

# Exercício: Gerenciamento de Pedidos em Loja Virtual com Herança e Polimorfismo

## Objetivo:

Desenvolver um sistema de gerenciamento de pedidos em uma loja virtual utilizando conceitos de herança e polimorfismo em Java.

## Descrição:

Crie as seguintes classes e interfaces:

### 1. Interface Pedido:

- Define os métodos básicos para um pedido, como:
  - `obterId()`: Retorna o identificador único do pedido.
  - `obterCliente()`: Retorna o cliente que realizou o pedido.
  - `obterDataPedido()`: Retorna a data em que o pedido foi realizado.
  - `obterItens()`: Retorna uma lista de itens do pedido.
  - `calcularValorTotal()`: Calcula e retorna o valor total do pedido, considerando preços dos itens e frete.
  - `aplicarDesconto(double desconto)`: Aplica um desconto percentual no valor total do pedido.

### 2. Classe PedidoSimples :

- Representa um pedido simples com frete fixo.
- Implementa o método `calcularValorTotal()` para calcular o valor total considerando o preço dos itens, frete fixo e desconto (se aplicado).

### 3. Classe PedidoExpress :

- Representa um pedido expresso com frete diferenciado.
- Implementa o método `calcularValorTotal()` para calcular o valor total considerando o preço dos itens, frete expresso e desconto (se aplicado).

### 4. Classe ItemPedido:

- Representa um item em um pedido, com atributos como:
  - `produto`: O produto adquirido.
  - `quantidade`: A quantidade do produto.
  - `precoUnitario`: O preço unitário do produto.

### 5. Classe Produto:

- Representa um produto à venda na loja virtual, com atributos como:

- nome: O nome do produto.
- descricao: Uma breve descrição do produto.
- preco: O preço unitário do produto.

## 6. Classe Cliente:

- Representa um cliente da loja virtual, com atributos como:
  - nome: O nome do cliente.
  - endereco: O endereço de entrega do cliente.

## 7. Classe Loja:

- Contém uma lista de produtos disponíveis para venda e métodos para:
  - adicionarProduto(Produto produto): Adiciona um novo produto à lista de produtos da loja.
  - removerProduto(Produto produto): Remove um produto da lista de produtos da loja.
  - criarPedido(Cliente cliente, List<ItemPedido> itens): Cria um novo pedido com base no cliente e nos itens selecionados.
  - listarPedidos(): Retorna uma lista com todos os pedidos realizados.
  - obterPedidoPorId(int idPedido): Retorna um pedido específico por meio do seu identificador.
  - calcularFaturamentoTotal(): Calcula e retorna o faturamento total da loja, considerando o valor total de todos os pedidos.

## 8. Classe Principal:

- Cria instâncias das classes Produto, Cliente, Loja e Pedido.
- Adiciona produtos à loja.
- Cria um pedido simples e um pedido expresso com diferentes itens e clientes.
- Calcula o valor total de cada pedido e aplica descontos.
- Exibe informações sobre os pedidos e o faturamento total da loja.

## Orientações:

- Utilize herança para modelar as classes PedidoSimples e PedidoExpress, que herdam da interface Pedido e compartilham características comuns.
- Implemente o polimorfismo no método calcularValorTotal() para que cada tipo de pedido utilize a lógica de cálculo de frete adequada.
- Utilize agregação nas classes Pedido e Loja para gerenciar as relações entre objetos.
- Teste as classes e interfaces criadas para garantir seu funcionamento correto.

## Observações:

- O sistema pode ser expandido para incluir funcionalidades como:
  - Gerenciamento de estoque de produtos.
  - Notificações de status do pedido para o cliente.

- Integração com sistemas de pagamento e entrega.
  - Geração de relatórios de vendas.
- As classes podem ser aprimoradas com métodos adicionais para atender às necessidades específicas do negócio.
- O exercício pode ser adaptado para outros cenários de gerenciamento de pedidos, como em restaurantes, serviços ou plataformas de e-commerce.
-