**Partie 2 : Programmer les classes**

Dans cette partie de l’évaluation certificative finale, vous devez implémenter les classes en code python.

Créer un projet nommé **mesClasses.**

Respecter les normes de programmation et documenter votre code.

**Classe Medicament**

* Ajouter au projet **mesClasses** la classe **Medicament.**

Ajouter à cette classe les attributs d’instance:

* **code\_medicament** qui doit être composé de trois lettres suivies de 3 chiffres. Exemple : ABC123
* **nom\_chimique** qui est composé d’un maximum de 50 lettres.
* **Nom\_commercial** qui est composé d’un maximum de 50 lettres.
* **prix** qui a une valeur entre 5 et 100$.
* **categorie** qui est : Antibiotiques, Analgésiques ou Corticoïde.
* Ajouter les méthodes d’accès et propriétés pour les attributs avec validation.
* Ajouter le constructeur avec paramètres et valeurs par défaut.
* Ajouter la méthode magique \_\_str\_\_ ().
* Ajouter la méthode \_\_add\_\_() qui permet d’ajouter le prix de deux médicaments.

# **Classe Fournisseur**

* Créer une classe **Fournisseur** dans votre projet **mesClasses**.
* Ajouter les attributs d’instance suivants :

1. **code\_fournisseur** : doit commencer par la lettre F en majuscule suivie par cinq caractères alphanumériques. Exemple : F-41524.
2. **Nom\_compagnie** : maximum de 30 caractères alphanumériques.
3. L’attribut d’instance **ls\_patients** de type list [**Patient**] qui contient la liste des patients qui achètent les médicaments du fournisseur via la pharmacie.

* Ajouter l’attribut de classe **nb\_fournisseurs** qui permet de compter le nombre de fournisseurs instanciés.
* Ajouter l’attribut de classe **ls\_fournisseurs** qui contient la liste des objets de type **Fournisseur** instanciés.
* Ajouter le constructeur avec paramètres et valeurs par défaut.
* Ajouter les méthodes d’accès et propriétés pour les attributs d’instance qui le nécessitent.
* Ajouter la méthode d’instance **serialiserFournisseur ()** qui permet de sérialiser un objet de type **Fournisseur** vers un fichier de type **json**. Elle reçoit le nom du fichier en paramètre.
* Ajouter la méthode d’instance **deserialiserFournisseur ()** qui permet de désérialiser un fichier en format **json** qu’ellereçoit en paramètreenun objet de type **Fournisseur**. Elle reçoit le nom du fichier en paramètre.
* Ajouter la méthode d’instance **ajouterMedicamentPatient ()** qui permet de chercher un patient dans la liste des patients **ls\_patients** afin de lui ajouter un médicament à la liste de ses médicaments. Cette méthode prend en paramètre un **numero\_patient** et un objet de type **Medicament.**
* Ajouter la méthode de classe **afficherLstFournisseurs** qui permet d’afficher la liste des fournisseurs de façon professionnelle.
* Ajouter la méthode magique \_\_str\_\_ ().

**Classe Antibiotique**

* Ajouter au projet **mesClasses** la classe **Antibiotique** qui hérite de la classe **Medicament.**
* Ajouter à cette classe l’attribut d’instance **duree\_prise\_max** qui est en nombre de jours.
* Ajouter le constructeur avec paramètres et valeurs par défaut.
* Ajouter les méthodes d’accès et propriétés si nécessaire.
* Ajouter la méthode magique \_\_str\_\_ ().

**Classe Analgesique**

* Ajouter au projet la classe **Analgesique.**
* Ajouter à cette classe l’attribut d’instance **dose\_quot\_max** qui doit être en nombre de comprimés. **Dose\_quot\_max** est la dose quotidienne maximale du médicament.
* Ajouter le constructeur avec paramètres.
* Ajouter les méthodes d’accès et propriétés si nécessaire.
* Ajouter la méthode magique \_\_str\_\_().

**Classe Corticoide**

* Ajouter au projet **mesClasses** la classe **Corticoide.**
* Ajouter à cette classe l’attribut d’instance **effet\_medic** qui décrit l’effet du médicament. Cet attribut peut être court, intermédiaire ou prolongé.
* Ajouter le constructeur avec paramètres.
* Ajouter les méthodes d’accès et propriétés si nécessaire.
* Ajouter la méthode magique \_\_str\_\_().

**Classe Patient**

* Ajouter la classe **Patient** au projet.
* Ajouter les attributs d’instance suivants à la classe **Patient** :

1. Le **numero\_patient** composé de 7 chiffres dont le premier est pair. Exemple : 2123456
2. Le **nom** du patient qui a une longueur de 50 caractères maximum.
3. Le **prenom** du patient qui a une longueur de 50 caractères maximum.
4. Le courriel est composé du nom suivi du prénom suivi de « @gmail.com ». Exemple : HociniHasna @gmail.com
5. La **date\_naiss** du patient qui est de type **Date**.
6. La **list\_medic** qui est de type list [**Medicament**] et qui contient la liste des médicaments achetés par le patient.

* Ajouter les méthodes d’accès et propriétés pour les attributs avec validation
* Ajouter un constructeur avec paramètres et valeurs par défaut
* Ajouter la méthode d’instance privée nommée **calculerAge**() qui permet de calculer l’âge du patient.
* Ajouter la méthode d’instance publique **estAdule** () qui utilise la méthode privée **calculerAge** () pour retourner l’une des deux valeurs True ou False selon que le patient ait dépassé 18 ans ou pas.
* Ajouter la méthode d’instance publique **afficherMedicaments ()** qui affiche de façon professionnelle les attributs de la liste des médicaments achetés par le patient.
* Ajouter la méthode d’instance publique **ajouterMedicament ()** qui prend en paramètre un objet de type **Medicament** et l’ajoute à la liste des médicaments achetés par le patient.
* Ajouter la méthode d’instance **supprimerMedicament()** qui prend en paramètre un objet de type Medicament et le supprime de la liste des médicaments achetés par le patient.

Bonne persévérance ! 😊