

Module : Architecture des SI II(Spring)

Classe (s) :TWIN GAMIX SIM

Enseignant (s) : Équipe Spring

Nombre de pages : 4 pages

Documents autorisés : OUI

Internet autorisée : NON

Calculatrice autorisée : NON

Durée : 01h30

Date : 09/01/2025

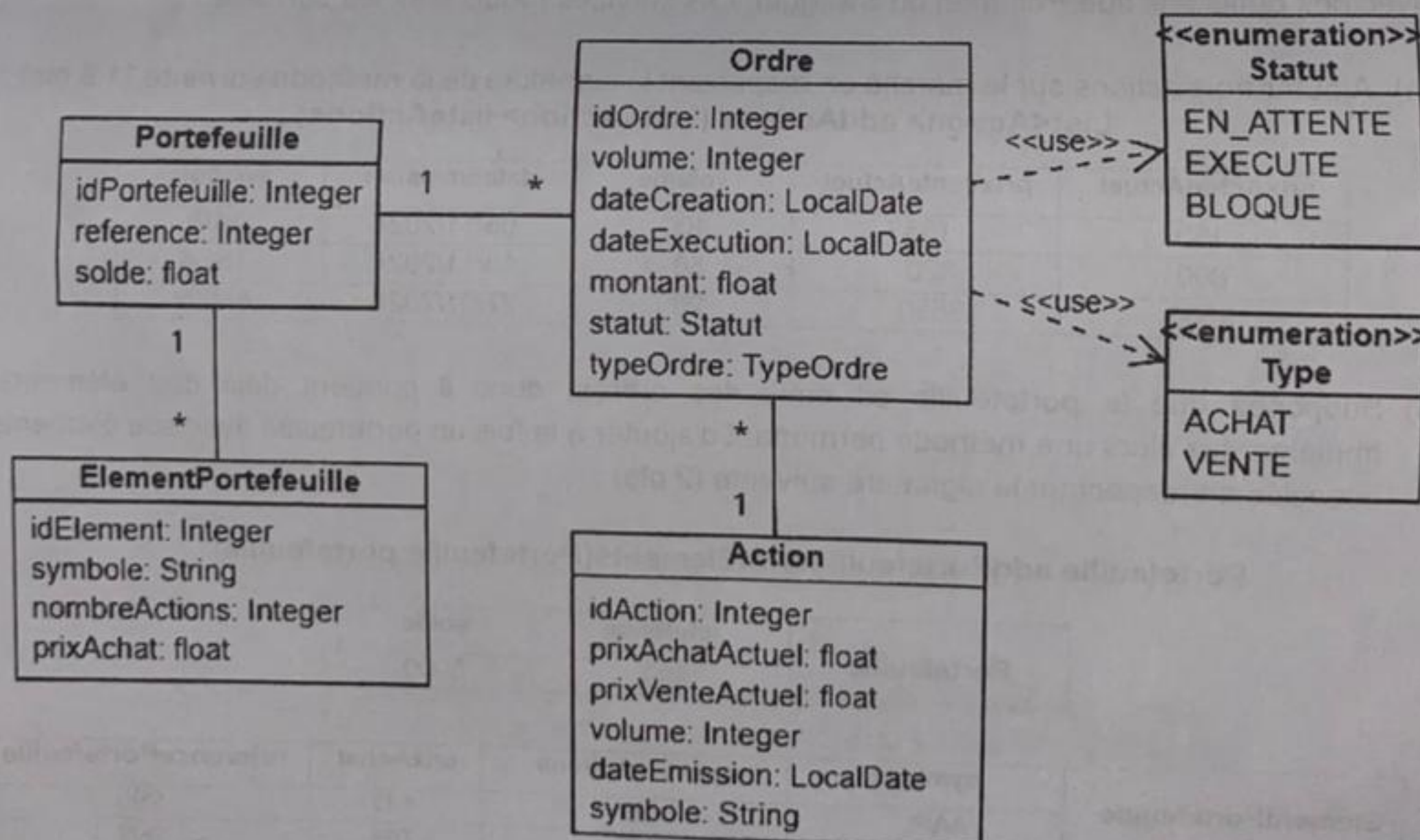
Heure : 09 :00h

La validation de l'épreuve est appliquée sur la base d'un code source exécutable.

Aucun code source non fonctionnel n'est comptabilisé lors de la validation.

Nous vous suggérons de mettre en place une application simplifiée conçue pour la gestion de portefeuilles en ligne, où les actions peuvent être gérées.

Ci-dessous le diagramme des classes :





## I. Entités/associations (6 points) :

Créez les entités nécessaires pour construire le schéma de la base de données conformément au diagramme des classes, en tenant compte des directives suivantes :

- Un portefeuille peut effectuer plusieurs ordres, et chaque ordre est émis par un seul portefeuille.
- Chaque ordre concerne une seule action qui peut être l'objet de plusieurs ordres.
- Chaque portefeuille peut consulter en détail ses éléments, et chaque élément appartient à un portefeuille.
- Les identifiants des entités sont auto-générés selon la stratégie "IDENTITY".
- Les énumérations doivent être enregistrées en tant que chaînes de caractères dans la base de données.
- La **référence** du portefeuille et le **symbole** de l'action doivent être **uniques**.

## II. Services (14points) :

Développez une classe annotée avec @RestController pour gérer les requêtes HTTP et interagir avec divers services. Exposez ces services à l'aide de Spring REST MVC, puis effectuez des tests avec des outils tels que Postman ou Swagger. Les services requis sont les suivants :

- a) Ajouter trois actions sur le marché en respectant la signature de la méthode suivante (1.5 pts) :
- List<Action> addActions (List<Action> listeActions) ;**

prixAchatActuel	prixVenteActuel	volume	dateEmission	symbole
150	155	100	05/11/2024	AAPL
800	820	50	13/11/2024	TSLA
3500	3550	20	27/07/2023	AMZN

- b) Supposez que le portefeuille ait émis des ordres, donc il contient déjà des éléments. Implémentez alors une méthode permettant d'ajouter à la fois un portefeuille avec ses éléments associés en respectant la signature suivante (2 pts) :

**Portefeuille addPortefeuilleWithElements(Portefeuille portefeuille) ;**

Portefeuille	reference	solde
	999	6000

ElementPortefeuille	symbole	nombreActions	prixAchat	referencePortefeuille
	AAPL	30	147	999
	TSLA	20	765	999



- c) Maintenant, nous abordons la gestion des ordres. En respectant la signature suivante, implémentez une méthode permettant à un portefeuille de passer un ordre. Chaque ordre créé doit être placé par défaut **en attente**, avec sa date de création fixée à la date **système** et son montant calculable selon la formule ci-dessous. (3.5pts)

Un ordre peut être soit d'achat, soit de vente. Le calcul du montant de l'ordre est défini comme suit :

- Si c'est un ordre **d'achat**, le montant de l'ordre est égal au prix d'achat actuel de l'action multiplié par le volume de l'ordre.
- Si c'est un ordre de **vente**, le montant de l'ordre est déterminé par le prix de vente actuel de l'action multiplié par le volume de l'ordre.

Vu que les ordres sont encore en attente, aucune modification du solde portefeuille ou de volume de l'action n'est envisagée à ce stade.

**Ordre addOrdreAndAffectToActionAndPortefeuille(Ordre ordre, String symbole, Integer reference) ;**

dateCreation	statut	typeOrdre	volume	symbole Action	reference Portefeuille
09/01/2025	EN_ATTENTE	ACHAT	20	AAPL	999
09/01/2025	EN_ATTENTE	ACHAT	20	TSLA	999
09/01/2025	EN_ATTENTE	VENTE	10	TSLA	999
09/01/2025	EN_ATTENTE	ACHAT	2	AMZN	999

- d) Après chaque ordre, on passe à la phase de vérification. Utilisez Spring AOP pour créer un Aspect qui affiche le message suivant (**en utilisant le logger**) après une bonne exécution de la méthode **addOrdreAndAffectToActionAndPortefeuille** de la couche service : "L'ordre est en train d'être traité !". (1.5 pts)
- e) En respectant la signature suivante et en utilisant obligatoirement le JPQL ou les Keywords, créer un service qui permet d'afficher la liste des symboles des actions qui ont fait l'objet d'au moins un ordre pour un portefeuille donné et dont la dateEmission de l'action est incluse entre les deux dates envoyées en paramètres. (2pts)

**List<String> listSymbolesActionsParPortefeuilleEtDateEmission(Integer referencePorteFeuille, LocalDate startDate, LocalDate endDate);**



f) Passons à la phase de vérification qui se fait sur tout ordre en **attente**. Selon le type de cet ordre, des vérifications de disponibilité de volume d'action et de solde de portefeuille sont prises en compte.

- Pour un ordre **d'achat**, il est essentiel de vérifier si le **volume d'actions** demandé est disponible et si le **solde** du portefeuille est suffisant pour couvrir le **montant** de l'achat. Si ces conditions sont remplies, le solde du portefeuille est mis à jour (solde portefeuille – montant ordre) ainsi que le volume de l'action (volume initial – volume ordre).
- Si les conditions **d'achat** ne sont pas remplies, un message d'avertissement s'affiche : "Solde insuffisant ou actions non disponibles" et l'ordre est **bloqué**.
- Dans le cas d'un ordre de **vente**, le solde du portefeuille est mis à jour (solde portefeuille + montant ordre). Ainsi que le volume de l'action est mis à jour (volume initial + volume ordre).

Tout ordre vérifié change de statut en **exécuté**, et prend la date actuelle comme **date d'exécution**.

En respectant la signature fournie, implémentez une méthode qui met en place la phase de vérification d'ordres **chaque deux minutes** en utilisant Spring Scheduler (3.5 pts)

`void checkOrdre() ;`

Les résultats dans la base de données suite à l'exécution du service sont les suivants :

Ordre						Portefeuille		Action	
dateExecution	montant	statut	typeOrdre	volume	symbole Action	reference Portefeuille	solde	symbole Action	volume
09/01/2025	3000	EXECUTE	ACHAT	20	AAPL	999	3000	AAPL	80
-	16000	BLOQUE	ACHAT	20	TSLA		3000	TSLA	50
09/01/2025	8200	EXECUTE	VENTE	10	TSLA		11200	TSLA	60
09/01/2025	7000	EXECUTE	ACHAT	2	AMZN		4200	AMZN	18

Bon travail 😊