

Module : Architecture des SI II (Spring)

Enseignante : Chahnez Sardouk

Documents autorisés : OUI

Calculatrice autorisée : NON

Date : 06/05/2023

Heure : 14 :00

Classe : 1 ALINFO6

Nombre de pages : 4 pages

Internet autorisées : NON

Durée : 01h30

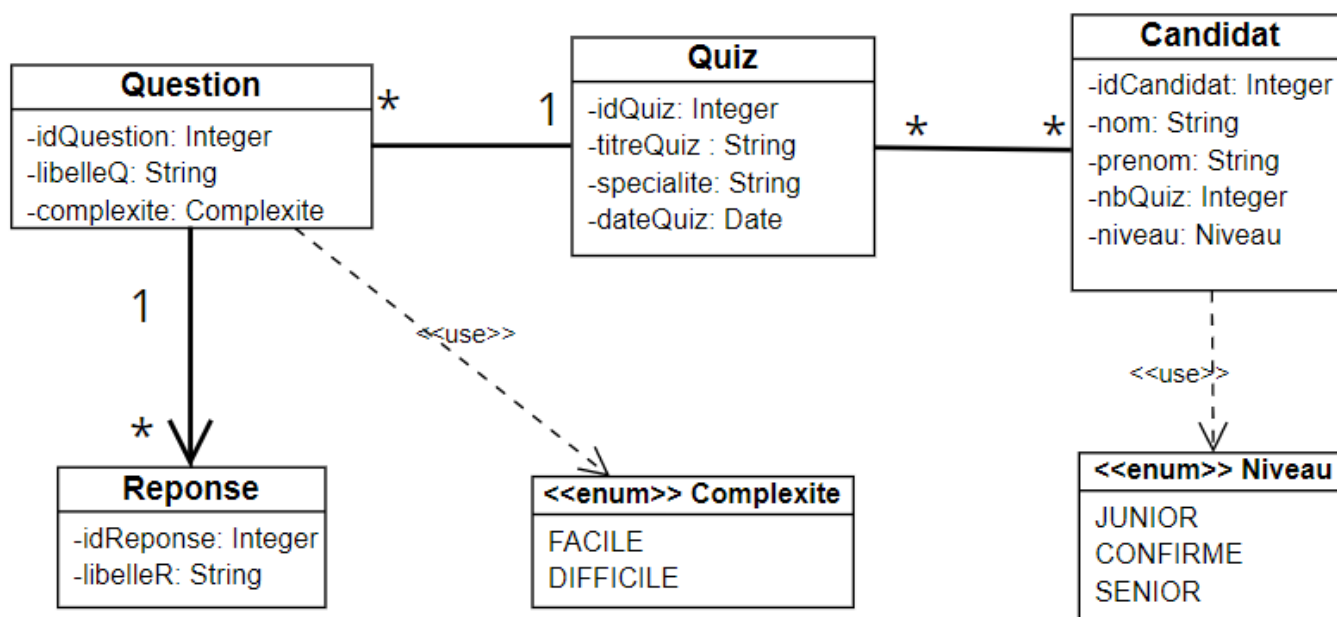
La validation de l'épreuve est appliquée sur la base d'un code source exécutable.

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 Quiz en ligne.

- Un candidat peut avoir un niveau : JUNIOR, CONFIRME ou bien SENIOR, et peut participer à plusieurs quizzes.
- Un quiz contient plusieurs questions mais une question ne peut être affectée qu'à un seul quiz.
- Une question contient plusieurs réponses.
- Une question peut avoir les niveaux de complexité suivants : FACILE, DIFFICILE.

Ci-dessous le diagramme de classes :



I. Entités/associations (6 points) :

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 **Quiz-Candidat**, l'entité Candidat est le **Child**.
- Les énumérations doivent être stockées en tant que **chaines de caractères** dans la base de données.

II. Services (14 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 candidats ci-dessous (1pt) :

public Candidat ajouterCandidat(Candidat candidat) ;

Candidat			
nom	prenom	nbQuiz	niveau
Abidi	Oussema	0	JUNIOR
Ben Ahmed	Chaima	0	CONFIRME
Romdhani	Mehdi	0	CONFIRME

2. On souhaite ajouter les quizzes suivants en respectant la signature de la méthode suivante (1pt) :

public Quiz ajouterQuiz(Quiz quiz) ;

Quiz		
titreQuiz	specialite	dateQuiz
Jenkins	DevOps	2023-05-20
Java	Developpement	2023-05-25
Spring	Developpement	Aujourd'hui

3. Affecter les quizzes aux candidats donnés en respectant la signature suivante (2.5pts) :

public Quiz affecterQuizCandidat(String titreQuiz, Integer idCandidat) ;

NB : Lors de l'affectation d'un nouveau quiz le nombre des quizzes d'un candidat doit être incrémenté par 1.

Candidat		Quiz
nom	prenom	titreQuiz
Abidi	Oussema	Jenkins
Ben Ahmed	Chaima	Java
Abidi	Oussema	Spring
Romdhani	Mehdi	Spring

4. Afficher la liste des candidats qui ont un niveau **CONFIRME** et qui vont passer un quiz de spécialité **Developpement** dans **les prochains jours** en utilisant les **KEYWORD** et en respectant la signature suivante (2.5pts) :

public List<Candidat> recupererCandidat (String specialite, Niveau niveau) ;

5. Implémenter une méthode permettant d'ajouter **à la fois** les questions et ses réponses et les affectent aux quizzes correspondent en respectant la signature suivante (2.5pts) :

public Question ajouterQuestEtRepEtAffecterQuestAQuiz(Question question, Integer idQuiz) ;

NB : Lors de l'ajout d'une question, il faut créer en même temps ses réponses.

Question		Reponse	Quiz
libelleQ	complexite	libelleR	titreQuiz
exception	DIFFICILE	try...catch	Java
batch	DIFFICILE	step	Spring
ioc	DIFFICILE	annotation	Spring
		xml	
jpa	FACILE	jpaRepository	Spring

6. En utilisant **Spring AOP**, créer un **Aspect** qui permet d'afficher le message suivant :
- " Début méthode « nom de la méthode » "

Cet aspect s'exécute **avant l'exécution** des méthodes d'**ajout** de la couche service (2 pts).

7. Proposer une méthode qui se déclenche toutes les **30 secondes** qui permet d'afficher **le quiz le plus difficile** en respectant la signature suivante (2.5 pts) :

public void recupererQuizPlusDifficile() ;

NB : Le quiz le plus difficile est celui qui comporte le nombre maximal des questions difficiles.

Bon courage 😊