

Encapsulation et Constructeurs

Créez une classe `Personne` avec les attributs `nom`, `âge` et `ville` (tous privés), ainsi que des constructeurs pour initialiser ces attributs. Ajoutez également des méthodes `getter` et `setter` pour accéder et modifier ces attributs de manière encapsulée.

Polymorphisme

Définissez une classe abstraite `Forme` avec une méthode abstraite `calculerAire()` et une méthode non abstraite `afficherType()` qui affiche le type de forme. Créez ensuite des classes `Cercle` et `Rectangle` qui héritent de `Forme` et implémentent les méthodes nécessaires. Utilisez le polymorphisme pour calculer l'aire de différentes formes.

Fonctions et Méthodes

Écrivez une méthode `estPalindrome(String mot)` qui prend un mot en entrée et renvoie `true` s'il s'agit d'un palindrome (un mot qui se lit de la même manière de gauche à droite et de droite à gauche), sinon renvoie `false`.

Polymorphisme et Interfaces

Créez une interface `Vehicule` avec une méthode `deplacer()`. Ensuite, créez des classes `Voiture`, `Avion` et `Bateau` qui implémentent cette interface. Chacune de ces classes devrait avoir sa propre implémentation de la méthode `deplacer()`, affichant un message spécifique au type de véhicule.

Méthodes Statiques

Écrivez une classe `MathUtil` avec une méthode statique `calculerMoyenne` qui prend un tableau de nombres en entrée et renvoie la moyenne de ces nombres.

Surcharges de Méthodes

Définissez une classe Calculatrice avec des méthodes surchargées pour effectuer des opérations mathématiques telles que l'addition, la soustraction, la multiplication et la division pour différents types de données (entiers, doubles, etc.).

Constructeurs surchargés

Créez une classe Article pour représenter un article en vente. Cette classe devrait avoir plusieurs constructeurs surchargés pour permettre l'initialisation de l'article avec différentes combinaisons d'attributs tels que le nom, le prix, la quantité en stock, etc.

Polymorphisme avec Héritage

Créez une hiérarchie de classes pour représenter différents types d'animaux (par exemple, Oiseau, Chien, Chat, etc.). Chaque classe devrait avoir une méthode crier() qui affiche un cri spécifique à l'animal. Utilisez le polymorphisme pour appeler la méthode crier() sur des objets d'animaux de différentes classes.

Utilisation de Méthodes d'Objet

Créez une classe CompteBancaire avec des méthodes pour déposer, retirer et consulter le solde. Réalisez une série d'opérations bancaires pour tester ces méthodes.

Méthodes Récursives

Écrivez une méthode récursive calculerFactorielle(int n) qui calcule la factorielle d'un nombre entier n.