

Lista de Exercícios 13

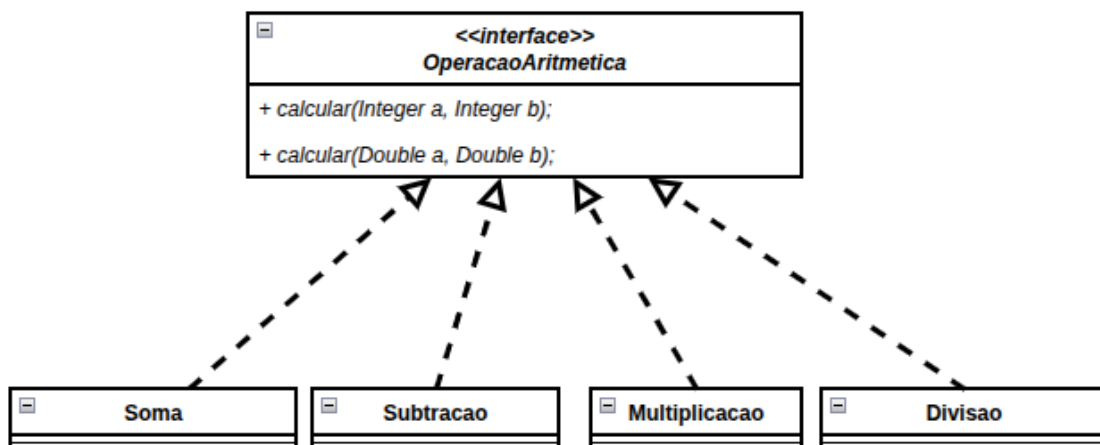
Interface

Cada aluno deverá, individualmente, resolver os exercícios abaixo:

Orientações:

- Para cada exercício faça:
 - a) Elabore o diagrama de classes utilizando UML quando o diagrama não for dado;
 - b) Implemente ou complemente o código em Java quando o código não for dado;
 - c) Implemente um aplicativo específico para demonstração dos códigos.

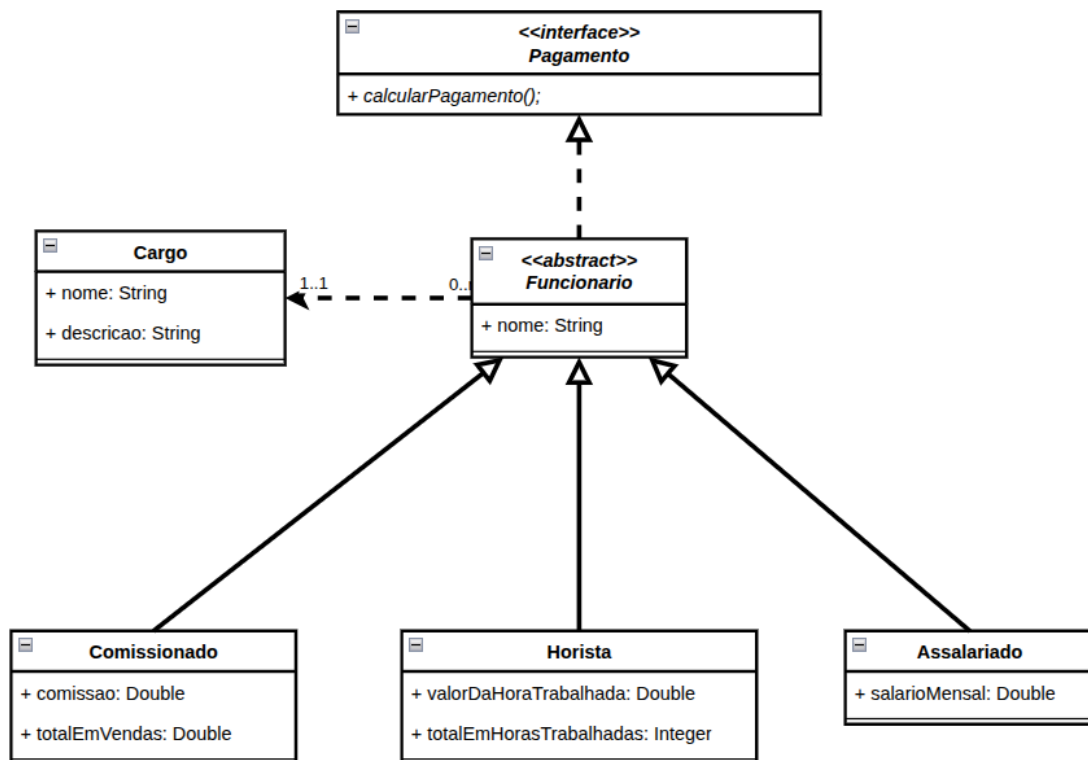
1. Operações Aritméticas



Para demonstrar sua solução faça:

- a. Complemente a UML se necessário.
- b. Implemente o código Java para a UML.
- c. Implemente um aplicativo que instancie um objeto de cada classe concreta. Demonstre o funcionamento do sistema.

2. Gerenciamentos de Salários



Considere um sistema de pagamento de funcionários de três categorias: a) Comissionado, b) Horista e c) Assalariado. Implemente uma solução que calcule o total a pagar para cada tipo de funcionário, conforme as características de sua categoria.

Para demonstrar sua solução faça:

- Complemente a UML se necessário.
- Implemente o código Java para a UML.
- Crie uma instância para cada categoria de funcionário
- Calcular o valor que ele deverá receber a partir de valores aleatórios definidos por você e escreva o resultado da seguinte maneira:
 - “O funcionário <NOME> com cargo <CARGO> deverá receber R\$ <TOTAL>.”, onde <NOME> é o valor do atributo

nome, <CARGO> é o valor do atributo nome e <TOTAL> é o valor calculado pelo método “*carcularPagamento()*”.

- e. Para cada classe concreta, execute todos os seus métodos e escreva o respectivo resultado.

3. Catálogo de Produtos para -E-commerce

Uma loja online vende diferentes tipos de produtos (Eletrônicos, Livros, Roupas) e precisa de um sistema para gerenciar o catálogo e calcular descontos.

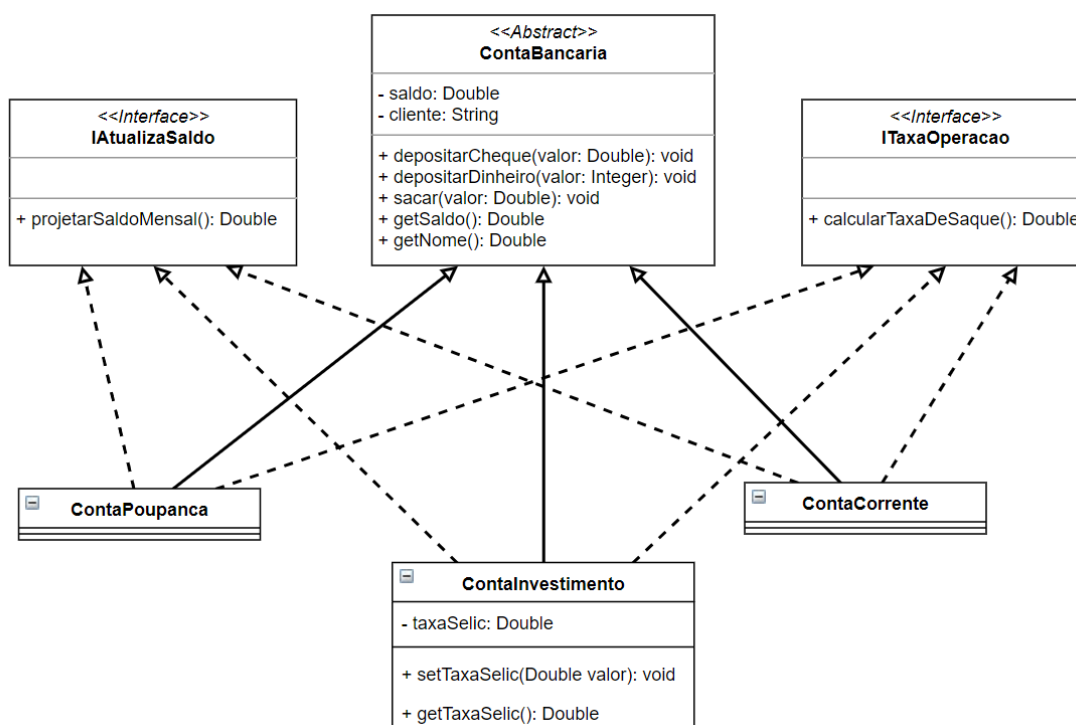
1. Crie uma **interface** Descontavel com o método `calcularDesconto()`.
2. Crie uma **classe abstrata** Produto com:
 - Atributos como nome, preço, categoria.
 - Método abstrato `exibirDetalhes()` que deve ser sobrescrito pelas subclasses.
3. Implemente **três classes concretas** que herdem de Produto:
 - Eletronico
 - Livro
 - Roupas
 - Cada uma deve ter comportamento específico para `exibirDetalhes()` e implementar (ou não) `Descontavel`, aplicando regras diferentes de desconto.
4. Utilize **polimorfismo** para mostrar os detalhes e calcular o desconto para cada tipo de produto.
5. Implemente um aplicativo que instancie um objeto de cada classe concreta. Demonstre o funcionamento do sistema.
6. Elabore a UML da solução.

4. Sistema de Jogo RPG em Cartas

Desenvolva um mini-sistema de jogo RPG em cartas em que cada carta possui um efeito diferente ao ser usada.

1. Crie uma **interface** `Jogavel` com o método `usarCarta()` que descreve o efeito da carta em um jogador.
2. Crie uma **classe abstrata** `Carta` que:
 - Tenha atributos como nome, custo de mana, descrição.
 - Método abstrato `efeito()` que descreve o que a carta faz.
 - Se necessário, implemente parcialmente `Jogavel`
 - i. Por exemplo, `usarCarta()` pode ser abstrato ou fornecer parte da lógica.
3. Crie **três classes concretas** que estendam `Carta`:
 - `CartaAtaque` (diminui pontos de vida do oponente).
 - `CartaDefesa` (aumenta a defesa do jogador).
 - `CartaCura` (recupera pontos de vida do jogador).
 - Cada uma deve implementar sua própria lógica de `efeito()` e `usarCarta()`.
4. Implemente um aplicativo que instancie um objeto de cada classe concreta. Demonstre o funcionamento do sistema. Simule uma pequena partida. Mostre como cada tipo de carta se comporta de maneira distinta ao ser usada.
5. Elabore a UML da solução.

5. Sistema Bancário



Considere que uma Instituição Bancária procurou então os maravilhosos alunos do 2º ano do curso TIII do IFG campus Luziânia para construir então um sistema para controlar as contas dos seus clientes. Logo, cada conta bancária pertence a um cliente e possui saldo da conta. É possível depositar e sacar valores de uma conta, sendo que, a ação depositar atualiza o saldo adicionando o valor depositado e a ação sacar atualiza o saldo subtraindo o valor do saldo. O saldo de uma conta nunca poderá ser alterado diretamente, somente por alguma operação bancária. No momento da criação de uma conta bancária o valor do saldo inicial, bem como o nome do cliente deverá ser informado. Toda conta deve permitir saber o valor atual do saldo. Essas contas bancárias podem ser de três tipos: **1) conta poupança, 2) conta corrente ou c) conta investimento**. Ambas as contas possuem uma atualização de saldo mensal, porém, essa atualização é diferente conforme o tipo da conta. Para:

a) Uma conta do tipo poupança, incrementa em seu saldo 1% do valor na atualização mensal, referente a rentabilidade por juros.

b) Já uma conta do tipo corrente, decrementa do seu saldo uma taxa de manutenção de conta no valor fixo de R\$ 15,00. Após a subtração da taxa de manutenção é incrementado no saldo restante o valor de 0,5% referente à rentabilidade por juros.

c) Uma conta do tipo investimento, incrementa em seu saldo, o valor da taxa Selic do mês.

Há diferenças também nas operações de saque conforme o tipo da conta:

a) Em uma conta do tipo corrente, cada operação de saque paga uma taxa de R\$ 0,10.

b) Em conta do tipo poupança, todas as operações são gratuitas.

c) Em conta do tipo investimento, cada operação de saque pague uma taxa de 0,38% referente a IOF (Imposto sobre Operações Financeiras).

Contas do tipo poupança e investimento, não podem permitir saldo negativo.

É importante mencionar que uma conta bancária, independente do seu tipo, permite operações de depósitos de duas maneiras: **1) dinheiro e 2) cheque**.

Depósitos em dinheiro são apenas em cédulas, isto é, o valor depositado sempre será um valor inteiro (sem centavos) e depósitos em cheques podem possuir centavos.

Para demonstrar sua solução faça:

- a) Implemente a UML acima e complemente-a se necessário.
- b) Implemente uma classe Aplicativo para simular o banco.
- c) Crie 3 contas, uma de cada tipo.
- d) Para cada conta criada, realize pelo menos uma operação de depósito, saque e atualização mensal.
- e) Após cada uma das operações do item d), exiba o valor do saldo atual da conta.

- f) Exiba o valor total no banco, isto é, a soma do saldo de todas as contas depois da execução do item c) e antes de encerrar o aplicativo, após a execução do item e).

Bom trabalho.