

## ANNEXE - 13 Résumé package java.io

- On définit les entrées / sorties en termes de flots de données ( streams )

Flot de données : c'est une suite ordonnée d'infos qui provient d'une source ( input stream ) ou qui se dirige vers une destination ( output stream )

Fichier → flot d'entrée /lecture → Programme

Programme → flot de sortie / écriture → Fichier

De plus, on peut séparer les flux de données par rapport à leur contenu. 2 grandes catégories existent :

Flots d'octets ( `FileInputStream`, `FileOutputStream` ... ) pour travailler avec des Images, du Son ou pour faire de la sérialisation d'objets

Flots de caractères : ( `FileReader`, `FileWriter`, `StringReader`... ) pour travailler avec des caractères

\*\*\* On utilise toujours ces flux de communication en collaboration avec un flux de traitement, un Buffer ( tampon ) → permet de régulariser l'accès aux données, évite que chaque lecture d'octet ou de caractère soit lu directement du fichier texte. Permet d'accumuler une ligne de caractères entre autres.

Les principaux flux de traitement : `BufferedReader`, `BufferedWriter`, `BufferedInputStream`, `BufferedOutputStream`

Exercice : Lire un texte et l'afficher dans JDeveloper

Exercice : Compter le nombre de 'e' dans le fichier texte

En écriture

- Importance du `flush`, `flush` appelé par `close` mais nécessaire quand on lit et écrit simultanément
- Paramètre `append` au `FileWriter`