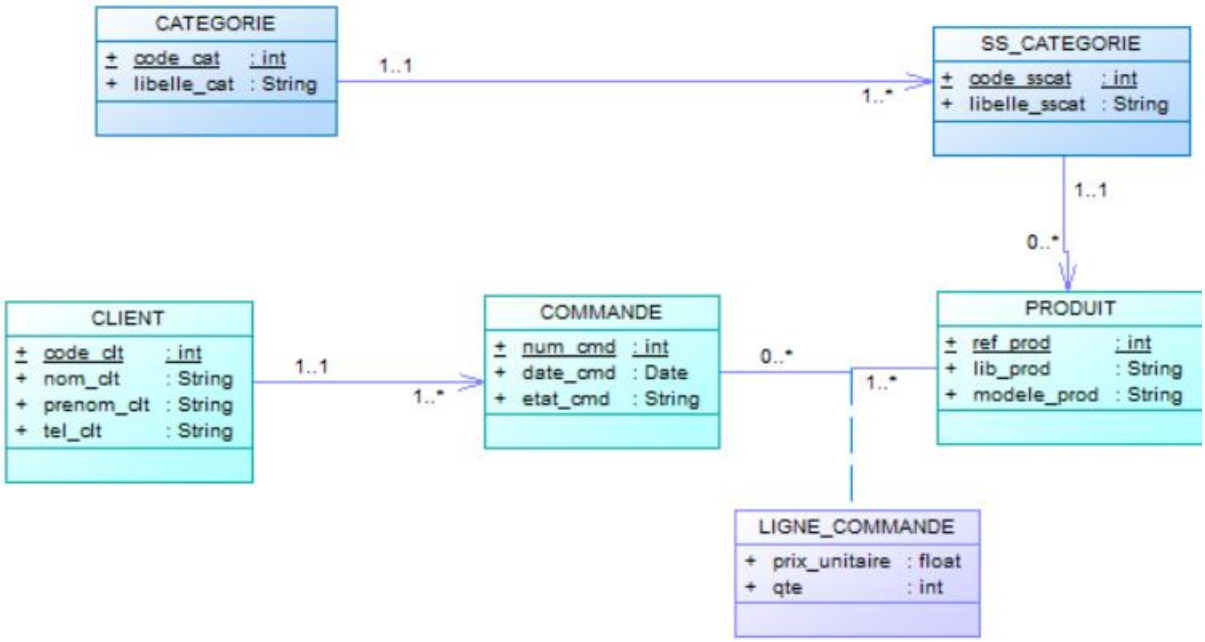
Atelier 03 – Diagramme de classes - Solution

Exercice 1 :

Transformez le modèle logique suivant en diagramme de classes :

CATEGORIE (code_cat, libelle_cat)
SS_CATEGORIE (code_sscat, libelle_sscat, #code_cat)
PRODUIT (ref_prod, lib_prod, modele_prod, #code_sscat)
CLIENT (code_clt, nom_clt, prenom_clt, tel_clt)
COMMANDE (num_cmd, date_cmd, etat_cmd, #code_clt)
LIGNE_COMMANDE (#ref_prod, #num_cmd, prix_unitaire, qte)

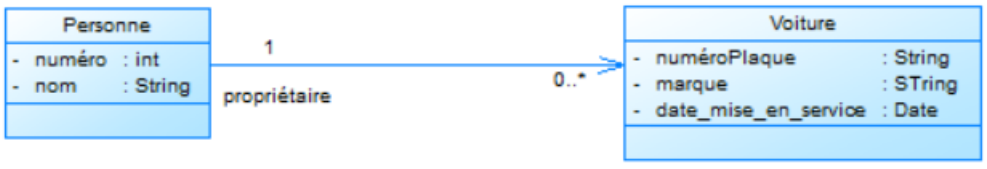


Exercice 2 :

Soient un ensemble de personnes et un ensemble de voitures. Une personne est caractérisée par un numéro qui l'identifie, son nom et par les voitures dont elle est l'unique propriétaire. Une voiture est caractérisée par un numéro de plaque, une marque et une date de mise en circulation.

Une personne peut avoir plusieurs voitures tandis qu'une voiture est la propriétére d'une seule personne.

Travail à faire : Réaliser un diagramme de classe.



Code Javascript

```
class Voiture {
    constructor(numeroPlaque, marque, dateMiseEnService, proprietaire = null)
{
        this.numeroPlaque = numeroPlaque;
        this.marque = marque;
        this.dateMiseEnService = dateMiseEnService;
        this.proprietaire = proprietaire;  // Propriétaire de la voiture
    }

    // Getter pour le propriétaire
    getProprietaire() {
        return this.proprietaire;
    }

    // Setter pour le propriétaire
    setProprietaire(proprietaire) {
        this.proprietaire = proprietaire;
    }
}


class Personne {
    constructor(numero, nom) {
        this.numero = numero;
        this.nom = nom;
        this.voitures = new Set();
        // Utilisation de Set pour représenter une collection sans doublons
    }

    // Getter pour la collection de voitures
    getVoitures() {
        return this.voitures;
    }

    // Setter pour la collection de voitures
    setVoitures(newVoitures) {
        this.voitures.clear(); // Supprimer toutes les voitures existantes
        // Ajouter les nouvelles voitures
        newVoitures.forEach(voiture => this.addVoiture(voiture));
    }

    // Ajouter une voiture à la collection
    addVoiture(newVoiture) {
        if (newVoiture) {
            // Ajouter à l'ensemble si la voiture n'existe pas déjà
            this.voitures.add(newVoiture);
        }
```

```
    }

    // Supprimer une voiture de la collection
    removeVoiture(oldVoiture) {
        if (oldVoiture) {
            this.voitures.delete(oldVoiture); // Supprimer de l'ensemble
        }
    }

    // Supprimer toutes les voitures
    removeAllVoitures() {
        this.voitures.clear();
    }
}
```

Code PHP

```
<?php
class Voiture {
    private $numeroPlaque;
    private $marque;
    private $dateMiseEnService;
    // Propriétaire de la voiture (objet de type Personne)
    private $proprietaire;

    public function __construct($numeroPlaque, $marque, $dateMiseEnService,
$proprietaire = null) {
        $this->numeroPlaque = $numeroPlaque;
        $this->marque = $marque;
        $this->dateMiseEnService = $dateMiseEnService;
        $this->proprietaire = $proprietaire;
    }

    // Getter pour le propriétaire
    public function getProprietaire() {
        return $this->proprietaire;
    }

    // Setter pour le propriétaire
    public function setProprietaire($proprietaire) {
        $this->proprietaire = $proprietaire;
    }
}

class Personne {
    private $numero;
    private $nom;
    private $voitures;
```

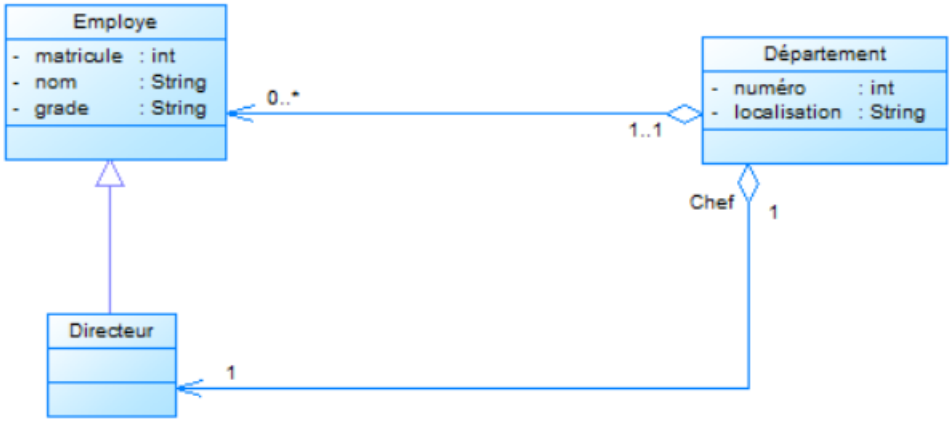
```php
    // Constructeur pour initialiser les propriétés
    public function __construct($numero, $nom) {
        $this->numero = $numero;
        $this->nom = $nom;
        // Initialiser un tableau pour la collection de voitures
        $this->voitures = array();
    }

    // Getter pour la collection de voitures
    public function getVoitures() {
        return $this->voitures;
    }

    // Setter pour la collection de voitures
    public function setVoitures($newVoitures) {
        $this->voitures = array(); // Réinitialiser les voitures
        foreach ($newVoitures as $voiture) {
            $this->addVoiture($voiture); // Ajouter les nouvelles voitures
        }
    }

    // Ajouter une voiture à la collection
    public function addVoiture($newVoiture) {
        if (!in_array($newVoiture, $this->voitures)) {
            // Ajouter la voiture si elle n'existe pas encore
            $this->voitures[] = $newVoiture;
        }
    }

    // Supprimer une voiture de la collection
    public function removeVoiture($oldVoiture) {
        $index = array_search($oldVoiture, $this->voitures);
        if ($index !== false) {
            // Supprimer la voiture si elle existe
            unset($this->voitures[$index]);
        }
    }

    // Supprimer toutes les voitures
    public function removeAllVoitures() {
        $this->voitures = array(); // Réinitialiser la collection
    }
}

?>
```

Exercice 3 :

On souhaite gérer les employés d'une entreprise. Dans cette entreprise il existe plusieurs départements.

Chaque département est occupé par au moins un employé. Un employé est décrit par son numéro matricule (unique dans l'entreprise), son nom, son grade et le département dans lequel il travaille. Un département est décrit par son numéro dans l'entreprise et sa localisation. Un département est dirigé par un directeur qui doit être un de ses employés.

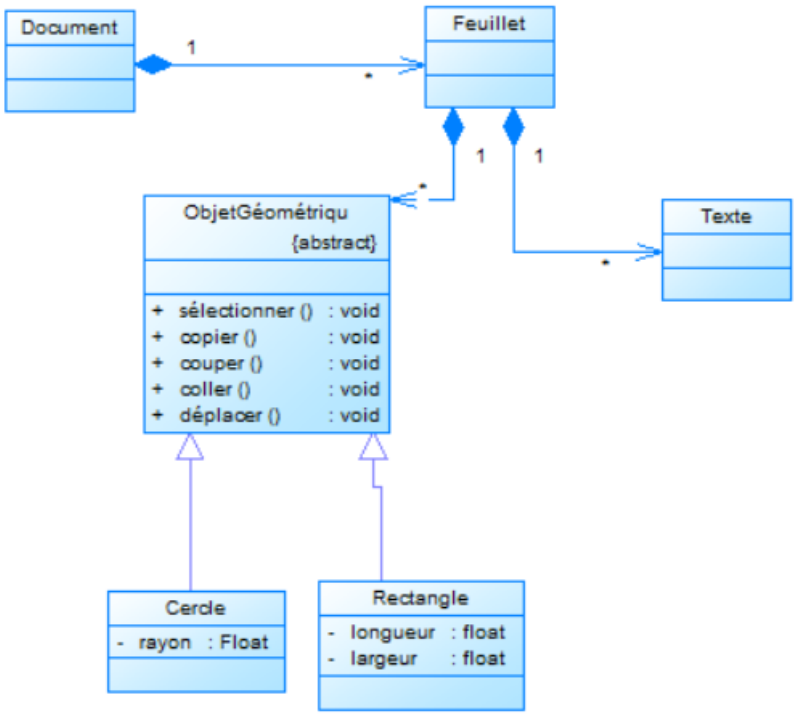
Travail à faire : Réaliser un diagramme de classe.



Exercice 4

Soit un document composé d'un ou plusieurs feuillets. Le feuillet comporte des objets géométriques et des textes. Les objets géométriques supportent des opérations de type : sélectionner, copier, couper, coller et déplacer. On suppose les deux objets géométriques suivants : cercle et rectangle.

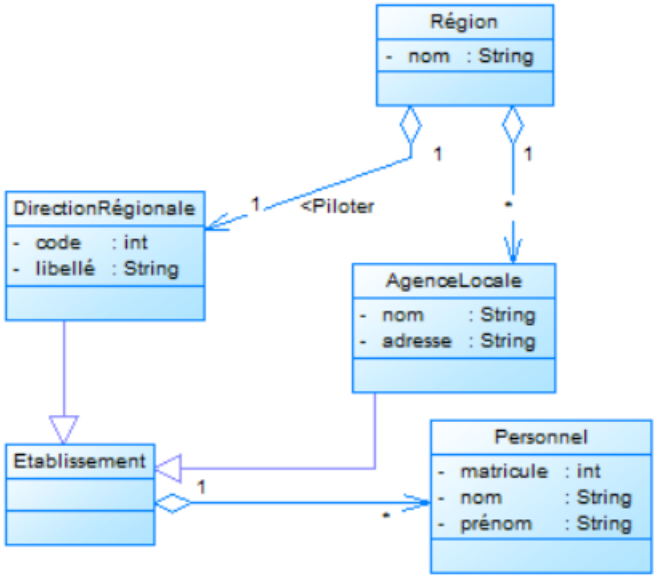
Travail à faire : Réaliser un diagramme de classe.



Exercice 5

Soit un système d'information qui concerne le suivi des personnels d'un ensemble d'agences locales. Chaque agence se trouve dans une région, chaque région est pilotée par une direction régionale. La direction régionale se charge d'un ensemble d'agences locales. une direction régionale est caractérisée par un code et un libellé.

Travail à faire : Réaliser un diagramme de classe.



Exercice 6.

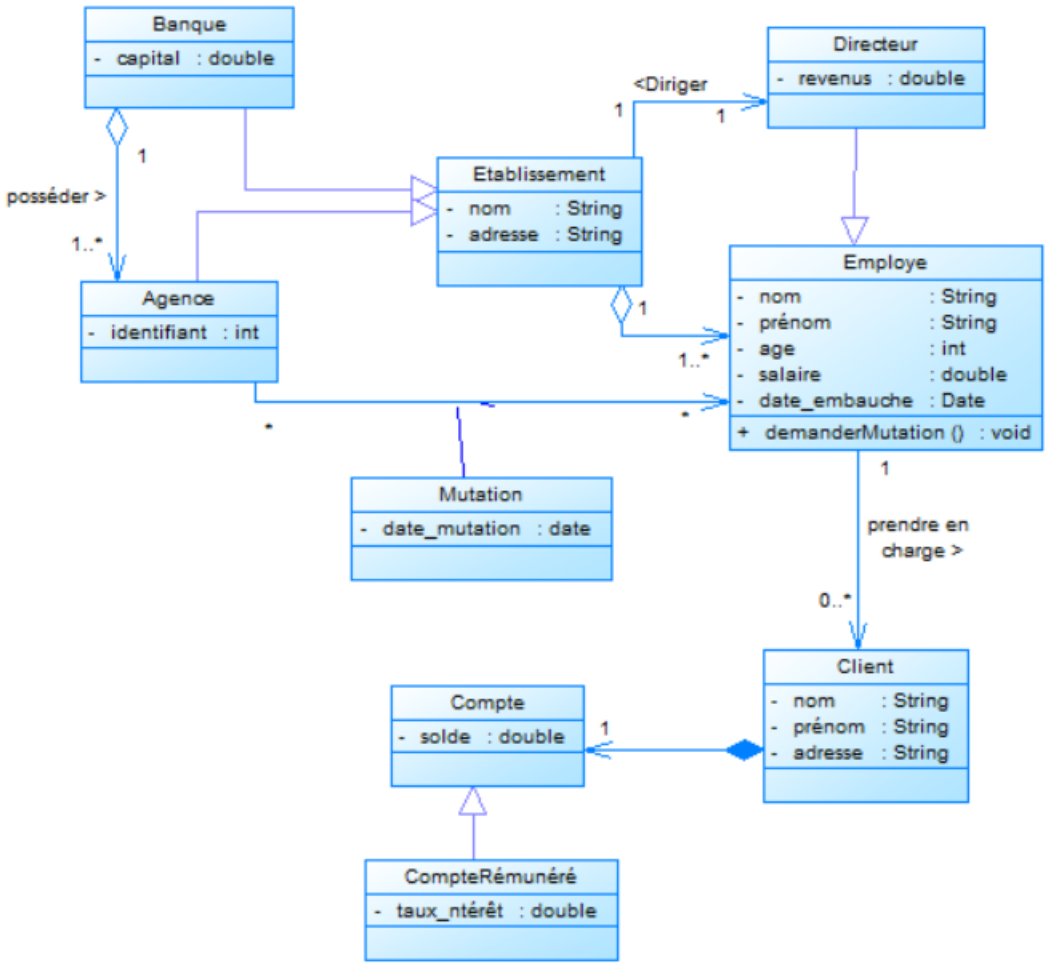
Une banque contient plusieurs agences, elle est caractérisée par son nom (banque populaire) et dirigée par un directeur, possède un capital et une adresse. Le directeur de la banque possède un nom, prénom, âge et son salaire (ou revenue).

Chaque agence possède une adresse, des employés et un identifiant. Chaque employé possède aussi un nom, prénom, age et date d'embauche.

Un employé travail seulement dans une seule agence, mais peut demander une mutation vers une autre agence. Un employé peut gérer un ou plusieurs clients, ce dernier possède un seul compte dans une agence donnée, un nom, prénom et une adresse. Chaque client inscrit est attribué à un employé dans l'agence.

Un compte peut soit être un compte rémunéré ou non. Les comptes rémunérés possèdent un taux d'intérêt versés annuellement

Travail à faire : Réaliser un diagramme de classe.

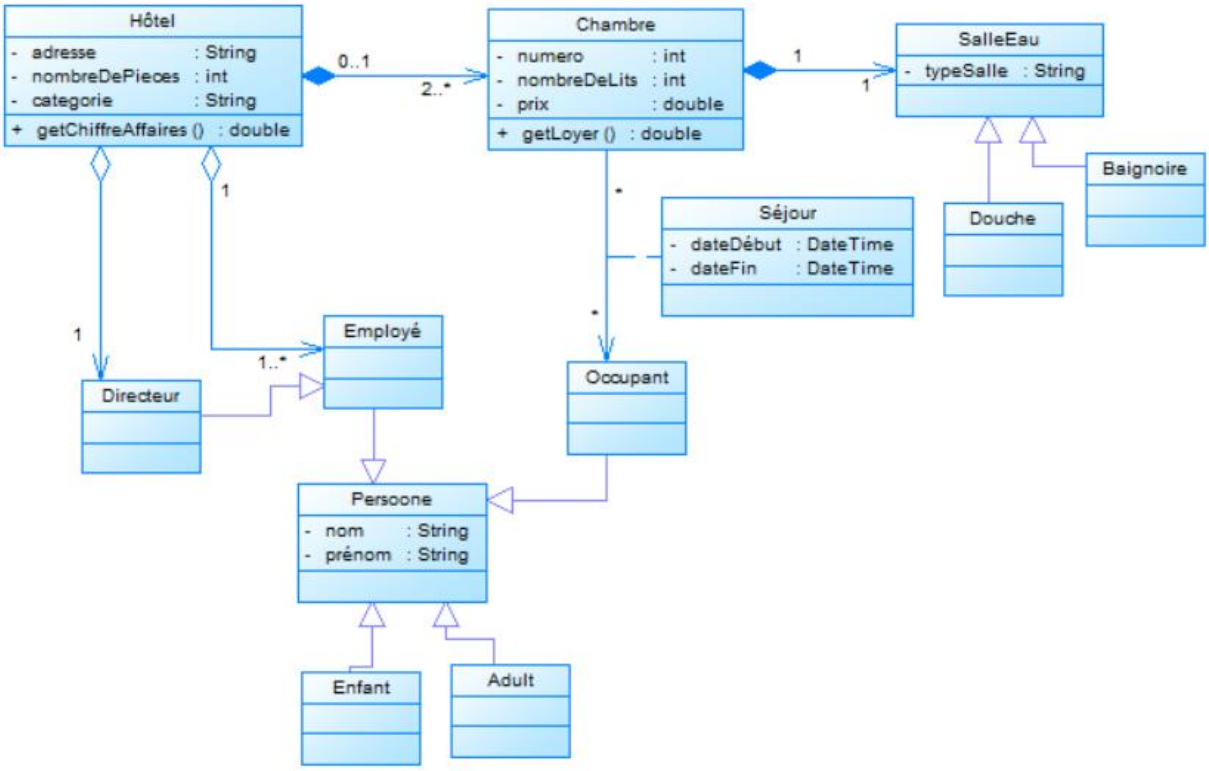


Exercice 7.

Un hôtel est composé d'au moins deux chambres. Chaque chambre dispose d'une salle d'eau : douche ou bien baignoire.

Un hôtel héberge des personnes. Il peut employer du personnel et il est impérativement dirigé par un directeur. On ne connaît que le nom et le prénom des employés, des directeurs et des occupants. Certaines personnes sont des enfants et d'autres des adultes (faire travailler des enfants est interdit).

Un hôtel a les caractéristiques suivantes : une adresse, un nombre de pièces et une catégorie. Une chambre est caractérisée par le nombre et de lits qu'elle contient, son prix et son numéro. On veut pouvoir savoir qui occupe quelle chambre à quelle date. Pour chaque jour de l'année, on veut pouvoir calculer le loyer de chaque chambre en fonction de son prix et de son occupation (le loyer est nul si la chambre est inoccupée). La somme de ces loyers permet de calculer le chiffre d'affaires de l'hôtel entre deux dates.



Exercice 8.

Donnez un diagramme de classes correspondant au code source suivant :

```
public interface Dessinable
{
        public void dessiner ( );
        public void effacer ( );
}
```

```
abstract public class Figure implements Dessinable
{
        protected String couleur;
        protected String getCouleur ( ) { return couleur; }
        protected void setCouleur ( String c ) { couleur = c; }
}
```

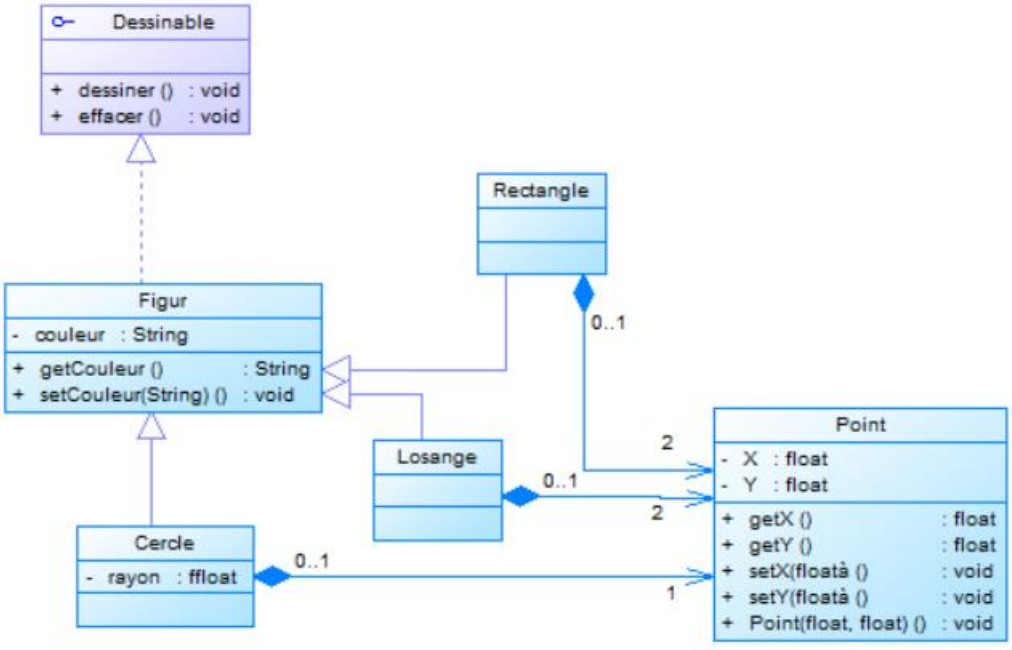
```
public class Point
{
        private float x;
        private float y;
        public float getX ( ) { return x; } public float getY ( ) { return y; }
        public void Point ( float x, float y) { ... }
}
```

```
public class Cercle extends Figure
{
```

```
        private float rayon;
        private Point centre;
        public Cercle ( Point centre, float rayon) { ... }
        public void dessiner ( ) { ... } public void effacer ( ) { ... }
}
```

```
public class Rectangle extends Figure
{
        protected Point sommets[] = new Point[2];
        public Rectangle ( Point p1, Point p2) { ... }
        public void dessiner ( ) { ... }
        public void effacer ( ) { ... }
}
```

```
public class Losange extends Figure
{
        protected Point sommets[] = new Point[2];
        public Losange ( Point p1, Point p2) { ... }
        public void dessiner ( ) { ... }
        public void effacer ( ) { ... }
}
```



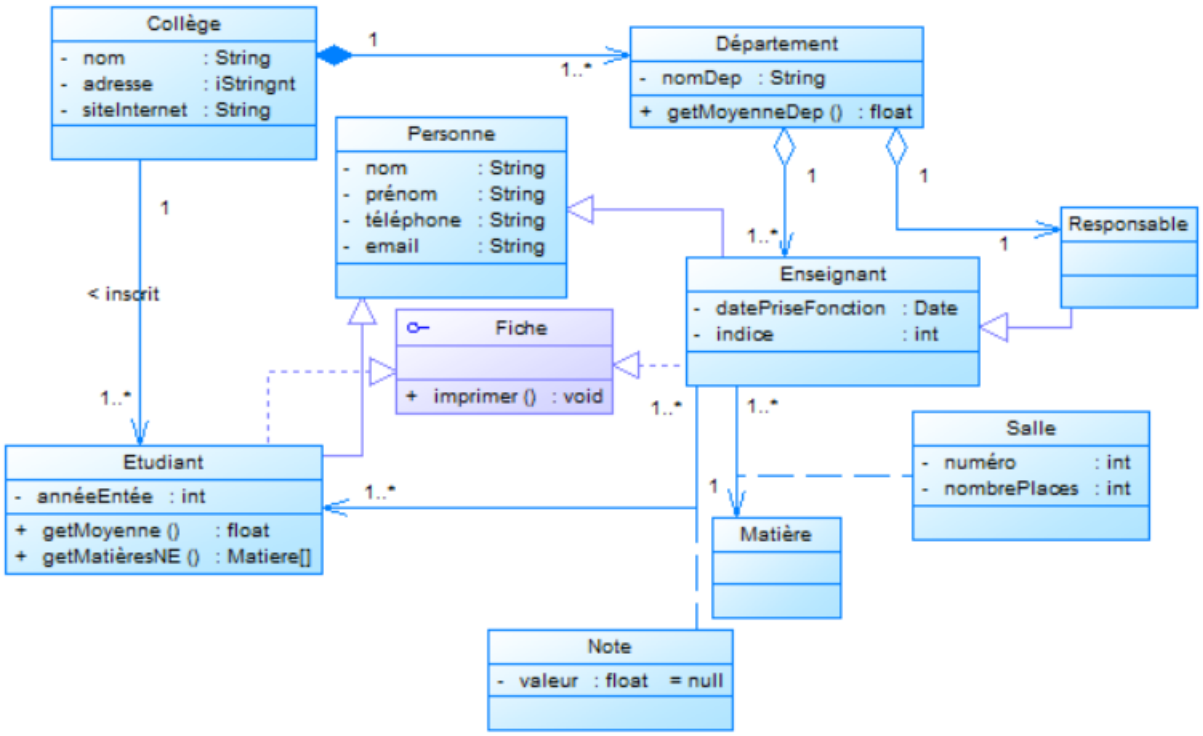
Exercice 9.

Une académie souhaite gérer les cours dispensés dans plusieurs collèges. Pour cela, on dispose des renseignements suivants :

1) Chaque collège possède un site Internet.
2) Chaque collège est structuré en départements, qui regroupent chacun des enseignants spécifiques. Parmi ces enseignants, l'un d'eux est responsable du département.
3) Un enseignant se définit par son nom, prénom, tél, mail, date de prise de fonction et son indice.

4) Chaque enseignant ne dispense qu'une seule matière.
5) Les étudiants suivent quant à eux plusieurs matières et reçoivent une note pour chacune d'elle.
6) Pour chaque étudiant, on veut gérer son nom, prénom, tél, mail, ainsi que son année d'entrée au collège.
7) Une matière peut être enseignée par plusieurs enseignants mais a toujours lieu dans la même salle de cours (chacune ayant un nombre de places déterminé).
8) On désire pouvoir calculer la moyenne par matière ainsi que par département
9) On veut également calculer la moyenne générale d'un élève et pouvoir afficher les matières dans lesquelles il n'a pas été noté.
10) Enfin, on doit pouvoir imprimer la fiche signalétique (, prénom, tél, mail) d'un enseignant ou d'un élève.

Travail à faire : Proposer un diagramme de classes.



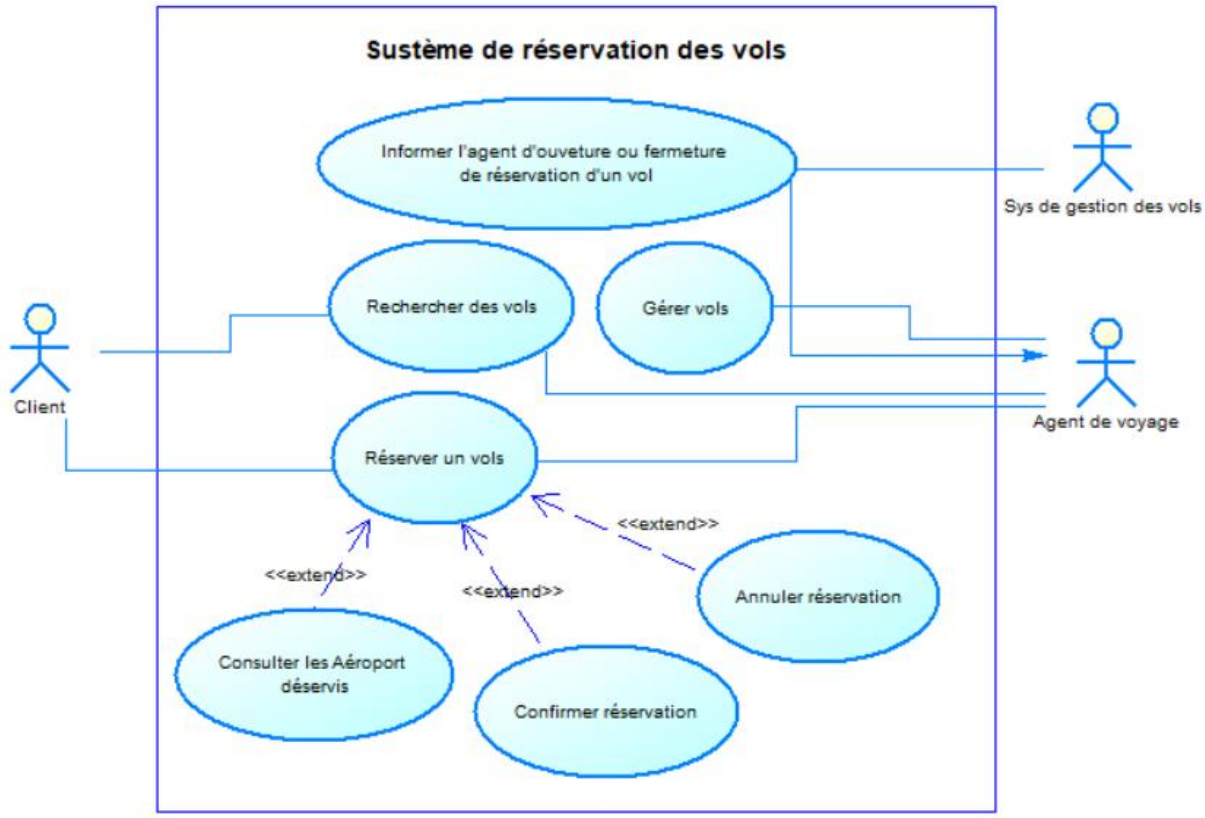
Exercice 10.

On souhaite gérer les réservations de vols effectués dans une agence. D'après les interviews réalisées avec les membres de l'agence, on sait que : •

1) Les compagnies aériennes proposent différents vols.
2) Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.
3) Un client peut réserver un ou plusieurs vols, pour des passagers différents.
4) Une réservation concerne un seul vol et un seul passager.
5) Une réservation peut être confirmée ou annulée.
6) Un vol a un aéroport de départ et un aéroport d'arrivée.
7) Un vol a un jour et une heure de départ, et un jour et une heure d'arrivée.
8) Un vol peut comporter des escales dans un ou plusieurs aéroport(s).
9) Une escale à une heure de départ et une heure d'arrivée.
10) Chaque aéroport dessert une ou plusieurs villes.

Travail à faire :

Proposer le diagramme des cas d'utilisation



Proposer un diagramme de classes.