Formation Java Pôle Emploi

Nicolas Rousset

Sujet n°2 - lecture de fichiers avec gestion des erreurs

Dans ce sujet on reprendra le projet de gestion de reconversion, en se concentrant sur 3 classes :

- la classe Personne
- la classe Formateur
- la classe PersonneEnReconversion

Auxquelles on rajoutera une classe Competence, avec un id et un String.

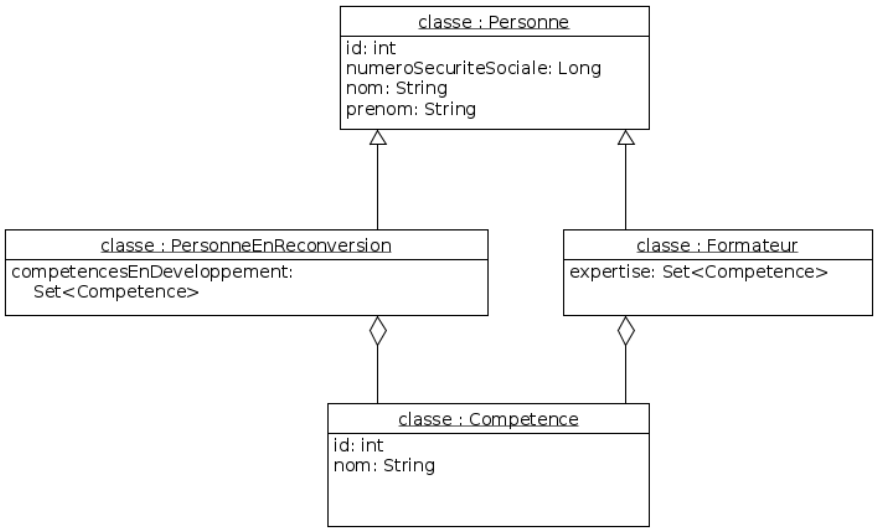

Figure 1: Diagramme classe

Les fichiers de données d'entrées auront la forme :

Fichier **formateurs.csv**
et **personneEnReconversions.csv**

```
numeroSecuriteSociale;competences
223357989824420;Java,C++
```

(Les colonnes sont séparées par des **;**, les compétences par des **,**).

On dispose d'un fichier de compétences réalisé **à posteriori** :

```
competences
Java
C++
Php
HTML
CSS
```

Et enfin du fichier **personnes.csv**

```
numeroSecuriteSociale;nom;prenom
121805264266599;Shelley;Renaud
202332172011871;Watson;Michaël
```

Hélas, le référentiel de compétences a été créé après coup, et donc certaines compétences ont été entrées à la main et diffèrent légèrement ("Java" ou "java", "back-end" ou "Backend").

De plus les formateurs et personnes en reconversion ne sont pas toute référencées dans le fichier **personnes.csv**

On veut donc pouvoir lire ces 3 fichiers en étant tolérant aux erreurs, à savoir :

- une compétence qui n'est pas dans le référentiel est simplement ignorée
- une personne en reconversion ou un formateur qui n'existe pas dans le fichier personnes.csv est également ignoré.

Mais toutes ces anomalies sont loggées dans un fichier spécifique (du nom que vous voulez).

On commencera donc par lire le fichier personne et soit :
- ne créer l'objet Personne que si aucune correspondance n'a été trouvée dans les fichiers formateurs et personnesEnReconversion.
- ou tous les créer et les remplacer ensuite si une correspondance est trouvée

Il n'est pas possible d'"upgrader" une instance en une instance de classe héritée (on peut toutefois créer une copie)

1 Déroulement de l'exercice

1.1 Exercice 1

Implémentez les méthodes pour lire les fichiers et les stocker les objets créés dans un objet central.

1.2 Exercice 2

Créez dans les classes PersonneEnReconversion et Formateur des méthodes matches qui prennent chacun en argument l'autre objet (celle de la classe Formateur prend en argument une PersonneEnReconversion et réciproquement) et renvoit true si il y a un matche (toutes les compétences de la PersonneEnReconversion recouvre celles du Formateur).

1.3 Exercice 3

Créez une méthode statique dans une classe dédiée qui prend en argument une liste de Formateur et une liste de PersonneEnReconversion et effectue le meilleur matching possible.