

## Travaux pratiques d'informatique N° 17

Le but de cette séance est de vous permettre de consolider vos connaissances concernant les entrées/sorties Java et d'utiliser les classes **FileOutputStream**, **DataOutputStream**, **FileInputStream**, **DataInputStream**, **RandomAccessFile**, **FileWriter**, **PrintWriter**, **FileReader**, **BufferedReader** et **StringTokenizer**.

### 1. Ecrire une application autonome interactive qui :

- demande à l'utilisateur :
  - le nom d'un étudiant ;
  - ses notes aux Contrôles numéros 1, 2, 3 et 4 ;
  - s'il veut continuer à introduire d'autres données ou pas ;
- reprend les mêmes questions précisées ci-dessus, si l'utilisateur veut continuer ;
- affiche un message de fin de session, si l'utilisateur veut s'arrêter.

De plus, l'application doit stocker toutes les données introduites par l'utilisateur dans un fichier binaire qui pourra être lu par la suite (voir l'exercice numéro 2.).

*Indication : Vous pouvez utiliser les méthodes **writeUTF( )** et **writeDouble( )** de la classe **DataOutputStream**.*

### 2. Ecrire une application autonome qui :

- lit le fichier binaire obtenu par l'exercice numéro 1. ;
- pour chaque étudiant, calcule la moyenne de ses notes aux quatre contrôles ;
- affiche à l'écran, sur des lignes distinctes, le nom et la moyenne de chaque étudiant.

*Indication : Vous pouvez utiliser les méthodes **readUTF( )** et **readDouble( )** de la classe **DataInputStream**.*

**3.** Ecrire une application autonome interactive similaire à celle de l'exercice numéro **1.**, mais qui stocke les données introduites par l'utilisateur dans un fichier texte qui pourra être lu par la suite (voir l'exercice numéro **4.**).

*Indication : Vous pouvez utiliser les méthodes **print()** et **println()** de la classe **PrintWriter**.*

**4.** Ecrire une application autonome interactive similaire à celle de l'exercice numéro **2.**, qui lit le fichier texte obtenu par l'exercice numéro **3.** et qui écrit dans un fichier texte, sur des lignes distinctes, le nom et la moyenne de chaque étudiant.

*Indication : Vous pouvez utiliser la méthode **readLine()** de la classe **BufferedReader** pour la lecture, la classe **StringTokenizer** pour le découpage en sous-chaînes de caractères et les méthodes **print()** et **println()** de la classe **PrintWriter** pour l'écriture.*

**5.** (Exercice intéressant mais facultatif) Ecrire une application autonome qui :

- réalise d'abord les mêmes tâches que celles précisées dans l'exercice numéro **1.**, mais qui écrit les données dans un fichier binaire à accès direct ouvert en lecture/écriture ;
- lit ensuite le fichier susmentionné afin d'afficher à l'écran, sur des lignes distinctes, le nom et la note au quatrième contrôle de chaque étudiant.

*Indication : Vous pouvez utiliser les méthodes **writeUTF()**, **writeDouble()**, **readUTF()**, **readDouble()**, **readShort()** et **seek()** de la classe **RandomAccessFile**.*