

Exercice : indigage naturel

Consigne

Modalités de rendu

Vous devez me rendre le fichier **.java** correspondant à votre travail via la plateforme Moodle. Veuillez tout mettre dans un seul et unique fichier java, ne pas faire de package ou d'import non inclus dans le JDK.

**** Pensez à ajouter dans votre code votre numéro d'auditeur, commençant par GET ****

Notation

La notation se basera sur :

- Le caractère fonctionnel de votre code (compilation et exécution) ;
- Le respect de la spécification ci-dessous ;
- La clarté et la propreté de votre code ;
- L'application des notions vues en cours nécessaires pour réaliser cet exercice.

N'hésitez surtout pas à me poser des questions sur Teams. Le barème est disponible [ici](#).

Spécification de l'exercice

But de l'exercice

En JAVA, comme dans d'autres langages, les tableaux sont indicés de *0 à taille-1*, où *taille* correspond à la longueur du tableau. Naturellement, en tant qu'humain, il nous paraît plus logique de parler d'indices allant de 1 à *taille*.

Ainsi il peut être intéressant, en conservant l'utilisation des tableaux basiques, de faire des méthodes qui masquent ce problème d'indice. Pour ce faire, nous avons besoin de plusieurs méthodes :

1. Une méthode d'affichage du tableau
2. Une méthode qui insère un élément dans un tableau
3. Une méthode qui supprime un élément d'un tableau

Une fois ces méthodes créées, l'utilisateur pourra les utiliser à sa guise sans se préoccuper de l'indigage machine de *0 à taille-1*.

Exemple

Soit le tableau `tab={'c','n','a','m'}`.

Si on appelle la méthode d'insertion avec la lettre `w` à l'indice 2, on obtiendra le tableau suivant `{'c','w','n','a','m'}`.

On a ajouté `'w'` à la deuxième position du tableau qui correspond à l'indice 1 en JAVA.

S'il avait précisé l'indice 5, nous aurions obtenu `{'c','n','a','m','w'}`

Dans le cadre de cet exercice, nous travaillerons avec un **tableau de char**.

À vous de jouer !

Voici l'ensemble des sous-programmes demandés :

- Une méthode qui affiche un tableau
- Une méthode qui ajoute un élément à un tableau sans préciser d'indice. L'insertion se fera alors à la fin du tableau.
- Une méthode qui ajoute un élément à un tableau en précisant un indice. **Attention** on doit pouvoir saisir des indices dits "naturels" comme vu plus haut dans l'exemple.
- Une méthode qui supprime un élément d'un tableau à un indice. **Attention** on doit pouvoir saisir des indices dits "naturels" comme vu plus haut dans l'exemple.
- Une méthode qui supprime un élément d'un tableau en précisant l'élément que l'on souhaite supprimer. Si l'élément n'est pas trouvé, on le précisera via un message.

Une fois toutes ces méthodes implémentées, vous pouvez les tester au sein de la méthode *main* une à une, en précisant le résultat console attendu en commentaire, si toutefois il est sensé en avoir un .

Exemple

```
display(tabToTest); // c | n | a | m | w |
```

Remarques importantes

- Les méthodes doivent porter sur des tableaux et être totalement native. **Vous ne devez pas utiliser la classe Arrays, ArrayList ou tout autre librairie.**
- Chaque méthode devra être sous la forme d'un sous-programme avec un nom explicite.
- Concernant les méthodes qui prennent un indice, vous rajouterez un test qui permet de s'assurer que l'indice est valable, c'est-à-dire dans le tableau.