

Qualité de Développement – TP3

Compléter le codage réalisé dans le TP2 afin de prendre en charge la partie enregistrement / chargement et verrouillage / déverrouillage des données.

Faire des tests en exécutant l'application dans plusieurs consoles en parallèle.

Enregistrement d'objets Java par sérialisation dans un fichier

Pour pouvoir enregistrer un objet Java par sérialisation dans un fichier, il faut que l'objet soit une instance d'une classe **sérialisable**, c'est-à-dire une classe qui implémente l'interface **java.io.Serializable** et dont les attributs sont eux-mêmes sérialisables (types primitifs, String, attributs typés par des classes sérialisables, tableau de données sérialisables).

Ensuite, il faut créer un flux de type **java.io.ObjectOutputStream** relié à un flux de type **java.io.FileOutputStream** et appeler l'opération **writeObject** en lui passant l'objet en paramètre.

Exemple : Soit objet une variable contenant l'objet à enregistrer. Le code suivant permet d'enregistrer objet par sérialisation dans un fichier nommé "donnees" (relativement au répertoire courant) :

```
ObjectOutputStream donnees = new ObjectOutputStream(new FileOutputStream("donnees"));
donnees.writeObject(objet);
donnees.close();
```

A noter : la sérialisation d'un objet a pour effet de sérialiser toute la « grappe » d'objets sérialisables liés à cet objet.

Chargement d'objets Java sérialisés dans un fichier

Pour le chargement d'un objet sérialisé, il faut créer un flux de type **java.io.ObjectInputStream** relié à un flux de type **java.io.FileInputStream** et appeler l'opération **readObject** sans paramètre qui renvoie un objet de type **java.lang.Object** (il faut donc faire un casting).

Exemple : Supposons que l'objet à charger est de type **TypeObject** et qu'il a été sérialisé dans un fichier nommé "donnees" (relativement au répertoire courant). Le code suivant permet de réaliser le chargement dans la variable nommée objet :

```
ObjectInputStream donnees = new ObjectInputStream(new FileInputStream("donnees"));
TypeObjet objet = (TypeObjet)donnees.readObject();
donnees.close();
```

Verrouillage / déverrouillage des données

Une solution simple consiste à créer un fichier (dans le répertoire courant par exemple, le contenu du fichier peut rester vide) pour indiquer que le verrou est posé et de supprimer ce fichier pour indiquer que le verrou est levé.