

# Applications Informatiques II

Année académique : 2021-2022

## 1. Enoncé général :

Pour chacun des exercices, il vous est demandé :

1°) de permettre l'encodage, la modification, la suppression et la recherche sur différents critères des différents éléments cités dans l'énoncé.

2°) L'affichage de statistiques (totaux, moyennes, ...) et de rapports triés issus de projections et de jointures à partir des différents éléments du modèle.

3°) de gérer la persistance des données par enregistrement dans une base de données (API JDBC).

4°) De respecter l'architecture MVP, pour une application en mode texte.

RQ : ces consignes sont incomplètes, d'autres suivront.

D'une manière générale, les informations générées par vos traitements et requêtes seront présentées de manière LISIBLE (tri, regroupements, etc ...)

D'une manière générale prévoir les accesseurs correspondant aux différentes variables d'instance indiquées dans le diagramme de classe

## 2. Enoncés détaillés :

### Exercice 1\_1

Dans une école, on désire répertorier les éléments suivants :

- Les différents cours qui y sont organisés (Math, Français,Anglais,Informatique, ...)
- Les différents enseignants qui y enseignent( Mr Dupont, Mme Durant,...)
- Les différentes salles de l'école (F1,F2,C3,...)
- Les différentes classes qui y sont organisées (1<sup>ère</sup> A, 1<sup>ère</sup> B,2<sup>ème</sup> A,...)

Un même cours peut être organisé plusieurs fois dans des classes différentes , mais pour un couple classe-cours donné, on désigne toujours le même titulaire et la même salle :

Exemple : **Math en 1<sup>ère</sup> A** : en F1 avec Mr Dupont, **Math en 1<sup>ère</sup> B** : en C3 avec Mme Durant ,...

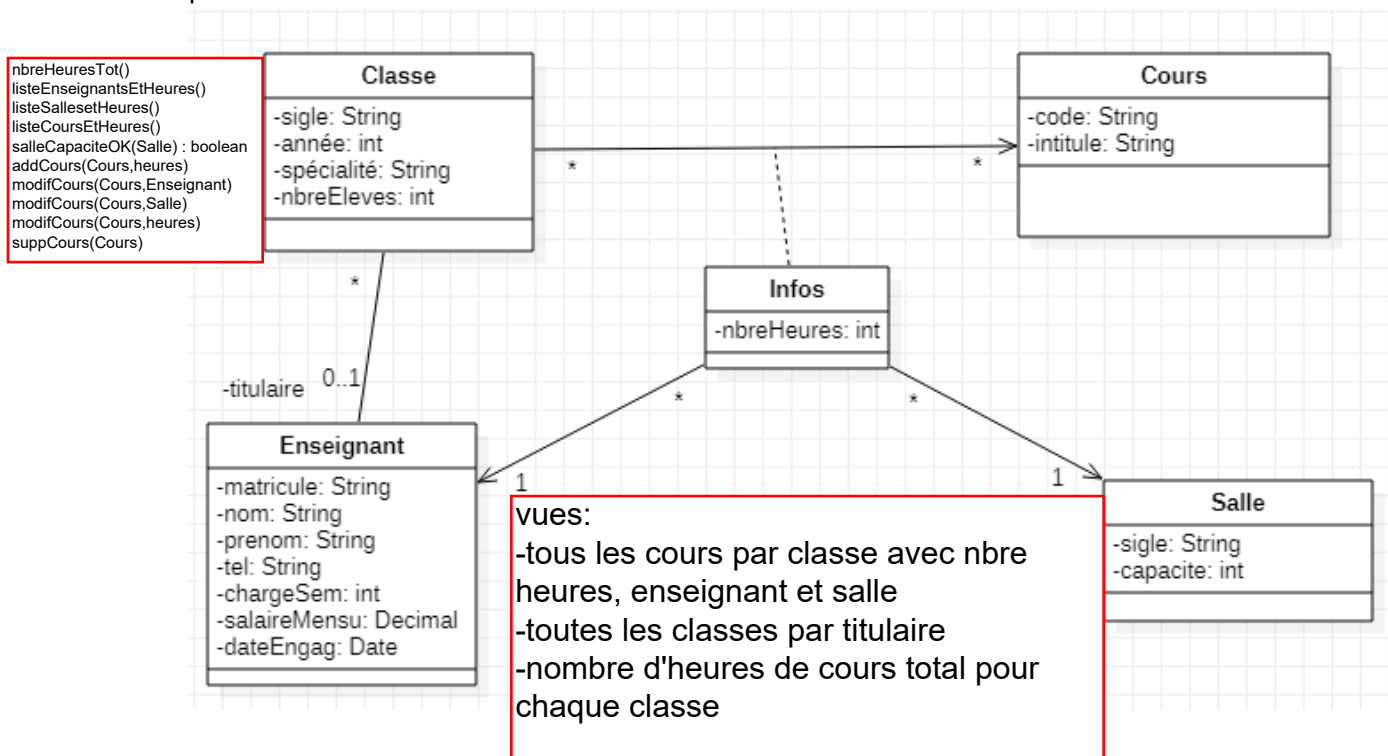
Pour répertorier les cours, on utilise la grille officielle du ministère qui octroie à chaque cours un code alphanumérique unique.

Pour répertorier les classe, on utilise notamment un sigle alphanumérique unique (Exemple : 1A,3D,...)

Pour répertorier les différents enseignants, on utilise leur numéro matricule unique, leur nom et leur prénom, leur numéro de téléphone nom+prénom+tel : triplet unique), le nombre d'heures qu'ils prestent chaque semaine (charge horaire hebdomadaire), leur salaire mensuel et leur date d'engagement.

Pour chacune des salles de l'école, on a enregistré son signe alphanumérique unique et son capacité.

Pour chaque classe on désigne un enseignant titulaire, un enseignant pouvant être titulaire de plusieurs classes différentes.



### Design Patterns :

-builder : Classe : sigle et année obligatoires.

-composite : Les classes appartiennent à des sections (id:int,nom:String) qui peuvent elles-mêmes comporter des sections. Totaliser le nombre d'élèves .

-observer : Les enseignants observent les classes en ce qui concerne la modification du nombre d'élèves

## Exercice 1\_2

Dans une école, on désire répertorier les éléments suivants :

- Les différents cours qui y sont organisés (Math, Français,Anglais,Informatique, ...)
- Les différents enseignants qui y enseignent( Mr Dupont, Mme Durant,...)
- Les différentes salles de l'école (F1,F2,C3,...)
- Les différentes classes qui y sont organisées (1<sup>ère</sup> A, 1<sup>ère</sup> B,2<sup>ème</sup> A,...)

Un même cours peut être organisé plusieurs fois dans des classes différentes , mais pour un couple classe-cours donné, on désigne toujours le même titulaire et la même salle :

Exemple : **Math en 1<sup>ère</sup> A** : en F1 avec Mr Dupont, **Math en 1<sup>ère</sup> B** : en C3 avec Mme Durant ,...

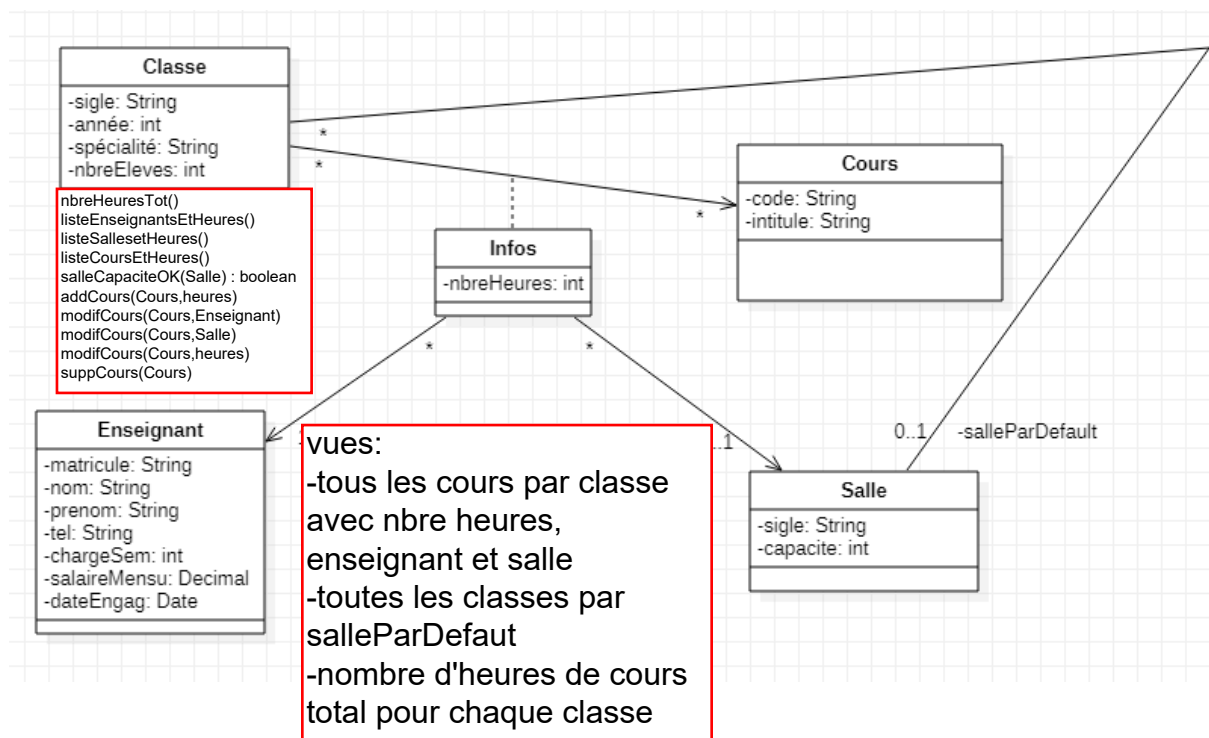
Pour répertorier les cours, on utilise la grille officielle du ministère qui octroie à chaque cours un code alphanumérique unique.

Pour répertorier les classe, on utilise notamment un sigle alphanumérique unique (Exemple : 1A,3D,...)

Pour répertorier les différents enseignants, on utilise leur numéro matricule unique, leur nom et leur prénom, leur numéro de téléphone nom+prénom+tel : triplet unique), le nombre d'heures qu'ils prestent chaque semaine (charge horaire hebdomadaire), leur salaire mensuel et leur date d'engagement.

Pour chacune des salles de l'école, on a enregistré son signe alphanumérique unique et son capacité.

Pour chaque classe on désigne une salle par défaut(représentant la salle ou la classe est censée se trouver quand elle n'a pas cours) , une même salle pouvant héberger plusieurs classes différentes.



### Exercice 1\_3

Dans une école, on désire répertorier les éléments suivants :

- Les différents cours qui y sont organisés (Math, Français,Anglais,Informatique, ...)
- Les différents enseignants qui y enseignent( Mr Dupont, Mme Durant,...)
- Les différentes salles de l'école (F1,F2,C3,...)
- Les différentes classes qui y sont organisées (1<sup>ère</sup> A, 1<sup>ère</sup> B,2<sup>ème</sup> A,...)

Un même cours peut être organisé plusieurs fois dans des classes différentes , mais pour un couple classe-cours donné, on désigne toujours le même titulaire et la même salle :

Exemple : **Math en 1<sup>ère</sup> A** : en F1 avec Mr Dupont, **Math en 1<sup>ère</sup> B** : en C3 avec Mme Durant ,...

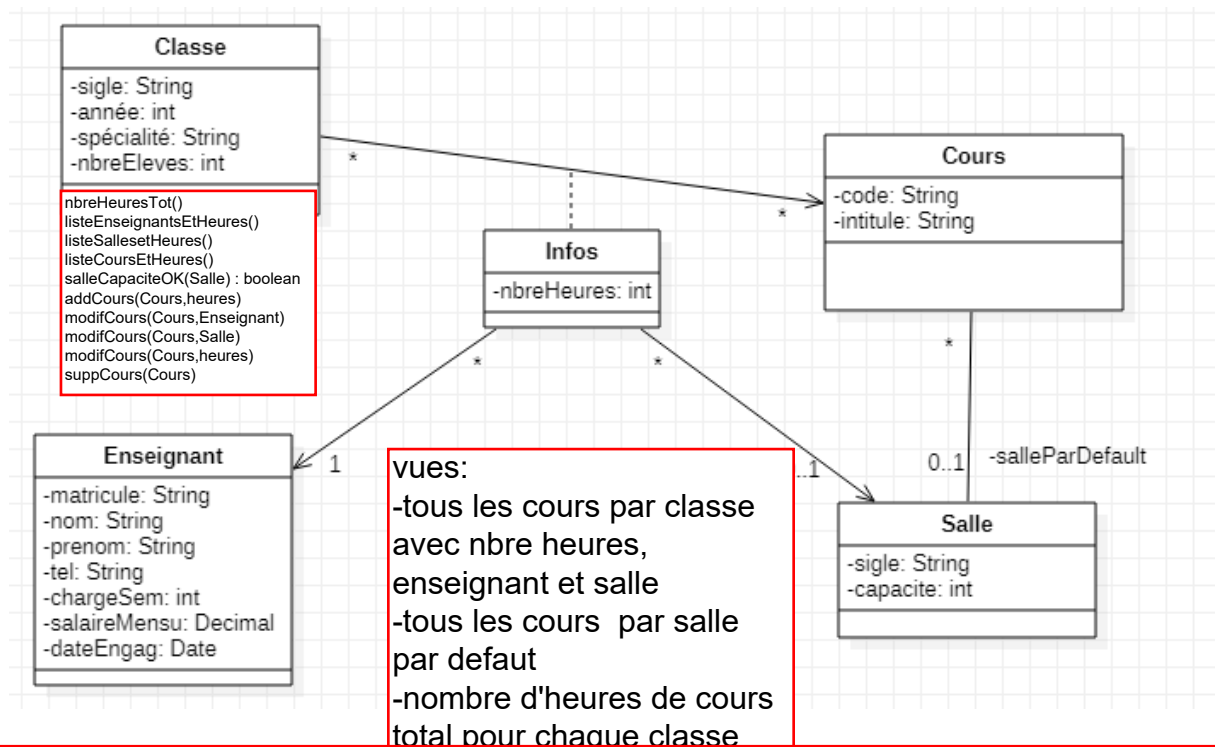
Pour répertorier les cours, on utilise la grille officielle du ministère qui octroie à chaque cours un code alphanumérique unique.

Pour répertorier les classe, on utilise notamment un sigle alphanumérique unique (Exemple : 1A,3D,...)

Pour répertorier les différents enseignants, on utilise leur numéro matricule unique, leur nom et leur prénom, leur numéro de téléphone nom+prénom+tel : triplet unique), le nombre d'heures qu'ils prestent chaque semaine (charge horaire hebdomadaire), leur salaire mensuel et leur date d'engagement.

Pour chacune des salles de l'école, on a enregistré son signe alphanumérique unique et son capacité.

Pour chaque cours on désigne une salle par défaut(représentant la salle où le cours devrait se donner de préférence, exemple Chimie au Labo de chimie) , une même salle pouvant héberger plusieurs cours différents.



## Exercice 1\_4

Dans une école, on désire répertorier les éléments suivants :

- Les différents cours qui y sont organisés (Math, Français,Anglais,Informatique, ...)
- Les différents enseignants qui y enseignent( Mr Dupont, Mme Durant,...)
- Les différentes salles de l'école (F1,F2,C3,...)
- Les différentes classes qui y sont organisées (1<sup>ère</sup> A, 1<sup>ère</sup> B,2<sup>ème</sup> A,...)

Un même cours peut être organisé plusieurs fois dans des classes différentes , mais pour un couple classe-cours donné, on désigne toujours le même titulaire et la même salle :

Exemple : **Math en 1<sup>ère</sup> A** : en F1 avec Mr Dupont, **Math en 1<sup>ère</sup> B** : en C3 avec Mme Durant ,...

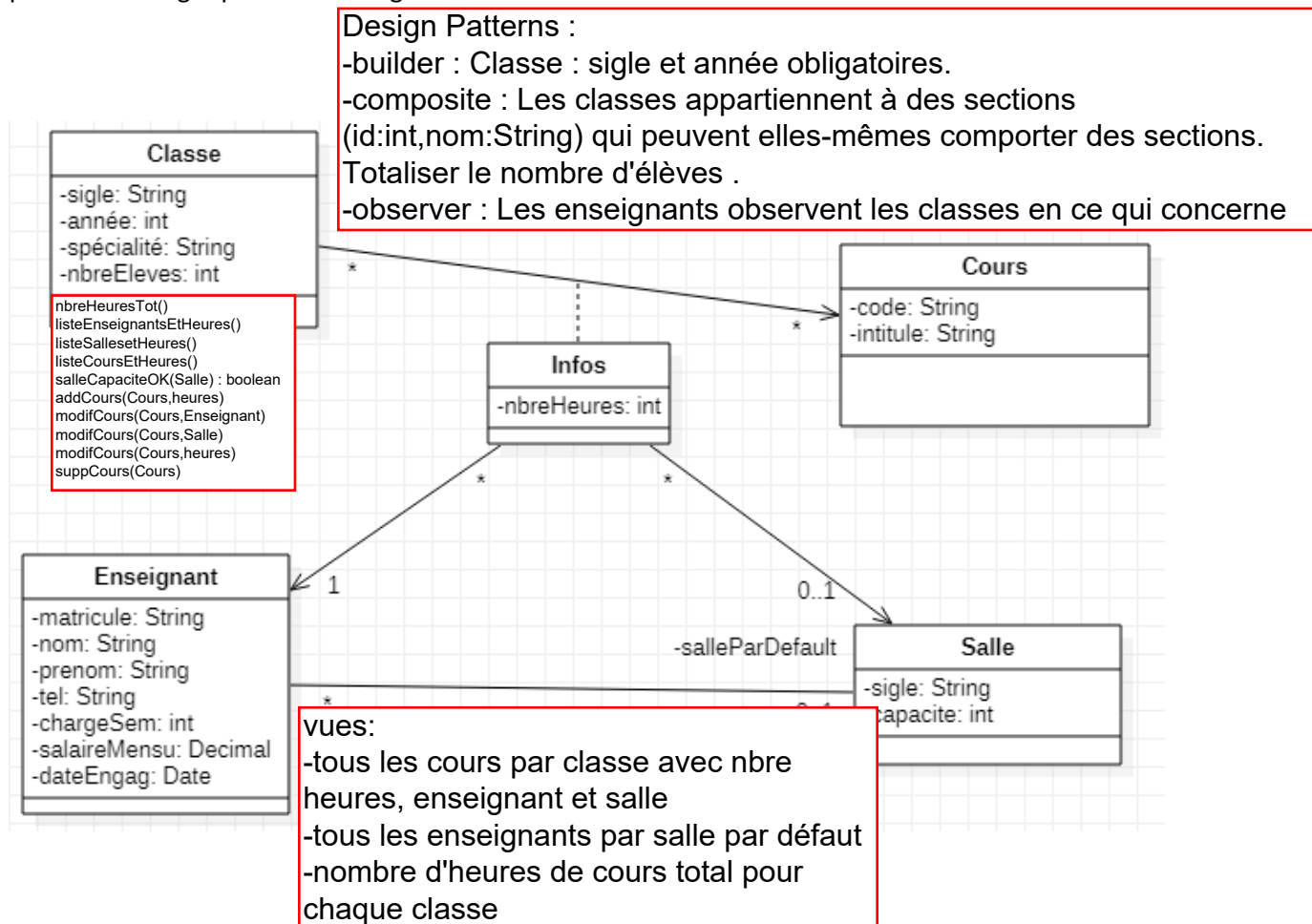
Pour répertorier les cours, on utilise la grille officielle du ministère qui octroie à chaque cours un code alphanumérique unique.

Pour répertorier les classe, on utilise notamment un sigle alphanumérique unique (Exemple : 1A,3D,...)

Pour répertorier les différents enseignants, on utilise leur numéro matricule unique, leur nom et leur prénom, leur numéro de téléphone nom+prénom+tel : triplet unique), le nombre d'heures qu'ils prestent chaque semaine (charge horaire hebdomadaire), leur salaire mensuel et leur date d'engagement.

Pour chacune des salles de l'école, on a enregistré son signe alphanumérique unique et son capacité.

Pour chaque enseignant on désigne une salle par défaut(représentant la salle où l'enseignant donne cours de préférence, exemple Mr Dupont donne cours de préférence au Local C3) , une même salle pouvant héberger plusieurs enseignant différents.



## Exercice 2\_1

Lors d'un championnat automobile, on désire répertorier les différentes courses auxquelles sont inscrits les différents pilotes participant au championnat. On désire notamment afficher pour chaque course, la liste de ses participants. Un pilote peut très bien ne pas s'inscrire à l'une des courses et une course peut très bien n'avoir aucun participant.

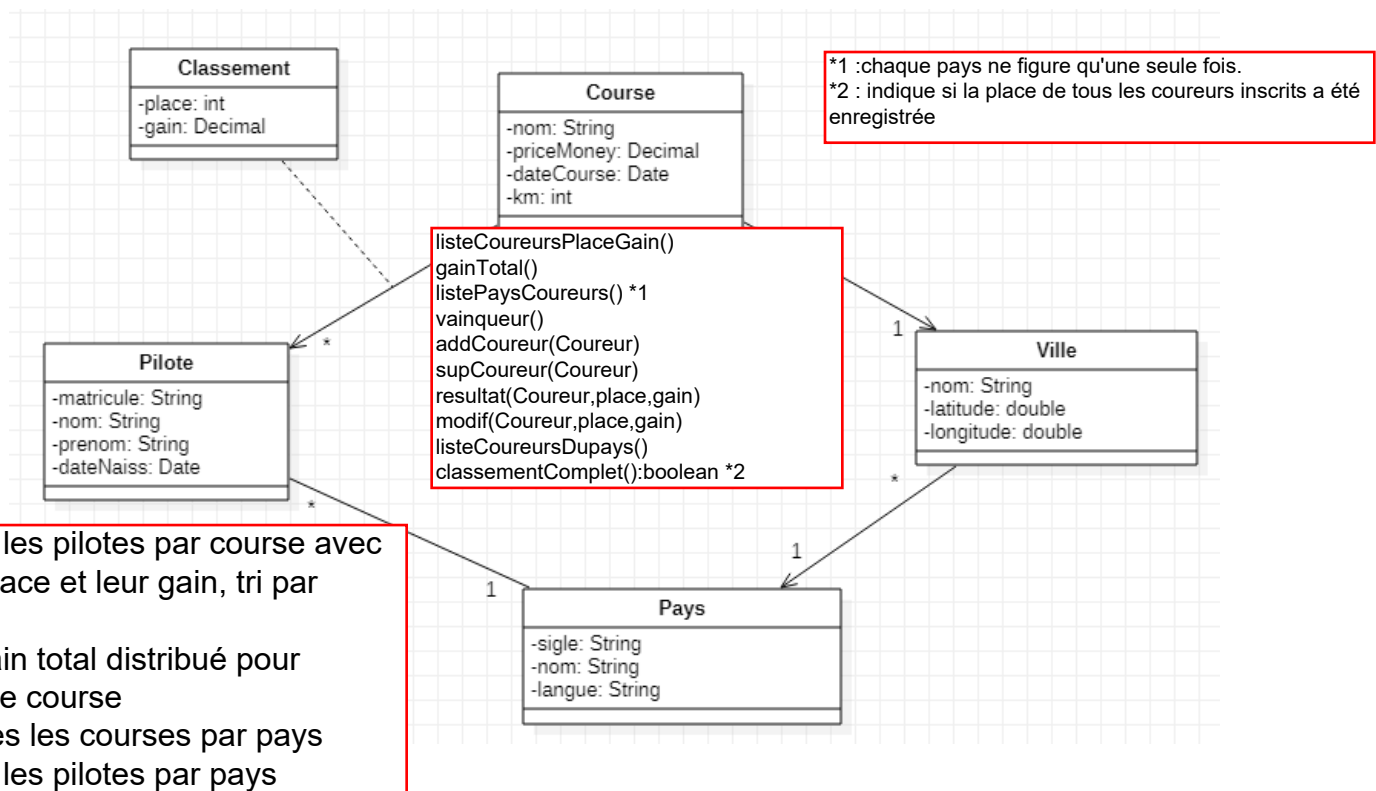
Pour chacun des pilotes on a enregistré leur numéro de licence unique, leur nom, prénom, date de naissance et nationalité (en fait leur pays).

Pour chacune des courses on a enregistré son nom unique, son prix money, la date à laquelle elle se déroule, son nombre total de km et la ville laquelle elle se déroule.

Pour chacune des villes, on a enregistré son nom, ses coordonnées géographiques (latitude et longitude) et le pays où elle se situe. Le couple latitude-longitude est unique. Les noms de ville **ne sont pas** uniques.

Pour chacun des pays, on a enregistré son sigle international unique (be, fr,...), son nom et sa langue principale.

On enregistre également pour chaque coureur ayant participé à une course son classement final et son gain en argent pour cette course. Si le coureur abandonne, la place vaut -1.



### Design Patterns :

- builder : Pays : sigle et nom obligatoires.
- composite : Les courses appartiennent à des championnats (id:int,nom:String) qui peuvent eux-mêmes comporter des championnats. Totaliser le prize money .
- observer : Les pilotes observent les courses en ce qui concerne la modification de la date de la course.

## Exercice 2\_2

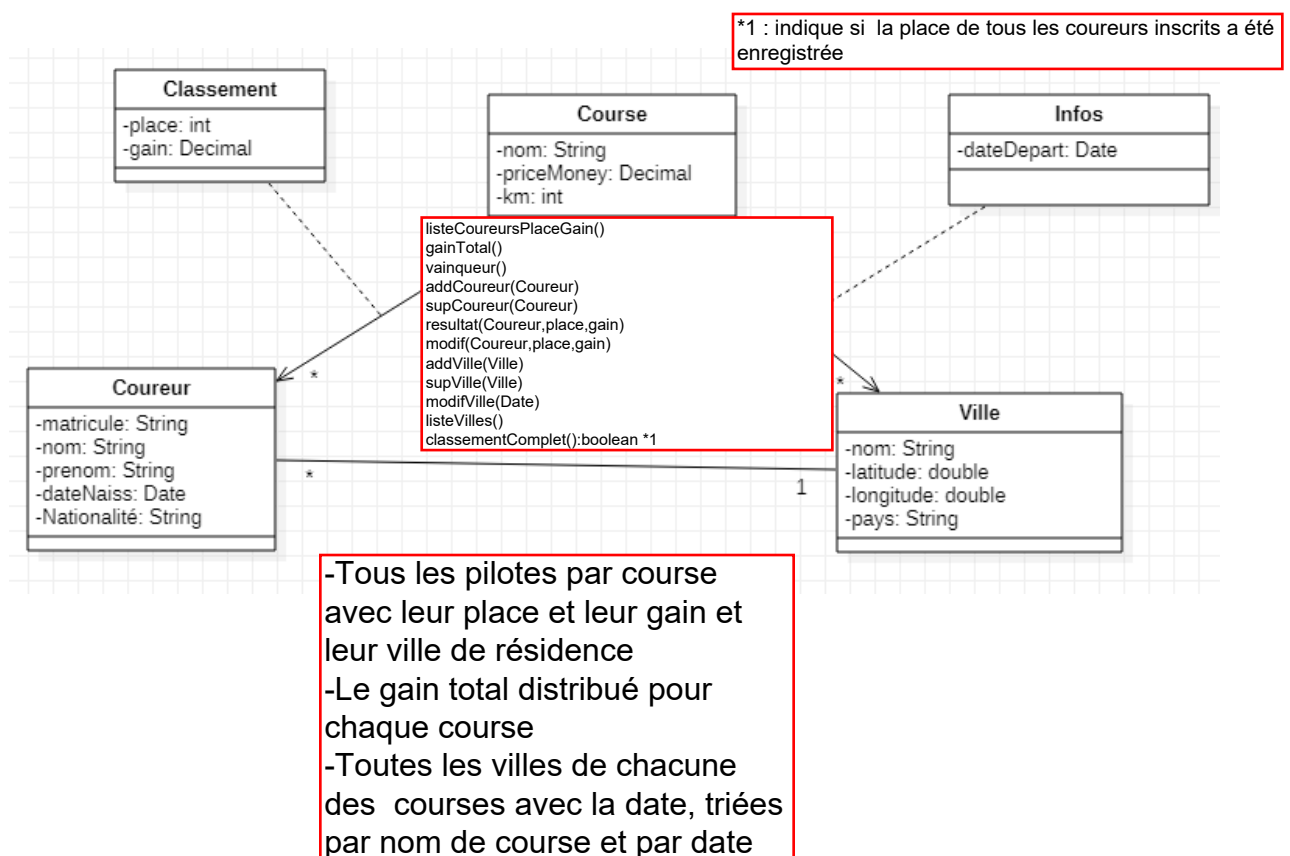
Dans le cadre d'un sport constitué de courses par étapes, on désire répertorier les différentes courses auxquelles sont inscrits les différents coureurs participant au championnat. On désire notamment afficher pour chaque course, la liste de ses participants. Un coureur peut très bien ne pas s'inscrire à l'une des courses et une course peut très bien n'avoir aucun participant.

Pour chacun des coureur on a enregistré leur numéro de licence unique, leur nom, prénom, date de naissance et nationalité, ainsi que sa ville de résidence.

Pour chacune des courses on a enregistré son nom unique, son priceMoney, la date à laquelle elle se déroule, son nombre total de km et la liste des villes dans lesquelles elle se déroule.

Pour chacune des villes, on a enregistré son nom, ses coordonnées géographiques (latitude et longitude) et le pays où elle se situe. Le couple latitude-longitude est unique. Les noms de ville **ne sont pas** uniques.

On enregistre également pour chaque coureur ayant participé à une course son classement final et son gain en argent pour cette course. Si le coureur abandonne, la place vaut -1.



### Design Patterns :

- builder : Pilote : matricule, nom et prénom obligatoires.
- composite : Les courses appartiennent à des championnats (id:int,nom:String) qui peuvent eux-mêmes comporter des championnats. Totaliser le prize money .
- observer : Les pilotes observent les courses en ce qui concerne la modification du prizeMoney de la course.

## Exercice 2\_3

Dans le cadre d'un sport constitué de courses par étapes, on désire répertorier les différentes courses auxquelles sont inscrits les différents coureurs participant au championnat. On désire notamment afficher pour chaque course, la liste de ses participants. Un coureur peut très bien ne pas s'inscrire à l'une des courses et une course peut très bien n'avoir aucun participant.

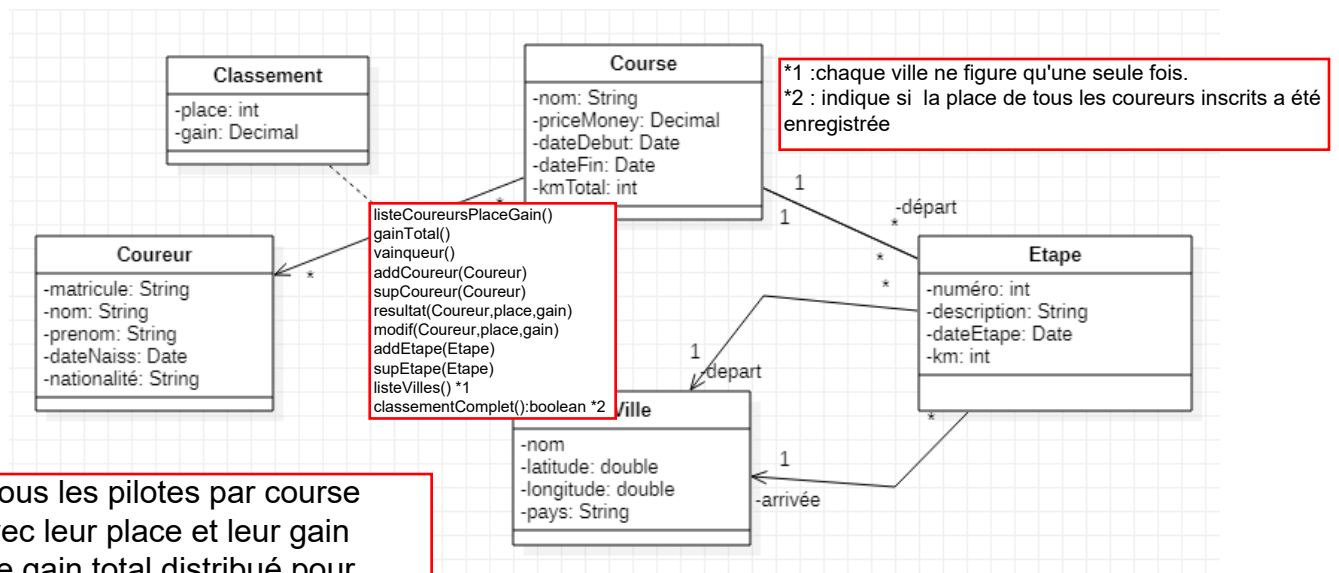
Pour chacun des coureur on a enregistré leur numéro de licence unique, leur nom, prénom, date de naissance et nationalité, ainsi que sa ville de résidence.

Pour chacune des courses on a enregistré son nom unique, son priceMoney, la date à laquelle elle se déroule, son nombre total de km et la liste des étapes dont elle est constituée.

Pour chaque étape, on enregistre son numéro d'ordre dans la course où elle se situe (deux étapes différentes reprises dans la même course ne peuvent pas avoir le même numéro d'ordre), sa description (exemple : « parcours difficile avec beaucoup de côtes »), sa date et son kilométrage ainsi que sa ville de départ et sa ville d'arrivée.

Pour chacune des villes, on a enregistré son nom, ses coordonnées géographiques (latitude et longitude) et le pays où elle se situe. Le couple latitude-longitude est unique. Les noms de ville **ne sont pas** uniques.

On enregistre également pour chaque coureur ayant participé à une course son classement final et son gain en argent pour cette course. Si le coureur abandonne, la place vaut -1.



-Tous les pilotes par course avec leur place et leur gain  
-Le gain total distribué pour chaque course  
-Toutes les étapes de chacune des courses avec la ville de départ et celle d'arrivée, triées par nom de cours et par date

### Design Patterns :

- builder : Pilote : matricule,nom et prénom obligatoires.
- composite : Les courses appartiennent à des championnats (id:int,nom:String) qui peuvent eux-mêmes comporter des championnats. Totaliser le prize money .
- observer : Les pilotes observent les courses en ce qui concerne la modification du prizeMoney de la course.



### Exercice 3\_1

Une entreprise informatique gère un ensemble de projets .

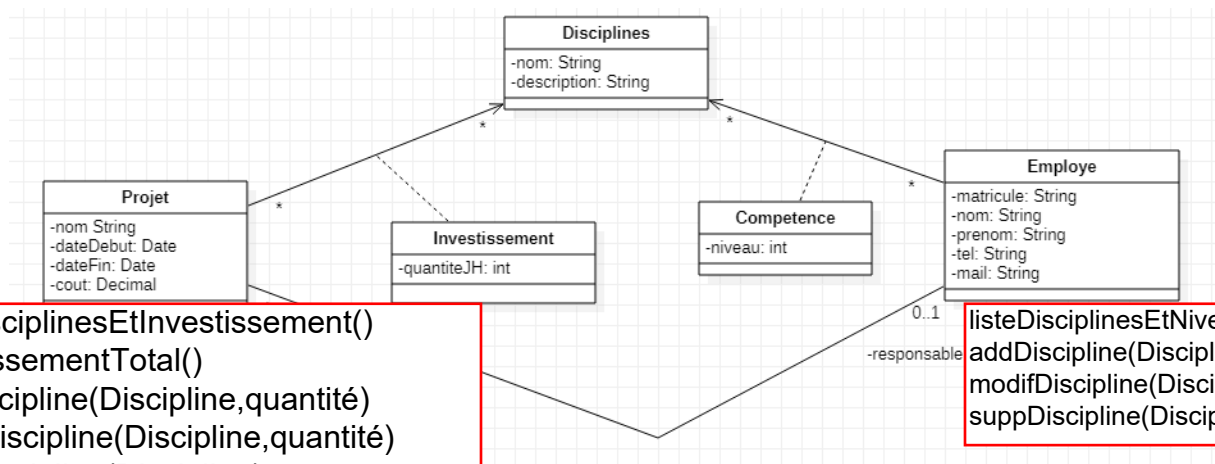
Chaque projet est caractérisé par un titre unique , une date de démarrage ,une date butoir et un coût.

Chaque projet demande un investissement en temps (journée/homme) dans un ensemble de disciplines telles que programmation JAVA, programmation C++,gestion et développement de bases de données, web design etc . (nom de la discipline unique).

Pour chaque membre du personnel de l'entreprise, on a enregistré ses coordonnées (matricule unique, nom ,prénom, gsm unique ,email unique) ainsi que ses compétences dans les disciplines citées ci-dessus.

Pour chaque compétence on a également enregistré le degré de compétence de la personne concernée (1 :peu compétent,2 :moyennement compétent,3 :très compétent) .~~Pour chaque membre du personnel on a également enregistré les différents projets dans lesquels il intervient, la date d'engagement dans le projet ainsi que le pourcentage de son temps qu'il y consacre (exemple : 30% projet1, 20% projet 2, 50% projet 3).~~

Pour chaque projet on a également enregistré un employé responsable, un même employé pouvant être responsable de plusieurs projets.



-toutes les disciplines de chacun des projets avec l'investissement en JH. Indiquer également les infos du responsable.  
-toutes les compétences avec le niveau correspondant de chacun des employés  
-Investissement total de chacun des projets en JH.

\*1 : retourne pour chacune des disciplines faisant l'objet d'un investissement du projet, le niveau de compétence du responsable.

Design Patterns :

-builder : Employé : matricule,nom et prénom obligatoires.  
-composite : Les projets appartiennent à des Groupes de projets (id:int,nom:String) qui peuvent eux-mêmes comporter des groupes de projets. Totaliser le coût total .  
-observer : Les employés observent les projets en ce qui concerne la modification de la date de fin du projet.

### Exercice 3\_2

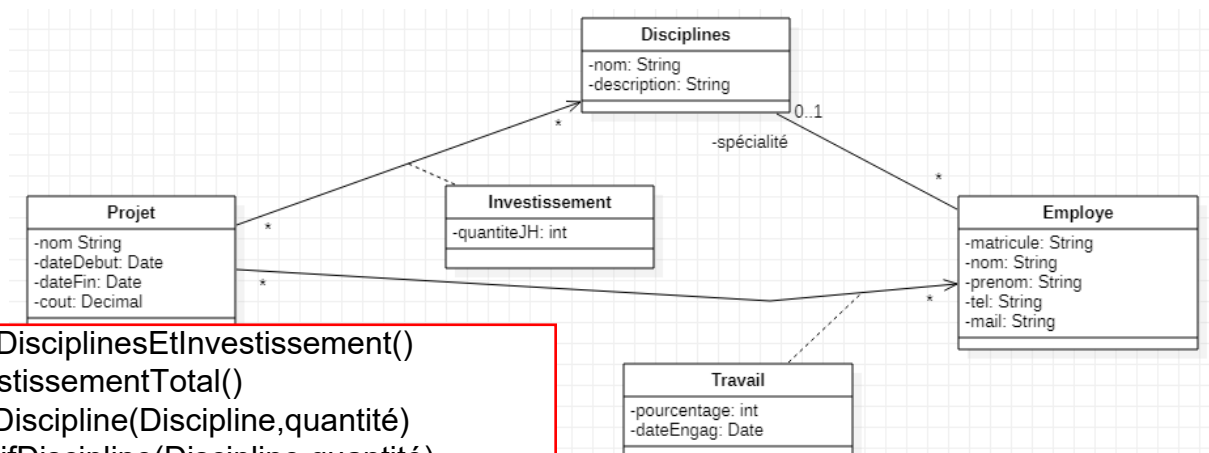
Une entreprise informatique gère un ensemble de projets .

Chaque projet est caractérisé par un titre unique , une date de démarrage ,une date butoir et un coût.

Chaque projet demande un investissement en temps (journée/homme) dans un ensemble de disciplines telles que programmation JAVA, programmation C++,gestion et développement de bases de données, web design etc . (nom de la discipline unique).

Pour chaque membre du personnel de l'entreprise, on a enregistré ses coordonnées (matricule unique, nom ,prénom, gsm unique ,email unique) ainsi qu'une spécialité parmi les disciplines citées ci-dessus.

Pour chaque projet on a également enregistré les différents employés qui y travaillent, la date d'engagement de l'employé concerné dans le projet ainsi que le pourcentage du projet pris en charge par cet employé (exemple : projet 1 : 30% pris en charge par Jean Dupont, projet1 : 20% pris en charge par Eric Lenoir, projet 2 : 50% pris en charge par Jean Dupont, ...).



```
listeDisciplinesEtInvestissement()
investissementTotal()
addDiscipline(Discipline,quantité)
modifDiscipline(Discipline,quantité)
suppDiscipline(Discipline)
listeEmployesEtPourcentageEtDate()
addEmploye(Employe,pourcentage,date)
supEmploye(Employe)
modifEmploye(Employe,pourcentage)
totalPourcentage()
ListeSpécialités() *1
```

-toutes les disciplines de chacun des projets avec l'investissement en JH.  
-Pour chacun des projets tous les employés avec le pourcentage et la date d'engagement correspondant de chacun des employés(indiquer aussi la spécialité des employés).  
-Investissement total de chacun des projets en JH.

\* 1 : Retourne la liste des spécialités des employés travaillant sur le projet, chaque spécialité n'apparaît qu'une seule fois.

#### Design Patterns :

-builder : Employé : matricule,nom et prénom obligatoires.  
-composite : Les projets appartiennent à des Groupes de projets (id:int,nom:String) qui peuvent eux-mêmes comporter des groupes de projets. Totaliser le coût total .  
-observer : Les employés observent les projets en ce qui concerne la modification de la date de fin du projet.

### Exercice 3\_3

Une entreprise informatique gère un ensemble de projets .

Chaque projet est caractérisé par un titre unique , une date de démarrage ,une date butoir et un coût.

Chaque projet demande d'utiliser une discipline de base choisie parmi un ensemble de disciplines telles que programmation JAVA, programmation C++,gestion et développement de bases de données, web design etc . (nom de la discipline unique).

Pour chaque membre du personnel de l'entreprise, on a enregistré ses coordonnées (matricule unique, nom ,prénom, gsm unique ,email unique).

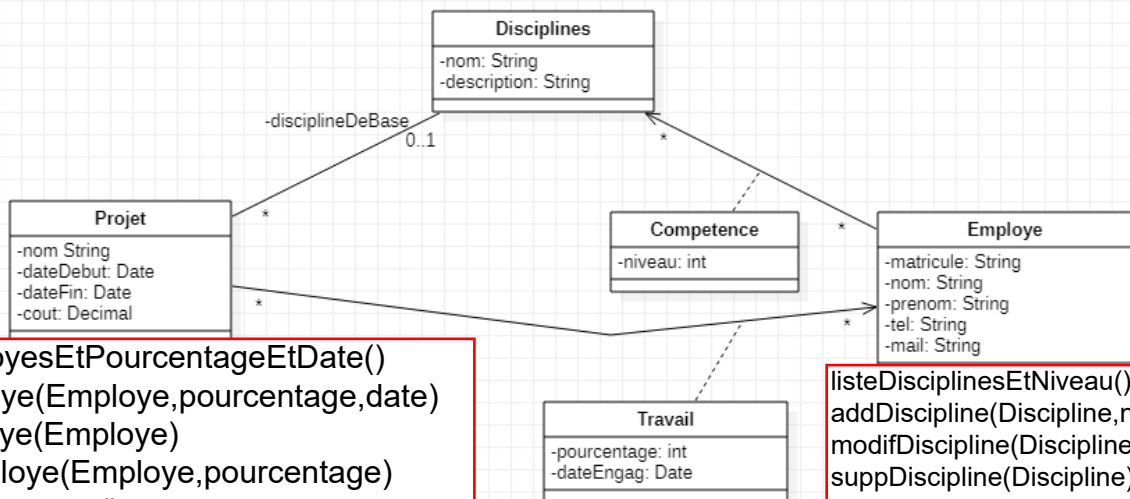
Pour chaque projet on a également enregistré les différents employés qui y travaillent, la date d'engagement de l'employé concerné dans le projet ainsi que le pourcentage du projet pris en charge par cet employé (exemple : projet 1 : 30% pris en charge par Jean Dupont, projet1 : 20% pris en charge par Eric Lenoir, projet 2 : 50% pris en charge par Jean Dupont, ...).

#### Design Patterns :

-builder : Employé : matricule,nom et prénom obligatoires.

-composite : Les projets appartiennent à des Groupes de projets (id:int,nom:String) qui peuvent eux-mêmes comporter des groupes de projets. Totaliser le coût total .

-observer : Les employés observent les projets en ce qui concerne la modification de la date de fin du projet.



listeEmployesEtPourcentageEtDate()  
addEmploye(Employe,pourcentage,date)  
supEmploye(Employe)  
modifEmploye(Employe,pourcentage)  
totalPourcentage()  
listeEmployesDisciplineBase(niveau) \*1

\*1 : retourne la liste des employés travaillant sur le projet et ayant un niveau de compétence égal ou plus grand que celui indiqué en paramètre dans la discipline de base du projet.

listeDisciplinesEtNiveau()  
addDiscipline(Discipline,niveau)  
modifDiscipline(Discipline,niveau)  
suppDiscipline(Discipline)

-tous les employés avec leurs disciplines et le niveau de compétence correspondant.  
-Pour chacun des projets tous les employés avec le pourcentage et la date d'engagement correspondant de chacun des employés. Indiquer aussi la discipline de base de chaque projet.  
-Pourcentage de travail total de chacun des projets.