1. Les collections : TP Cyclistes

JAVA AVANCE

Etape 1

(IEF Christophe Louër)

1. Description

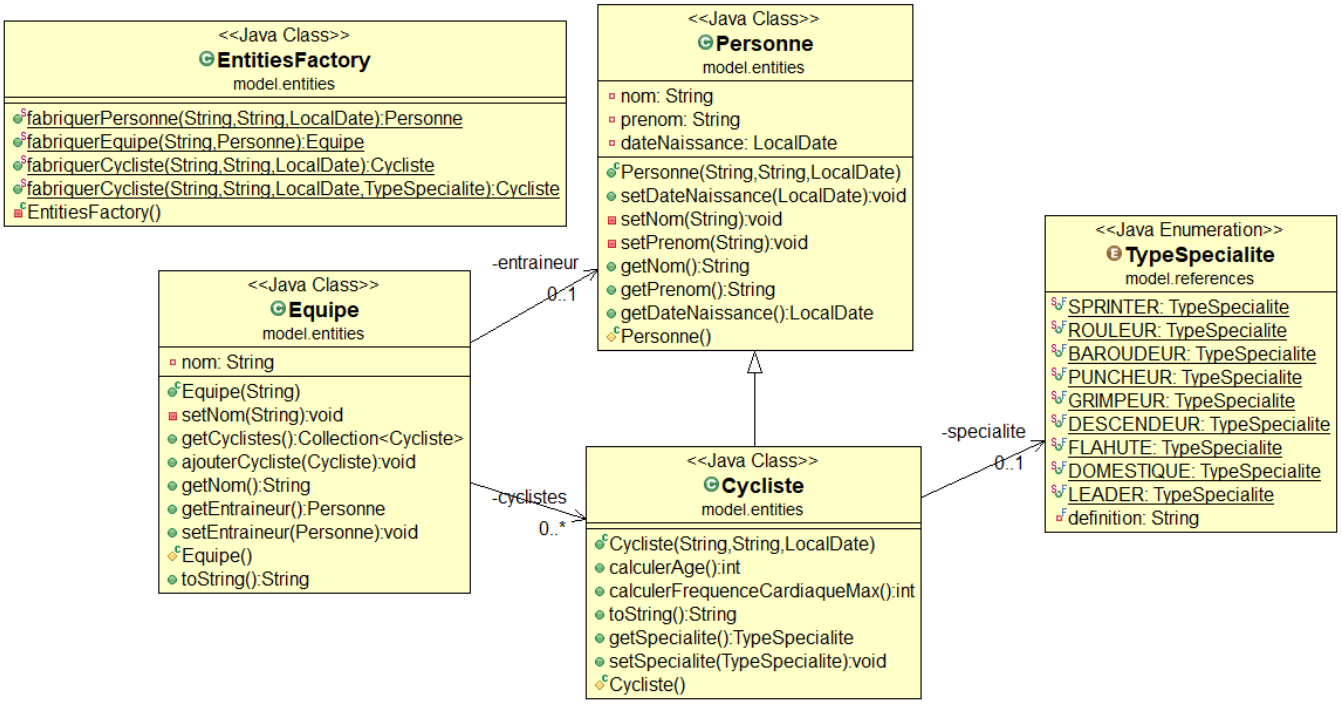
Nous allons développer, progressivement, une application de gestion d'équipes cyclistes :

```
+------------- Equipes cycliste -------------+
|  1 - Lister les équipes                     |
|  2 - Lister les coureurs d'une équipe       |
|  3 - Ajouter une équipe                     |
|  4 - Ajouter un coureur à une équipe        |
|  0 - sortir                                 |
+---------------------------------------------+
CHOIX :
```

Ce document présente le métier de l'application que vous devez implémenter.
Seront abordés ici les technologies suivantes :

- Maven : gestion de dépendance
- Lombok : annotations pour les éléments de base d'une classe
- Les collections

2. Diagramme de classe métier



DESCRIPTION DES CLASSES :

Enum ***TypeSpecialite***	Représente l'ensemble des spécialités dans le monde des cyclistes.
	Attention, cette énumération embarque une définition dont voici la liste :
	`SPECIALITE_SPRINTER = "Je suis relativement lourd et je cherche la victoire. Je mets à profit ma grande puissance en fin d'épreuve dans le but de gagner l'étape.";`
	`SPECIALITE_ROULEUR = "Je suis grand et adepte de la vitesse et des contre-la-montre.";`
	`SPECIALITE_BAROUDEUR = "Je m'échappe du peloton sur de longues distances.";`
	`SPECIALITE_PUNCHEUR = "Je dynamise le peloton et j'attaque les côtes courtes avec confiance.";`
	`SPECIALITE_GRIMPEUR = "Ayant un gabarit léger, je suis adepte des routes montagneuses.";`
	`SPECIALITE_DESCENDEUR = "Je suis très adroit dans les descentes, et j'en profite pour créer de l'écart.";`
	`SPECIALITE_FLAHUTE = "La pluie, le vent… je dompte les pavés et le mauvais temps.";`
	`SPECIALITE_DOMESTIQUE = "Je suis dévoué au leader qui ne pourrait pas réussir sans mon aide.";`
	`SPECIALITE_LEADER = "Je suis le cycliste désigné au sein de l'équipe comme ayant le meilleur potentiel pour remporter l'épreuve, la compétition.";`

Classe ***Personne***	Représente une personne dans l'application. L'identité est codée sur le nom, le prénom et la date de naissance (pas de gestion des homonymes) Le nom d'une personne est ~~composé de lettres alphabétiques~~ en majuscules. Le prénom est stocké en minuscule à l'exception de la première lettre. Une personne doit être majeur (plus de 18 ans).
Classe ***Cycliste***	Représente un cycliste et hérite de Personne. L'attribut ***specialite*** : c'est la spécialité, correspondant à l'énumération ***TypeSpecialite***, du cycliste (sa spécialité principale). Cet attribut n'est pas forcement renseigné à l'instanciation. La méthode ***calculerAge*** retourne l'age du cycliste. La méthode ***calculerFrequenceCardiaqueMax*** retourne la fréquence cardiaque du cycliste selon la formule : (plafond fréquence cardiaque) – age. Le plafond de la fréquence cardiaque est 220 pulsations.
Classe ***Equipe***	Représente une équipe dans l'application. Une équipe est instanciée avec son nom. Le nom de l'équipe est stocké en majuscules. L'attribut ***cyclistes*** représente la liste des cyclistes de l'équipe. Un cycliste n'est présent qu'une seule fois dans la liste.
Classe ***EntitiesFactory***	Regroupe les fabriques d'objets.

Vous disposez des classes de tests unitaires pour contrôler votre implémentation.