

Examen Master IPS - UMINS101

Programmation Java - Session juin 2007

Problème 1 - Etudiants et modules

Pour représenter les notes obtenues par un étudiant à un module, on va définir la classe `ModEtud`, avec comme attributs un nom de module (`String nomMod`), un coefficient de module (`int coeff`), et un ensemble (`Vector notesMod`) de notes représentant les notes obtenus par l'étudiant aux différentes épreuves du module. Commencez par écrire en java les attributs de la classe `ModEtud`. Un étudiant (simplifié) a maintenant comme attributs son nom, et un `Vector` dont chaque élément est de la classe `ModEtud`. Ecrivez en java les attributs de la classe `Etudiant`. La note d'un étudiant à un module est obtenue en faisant la somme des notes présentes dans `notesMod`, divisée par le nombre de ces notes. La moyenne générale d'un étudiant est obtenue en calculant la moyenne des notes obtenues dans les différents modules, pondérée par les coefficients.

Question 1 : Ecrivez la méthode `public double moyGen` de la classe `Etudiant` qui calcule la moyenne générale d'un étudiant.

Question 2 : Affichage de la mention

Règle : Un étudiant est ajourné si l'une des notes obtenues à un module est inférieure ou égale à 5 (note éliminatoire) ou si sa moyenne générale est inférieure à 10.

Complétez la méthode de la question 1 en écrivant une méthode `afficheMention` qui affiche le nom, la moyenne générale, et la mention d'un étudiant.

Question 3 : Affichage des résultats

Complétez la méthode précédente en écrivant une méthode `afficheRes` qui affiche le nom, la moyenne générale, la mention, et la liste des modules « à repasser » (note du module inférieure à 10).

Problème 2 - Fichiers

Nous étudions une représentation simplifiée des fichiers à l'aide d'une hiérarchie de trois classes. La plus haute est la classe `Fichier`, qui se spécialise en `FichierOrdinaire` et `FichierRépertoire`. Tous les fichiers ont un nom qui est une chaîne de caractères et des droits d'accès pour l'utilisateur, le groupe et les autres en lecture, écriture et exécution/traversée.

- Un fichier ordinaire est décrit également par une taille en nombre d'octets.
- Un fichier répertoire contient d'autres fichiers (fichiers ordinaires ou répertoires). Un fichier répertoire est créé vide et on peut lui ajouter un fichier grâce à une méthode `ajoute(Fichier f)`.

Question 1 Ecrivez les classes `Fichier`, `FichierOrdinaire` et `FichierRépertoire` avec leurs attributs et munies de constructeurs (n'écrivez pas les accesseurs mais supposez qu'ils existent).

Question 2 Ecrivez pour ces trois classes une méthode `public int taille()` retournant la taille du fichier.

Question 3 Ecrivez pour ces trois classes une méthode `public String toString()`. Pour un fichier simple, on retourne une chaîne contenant son nom et sa taille, pour un fichier répertoire, on retourne une chaîne contenant son nom, sa taille, et son contenu (et le contenu des sous-répertoires jusqu'au plus bas niveau de l'arborescence).