

A) Lecture d'un fichier texte :

À partir d'un programme java, on peut lire un fichier texte. On suppose que le fichier texte à lire est situé dans le même dossier (projet) que le programme.

```
// inclure l'importation suivante
import java.io.*;

// dans main, préciser qu'il peut y avoir des erreurs d'entrées/sorties
public static void main(String args[]) throws IOException {

// il faut déclarer le fichier et son tampon de lecture
FileReader fich = new FileReader("fich.txt");
BufferedReader fichier = new BufferedReader(fich);

    ou

BufferedReader fichier = new BufferedReader(new FileReader("fich.txt"));

// pour lire une ligne du fichier
String ligneLue = fichier.readLine();

// boucle tant que la ligne lue n'est pas null car alors le fichier est terminé
// il ne faut pas oublier de relire à la fin de la boucle while
while (ligneLue != null)

// lorsque la lecture est terminée, fermer le fichier
fichier.close();
```

B) Écriture d'un fichier texte :

À partir d'un programme java, on peut écrire un fichier texte. On suppose que le fichier texte à écrire est situé dans le même dossier (projet) que le programme.

```
// inclure l'importation suivante
import java.io.*;

// préciser qu'il peut y avoir des erreurs d'entrées/sorties
public static void main(String args[]) throws IOException {

// il faut déclarer le fichier et son tampon d'écriture
FileWriter fich = new FileWriter("fich.txt");
BufferedWriter fichier = new BufferedWriter(fich);

    ou

BufferedWriter fichier = new BufferedWriter(new FileWriter("fich.txt"));

// pour écrire une ligne dans le fichier après avoir
// rempli la ligne avec ce qu'on veut écrire
fichier.write(ligne);
fichier.newLine();

// lorsque l'écriture est terminée, fermer le fichier
fichier.close();
```

dans une boucle pour écrire
plusieurs lignes !

On utilise normalement une méthode pour lire ou écrire un fichier. Ceci permet de dégager la logique principale. La méthode main devra quand même avoir `throws IOException` puisqu'elle appelle une méthode ayant `throws IOException` dans sa signature.

Exemple :

```
// Méthode permettant la lecture et l'écriture d'un fichier texte
static void lireEcrire() throws IOException
{
    char    code;           // A = désaltérante, D = dégustation
    String  ligneEntree,    // ligne lue
           ligneSortie,    // ligne à écrire
           nom,             // nom de la bière
           pays;            // pays d'origine
    double  alcool,         // % d'alcool
           prix;            // prix au détail

    // description des fichiers d'entrée et de sortie
    BufferedReader ficEntree = new BufferedReader(
        new FileReader("fichIn.txt"));
    BufferedWriter ficSortie = new BufferedWriter(
        new FileWriter("fichOut.txt"));

    // redéfinition du symbole pour le séparateur des décimales
    // car certains systèmes utilisent la virgule comme séparateur
    DecimalFormatSymbols pointDecimal = new DecimalFormatSymbols();
    pointDecimal.setDecimalSeparator('.');

    // définition des formats des nombres réels pour l'écriture
    DecimalFormat formatAlcool = new DecimalFormat("0.0", pointDecimal);
    DecimalFormat formatPrix  = new DecimalFormat("0.00", pointDecimal);

    // lecture de la première ligne du fichier d'entrée
    ligneEntree = ficEntree.readLine();

    while (ligneEntree != null)    // tant que pas la fin du fichier
    {
        code    = ligneEntree.charAt(0);
        nom     = ligneEntree.substring(1, 30);
        pays    = ligneEntree.substring(31, 42);
        alcool  = Double.parseDouble(ligneEntree.substring(43, 46));
        prix    = Double.parseDouble(ligneEntree.substring(47));

        // écriture des données dans le fichier de sortie
        ligneSortie = code + nom + " " + pays + " " +
            formatAlcool.format(alcool) + " " +
            formatPrix.format(prix);
        ficSortie.write(ligneSortie);
        ficSortie.newLine();

        // lecture de la ligne suivante, dans le fichier d'entrée
        ligneEntree = ficEntree.readLine();
    }

    ficEntree.close();
    ficSortie.close();
}
```

Contenu du fichier

fichIn.txt

Le fichier écrit

fichOut.txt

a exactement le même contenu

AMort Subite Kriek Lambic	Belgique	4.0	4.65
DPentagel Superior English Ale	Angleterre	4.5	3.15
Ala Trappe	Hollande	8.0	2.90
AGuinness Pub Draught	Irlande	4.8	2.40
ABishop's Finger Ale	Angleterre	5.4	3.85
DBlanche de Bruxelles	Belgique	4.5	2.55
DDouglass Scotch Ale	Ecosse	8.6	3.30
ASEigneuriale Réserve	Québec	7.5	4.30