

ELE078 - Programação Orientada a Objetos

Atividade Prática 10 - Polimorfismo, Funções Virtuais e Classes Abstratas

Lidando com Contas Bancárias

Implemente a hierarquia de classes **ContaBancaria** (classe Base e abstrata), **ContaCorrente** (com senha, número, saldo e quantidade de transações realizadas) e **ContaPoupanca** (com senha, número, saldo e taxa de rendimento). Observação: os atributos comuns para os dois tipos de conta podem ficar na classe Base.

Quando uma **ContaBancaria** for criada, informe a senha da conta por parâmetro. Na classe **ContaBancaria**, crie os seguintes métodos abstratos, como funções virtuais puras):

- `saca(double valor)`
- `deposita(double valor)`
- `tiraExtrato()`

Nesta mesma classe, crie o método `alteraSenha()`, que recebe uma senha por parâmetro e deve confirmar a senha anterior (via teclado), e somente se a senha anterior estiver correta a senha recebida por parâmetro deve ser atribuída.

Implemente os métodos abstratos nas classes **ContaCorrente** e **ContaPoupanca**.

Se necessário, crie métodos de acesso e modificação (GET and SET) para os atributos de **ContaCorrente** e **ContaPoupanca**.

Testando suas Classes

Crie um programa para testar a hierarquia de classes que manipula contas bancárias.

- pergunte (via teclado) quantas contas o usuário deseja criar e crie-as (com a utilização de uma lista (vector de **ContaBancaria**) para armazenar as contas.
- a cada conta criada, pergunte ao usuário se trata-se de uma **ContaCorrente** ou de uma **ContaPoupanca**, e crie a conta de acordo com o informado pelo usuário.
- após as contas terem sido criadas, informe a taxa de rendimento de cada **ContaPoupanca** armazenada.
- realize saques, depósitos e extratos nestas contas.
- imprima a quantidade de transações realizadas nas contas correntes e as taxas de rendimento das contas poupança.

In []: