

Lista 04 - Introdução a POO

Crie um projeto Java chamado **lista4-resolucao** e para cada exercício crie um pacote(java package).

Exercício 01 - Confeitaria:

A confeitaria da Dona Maria, precisa de um sistema para gestão dos bolos vendidos em sua loja, que possibilite a geração de um relatório de vendas para cada tipo de bolo, para implementação do sistema em Java com OO (orientação a objetos), crie:

1. Uma classe Bolo que contenha os **atributos**:

- a) sabor (chocolate, morango ou abacaxi)
- b) valor (entre 30.00 e 50.00)
- c) quantidadeVendida (até 100 por dia)

A classe deve conter os seguintes **métodos**:

- d) comprarBolo: recebe a quantidade desejada do bolo, e exibe a frase "Seu pedido ultrapassou nosso limite diário para esse bolo." caso a quantidade seja maior que 100, caso contrário, a quantidade de bolos vendidos deve ser atualizada.
- e) exibirRelatorio: exibe a frase "O bolo sabor X, foi comprado Y vezes hoje, totalizando R\$ Z"

2. Crie uma classe executável chamada App, que:

- a) Crie 3 bolos, com preço e sabor de sua preferência,
- b) Compre pelo menos 5 vezes qualquer um dos bolos,
- c) Exiba o relatório de cada um dos bolos criados.

Exercício 02 - Encomendas:

Crie um programa para gerenciar as encomendas de uma transportadora, que gere etiquetas para envio, e calcule o frete.

1. Seu programa deve conter uma classe chamada **Encomenda**, que contenha os seguintes atributos:

- a) altura (Ex: 5.5)
- b) largura (Ex: 15.5)
- c) enderecoRemetente (Ex: Rua Santos da Glória, 18)
- d) enderecoDestinatario (Ex: Av Dr. Pedro, 255)
- e) distancia(Ex: 42.2)

f) valorEncomenda (Ex: 87.50)

A classe deve conter os seguintes **métodos**:

calcularFrete: o método deve calcular o frete de acordo com a distância e tamanho da encomenda, use as seguintes tabelas para o cálculo:

Tamanho	Altura	Largura	Frete
Pequeno	até 6cm	até 16cm	1% do valor do produto
Médio	7cm a 15cm	16cm a 50cm	3% do valor do produto
Grande	16cm ou mais	51 cm ou mais	5% do valor do produto

Distância	Preço
Até 50km	R\$ 3,00
51 a 200km	R\$ 5,00
mais de 200km	R\$ 7,00

emitirEtiqueta: ao ser invocado, o esse método deverá exibir as informações da encomenda, exemplo:

```
***** ETIQUETA PARA ENVIO *****
Endereço do remetente: xxxxxx
Endereço do destinatário: xxxxxx
Altura da encomenda: 00.0 cm
Largura da encomenda: 00.0cm
-----
Valor da encomenda: R$00.00
Valor do frete: R$00.00
-----
Valor total: R$00.00
```

Exercício 03 - Empregado:

Crie uma classe chamada **Empregado**
Crie os atributos **nome**, **cargo** e **salario**.

Crie um método chamado **reajustarSalario**, que receberá como parâmetro a porcentagem do reajuste, e atualiza o atributo **salario**.

Crie uma classe de teste chamada **TesteEmpregado**.

- Dentro do método **main**, instancie um objeto da classe **Empregado** que tenha como nome "João", salário inicial de R\$ 5.400,00 e cargo "Analista de Sistemas".
- Imprima na tela o nome, o cargo e o salário reajustado, sabendo que este empregado recebeu um reajuste salarial de 15%.
- Crie outro objeto da classe **Empregado** e forneça atributos de sua escolha. Depois imprima seus dados e também seu salário reajustado.

Exercício 04 - Termômetro:

Crie uma classe chamada **Termometro**.

- Crie os atributos **temperaturaAtual**, **temperaturaMax**, **temperaturaMin**.
- Crie um método chamado **aumentaTemperatura**, que recebe um valor para aumentar a temperatura atual. Caso a nova temperatura seja maior do que a **temperaturaMax**, atribua para a temperatura atual a **temperaturaMax**.
- Crie um método chamado **diminuiTemperatura**, que recebe um valor para diminuir a temperatura atual. Caso a nova temperatura seja menor do que a **temperaturaMin**, atribua para a temperatura atual a **temperaturaMin**.
- Crie um método chamado **exibeFahrenheit**, que calcula e exibe a temperatura atual na escala de Fahrenheit. Pesquise como é a fórmula para converter de Celsius para Fahrenheit.
- Crie uma outra classe chamada **TesteTermometro**, crie uma instância e seus execute os métodos para testar sua classe recém criada.

Exercício 05 - Recursos Humanos:

1. Crie uma classe chamada Colaborador
 - a) Deve conter nome, cargo e salario.
2. Crie uma classe chamada RecursosHumanos
 - a) Deve conter total de promovidos e total de salarios reajustados.
 - b) Deve conter um método reajustarSalario, que recebe um **colaborador** e o valor do reajuste(utilize Double).
 - c) Deve conter um método promoverColaborador, que recebe um colaborador, novo cargo e um novo salario.
 - d) Somente deverá ser realizada a promoção caso o novo salario informado seja maior que o salario atual do funcionario, caso não seja, exiba "operação inválida".
3. Crie uma classe chamada TesteColaborador
 - a) Crie dois objetos do tipo Colaborador.
 - b) Crie um objeto do tipo RecursosHumanos.
 - c) Exiba as informações dos colaboradores.
 - d) Promova um colaborador com salário maior.
 - e) Promova um colaborador com salário menor.
 - f) Faça o reajuste de salário de um colaborador.
 - g) Exiba novamente as informações dos colaboradores.
 - h) Exiba o total de promovidos.
 - i) Exiba o total de salários reajustados.