



KIV/DB2 - Objektově relační mapování: Hibernate - model

Objektově relační mapování: Hibernate

Toto cvičení může sloužit jako návod jak pomocí objektově relačního mapování Hibernate navrhnout a ve vybraném databázovém systému vytvořit fyzický datový model včetně odpovídajících Java tříd a mapovacích souborů. Též se stručně seznámíme s konfigurací objektově relačního mapování. To vše bude realizováno automaticky, téměř bez napsání jakéhokoliv programového kódu, pluginem HiberObjects do vývojového prostředí Eclipse.

Proč objektově relační mapování

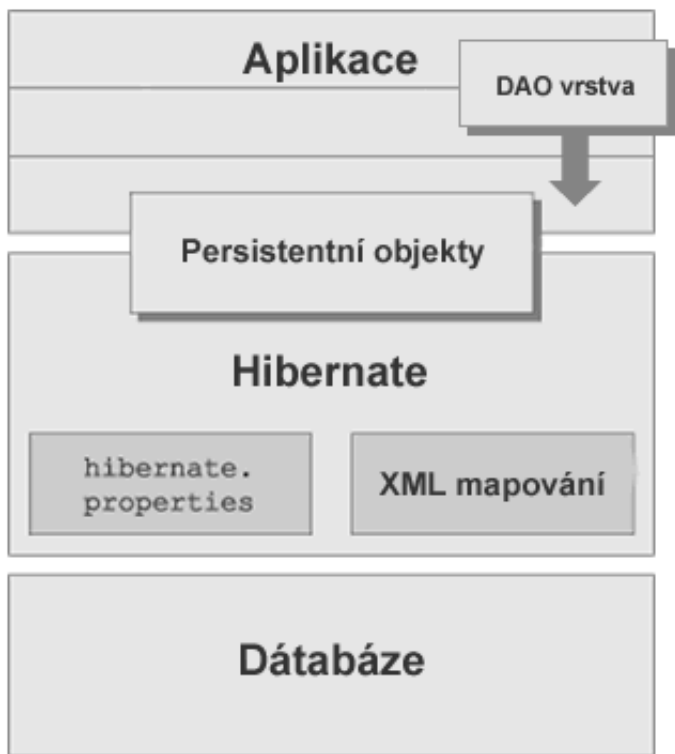
V současnosti je drtivá většina aplikací vyvíjena v objektově orientovaném stylu, ale data těchto aplikací jsou ukládána do relačních databází. To může přinášet problémy a je vhodné navrhnout takové mapování, aby jeden záznam tabulky byl reprezentován jako perzistentní (trvale uložený) objekt příslušné třídy. Tento proces může zabrat 50 až 70 procent času vývoje aplikace. Dalším problémem je také ta skutečnost, že dialekty jazyka SQL jednotlivých SRBD jsou nekompatibilní, i když by měly dodržovat příslušný standard.

Řešením uvedeného problému je použití objektově relačního mapování, např. framework Hibernate, který:

- zajišťuje mapování na prostředí programovacího jazyka Java,
- odstiňuje uživatele (i programátora) od konkrétní databáze,
- zajišťuje trvanlivost (perzistenci) datové vrstvy,
- mapuje tabulky na třídy, databázové relace na množiny, databázové atributy na java atributy.

Zajištění objektově relačního mapování

Použití frameworku Hibernate nevyžaduje žádné speciální rozhraní ani dědičnost. Každá třída, která odpovídá jedné tabulce v databázi, má shodné atributy, ke kterým se přistupuje metodami typu `get` a `set`. Třída dále obsahuje prázdný konstruktor a pro jednoznačnou identifikaci je definován atribut `id` typu `Long`. Informace, jaká třída odpovídá které tabulce v databázi, je uložena v příslušných mapovacích souborech ve formátu XML. K jednotlivým perzistentním objektům se přistupuje prostřednictvím *Data Access Objects* (DAO) vrstvy, která nabízí základní **CRUD** opearace s objekty (Create, Read, Update a Delete). Pro správný chod frameworku Hibernate je potřeba jej nakonfigurovat. Konfigurace je uložena buď v souboru `.properties` nebo v jiném XML souboru. V konfiguraci je nutné uvést základní spojení s databází, kde budou spravovaná data aplikací fyzicky uložena. Obecné schéma aplikace využívající framework Hibernate ukazuje tento obrázek:



Instalace vývojového prostředí Eclipse s pluginem HiberObjects

Vývojové prostředí Eclipse s integrovaným pluginem HiberObjects je ke stažení z tohoto URL: <http://www.kiv.zcu.cz/studies/predmety/db2/eclipse.zip>, které je uvedeno i na Coursewarových stránkách předmětu. Pro úplnost uvádím návod, jak plugin nainstalovat:

1. Stáhněte Eclipse ze stránek www.eclipse.org ze sekce *Downloads* - např. verzi *Eclipse Classic*.
2. Stažený archiv rozbalte na disk a z adresáře `eclipse` spusťte aplikaci. Zvolte/nechte pracovní adresář (*workspace*).
3. Vyberte nabídku *Help/Install new software* a poté stiskněte tlačítko *Add*.
4. Do položky *Location* vyplňte toto URL: `http://objectgeneration.com/update` a potvrďte tlačítkem *OK*.
5. Vyberte plugin HiberObjects (případně i DB Importer) a dále postupujte dle instrukcí instalačního průvodce.
6. Po dokončení instalace a restartu Eclipse můžete daný plugin používat.

pozn: návod je pro verzi Helios (3.6.2)

Práce s pluginem HiberObjects

Demonstrace použití frameworku Hibernate bude na jednoduchém projektu **Bar**. V Baru budeme nabízet zákazníkům *koktejly*, které bude třeba namíchat z příslušných *přísad*. Pro jednoduchost nebudeme evidovat množství těchto přísad, ani velikost požadovaného kotejlu.

Spusťte Eclipse a změňte perspektivu na *HiberObjects (Window/Open Perspective)*. Založte nový *HiberObjects Project*, který bude mít nastavenou perzistentní platformu na *Hibernate 3* a ponechte zaškrtnuté pole *Create Model Diagram*.

V příslušném UML editoru vytvořte dvě třídy: *Koktejl* a *Prisada*. Přidejte vhodné atributy, např. název a cena pro obě třídy. Plugin automaticky generuje Java kód příslušných tříd. Přidáním stereotypu *persistent* budou též automaticky generovány příslušné mapovací soubory. Každý perzistentní objekt získá unikátní `id` typu `Long`, které můžeme v návrhu také zobrazit. V automaticky generovaných mapovacích souborech jsou tabulky pojmenovány `tb_nazevtridy`. Pokud je chcete pojmenovat jinak,

je možné třídě přidat tag `@hibernate` s parametrem `table=nazev_tabulky`. Tento tag je možné použít i pro definici atributu třídy. Navržené třídy spojíme *obousměrnou asociací* a upravíme její typ na **M:N**. Všimněte si, jakým způsobem je tato asociace realizována ve zdrojovém kódu tříd i mapovacích souborech. Na závěr v editoru pro každou z obou tříd vytvoříme příslušnou *DAO třídu*.

Konfigurace Hibernate

Z adresáře `conf` projektu otevřete v editoru vlastností (v Eclipse) konfigurační soubor `hibernate.properties`. V souboru je potřeba vyplnit informace o připojení k databázi, kde budou fyzicky uloženy spravovaná data.

SŘDB Oracle na stroji `students.kiv.zcu.cz`:

```
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
hibernate.connection.driver_class=oracle.jdbc.driver.OracleDriver
hibernate.connection.url=jdbc:oracle:thin:@students.kiv.zcu.cz:1521:students
```

SŘDB PostgreSQL na stroji `students.kiv.zcu.cz`:

```
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.connection.driver_class=org.postgresql.Driver
hibernate.connection.url=jdbc:postgresql://students.kiv.zcu.cz:5432/db2_vyuka
```

Ještě je nutné vyplnit správné přihlašovací jméno a heslo do nastavené databáze a základní konfigurace Hibernate je ukončena:

```
hibernate.connection.username=login
hibernate.connection.password=heslo
```

Automatické vytvoření tabulek v databázi

V projektu vytvoříme adresář `lib` a naimportujeme do něj příslušné JDBC, buď [ojdbc6.jar](#) pro SŘBD Oracle nebo [postgresql-9.1-903.jdbc3.jar](#) pro PostgreSQL. Příslušný `jar` soubor zařadíme do *Java PATH*. Nyní je vše připraveno ke generování příslušných tabulek ve vybrané databázi. Stačí spustit metodu `main` třídy `HibernateHelper` z balíku `util`.