Projet à rendre

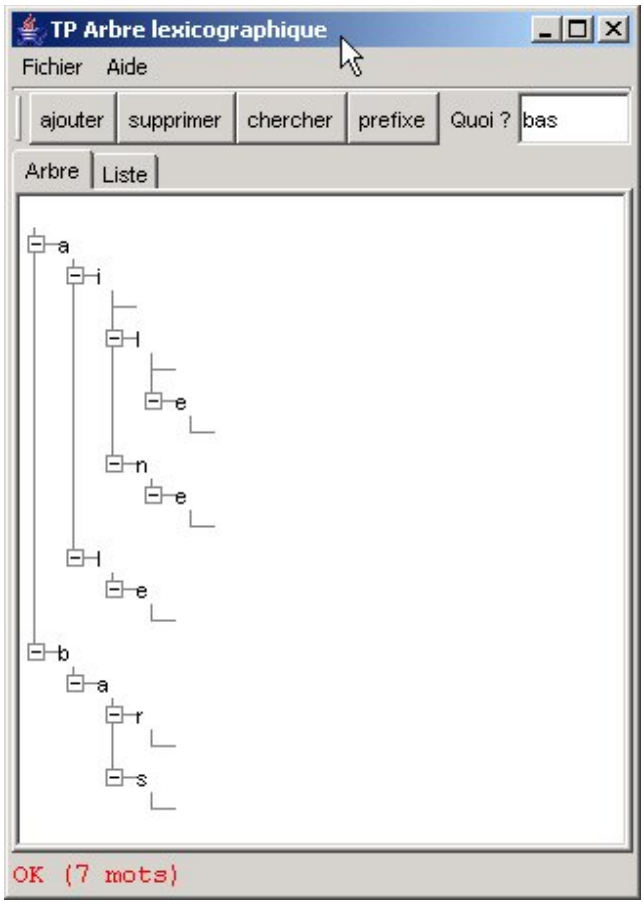
1. Aspect `Serialisation`

Définir un aspect `Serialisation` permettant de rendre la classe `ArbreLexicographique` sérialisable. Cet aspect doit contenir :

- une déclaration inter-type indiquant que `ArbreLexicographique` implémente l'interface `Serializable` ;
- une introduction de la méthode `void sauve(String nomFichier)` dans `ArbreLexicographique` qui permet de sauvegarder un arbre lexicographique dans un fichier ;
- une introduction de la méthode `void charge(String nomFichier)` dans `ArbreLexicographique` qui permet de charger un arbre lexicographique depuis un fichier.

2. Interface graphique

Réaliser une interface graphique proche de cet exemple :



Le menu fichier doit comporter deux items permettant de sauvegarder ou charger un arbre lexicographique depuis un fichier, ainsi qu'un item permettant de quitter l'application.
La synchronisation entre le `Jtree` utilisé et l'arbre lexicographique doit être faite essentiellement au travers de l'aspect `Visualisation` (voir ci-dessous).
De nombreuses indications sur l'utilisation des composants Swing utiles pour la réalisation de cette interface sont disponibles sur
`http://imss-www.upmf-grenoble.fr/prevert/Prog/Java/swing/tableDesMatieres.html`

3. Aspect `Visualisation`

L'aspect `Visualisation` a pour but de pouvoir visualiser un arbre lexicographique en utilisant le composant `Jtree` utilisé dans l'interface graphique. Un `Jtree` est conçu pour afficher une arborescence de type `TreeModel`, dont les nœuds sont de type `TreeNode`. Il existe en Java notamment deux classes, `DefaultTreeModel` et `DefaultMutableTreeNode`, qui implémentent respectivement les interfaces `TreeModel` et `TreeNode`. On peut donc définir l'aspect `Visualisation` en se basant sur les indications données ci-dessous.

3.1 Déclarations inter-type

Prévoir une déclaration inter-type indiquant que `ArbreLexicographique` implémente l'interface `TreeModel` et une autre indiquant que `NoeudAbstrait` implémente l'interface `TreeNode`.

Définir l'introduction d'un attribut privé de type `DefaultTreeModel` dans `ArbreLexicographique` et l'introduction d'un attribut privé de type `DefaultMutableTreeNode` dans `NoeudAsbtrait`.

Définir les introductions de méthodes nécessaires pour que les interfaces `TreeModel` et `TreeNode` soient correctement implémentées.

Définir l'introduction d'un attribut privé `vue` de type `JTree` dans `ArbreLexicographique`.

Définir l'introduction d'une méthode `public void setVue(JTree jt)` dans `ArbreLexicographique` permettant de modifier la valeur de l'attribut `vue`. Cette méthode peut être appelée depuis la classe d'application graphique.

3.2 Synchronisation dynamique

Définir des pointcuts et advices de telle sorte que chaque modification de structure dans un `ArbreLexicographique` soit prise en compte dans le `DefaultTreeModel` associé.

Définir un pointcut et un advice permettant de rafraîchir le `JTree` associé à un `ArbreLexicographique` après tout changement de structure.