

| |
|--|
| <p>Classe RegistroIdentificavel</p> <p>Criar tal classe abstrata, que representa dados gerais que toda entidade possa ter, e a característica de que toda entidade é identificável por uma chave única, que será definida em cada uma das subclasses desta classe, através da implementação obrigatória de um método abstrato.</p> |
| <p>Atributos: por enquanto, nenhum.</p> |
| <p>Métodos:</p> <p><u>String obterChave()</u> – deve ser abstrato.</p> |
| <p>Alterar as classes Produto (do projeto de referência) e ContaCorrente</p> <p>Fazer as duas classes herdarem de RegistroIdentificavel, e nelas implementar o método obterChave(). A chave de um Produto é seu código, e a chave de uma ContaCorrente é seu número.</p> |
| <p>Classe DAOGenerico</p> <p>Criar tal classe, que terá uma implementação genérica para as operações básicas de um cadastro, baseando-se no tipo RegistroIdentificavel. Usar a mesma lógica de DAOProduto.</p> |
| <p>Atributos:</p> <p>cadastro, do tipo CadastroObjetos.</p> |
| <p>Métodos:</p> <p><u>Construtor</u> que recebe como parâmetro tipoEntidade, do tipo Class. Inicializar no construtor o atributo cadastro com um new CadastroObjetos(...), e passar para o construtor de CadastroObjeto o tipoEntidade recebido.</p> <p><u>boolean incluir(RegistroIdentificavel reg)</u></p> <p>Implementar seguindo o modelo do DAOProduto. A chave de reg é o retorno do método obterChave().</p> <p><u>boolean alterar(RegistroIdentificavel reg)</u></p> <p>Implementar seguindo o modelo do DAOProduto. A chave de reg é o retorno do método obterChave().</p> <p><u>boolean excluir(String chave)</u></p> <p>Implementar usando o método excluir do cadastro (atributo da classe).</p> <p><u>RegistroIdentificavel buscar(String chave)</u></p> <p>Implementar seguindo o modelo do DAOProduto.</p> <p><u>RegistroIdentificavel[] buscarTodos()</u></p> <p>Implementar seguindo o modelo do DAOProduto.</p> |
| <p>Alterar as classes DAOProduto (do projeto de referência) e DAOContaCorrente</p> <p>A “interface” destas classes devem ser mantidas, ou seja, as assinaturas dos seus métodos públicos devem permanecer exatamente como estão. A ideia é otimizar a implementação, evitando duplicação de código. Para isso, estas classes devem ter um atributo do tipo DAOGenerico. Ver modelo de implementação e exemplo com os facilitadores.</p> |