

Relatório de Prática de Desenvolvimento em Java

Disciplina: RPG0016 - Desenvolvimento de Aplicações Backend

Nome do Aluno: Aliffer Farias

Turma: 2023.1

1. Desenvolvimento de Aplicativo Java com Persistência em Banco de Dados SQL Server

Nesta prática, desenvolvemos um aplicativo Java para realizar operações de cadastro com persistência em um banco de dados SQL Server. Utilizamos o middleware JDBC para a comunicação com o banco de dados e implementamos o padrão DAO (Data Access Object) para o manuseio dos dados.

2. Objetivo da Prática

Implementar persistência utilizando o middleware JDBC.

Utilizar o padrão DAO para manipulação de dados.

Desenvolver um sistema cadastral com persistência em banco relacional.

Criar um aplicativo que realiza operações CRUD (Create, Read, Update, Delete) com dados armazenados no SQL Server.

3. Resultados da Execução dos Códigos

Figura 1: Inclusão e alteração de Pessoa Física na classe CadastroBDTeste.

Figura 2: Exclusão de Pessoa Física, inclusão e alteração de Pessoa Jurídica na classe CadastroBDTeste.

Figura 3: Exclusão de Pessoa Jurídica na classe CadastroBDTeste.

Figura 4: Inclusão de pessoa na classe CadastroBD.

Figura 5: Confirmação de inclusão bem-sucedida de Pessoa.

Figura 6: Alteração de pessoa na classe CadastroBD.

Figura 7: Confirmação de alteração bem-sucedida de Pessoa.

Figura 8: Exclusão de pessoa na classe CadastroBD.

Figura 9: Busca por ID na classe CadastroBD.

Figura 10: Exibição de todas as pessoas na classe CadastroBD.

4. Análise e Conclusão

(a) Importância dos Componentes de Middleware, como o JDBC:

O JDBC desempenha um papel crucial na comunicação entre aplicativos Java e bancos de dados relacionais. Ele fornece uma interface padrão para acessar diferentes sistemas de gerenciamento de banco de dados, promovendo assim a portabilidade e interoperabilidade das aplicações.

(b) Diferença no Uso de Statement ou PreparedStatement:

O PreparedStatement é preferível ao Statement devido à sua segurança aprimorada contra ataques de injeção SQL e melhor desempenho, especialmente em consultas repetidas com diferentes parâmetros. Além disso, o PreparedStatement torna o código mais legível e fácil de manter.

(c) Melhoria da Manutenibilidade com o Padrão DAO:

O padrão DAO melhora a manutenibilidade do software ao separar a lógica de acesso a dados do restante da aplicação. Isso facilita a realização de alterações na camada de acesso a dados sem afetar outras partes do sistema, promovendo reutilização de código e legibilidade.

(d) Reflexo da Herança em Modelos Estritamente Relacionais:

Em modelos estritamente relacionais, a herança é geralmente representada usando tabelas de junção, onde uma tabela base contém atributos comuns e tabelas de subclasse contêm atributos específicos. Isso permite representar hierarquias de herança de forma eficiente, mantendo a integridade referencial.

5. Desenvolvimento da Base de Dados no SQL Server

1. Objetivo da Prática

Identificar requisitos de um sistema e transformá-los em modelo de banco de dados.

Utilizar ferramentas de modelagem para criar estruturas de banco de dados relacionais.

Explorar a sintaxe SQL para criação de estruturas (DDL) e manipulação de dados (DML) no SQL Server.

2. Resultados da Execução dos Códigos

Figura 1: Inclusão de dados na classe CadastroBD.

Figura 2: Confirmação de inclusão bem-sucedida de Pessoa.

Figura 3: Alteração de dados na classe CadastroBD.

Figura 4: Confirmação de alteração bem-sucedida de Pessoa.

Figura 5: Exclusão de dados na classe CadastroBD.

Figura 6: Busca por ID na classe CadastroBD.

Figura 7: Exibição de todas as pessoas na classe CadastroBD.

3. Análise e Conclusão

(a) Diferenças entre Persistência em Arquivo e Banco de Dados:

A persistência em arquivo armazena dados diretamente em sistemas de arquivos, enquanto a persistência em banco de dados utiliza SGBDs para gerenciar dados de forma estruturada. Banco de dados oferece recursos avançados, como transações, segurança e escalabilidade, em comparação com a persistência em arquivo.

(b) Simplificação da Impressão de Valores com Operador Lambda:

O uso de operadores lambda simplifica a impressão de valores em coleções, tornando o código mais legível e expressivo, eliminando a necessidade de loops explícitos.

(c) Necessidade de Marcação como Static em Métodos Acionados pelo Main:

Métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static para permitir sua execução sem a necessidade de instanciação de objetos da classe. Isso facilita a chamada direta do método main sem a criação de instâncias da classe.

Conclusão:

Essas práticas proporcionaram uma compreensão mais profunda do desenvolvimento de aplicativos Java com acesso a banco de dados SQL Server, além de destacar a importância do padrão DAO, do JDBC e das boas práticas de modelagem de banco de dados.