

Programmation orientée objet

Exercice 1

1. Créer une classe Gateau
2. Ajouter les attributs suivants et les initialiser dans le constructeur :
 - nom gâteau : str
 - temps cuisson : int
 - liste ingrédients : list de str
 - étapes recettes : list de str
 - nom du créateur : str
3. Ajouter une méthode qui affiche les ingrédients de la recette
4. Ajouter une méthode qui affiche les étapes de la recette
5. Instancier un objet gâteau qui affiche les ingrédients ainsi que les étapes de préparation du gâteau

Exercice 2

1. Créer une classe nommée CompteBancaire qui représente un compte bancaire, ayant pour attributs :
 - numeroCompte (type numérique)
 - nom (nom du propriétaire du compte du type chaîne)
 - solde
2. Créer un constructeur ayant comme paramètres : numero_compte, nom, solde
3. Créer une méthode Versement() qui gère les versements
4. Créer une méthode Retrait() qui gère les retraits
5. Créer une méthode Agios() permettant d'appliquer les agios à un pourcentage de 5 % du solde
6. Créer une méthode afficher() permettant d'afficher les détails sur le compte

Exercice 3

1. Créer une classe WaterTank qui possédera les attributs d'instance suivants :
 - Poids de la citerne à vide : float
 - Capacité maximale : float
 - Niveau de remplissage : float
2. Créer les méthodes suivantes propre à chaque instance de classe :
 - Méthode indiquant le poids total
 - Méthode pour remplir la citerne avec un nombre de litre d'eau
 - Méthode pour vider la citerne d'eau d'un nombre de litre d'eau

3. Créer un attribut de classe qui contiendra la totalité des volumes d'eau des citernes
4. Testez votre programme

Exercice 4

1. Créer une classe `Personne`, contenant le nom de la personne, son prénom, son numéro de téléphone et son email. Une méthode `__toString` pour afficher les données de la personne.
2. Créer une classe `Travailleur`, qui hérite de la classe `Personne` et étend avec les attributs nom d'entreprise, adresse entreprise et téléphone professionnel. Une méthode `__toString` pour afficher les données et qui réutilise celle de `Personne`.
3. Créer une classe `Scientifique` qui hérite de la classe `Travailleur` et étend avec les attributs de type list disciplines (physique, chimie, mathématique, ...) et types du scientifique (théorique, expérimental, informatique...) Une méthode `__toString` pour afficher les données et qui réutilise celle de `Travailleur`.