EX1 : Création séquentielle d'un fichier binaire

Écrire un programme permettant de créer séquentiellement un fichier binaire comportant pour différentes personnes les informations suivantes : nom, prénom et année de naissance.

Le dialogue de saisie de l'information s'effectuera en fenêtre console comme dans cet exemple :

Nom du fichier a creer :
e:\repert
nom 1 : Carre
Prenom : Thibault
annee naissance : 1997
.....
nom 5 : Mitenne
Prenom : Thomas
annee naissance : 2001
nom 6 :
****** fin creation fichier ******

On proposera deux solutions :

1. Les informations relatives au nom et au prénom seront conservées dans le fichier sous la forme d'une suite de 20 caractères (comportant d'éventuels espaces à la fin).
2. Ces mêmes informations seront conservées sous la forme d'une chaîne codée dans le format UTF; aucune contrainte ne portera sur leur longueur.

UTF : Ce format (Unicode Text Format) permet de coder une chaîne sous forme d'une suite d'octets en nombre variable (chaque caractère étant codé sur un à trois octets). La méthode writeUTF de la classe DataOutputStream réalise cette transformation d'une chaîne en une suite de caractères UTF.

Outils : Vous utiliserez la démarche la plus classique qui consiste à exploiter les méthodes de la classe flux *DataOutputStream*. Pour ce faire, vous associerez un objet de ce type (nommé *sortie*) à un fichier dont le nom est fourni par l'utilisateur dans la chaîne *nomFichier* :

```
DataOutputStream sortie = new DataOutputStream (new FileOutputStream (nomFichier)) ;
```

Les variables *chNom* et *chPrenom* servent à lire les informations nom et prénom sous forme de chaînes de caractères. Nous en transférons ensuite chacun des caractères (à concurrence de 20) dans des tableaux de 20 caractères *nom* et *prenom*, préalablement remplis avec des espaces. L'écriture dans le fichier est réalisée à l'aide des méthodes *writeChar* (écriture d'un caractère) et *writeInt* (écriture d'un entier) de la classe *DataOutputStream*.

Ex2 : Liste (lecture) séquentielle d'un fichier binaire

Écrire un programme permettant de lister en fenêtre console le contenu d'un fichier binaire tel que celui créé par l'exercice. On proposera deux solutions correspondant aux deux situations :
1. Les informations relatives au nom et au prénom ont été enregistrées dans le fichier sous la forme d'une suite de 20 caractères (comportant d'éventuels espaces à la fin).
2. Ces mêmes informations ont été enregistrées sous la forme d'une chaîne codée dans le format UTF ; aucune contrainte ne portera sur leur longueur.

Outils : Vous exploitez les méthodes de la classe flux *DataInputStream*. Pour ce faire, vous associez un objet de ce type (nommé *entree*) à un fichier dont le nom est fourni par l'utilisateur dans la chaîne *nomFichier* :

```
DataInputStream entree = new DataInputStream (new FileInputStream (nomFichier)) ;
```

Les informations relatives au nom et au prénom sont lues dans des tableaux de 20 caractères *nom* et *prénom* à l'aide de la méthode *readChar* de la classe *DataInputStream*.

La gestion de la fin de fichier est réalisée en interceptant l'exception *EOFException* : la boucle de lecture des informations est contrôlée par un indicateur booléen *eof* initialisé à *false* et mis à *true* par le gestionnaire d'exception.

Problème : Sérialisation :

1. Ecrire une classe sérialisable Etudiant caractérisée par les attributs nom, prénom, age et les méthodes affichage, getNom, getPrenom et getAge.
2. Ecrire une classe sérialisable Salle caractérisée par les attributs liste des étudiants et par les méthodes ajouterEtudiant et affichageTotales des étudiants dans la liste.

3 - Pour écrire la sérialisation d'une Salle dans un fichier on utilise :

*- Un **ObjectOutputStream** qui sérialise l'objet et produit un flux binaire.*
*- Un **FileOutputStream** qui redirige ce flux vers un fichier*
*- Un **BufferedOutputStream** pour améliorer les performances*

Ecrire une classe SerialSalle qui a deux méthodes :

*a- **saveSalleToSerialFile** pour sauvegarder l'objet Salle dans un fichier donné. Cette dernière a deux arguments le nom de fichier et un objet de type Salle.*
*b- **getRealisateurToSerialFile** pour récupérer l'objet de type Salle à partir du fichier de sérialisation.*

***4** – Ecrire une classe d'utilisation qui comporte la méthode main pour créer les objets Etudiants, Salle et les sauvegarder dans un fichier et finalement les récupérer en les affichant sur la console.*