

# Java avancé - TP 7 - Streams

Romain Péchoux

19 janvier 2010

## Exercice : Sérialisation

Ecrire une classe `Paysan` avec un age, un nom et un prénom, un constructeur public et un accesseur public `getAge`. Dans une classe test, sérialiser 3 paysans “Raymond”, “Robert” et “André” dans un fichier “temp.txt”. Récupérer et afficher l’age des trois paysans en utilisant la désérialisation.

## Problème :

Voici le code source d’une classe java `Copieur` dont la méthode `copier(Reader source, Writer destination)` copie un flot de caractères d’un stream d’entrée vers un stream de sortie :

```
import java.io.*;
public class Copieur {
    public void copier(Reader source, Writer destination) throws IOException
    {
        // character stream
        int c;
        while ((c = source.read()) != -1) { destination.write(c); }
    }
}
```

Vous souhaitez permettre à un objet (dont vous ne connaissez pas la classe à l’avance) d’être informé à chaque fois qu’un certain nombre de caractères ont été copiés de la source vers la destination, par exemple à chaque fois qu’un bloc de 200 caractères a été copié (le nombre de caractères doit être une variable de l’objet). Par exemple, une interface graphique qui utilise ce copieur pourrait afficher une barre de progression pour représenter visuellement à l’utilisateur l’avancement de la copie. Il ne doit pas être possible de donner une taille de bloc non positive (toute taille négative ou nulle sera tout simplement ignorée).

Pour qu’un objet soit informé, il faut qu’il s’inscrive auprès du copieur. Les objets informés seront appelés des listeners (ou écouteurs) du copieur (par analogie avec les listeners étudiés dans le cours sur les interfaces graphiques). Un copieur peut avertir plusieurs objets s’ils se sont inscrits auprès de lui.

- Modifier le code de la classe `Copieur` et ajoutez éventuellement des classes et des interfaces pour avoir les fonctionnalités décrites ci-dessus et dans les questions suivantes (lire l'énoncé de cet exercice jusqu'au bout avant de commencer à coder).
- Ecrire une classe dont les instances seront des écouteurs qui seront informés par le copieur écrit à la question précédente. Une instance de cette classe affichera un caractère, par exemple le caractère "+", sur la sortie standard à chaque fois qu'elle sera informée par le copieur qu'un bloc de caractères aura été copié. Le caractère affiché dépend de l'instance de la classe; ça peut être "+" ou tout autre caractère.
- Écrire une méthode main qui crée un copieur et 2 écouteurs du copieur (de la classe écrite dans la question précédente) et démarre la copie. Un des écouteurs souhaite être prévenu tous les 50 caractères et affiche le caractère "+" (à chaque copie de 50 caractères) et l'autre affiche le caractère "\*" et souhaite être prévenu tous les 70 caractères. Arrangez vous pour que la copie se fasse en utilisant un buffer afin de réduire les lectures-écritures réelles.
- Si le copieur lève des exceptions, vers quelle méthode vont-elles remonter?
- Autres consignes à respecter :
  - Mettre toutes les classes dans un paquetage. Cependant, la classe `Copieur` et les écouteurs ne devront pas être dans le même paquetage.
  - Ne pas utiliser d'évènements (au sens interface graphique).
  - Fermer les ressources quand il le faut et tenir compte des exceptions.
  - Ne pas capturer les exceptions dans le code.

## Problème (add on) :

Reprendre l'exercice précédent. Cette fois-ci, on souhaite que les écouteurs soient prévenus aussi dans les 2 cas suivants :

- La fin du fichier est rencontrée avant que le nombre de caractères associé à l'écouteur ne soit atteint. Par exemple si la fin du fichier est rencontrée alors que seulement 5 caractères viennent d'être lus depuis la lecture du dernier bloc entier de 50 caractères pour le premier écouteur de l'exercice précédent. Dans ce cas, l'écouteur doit pouvoir savoir (directement ou par un calcul) combien de caractères ont été copiés depuis la dernière fois qu'il a été prévenu. De même, si une exception a été levée (au lieu de la fin de fichier). Attention, en ce cas, l'exception doit toujours être levée par la méthode copier pour que la méthode qui a appelé copier la reçoive.
- On souhaite aussi que les écouteurs de l'exercice précédent affichent leur caractère entre "!" pour indiquer la fin de la copie (par exemple "!!+!").

Modifier les classes de l'exercice précédent pour pouvoir obtenir le comportement décrit.