

## Exemple d'utilisation de FileReader/FileWriter

```
import java.io.*;

public class Copy {
    public static void main(String[] args) throws IOException {
        File inputFile = new File("farrago.txt");
        File outputFile = new File("outagain.txt");

        FileReader in = new FileReader(inputFile);
        FileWriter out = new FileWriter(outputFile);
        int c;

        while ((c = in.read()) != -1)
            out.write(c);

        in.close();
        out.close();
    }
}
```

## BufferedReader et PrintWriter

- utiles pour la lecture et l'écriture de fichiers textes.

```
BufferedReader in = new BufferedReader(new FileReader("fichier.txt"));
String str;
while ((str=in.readLine()) != null) {
    // faire quelque chose avec la ligne
}
in.close();

// Pour la lecture au clavier
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

PrintWriter out = new PrintWriter(new FileWriter("fichier.txt"));
out.println(String s)
out.print(String s)
out.close();
```

## La classe File

- Une classe qui permet de manipuler des fichiers et des répertoires sans toutefois accéder aux données dans les fichiers.

```
File myFile = new File("/u/user/ift6810/fichier.txt");

// Si on se trouve dans le répertoire /u/user/ift6810
String relative = myFile.getPath(); // retourne "fichier.txt"

String absolute = myFile.getAbsolutePath();
// retourne "/u/user/ift6810/fichier.txt"

boolean there = myFile.exists(); // vrai ou faux si le fichier existe ou non
boolean checkDir = myFile.isDirectory() // si le fichier est un répertoire

long myLength = myFile.length(); // taille du fichier.

// retourne les différents fichiers dans un répertoire
String[] allfiles = myFile.list();
```

# ObjectInputStream et ObjectOutputStream

- permettent de lire et écrire des objets sur un stream.
- Un objet doit implémenter l'interface Serializable pour pouvoir être écrit ou lu.

L'interface Serializable contient les deux méthodes suivantes :

```
private void readObject(java.io.ObjectInputStream stream)
    throws IOException, ClassNotFoundException;
private void writeObject(java.io.ObjectOutputStream stream)
    throws IOException;
```

Exemple :

```
class Etudiant implements Serializable {
    String nom;
    String prenom;
    double note;

    Etudiant(String n, String p, double n) {
        nom = n;
        prenom = p;
        note = n;
    }

    private void readObject(java.io.ObjectInputStream stream) {
        nom = (String) stream.readObject();
        prenom = (String) stream.readObject();
        note = stream.readDouble();
    }

    private void writeObject(java.io.ObjectOutputStream stream) {
        stream.writeObject(nom);
        stream.writeObject(prenom);
        stream.writeDouble(note);
    }
}

Etudiant e = new Etudiant("smith", "john", 86);

FileOutputStream fos = new FileOutputStream("t.tmp");
ObjectOutputStream oos = new ObjectOutputStream(fos);
oos.writeObject(e);
oos.close();

FileInputStream fis = new FileInputStream("t.tmp");
ObjectInputStream ois = new ObjectInputStream(fis);
Etudiant e2 = (Etudiant) ois.readObject();
ois.close();
```

# StreamTokenizer

- Permet de lire à partir d'un stream de caractères et de séparer l'information en tokens.

```
StreamTokenizer(Reader r) // Constructeur
int nextToken(); // Parser le prochain token

nval // champ indiquant la valeur numérique du token si le token NUMBER
sval // champ indiquant la valeur textuelle du token si le token WORD
ttype // Le type du token
```

## Les types :

```
StreamTokenizer.TT_EOF
StreamTokenizer.TT_EOL
StreamTokenizer.TT_NUMBER
StreamTokenizer.TT_WORD
```

```
// Read space or comma-delimited file of integers
// Convert from ASCII to internal binary
import java.io.*;
import java.io .StreamTokenizer;

BufferedReader in = new BufferedReader(new FileReader("/home/user/foo.txt" ) );
StreamTokenizer st = new StreamTokenizer(in);
// process the entire file, of space or comma-delimited ints
int found = StreamTokenizer.TT_NUMBER;
while ( found != StreamTokenizer.TT_EOF )
{
    found = st.nextToken ();
    if ( found == StreamTokenizer.TT_NUMBER )
    {
        int i = ( int) st.nval;
        System.out.println (i);
    } // end if
} // end while
```