Campus: Polo Taguatinga Sul

Curso: **Desenvolvimento Full Stack**

Disciplina: Back-end Sem Banco Não Tem (RPG0016)

Turma: 9001

Semestre: 3º Semestre

Integrantes: Yan Silva Sales

Mapeamento Objeto-Relacional e DAO

1. Implementar persistência com base no middleware JDBC.
2. Utilizar o padrão DAO (Data Access Object) no manuseio de dados.
3. Implementar o mapeamento objeto-relacional em sistemas Java.
4. Criar sistemas cadastrais com persistência em banco relacional.
6. SQL Server na persistência de dados.

Introdução

O modelo proposto descreve a estrutura e as operações de um sistema de cadastro de pessoas em um banco de dados. O documento aborda a criação de classes para representar pessoas físicas e jurídicas, além de classes para interação com o banco de dados, seguindo o padrão DAO (Data Access Object).

Definição das Classes Model:

São apresentadas as classes Pessoa, PessoaFisica e PessoaJuridica, com seus atributos, construtores, métodos getters e setters, além do método exibir para visualização dos dados no console. As classes PessoaFisica e PessoaJuridica herdam de Pessoa, demonstrando o conceito de herança.

Definição das Classes Utilitárias:

São apresentadas as classes utilitárias ConectorBD e SequenceManager, responsáveis pela conexão com o banco de dados e pela obtenção de valores de sequência, respectivamente. O documento detalha os métodos dessas classes e explica como são utilizadas para acessar o banco de dados.

Implementação das Classes DAO:

São detalhadas as classes PessoaFisicaDAO e PessoaJuridicaDAO, que seguem o padrão DAO para realizar operações de CRUD (Create, Read, Update, Delete) no banco de dados. Cada classe possui

métodos para inclusão, alteração, exclusão e consulta de pessoas físicas e jurídicas.

Classe Principal de Testes:

É apresentada a classe CadastroBDTeste, responsável por testar todas as operações do sistema. No método main, são realizadas operações como inclusão, alteração, consulta e exclusão de pessoas físicas e jurídicas no banco de dados.
Análise das Classes

Classes Model (Pessoa, PessoaFisica, PessoaJuridica):

As classes são bem estruturadas, com atributos, construtores, métodos getters e setters adequados. O método exibir permite visualizar os dados de uma pessoa no console de forma organizada.
A utilização de herança permite reutilizar e estender funcionalidades entre as classes PessoaFisica e PessoaJuridica.

Classes Utilitárias (ConectorBD, SequenceManager):

As classes utilitárias encapsulam a lógica de acesso ao banco de dados e à obtenção de valores de sequência, promovendo uma maior modularidade e reutilização de código.
Os métodos getConnection, getPrepared, getSelect e close na classe ConectorBD permitem a interação segura com o banco de dados, garantindo o fechamento adequado de recursos.

Classes DAO (PessoaFisicaDAO, PessoaJuridicaDAO):

As classes DAO implementam operações de CRUD de forma independente do tipo de banco de dados utilizado, promovendo a separação de responsabilidades e facilitando a manutenção do código.
O controle explícito de transações garante a consistência das operações realizadas no banco de dados, evitando alterações não desejadas em caso de erro.

Conclusão

O modelo proposto apresenta uma estrutura sólida e bem definida para um sistema de cadastro de pessoas em um banco de dados. A utilização de classes model, classes utilitárias e classes DAO permite uma implementação modular, coesa e de fácil manutenção. No entanto, é importante realizar testes abrangentes para validar o

funcionamento correto do sistema em diferentes cenários e ambientes de execução.