

Exercises on Inheritance in Java

Single Inheritance

1. Vehicle and Car

- **Task:** Create a class `Vehicle` with a method `start()` that prints "Vehicle started". Create a subclass `Car` that extends `Vehicle` and overrides the `start()` method to print "Car started". Instantiate both classes and call their `start()` methods.

2. Person and Employee

- **Task:** Create a class `Person` with fields `name` and `age`, and a method `display()` that prints the name and age. Create a subclass `Employee` that adds a field `salary` and overrides `display()` to include salary information. Instantiate both classes and call their `display()` methods.

Multilevel Inheritance

3. Animal Hierarchy

- **Task:** Create a base class `Animal` with a method `makeSound()`. Create a subclass `Dog` that extends `Animal` and overrides `makeSound()` to print "Woof!". Then, create another subclass `Puppy` that extends `Dog` and overrides `makeSound()` to print "Yap!". Instantiate all classes and call their `makeSound()` methods.

Hierarchical Inheritance

4. Shape Hierarchy

- **Task:** Create a class `Shape` with a method `getArea()`. Create two subclasses: `Rectangle` and `Circle`. The `Rectangle` class should have fields for `length` and `width`, and override `getArea()` to return the area of the rectangle. The `Circle` class should have a field for `radius` and override `getArea()` to return the area of the circle. Instantiate all classes and call their `getArea()` methods.

5. Bank Account Hierarchy

- **Task:** Create a class `BankAccount` with a method `deposit()`. Create two subclasses: `SavingsAccount` and `CheckingAccount`. The `SavingsAccount` class should override `deposit()` to add interest, while `CheckingAccount` should display a message when depositing. Instantiate both subclasses and call their `deposit()` methods.

Additional Exercises

6. Employee Management

- **Task:** Create a base class `Employee` with methods `work()` and `getSalary()`. Create subclasses `Manager` and `Developer`. The `Manager` class should override `work()` to include management tasks, and the `Developer` class should add a method `writeCode()`. Instantiate both subclasses and demonstrate their methods.

7. Animal Movement

- **Task:** Create a class `Animal` with a method `move()`. Create subclasses `Bird` and `Fish`, where `Bird` overrides `move()` to print "The bird is flying" and `Fish` overrides it to print "The fish is swimming". Instantiate both subclasses and call their `move()` methods.