


DEVE514 - Développement Java

 **Consigne importante** : Vous serez notés bien évidemment sur un code fonctionnel et qui réalise le comportement attendu, mais aussi et surtout sur les commentaires que vous écrirez dans votre code. Vous devrez notamment justifier le choix des modifieurs (*public*, *private*, *static*, etc.), de la gestion des exceptions, de l'héritage, de l'usage du polymorphisme, etc. Le code étant le seul livrable, il doit pouvoir exprimer vos choix et votre démarche. N'hésitez donc pas à abuser des commentaires (jusqu'à ce qu'ils prennent plus de place que le code !)

Objectif du programme à réaliser

Le but du programme est de pouvoir construire un rapport des ventes et commissions de l'ensemble des vendeurs d'une entreprise. Pour cela, vous disposez d'un fichier contenant les dernières ventes de chaque vendeur, sous la forme suivante :
chaque ligne du fichier représente un unique vendeur, dont les informations sont formatées de la manière suivante :

```
<Nom du vendeur>:<age>:<date de la dernière vente au format  
jj/mm/aaaa>:<montant vente 1>;<montant vente 2>;<montant vente 3>; ...
```

Le montant des ventes est censé être un entier, leur nombre n'est pas forcément le même pour chacun des vendeurs.

Exemple :

```
James:42:11/02/2020:2399;349;450;356  
Charlotte:23:14/07/2020:1300;290;450  
Olfa:51:09/12/2020:532;790  
Youcef:25:12/05/2020:120;349;540;78;450
```

Les vendeurs touchent une commission selon le montant de leur vente, mais aussi de leur âge. Un vendeur qui a plus de 40 ans touchera 15% sur ses ventes, les autres seulement 10%.

Votre programme devra réaliser les tâches suivantes :

- Lire le fichier "vendeurs.txt" (fichier fourni), dont le chemin d'accès devra être renseigné en paramètre de votre programme (en ligne de commande)
- Alimenter à partir de son contenu une liste contenant des d'objets "vendeur" et "vendeur sénior"
- Effectuer 2 tris différents de cette liste selon deux critères : le montant total des ventes, et la date de dernière vente (montant le plus élevé en premier, date la plus proche en premier)
- Chacun de ces tris affichera la liste des vendeurs sous le format suivant :

<nom du vendeur> a vendu pour <montant total> € en date du <date au format complet>, sa commission est de <montant commission> €
Exemple :

Charlotte a vendu pour 2040 € en date du 14 juillet 2020, sa commission est de 204 €

Votre programme comportera à minima 5 classes :

- Une classe modélisant un “vendeur” avec les attributs et méthodes que vous jugerez pertinents
- Une classe modélisant un “vendeur sénior”, qui devra hériter de la classe précédente. Vous tirerez parti de l’héritage pour ne coder que le comportement et/ou les attributs spécifiques au vendeur sénior dans cette classe.
- Une classe dite “utilitaire”, qui permettra de lire le contenu d’un fichier et manipuler les données sur les vendeurs (par exemple les trier selon différents critères)
- Une classe représentant une Exception lors de la lecture d’une ligne de fichier (format ne correspondant pas à celui attendu)
- Une classe “point d’entrée” de votre programme, qui sera celle exécutée par l’interpréteur Java, et que vous chercherez à rendre la plus concise possible

Libre à vous de créer d’autres classes si vous le jugez nécessaire, notamment dans l’optique de rendre votre programme davantage “orienté objet” que “procédural”.

Votre programme devra réagir de manière correcte pour l’utilisateur en l’informant d’une erreur d’utilisation ou de syntaxe dans les cas suivants :

- Si le chemin du fichier passé en paramètre ne peut être résolu, ou tout autre problème d’accès au fichier, un message clair d’erreur est affiché à l’utilisateur et le programme s’arrête
- Si une ligne contient une donnée mal formatée (date de la dernière vente, ou montant d’une vente non numérique, etc.) la ligne correspondante est ignorée, l’utilisateur en est averti par un message, mais le traitement se poursuit avec les autres données présentes dans le fichier. Le fichier fourni comporte intentionnellement des erreurs. Si vous ne souhaitez pas commencer par traiter les erreurs, vous pouvez supprimer temporairement ces lignes.

Quelques pistes...

Le parsing d’une ligne peut se faire avec un Scanner, ou plus simplement encore avec la méthode `split` de la classe `String`.

La manipulation des dates au format textuel (formatage ou parsing) peut se faire avec la classe `java.time.format.DateTimeFormatter`

Le tri d'une liste peut se faire rapidement avec une méthode de la classe utilitaire `java.util.Collections` qui prend en paramètre supplémentaire un `Comparator` décrivant la manière d'effectuer la comparaison entre 2 objets.

La gestion des erreurs en Java se fait à base d'`Exception`, en héritant de cette classe ou en utilisant une existante..

Pour rappel, la documentation de l'API Java est disponible sur le site d'Oracle
<https://docs.oracle.com/en/java/javase/21/docs/api/java.base/module-summary.html>
Ou encore sur celui de devdocs
<https://devdocs.io/openjdk~21/java.base/module-summary>