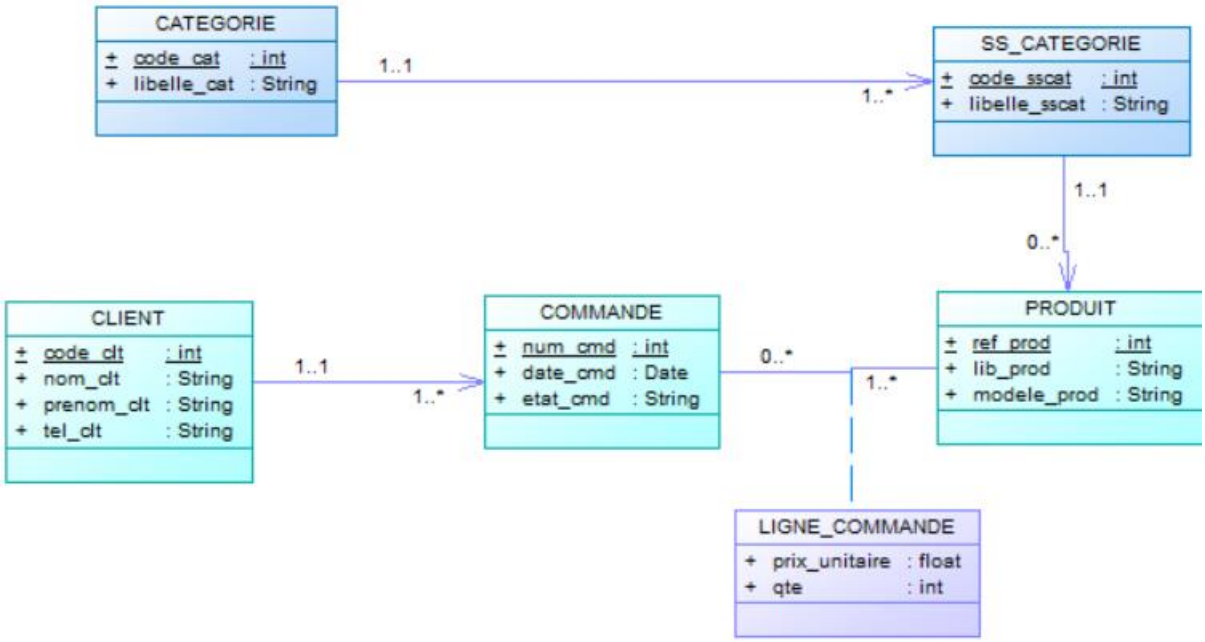
Atelier 03 – Diagramme de classes - Solution

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)

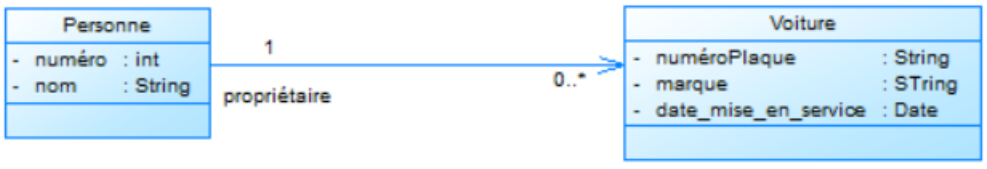


Exercice 2 :

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();
    }
}

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;
    }
}
```

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;
```

```php
        $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;

    // 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;
        }
    }
```

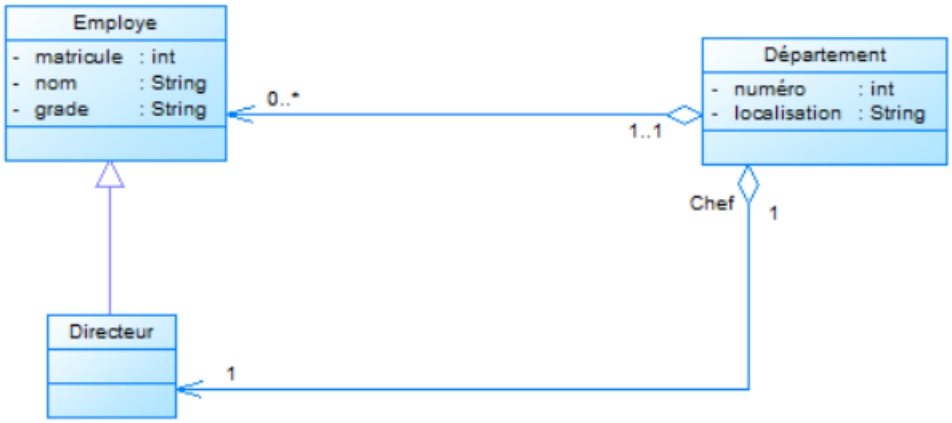
```php
    // Supprimer une voiture de la collection
    public function removeVoiture($oldVoiture) {
        $index = array_search($oldVoiture, $this->voitures);
        if ($index !== false) {
            // Supprimer la voiture si elle existe
            unset($this->voitures[$index]);
        }
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

    // 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.

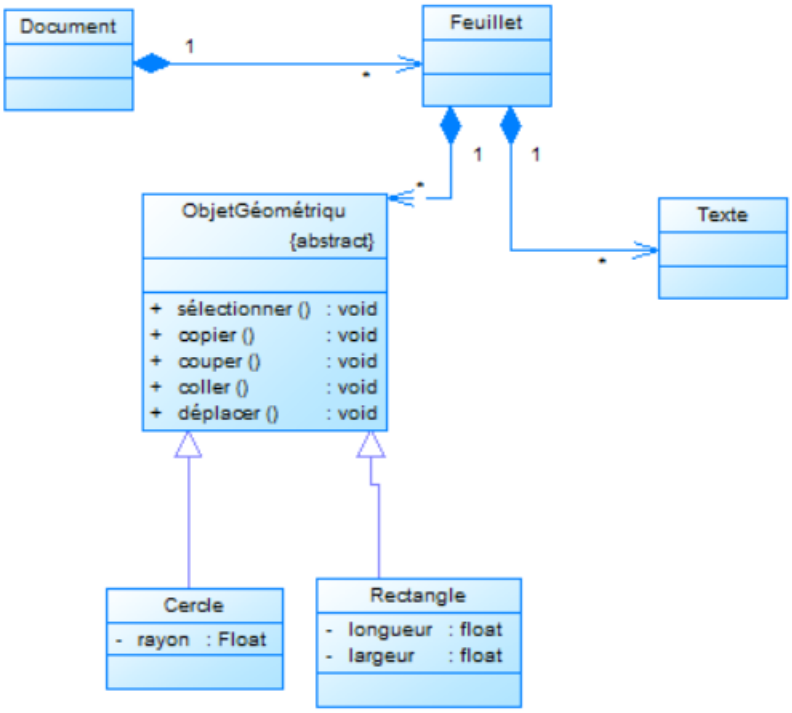
Travail à faire : Réaliser un diagramme de classe.



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.