

Feuille d'exercices n° UnknownTdNumberException

Exercice 1.

Dans le TD sur la gestion d'un ensemble de `Note`, nous choisissons d'écrire nos représentations d'objets `Note` sous un format texte, chaque ligne représentant une note. Il est possible de choisir une approche différente, celle qui permet de stocker des objets *sérialisés* dans un fichier. La première méthode (format texte) a l'avantage de créer un fichier lisible par l'homme. La seconde méthode (sérialisation) encode l'objet dans un format illisible mais elle demande moins de lignes de codes et est plus générique : la manière de sérialiser un objet est la même quelque soit l'objet, il suffit qu'il implémente l'interface `Serializable`. L'idée, c'est que c'est Java qui se charge de l'écriture de l'objet et de tous ses champs.

Pour sérialiser un objet, il suffit de :

- faire en sorte qu'il soit sérialisable ainsi que tous ces champs
- l'écrire dans un fichier en utilisant `FileOutputStream` et `ObjectOutputStream`

Pour retrouver cet objet sérialisé, il faut :

- récupérer l'objet sous forme de `Serializable` dans le fichier dans lequel il est stocké en utilisant `FileInputStream` et `ObjectInputStream`
- caster cet objet dans le type qui convient

Interface `Serializable` Faire en sorte que `Note` et `EnsembleNote` implémentent l'interface `Serializable`.

Les méthodes de sérialisation Ecrire une méthode qui écrit un objet `Note` dans un fichier en utilisant :

- `public FileOutputStream(String name) throws FileNotFoundException` qui ouvre un flux pour écrire dans le fichier d'emplacement `name`
- `public ObjectOutputStream(OutputStream out) throws IOException` qui permet d'écrire des objets sur le flux `out`
- `public final void writeObject(Object obj) throws IOException` de la classe `ObjectOutputStream` qui écrit un objet `obj`

Puis récupérer l'objet `Note` que l'on vient d'écrire à partir du fichier en vous servant de :

- `public FileInputStream(String name) throws FileNotFoundException` qui ouvre un flux pour lire dans le fichier d'emplacement `name`
- `public ObjectInputStream(InputStream in) throws IOException` qui permet de lire des objets avec le flux `in`
- `public final Object readObject() throws IOException, ClassNotFoundException` de la classe `ObjectInputStream` qui lit un objet

Effectuer le même processus avec un objet `EnsembleNote`.