

Exercices de Bac NSI – Programmation Orientée Objet (POO)

Série 1 – Notions de base de la POO

Exercice 1 : Classe simple – Lampe

On souhaite modéliser une lampe :

```
class Lampe:
    def __init__(self, couleur):
        self.couleur = couleur
        self.allumee = False

    def allumer(self):
        self.allumee = True

    def eteindre(self):
        self.allumee = False

    def __str__(self):
        etat = "allumée" if self.allumee else "éteinte"
        return f"Lampe {self.couleur} ({etat})"
```

1. Que contient l'attribut `self.allumee` juste après la création d'une lampe ?
2. Écrire les instructions permettant : - de créer une lampe rouge ; - de l'allumer ; - d'afficher son état.
3. Modifier la classe pour compter le **nombre total de lampes créées**.

Exercice 2 : Compte bancaire

On souhaite modéliser un compte bancaire :

```
class Compte:
    def __init__(self, titulaire, solde=0):
        self.titulaire = titulaire
        self.solde = solde

    def deposer(self, montant):
        self.solde += montant

    def retirer(self, montant):
        if montant <= self.solde:
            self.solde -= montant
        else:
            print("Solde insuffisant.")

    def __str__(self):
        return f"Compte de {self.titulaire} : {self.solde} €"
```

1. Créez un compte pour "Alice" avec un solde initial de 100 €.
2. Faites un dépôt de 50 €, puis un retrait de 120 €.
3. Ajoutez une méthode `virement(self, autre_compte, montant)`.
4. Que se passe-t-il si le montant du virement dépasse le solde ?

Série 2 – Héritage et polymorphisme

Exercice 3 : Formes géométriques

On souhaite représenter des formes avec une classe mère *Forme* et deux classes filles *Cercle* et *Rectangle*.

1. Écrire la classe *Forme* contenant : - un attribut *nom* ; - une méthode *aire()* qui renvoie 0 par défaut.
2. Écrire la classe *Cercle(Forme)* avec un attribut *rayon* et une méthode *aire()* calculant $\pi \cdot r^2$.
3. Écrire la classe *Rectangle(Forme)* avec les attributs *largeur* et *hauteur* et une méthode *aire()* calculant *largeur* x *hauteur*.
4. Écrire une fonction *afficher_aire(forme)* qui affiche l'aire de n'importe quelle forme.

Exercice 4 : Zoo

```
class Animal:
    def __init__(self, nom):
        self.nom = nom

    def parler(self):
        return "..."
```

Créer les classes *Chien* et *Chat* héritant de *Animal*.

- Le chien "aboie" et le chat "miaule".
- Compléter pour que la méthode *parler()* renvoie respectivement "Woof!" et "Miaou!".
- Créer une liste d'animaux et afficher leur cri grâce à une boucle.

Série 3 – Sujet type Bac

Exercice 5 : Réseau social simplifié

```
class Utilisateur:
    def __init__(self, nom):
        self.nom = nom
        self.amis = []

    def ajouter_ami(self, autre):
        if autre not in self.amis and autre != self:
            self.amis.append(autre)
            autre.amis.append(self)

    def est_ami(self, autre):
        return autre in self.amis

    def __str__(self):
        return f"{self.nom} ({len(self.amis)} amis)"
```

1. Créer trois utilisateurs : *alice*, *bob*, *carla*.
2. *Alice* et *Bob* deviennent amis, *Bob* et *Carla* aussi.
3. Vérifier si *Alice* et *Carla* sont amies.
4. Ajouter une méthode *suggestions(self)* qui renvoie la liste des amis de ses amis qui ne sont pas encore ses amis.