

Exercice 2 : Système de Gestion d'Inventaire en Scala

Objectif : Créer un système de gestion d'inventaire simple pour gérer des articles, leur quantité et leur valeur. Le but est de permettre à l'utilisateur d'ajouter des articles, d'afficher l'inventaire actuel, et de calculer la valeur totale de l'inventaire.

Contexte

Vous êtes chargé de développer un système de gestion d'inventaire en Scala qui peut :

1. Ajouter de nouveaux articles avec un nom, un prix et une quantité.
2. Afficher tous les articles actuellement en stock.
3. Calculer et afficher la valeur totale de tous les articles de l'inventaire.

Exigences

Votre programme doit être structuré en plusieurs fichiers pour encourager une séparation claire des responsabilités. Voici les détails pour chaque fichier :

1. **Article.scala** : Définissez une classe Article pour représenter chaque article dans l'inventaire. Cette classe doit inclure les attributs suivants :
 - nom : Le nom de l'article (type String).
 - prix : Le prix unitaire de l'article (type Double).
 - quantite : La quantité de cet article dans l'inventaire (type Int).

La classe doit aussi inclure :

- Une méthode valeurTotale pour calculer la valeur totale d'un article en fonction de sa quantité.
 - Une méthode toString pour afficher les informations d'un article dans un format lisible.
2. **Inventory.scala** : Définissez une classe Inventory pour gérer une collection d'articles. Cette classe doit inclure :
 - Une collection articles (utilisez ListBuffer).
 - Une méthode ajouterArticle pour ajouter un nouvel article à la collection.
 - Une méthode afficherInventaire pour afficher tous les articles de l'inventaire.

- Une méthode `valeurTotalInventaire` pour calculer et retourner la valeur totale de tous les articles.
- 3. **Main.scala** : Créez un programme principal qui permet à l'utilisateur d'interagir avec le système d'inventaire via un menu. Les options du menu doivent inclure :
 - **Ajouter un article** : Demander à l'utilisateur d'entrer le nom, le prix, et la quantité, puis ajouter cet article dans l'inventaire.
 - **Afficher l'inventaire** : Afficher tous les articles actuellement en stock, avec leurs informations et leur valeur totale.
 - **Afficher la valeur totale de l'inventaire** : Afficher la somme totale de la valeur de tous les articles dans l'inventaire.
 - **Quitter** : Permettre à l'utilisateur de quitter le programme.

Exemple d'Exécution

Votre programme devrait fonctionner de la manière suivante :

```
Menu :
1. Ajouter un article
2. Afficher l'inventaire
3. Afficher la valeur totale de l'inventaire
4. Quitter
Choisissez une option : 1

Entrez le nom de l'article : Chaise
Entrez le prix de l'article : 25.5
Entrez la quantité de l'article : 4
Chaise a été ajouté à l'inventaire.

Menu :
1. Ajouter un article
2. Afficher l'inventaire
3. Afficher la valeur totale de l'inventaire
4. Quitter
Choisissez une option : 2

Inventaire actuel :
Article(nom=Chaise, prix=25.5, quantite=4, valeurTotale=102.0)

Menu :
1. Ajouter un article
2. Afficher l'inventaire
3. Afficher la valeur totale de l'inventaire
4. Quitter
Choisissez une option : 3

valeur totale de l'inventaire : $ 102.0
```

Détails Techniques

- **Structure des Fichiers** : Assurez-vous de créer les trois fichiers Scala suivants dans le même dossier :
 - Article.scala
 - Inventory.scala
 - Main.scala

- **Compilation et Exécution** :

- Compilez chaque fichier avec la commande suivante :

```
scalac Article.scala Inventory.scala Main.scala
```

- Exécutez le programme principal avec la commande suivante :

```
scala Main
```

Critères d'Évaluation

Votre solution sera évaluée sur les critères suivants :

1. Respect de la structure demandée.
2. Fonctionnalité et exécution correcte de chaque option du menu.
3. Utilisation appropriée des classes, objets, et méthodes.
4. Affichage clair et formaté des informations dans le terminal.