

Objectifs: Mettre en pratique l'externalisation et la persistance des données dans des fichiers.

1 Externalisation des données pour le jeu du TD n°12 avec le type `File`

Dans le TD n°12, on a modélisé un jeu consistant à chercher des mots d'une catégorie donnée. Il s'agit dans cet exercice d'écrire le code permettant de charger les catégories renseignées dans des fichiers et de permettre d'en choisir une au début du jeu. Pour ce faire, on va appliquer ce qui a été vu durant le cours sur les fichiers. N'hésitez pas à le consulter durant ce TP ainsi que [la documentation sur les fichiers](#).

1. Téléchargez l'archive `tp12-jeuMots` disponible sur moodle et décompressez la dans votre répertoire de travail du tp12 de DÉV. Elle comporte :
 - les fichiers `Mot.java` et `Categorie.java` décrivant les types correspondants tels que modélisés avec le groupe C (si vous avez opté pour une autre modélisation, libre à vous d'adapter)
 - le programme `JeuMots.java` contenant les fonctions permettant de créer un `Mot`, une `Catégorie` ou d'afficher tous les mots d'une catégorie.
 - le répertoire `categories` contenant 3 fichiers textes décrivant chacun le nom et les éléments d'une catégorie différente.
2. Ici, on va recourir à des fichiers textes. Pour pouvoir utiliser les fonctions qui s'y rapportent, il faut commencer par ajouter cette instruction au tout début de votre fichier : `import extensions.File;`
3. Décommentez et implémentez la fonction nommée `choixFichier` permettant au joueur de choisir un nom de fichier parmi ceux disponibles dans le répertoire passé en paramètre. Vérifiez ensuite son bon fonctionnement via le `void algorithm()`.
4. Décommentez et réalisez la fonction nommée `nbLignes` qui pour un nom de fichier donné calcule le nombre de lignes qu'il contient.
5. Décommentez et réalisez la fonction nommée `chargerCategorie` qui crée une `Catégorie` à partir du chemin du fichier correspondant. Plusieurs fonctions pourront vous être utiles pour ce faire.
6. Décommentez maintenant le reste du `void algorithm()` et exécutez votre programme afin de voir s'il affiche bien l'intégralité des mots contenus dans une catégorie choisie.
7. Prolongement : Implémenter entièrement le jeu du TD12.

2 Externalisation et persistance de données structurées avec le type `CSVFile`

Dans cet exercice, vous allez utiliser le type `CSVFile` défini dans la bibliothèque `program.jar` et qui permet de représenter un fichier au format CSV¹. Vous utiliserez le fichier nommé `"USPresident.csv"` qui contient différentes informations concernant des présidents américains.

Pour manipuler les fichiers CSV, c'est-à-dire des fichiers textes contenant des éléments séparés par des virgules et retour à la ligne, vous pourrez utiliser le nouveau type `CSVFile`. Les **fonctions suivantes** permettant de manipuler ce type vous **sont fournies** :

- `CSVFile loadCSV(String filename)` : charge le fichier CSV stocké dont le nom est reçu dans le paramètre `filename`
- `int rowCount(CSVFile file)` : retourne le nombre de lignes du fichier
- `int columnCount(CSVFile file)` : retourne le nombre de colonnes du fichier
- `String getCell(CSVFile file, int line, int column)` : retourne le contenu à la coordonnée (`line`, `column`)
- `void saveCSV(String[][] tab, String filename)` : sauve le tableau `tab` sous la forme d'un fichier CSV.

1. CSV est l'abréviation pour « comma separated values »

Cependant, pour pouvoir accéder à ce nouveau type, `CSVFile`, qui ne fait pas partie des types de base (comme `int` ou `String`), il est nécessaire d'informer le compilateur que l'on souhaite l'utiliser dans notre programme. Pour cela, il est nécessaire d'utiliser l'instruction suivante:

```
import extensions.CSVFile;
```

au tout début de votre programme, avant le mot clé `class`.

Lorsqu'un fichier CSV est chargé, il est stocké dans une variable de type `CSVFile` sous la forme d'un tableau à deux dimensions de chaînes de caractères. Il est possible de connaître le nombre de lignes grâce à l'instruction `rowCount(CSVFile)` et le nombre de colonnes avec `columnCount(CSVFile)`. Pour accéder à une case, il suffit d'appeler `getCell` en précisant les indices de colonne et de ligne (sachant que les indices commencent à 0). Finalement, si vous souhaitez créer un fichier CSV, vous pouvez sauvegarder un tableau à deux dimensions de chaînes de caractères à l'aide de l'instruction `saveCSV`.

2.1 Externalisation des données

1. Ecrivez un algorithme qui charge le fichier `USPresident.csv` (attention à son emplacement) et affiche l'intégralité de son contenu.
2. On souhaite désormais structurer une partie de ces données dans la mémoire afin de réaliser un certain nombre de traitements dessus. Créez un type `President` contenant les champs `nom`, `parti`, `anneeDebutMandat` et `anneeFinMandat`.
3. Réalisez une fonction permettant de créer un nouvel élément de type `President` à partir de l'ensemble des informations qui le caractérisent.
4. Réalisez une fonction `President[] load(String nomFichier, int casesSuppl)` qui
 - (a) crée un tableau de `President` avec autant de cases que le nombre de présidents du fichier en paramètre + le nombre de cases supplémentaires indiquées.
 - (b) le remplit de `President` tels que définis ci-dessus en les créant à partir des colonnes utiles du fichier CSV fourni. Les `casesSuppl` dernières cases resteront vides pour l'instant.

Pour récupérer les années de début et fin de mandat dans un format adéquat, vous pouvez utiliser la fonction `int stringToInt(String s)` dont [la documentation est fournie ici](#).

2.2 Persistance des données

1. Réalisez une fonction `void save(USPresident[] presidents, String nomFichier)` qui sauvegarde dans un fichier dont le nom est fourni l'ensemble des présidents du tableau reçu en paramètre. À noter qu'il faudra passer par une première étape de transformation des données en `String[][]` pour les sauvegarder ensuite dans le CSV à partir des fonctions fournies et présentées au début de ce sujet.
2. Réalisez une fonction principale `algorithm` qui
 - (a) liste l'ensemble des informations du fichier
 - (b) demande combien de présidents il y a eu depuis et de saisir leurs informations (nom, parti, année de début et de fin mandat)
 - (c) crée un fichier `USPresident2022.csv` comportant uniquement les informations que nous avons choisies de retenir pour l'ensemble des présidents des états unis (ceux du fichier ainsi que ceux saisis). Après exécution, vérifiez que votre fichier a bien été créé et qu'il contient bien les informations présentes dans votre tableau.

2.3 Prolongements

1. Écrivez une fonction `int nombreDe(President[] presidents, String parti)` qui calcule pour un tableau donné combien de présidents sont d'un parti donné
2. Écrivez une fonction qui calcule la durée maximale en années pendant laquelle un même parti est resté au pouvoir.

3 SAÉ1.2 Logiciels ludo-pédagogiques

Réfléchissez à comment vous allez intégrer l'externalisation et la persistance des données dans votre Logiciel ludo-pédagogique.