



TD Tableaux

Résumé

Ces exercices ont pour but de vérifier que vous avez fixé les tableaux, une structure qui peut contenir plusieurs exemplaires de données similaires.

1	Les tableaux	2
1.1	UtilTabInt	2



1 Les tableaux

N'oubliez pas nos quelques conseils pour vous guider dans la résolution de tels problèmes :

- il convient d'abord de bien comprendre le problème posé ; assurez-vous qu'il est parfaitement spécifié ;
- résolvez le problème via quelques exemples précis ;
- mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;
- n'hésitez pas à faire une ébauche de résolution en français avant d'élaborer l'algorithme définitif pseudo-codé ;
- déclarez ensuite les variables (et leur type) qui interviennent dans chaque module ; les noms des variables risquant de ne pas être suffisamment explicites.
- Écrivez la partie algorithmique **AVANT** de vous lancer dans la programmation en Java.
- Demandez-vous si vous avez besoin de parcourir tout le tableau ou de sortir prématurément (si on a trouvé ce qu'on cherche par exemple).
- Pour la partie Java, dessinez l'arborescence des fichiers.
- **Écrivez le plan de tests en écrivant l'algorithme. Codez les tests après avoir écrit le code Java.**

1.1 UtilTabInt

On vous demande d'écrire une classe `UtilTabInt` qui contiendra des méthodes de classe utiles pour la manipulation de tableaux d'entiers.

- Il convient d'abord de bien comprendre le problème posé ; assurez-vous qu'il est parfaitement spécifié ;
- déclarez ensuite les variables (et leur type) qui interviennent dans l'algorithme ; les noms des variables risquant de ne pas être suffisamment explicites ;
- **mettez en évidence les variables «données », les variables «résultats »et les variables de travail ;**
- n'hésitez pas à faire une ébauche de résolution en français avant d'élaborer l'algorithme définitif pseudo-codé.
- **Écrivez la partie algorithmique AVANT de vous lancer dans la programmation en Java.**
- Demandez-vous si vous avez besoin de parcourir tout le tableau ou de sortir prématurément (si on a trouvé ce qu'on cherche par exemple).
- **Écrivez le plan de tests en écrivant l'algorithme. Codez les tests après avoir écrit le code Java.**

Attention ! Codez, commentez et testez en profondeur (pensez aux cas particuliers) chaque méthode avant de passer à la suivante.

initialiser

1. Écrivez, dans la classe `UtilTabInt`, une méthode de classe `initialiser` qui reçoit en paramètre un tableau d'entiers. Cette méthode initialisera toutes les valeurs du tableau à la valeur de l'indice dans le tableau. Faites bien attention aux cas particuliers que vous pouvez rencontrer. Que se passe-t-il si le tableau est null ? Et si il est vide ?
2. Dans la méthode principale de la classe `Test`, créez un tableau d'entiers de taille 10 et appelez la méthode `initialiser` ci-dessus pour initialiser votre tableau. Testez également les cas particuliers.
3. Écrivez déjà la **javadoc** pour la méthode `initialiser`. N'oubliez pas de la générer dans un répertoire nommé `javadoc` et de vérifier le résultat obtenu.

afficher

1. Écrivez, dans la classe `UtilTabInt`, une méthode de classe `afficher` qui reçoit en paramètre un tableau d'entiers et qui affiche tous les éléments du tableau.
2. Dans la méthode principale de la classe `Test`, appelez la méthode `afficher` pour afficher les éléments du tableau que vous avez créé.
3. Vous avez encore ajouté une méthode et modifié votre méthode `main`. N'oubliez pas de **recompiler** vos classes, d'**exécuter** votre programme et de compléter la **javadoc**. À partir de maintenant, je ne vous le rappellerai plus !

sommer

Écrivez une méthode `sommer` qui reçoit un tableau d'entiers et retourne la somme de ses éléments.

maximum

Écrivez une méthode `maximum` qui reçoit un tableau d'entiers et retourne la valeur maximale rencontrée.

indiceMax

Écrivez une méthode **indiceMax** qui reçoit un tableau d'entiers et retourne l'indice de la valeur maximale rencontrée. Si la valeur maximale est présente plusieurs fois, on retourne l'indice le plus petit.

créer

Écrivez une méthode **créer** qui reçoit en paramètre un entier représentant la taille du tableau et qui retourne le tableau d'entiers créé de la taille donnée et dont les éléments sont initialisés à 0. Aide : Pensez aux valeurs initiales des éléments d'un tableau.

estTrié

Écrivez une méthode **estTrié** qui indique si le tableau reçu en paramètre est trié ou pas.

créer

Écrivez une méthode **créer** qui reçoit en paramètre deux entiers

- un entier représentant la taille du tableau
- et un entier représentant la valeur initiale des éléments du tableau

et qui retourne le tableau d'entiers

- créé de la taille donnée
- et dont les éléments sont initialisés à la valeur du deuxième paramètre reçu.

inverser

Écrivez une méthode **inverser** qui inverse les éléments du tableau reçu en paramètre. Le premier élément est échangé avec le dernier et ainsi de suite.

positionsMin

Écrivez une méthode **positionsMin** qui **retourne** un tableau donnant les positions de toutes les valeurs minimales du tableau reçu en paramètre.

Pour plus d'exercices, révisez ici (www.heb.be/esi/InitTableau/fr/../../../../TDDTableau/fr/html/Exercices_learningObject2.html)