



++

45697056

Sistemas para Internet

Domain Driven Design

45697056

Encapsulamento, Modificadores de Acesso e Herança - Exercícios

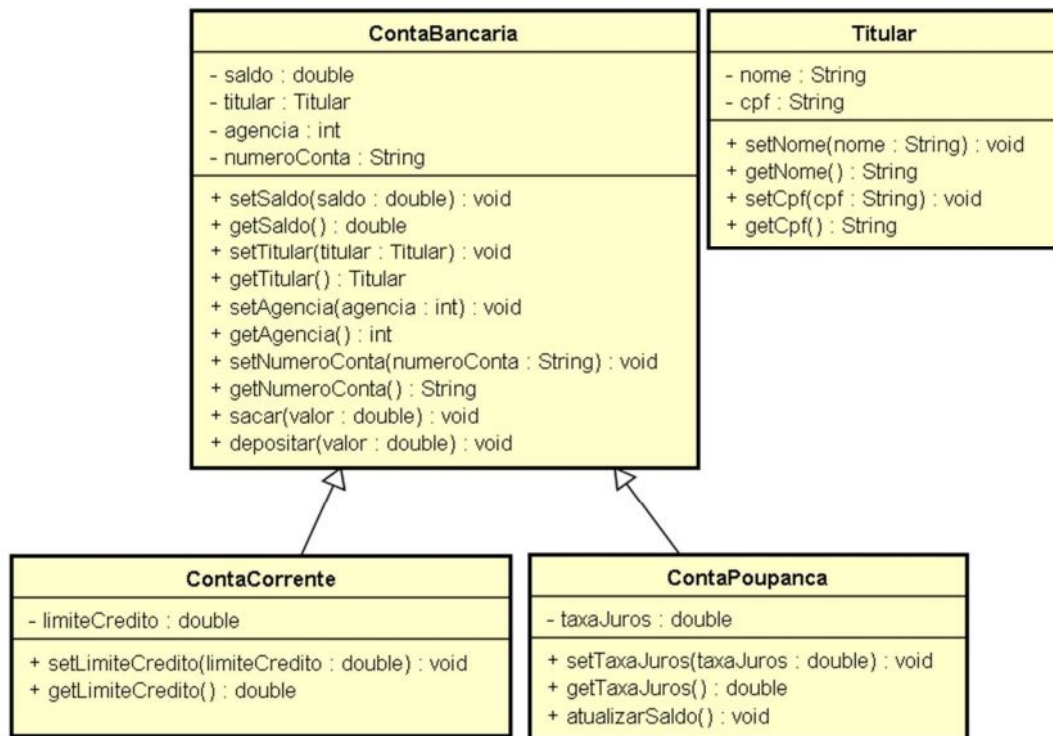
45697056

...



Exercício 01

Implemente em Java o seguinte diagrama de classes:



Exercício 01.1

Faça as seguintes alterações:

- 1) O método **depositar** da classe **ContaBancaria** deve adicionar ao atributo **saldo** o valor passado como parâmetro.
- 2) O método **sacar** da classe **ContaBancaria** deve subtrair do atributo **saldo** o valor passado como parâmetro.
- 3) O método **atualizarSaldo** da classe **ContaPoupanca** deve atualizar o valor do **saldo** através do seguinte cálculo:
 $\text{saldo} + (\text{saldo} * (\text{taxa de juros} / 100))$

Exercício 01.2



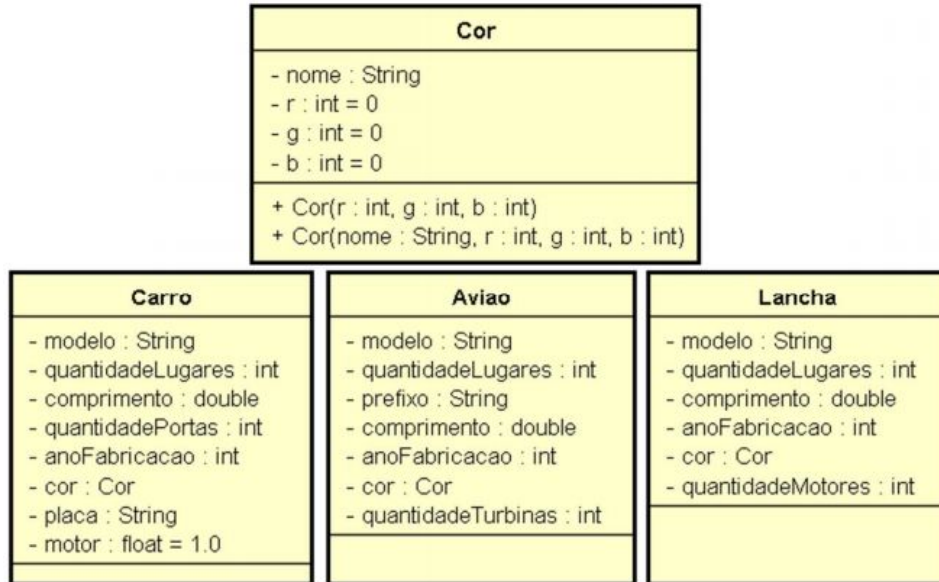
Faça um programa para testar as classes **ContaCorrente** e **ContaPoupanca** e invoque os métodos **sacar** e **depositar** das instâncias destas classes.

Exercício 02

Utilizando seus conhecimentos sobre modelagem de classes e herança, analise as classes exibidas abaixo e em seguida implemente uma nova hierarquia de classes de forma que o conceito de herança seja aplicado e evite as duplicações expostas.

Nota 1: Não é preciso desenhar um novo diagrama, apenas implemente em Java a nova hierarquia de classes.

Nota 2: Os atributos devem estar encapsulados (métodos get e set).



Exercício 03



- 1) Defina o que é encapsulamento e aponte seus benefícios.
- 2) Defina o que são os modificadores de acesso e cite os disponíveis na linguagem Java.
- 3) Defina o que é JavaBean e aponte suas principais características.
- 4) Defina o que é herança e em seguida escreva um código-fonte em Java onde é demonstrada a aplicação deste recurso.



Copyright © 2017 Prof. Douglas Cabral <douglas.cabral@fiap.com.br> <https://www.linkedin.com/in/douglascabral/>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).