JAVA Avancé TEST Intermédiaire

Projet JUDO

(4H)

SGC LEDEME

2022



# Règles de base

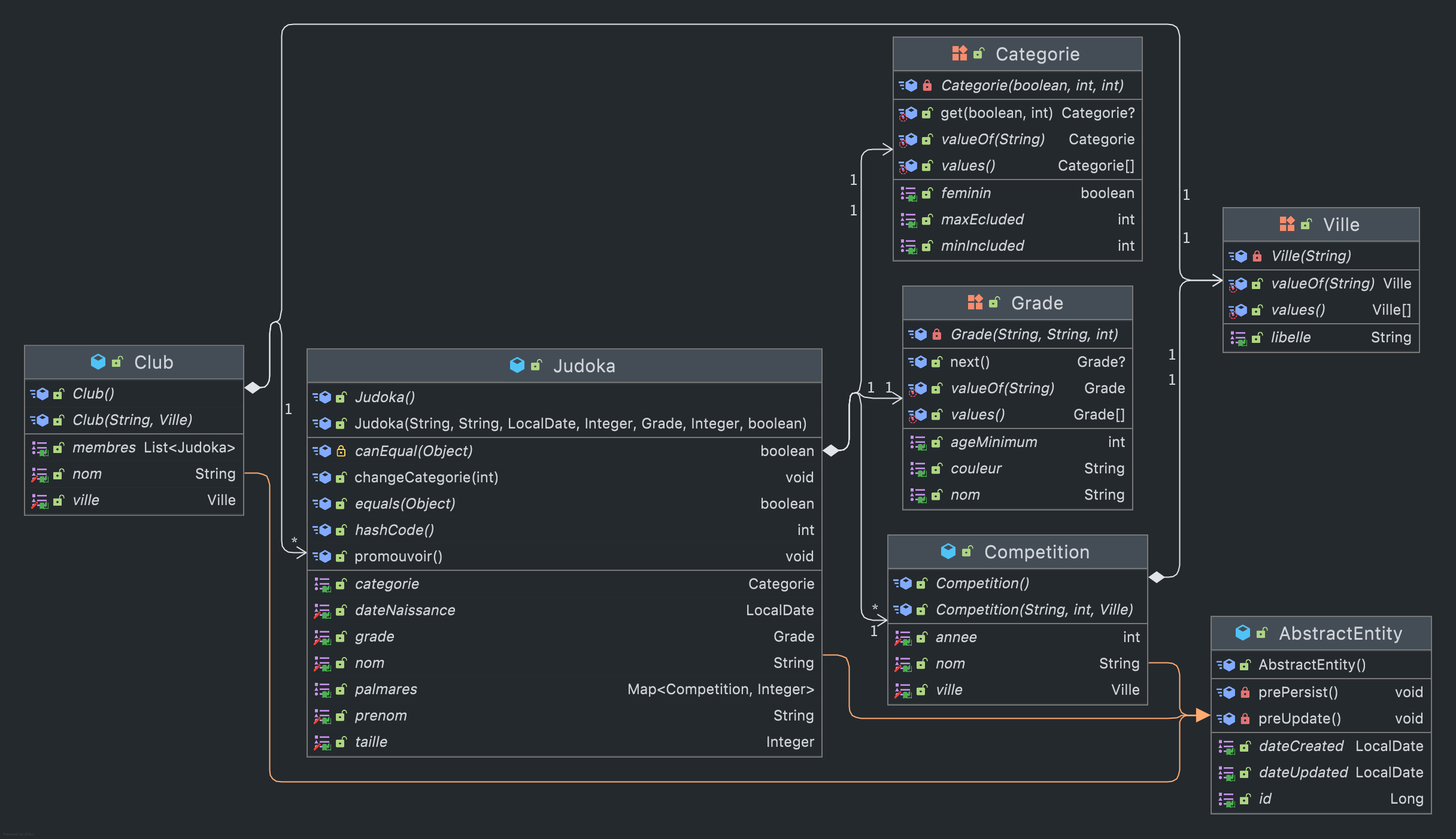
* Vous ne devez développer que la partie métier de l’application.
* Le code devra être entièrement documenté.
* Vous devez découper correctement votre code et interfacer les couches.
* Mettre en œuvre toutes les bonnes pratiques données en cours.
* Toutes les erreurs de validation devront pouvoir être transmises jusqu’à la vue non développée pendant le test.

# La base de données

La base de données devra se nommer « JAVA\_AVA\_TEST\_INT\_JUDO ».

Le nom du persistence-unit sera « pu-judo ».

# Les entités



Le fichier “entities.png” est aussi disponible dans l’archive du test.

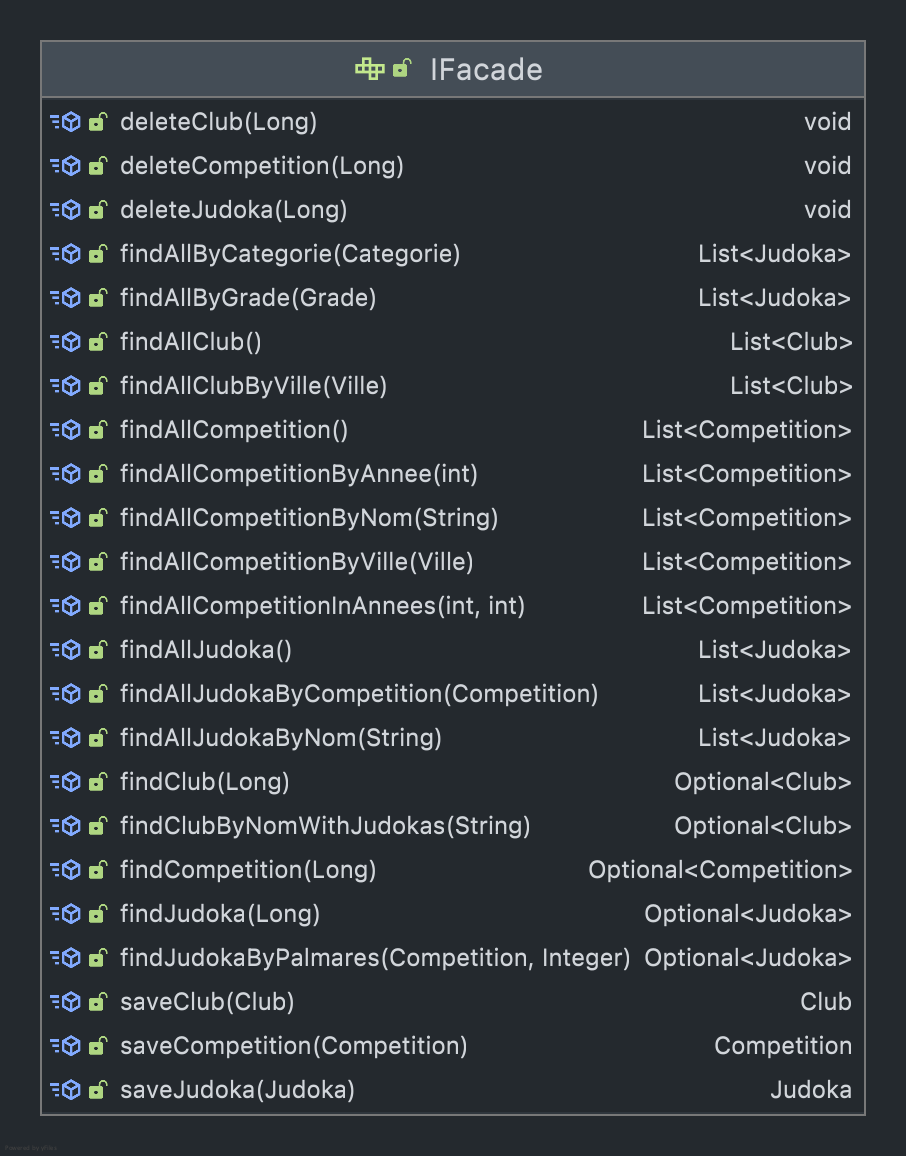
Règles de gestion :

1. Un club a :
   1. Un nom de taille 50 max, unique, et obligatoire.
   2. Une ville de taille 50 max obligatoire.
   3. Une liste de Judoka membres avec au moins 1 judoka.
   4. Son identité est basée sur le nom.
2. Un judoka a :
   1. Un nom de taille 50 max obligatoire
   2. Un prénom de taille 50 max obligatoire
   3. Une date de naissance obligatoire et obligatoirement dans le passé.
   4. Une taille positive.
   5. Une catégorie de taille 15 max obligatoire.
   6. Un grade de taille 10 max obligatoire.
   7. Un palmarès géré par un dictionnaire avec en clé une compétition et en valeur son classement.
   8. Son identité est basée sur le nom, prénom, date de naissance.
3. Une compétition a :
   1. Un nom de taille 50 max obligatoire.
   2. Une année obligatoire.
   3. Une ville de taille 50 max obligatoire.
   4. Son identité est basée sur le nom, l’année et la ville.

Méthodes à coder :

1. Dans l’énumération Catégorie :
   1. La méthode « get » prenant un booléen « pFeminin » et un entier « poids » permettant de renvoyer la bonne catégorie associée.
2. Dans l’énumération Grade :
   1. La méthode « next » permettant de renvoyer le grade suivant.
3. Dans la classe Judoka :
   1. Le constructeur prenant le nom, le prénom, la date de naissance, la taille, le grade, le poids et un booléen « isFeminine ».
   2. Le setter de grade vérifiant que le judoka a bien l’âge requis et sinon levant une exception « GradeMinimumAgeException »
   3. La méthode « promouvoir » qui donne le grade suivant au judoka.
   4. La méthode « changeCategorie » prenant un entier « pPoids » et qui associe au judoka la nouvelle catégorie par rapport à son nouveau poids.

# Les façades



Le fichier “IFacade.png” est aussi disponible dans l’archive du test.

La façade fournie est un exemple il sera peut-être judicieux de la découper en plusieurs et de transformer les types de retour en DTO et des DTO comportant vraiment le nécessaire.

Vous devez développer le métier d’une application connectée à une base de données PostgreSQL grâce à l’ORM Hibernate en respectant la spécification JPA.

Il vous faut mettre tout en œuvre pour respecter les méthodes demandées dans l’exemple de façade qui vous est fourni dans l’archive du test.

Pour finir il vous faut créer un fichier CSV avec au moins 5 lignes de données au format :

$NOM;$PRENOM;$DATENAISSANCE(dd/MM/YYYY);$TAILLE ;$POIDS ;$SEXE(M/F);$GRADE;$NOMCLUB;$VILLECLUB

Puis ajouter dans une façade une méthode « init » qui permet de les charger en BDD.

# Bonus

1. Charger le fichier CSV dans une seule transaction.
2. Loguer les erreurs rencontrées lors de l’insertion du fichier CSV

# Allons-y

A la fin du temps imparti vous fournirez le projet au format zip qui respectera le format suivant : $NOM\_$STAGE\_JAVAAVA\_INT\_TEST.zip

Une erreur dans le nom du zip vous fera perdre des points.

Un retard dans l’heure de rendu du test vous fera perdre des points.

勇敢些

« **Courage!** (yǒng gǎn xiē) »