

Aula 11

Java Persistence API

Visão Geral

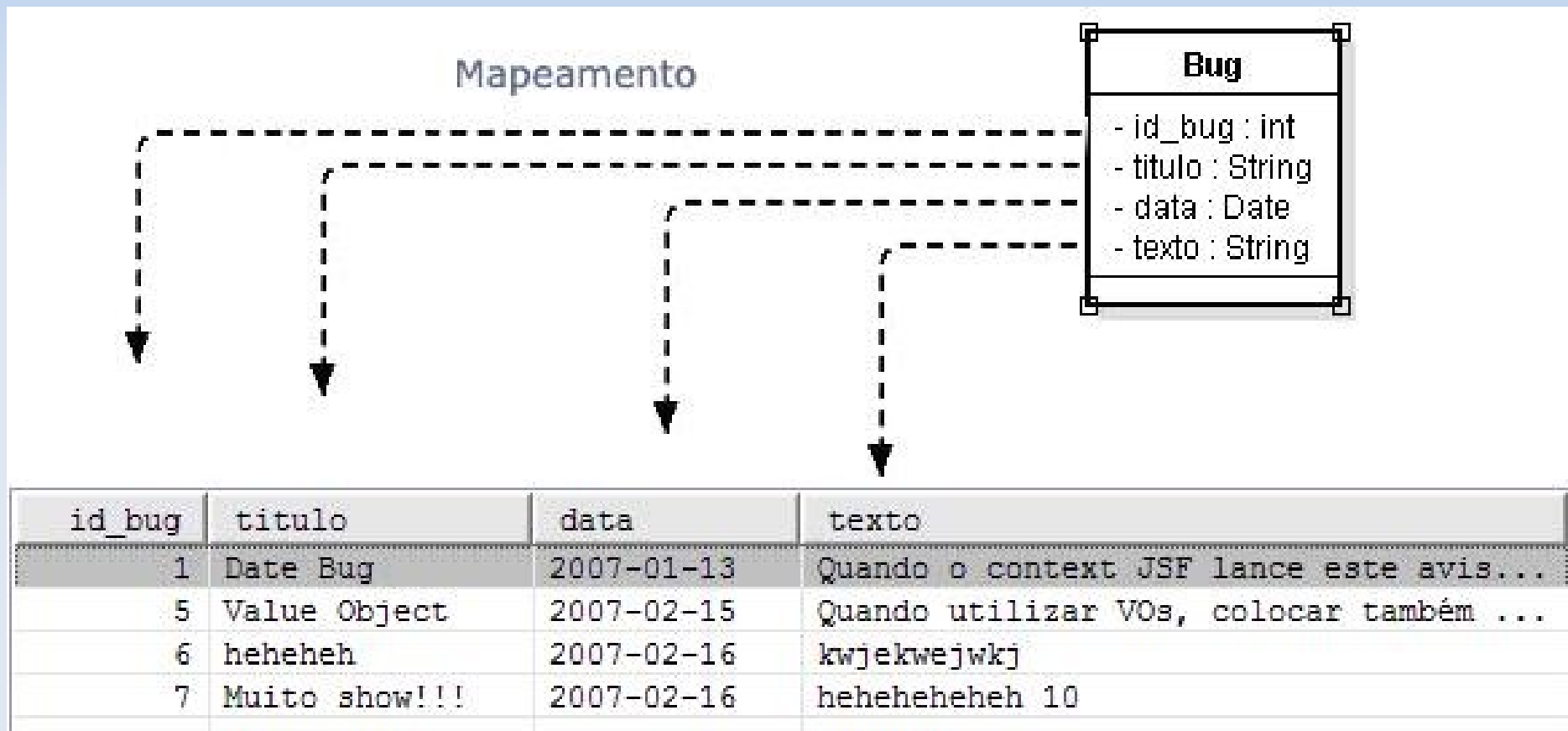
- O que é ORM
- JPA
- O EntityManager
- O persistence unit
- Tipos de Relacionamentos
- JPA-QL
- Configurações do Hibernate

O que é ORM

- Reduz o trabalho com SQL
- Automatiza a camada de persistência
- Facilita a programação orientada aos objetos utilizando bancos de dados relacionais

O que é ORM

- Exemplo



JPA

- JPA (Java Persistence API) faz o mapeamento objeto-relacional
- Traduz o mundo objetos do java para o mundo relacional dos bancos de dados
- Trata-se apenas de uma especificação
- A implementação fica por conta de terceiros

JPA

- Engines:
 - Hibernate (será utilizado nos exemplos)
 - TopLink
 - Kodo
 - etc

O EntityManager

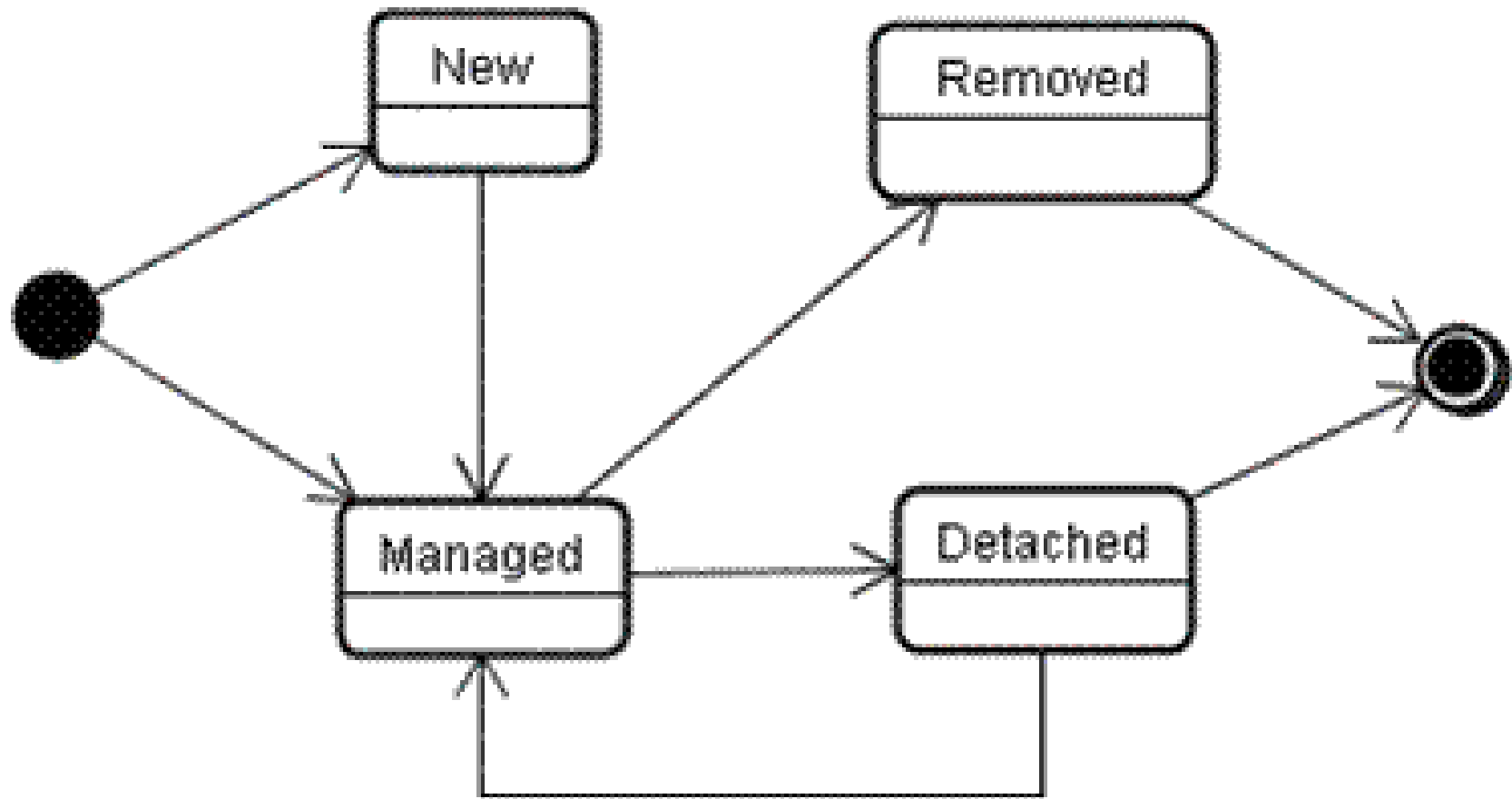
- É o gerente de entidades
- Responsável por manipular (persistir, alterar, consultar e excluir) os objetos
- Realiza o controle transacional

O EntityManager

- Estado das entidades
 - Novas (new)
 - Gerenciadas (Managed)
 - Não gerenciadas (Detached)
 - Removidas (Removed)

O EntityManager

- Estado das entidades



O EntityManager

- Observações
 - Só podem ser excluídas entidades gerenciadas
 - Uma entidade pode ser diretamente persistida
 - Para tornar uma entidade gerenciada é necessário utilizar o método merge.

O EntityManager

- Métodos do EntityManager (incluir e alterar):
 - void persist(Object entity) torna o entity gerenciado e persiste a instância no banco de dados quando a transação terminar.
 - T merge(T entity) faz com que um entity detached (desconectado) volte a ser gerenciado. Também pode ser utilizado para inserir novas entidades no banco de dados, mas comumente é utilizado para operações de alterações.

O EntityManager

- Métodos do EntityManager (excluir):
 - void remove(Object entity) remove o entity do banco de dados.
 - o entity deve estar no estado gerenciado.
 - Qualquer tentativa de remover um detached entity resultará num IllegalArgumentException,
 - Faça assim: `em.remove(em.merge(produto))`

O EntityManager

- Métodos do EntityManager (consultar):
 - A consulta de entities do banco de dados é realizada através de `T find(Class entityClass, Object primaryKey)`,
 - Retorna entidades gerenciadas
 - ou queries criadas com:
 - `Query.createNamedQuery(String query)`,
 - `Query.createNativeQuery(String sqlQuery)`
 - `Query.createQuery(String sqlQuery)`

O EntityManager

- Métodos do EntityManager (demais métodos):
 - void clear() encerra o persistence context. Como consequência, todos os entity beans tornam-se detached.
 - boolean contains(Object entity) determina se o persistence context contém um entity.
 - void close() fecha um entity manager de uma application-managed transaction. boolean isOpen() verifica se o persistence context está aberto.

O EntityManager

- Métodos do EntityManager (demais métodos):
 - void flush() força a sincronização do persistence context com o banco de dados.
 - void refresh(Object entity) atualiza o entity bean com o estado persistido no banco de dados, eliminando às mudanças feitas no bean, caso haja alguma.

O EntityManager

- Métodos do EntityManager (flush):
 - void flush() força a sincronização do persistence context com o banco de dados.
 - void setFlushMode(FlushModeType modeType) ajusta o flush mode aplicado aos objetos do persistence context.

O persistence unit

- Entity beans são empacotados em persistence units
- contém um arquivo descriptor denominado persistence.xml, no diretório META-INF.
- Um persistence unit define o conjunto de classes gerenciadas pelo entity manager e que representam as tabelas de um banco de dados.

O persistence unit

- Exemplo:

```
<persistence>
  <persistence-unit name="IBoi-Corp-ejbPU" transaction-type="JTA">
    <provider>oracle.toplink.essentials.ejb.cmp3.EntityManagerFactoryProvider
    </provider>
    <jta-data-source>jdbc/IBoi-Corp</jta-data-source>
    <properties>
      <property name="toplink.ddl-generation" value="create-tables"/>
    </properties>
  </persistence-unit>
</persistence>
```

O persistence unit

- Elementos do persistence unit
 - description: descrição do persistence unit
 - transaction-type: atributo de persistence-unit; conta com as opções JTA (padrão) e RESOURCE_LOCAL
 - non-jta-data-source: tem a mesma finalidade que jta-data-source

O persistence unit

- Elementos do persistence unit
 - mapping-file: a definição de mapeamento O/R pode ser feita diretamente nas classes entities via annotations, ou podem ser especificadas num arquivo XML denominado orm.xml, presente no mesmo diretório META-INF onde está o persistence.xml. mapping-file pode conter um ou mais desses arquivos de mapeamento O/R.
 - jar-file: define arquivos nos quais procurar por classes entities

O persistence unit

- Elementos do persistence unit
 - class: define as classes entidades presentes no persistence unit
 - exclude-unlisted-classes: se esse elemento for especificado, apenas arquivos e classes definidos em mapping-file, jar-file e class farão parte do persistence unit.

Tipos de Relacionamentos

- OneToOne
- OneToMany
- ManyToOne
- ManyToMany

Tipos de Relacionamentos

- OneToOne
 - Um para um
 - Unidirecional (@JoinColumn) e bidirecional (atributo mappedBy)

Tipos de Relacionamentos

- OneToMany
 - Um para muitos
 - Unidirecional (@JoinColumn) e bidirecional (atributo mappedBy)

Tipos de Relacionamentos

- ManyToOne
 - Muitos para um
 - Unidirecional (@JoinColumn)

Tipos de Relacionamentos

- ManyToMany
 - Muitos para muitos
 - Unidirecional e Bidirecional com @JoinTable

JPA-QL

- `SELECT revista FROM Magazine revista`
- `SELECT revista FROM Magazine revista where revista.nome = 'Java'`

JPA-QL

- Código:

```
EntityManager em = ...
```

```
Query q = em.createQuery ("SELECT x FROM Magazine x WHERE x.title = ?1 and x.price > ?2");
```

```
q.setParameter (1, "JDJ").setParameter (2, 5.0);
```

```
List<Magazine> results = q.getResultList ();
```

JPA-QL

- Código:
 - getResultList() - retorna uma lista de resultados
 - getSingleResult() - retorna apenas um resultado
 - Lança exceção quando nada é encontrado

Configurações do Hibernate

- `hibernate.connection.driver_class`
 - Classe do driver jdbc
- `hibernate.connection.url`
 - Url do banco de dados
- `hibernate.connection.username`
 - Usuário do banco de dados

Configurações do Hibernate

- `hibernate.connection.password`
 - Senha do banco de dados
- `hibernate.connection.pool_size`
 - Máximo de conexões do pool

Configurações do Hibernate

- **hibernate.dialect**
 - Ajusta o dialeto do banco de dados para otimizando para um banco de dados particular

RDBMS	Dialect
DB2	org.hibernate.dialect.DB2Dialect
DB2 AS/400	org.hibernate.dialect.DB2400Dialect
DB2 OS390	org.hibernate.dialect.DB2390Dialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
MySQL	org.hibernate.dialect.MySQLDialect
MySQL with InnoDB	org.hibernate.dialect.MySQLInnoDBDialect
MyISAM	org.hibernate.dialect.MySQLMyISAMDialect
Oracle (any version)	org.hibernate.dialect.OracleDialect
Oracle 9i/10g	org.hibernate.dialect.Oracle9Dialect
FrontBase	org.hibernate.dialect.FrontbaseDialect
Firebird	org.hibernate.dialect.FirebirdDialect

Configurações do Hibernate

RDBMS	Dialect
Sybase	org.hibernate.dialect.SybaseDialect
Sybase Anywhere	org.hibernate.dialect.SybaseAnywhereDialect
Microsoft SQL Server	org.hibernate.dialect.SQLServerDialect
SAP DB	org.hibernate.dialect.SAPDBDialect
Informix	org.hibernate.dialect.InformixDialect
HypersonicSQL	org.hibernate.dialect.HSQLDialect
Ingres	org.hibernate.dialect.IngresDialect
Progress	org.hibernate.dialect.ProgressDialect
Mckoi SQL	org.hibernate.dialect.MckoiDialect
Interbase	org.hibernate.dialect.InterbaseDialect
Pointbase	org.hibernate.dialect.PointbaseDialect

Configurações do Hibernate

- `hibernate.show_sql`
 - Escreve no console todo o sql gerado pelo hibernate
 - opções: `true` | `false`

Dúvidas

Fim