Projet LU2IN002 - 2020-2021

Numéro du groupe de TD/TME : 9

Nom : Oumar	*Nom :* Habbi	*Nom :*
Prénom : Youssouf	*Prénom :* Sarah	*Prénom :*
N° étudiant : 28610254	*N° étudiant :* 28601257	*N° étudiant :*

Thème choisi (en 2 lignes max.)
Gestionnaire de tâches administratives au sein d'un gymnase

Description des classes et de leur rôle dans le programme (2 lignes max par classe)

- Personne: permet de créer un objet Personne avec un nom et un prénom, classe mère de Membre.

- Etudiant : hérite de Personne dispose d'un attribut identifiant qui va contenir l'identifiant de l'étudiant, et embarque la méthode verifieIdentifiant(identifiant) pour le valider

- Membre: hérite de Etudiant et permet de créer un nouveau membre pour l'ajouter à un club du gymnase grâce à la méthode creerMembre().

- Club: classe abstraite, héritée par Boxe, Danse, et Judo, qui peuvent utiliser la méthode inscrireMembreClub() et afficherMembres().

- Boxe, Judo, Danse: classes pour définir des club différents par leur attribut nom et leur liste de membres.

- ClubPleinException: pour lancer une exception lorsqu'on souhaite ajouter un membre dans un club plein.

- ClubInexistantException: pour lancer une exception lorsqu'on souhaite ajouter un membre dans un club qui ne fait pas partie du gymnase.

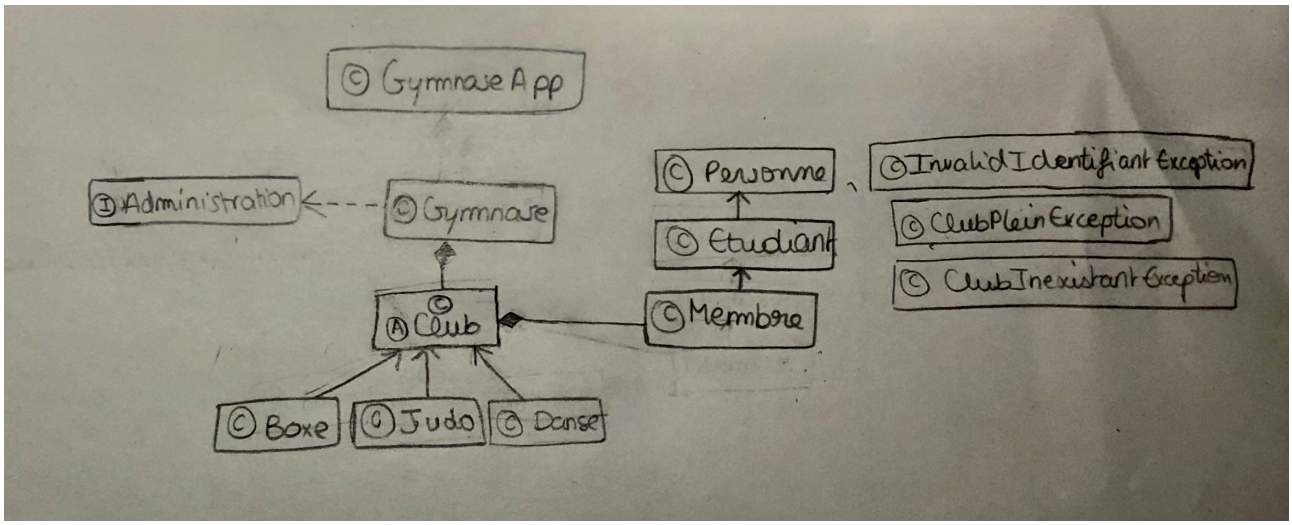
- InvalidIdentifiantException : pour vérifier que le membre qu'on veut inscrire est bien un étudiant, donc possède un bon identifiant (entier qui commence par 286 et compte 8 chiffre)

- Administration: interface contenant les méthodes spécifiques aux fonctionnalités implémentée par la classe Gymnase.

- Gymnase: classe contenant toutes les fonctionnalités de notre programme : inscrire un nouveau membre, afficher des informations sur un club, afficher des informations sur le gymnase…

- GymnaseApp: classe principale pour instancier un gymnase et tester notre programme.

Schéma UML des classes vision fournisseur (dessin "à la main" scanné ou photo acceptés)



Checklist des contraintes prises en compte:	*Nom(s) des classe(s) correspondante(s)*
Classe contenant un tableau ou une ArrayList	Gymnase(ArrayList<Club>), Club (ArrayList<Membre>)
Classe avec membres et méthodes statiques	Membre (méthode statique), Etudiant(méthode statique), Boxe (attribut statique), Judo (attribut statique), Danse (attribut statique)
Classe abstraite et méthode abstraite	Classe : Club Méthode : **public abstract void** inscrireMembreClub(Membre m) **throws** ClubPleinException
Interface	Administration
Définition de classe étendant Exception	ClubPleinException ClubInexistantException InvalidIdentifiantException
Gestion des exceptions	ClubPleinException ClubInexistantException NumberFormatException InvalidIdentifiantException
Gestion de flux	Gymnase Membre

Javadoc complète

Présentation de votre projet (max. 2 pages) : texte libre expliquant en quoi consiste votre projet.

Notre projet consiste en la gestion de tâches administratives au sein d'un gymnase. En effet, notre programme offre diverses fonctionnalités pour la gestion de différents clubs de sport : judo, boxe, et danse. La personne en charge des tâches administratives peut ajouter des adhérents, afficher les différentes informations et les membres d'un club.

Chaque club a une capacité d'accueil spécifique. Lors de l'inscription d'un nouveau membre (qui doit fournir un identifiant étudiant valide, gérer grâce à une exception) à un club, cette limite d'adhérents est respectée par la gestion aussi d'une exception.

Les différentes fonctionnalités sont accessibles via un menu principal s'affichant de nouveau lorsque l'utilisateur à terminé une tâche souhaitée. Tout cela s'effectue dans la console du terminal et grâce à des flux permettant d'interagir avec l'utilisateur.