

Module : Architecture des SI II (Spring)
Enseignant (s) : Équipe Spring
Classe (s) : 4ARCTIC / 4NIDS
Nombre de pages : 4 pages

Documents autorisés : OUI
Calculatrice autorisée : NON
Internet autorisées : NON
Durée : 01h30

Date : 20/05/2022

Heure : 08h30

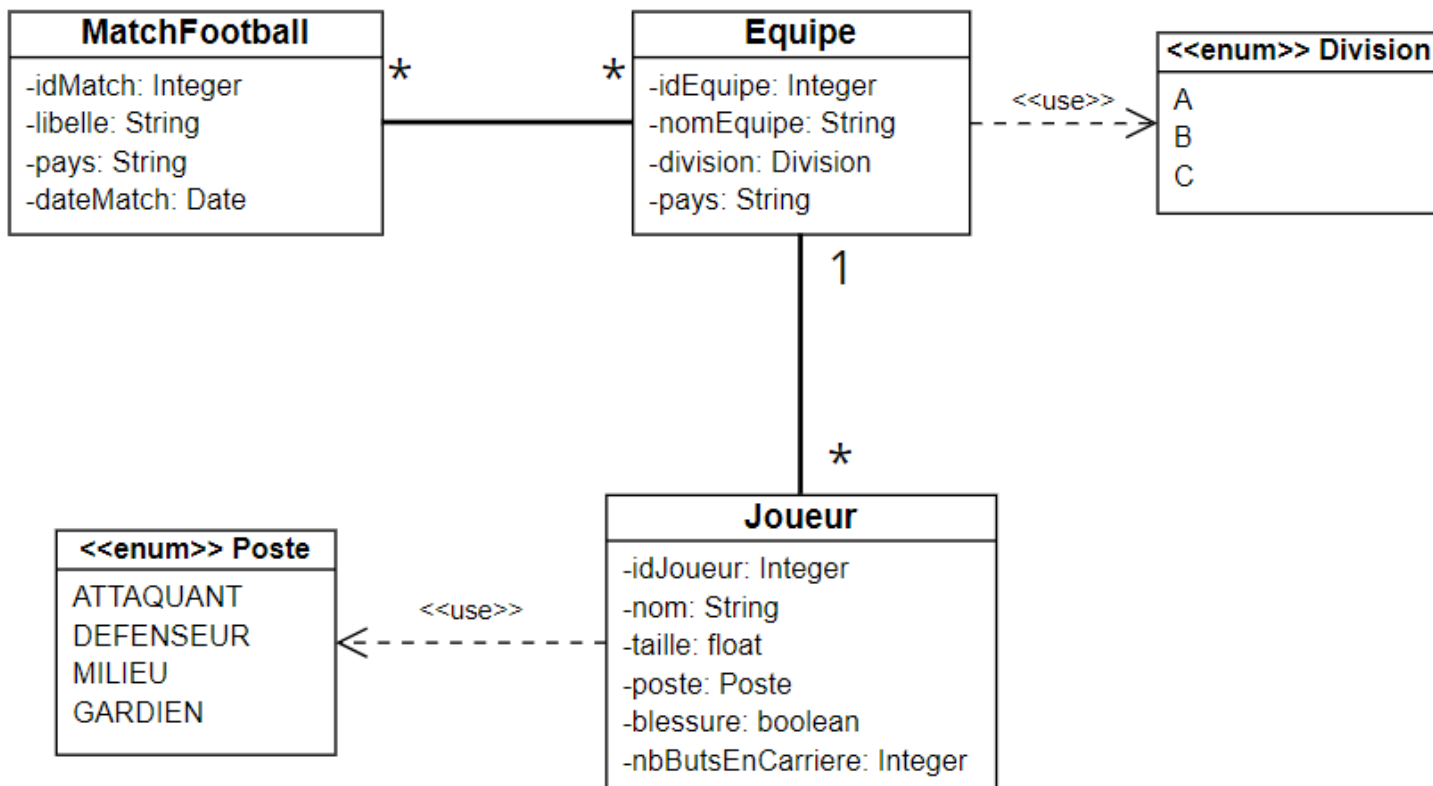
La validation de l'épreuve est appliquée sur la base d'un code source exécutable.

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

On vous propose d'implémenter une application simplifiée pour la gestion d'un match de football.

- Une équipe appartenant à la division A ou B ou C peut participer à plusieurs Matches
- Une équipe contient plusieurs joueurs et un joueur ne peut être affecté qu'à une seule équipe.
- Un joueur occupe l'un des postes suivants : Attaquant, Défenseur, Milieu, Gardien.

Ci-dessous le diagramme de classes :



I. Entités/associations (5points) :

Implémenter les entités qui permettent de générer le schéma de la base de données comme illustré dans le diagramme de classes sachant que :

- Les identifiants des entités sont auto-générés avec la stratégie « IDENTITY ».
- Dans l'association bidirectionnelle MatchFootball-Equipe, l'entité Equipe est le Child.
- Les énumérations doivent être stockées en tant que chaînes de caractères dans la base de données.

II. Services (15 points) :

Développer le code nécessaire dans une classe annotée par @RestController qui fait appel aux différents services. (Exposition des services avec Spring REST MVC, et Tests avec Postman ou Swagger).

Voici les Services demandés :

- 1) En respectant la signature de la méthode suivante, ajouter les équipes ci-dessous (1pt) :

public Equipe ajouterEquipe (Equipe equipe)

Equipe	nomEquipe	division	pays
	Barnachea	B	Chile
	Liverpool	A	Angleterre
	Real	A	Espagne

- 2) En respectant la signature de la méthode suivante, ajouter les joueurs ci-dessous et les affecter à leurs équipes (2pts) :

public Joueur ajouterJoueurEtAffecterEquipe(Joueur joueur, Integer idEquipe)

Joueur	nom	taille	blessure	nbButsEn Carriere	poste	equipe
	Carvajal	1.73	Oui	13	DEFENSEUR	Real
	Mohamed Salah	1.75	Non	357	ATTAQUANT	Liverpool
	Juan Ignacio	1,78	Non	25	DEFENSEUR	Barnachea
	Bale	1.85	Oui	285	ATTAQUANT	Real

- 3) Afficher la liste des joueurs qui occupent le poste Attaquant et dont la taille est supérieure à 1m79 en respectant la signature suivante (1.5pts) :

public List<Joueur> afficherJoueursParPosteEtTaille (Poste poste, float taille)

- 4) On souhaite ajouter les matchs de football suivants en les affectant aux équipes associées tout en vérifiant qu'une équipe ne peut pas être affectée deux fois au même match en respectant la signature suivante (2pts) :

public MatchFootball ajouterMatchFootballEtAffecterEquipe

(MatchFootball matchFootball, Integer idEquipe1, Integer idEquipe2)

Rq : Pour affecter une équipe, on peut initialiser la liste des équipes.

Match Football	libelle	pays	dateMatch	equipes
	Amical	Espagne	Aujourd'hui	Barnachea- Real
	UEFA	Espagne	2022-05-22	Real - Liverpool

- 5) Afficher la liste des défenseurs de la division A participant au match amical Barnachea- Real en respectant la signature suivante (2.5pts) :

**public List <Joueur> afficherJoueursDuMatchParDivisionEtPoste (Division division,
Poste poste, Integer idMatch)**

- 6) Afin de prendre des métriques associées à l'efficacité des joueurs, on souhaite afficher la liste des joueurs polyvalents disponibles.

- Un joueur est considéré polyvalent si son poste est défenseur avec un nombre de buts en carrière supérieur ou égale à 20.
- Un joueur disponible est un joueur non blessé.

En utilisant SpringScheduler, créer un service qui se déclenche toutes les 60 secondes en respectant la signature suivante (2.5pts) :

public void afficherJoueursPolyvalentsDisponibles ()

Le joueur Juan Ignacio est polyvalent

- 7) Implémentez une méthode permettant de reporter le match du 2022-05-22 d'une semaine (date du match +7 jours) sachant qu'un match peut être reporté s'il n'est pas encore joué et qu'il y a au moins une équipe avec deux blessures ou plus en respectant la signature suivante (2pts) :

```
public MatchFootball reporterMatch(@PathVariable Integer idMatch,  
  
@PathVariable @DateTimeFormat(pattern = "yyyy-MM-dd") Date dateReportee)
```

- 8) En utilisant Spring AOP afficher le message « Gestion match : Ajout réalisé avec succès » après chaque **bonne exécution** d'une méthode de la couche service commençant par «**ajouter**» (1.5pts)

Bon courage 😊