

EXAMEN

Semestre: **I**
Session: Principale



Module : **Architecture des Systèmes d'Information II (Spring)**

Enseignants : **Spring Team**

Classes : **4SE – 4SLEAM – 4GAMIX**

Documents autorisés: **OUI**

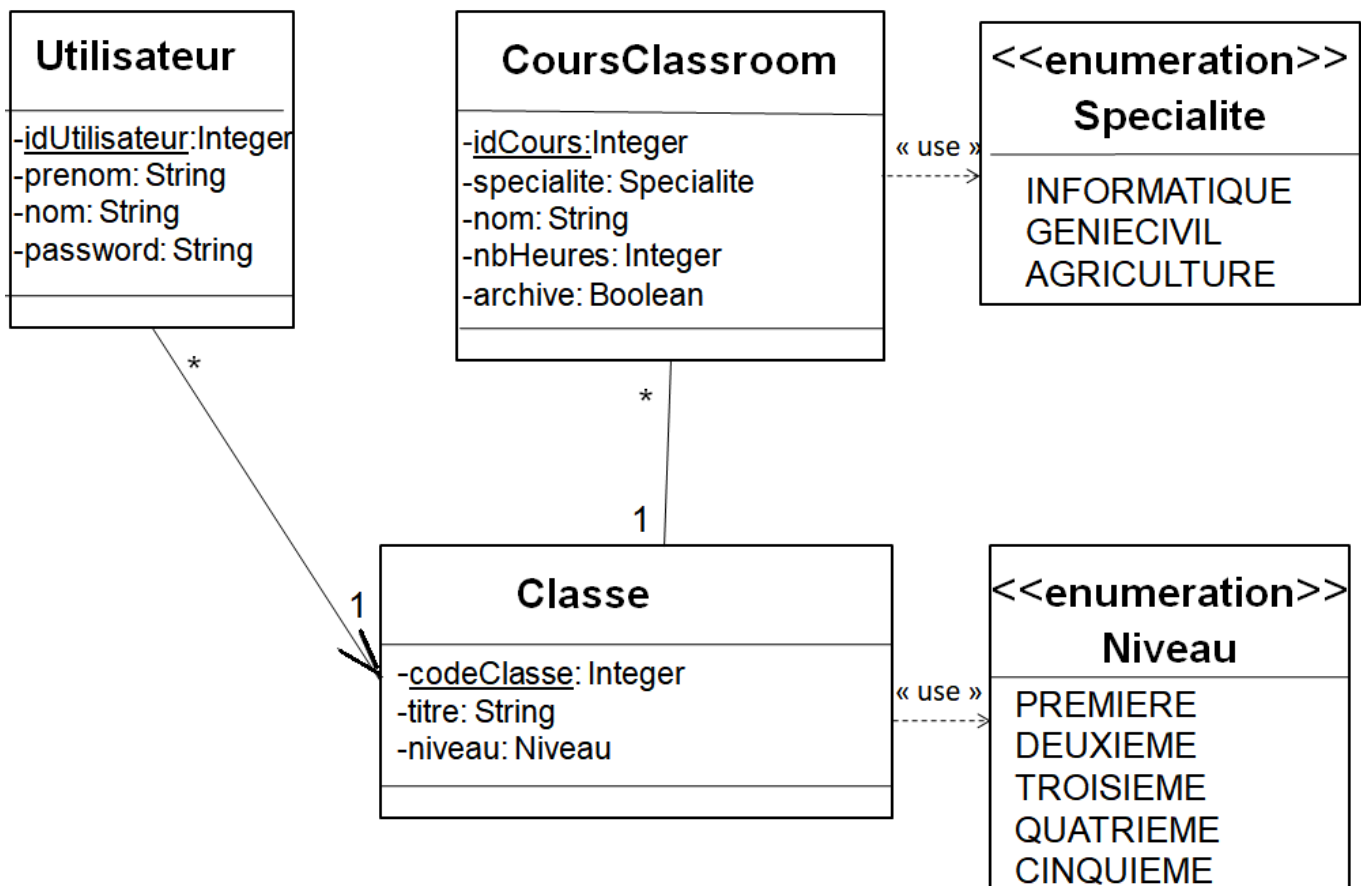
Internet autorisée : **NON**

Date : **12/01/2022** Heure : **09h00** Durée : **90 minutes**

Nombre de pages : **3**

**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 de gestion de cours créés dans des Classrooms pour les différents niveaux et pour les différentes spécialités. L'application est destinée à un établissement universitaire (exemple : ESPRIT). Voici son **diagramme de classes** :



Partie I (5 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 de toutes les entités sont auto-générés avec la stratégie «IDENTITY»
- La relation bidirectionnelle CoursClassroom - Classe indique qu'une Classe peut avoir plusieurs CoursClassroom, et un CoursClassroom ne peut être affecté qu'à une et une seule classe.
- Les énumérations doivent être stockées en tant que chaînes de caractères dans la base de données.

Partie II (15 points) :

Développer le code nécessaire dans une classe annotée @RestController qui fait appel aux différents services (Exposition de services avec Spring MVC REST).

Toutes les méthodes seront testées à travers **Swagger ou Postman**.

- a) Ajouter 2 utilisateurs ayant les détails ci-dessous en respectant la signature suivante **(1pt)** :

public Utilisateur ajouterUtilisateur (Utilisateur utilisateur)

prenom	nom	password
Amna	Ammar	etudiant
Ahmed	Slama	admin

- b) Ajouter 2 Classes ayant les détails ci-dessous en respectant la signature suivante **(1pt)** :

public Classe ajouterClasse (Classe c)

titre	niveau
4AG1	QUATRIEME
5EM1	CINQUIEME

- c) Ajouter 2 CoursClassrooms ayant les détails ci-dessous (en affectant chacun à la classe indiquée dans le tableau), en respectant la signature suivante **(2pts)** :

public CoursClassroom ajouterCoursClassroom (CoursClassroom cc, Integer idClasse)

specialite	nom	nbHeures	archive	classe
INFORMATIQUE	Programmation C	42	false	4AG1
AGRICULTURE	Plantes	25	false	4AG1
AGRICULTURE	Sciences Naturelles	40	false	4AG1

- d) Affecter les éléments déjà ajoutés (au niveau des questions a et b) de la sorte :
Affecter l'utilisatrice «Amna Ammar» à la classe «4AG1» et l'utilisateur «Ahmed Slama» à la classe «5EM1» en respectant la signature suivante **(2pts)**:

public void affecterUtilisateurClasse (Integer idUtilisateur, Integer idClasse)

- e) Indiquer le nombre d'utilisateurs pour le niveau «QUATRIEME», en respectant la signature suivante **(2pts)** :

public Integer nbUtilisateursParNiveau(Niveau nv)

- f) Désaffecter le coursClassroom «Plantes» de la classe «4AG1» en respectant la signature suivante **(1.5pts)** :

public void desaffecterCoursClassroomClasse(Integer idCours)

- g) En utilisant **SpringScheduler**, proposer une méthode qui se déclenche toutes les 60 secondes et qui archive (archive=true) tous les coursClassroom existants, en respectant la signature suivante **(1.5pts)** :

public void archiverCoursClassrooms()

- h) Créer une méthode qui détermine le nombre d'heures enseignées à une spécialité et un niveau donné, en respectant la signature ci-dessous. Tester la méthode sur le niveau «QUATRIEME» et la spécialité «AGRICULTURE» **(2pts)** :

public Integer nbHeuresParSpecEtNiv(Specialite sp, Niveau nv)

- i) Créer un **Aspect** qui permet de calculer et afficher sur la console de votre IDE le message «Début Exécution :>» avant l'exécution de chaque méthode, de la couche service, dont le nom commence par le mot **ajouter...** **(2pts)**

Excellent Travail !