

Labs

Exercice 1

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 :

```
1 Nom du fichier a creer : e:\repert
2 nom 1 : Carre
3 Prenom : Thibault
4 annee naissance : 1997 .....
5 nom 5 : Mitenne
6 Prenom : Thomas
7 annee naissance : 2001
8 nom 6 :
9 **** 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 UTFa ; aucune contrainte ne portera sur leur longueur.

Exercice 2

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.

Exercice 3

Écrire un programme qui liste en fenêtre console le contenu d'un fichier texte en en numé- rotant les lignes. On prévoira 4 caractères pour l'affichage du numéro de ligne. Les lignes de plus de 60 caractères seront affichées sur plusieurs lignes d'écran comme dans cet exemple

```
1 Donnez le nom du fichier texte a lister : e:\book\essai.txt
2 1 Ceci est la premiere ligne d'un exemple de fichier texte
3 2 Il contient des lignes de chiffres de longueurs variables
4 dont une de 59 caracteres, une de 60 caracteres et une de 61
5 caracteres
```

```
6 | 3 12345678901234567890
7 | 4 12345678901234567890123456789012345678901234567890
8 | 5 1234567890123456789012345678901234567890123456789
9 | 6 123456789012345678901234567890123456789012345678901
10 | 7 1234567890123456789012345678901234567890
11 | 8 la ligne suivante est vide
12 | 9
13 | 10 les deux lignes suivantes sont également vides 11
14 | 12
15 | 13 Ceci est la derniere ligne du fichier
16 | *** fin liste fichier ***
```

Exercise 4

Écrivez un programme Java pour trier un ArrayList donné.

Exemple :
PHP, Java, C++, Ada

Sortie prévue :

Liste avant le tri : [PHP, Java, C++, Ada]

Liste après le tri : [Ada, C++, Java, PHP]

Exercise 5

Écrivez un programme Java pour inverser les éléments d'un ArrayList.

Exemple :
PHP, Java, C++, Ada

Sortie prévue :

Liste avant l'inversion : [PHP, Java, C++, Python]

Liste après l'inversion : [Python, C++, Java, PHP]

Exercise 6

Ecrivez une méthode displayEvenIndex afin qu'elle affiche les entiers aux index pairs de la liste passée en argument.

Exemple :

1,2,3,4,5,6,7,8,9

Sortie prévue :

1, 3, 5, 7, 9

Exercise 7

Ecrivez une méthode `displayEvenElements` afin qu'elle affiche tous les nombres entiers pairs qui figurent dans l'ArrayList transmise en argument.

Exemple : 1,2,3, 4, 5, 6, 7, 8, 9

Sortie prévue : 2, 4, 6, 8

Exercise 8

Écrivez un programme Java pour comparer deux ArrayList. Affiche TRUE si il existe dans la première liste sinon FALSE

Exemple :

ArrayList 1 : [PHP, Java, C++, Python]

ArrayList 2 : [PHP, Java, C, Python]

Sortie prévue :
, TRUE, FALSE, TRUE

Bon courage !!!