Telecom Nancy 1ère année
Structures de Données - 2013/2014

Flots

Exercice 1 Générateur de fichier

L'objectif de cet exercice est d'écrire un générateur de fichiers d'entiers pris au hasard. L'application `flots.Generateur` utilise les arguments de la ligne de commande : le nom du fichier destiné à contenir les entiers générés, les nombres minimum et maximum de données à générer et une indication sur l'ordre dans lequel les données sont générées. Le nombre de données générées est pris au hasard dans les bornes spécifiées ; les données sont prises au hasard.

Exemples d'appel :

```
java flots.Generateur donnees1.txt 5 1000 Croissant
java flots.Generateur donnees2.txt 100 200 Decroissant
java flots.Generateur donnees3.txt 10 20
```

Dans les exemples ci-dessus, le fichier `donnees1.txt` est rempli avec des entiers quelconques rangés dans l'ordre croissant (il y en a entre 5 et 1000), le fichier `donnees2.txt` avec des entiers quelconques rangés dans l'ordre décroissant et le fichier `donnees3.txt` avec des entiers quelconques sans rangement particulier.

Un peu d'aide :

- *les entiers sont écrits au fur et à mesure de leur génération, sans stockage temporaire ;*
- *écrire trois fonctions de génération dans la classe `flots.Generateur`, chacune d'elles gérant un des trois cas (croissant, décroissant, quelconque) ;*
- *la classe `java.util.Random` fournit un générateur de nombres aléatoires plus pratique que `Math.random`.*

Exercice 2 Création et sauvegarde de polygônes.

L'objectif de cet exercice est d'écrire une application `flots.PolyFlots` qui crée des instances de la classe `Polygone` en lisant les coordonnées successives des points dans des fichiers distincts (un fichier par polygône, dans chaque fichier alternance des abscisses et des ordonnées). Tous les polygônes ainsi créés sont sauvegardés dans un fichier. La commande utilise deux arguments : le chemin d'accès au répertoire qui contient tous les fichiers contenant des coordonnées de points et le chemin d'accès au fichier destiné à sauvegarder les polygônes.

Exemple d'appel :

```
java flots.PolyFlots donnees poly
```

Dans l'exemple ci-dessus, tous les fichiers de suffixe `txt` contenus dans le répertoire `donnees` sont utilisés pour créer des polygônes (un par fichier) ; ces polygônes sont sérialisés dans le fichier `poly`. Tout fichier du répertoire qui contient moins de 6 coordonnées est ignoré. Si un fichier contient un nombre impair de coordonnées, la dernière est ignorée.

Consignes :

- *utiliser les fonctions de `java.io.File` pour retrouver tous les fichiers d'un répertoire ;*
- *écrire une classe `Polygone` rudimentaire, qui stocke les points dans une `ArrayList`.*
- *utiliser les fichiers générés par `flots.Generer` pour réaliser les tests.*
- *tester l'application en écrivant une classe qui relit le fichier produit et affiche le polygône lu ;*

Exercice 3 Calcul de moyennes de jurys

On souhaite développer une application permettant de calculer les moyennes des différents jurys d'une formation. On dispose pour cela de fichiers de notes ayant un format identique. On suppose ces fichiers correctement construits.
La première ligne d'un fichier de données contient le nom de l'UE ; la seconde ligne contient le nombre de notes, suivi des coefficients des différentes notes ; sur chaque ligne suivante, on trouve le nom et le prénom de l'étudiant suivi de ses différentes notes. Voici un exemple de fichier de données :

Informatique
4 2 3 3 1
Dupond Isidore 12 13 19 12
Legrand Raymond 8 18 14 6
Mizet Félix 7 13 15 9

Ecrire (et tester) la classe `jury.UE` avec au moins les différentes fonctions suivantes :

- Le constructeur admet en paramètre le nom d'un fichier de données suffixé `txt` ; le constructeur ouvre le fichier si c'est possible et déclenche une exception sinon.
- La méthode `public int nbreNotes()` permet de consulter le nombre de notes.
- La méthode `public int nbreEtudiants()` permet de consulter le nombre d'étudiants.
- La méthode `public void calculerResultats()` crée un fichier de même nom que le fichier de données mais suffixé `res` ; ce fichier contient les résultats de l'UE : la première ligne indique le nom de l'UE et sur les lignes suivantes, on retrouve le nom de chaque étudiant et sa moyenne à l'UE, arrondie au dixième.

Voici un exemple de fichier résultats :

Informatique
Dupond Isidore 14.6
Legrand Raymond 13.1
Mizet Félix 11.8