

Licence Mention Informatique, Programmation avec des Objets

Cours de Robert Cori

Devoir surveillé du 24 avril 2006

*Documents autorisés : sujets, corrigés de travaux dirigés et copies des transparents.
Les livres et photocopies de livres sont interdits. Le barème est indicatif.*

Exercice 1 : Fonctions statiques (3 points)

A tout nombre entier a on associe la somme de ses diviseurs qui sont plus petits que lui. On note ici $\sigma(a)$ cette somme ; par exemple les diviseurs de 6 qui lui sont inférieurs sont 1, 2, 3 leur somme vaut 6. On a aussi $\sigma(8) = 1 + 2 + 4 = 7$ et $\sigma(12) = 1 + 2 + 3 + 4 + 6 = 16$.

On vous demande d'écrire une fonction statique

```
static int sommeDiviseurs(int a)
```

qui a pour résultat la somme des diviseurs de l'entier a .

On rappelle qu'en JAVA pour vérifier que l'entier k divise l'entier a il suffit de tester si $a \% k$ est égal à 0. On ne demande pas d'optimiser le nombre d'opérations effectuées.

Exercice 2 : Chaînes de caractères et tableaux (5 points)

On suppose donné un tableau de chaînes `String[] u` et on souhaite construire un tableau de tableaux de caractères `char[][] w` tel que `w[i]` contienne la chaîne `u[i]` décomposée en caractères et complétée par des symboles - de façon telle que tous les `w[i]` aient même longueur, celle de la plus longue chaîne.

Question 2.1. Ecrire une fonction :

```
static int maxLong(String[] u)
```

qui donne la longueur de la plus longue chaîne parmi les `u[i]`

Question 2.2. Ecrire une fonction :

```
static char[][] transform(String[] u)
```

qui transforme un tableau `u` de chaînes en un tableau `w` de tableaux de caractères tous de même taille contenant la décomposition des `u[i]` complété éventuellement par des symboles -. Ainsi `w[i][j]` est égal au $(j + 1)$ -ième caractère de `u[i]` pour j plus petit que la longueur de `u[i]` et à - pour j supérieur ou égal à cette longueur. On utilisera la méthode `char charAt(i)` qui donne le $i + 1$ -ième caractère de la chaîne d'appel.

Ainsi cette fonction pourra être insérée dans une classe `Exo2` qui pourra donner le résultat suivant après avoir calculé `w`. On ne demande pas d'écrire la fonction `main`, ni la fonction d'affichage de `w` dans cette classe.

```
% java Exo2 Begles Pessac Merignac Gradignan Talence
B e g l e s - - -
P e s s a c - - -
M e r i g n a c -
G r a d i g n a n
T a l e n c e - -
```

Exercice 3 : Construction d'une classe (8 points)

On se propose de construire une classe `Copie` qui pour chaque objet de type `Copie` contienne un entier (numéro d'anonymat), un tableau de nombres de type flottant (note obtenues à chacun des exercices) et une chaîne de caractères contenant les commentaires du correcteur sur la copie. Cette classe devra permettre de réaliser les opérations suivantes :

- Constructeur d'un objet `Copie` à partir d'un entier, une référence sur tableau de flottants (`float[]`) et une chaîne (`String`).
- Méthode (non statique) `float note()` qui calcule la note de la copie en additionnant les notes obtenues aux différents exercices.
- Méthode `String toString()` qui construit une chaîne de caractères contenant le numéro d'anonymat suivi de la note globale et du commentaire.
- Méthode `boolean estMeilleure(Copie c)` telle que `c1.estMeilleure(c2)` a pour résultat `true` si la copie `c1` a une note strictement supérieure que `c2`.

Question 3.1. Ecrire la définition de cette classe, du constructeur et des trois méthodes.

Question 3.2. Ecrire en plus deux fonctions statiques qui seront insérées dans une autre classe :

```
public static float meilleureNote(Copie[] tab)
public static int meilleureNotePar(Copie[] tab)
```

La première donne la valeur de la meilleure note obtenue par une copie du tableau `tab` et la seconde le numéro d'anonymat d'une des copies ayant obtenu cette meilleure note.

Exercice 4 : Listes (4 points)

On se propose de compléter la classe `Liste` étudiée en cours et plus particulièrement les méthodes `boolean estVide()`, `void ajoutDebut(int a)`, `boolean appartient (int a)`, `Liste reste()`, `int valeur()`.

Ecrire une méthode

```
void ajoutListe(Liste v)
```

qui ajoute les éléments d'une liste `v` dans la liste d'appel s'ils n'y sont pas déjà. Par exemple si `u` contient 1, 3, 7, 5, 9 et `v` contient 1, 2, 6, 9 alors `u.ajoutListe(v)` aura pour effet de faire que `u` contienne 6, 2, 1, 3, 7, 5, 9.

Vous écrirez cette méthode de façon qu'elle fasse partie de la classe `Liste`.