

# JAVA Avancé Test Intermédiaire LE24 :

# Gestionnaire de jeux vidéo

# 4 heures

# ADC POTACZALA

## 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\_jv ».

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

## Contexte :

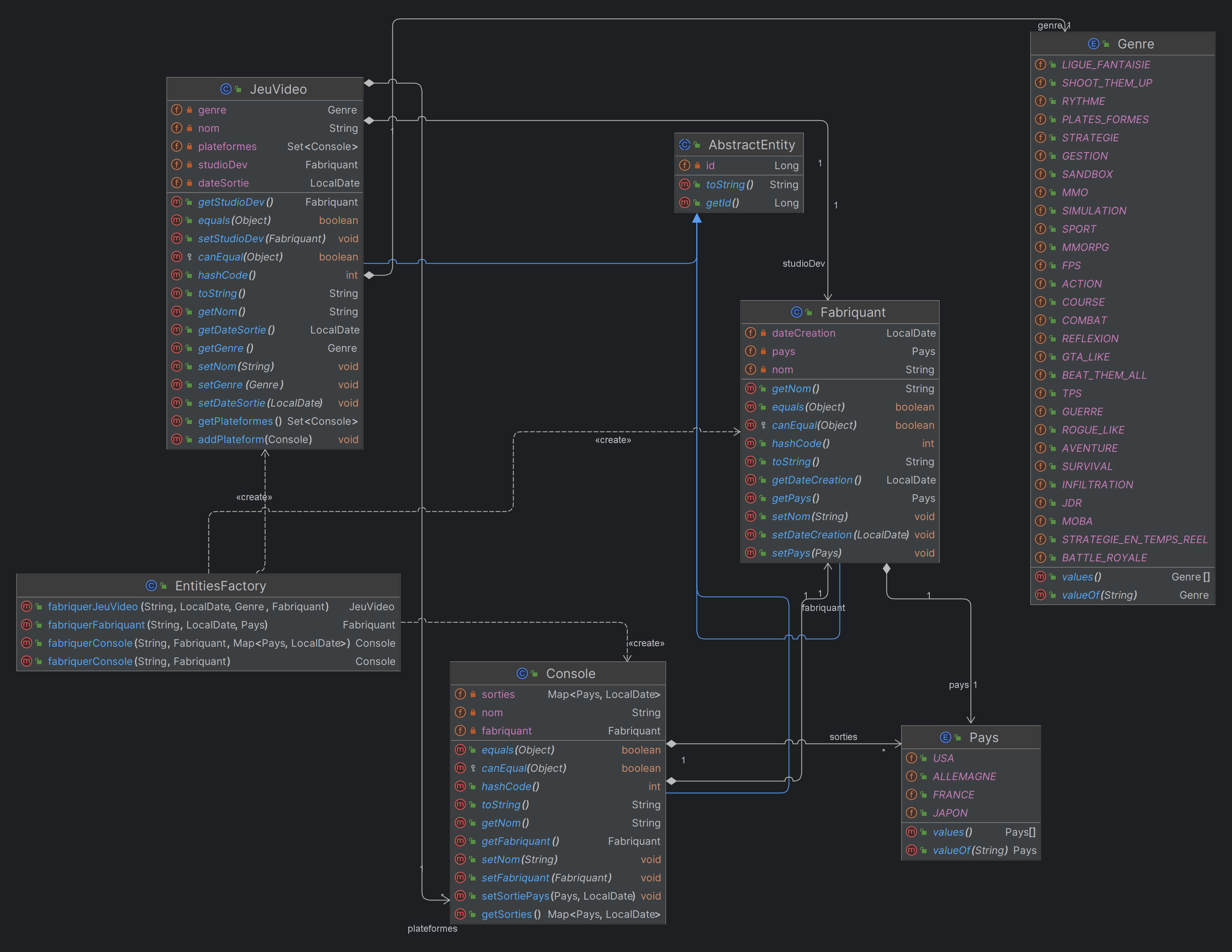
Vous êtes mandatés pour réaliser un Système d’Information (SI) au profit de la société MicroMegaGames.

En effet, il voudrait moderniser leur site internet pour vendre plus de jeux vidéo mais aussi faire de l’histoire sur ceux-ci.

La partie Web est gérée par une autre équipe de votre entreprise. Vous devez vous occupé de la partie Métier (Back).

## Objectifs :

Le but du SI est donc de gérer des jeux vidéo, notre analyste vous fournit le diagramme de classe suivant :



## Contraintes Métiers :

Pour le jeu vidéo :

* L’identité est faite sur le nom du jeu vidéo
* Le nom doit avoir au minimum 3 caractères max 80, unique, et obligatoire.
* La date de sortie est forcément dans le passé et obligatoire.
* Un jeu vidéo a un seul Fabriquant
* Un Set des Console qui sera toujours trié par ordre naturel, un jeu vidéo peut être sur plusieurs consoles
* Le genre du jeu vidéo doit avoir un maximum de 40 caractères
* Vous devez coder les méthode d’ajout et readAll du Set.

Pour la console :

* L’identité est faite sur le nom de la console
* Le nom de la console possède entre 2 et 50 caractères, unique, et obligatoire.
* La console possède un seul fabriquant.
* La console a une Map qui permet de connaître la date de sortie pour un pays.
* Les méthodes readAll et SetSortiePays (permettant d’ajouter dans la map) sont à coder.

Pour le fabriquant:

* L’identité se fait sur le nom du fabriquant
* Le nom de taille 5 min, 100 max, unique, et obligatoire.
* La date de création dans le passé et obligatoire
* Le Pays de taille 50 max et obligatoire

La façade fournie est un exemple il sera peut-être judicieux de la découper en plusieurs.

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.



Les méthodes particulières :

* recupererJeuxVideoParGenre(Genre) : Permet de récupérer tous les jeux vidéo du genre choisi
* recupererJeuxVideoDontLeNomCommenceParEtLeFabriquantEstDeTelPays(String,Pays) : Permet de récupérer tous les jeux vidéo dont le nom commence par : et dont le fabriquant est du pays sélectionner
* recupererLaConsoleDontLaSortieEstlaPlusAnciennePourUnPays(Pays) : Permet de récupérer la console qui est la plus vieille pour un pays(Attention une requête native peut simplifier la tâche)
* recupererConsoleAvecLeurJeuxVideo() : Permet de récupérer un dictionnaire contenant toutes les consoles avec leur jeux vidéo. (Attention : une requête n’est pas forcément attendue.)
* recupererDureeMoyenneSortieEntreJeuVideoPourUneConsole(Console console)

Permet de renvoyer une chaîne de caractère au format suivant : X années, Y mois, Z jours arrondi au jours près donnant la durée moyenne entre 2 sortie de jeux vidéo pour une console. (Attention c’est un algorithme qui est attendu et non une requête)

* Vous devrez charger dans une méthode initialisation() le fichier inti\_gestion\_jeuxvideo.csv, celui-ci sera placé dans les ressources du projet.

GL HF

