

**Lista Prática 10**  
**Paradigma Orientado a Objetos (Java)**

**Problema 1:** Elabore na linguagem Java códigos que atendam aos seguintes requisitos:

(a) Escreva a classe “Sobremesa” com *gets* e *sets* para os atributos “nome” e “calorias” (que devem ser privados). O construtor dessa classe deverá receber como parâmetros nome e calorias.

(b) Defina as operações de instância “ehSaudavel”, que retorna “true”, se e somente se, a sobremesa tem menos de 200 calorias, e “ehDeliciosa”, que retorna “true” para todas as sobremesas.

(c) Crie a classe “GeleiaEmCompota” que herdará da classe Sobremesa. O seu construtor deverá aceitar um único argumento denominado “sabor”; a sua quantidade padrão de calorias é 5 e seu nome deverá ser precedido de “Geléia em Compota de ”, por exemplo, “Geléia em Compota de Morango”.

(d) Inclua métodos *get* e *set* para o atributo sabor.

(e) Modifique a operação “ehDeliciosa” (*overriding*) para retornar “false” se o sabor é alcaçuz e “true” para todos os outros sabores. O comportamento dessa operação para sobremesas que não são geléias em compotas não deve ser alterado.

(f) No *main*, instancie objetos de ambas as classes e chame cada um dos métodos.

**Problema 2:** Elabore na linguagem Java códigos que atendam aos seguintes requisitos:

(a) Escreva a classe “ContaCorrente” que tenha os atributos “cpf do titular” e “saldo da conta”. Esta classe deverá ter métodos *get* e *set* para cada atributo.

(b) Defina métodos para: depositar uma quantia na conta, sacar uma quantia da conta e imprimir os dados referentes a uma conta (cpf e saldo). Para cada saque será debitada também uma taxa de operação equivalente à 0.5% do valor sacado. Os valores a serem sacados e depositados deverão ser passados como parâmetros para os respectivos métodos.

(c) Escreva uma classe “ContaCorrenteEspecial” que herda da classe ContaCorrente para representar uma conta corrente de um cliente especial. Clientes especiais pagam taxas de operação de apenas 0.1% do valor sacado (e não 0.5% como na classe ContaCorrente).

(d) No *main*, faça simulações de depósitos e saques com as duas classes e imprima os resultados através de um método de impressão.