

Test TP (1)

Exercice : L'objectif de cet exercice est de créer un programme en Python utilisant la programmation orientée objet (POO) afin de gérer les informations d'un étudiant, de calculer la moyenne de ses notes et évaluer son résultat académique.

I. Créer une classe nommée **Etudiant** contenant :

Attributs privés :

1. **nom** : nom de l'étudiant
2. **prenom** : prénom de l'étudiant
3. **age** : âge de l'étudiant
4. **numero** : numéro d'inscription
5. **n** : nombre de notes
6. **notes** : liste contenant les notes de l'étudiant

Méthodes publiques :

1. **initialiser()** : initialise tous les attributs (**nom=""**, **prenom=""**, **age=0**, **numero=0**, **n=0**, **notes=[]**)
2. **infos()** : saisit les informations personnelles (**nom**, **prenom**, **age**, **numero**)
3. **notes()** : saisit le nombre de notes et remplit la liste **notes**
4. **calmoy()** : calcule et retourne la moyenne des notes
5. **mention()** : retourne "Admis" si moyenne ≥ 10 , "Ajourné" sinon
6. **afficher()** : affiche toutes les informations de l'étudiant, la moyenne et la mention

II. Programme principal

1. Créer un objet **Etudiant**
2. Initialiser l'objet
3. Saisir les informations et les notes
4. Afficher toutes les informations, la moyenne et la mention

Test TP (2)

Exercice : L'objectif de cet exercice est de créer un programme en Python utilisant la programmation orientée objet afin de gérer les informations d'un patient, calculer la moyenne de ses frais médicaux et évaluer sa situation financière.

I. Créer une classe nommée **Patient** contenant:

Attributs privés :

1. **nom** : nom du patient
2. **prenom** : prénom du patient
3. **age** : âge du patient
4. **id** : identifiant du patient
5. **n** : nombre de factures
6. **frais** : liste contenant les frais médicaux du patient

Méthodes publiques :

1. **initialiser()** : initialise tous les attributs (**nom=""**, **prenom=""**, **age=0**, **id=0**, **n=0**, **frais=[]**)
2. **sinfos()** : saisit les informations personnelles (**nom**, **prenom**, **age**, **id**)
3. **sfrais()** : saisit le nombre de factures et remplit la liste **frais**
4. **calmoy()** : calcule et retourne la moyenne des frais
5. **eval()** : retourne "**stable**" si la moyenne ≤ 2000 , "**difficile**" sinon
6. **afficher()** : affiche toutes les informations du patient, la moyenne des frais et l'évaluation

II. Programme principal

1. Créer un objet **Patient**
2. Initialiser l'objet
3. Saisir les informations et les frais médicaux
4. Afficher toutes les informations, la moyenne et l'évaluation

TP test (1)

```
class Etudiant:
    # Méthode d'initialisation
    def initialiser(self):
        self.__nom = ""
        self.__prenom = ""
        self.__age = 0
        self.__numero = 0
        self.__n = 0
        self.__notes = []

    # Saisie des informations personnelles
    def sinfos(self):
        self.__nom = input("Entrer le nom : ")
        self.__prenom = input("Entrer le prénom : ")
        self.__age = int(input("Entrer l'âge : "))
        self.__numero = int(input("Entrer le numéro d'inscription : "))

    # Saisie des notes
    def snotes(self):
        self.__n = int(input("Entrer le nombre de notes : "))
        self.__notes = []

        for i in range(self.__n):
            note = float(input(f"Entrer la note {i + 1} : "))
            self.__notes.append(note)

    # Calcul de la moyenne
    def calmoy(self):
        return sum(self.__notes) / self.__n

    # Affichage de la mention
    def mention(self):
        if self.calmoy() >= 10:
            return "Admis"
        else:
            return "Ajourné"

    # Affichage des informations complètes
    def afficher(self):
        print("\n--- Informations de l'étudiant ---")
        print("Nom :", self.__nom)
        print("Prénom :", self.__prenom)
        print("Âge :", self.__age)
        print("Numéro d'inscription :", self.__numero)
        print("Moyenne :", self.calmoy())
        print("Mention :", self.mention())

# Programme principal
# Création de l'objet
etu = Etudiant()
# Initialisation
etu.initialiser()
# Saisie des informations
etu.sinfos()
etu.snotes()
# Affichage des résultats
etu.afficher()
```

TP test (2)

```
class Patient:
    # Méthode d'initialisation
    def initialiser(self):
        self.__nom = ""
        self.__prenom = ""
        self.__age = 0
        self.__id = 0
        self.__n = 0
        self.__frais = []

    # Saisie des informations personnelles
    def sinfos(self):
        self.__nom = input("Entrer le nom du patient : ")
        self.__prenom = input("Entrer le prénom du patient : ")
        self.__age = int(input("Entrer l'âge du patient : "))
        self.__id = int(input("Entrer l'identifiant du patient : "))

    # Saisie des frais médicaux
    def sfrais(self):
        self.__n = int(input("Entrer le nombre de factures : "))
        self.__frais = []

        for i in range(self.__n):
            montant = float(input(f"Entrer les frais de la facture {i + 1} : "))
            self.__frais.append(montant)

    # Calcul de la moyenne des frais
    def calmoy(self):
        return sum(self.__frais) / self.__n

    # Évaluation financière
    def eval(self):
        if self.calmoy() <= 2000:
            return "stable"
        else:
            return "difficile"

    # Affichage des informations complètes
    def afficher(self):
        print("\n--- Informations du patient ---")
        print("Nom :", self.__nom)
        print("Prénom :", self.__prenom)
        print("Âge :", self.__age)
        print("Identifiant :", self.__id)
        print("Moyenne des frais médicaux :", self.calmoy())
        print("Évaluation financière :", self.eval())

# Programme principal
# Création de l'objet
patient = Patient()
# Initialisation
patient.initialiser()
# Saisie des informations
patient.sinfos()
patient.sfrais()
# Affichage des résultats
patient.afficher()
```