

Exercício 1:

Crie uma classe em Java que represente uma conta corrente. A conta corrente possui um número, o saldo, um status para informar se ela é especial ou não e um limite. Desenvolva pelo menos um construtor, gets e sets pertinentes e métodos para realizar as seguintes operações:

- a) **Sacar:** recebe um valor para saque como parâmetro e caso tenha saldo suficiente retorne true e subtraia o valor do saldo. Considere que o saldo de contas especiais tem valor somado ao limite, permitindo assim que o saldo fique negativo até o valor do limite.
- b) **Depositar:** recebe um valor e o adiciona ao saldo.
- c) **Consultar saldo:** retorna o saldo sem considerar o limite
- d) **Verificar Especial:** verifica se o cliente está usando cheque especial ou não, retornando true ou false

Crie as validações necessárias e desenvolva um programa para testar essa classe.

Exercício 2:

Crie a classe `Produto` com os atributos `nome` e `preco`. Adicione métodos para calcular o preço com desconto e outro para calcular o imposto sobre o preço do produto.

Passos:

- a) Implemente um construtor para inicializar os atributos.
- b) Adicione um método que aplica um desconto ao produto (recebe um percentual como parâmetro).
- c) Adicione um método que calcula e retorna o valor do imposto (o imposto é 10% do preço).

No Main() crie uma instância da classe `Produto`, utilize os métodos para calcular o imposto e dar o desconto e exiba os resultados no console.

Exercício 3:

Globo de Bingo. Desenvolva uma classe para representar um globo utilizado nos jogos de bingo. A classe Globo deverá fornecer através de um método, números aleatórios não repetidos entre 1 e um valor máximo definido na sua instanciação. No projeto da classe, defina construtores, métodos gets, sets e demais métodos necessários. Não permita o sorteio de números duplicados.

Exercício 4:

A linguagem Java dispõe de um suporte nativo a vetores, que exige a definição de seu tamanho no momento da instanciação. Depois de instanciado, o tamanho do vetor não pode ser modificado, sem que perca seus elementos. Escreva uma classe chamada `MeuVetor` cujos objetos simulem vetores de Strings de tamanho variável. A classe define os seguintes métodos:

construtor: recebe como parâmetro o tamanho inicial do vetor.

insert: recebe como parâmetro uma string e a coloca na próxima posição disponível do vetor; note que o vetor cresce automaticamente, portanto, se a inserção ultrapassar o tamanho inicial estabelecido na criação, o vetor deve aumentar seu tamanho automaticamente. Crie um algoritmo para redimensionar um vetor de forma a não perder seus valores.

get: recebe como parâmetro uma posição do vetor e retorna a string que estiver naquela posição; se a posição não estiver ocupada ou ultrapassar o tamanho do vetor, este método retorna nulo.

get: não recebe parâmetros, retorna o último elemento inserido no vetor, retorna nulo caso o vetor esteja vazio.

size: retorna o número de elementos inseridos no vetor (independente do tamanho físico do mesmo)