Projet : Gestion d'un distributeur automatique de boissons :

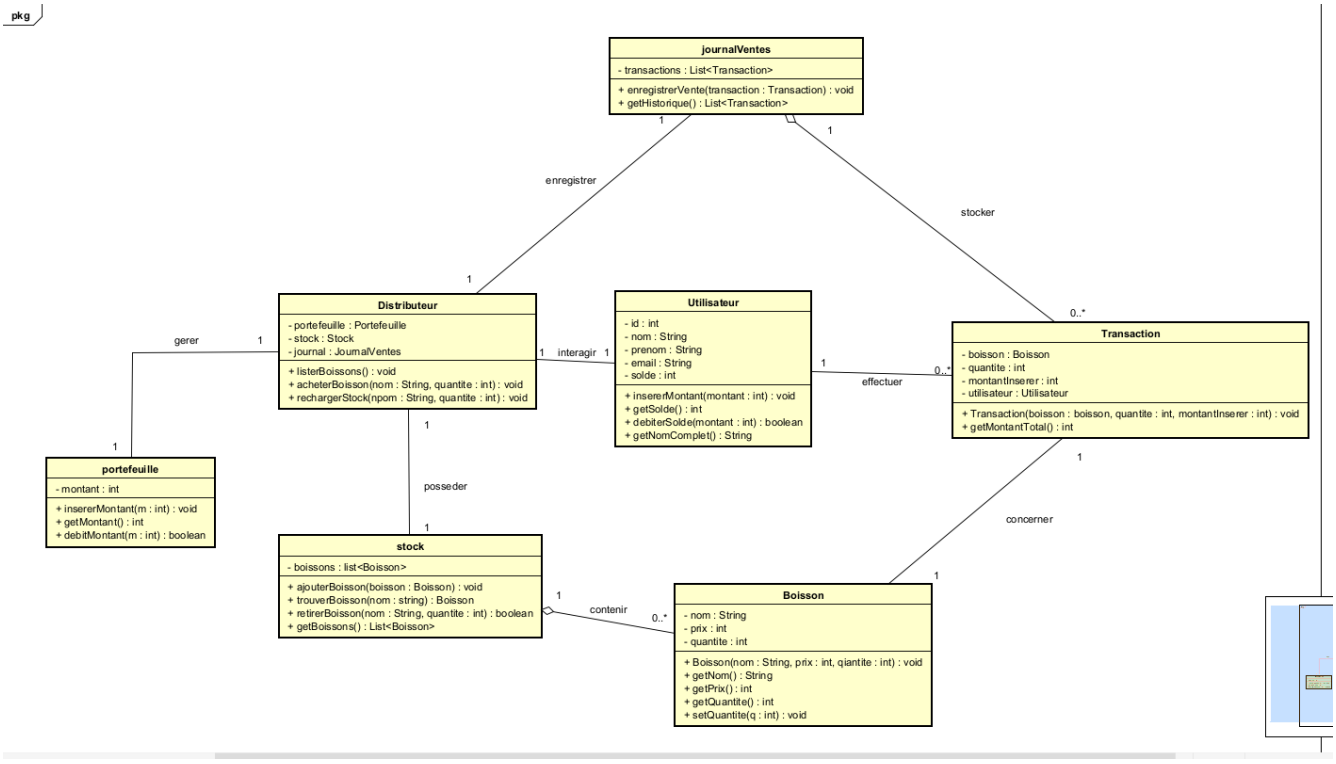
Nom des étudiants :

Cheikhouna Khadim Kebe

Serigne Fallou Seck

Mor Dia

Diagramme de classe :



Voici une explication du rôle de chaque classe de notre système de gestion de distributeur automatique de boissons, ainsi que les relations principales:

---

1. Classe Distributeur

Rôle :

Classe principale du système. Elle gère les opérations : afficher les boissons, accepter un montant, vendre une boisson, recharger le stock, enregistrer les ventes.

Relations principales :

- Contient un Portefeuille pour gérer le montant inséré.

- Possède un Stock pour accéder aux boissons disponibles.
- Utilise un JournalVentes pour enregistrer les transactions.

2. Classe Boisson

Rôle :

Représente une boisson vendue par le distributeur, avec son nom, prix et quantité.

Relations principales :

- Est contenue dans la liste de boissons du Stock.
- Est utilisée dans chaque Transaction pour identifier ce qui a été acheté.

3. Classe Portefeuille

Rôle :

Gère temporairement le montant inséré par l'utilisateur pour un achat.

Relations principales :

- Est composé dans le Distributeur (relation 1:1).
- Ne dépend pas du Stock ni des Transactions.

4. Classe Stock

Rôle :

Gère toutes les boissons disponibles dans le distributeur (ajout, retrait, recherche).

Relations principales :

- Contient une liste de Boisson (relation d'agrégation 1 → 0..*).
- Est composé dans le Distributeur (1:1).

5. Classe Transaction

Rôle :

Représente un achat effectué : boisson choisie, quantité, montant inséré, et utilisateur (optionnel).

Relations principales :

- Associe une Boisson à un achat.
- Est enregistrée dans le JournalVentes.
- Peut être liée à un Utilisateur (facultatif mais utile).

6. Classe JournalVentes

Rôle :

Archive toutes les transactions effectuées dans le distributeur.

Relations principales :

- Contient une liste de Transaction (agrégation 1 → 0..*).
- Est utilisé par le Distributeur pour l'historique des ventes.

7. Classe Utilisateur *(ajout optionnel mais recommandé)*

Rôle :

Représente une personne qui utilise le distributeur. Elle a un nom, un email et un solde personnel.

Relations principales :

- Peut effectuer plusieurs Transactions (1 → 0..*).
- Peut être lié à une transaction pour traçabilité.

Liste des 30 tests unitaires avec description :

N°		Description
1	testAffichageBoissons()	Vérifie que la liste des boissons est correctement affichée et contient 2 éléments.
2	testAchatBoissonAvecMontantSuffisant()	Vérifie qu'un achat est effectué correctement si le montant est suffisant.
3	testAchatMontantInsuffisant()	Vérifie que l'achat est refusé si le montant est insuffisant.
4	testAchatRuptureStock()	Vérifie qu'un achat est bloqué lorsque la boisson est en rupture de stock.
5	testAchatBoissonInexistante()	Vérifie qu'un achat échoue si la boisson n'existe pas dans le stock.
6	testRechargementStock()	Vérifie qu'on peut recharger une boisson existante dans le stock.
7	testRechargeBoissonInexistante()	Vérifie que la recharge échoue si la boisson est absente du stock.
8	testMontantNegatif()	Vérifie que l'insertion d'un montant négatif est ignorée.

N°		Description
9	testDebitImpossible()	Vérifie qu'on ne peut pas débiter plus que le montant disponible.
10	testAchatQuantiteMultiple()	Vérifie qu'on peut acheter plusieurs unités d'une boisson.
11	testGetMontant()	Vérifie que le montant inséré est bien enregistré dans le portefeuille.
12	testGetNomBoisson()	Vérifie que le nom d'une boisson est correctement retourné.
13	testGetPrixBoisson()	Vérifie que le prix d'une boisson est correct.
14	testGetQuantiteBoisson()	Vérifie que la quantité d'une boisson est correcte.
15	testSetQuantiteBoisson()	Vérifie qu'on peut modifier la quantité d'une boisson.
16	testVenteEnregistree()	Vérifie qu'une vente est bien enregistrée dans le journal après un achat.
17	testMontantTotalTransaction()	Vérifie que le montant total d'une transaction est correct (prix × quantité).
18	testHistoriqueVideInitialement()	Vérifie que le journal des ventes est vide au lancement.
19	testTransactionToString()	Vérifie que le texte affiché pour une transaction est lisible et contient le nom.
20	testJournalAjouteTransaction()	Vérifie que le journal ajoute correctement une nouvelle transaction.
21	testAjouterNouvelleBoisson()	Vérifie qu'on peut ajouter une nouvelle boisson au stock.
22	testAjoutMontantPlusieursFois()	Vérifie que plusieurs insertions de montant s'additionnent.
23	testRetraitBoissonInvalide()	Vérifie qu'on ne peut pas retirer une quantité supérieure au stock.
24	testRetraitBoissonValide()	Vérifie qu'on peut retirer une quantité disponible d'une boisson.
25	testStockVideSiToutesVentes()	Vérifie que le stock passe à 0 après avoir vendu toutes les unités.
26	testBoissonToStringFormat()	Vérifie que l'affichage de la boisson contient le nom et le prix.

N°	Description
27 testBoissonPrixInchange()	Vérifie que le prix d'une boisson reste inchangé après achat.
28 testSetQuantiteNegative()	Vérifie qu'on peut (actuellement) mettre une quantité négative (à améliorer).
29 testRetraitExactQuantiteDisponible()	Vérifie qu'on peut retirer exactement le stock restant.
30 testGetBoissonsListeNonVide()	Vérifie que la liste des boissons n'est pas vide après ajout.

TESTS D'ACCEPTANCE

1. Afficher les boissons disponibles

- Contexte : Le distributeur contient plusieurs boissons.
- Actions : L'utilisateur démarre le distributeur et demande la liste des boissons.
- Résultat attendu : La liste des boissons (nom, prix, quantité) s'affiche.

2. Achat réussi avec montant exact

- Contexte : L'utilisateur insère exactement 500 F pour une boisson à 500 F.
- Actions : L'utilisateur insère 500 F, choisit "Coca", quantité 1.
- Résultat attendu : La boisson est délivrée, le solde passe à 0, et le stock diminue de 1.

3. Achat échoué (montant insuffisant)

- Contexte : Boisson à 500 F, montant inséré : 300 F.
- Actions : L'utilisateur tente d'acheter une boisson sans solde suffisant.
- Résultat attendu : Message "Montant insuffisant", aucun changement dans le stock.

4. Achat échoué (boisson introuvable)

- Contexte : L'utilisateur veut acheter une boisson qui n'existe pas.
- Actions : L'utilisateur saisit "Ice Tea", qui n'est pas dans le stock.
- Résultat attendu : Message "Boisson non trouvée".

5. Achat échoué (stock insuffisant)

- Contexte : "Fanta" n’a que 2 unités restantes.
- Actions : L'utilisateur insère 1200 F et tente d’en acheter 3.
- Résultat attendu : Message "Stock insuffisant", rien ne change.

6. Recharge de stock réussie

- Contexte : Le personnel veut recharger "Fanta" de 5 unités.
- Actions : Appel à rechargerStock("Fanta", 5)
- Résultat attendu : La quantité de "Fanta" augmente de 5.

7. Recharge échouée (boisson inconnue)

- Contexte : Tentative de recharge d’une boisson non présente dans le stock.
- Actions : Appel à rechargerStock("Sprite", 3)
- Résultat attendu : Message "Boisson introuvable pour rechargement."

8. Affichage du solde inséré

- Contexte : L’utilisateur insère plusieurs montants successifs.
- Actions : 200 F puis 300 F.
- Résultat attendu : Solde total affiché : 500 F.

9. Rejet d’un montant négatif

- Contexte : L’utilisateur tente d’insérer -100 F.
- Actions : insererMontant(-100)
- Résultat attendu : Le montant n’est pas pris en compte.

10. Achat de plusieurs unités d’une même boisson

- Contexte : L’utilisateur souhaite acheter 2 Coca à 500 F l’unité.
- Actions : Il insère 1000 F, achète 2 Coca.
- Résultat attendu : Stock -2, solde = 0, transaction enregistrée.

11. Enregistrement de la transaction

- Contexte : Une boisson est achetée.
- Actions : Achat d’un "Fanta", 1 unité.
- Résultat attendu : Le journal contient la transaction correspondante.

12. Achat exact du dernier exemplaire

- Contexte : Il reste 1 Fanta.
- Actions : L’utilisateur achète ce dernier Fanta.
- Résultat attendu : Stock passe à 0, achat validé.

13. Retrait automatique du montant après achat

- Contexte : L’utilisateur insère 500 F et achète 1 Coca à 500 F.
- Actions : Achat exécuté.
- Résultat attendu : Le portefeuille est vidé (montant = 0).

14. Vérification du montant total d’une transaction

- Contexte : Achat de 2 boissons à 400 F.
- Actions : Création d’une transaction.
- Résultat attendu : Montant total = 800 F.

15. Historique initial vide

- Contexte : Distributeur tout juste lancé.
- Actions : Consultation du journal.
- Résultat attendu : Le journal est vide.