

TP3 : Classes et Objets

Le but de ce TP est de :

- Comprendre le concept de classe et objet
- Savoir créer et utiliser des méthodes et attributs
- Manipuler le constructeur
- Construire un mini-système orienté objets simple

Exercice 1:

1. Créer une classe **Etudiant** avec :

Les attributs : **nom, age, moyenne**

Et les méthodes :

- **afficher_infos()** : permettant d'affiche le nom, l'âge et la moyenne
- **mention()** : permettant d'affiche la mention correspondant à la moyenne :
 - $moyenne \geq 16 \rightarrow$ "**Excellent**"
 - $14 \leq moyenne < 16 \rightarrow$ "**Très Bien**"
 - $12 \leq moyenne < 14 \rightarrow$ "**Bien**"
 - $10 \leq moyenne < 12 \rightarrow$ "**Passable**"
 - $moyenne < 10 \rightarrow$ "**Insuffisant**"

2. Créer un objet de la classe **Etudiant**, en utilisant vos propres informations :

- Votre nom
- Votre âge
- Votre moyenne

Exercice 2 :

1. Créer une classe **CompteBancaire** avec :

- numéro
- titulaire
- solde

Méthodes :

- **deposer(montant)**
- **retirer(montant)** (refuser si solde insuffisant)
- **afficher()**

2. Créer un compte et faire plusieurs opérations.

Exercice 3 :

Créer une classe **Film** avec titre, réalisateur, durée.

Méthodes :

- **afficher()**
- **est_long()** : True si durée > 120 minutes

Créer 2 films et tester.

Exercice 4:

On souhaite créer un programme orienté objet permettant de gérer un petit parc automobile.

1. Créer une classe **Voiture** contenant :

Attributs :

- marque : chaîne de caractères
- modele : chaîne de caractères
- annee : entier représentant l'année de fabrication

Méthodes :

- age(annee_courante=2025) : retourne l'âge de la voiture (année courante – année de fabrication).
- __str__() : retourne une représentation textuelle d'une voiture sous la forme : "**Marque Modèle (Année)**"

2. Créer une classe **ParcAuto** permettant de gérer plusieurs voitures.

Attribut :

- voitures : une liste contenant les objets de type Voiture

Méthodes :

- ajouter_voiture(voiture) : ajoute un objet Voiture au parc
- afficher_parc() : affiche la liste complète des voitures du parc
- age_moyen() : calcule et renvoie l'âge moyen des voitures du parc (0 si le parc est vide)

3. Dans le programme principal :

- Créer un objet ParcAuto
- Ajouter au moins **trois voitures** avec leur marque, modèle et année
- Afficher toutes les voitures du parc

#exercice1

```
class Etudiant:
```

```
    def __init__(self, nom, age, moyenne):
```

```
        self.nom = nom
```

```
        self.age = age
```

```
        self.moyenne = moyenne
```

```
    def afficher_infos(self):
```

```
        print(f'Nom : {self.nom}, Âge : {self.age}, Moyenne : {self.moyenne}')
```

```
    def mention(self):
```

```
        if self.moyenne >= 16:
```

```
            return "Excellent"
```

```
        elif self.moyenne >= 12:
```

```
            return "Bien"
```

```
        elif self.moyenne >= 10:
```

```
            return "Passable"
```

```
        else:
```

```
            return "Insuffisant"
```

```
r=Etudiant("ahmed",23,13.5)
```

```
r.afficher_infos()
```

```
m=r.mention()
```

```
print(m)
```

#exercice2

```
class CompteBancaire:
```

```
    def __init__(self, numero, titulaire, solde=0):
```

```
        self.numero = numero
```

```
        self.titulaire = titulaire
```

```
        self.solde = solde
```

```
    def deposer(self, montant):
```

```
        if montant > 0:
```

```
            self.solde += montant
```

```
            print(f'{montant} DA déposés.")
```

```
        else:
```

```
            print("Montant invalide !")
```

```
    def retirer(self, montant):
```

```
        if montant <= 0:
```

```
            print("Montant invalide !")
```

```
        elif montant > self.solde:
```

```
            print("✘ Retrait refusé : solde insuffisant.")
```

```
        else:
```

```
            self.solde -= montant
```

```
            print(f'{montant} DA retirés.")
```

```
    def afficher(self):
```

```
        print("=== Informations du Compte Bancaire ===")
```

```

    print(f"Numéro : {self.numero}")
    print(f"Titulaire : {self.titulaire}")
    print(f"Solde : {self.solde} DA")
    print("-----")

# --- Programme principal ---
#if __name__ == "__main__":
# Création d'un compte
compte1 = CompteBancaire("C123", "Ahmed", 500)

# Opérations
compte1.afficher()
compte1.deposer(200)
compte1.retirer(100)
compte1.retirer(700) # Retrait refusé
compte1.afficher()

#exercice 3
class Film:
    def __init__(self, titre, realisateur, duree):
        self.titre = titre
        self.realisateur = realisateur
        self.duree = duree # en minutes

    def afficher(self):
        print("=== Informations du Film ===")
        print(f"Titre : {self.titre}")
        print(f"Réalisateur : {self.realisateur}")
        print(f"Durée : {self.duree} minutes")
        print("-----")

    def est_long(self):
        return self.duree > 120

# --- Programme principal ---

# Création de quelques films
f1 = Film("Inception", "Christopher Nolan", 148)
f2 = Film("Le Roi Lion", "Disney", 88)
f3 = Film("Avatar", "James Cameron", 162)

# Test des méthodes
f1.afficher()
print("Film long ?", f1.est_long()) # True

f2.afficher()
print("Film long ?", f2.est_long()) # False

f3.afficher()
print("Film long ?", f3.est_long()) # True

```

**#exercice 4****class Voiture:**

```
def __init__(self, marque, modele, annee):  
    self.marque = marque  
    self.modele = modele  
    self.annee = annee
```

```
def age(self, annee_courante=2025):  
    return annee_courante - self.annee
```

```
def __str__(self):  
    return f'{self.marque} {self.modele} ({self.annee})'
```

class ParcAuto:

```
def __init__(self):  
    self.voitures = []
```

```
def ajouter_voiture(self, voiture):  
    self.voitures.append(voiture)
```

```
def afficher_parc(self):  
    print("=== Liste des voitures ===")  
    for v in self.voitures:  
        print(v)
```

```
# --- Programme principal ---
```

```
if __name__ == "__main__":
```

```
    p = ParcAuto()
```

```
    p.ajouter_voiture(Voiture("Peugeot", "208", 2019))
```

```
    p.ajouter_voiture(Voiture("Renault", "Clio", 2021))
```

```
    p.ajouter_voiture(Voiture("Tesla", "Model 3", 2023))
```

```
    p.afficher_parc()
```