C. Recanati
Java Graphique
AIR2

Feuille de TP 6

Exercice 1

On va créer une applet qui présente une liste scrollée dans sa partie centrale et, au dessus, deux boutons alignés horizontalement dans un panneau qui permettront respectivement :

1. de détruire les items sélectionnés dans la liste
2. d'ajouter un item à la liste.

Remarque: si on ne souhaite pas créer d'applet, on peut alors créer un cadre JFrame d'application.

1) Compléter pour cela le squelette de programme suivant, en réalisant tout d'abord simplement l'affichage du contenu du cadre.

```
public class Test extends JApplet {
        private JList list = new JList();

        String[] items = { "item[0]", "item[1]", "item[2]", "item[3]",
                           "item[4]", "item[5]", "item[6]", "item[7]",
                           "item[8]", "item[9]" };

        public void init() {
                Container contentPane = getContentPane();
                ControlPanel controlPanel = new ControlPanel(list); //+ loin
                contentPane.add(controlPanel, BorderLayout.NORTH);
                contentPane.add(new JScrollPane(list),
                                        BorderLayout.CENTER);
                populateList();
        }
        public void populateList() {
        // construire un modèle à partir de DefaultListModel
        // l'initialiser avec le tableau d'items et l'affecter à la liste privée
        //  ...
        }
}
class ControlPanel extends JPanel {
        private JButton remove = new JButton("Supprimer la selection");
        private JButton add = new JButton("Ajouter un item");
        private JList list ;
```

```
    public ControlPanel(JList ls) {
        this.list = ls ;
        add(remove);
        add(add);

        remove.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
            // récupérer les indices sélectionnés de la liste
            // et récupérer le modèle
            // ...
            // puis supprimer les items correspondants du modèle

            // ... on pourra présenter d'abord une boîte
            // demandant à l'utilisateur s'il souhaite vraiment
            // supprimer la sélection (utiliser une méthode de la
            // classe JOptionPane)
            }
        });
        add.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // récupérer le modèle de la liste

                // récupérer la chaîne (l'item) à rajouter dans
                // une boîte de dialogue modale prête à l'emploi
                // fournie par la classe JOptionPane

                // ajouter cette chaîne au modèle

                SwingUtilities.invokeLater(new Runnable() {
                    public void run() {
                        list.ensureIndexIsVisible(
                                model.getSize()-1);
                    }
                });
            }
        });
    }
}
```

Remarque : avec `invokeLater`, la procédure `ensureIndexIsVisible` n'est effectuée qu'après que les événements de mise à jour aient été traités.

2) Encadrer joliment la liste scrollée : on placera le `JScrollPane` contenant la liste dans un `JPanel` muni d'un bord d'une certaine épaisseur et d'un encadrement. (Utiliser `createEmptyBorder` de `BorderFactory` pour avoir un bord vide, et `new LineBorder(Color, int)` pour avoir un encadrement noir, et rajouter un titre comme par exemple « Liste des éléments » en affectant au `JPanel` un bord composé à partir du bord vide et du bord noir finalement titré et créé avec la méthode `createCompoundBorder` de `BorderFactory`).

3) On rajoute au panneau de contrôle du haut une `JComboBox` présentant le mode de sélection utilisée par la liste, soit : `SINGLE_SELECTION, SINGLE_INTERVAL_SELECTION,` ou `MULTIPLE_INTERVAL_SELECTION`. On écrira une procédure privée `initializeSelectionMode`() qui initialisera l'item sélectionné de la comboBox conformément au mode de sélection initial de la liste, et on ajoutera un `ItemListener` à la comboBox pour mettre à jour le mode de sélection de la liste quand un nouveau mode sera sélectionné sur la comboBox.

4) Mettre en place les réactions sur les boutons add et remove du panneau de contrôle conformément aux commentaires du squelette de programme (utiliser des boîtes de dialogue).

5) Modifier encore le panneau de contrôle `ControlPanel` pour qu'il affiche sous les boutons précédents:

1. les indices des items sélectionnés dans la liste (sous une forme du style "Index des items sélectionnés : 3, 4, 5, 9 ") ;
2. et au dessous les index minimum et maximum des items sélectionnés.

Pour la mise à jour de ces labels particuliers, on écrira une méthode `updateLabel`() pour le panneau de contrôle qui sera appelée quand la sélection de la liste aura changée (utiliser un écouteur `ListSelectionListener` sur la liste).

Pour le placement des composants dans le `ControlPanel`, on pourra utiliser un `BoxLayout` vertical. En haut, un `JPanel` servira à placer les boutons et la comboBox comme précédemment, et au centre et en bas on placera les labels adéquats.