

Claro! Aqui está a lista de 10 exercícios, agora incorporando dicionários, listas, tuplas e conjuntos em diferentes níveis de dificuldade.

---

### Nível Fácil (3 exercícios)

1. **Herança e Listas:** Crie uma classe `Animal` com um atributo `nome` e um método `fazer_som()`. Crie uma classe `Gato` que herda de `Animal` e uma classe `Cachorro` que herda de `Animal`. Crie uma **lista** que contenha um objeto de cada classe (`Gato` e `Cachorro`) e itere sobre ela para chamar o método `fazer_som()`.
  2. **Encapsulamento e Dicionários:** Crie uma classe `Pessoa` com os atributos privados `__nome` e `__idade`. Use um método `get_info()` que retorne os dados da pessoa como um **dicionário** no formato `{'nome': '...', 'idade': '...'}`.
  3. **Polimorfismo e Tuplas:** Crie uma classe `Conta` com um método `obter_saldo()`. Crie uma classe filha `ContaPoupanca` que sobrescreve o método `obter_saldo()` para retornar o saldo e a taxa de juros como uma **tupla** (saldo, juros).
- 

### Nível Médio (4 exercícios)

4. **Hierarquia de Classes e Dicionários:** Crie uma hierarquia de classes para um sistema de controle de estoque. Comece com uma classe `Produto` com atributos como `nome` e `preco`. Em seguida, crie classes filhas como `Eletronico` e `Roupa`. A classe `Estoque` deve usar um **dicionário** para armazenar os produtos, onde a chave é o nome do produto e o valor é o objeto do produto.
  5. **Polimorfismo e Listas de Dicionários:** Usando as classes do exercício 4, crie uma **lista de dicionários**, onde cada dicionário representa um produto e suas informações. Crie uma classe `CarrinhoDeCompras` que receba essa lista e use polimorfismo para calcular o total da compra.
  6. **Encapsulamento e Conjuntos:** Crie uma classe `RedeSocial` com um atributo privado `__amigos`, que deve ser um **conjunto** para garantir que não haja duplicatas. Implemente os métodos `adicionar_amigo()` e `remover_amigo()`.
  7. **Herança e Tuplas:** Crie uma classe base `Veiculo` com um método `get_info()`. Crie duas classes filhas, `Carro` e `Moto`. O método `get_info()` de cada classe filha deve retornar informações específicas (como `marca`, `modelo` e `ano`) como uma **tupla** de strings.
- 

### Nível Intermediário (3 exercícios)

8. **Herança Múltipla com Dicionários e Conjuntos:** (Este é um desafio um pouco mais avançado) Crie uma classe Funcionario com atributos como nome e um método trabalhar(). Crie uma classe Gerente que herda de Funcionario e adicione um atributo privado \_\_equipe que é um **conjunto**. Use um método add\_membro() para adicionar membros.
9. **Encapsulamento com Listas e Dicionários:** Crie uma classe SistemaDeNotas com um atributo privado \_\_alunos, que é um **dicionário** onde a chave é o nome do aluno e o valor é uma **lista** de suas notas. Implemente os métodos adicionar\_nota(aluno, nota) e calcular\_media(aluno).
10. **Projeto de Sistema de Pedidos:** Crie um sistema de pedidos de restaurante.
  - Crie uma classe ItemMenu com atributos como nome e preco.
  - Crie classes filhas, Bebida e Prato, que herdam de ItemMenu e podem ter atributos adicionais (como tamanho para a Bebida).
  - Crie uma classe Pedido que use um **dicionário** para armazenar os itens e suas quantidades, e tenha um método polimórfico calcular\_total() que itera sobre os itens do pedido para somar o valor final.