

# M3103 Algorithmique avancée – TP 1

## Implémentation d'une file

Nous allons implémenter une file d'entiers en Java en respectant l'API `java.util.Queue<Integer>`.

Nous souhaitons que cet implémentation utilise un *tableau circulaire*, c'est-à-dire un tableau d'entiers, avec un indice de **tete** et une **longueur** (nombre d'éléments dans la file). L'indice de *tete* pointe toujours sur l'élément de tête (si la file est non vide), et se déplace vers la droite lorsqu'on supprime un élément. De même, l'insertion ajoute l'élément à droite du dernier élément inséré. L'aspect "circulaire" vient du fait que, si cet indice d'insertion dépasse la capacité du tableau, on recommence à insérer dans la première case du tableau (si elle est libre).

Exemple avec une file de capacité 4 :

Situation initiale :					tete=0, longueur=0 (la file est vide)
Insertion de 7 :	7				tete=0, longueur=1
Insertion de 2 :	7	2			tete=0, longueur=2
Insertion de 6 :	7	2	6		tete=0, longueur=3
Suppression :		2	6		tete=1, longueur=2
Suppression :			6		tete=2, longueur=1
Insertion de 4 :			6	4	tete=2, longueur=2
Insertion de 9 :	9		6	4	tete=2, longueur=3
Insertion de 3 :	9	3	6	4	tete=2, longueur=4 (la file est pleine)
Suppression :	9	3		4	tete=3, longueur=3
Suppression :	9	3			tete=0, longueur=2
Suppression :		3			tete=1, longueur=1
Suppression :					tete=2, longueur=0 (la file est vide)
Insertion de 8 :			8		tete=2, longueur=1

Remarques :

- pour passer à la case "suivante", il suffit d'incrémenter modulo la taille du tableau
- en cas d'insertion, connaissant *tete* et *longueur*, il n'est pas bien compliqué de trouver l'indice où doit avoir lieu l'insertion.

1) Créez une classe Java :

- se nommant `FileTableauCirculaire`,
- faisant partie du package `implemFile`,
- implémentant l'interface `java.util.Queue<Integer>` et
- qui compile (créez automatiquement les squelettes des fonctions).

Faites tourner les tests unitaires de la classe `FileTableauCirculaireTests` disponible sur Moodle. Vous aurez peut-être besoin d'ajouter JUnit4 parmi les bibliothèques de votre projet.

2) Implémentez les fonctions `contains`, `add` et `remove()` (pas `remove(Object o)`). Votre file aura une capacité maximale de 1000 entiers. Vérifiez que les tests unitaires suivants sont validés : `testContains`, `testAdd`, `testAddException1`, `testAddException2`, `testRemove`, `testRemoveException1`, `testAddRemove`, `testFillEmpty`.

3) [facultatif] Complétez toutes les fonctions, et vérifiez que tous les tests sont validés.



Ce document est publié sous Licence Creative Commons « By-NonCommercial-ShareAlike ». Cette licence vous autorise une utilisation libre de ce document pour un usage non commercial et à condition d'en conserver la paternité. Toute version modifiée de ce document doit être placée sous la même licence pour pouvoir être diffusée.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>