

**GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**DEPARTAMENTO DE COMPUTAÇÃO - UNIVERSIDADE ESTADUAL DE LONDRINA**  
**PROGRAMAÇÃO ORIENTADA A OBJETO – PROF. BRUNO B. ZARPELÃO**  
**LISTA DE EXERCÍCIOS – INTERFACES**

**Exercício 1:**

Crie uma interface ContaBancaria.

Inclua os seguintes métodos:

- sacar;
- depositar;

Crie duas classes que implementam ContaBancaria: ContaCorrente e ContaPoupanca.

Na classe ContaPoupanca, inclua os seguintes atributos:

- cliente: String;
- numeroConta: String;
- saldo: BigDecimal;
- diaRendimento: int;

Na classe ContaPoupanca, crie os seguintes métodos:

- sacar (implementação do método previsto na interface ContaBancaria): só deve permitir o saque caso o saldo resultante seja maior ou igual a zero;
- depositar (implementação do método previsto na interface ContaBancaria)
- calcularNovoSaldo: recebe como parâmetro a taxa de rendimento e atualiza o saldo;

Na classe ContaCorrente, inclua os seguintes atributos:

- cliente: String;
- numeroConta: String;
- saldo: BigDecimal;
- limite: BigDecimal;

Na classe ContaCorrente, crie os seguintes métodos:

- sacar (implementação do método previsto na interface ContaBancaria): o valor do saque pode ser maior que o saldo disponível, mas deve respeitar o limite previsto para a conta;
- depositar (implementação do método previsto na interface ContaBancaria)

Crie uma interface Tributavel. Esta interface deve ter o seguinte método: calcularTributo(BigDecimal taxaRendimento).

Crie uma classe denominada ContalInvestimento. Ela deve implementar as interfaces ContaBancaria e Tributavel. Inclua os seguintes atributos nesta classe:

- cliente: String;
- numeroConta: String;
- saldo: BigDecimal;

Na classe ContalInvestimento, crie os seguintes métodos:

- sacar (implementação do método previsto na interface ContaBancaria): só deve permitir o saque caso o saldo resultante seja maior ou igual a zero;
- depositar (implementação do método previsto na interface ContaBancaria)
- calcularNovoSaldo: recebe como parâmetro a taxa de rendimento e atualiza o saldo;
- calcularTaxaAdministração: recebe como parâmetro a taxa de rendimento e deve retornar um valor correspondente a 1% do lucro obtido com o rendimento;
- calcularTributo (implementação do método previsto na interface Tributavel): recebe como parâmetro a taxa de rendimento e deve retornar um valor correspondente a 0,5% do lucro obtido com o rendimento;

Faça uma classe Principal que permita ao usuário realizar as seguintes tarefas:

- Cadastrar a conta de um cliente;
- Sacar um valor da sua conta;
- Atualizar uma conta poupança com o seu rendimento;
- Depositar um determinado valor na conta;
- Mostrar o saldo de uma conta;
- Calcular os tributos de uma conta;
- Calcula a taxa de administração de uma conta investimento;