

Unité d'enseignement : **Architecture N-tiers**

Modules : **Architecture SI II**

Documents autorisés : OUI ☐ NON ☒ Nombre de pages : 3

Internet autorisée : OUI ☐ NON ☒

Date : Décembre 2021

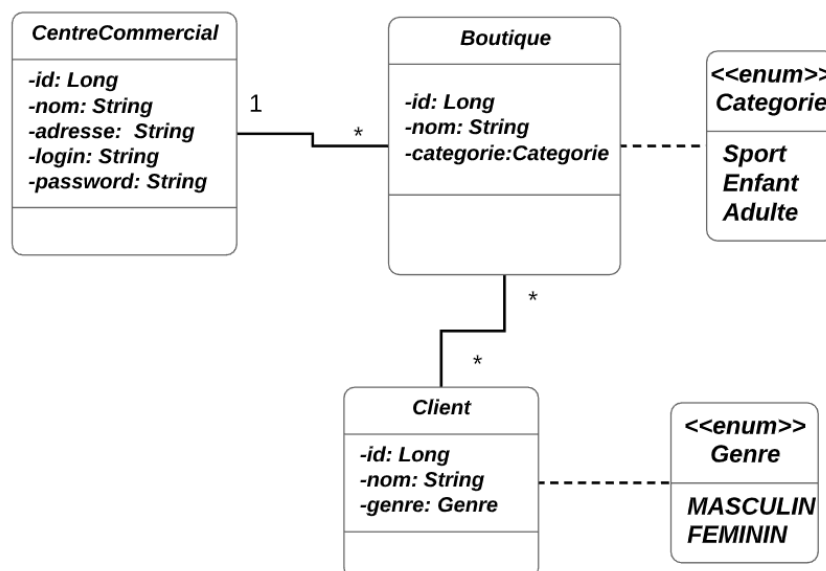
Durée : 1h30

**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.**

Le projet consiste à implémenter une application simplifiée de gestion d'un centre commercial.

**Ci-dessous le diagramme de classes :**

- La relation bidirectionnelle **Centre Commercial – Boutique : OneToMany** modélise le fait que dans un centre commercial, il peut y avoir plusieurs boutiques et une boutique n'appartient qu'à un seul centre commercial.
- La relation bidirectionnelle **Boutique- Client : ManyToMany** modélise le fait que plusieurs clients peuvent s'abonner à plusieurs boutiques et une boutique peut avoir plusieurs clients abonnés.



## Partie I : Spring DataJPA (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 sont auto-générés avec la stratégie « Identity ».
- L'énumération doit être stockée en tant que chaîne de caractères dans la base.

## Partie II : Service et Web Services (15 points) :

Pour **chacun** des exercices suivants, développer le code nécessaire dans **une classe service** qui injecte chacun un bean `@Repository`. Les services sont exposés par la suite entant que Web Services dans des beans Spring `@RestController`.

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

1. En respectant la signature de la méthode suivante, ajouter le centre commercial suivant : (3pts)

**void ajoutCentre(CentreCommercial centre)**

Centre Commercial	nom	adresse	login	password
	Ariana Center	Ariana	centreA	centreA

**NB:** Pour l'ajout d'un centre, il faut créer en même temps au moins une boutique (entité CentreCommercial avec l'entité associé Boutique).

Boutique	Categorie	nom
	Enfant	Funny Kids

2. En respectant la signature de la méthode suivante, ajouter à la fois la liste des boutiques suivante et l'affecter au centre commercial créé dans la question 1) : (2pts)

**void ajouterEtAffecterlisteBoutiques (List<Boutique> lb, Long idCentre)**

Boutique	Categorie	nom
	Adulte	New collection
	Sport	Sport City

3. En respectant la signature de la méthode suivante, ajouter les clients ci-dessous et affecter les aux boutiques créées dans la question 2) : (2pts)

**void ajouterEtAffecterClientBoutiques(Client client, List<Long> idBoutiques)**

Client	nom	genre	Liste des boutiques
	Ahmed Ali	MASCULIN	Sport City Funny Kids
	Hend Salem	FEMININ	New collection
	Sawsan salhi	FEMININ	New collection Funny Kids

4. En respectant la signature suivante, afficher la liste des clients d'une boutique donnée : (1.5pts)

**List<Client> listeClients(Long idBoutique)**

5. En respectant la signature suivante, afficher la liste des boutiques d'un centre commercial : (1.5pts)

**List<Boutique> listeBoutiques(Long idCentre)**

6. En respectant la signature suivante, afficher la liste de clients pour une catégorie donnée : (2pts)

**List<Client> listeDeClientsParCategorie(Categorie categorie)**

7. En respectant la signature suivante, créer un service programmé automatiquement, toutes les 30s, permettant d'afficher le nombre des clients par genre : (2pts)

**void nbreClientParGenre()**

8. Créer un Aspect qui permet de calculer et afficher dans les logs la durée d'exécution de chaque méthode appelée. (1pt)

**Bon travail ☺**