1 Teoria

- Che cos'è l'ereditarietà in Java? A cosa serve?
- Come si dichiara una classe che estende un'altra classe?
- Qual è la differenza tra classe base (superclasse) e classe derivata (sottoclasse)?
- Cos'è il metodo `super` e quando si usa?
- Che cos'è il polimorfismo in Java? Come si manifesta?
- Cos'è l'overriding di un metodo? Come si differenzia dall'overloading?
- È possibile chiamare un metodo della superclasse da una sottoclasse? Come?
- Cos'è il binding dinamico (late binding) nei metodi?
- Che ruolo hanno i metodi astratti e le classi astratte nell'ereditarietà?
- Cos'è un'interfaccia in Java e come differisce dall'ereditarietà delle classi?

2 Ereditarietà e Polimorfismo

1. **Classe base Animale**
   Crea una classe `Animale` con un metodo `verso()` che stampa un messaggio generico.
2. **Classi derivata Cane e Gatto**
   Crea le classi `Cane` e `Gatto` che estendono `Animale` e sovrascrivono il metodo `verso()` per stampare suoni specifici.
3. **Polimorfismo con Animale**
   Scrivi un metodo che riceve un oggetto `Animale` e chiama `verso()`. Prova a passare oggetti di tipo `Cane` e `Gatto`.
4. **Costruttore con super**
   Nella classe derivata `Cane`, crea un costruttore che richiama il costruttore della superclasse `Animale` usando `super`.
5. **Override e overloading**
   Nella classe `Gatto`, scrivi un metodo `verso()` che sovrascrive quello della superclasse e un metodo `verso(String emozione)` che lo sovraccarica.
6. **Classe astratta Veicolo**
   Definisci una classe astratta `Veicolo` con un metodo astratto `muovi()`. Crea due classi derivate `Auto` e `Bicicletta` che implementano `muovi()`.

7. **Interfaccia Volante**
   Crea un'interfaccia `Volante` con metodo `vola()`. Fai in modo che la classe `Uccello` implementi questa interfaccia.

8. **Ereditarietà multipla tramite interfacce**
   Crea due interfacce `Nuotatore` e `Volante`. Crea una classe `Anatra` che implementa entrambe.

9. **Casting e istanze**
   Dato un oggetto di tipo `Animale`, verifica se è istanza di `Cane` usando `instanceof` e fai un cast per chiamare un metodo specifico.

10. **Metodo finale**
    Spiega e mostra un esempio in cui un metodo è dichiarato `final` e non può essere sovrascritto nella sottoclasse.