M2105	Séance 1 / Semaine 13	2016-2017

Remarques préliminaires :
Cet énoncé est par nature incomplet, il invite à prendre conscience de l'importance de la documentation de l'API Java pour résoudre les problèmes posés. Il est prévu de répondre à l'ensemble des questions en 3h environ, avec une évaluation en classe par l'enseignant lors du premier créneau de 1h30 (fin de partie A) et lors du second créneau de 1h30 (fin de partie B). Bien qu'il soit demandé d'écrire du code Java de façon incrémentale, en modifiant le code au fur et à mesure des questions posées, il reste fortement conseillé d'archiver chacune des réponses sous forme d'un fichier de sauvegarde.

Le Professeur Tournesol souhaite conserver une trace des évaluations de ses élèves. Un élève est représenté par un nom et un prénom. Une évaluation correspond à un coefficient (entier entre 0 et 10) et une note (entier entre 0 et 20). Les différentes évaluations d'un même élève donnent lieu au calcul d'une moyenne.

A. Lire et écrire des fichiers texte

1. Ecrire la classe Evaluation et la compiler. Ecrire la classe Eleve et la compiler.

2. Ecrire un programme de test et l'exécuter.

3. Ajouter des méthodes d'écriture / lecture dans un fichier texte aux classes Eleve et Evaluation. Ces méthodes pourront prendre en paramètre le nom du fichier de sauvegarde à lire ou écrire. Procéder à l'écriture et la lecture des données depuis votre programme de test.

4. Modifier l'application pour ne pas lire / écrire la moyenne, mais plutôt pour la recalculer au moment de la lecture du fichier.

5. Faire évaluer le programme (survol du code source + trace d'exécution) par l'enseignant.

B. Lire et écrire des fichiers binaires

1. Modifier l'application pour lire et écrire les données dans des fichiers binaires, en utilisant les classes DataInputStream et DataOutputStream, mais aussi d'autres classes si nécessaire.

2. Apporter les modifications nécessaires au code pour sauvegarder la moyenne dans le fichier binaire, et vérifier le bon comportement de l'application.

3. Comparer les tailles des fichiers de sauvegarde en mode texte et en mode binaire. Quelles conclusions peut-on tirer ?

4. Supprimer (provisoirement) l'appel à la méthode close() dans les programmes de lecture et écriture de fichiers texte et binaire. Que peut-on observer quant au fonctionnement des programme, et au contenu des fichiers ?

5. Faire évaluer le programme (survol du code source + trace d'exécution) par l'enseignant.

C. Lire et écrire des objets

1. En s'appuyant sur la javadoc de l'interface java.io.Serializable, apporter les modifications nécessaires aux classes dont les instances pourraient être lues et écrites par l'application.

2. Modifier l'application pour lire et écrire les objets, en utilisant les classes ObjectInputStream et ObjectOutputStream.

3. En s'appuyant sur la javadoc des classes précédentes, faire évoluer l'application en redéfinissant le mécanisme de lecture/écriture des objets pour ne pas sauvegarder la moyenne et la recalculer au moment de la lecture du fichier.

4. Faire évaluer le programme (survol du code source + trace d'exécution) par l'enseignant si possible. Sinon, finir le travail à la maison.

5. Déposer individuellement votre travail (codes sources des questions A, B, et C) prévu à cet effet sur Moodle, sous la forme d'une archive M2105_TP1_nom1_nom2.zip si le travail a été réalisé en binôme (un dépôt par étudiant) ou M2105_TP1_nom.zip si le travail a été réalisé tout seul. La date limite de rendu est fixée à la semaine suivante, lundi 8h.