

Java DUT 1 Feuille TP10
Université Paris-Est Marne-la-Vallée

Exercice 1.—

On veut écrire un programme permettant de manipuler les joueurs d'un jeu de rôle en ligne où le nombre de joueurs est important.

Pour chaque joueur, on dispose de données diverses dont au moins:

- le pseudo de type **String** qui est un identifiant unique dans le jeu;
- le nom de type **String**;
- le niveau du personnage qui sera un entier;
- l'ensemble des pseudos de ses amis dans le jeu.

1. Écrire une classe **Player** contenant les données du joueur et un constructeur qui prend en paramètre **tous** les champs et les initialise. Ce constructeur ne doit pas être visible de l'extérieur de la classe. Pourquoi?
2. Ajouter des *getters* pour le pseudo et le niveau ainsi qu'une méthode d'affichage (voir exemple ci-dessous).
3. Les données de l'ensemble des joueurs du jeu seront stockées dans un fichier. Chaque ligne représente un joueur sous la forme suivante: entre crochets, le pseudo, le nom, le niveau, et les amis. Chaque champ entre les crochets est séparé par une virgule. Par exemple, une ligne de du fichier peut être:

```
<Yampai804, lennox, 93, Eudora969, Pillow644, FairHill851>
```

et l'affichage du **Player** correspondant sera:

```
*** pseudo = Yampai804, name = lennox, level = 93, friends =  
[Eudora969, Pillow644, FairHill851] ***
```

Écrire une méthode **makePlayerFromLine** qui prend en paramètre une ligne de données (c'est à dire une chaîne de caractères) formatée comme indiqué ci-dessus et renvoie un nouveau **Player** fabriqué à partir de ces données. Attention, cette méthode n'est pas un constructeur mais fait appel au constructeur de la classe (on appelle cela une méthode *factory*).

Exercice 2.— On va créer une classe **Game** pour stocker l'ensemble des joueurs. La classe **Game** doit permettre de trouver rapidement les données d'un joueur ayant un certain pseudo. Pour ceci, on demande que la classe **Game** contienne une **Map players** qui associe à chaque pseudo le joueur ayant ce pseudo.

1. Écrire la classe **Game** avec un constructeur qui prend en paramètre la **Map** et n'est pas visible à l'extérieur de la classe. Ajouter un *getter* qui permet d'obtenir un joueur à partir de son pseudo et une méthode d'affichage (un joueur par ligne).

2. Écrire une méthode `makeGameFromFile` qui prend en paramètre un fichier (son chemin) contenant les données d'un ensemble de joueurs selon le format décrit ci-dessus et fabrique un `Game` à partir de ces données. Attention, cette méthode est aussi une *factory* qui construit un `Game`.

Par exemple, pour un `Game` crée à partir du fichier:

```
<Mesa682, marylou, 46, Alex419, MaunaLoa295>
<Malta493, corbin, 43, MaunaLoa295>
<Alex419, xenia, 76, Mesa682, MaunaLoa295>
<MaunaLoa295, cyndi, 28, Mesa682>
```

l'affichage sera:

```
*** pseudo = Mesa682, name = marylou, level = 46, friends = [Alex419, MaunaLoa295] ***
*** pseudo = Malta493, name = corbin, level = 43, friends = [MaunaLoa295] ***
*** pseudo = Alex419, name = xenia, level = 76, friends = [Mesa682, MaunaLoa295] ***
*** pseudo = MaunaLoa295, name = cyndi, level = 28, friends = [Mesa682] ***
```

3. Écrire une classe `GameTest` pour tester la création et l'affichage d'un jeu à partir des fichiers de données fournis.

Exercice 3.—

1. Écrire une méthode `highestLevel()` qui permet de calculer le plus haut niveau présent parmi les joueurs d'un jeu. Pour cela, vous utiliserez sur l'ensemble des joueurs la méthode de `Collections` qui permet de calculer le maximum. Attention, il ne s'agit pas de calculer la liste des niveaux puis de calculer son maximum.
2. Écrire une méthode `highestLevelPlayers` construisant l'ensemble des pseudos des joueurs qui ont les personnages de plus haut niveau. Pour le fichier `players_data_small.txt`, on obtient par exemple:

```
[Elkader520, Brookings203, Oacoma699, NorthManitou715]
```

3. Écrire une méthode `commonFriends` renvoyant l'ensemble des pseudos des amis communs à deux joueurs identifiés par leurs pseudos. Pour cela, vous utilisez la méthode permettant de faire l'intersection de deux ensembles vue en cours. Avec le fichier `players_data_medium.txt`, pour "Warwick460" et "Interior294", on obtient par exemple:

```
[SaintOlaf212]
```

4. ★ Écrire une méthode `network` qui prend en paramètre le pseudo d'un joueur et renvoie l'ensemble des pseudos de ses amis directs ou indirects (ses amis, les amis de ses amis, les amis des amis de ses amis, ...). Avec le fichier `players_data_very_small.txt`, pour "Amo105", on obtient par exemple:

```
[Camak669, McDermott373, Barneston758, Midtown451, Amo105]
```