

Exercice 1: Classes de Base et Dérivées

Créer une classe de base Animal et une classe dérivée Dog qui hérite de Animal.

Instructions

1. Créez une classe Animal avec les attributs name et age, et une méthode makeSound() qui affiche un son générique.
2. Créez une classe Dog qui hérite de Animal et ajoute un attribut breed. Overridez la méthode makeSound() pour afficher “Woof”.

Exercice 2: Utilisation de super

Utiliser le mot-clé `super` pour accéder à la méthode de la classe parente.

Instructions

1. Ajoutez une méthode `displayInfo()` dans la classe `Animal` pour afficher les informations de base.
2. Overridez `displayInfo()` dans la classe `Dog` pour inclure le `breed` et utilisez `super` pour réutiliser la méthode de la classe parente.

Exercice 3: Héritage et Constructeurs

Comprendre comment les constructeurs fonctionnent avec l'héritage.

Instructions

1. Créez une classe de base `Vehicle` avec un constructeur qui initialise `make` et `model`.
2. Créez une classe dérivée `Car` avec un constructeur qui initialise `make`, `model` et `year`.
3. Appelez le constructeur de la classe parente depuis le constructeur de la classe dérivée en utilisant `super`.

Exercice 4: Polymorphisme

Utiliser le polymorphisme avec l'héritage.

Instructions

1. Créez une méthode `printSound()` qui accepte un objet `Animal` et appelle sa méthode `makeSound()`.
2. Passez différents objets `Animal` et `Dog` à cette méthode pour observer le comportement polymorphique.

Exercice 5: Classe Abstraite

Comprendre l'utilisation des classes abstraites et des méthodes abstraites.

Instructions

1. Créez une classe abstraite Shape avec une méthode abstraite calculateArea().
2. Créez des classes Circle et Rectangle qui étendent Shape et implémentent calculateArea().