

Tópicos Abordados



- Desvantagens do JDBC
- Frameworks ORM
- O que é o Hibernate
- Entidades
- Configuração do Hibernate
- SessionFactory e Session
- Associações
- HQL

2

Desvantagens do JDBC



- JDBC permite a integração entre aplicações Java e banco de dados
 - A forma de uso é a mesma para todos os bancos de dados
- JDBC expõe a linguagem SQL ao programador
 - SQL nem sempre é padronizado
 - Às vezes, as queries são complexas de montar

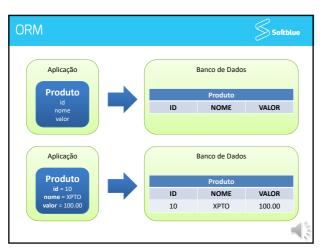
Frameworks ORM



- Em aplicações orientadas a objetos, normalmente são criadas classes para representar conceitos existentes na aplicação
- Estas classes são, no final das contas, mapeadas para tabelas do banco de dados, com o objetivo de persistir os dados
- Os frameworks do tipo <u>O</u>bject-<u>R</u>elational <u>M</u>apping facilitam este mapeamento



4



5

Hibernate



- O Hibernate é um framework ORM
- Gratuito e open source
- Site oficial
 - http://www.hibernate.org
- Pode ser utilizado em aplicações standalone e aplicações web



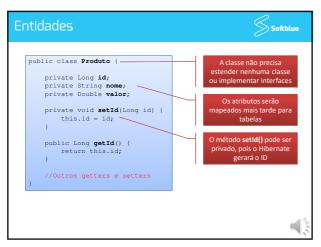
Entidades



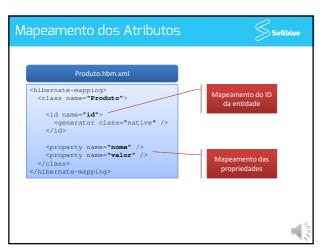
- Entidades são classes Java que serão mapeadas para tabelas no banco de dados
- As entidades são classes POJO
 - Devem ter uma ou mais propriedades que serão utilizadas como chave na tabela
 - Atributos e métodos getters e setters
 - O Hibernate consegue mapear atributos sem getters e/ou setters definidos, mas utilizá-los é mais adequado
 - Presença de um construtor sem argumentos



7



8



Configurando o Hibernate



- Para que o Hibernate funcione adequadamente, ele deve ser configurado
 - Os JARs necessários devem ser colocados no classpath
 - Criação dos arquivos .hbm.xml
 - Criação do arquivo hibernate.cfg.xml
 - Define como o Hibernate será integrado com o banco de dados da aplicação



10



11

SessionFactory



 A SessionFactory é responsável por criar os objetos Session

Configuration cfg = new Configuration().configure(); ServiceRegistry sr = new ServiceRegistryBuilder(). applySettings(cfg.getProperties()).buildServiceRegstry(); SessionFactory factory = cfg.buildSessionFactory(sr);

O ideal é usar apenas um objeto
 SessionFactory para toda a sua aplicação



Session

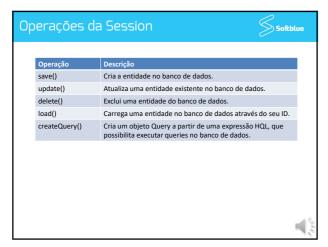


 A Session é utilizada para executar operações, que serão refletidas no banco de dados

Session session = factory.getCurrentSession();
session.beginTransaction();
Produto p = new Produto();
p.setNome("XPTO");
p.setValor(100.0);
session.save(p);
session.getTransaction().commit();

 A session pode ser utilizada para executar mais de uma operação

13

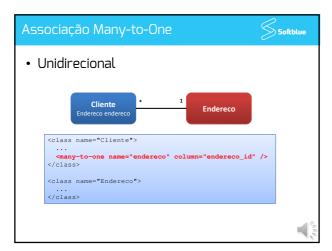


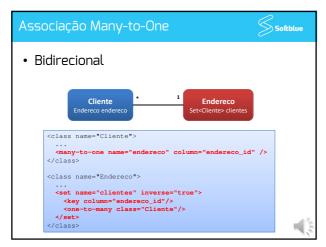
14

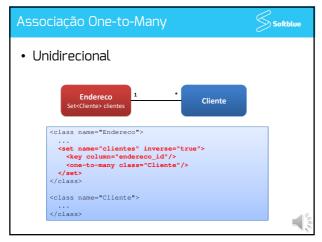
Associacões

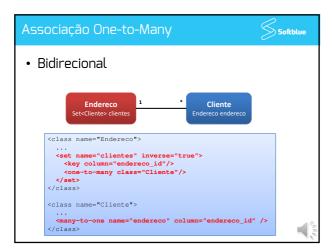


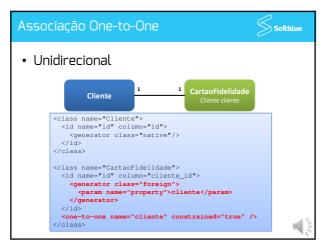
- O Hibernate gerencia também as associações entre entidades, escondendo a linguagem SQL usada no processo
- Tipos de associação
 - Many-to-One (Muitos-para-Um)
 - One-to-Many (Um-para-Muitos)
 - One-to-One (Um-para-Um)
 - Many-to-Many (Muitos-para-Muitos)
- Associações podem ser unidirecionais ou bidirecionais



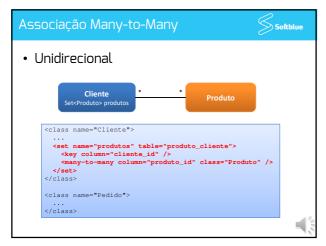


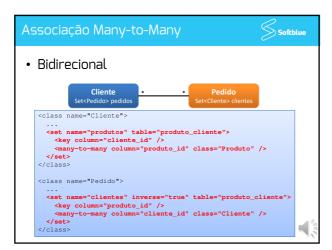


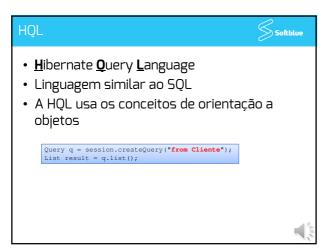


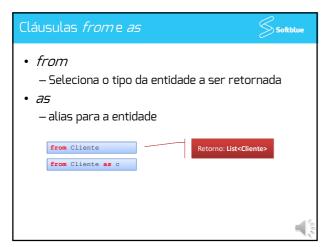


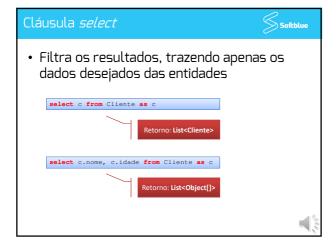






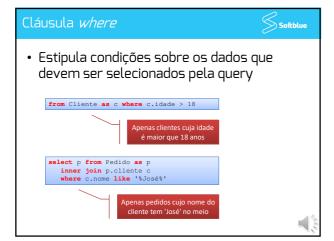


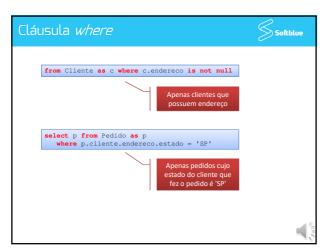


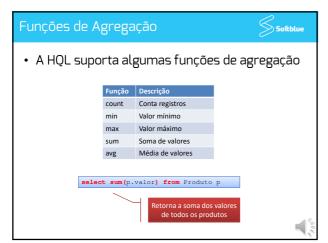












Permite escrever queries de forma orientada a objetos, sem pensar em termos de bancos de dados A HQL é bastante extensa, e o que foi visto aqui cobre o básico, que é o mais utilizado Para maiores informações, consulte a documentação do Hibernate