

# Lista de Exercícios de Composição

## 1. Montando um Carro (Fácil)

- **Classes:** Motor e Carro.
- **Classe Motor:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `ligar()` que imprime "O motor ligou."
- **Classe Carro:**
  - **Atributo (Composição):** `motor`, que deve ser uma instância de Motor.
  - **Método:** `__init__` que inicializa o atributo `motor`.
  - **Método:** `ligar_carro()` que chama o método `ligar()` do seu objeto `motor`.

## 2. Criando uma Cafeteira (Fácil)

- **Classes:** `GraoDeCafe`, `Agua` e `Cafeteira`.
- **Classe GraoDeCafe:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `moer()` que imprime "Os grãos de café foram moídos."
- **Classe Agua:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `aquecer()` que imprime "A água está aquecida."
- **Classe Cafeteira:**
  - **Atributos (Composição):** `grao` e `agua`, que devem ser instâncias das classes `GraoDeCafe` e `Agua`.
  - **Método:** `__init__` que inicializa os atributos `grao` e `agua`.
  - **Método:** `preparar_cafe()` que chama os métodos `moer()` do seu `grao` e `aquecer()` da sua `agua`.

### 3. Segmento de Reta (Fácil/Médio)

- **Classes:** Ponto e SegmentoDeReta.
- **Classe Ponto:**
  - **Atributos:** x e y.
  - **Método:** `__init__(x, y)`.
- **Classe SegmentoDeReta:**
  - **Atributos (Composição):** ponto1 e ponto2, que devem ser instâncias de Ponto.
  - **Método:** `__init__(ponto1, ponto2)`.
  - **Método:** `calcular_comprimento()` que retorna a distância entre os dois pontos.
- **Dica:** Use o módulo math e a fórmula da distância euclidiana:  
$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### 4. Dispositivos de um Computador (Fácil/Médio)

- **Classes:** Teclado, Mouse, Monitor e Computador.
- **Classes Teclado, Mouse, Monitor:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `ligar()` que imprime uma mensagem indicando que o dispositivo está ligado (ex: "O teclado foi ativado.").
- **Classe Computador:**
  - **Atributos (Composição):** teclado, mouse e monitor, que devem ser instâncias das classes correspondentes.
  - **Método:** `__init__` que inicializa os três atributos.
  - **Método:** `ligar_computador()` que chama o método `ligar()` de cada um dos seus dispositivos.

## 5. Casa e Cômodos (Médio)

- **Classes:** Comodo e Casa.
- **Classe Comodo:**
  - **Atributo:** nome.
  - **Método:** `__init__(nome)`.
- **Classe Casa:**
  - **Atributo (Composição):** `comodos`, que deve ser uma **lista vazia**.
  - **Método:** `__init__` que inicializa a lista `comodos`.
  - **Método:** `adicionar_comodo(nome)` que cria uma instância de `Comodo` e a adiciona na lista `comodos`.
  - **Método:** `listar_comodos()` que itera sobre a lista e imprime o nome de cada cômodo.

## 6. Playlist de Músicas (Médio)

- **Classes:** `Musica` e `Playlist`.
- **Classe Musica:**
  - **Atributos:** `titulo` e `artista`.
  - **Método:** `__init__(titulo, artista)`.
- **Classe Playlist:**
  - **Atributo (Composição):** `musicas`, que deve ser uma **lista vazia**.
  - **Método:** `__init__` que inicializa a lista `musicas`.
  - **Método:** `adicionar_musica(musica)` que recebe um objeto `Musica` e o adiciona na lista.
  - **Método:** `remover_musica(musica)` que remove um objeto `Musica` da lista.
  - **Método:** `tocar_playlist()` que itera sobre a lista de músicas e imprime "Tocando [titulo] de [artista]".

## 7. Carro com Motor e Rodas (Médio)

- **Classes:** Motor, Roda e Carro.
- **Classe Motor:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `ligar()` que imprime "Motor ligado."
- **Classe Roda:**
  - **Método:** `__init__` (sem atributos).
  - **Método:** `girar()` que imprime "A roda está girando."
- **Classe Carro:**
  - **Atributos (Composição):** motor e rodas (uma **lista**).
  - **Método:** `__init__` que inicializa o motor e cria uma lista com **4** instâncias de Roda.
  - **Método:** `ligar_carro()` que chama o método `ligar()` do motor e, em seguida, itera sobre a lista rodas para chamar o método `girar()` de cada uma.

## 8. Carrinho de Compras (Médio)

- **Classes:** Produto, ItemDeCompra e CarrinhoDeCompras.
- **Classe Produto:**
  - **Atributos:** nome e preco.
  - **Método:** `__init__(nome, preco)`.
- **Classe ItemDeCompra:**
  - **Atributos (Composição):** produto (um objeto Produto) e quantidade.
  - **Método:** `__init__(produto, quantidade)`.
- **Classe CarrinhoDeCompras:**
  - **Atributo (Composição):** itens, que deve ser uma **lista vazia**.
  - **Método:** `__init__` que inicializa a lista itens.
  - **Método:** `adicionar_item(produto, quantidade)` que cria um objeto ItemDeCompra e o adiciona na lista.
  - **Método:** `calcular_total()` que itera sobre a lista itens e retorna a soma total dos preços (preço do produto \* quantidade).

## 9. Conta Bancária (Médio/Difícil)

- **Classes:** Cliente, Transacao e ContaCorrente.
- **Classe Cliente:**
  - **Atributos:** nome e cpf.
  - **Método:** `__init__(nome, cpf)`.
- **Classe Transacao:**
  - **Atributos:** tipo (string, ex: "deposito", "saque"), valor e data (pode ser uma string simples ou um objeto datetime).
  - **Método:** `__init__(tipo, valor, data)`.
- **Classe ContaCorrente:**
  - **Atributos (Composição):** cliente (um objeto Cliente), saldo e historico (uma **lista vazia** de Transacao).
  - **Método:** `__init__(cliente, saldo)`.
  - **Método:** `sacar(valor)` que subtrai o valor do saldo e cria uma Transacao negativa no histórico.
  - **Método:** `depositar(valor)` que adiciona o valor ao saldo e cria uma Transacao positiva no histórico.
  - **Método:** `mostrar_historico()` que imprime cada transação na lista.

## 10. Veículos com Herança e Composição (Médio/Difícil)

- **Classes:** Motor, Roda e Veiculo (base), Carro e Moto (herdam de Veiculo).
- **Classe Motor e Roda:** As mesmas do Exercício 7.
- **Classe Veiculo:**
  - **Atributos (Composição):** motor e rodas (uma **lista**).
  - **Método:** `__init__(self, numero_de_rodas)`. O `__init__` deve instanciar um Motor e criar a lista de Rodas com o número especificado.
  - **Método:** `ligar()` que chama o método `ligar()` do motor e depois o método `girar()` de todas as rodas.
- **Classe Carro:**
  - **Herança:** Carro deve herdar de Veiculo.
  - **Método:** `__init__()` que chama `super().__init__(4)`.
- **Classe Moto:**
  - **Herança:** Moto deve herdar de Veiculo.
  - **Método:** `__init__()` que chama `super().__init__(2)`.