

Parte 1: Questões Conceituais

1. Padrão Creator: Na nossa atividade, por que a classe Venda é a candidata ideal para criar instâncias de ItemVenda? Qual é a regra fundamental do Creator que justifica essa escolha?

-A classe Vendas representa o geral, o todo que contém o itemvenda.

2. Padrão Information Expert: Qual é o objetivo principal do padrão Information Expert? Em termos simples, a que classe ele atribui uma responsabilidade (método)?

-determinar qual classe deve ser responsável por realizar uma determinada ação ou tarefa à classe que detém a maioria das informações essenciais.

3. Encapsulamento e Expert: Se tivéssemos colocado a lógica de cálculo do subtotal do item diretamente na classe Venda (em vez de em ItemVenda), que informação a Venda precisaria acessar do ItemVenda? E por que isso seria uma violação de boas práticas de orientação a objetos, como o encapsulamento?

-O ItemVenda seria forçado a expor publicamente seus atributos internos (quantidade e produto) através de métodos getters (como getQuantidade() e getProduto()).

4. Colaboração de Experts: Para o cálculo do total geral da venda, a classe Venda agiu como o Expert principal. Ela executou o cálculo sozinha ou colaborou com outra classe? Se sim, qual? E que método foi usado nessa colaboração?

-O cálculo do total requer a soma dos subtotais individuais, e a Venda não possui os dados brutos (preço e quantidade) para calcular o subtotal de cada item, utilizando o metodo calcularSubtotal().

5. Coesão e Acoplamento: O uso do Information Expert e Creator ajuda a promover a Alta Coesão e o Baixo Acoplamento no código. Explique brevemente como colocar o método `calcularSubtotal()` dentro de `ItemVenda` (Expert) contribui para o Baixo Acoplamento.

-Baixo acoplamento, pois foi atribuída a classe com todas as informações necessárias para executá-las.

6. Multiplicidade UML: No diagrama de classes, o relacionamento entre `Venda` e `ItemVenda` é marcado com "1 *-- *". O que o símbolo do diamante preenchido (*) representa em termos de relacionamento de classes?

- Relação de composição, indica que a classe `venda` possui sem restrições a classe `itemvenda`.

7. Propósito do GRASP: Qual é a função principal dos padrões GRASP (como Creator e Expert) em um projeto de software orientado a objetos?

-Servir como ferramenta para auxiliar na atribuição de responsabilidade às classes.

8. Information Expert vs. Dados: A classe `Produto` armazena o preço. Por que ela não foi escolhida como o Expert para calcular o subtotal, já que ela tem uma informação crucial (o preço)? Que informação essencial faltava a ela?

-Informação que faltava é a quantidade que está no `itemvenda`, o `itemvenda` foi escolhido como expert, pois ele possui acesso à informação da quantidade e `getpreçounitario`, pois a classe está ligada ao produto.