

Utilitaires d'entrées-sorties

```
public class Ecriture {
    // quelques primitives d'écriture à l'écran ou dans un fichier

    private static void erreur (IOException e)
        // affichage d'un message précisant l'exception levée puis arrêt de l'exécution
    public static OutputStream ouvrir (String nomFich)
        // délivre une référence sur le fichier de nom nomFich (null si erreur)
    public static void fermer (OutputStream f)
        // fermeture d'un fichier
    public static void ecrireChar (OutputStream f, char c)
        // écriture d'un caractère dans le fichier f
    public static void ecrireChar (char c)
        // écriture d'un caractère à l'écran
    public static void ecrireString (OutputStream f, String s)
        // écriture d'une chaîne dans le fichier f
    public static void ecrireString (String s)
        // écriture d'une chaîne à l'écran
    public static void ecrireStringln (OutputStream f, String s)
        // écriture d'une chaîne et de \n dans le fichier f
    public static void ecrireStringln (String s)
        // écriture d'une chaîne à l'écran et passage à la ligne suivante
    public static void ecrireStringln (OutputStream f)
        // écriture de \n dans le fichier f
    public static void ecrireStringln ( )
        // passage à la ligne suivante de l'écran
    public static void ecrireInt (OutputStream f, int x)
        // écriture d'un entier, sans formatage, dans le fichier f
    public static void ecrireInt (int x)
        // écriture d'un entier, sans formatage, à l'écran
    public static void ecrireInt (OutputStream f, int x, int longueur)
        // écriture d'un entier dans le fichier f, en utilisant longueur caractères
    public static void ecrireInt (int x, int longueur)
        // écriture d'un entier à l'écran, en utilisant longueur caractères
    public static void ecrireDouble (OutputStream f, double d)
        // écriture d'un réel, sans formatage, dans le fichier f
    public static void ecrireDouble (double d)
        // écriture d'un réel, sans formatage, à l'écran
} // class Ecriture
```

```
public class Lecture {
    // quelques primitives de lecture au clavier ou dans un fichier

    public static InputStream ouvrir (String nomFich)
        // délivre une référence sur le fichier de nom nomFich (null si erreur)
    public static boolean finFichier (InputStream f)
        // détermine si la fin du fichier est atteinte
    public static void fermer (InputStream f)
        // ferme un fichier (affiche un message si problème)
    public static char lireChar (InputStream f)
        /* lit le prochain octet dans le fichier f et capture l'exception en cas de tentative
        de lecture après la fin de fichier (⇒ arrêt de l'exécution) */
    public static char lireChar ()
        // lit un caractère saisi au clavier
    public static String lireString (InputStream f)
        /* lit, dans le fichier f, une chaîne terminée par un line feed (précédé ou non de
        retour-chariot) */
    public static String lireString ()
        // lit une ligne saisie au clavier
    public static int lireInt (InputStream f)
        /* lit, dans le fichier f, un entier éventuellement précédé par des blancs et suivi
        d'un blanc */
    public static int lireInt ()
        // idem que lireInt(f) mais pour un entier saisi au clavier
    public static int lireIntln (InputStream f)
        // idem lireInt(f) mais le reste de la ligne est ignoré
    public static int lireIntln ()
        // idem lireInt() mais le reste de la ligne est ignoré
    public static double lireDouble (InputStream f)
        /* lit, dans le fichier f, un réel éventuellement précédé par des blancs et suivi
        d'un blanc */
    public static double lireDouble ()
        // idem que lireDouble(f) mais pour un réel saisi au clavier
    public static double lireDoubleln (InputStream f)
        // idem lireDouble(f) mais le reste de la ligne est ignoré
    public static double lireDoubleln ()
        // idem lireDouble() mais le reste de la ligne est ignoré
    public static String lireUnite (InputStream f, boolean alaligne)
        /* lit, dans le fichier f, une unité (séquence encadrée par des blancs), puis ignore
        éventuellement la fin de la ligne (si alaligne vaut true) */
    public static String lireUnite (boolean alaligne)
        // idem mais l'unité est saisie au clavier
} // class Lecture
```