

WPA cvičení 5 - DB, JPA

Martin Ledvinka

10. listopadu 2015

Obsah

- 1 Dotazy k semestrálkám
- 2 DB a Java
- 3 JDBC
- 4 JPA
- 5 CP1 - požadavky
- 6 Literatura

Dotazy k semestrálkám

- Dotazy k semestrálkám?
- Checkpoint 1
- 20 bodů

Databáze v Javě

Java Database Connectivity (JDBC)

- Standard pro přístup k databázím,
- Veškeré operace s DB pomocí SQL,
- Nepříliš vhodné pro OO vývoj.

Java Persistence API (JPA)

- Objektově relační standard pro Javu,
- Umožňuje mapovat objekty na DB tabulky,
- Práce s objekty a high-level jazykem JPQL.

JDBC

Práce s databází v Javě pomocí JDBC

- parametry připojení k databázi - v kódu nebo pomocí property file (soubor je potřeba načíst programově)
 - url - protokol, port, database
 - username
 - password
- nativní dotaz - SQL
- ResultSet

JPA

- Java ORM standard,
- Současná verze – JPA 2.1 (JSR 338),
- "Configuration by exception",
 - Tj. konfigurace jen v případě nutnosti,
 - Většinou si vystačíte s defaultními hodnotami,
 - Často stačí k použití JPA jen 5-6 řádků v persistence.xml a anotace `@Entity` u tříd modelu.
- Implementace
 - **EclipseLink** – RI
 - Hibernate
 - Kodo(OpenJPA)

JPA

Práce s JPA

- Konfigurace připojení - obdobně jako u JDBC, ale v konfiguračním souboru
- Mapování relace (DB tabulky) a objektu
- Správa transakcí
- Generování databázového schéma z doménového modelu nebo naopak.

JPA - konfigurace

Čistá konfigurace je v META-INF/persistence.xml. Tento soubor obsahuje:

- Persistence unit *name*
- Typ transakčního zpracování - *transaction-type*
- Persistence *provider*
- (Seznam entity tříd)
 - *U většiny providerů již dnes není třeba*
- Nastavení připojení k DB a nastavení providera

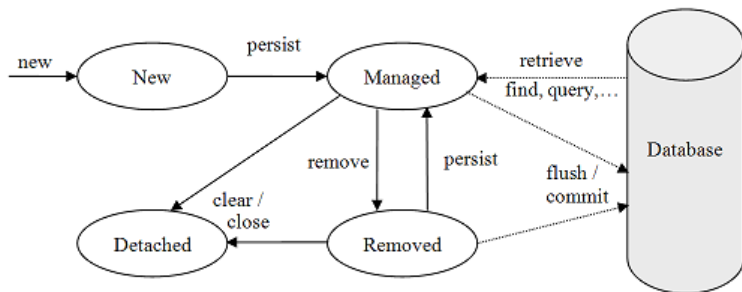
JPA - mapování pomocí anotací

- Mapování objektů na záznamy v DB tabulkách
 - `@Entity` - označí třídu jako entitu
 - `@Table` - nastavení odpovídající DB tabulky
- Mapování atributů
 - `@Id`
 - `@Column(name="col_name", unique=true/false, nullable=true/false)`

JPA - Vztahy

- @OneToOne (mappedBy=...), @JoinColumn
- @ManyToOne, @JoinColumn
- @OneToMany (mappedBy=...)
- @ManyToMany (mappedBy=...), @JoinTable

JPA - Životní cyklus entity



JPQL - Java Persistence Query Language

Jazyk je podobný SQL. Rozdíl je v tom, že pracuje s entitami a atributy.

- **SELECT**

- **alias** - `SELECT s FROM Student s <=> SELECT s FROM Student AS s`
- **atributy** - `SELECT s.name FROM Student s`

- **UPDATE**

- `UPDATE Account a SET a.deposit=a.deposit*1.01 where a.client.firstName='Josef'`

- **DELETE**

- `DELETE FROM User AS u WHERE u.username LIKE 'f%'`

JPQL cont.

- GROUP BY

- `SELECT a.client, sum(a.deposit) FROM Account a GROUP BY a.client`

- ORDER BY, HAVING, INNER JOIN, LEFT JOIN

- FETCH JOIN - vynucuje nahrání asociovaných entit

JPA - Podpora dědičnosti (Inheritance)

- Nadtřída není Entita
 - `@MappedSuperclass` - deklarace mapování je zděděna v podtřídách
- Nadtřída je Entita -
 - `@Inheritance` - používá se v nadtřídě. Parametr *strategy* určuje strategii generování tabulek
 - pouze jedna tabulka - `InheritanceType.SINGLE_TABLE`
 - každá třída = samostatná tabulka včetně zděděných atributů - `InheritanceType.TABLE_PER_CLASS`
 - `InheritanceType.JOINED` - každá třída má svou tabulku, parent třída též. Vyžaduje JOIN pro načítání parent atributů

Ordering (uspořádání kolekce podle atributu)

```
@Entity
public class Course {
    ...
    @ManyToMany
    // @OrderBy // ordering by primary key is
    // assumed
    @OrderBy("lastname ASC")
    // @OrderBy("zipcode.zip, zipcode.plusFour")
    public List<Student> getStudents() {...};
    ...
}
```

Pojmenované dotazy

```
@Entity
@NamedQuery(name="Country.findAll", query="
    SELECT c FROM Country c")
public class Country {
    ...
}
```


Kaskádní persist/update/merge/remove

Kaskádní persist

```
@Entity
class Employee {
    :
    @OneToOne(cascade=CascadeType.PERSIST)
    private Address address;
    :
}
```

Složené primární klíče

```
@Entity @IdClass(ProjectId.class)
public class Project {
    @Id int departmentId;
    @Id long projectId;
    :
}

Class ProjectId {
    int departmentId;
    long projectId;
}
```

Složené primární klíče

```
@Entity @IdClass(ProjectId.class)
public class Project {
    @Id int departmentId;
    @Id long projectId;
    :
}

Class ProjectId {
    int departmentId;
    long projectId;
}
```

Literatura

- WikiBooks – JPA

https://en.wikibooks.org/wiki/Java_Persistence

- JPA Step by Step – Albert Guo

[slideshare.net/junyuo/java-persistence-api-jpa-step-by-step-presentation](https://www.slideshare.net/junyuo/java-persistence-api-jpa-step-by-step-presentation)

- Pro JPA2 Mastering the Java™ Persistence API
– Mike Keith, Merrick Schincariol