



Universidade Federal do Piauí - UFPI  
Campus Senador Helvídio Nunes de Barros - Picos-PI  
Sistemas de Informação - SI  
Programação Orientada a Objeto

## Lista 04 – Orientação a Objetos, Herança e Interfaces - 18/07/2023

- Desenvolva os códigos em Python.
- Objetivo desta lista é revisar conceitos de introdução a programação orientada a objetos, herança, interfaces e praticar a linguagem Python.
- Data da entrega: 31/07/2023

Crie um sistema bancário em Python que satisfaça as condições descritas a seguir. **Além dos códigos, também deve ser enviado um diagrama de classes contendo as classes criadas, seus atributos, métodos e relações entre as mesmas.**

1 – Cadastrar cliente (Atributos: nome, cpf)

2 – Criar conta corrente (Atributos: número, saldo, histórico de transações).

3 – Criar conta poupança (Atributos: número, saldo, histórico de transações). Uma conta sempre é associada a um cliente. Cada cliente só pode ter no máximo duas contas, uma corrente e outra poupança. Ele não pode ter mais de uma conta do mesmo tipo.

4 – Criar seguro de vida (Atributos: valor mensal, valor total do seguro). Um seguro de vida sempre está associado a um cliente, e um mesmo cliente pode ter mais de um seguro.

5 – Calcular tributação. Quando essa opção é ativada, o banco calcula a tributação devida por cada cliente, que é um valor fixo de 10 reais, mais 1% do valor do saldo da conta corrente, mais 2% do valor mensal pago no seguro de vida. Nessa opção é calculado e mostrado o histórico do valor total da tributação cobrada. Por exemplo, se na primeira vez que essa opção foi chamada a tributação calculada foi 100 e na segunda 90, na segunda vez deverá aparecer “1 tributação = 100, 2 tributação = 90”. Essa tributação calculada e mostrada representa a soma da tributação de todos os clientes do banco. Não precisa debitar esse valor na conta dos clientes, é somente para computar e mostrar o total.

6 – Sacar.

7 – Depositar.

8 – Transferir. Nas operações de sacar, depositar e transferir, deverá ser selecionado um cliente, em seguida caso ele tenha mais de uma conta (poupança ou corrente), deverá perguntar de qual conta ele deseja fazer a operação. Em seguida a operação deve ser realizada caso ele tenha saldo suficiente, caso não tenha deve ser mostrada a mensagem apropriada.

9 – Imprimir histórico de uma conta. Mostrar todas as operações realizadas em uma conta.

10 – Exibir informações do banco. Mostra quantas contas estão cadastradas no banco, o nome de todos os clientes cadastrados e quantas contas e seguros de vida cada um possui.

**IMPORTANTE:**

Fique à vontade para criar as classes que achar necessário e use a criatividade para adaptar ou incrementar novas funcionalidades as opções acima.

**Utilize herança e interfaces onde for possível**, também utilize as boas práticas estudadas em aula, como declarar atributos privados, tratamentos de erros com try except, dentre outros.

**Não esqueça de enviar o diagrama com as classes, atributos, métodos e a relação entre as classes.**