

Fichiers

1 Exemples introductifs

Fichier texte

Dans cet exemple (`FichierTexte.java` sur Moodle), on va enregistrer dans un fichier texte la liste des livres d’une bibliothèque (en simplifiant les données). Chaque ligne de texte correspondra à un livre. Le premier mot de la ligne sera le code alphanumérique du livre, puis on trouvera le nombre d’exemplaires du livre et enfin, sur le reste de la ligne, le titre du livre. La procédure `ecrire()` va générer un fichier et la procédure `lire()` va relire un fichier. Dans un vrai programme de gestion d’une bibliothèque, la procédure `lire()` servirait à charger en mémoire la liste des livres au lancement du programme, le programme permettrait de consulter et modifier la liste des livres, et enfin le procédure `ecrire()` permettrait de sauver cette liste quand on le souhaite.

```
import java.io.*;
import java.util.Scanner;

class FichierTexte {
    static String[] titres={
        "Un drame au Mexique", ...
        "L’Invasion de la mer"
    };

    static void ecrire(String filename) {
        try {
            PrintWriter out=new PrintWriter(filename);

            for(int i=0;i<titres.length;i++)
                out.println("V"+i+" "+(i%3)+" "+titres[i]);

            out.close();
        }
        catch(Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }

    static void lire(String filename) {
        try {
            Scanner in=new Scanner(new File(filename));
            String code,titre;
            int nb;

            while(in.hasNext()) {
                code=in.next();
                nb=in.nextInt();
                in.skip("\s+");
                titre=in.nextLine();

                System.out.println(code+" "+nb+" "+titre);
            }

            in.close();
        }
        catch(Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }

    public static void main(String[] args) {
        ecrire("biblio.txt");
        lire("biblio.txt");
    }
}
```

2 Exercices

Fichiers textes

Exercice 1 : On utilise un fichier texte pour mémoriser la température relevée chaque jour dans un lieu donné. Ce fichier contient donc des lignes de la forme “JJ/MM/AAAA TEMP”. On supposera que les données sont fournies dans l’ordre chronologique.

- Écrivez un programme pour générer aléatoirement des températures sur plusieurs mois et les enregistrer dans un fichier.
- Écrivez un programme qui relit le fichier et calcule les moyennes de température pour chaque mois.

Exercice 2 : Programmez la commande `wc`.

Exercice 3 : Un fichier CSV est un fichier texte qui contient des lignes, chaque ligne contient des chaînes de caractères séparées par le caractère virgule. Pour simplifier, on considérera que le séparateur est toujours le caractère virgule et on ne gèrera pas les guillemets dans un premier temps. On choisit de représenter en mémoire les informations contenues dans le fichier CSV sous la forme d’un tableau de lignes, chaque ligne étant un tableau de string. Comme les fichiers CSV n’ont pas tous le même nombre de lignes et de colonnes, et que le nombre de colonnes peut varier d’une ligne à l’autre, on utilisera des `ArrayList`.

1. Ouvrez un tableur, remplissez quelques cellules, sauvez dans un fichier au format CSV et regardez le contenu de ce fichier.
2. Créez une classe `CSV` et déclarez ses attributs.
3. Écrivez une méthode `découperLigne(s)` qui retourne un tableau contenant les chaînes de caractères qui sont séparées par des virgules dans la chaîne `s`
4. Écrivez une méthode `lire(nomDuFichierCSV)` qui lit un fichier CSV
5. Écrivez une méthode `afficher()` qui affiche à l’écran les données de la classe `CSV`
6. Écrivez une méthode `ecrire(nomDuFichierCSV)` qui écrit dans le fichier les informations contenues dans la classe `CSV`.
7. Écrivez un programme de test qui lit un fichier CSV, modifie plusieurs cellules, puis sauvegarde le fichier
8. Modifiez votre programme pour gérer correctement les guillemets. Quand un champ contient le caractère guillemet double (“), le caractère virgule (,) ou un saut de ligne, il doit être encadré par des guillemets doubles dans le fichier CSV. Si le champ contient des guillemets doubles, ces derniers doivent être doublés pour qu’il n’y ait pas de confusion avec la fin du champ. Par exemple ‘John, “Big Jim” et Paul’ est enregistré dans le fichier CSV sous la forme ‘"John, ""Big Jim"" et Paul"’.

Exercice 4 : On veut remplacer dans un fichier des mots-clefs par leur valeur. On vous demande donc d’écrire un programme `remplacer keywords source destination` qui prend 3 paramètres sur la ligne de commande, lit le fichier texte `source`, remplace les mots-clefs par leur valeur, puis écrit le résultat dans le fichier `destination`. Le fichiers `keywords` contient la liste des mots clefs. Dans ce fichier, chaque ligne contient un mot-clef, le caractère tabulation ‘\t’ et la chaîne de caractères qu’il faut substituer à ce mot-clef. On précise que chaque mot-clef peut être une chaîne de caractères quelconques, à partir du moment où elle ne contient pas le caractère tabulation.

Exercice 5 : Écrivez des méthodes pour écrire et lire un fichier JSON (`ecrireJSON()`, `lireJSON()`). Les objets javascript seront représentés par des `TreeMap` et les tableaux par des `ArrayList`.