

TD n° 1

Java : Les flots

Exercice 1

Soit un fichier texte au nom de « fichier.txt » créé avec le code suivant :

```
public class Ex01 {  
    public static void main(String [] args) throws Exception {  
        FileOutputStream fos = new FileOutputStream("fichier.txt");  
        fos.write(10);  
        fos.write(11);  
        fos.write(12);  
        fos.write(13);  
        fos.write(14);  
        fos.write(15);  
        fos.close();  
    }  
}
```

On suppose que l'on a créé un buffer b contenant 256 octets.

1. Ecrivez le code permettant de créer le même fichier en utilisant :
 - a. La méthode write(x) // x un entier
 - b. La méthode write(b) // b un buffer d'octets
 - c. La méthode write(b, x, y)
2. Ecrivez le code permettant d'afficher les octets en utilisant :
 - a. La méthode read()
 - b. La méthode read(b)
 - c. La méthode read(b, x, y)

Exercice 2

Proposez une méthode permettant de lire n octets exactement à partir de offset sur un `InputStream`.

```
public int readN(byte [] b, int offset, int n)
```

Cette méthode doit utiliser la méthode `read(byte [] b, int offset, int len)`

Exercice 3 (examen 2017)

Soit le code suivant permettant de créer un fichier contenant 6 octets (10, 11, 12, 13, 14 et 15) :

CODE 1

```
1: FileOutputStream fos = new FileOutputStream("test.txt");  
2: byte [] b1 = {10, 11, 12, 13, 14, 15};  
3: fos.write(b1);  
4: fos.close();
```


UBO

Donnez le résultat d'exécution du **CODE 2** suivant :

CODE 2

```
1: FileInputStream fis = new FileInputStream("test.txt");
2: int n = 2;
3: byte [] b2 = new byte [n];
4: int x ;
5: while((x=fis.read(b2)) != -1) {
6:     System.out.println(x);
7:     for(int i=0; i<n; i++) {
8:         System.out.print(b2[i]+" ");
9:     }
10:    System.out.println();
11: }
12: fis.close();
```

Donnez le résultat d'exécution du **CODE 2**, pour :

- a. n=4 (ligne 2)
- b. n=10 (ligne 2).

Remplacez la ligne 3 du **CODE 1** ci-dessus par la ligne suivante :

```
3: fos.write(b1,1,4);
```

Ensuite, donnez le résultat d'exécution du **CODE 2**, pour :

- c. n=3 (ligne 2)

Exercice 4

Soit un flot de type `FileInputStream` où le fichier est de type texte. Nous voudrions lire ce fichier ligne par ligne en utilisant la méthode `readLine()`. Quelles sont les classes permettant de lire les flots texte ligne par ligne ?

Nous proposons d'utiliser la classe `BufferedReader`. Ecrivez le code permettant de lire un flot de type `FileInputStream` ligne par ligne.

Exercice 5

Écrire la méthode :

```
public void convert(InputStreamReader in, OutputStreamWriter out)
qui permet de convertir un fichier codé en Cp850 en un fichier codé en ISO8859_1
```

Exercice 6

Le but de cet exercice est de remplir un tableau d'octets statique avec la même taille que celle du flot.

Construire un tableau `byte tab[]` à partir d'un flot d'octets `InputStream in`.

Indication : utiliser la classe `ByteArrayOutputStream`.

Exercice 7

Soit une chaîne de caractères `str`. Afficher ses caractères un par un en utilisant les flots.

Exercice 8

Soit un `InputStream in`. Ecrivez le code qui permet de transformer `in` en une chaîne de caractères `String s`.

Indications : Créer un `String` à partir d'un tableau de `byte [] b`.

Exercice 9

Ecrivez un exemple qui permet d'écrire trois caractères dans un *pipe*. Puis, afficher les résultats de lecture de ce *pipe*.

Exercice 10

Programmer la commande Unix "`ls -l`" pour un répertoire donné.

Exercice 11

Afficher les lignes, d'un fichier, précédées chacune d'elle de son numéro. Pour ce dernier, utiliser la méthode `getLineNumber()`.

Exercice 12

Soit un entier `i=10` et un réel `x=3.14`. Mettez dans un tableau d'octets `byte [] tab` ces deux numéros.