce que le choix de l'item "Ouvrir" provoque l'affichage d'une boîte de dialogue permettant la représentation graphique et arborescente des fichiers du répertoire en cours avec possibilité de sélection du fichier à la souris.

- `JFileChooser` est la classe Swing qui est utilisée pour cela. On en créé une instanciation.
- `showOpenDialog(parent)` est la méthode de cette classe qui permet l'ouverture de la boîte de dialogue : elle a pour paramêtre le composant parent de l'appel (par exemple, le composant en cours `this`). Suite à l'appel de cette méthode, le flux d'exécution du programme ne revient pas tant que l'utilisateur n'a pas sélectionné un fichier ou annulé la sélection.
- La méthode précédente renvoie une valeur entière qui vaut une des deux constantes : `JFileChooser.APPROVE_OPTION` ou `JFileChooser.CANCEL_OPTION`. Dans le premier cas de valeur retournée, un fichier a été choisit par l'utilisateur : son nom est récupérable

sous la forme d'une chaîne de caractères renvoyée par l'appel de la méthode `getSelectedFile().getName()` sur l'instance de la classe `JFileChooser`.

4 Ajout d'un traitement de texte rudimentaire

On complète maintenant le programme

- en ajoutant un champ de données du type `JTextArea` stockant le contenu du texte à éditer ;
- en ajoutant un champ de données du type `String` stockant une portion de texte sélectionné ;
- le choix des items "Couper", "Copier" et "Coller" doit permettre d'implémenter les fonctionnalités attendues d'un traitement de texte courant sur les données du champ `JTextArea`.

On utilise, pour cela, les méthodes prédéfinies suivantes de la classe `JTextArea` :

- `getSelectedText()` renvoie la chaîne correspondant à la partie de texte sélectionnée à la souris ;
- `getSelectionStart()` et `getSelectionEnd()` renvoient chacune un entier qui correspond respectivement à la position de début du texte sélectionné et la position de la fin du texte sélectionné ;
- `replaceRange(str, i, j)` permet de substituer, à la partie de texte comprise entre les positions i et j, la chaîne de caractères `str`.