

# Buffer, Charset & Channel

## Exercice 1 - Compteur de lignes

On cherche à écrire une application permettant de compter le nombre de lignes d'un fichier (en fait le nombre de '\n') contenus dans un fichier. Le fichier sera pris en tant que premier argument de la ligne de commande.

- 1 Écrire la classe `IOLineCounter` utilisant les classes du paquetage `java.io` pour compter les lignes.  
Penser que par défaut, l'accès au fichier n'est pas bufferisé.
- 2 Quelle est la différence entre un buffer (`java.nio.ByteBuffer`) alloué en utilisant `allocate` et un buffer alloué en utilisant `allocateDirect()`.  
Quel est le buffer le plus efficace si l'on veut utiliser celui-ci pour coder un compteur de lignes ?
- 3 Lorsque l'on effectue un parcours d'un buffer doit-on faire une boucle en utilisant `get()` ou `get(int)` pour que celle-ci soit la plus efficace ?
- 4 Écrire la classe `NIOLineCounter` effectuant la même tâche mais en utilisant les classes `java.nio.channels.FileChannel` et `java.nio.ByteBuffer`.

## Exercice 2 - Compteur de lignes suite

- 1 On souhaite maintenant écrire une version du compteur de mot utilisant un buffer mappé en mémoire `MappedByteBuffer`.
- 2 Indiquer dans quels cas il est plus intéressant d'utiliser `read` que d'utiliser un fichier mappé en mémoire ?
- 3 Quelle est le mode (`FileChannel.MapMode`) que l'on doit choisir pour monter le fichier en mémoire ?
- 4 Implanter la classe `MappedLineCounter`.

## Exercice 3 - Transcoder

On souhaite écrire la classe `Transcoder` qui effectue la conversion d'un format de charset à un autre :

```
java Transcoder srcFile.txt UTF-8 dstFile.txt ISO-8859-1
```

Dans cet exemple, le transcoder lit le fichier `srcFile.txt` écrit en UTF-8 et transforme celui-ci en ISO-8859-1.

Dans le cas où le codage spécifié est "default" alors le programme utilisera le codage par défaut de la plateforme.

- 1 Rappeler ce qu'est un charset et pourquoi un tel concept existe.
- 2 Utiliser la classe `java.nio.charset.Charset` pour obtenir un charset puis créer l'encodeur et le décodeur correspondant.
- 3 Compléter la classe `Transcoder` en utilisant les méthodes `decode(ByteBuffer)` et `encode(CharBuffer)`.
- 4 Quel est le problème d'une telle implantation ?

Changer votre code pour utiliser les méthodes  
`decode(ByteBuffer, CharBuffer, boolean)` et

`encode(CharBuffer, ByteBuffer, boolean)`.

Comment dimensionner les différents buffers pour être sûr que, dans le pire cas, ce que  
`encode` ou `decode` génère tienne dans le buffer de sortie ?