

Exercice 1 : (ne cherchez aucun réalisme par rapport au comportement ou à l'implémentation de Unix !) : On désire simuler une version simplifiée de l'arborescence du système de fichiers d'Unix dans un langage à objets. Pour cela, on se restreint aux répertoires et aux fichiers, ces derniers étant supposés être de deux types: les fichiers « texte » et les fichiers exécutables. Les commandes du système sont vues comme des méthodes applicables aux fichiers et aux répertoires (par exemple `F.ls()` si `F` est un fichier). Les commandes considérées sont les suivantes : `ls`, `cat`, `cd`, `mkdir`, `mv`, `chmod`, `rm`, `rmdir`. On suppose que l'exécution d'un fichier exécutable se fait via la commande `exec`. La commande `cat` ne fonctionne que sur des fichiers « texte ».

Décrivez les classes que vous utiliseriez pour représenter les objets manipulés dans cette simulation. Ne donnez pas le code des méthodes, sauf celui de la commande `ls` que vous pouvez écrire dans un pseudo-langage à votre convenance. On suppose qu'on dispose d'une commande `Print` pour imprimer une chaîne de caractères.

Votre commande `ls` liste-t-elle aussi le contenu des sous-répertoires ?

Ecrire une commande `rm-r` qui fonctionne de manière récursive, en nettoyant automatiquement les sous-répertoires.

Exercice 2 : Une société emploie plusieurs personnes à différents niveaux hiérarchiques parmi Employé, Cadre, Cadre Supérieur et Directeur. Chaque employé est caractérisé par

- une identification (un entier)
- une catégorie
- une ancienneté exprimée en années
- un salaire brut annuel
- une note attribuée par la DRH.

Les augmentations de salaire ainsi que les promotions (passage d'une catégorie à la catégorie immédiatement supérieure) sont définies de la façon suivante :

	Employé	Cadre	Cadre sup.	Directeur
<i>promotion</i>	ancienneté > 3 et note > 70	ancienneté > 2 et note > 80	ancienneté > 6 et note > 90	NEANT
<i>augmentation</i>	2% * note /100	3% * note /100	3% * note /100	4% * note /100

L'ancienneté est calculée à partir de chaque changement de statut. On ne change que d'une catégorie à la fois.

Proposez un ensemble de classes permettant d'implémenter cette gestion des ressources humaines en tenant compte des contraintes suivantes :

- la promotion d'une personne ne doit pas donner lieu à destruction ou à création d'instances de Personnes
- la création ultérieure de nouvelles catégories ne doit pas entraîner de modifications importantes du système existant.
- On doit pouvoir modifier simplement certaines règles de calcul de promotion ou d'augmentation (par exemple distinguer l'ancienneté dans l'entreprise et l'ancienneté dans la catégorie courante) ou dans certaines catégories avoir un pourcentage d'augmentation qui dépend du salaire brut courant.