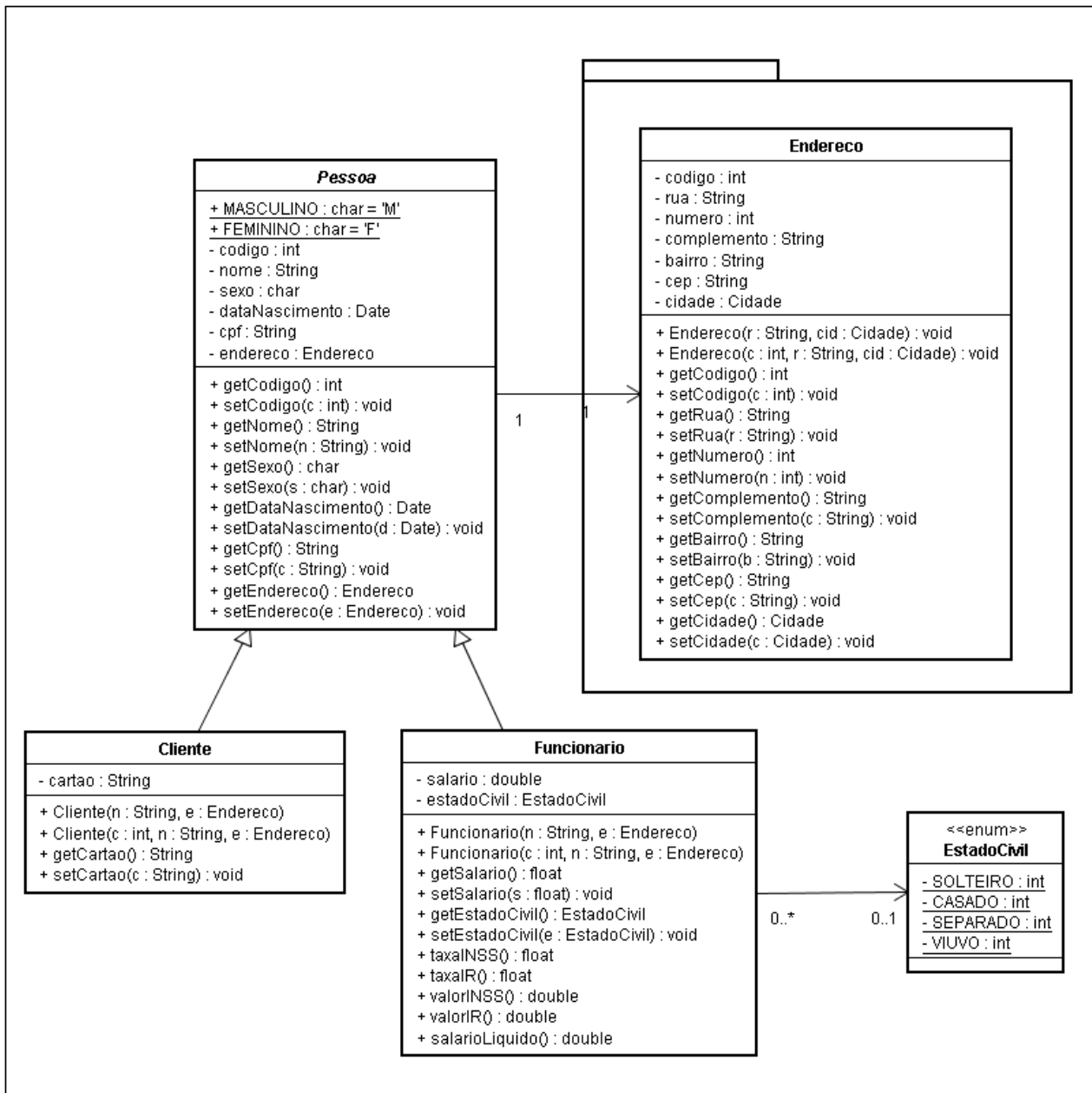


Exercícios em Java com JDBC e Utilizando Coleções



1) Utilizar parte do padrão MVC (caso sobre tempo. Se for utilizar MVC, instalar o plugin JiGloo, criar uma interface IView com os eventos para o controller). Criar os pacotes br.edu.qi.app, br.edu.qi.controller, br.edu.qi.dao, br.edu.qi.model, br.edu.qi.view. Uma classe de conexão padrão Singleton (chamada ConnectionManager), uma interface padrão para os métodos CRUD chamada IDao. Criar uma classe pai com o método padrão sql chamado executeSQL, de nome GenericDao. A classe GenericDao depende da classe de conexão. Criar uma Classe chamada ClienteDao que herda da GenericDao e implementa a interface IDao.

- Tabela: cliente(codigo integer not null primary key, nome varchar(50))

- Criar um arquivo de PROPRIEDADE , onde será definida as configuração da conexão com o Banco de Dados.

- Iniciar o teste a partir do mysql e depois o postgresql

Todos os pacotes dos próximos exercícios terá os pacotes br.edu.qi.model e br.edu.qi.view
Criar um programa chamado TesteColecoes

2)

- Criar a classe Cliente (pode ser copiada da lista de exercício 1);
- Criar uma classe chamada ClienteNegocio (pacote model). Com os métodos:
listarGeral(List obj) . O retorno são os dados para a impressão
alterarCliente(Cliente obj). void
- No método main do programa TesteColecoes, onde serão criados 4 objetos clientes e adicioná-los a um arraylist. Chamar o método listarGeral, criar outro objeto cliente e chamar o método alterarCliente e por fim o método listarGeral novamente.

3)

- Desenvolver classes e métodos iguais ao exercício anterior, mas em vez de cliente utilizar a classe Funcionario. Criar outro método chamado removerFuncionario que recebe a posição do elemento a ser removido ou o nome do funcionário a ser removido. Criar um método para adicionar funcionário chamado addFuncionario. Este método deve ter dois parâmetros: o primeiro é o objeto funcionário e o segundo é a posição onde desejamos adicioná-lo.
- No método main do programa TesteColecoes, criar 4 funcionários e chamar os métodos na mesma ordem do exercício 2 e por fim a rotina addFuncionario. Chamar novamente o método listarGeral.

4)

- Criar uma classe Endereco , conforme o diagrama de classe;
- Criar os seguintes métodos na classe Endereco:
- Criar uma classe chamada EnderecoNegocio (pacote model)
listarOrdenado(). O retorno são os dados ordenados;
- No método main do programa TesteColecoes, onde serão criados 4 objetos Endereco e adicioná-los a um coleção que permita ordenação. Chamar o método listarGeral

5)

- No método main do programa TesteColecoes, criar uma rotina chamada defineEnderPessoa que define um mapa com 4 objetos pessoas e os respectivos endereços e ao final listar todas as pessoas e os endereços