



Estácio

Missão prática 3: Back-end Sem Banco Não Tem

Gabriel Gonçalves Silva Costa - 202308025571

Campus Virtual

Nível 3: Back-end Sem Banco Não Tem - turma 9001, período: 2024.3

OBS: O template fornecido é praticamente IMPOSSÍVEL de ser editado, logo fiz esse aqui no canvas.

Parte 1:

a) Reposta: Sem o middleware, seria necessário criar um módulo de troca de dados para cada componente de software que se conecta à aplicação.

b) Resposta: PreparedStatement é uma interface mais especializada de Statement, logo, lhe permite maior flexibilidade ao interagir com o SGBD. Com um objeto do tipo Statement, por exemplo, você pode enviar uma SQL para o SGBD fazer o processamento, porém sua SQL já tem que estar "pronta", com os valores embutidos nela.

Já com um objeto PreparedStatement, você pode parametrizar os valores em sua SQL.

c) Reposta: Este padrão permite que possamos mudar a forma de persistência sem que isso influencie em nada na lógica de negócio, além de tornar nossas classes mais legíveis.

d) Resposta: Eu usei tabela por classe, então a estrutura é mais limpa, com poucos campos nulos e cada tabela é especializada para uma classe específica. Como contraponto as consultas podem ser mais complexas, pois requerem joins entre as tabelas.

Parte 2:

a) Resposta: Persistência em arquivos é simples e flexível, mas pode ser lenta e menos organizada, ideal para dados pequenos. Já a persistência em bancos de dados oferece estrutura, consultas rápidas, integridade e escalabilidade, sendo mais adequada para sistemas complexos e de grande volume. A segurança e a manutenção também são mais robustas em bancos de dados, embora possam ter custos mais altos.

b) Resposta: O uso de operadores lambda no Java simplificou a impressão de valores ao permitir que expressões anônimas sejam usadas em lugar de classes internas, tornando o código mais conciso e legível. Com lambdas, você pode facilmente passar funções como parâmetros, especialmente ao trabalhar com streams e coleções. Isso reduz a quantidade de código boilerplate, facilitando a manipulação e a iteração sobre entidades. Assim, a impressão de valores se torna mais direta e expressiva.

c) Resposta: Métodos acionados diretamente pelo main precisam ser marcados como static porque o método main é estático e pertence à classe, não a uma instância dela. Isso significa que você não pode chamar métodos não estáticos (que pertencem a instâncias) sem criar um objeto da classe. Ao marcar um método como static, ele pode ser chamado diretamente dentro do contexto estático do main, facilitando a execução de lógica sem a necessidade de instanciar a classe.

Testes:



