



## LISTA DE EXERCÍCIOS 09 - POO: CLASSES, OBJETOS, ATRIBUTOS, MÉTODOS

Utilize como base para as questões os arquivos feitos em sala de aula e disponíveis no GitHub `lib_banco.py`:

([https://github.com/ricardokleber/Programacao-Estruturada-e-Orientada-a-Objetos-2023/blob/main/Lista\\_de\\_Exercicios\\_09/Exercicios\\_Resolvidos/lib\\_banco.py](https://github.com/ricardokleber/Programacao-Estruturada-e-Orientada-a-Objetos-2023/blob/main/Lista_de_Exercicios_09/Exercicios_Resolvidos/lib_banco.py))

`banco.py`:

([https://github.com/ricardokleber/Programacao-Estruturada-e-Orientada-a-Objetos-2023/blob/main/Lista\\_de\\_Exercicios\\_09/Exercicios\\_Resolvidos/banco.py](https://github.com/ricardokleber/Programacao-Estruturada-e-Orientada-a-Objetos-2023/blob/main/Lista_de_Exercicios_09/Exercicios_Resolvidos/banco.py))

1) Implemente o **encapsulamento** (no atributo saldo):

- Altere o arquivo da biblioteca (`lib_banco.py`) alterando o atributo 'saldo' da classe 'Conta' de modo que ele passe a ser do tipo 'privado' (não consiga ser acessado diretamente quando instanciado em um objeto). Altere todas as referências a este atributo nos métodos da classe.
- Altere o arquivo do programa principal (`banco.py`) de modo que a consulta volte a funcionar (execute o método de consulta ao saldo e não tente exibir diretamente o valor do atributo saldo). Faça o mesmo em todos os demais itens do programa que exibem o saldo (depositar e sacar).

2) Utilize agora **herança** para criar duas novas classe na biblioteca (`lib_banco.py`):

- A nova classe 'ContaCorrente':
  - Deve ser **subclasse** da superclasse 'Conta'
  - Deve ter um novo atributo: 'limite' (com valor padrão = 1000)
  - Deve ter um novo método: 'alterar\_limite' (que quando instanciada recebe um valor por parâmetro e altera o valor do atributo 'limite')
  - O atributo 'limite' deve estar encapsulado (ser do tipo 'privado')
- A nova classe 'ContaPoupanca':
  - Deve ser **subclasse** da superclasse 'Conta'
  - Deve ter um novo atributo: 'taxa\_de\_juros' (com valor padrão = 0)
  - Deve ter um novo método: 'set\_taxa\_de\_juros' (que quando instanciada recebe um valor por parâmetro e altera o valor do atributo 'taxa\_de\_juros')
  - Deve ter um novo método: 'aplicar\_juros' (que quando instanciada incrementa o atributo 'saldo' com os juros calculados de acordo com a taxa)
  - O atributo 'taxa\_de\_juros' deve estar encapsulado (ser do tipo 'privado')

3) Altere o arquivo do programa principal (`banco.py`) de modo que:

- 'Cadastrar Cliente' crie/instancie uma conta corrente e uma conta poupança
- 'Gerar Número da Conta' gere números aleatórios de conta para cada conta criada
- 'Consultar Saldo' exiba o saldo das duas contas (corrente e poupança)
- 'Depositar' pergunte antes em qual conta será feito o depósito
- 'Sacar' pergunte antes de qual conta será feito o saque