

# Entrées-sorties en Java

Alain Giorgetti

1 Février 2000

## Flots de données, canaux

En Java, les entrées et les sorties sont organisées en « canaux » (anglais : *stream*) qui contrôlent des « flots » (ou « flux ») de données.

• *Flot* : source ou destination de données, progressivement disponible dans le temps, de manière permanente ou temporaire,

• *Canal* : objet Java qui prend le contrôle d'un flot en le découplant en éléments d'information (de même taille ou de taille différente)

Paquetage des classes de canaux : **java.io**

• Analogie hydraulique :

- ◆ *flot* : source ou puits
- ◆ *canal* : tuyau, conduite

## Fichier texte

Un *fichier texte* est une séquence de *caractères* organisés en *lignes*

La fin d'une *ligne* est indiquée par :

- ◆ '\n' dans les systèmes Unix et Linux
- ◆ '\r' + '\n' dans les systèmes MS-DOS et Windows
- ◆ **System.getProperty("line.separator")** en Java

Un *caractère* est codé sur :

- ◆ un octet (ASCII étendu) : classes **java.io.\*Stream**
- ◆ 2 octets (UNICODE) : classes **java.io.\*Reader** et **java.io.\*Writer**

Il existe d'autres sortes de fichiers (binaires, fichiers de données objet, ...). Dans ce cours, on se restreint à la lecture et à l'écriture de fichier texte.

## La classe java.io.FileReader

- Objet de la classe **FileReader** : canal d'entrées associé à un fichier texte
- Constructeur

```
public FileReader (String fileName)
    throws FileNotFoundException
```

- Hérité des méthodes suivantes de la classe **InputStreamReader**:

➤ **close()** Ferme le canal

➤ **getEncoding()** Retourne le nom du codage des caractères utilisés dans ce canal

➤ **read()** Lit un caractère

➤ **read(char[] j, int, int)** Lit une suite de caractères dans une portion de tableau