

Université Paris XII
IUT de Fontainebleau
Département informatique
Licence SIL
Structures de données abstraites
2004-2005

PARTIEL 2

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant imprimé (et celui-ci seulement) et les notes manuscrites (pas de photocopie) comprenant le nom de l'étudiant sur chaque page.

Les deux exercices doivent être présentés dans l'ordre (prévoir deux pages par exercice).

Il est conseillé d'écrire les exercices en langage C++. L'écriture en Java complique les entrées-sorties et l'écriture en langage C allonge le programme (ce qui ne rapportera pas de point supplémentaire dans un cas comme dans l'autre).

Exercice- (Logiciel de gestion des notes)

- 1°) Définir une classe `etudiant` comportant quatre attributs, un nom qui est une chaîne d'au plus 30 caractères, un tableau `note` d'au plus 20 réels (contenant les notes), un tableau `coeff` d'au plus 20 entiers (contenant les coefficients) et un réel `avg` contenant la moyenne des notes valides (différentes de -1) coefficientées. Les méthodes seront un constructeur ayant un argument chaîne de caractères (permettant d'initialiser le nom avec la valeur passé en argument, le tableau des notes avec que des -1, celui des coefficients à 1 et la moyenne à 0), une méthode `getNom()` permettant de récupérer le nom, une méthode `setNote(int i, float x, int c)` permettant d'entrer la i -ième note avec le coefficient c (la moyenne sera mise à jour) et une méthode `getMoyenne()` permettant de récupérer la moyenne.

- 2°) Définir une table de hachage dont les éléments seront des (pointeurs sur des) objets de la classe `etudiant`. Le hachage se fera sur le nom de l'étudiant. Les méthodes publiques seront : le constructeur par défaut ; un constructeur ayant un argument entier ; une méthode `insert()` ayant comme argument un (pointeur sur un) objet `etudiant` ; des méthodes `remove()` et `isIn()` ayant comme argument une chaîne de caractères, le nom de l'étudiant ; une méthode `find()` ayant comme argument une chaîne de caractères, renvoyant un (pointeur sur un) objet `etudiant` ; et enfin une méthode `replace()` ayant comme argument un (pointeur sur un) objet `etudiant`, dont le but est, s'il existe déjà

dans la table de hachage une structure dont le nom est celui-ci de celle passée en argument, l'ancienne structure par la nouvelle.

- 3^o) Le programme principal créera la promotion (on demande le nombre d'étudiants prévisibles et on crée la table de hachage) puis fera apparaître un menu :

1. Insertion d'étudiants
2. Retrait d'un étudiant
3. Saisie d'une note
4. Affichage d'une moyenne
5. Quitter

Lors de l'insertion, on insère plusieurs étudiants à la fois (en général en début d'année) en terminant par la valeur sentinelle le mot vide. Par contre on retire en général un étudiant à la fois (en cas de démission). Pour la saisie d'une note, on demandera le nom de l'étudiant, le numéro de la note puis, le coefficient. Pour l'affichage, on saisit le nom d'un étudiant et on affiche la moyenne correspondante.