**1. O que é um relacionamento bidirecional em um diagrama UML?**

Um relacionamento bidirecional significa navegável nos dois sentidos. Para que isso seja possível, precisamos de uma coleção de objetos de uma classe A em uma classe B e vice-versa. Além disso, precisamos manter a consistência do relacionamento. Para tal, se um objeto Aa aponta para Bb, Bb também deve apontar para Aa. Caso Aa deixe de apontar para Bb, Bb também deverá parar de apontar para Aa, ou seja, bidirecional. Além disso, a operação deve ser atômica e precisamos de uma condição de saída para não gerar um loop infinito na chamada de A para B e posteriormente de B para A.

**2. Como um relacionamento bidirecional pode ser implementado em Java? Descreva para um relacionamento 1 para 1, 1 para n e n para n (onde n = \*).**

1 <-> 1: Um objeto A apontaria para um único objeto B e vice-versa. (Ex: Casamento).

1 <-> n: Um objeto A apontaria para uma lista de objetos B e um objeto B apontaria para um único objeto A. (Ex: Pai e filhos).

n <-> n: Um objeto A apontaria para uma lista de objetos B e um objeto B apontaria para uma lista de objetos A. (Ex: Alunos e turma).

**3. Explique operação atômica (transação)? Qual a sua necessidade?**

É uma operação que só pode ser realizada por completo. Ou seja, caso haja uma exceção no meio de sua execução, ela deverá retornar para seu estado inicial. Não há como executar apenas uma parte de uma operação atômica. Em relação a sua necessidade, ela pode ser vista como essencial, por exemplo, em transações financeiras. Caso haja uma transferência de A para B, a operação atômica garantirá que o dinheiro fique, ou com A (caso haja alguma exceção) ou com B (caso haja sucesso na transferência). Isto é, não há a possibilidade do dinheiro que está sendo transferindo acabar em um limbo entre A e B.

**4. O que é uma classe de associação? Como ela é representada em UML e em Java?**

É uma classe que representa a ligação entre duas outras classes. Ela é tanto uma classe quanto uma associação. No exemplo do roteiro, nota não seria um atributo exclusivo nem de Class nem de Student. Nota seria um atributo da relação entre elas. Sendo assim, a utilização de uma classe de associação é necessária apenas quando a associação precisa de atributos para defini-la.

Ela é representada em UML, em visão lógica, por uma linha pontilhada que provém da linha (de associação) que está ligando as duas classes associadas. Uma classe de associação é uma classe que pertence mutualmente às duas classes que estão ligadas entre si. Em Java, ela possuiria duas variáveis essenciais, que apontariam para as classes associadas. Assim como as classes associadas apontariam para a classe de associação (operação atômica).

**5. Mostre um exemplo (diferente daquele apresentado pelo roteiro) para uma classe de associação (visão lógica) e mostre como seria sua representação em uma visão física (ainda em UML).**

Um Medicamento trata de uma certa Doença. Nesse caso, “tratar” seria a relação, e a “Eficácia” do tratamento poderia ser a classe de associação. Visto que um Medicamento tem diferentes eficácias dependendo da Doença. Cada instância do objeto Eficácia associaria um Medicamento com uma Doença. Um objeto Doença pode estar associado a vários objetos Eficácia, assim como um objeto Medicamento pode estar associado a vários objetos Eficácia.

Em visão lógica seria apenas incluir a linha pontilhada que provém da linha de associação entre Medicamento e Doença. Já na representação física deve representar como, fisicamente, ela seria implementada. Ou seja, tanto Medicamento quanto Doença estariam ligados a Eficácia, com suas respectivas cardinalidades.

**6. Modele em UML e implemente as classes em Java para representar pedidos de compra em uma loja de material de construção. Um pedido de compra está relacionado com o cliente, o vendedor e uma lista dos produtos com a sua respectiva quantidade. Um mesmo produto não poderia aparecer duas vezes em um mesmo pedido, pois a quantidade deve controlar quantos itens daquele produto estão sendo pedidos. Um produto, além de sua descrição, também possui a sua quantidade em estoque e seu preço. A sua solução não deve permitir que um pedido marque uma quantidade maior do que aquela disponível em estoque.**

**7. A partir do exercício anterior, modele em UML e implemente um sistema para o setor atendimento de uma loja de material de construção. Utilize separação em camadas e repositórios.**

**8. Continue a modelagem e implementação do nosso sistema acadêmico, considerando a interface com o usuário. Utilize as boas práticas. Pense em um cenário similar ao da Univali, onde o professor poderia cadastrar várias avaliações, cada uma com seu próprio peso. Depois, cada aluno teria uma nota em cada uma das avaliações. A nota de um aluno na turma seria a média ponderada das avaliações.**