

TP1

Gestion des flux

Exercice1 : Lire et d'écrire dans un fichier

L'objectif de cet exercice est de lire et d'écrire dans un fichier. Pour cela, il vous est demandé de suivre les étapes suivantes :

- 1) créer un nouveau projet sur Eclipse :
 - a. File/new/Java Project,
 - b. puis nommer le projet créé Tp1File
- 2) créer une nouvelle classe et nommer la Fichier.java
 - a. clique bouton droit sur le nom du projet Tp1File, puis cliquer sur new/class
- 3) créer une autre nouvelle classe, nommer la Main.java.
- 4) Ecriture de la classe Fichier.java

a. Variable d'instance:

On représentera le nom du fichier dans une variable de classe (static), qui se nomme name et qui est de type String.

b. Méthodes :

- i. La méthode void afficherFichier() permet de lire le fichier ligne par ligne et de l'afficher sur l'écran

Fichier
-static name: String
+static afficherFichier() +static ecrireFichier(s: String) +static afficherObjectFichier(): Object +static ecrireObjectFichier(o: Object)

```
public static void afficherFichier() throws FileNotFoundException{
    InputStream is;
    try {
        is = new FileInputStream(name);
        byte[] b = new byte[50];
        while(is.read(b) != -1) {
            System.out.println(new String(b));
        }
        is.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

- ii. La méthode ecrireFichier(String s) permet d'écrire un string dans un fichier.

```
public static void ecrireFichier(String s){
    OutputStream os;
    try {
        os = new FileOutputStream(name);
        s = s + "\n";
        os.write(s.getBytes());
        os.close();
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

- iii. La méthode `Object afficherObjectFichier()`, permet de lire du fichier un objet sérialisé, puis de l'afficher à l'écran.

```

public static Object afficherObjectFichier()
    throws FileNotFoundException{
    ObjectInputStream ois;
    try {
        ois = new ObjectInputStream(new FileInputStream(name));
        Object o;
        o = ois.readObject();
        System.out.println(o);
        ois.close();
        return o;
    } catch (IOException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    return null;
}

```

- iv. La méthode `ecrireObjectFichier (Object o)` permet d'écrire un objet sérialisable dans un fichier.

```

public static void ecrireObjectFichier(Object o){
    ObjectOutputStream oos;
    try {
        oos = new ObjectOutputStream(new FileOutputStream(name));
        oos.writeObject(o);
        oos.close();
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

5) Ecriture de la classe Main.java

Dans une méthode `main` dont la signature est `public static void main(String arg[])`, ajoutez les instructions suivantes:

- a. écrire un string dans un fichier puis le lire, en utilisant les méthodes `ecrireFichier()` et `afficherFichier()`

```

public static void main(String[] args) {

    try {
        Fichier.ecrireFichier("bonjour");
        Fichier.afficherFichier();
    }
}

```

```

    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

```

- b. écrire un string dans un fichier un objet sérialisable puis le lire, en utilisant les méthodes `ecrireObjectFichier()` et `afficherObjectFichier()`

```

public static void main(String[] args) {

    try {
        Object o = new String ("toto");
        Fichier.ecrireObjectFichier(o);
        String s = (String) Fichier.afficherObjectFichier();
        System.out.println(s);
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

- 6) Exécuter le programme

Exercice 2 : Viusaliseur de fichiers

Créer un programme Java qui permette d’afficher le contenu d’un fichier dans un composant `TextArea`. En bas de la fenêtre se trouve un `Panel` contenant un `FlowLayout` avec 2 boutons, l’un pour ouvrir et lire un fichier et l’autre pour fermer l’application. Par souci de simplicité, on pourra lire le contenu du fichier (supposé être un fichier texte) dans un tampon (un tableau de `char`), qui servira ensuite de source pour l’affichage dans le composant `TextArea`. On pourra à cette occasion se servir de la méthode `length()` de `File`, qui fournit la taille du fichier. On créera une méthode `actionPerformed()` pour les 2 boutons précédents.

Exercice 3 : Explorateur simple

Créer un programme Java permettant de se déplacer dans une arborescence de répertoires. Le contenu du répertoire courant est affiché dans un composant `List`. Lorsque l’on sélectionne un nom d’entrée de cette liste, le programme affiche des informations sur cette entrée (taille, date de dernière modification, . . .) dans un composant `TextField`. Lorsque l’on “double-clique” sur une entrée de répertoire, on entre dedans et on affiche son contenu. Lorsque l’on double clique sur une entrée de fichier, il est affiché dans une fenêtre `FileLister`.

Exercice 4 : Copie de fichier texte

Créer un programme ou qui permet de copier un fichier texte source vers un autre fichier texte de destination. Créer ce programme en utilisant les deux méthodes possibles : caractère par caractère ou ligne par ligne.