

„Vnější“ programování (přes rozhraní/knihovny) – rozhraní ODBC, JDBC, rozhraní podporující objektově-relační mapování (Java Hibernate).

Thursday, May 30, 2013 8:20 AM

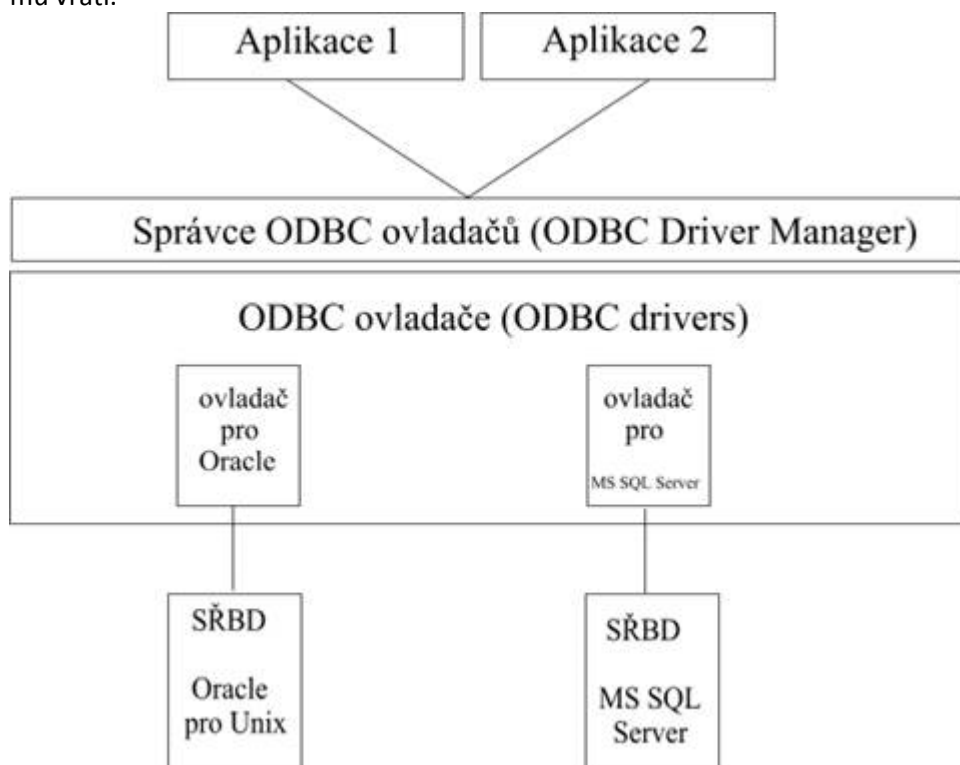
ODBC

Open DataBase Connectivity .Je standardizované softwarové API pro přístup k databázovým systémům (DBMS). Snahou ODBC je poskytovat přístup nezávislý na programovacím jazyku, operačním systému a databázovém systému. Je to čistě C-čkové API, které nemá žádný objektový základ.

Navrženo Microsoftem, proto primárně přístupné pouze přes C/C++. Založeno na specifikaci X/Open a ISO: SQL Call Level Interface (SQL/CLI)

Model struktury ODBC se dá znázornit pomocí čtyř vrstev:

- V první nejvrchnější vrstvě se nachází samotná aplikace. Ta v případě, že potřebuje data, provede volání ODBC funkcí (ve formě SQL dotazu).
- Druhou vrstvou je tzv. "Správce ODBC ovladačů" (ODBC Driver Manager). Úkolem správce ovladačů je zajistit propojení mezi aplikací a příslušným ODBC ovladačem (ODBC ovladače tvoří třetí vrstvu modelu, podrobněji viz dále). Jakmile aplikace potřebuje data, správce ovladačů vyhledá a nahraje příslušný ovladač. (ve formě DLL knihovny). Správce ovladačů také zjistí, jaké konkrétní funkce jsou podporovány jednotlivými ovladači, a uschová si jejich adresy v paměti do tabulky. V případě, že aplikace volá konkrétní funkci, správce souborů zjistí, ke kterému ovladači funkce patří a zavolá ji. Tímto způsobem může být prováděn souběžný přístup k více ovladačům, což se hodí v případě programování aplikací přístupujících souběžně k několika zdrojům dat.
- Třetí vrstvou zde již zmíněnou vrstvou jsou ODBC ovladače. Ty provedou zpracování volané ODBC funkce, přeložení požadavku do SQL pro příslušný SŘBD (DBMS) a jeho následné poslání.
- Poslední vrstvou je SŘBD, který provede zpracování operace požadované ODBC ovladačem a výsledky této operaci mu vrátí.



Open Database Connectivity (ODBC) is Microsoft's strategic interface for accessing data in a heterogeneous environment of relational and non- relational database management systems. Based on the Call Level Interface specification of the SQL Access Group, ODBC provides an open, vendor- neutral way of accessing data stored in a variety of proprietary personal computer, minicomputer, and mainframe databases.

ODBC alleviates the need for independent software vendors and corporate developers to learn multiple application programming interfaces. ODBC now provides a universal data access interface. With ODBC, application developers can allow an application to

concurrently access, view, and modify data from multiple, diverse databases.

ODBC is a core component of Microsoft Windows Open Services Architecture. Apple has endorsed ODBC as a key enabling technology by announcing support into System 7 in the future. With growing industry support, ODBC is quickly emerging as an important industry standard for data access for both Windows and Macintosh applications.

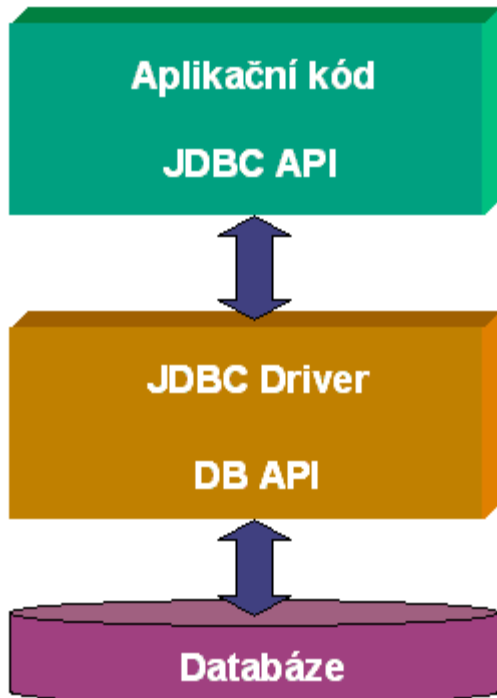
From <<http://support.microsoft.com/kb/110093>>

JDBC (Java Database Connectivity)

- jeho API poskytuje základní rozhraní pro unifikovaný přístup k databázím, aplikační programátor je tak odstíněn od specifického API databáze a může se naučit pouze jednotné rozhraní JDBC
- lze použít i mimo databáze – pro přístup k datům ve formě tabulek (CSV, XLS, ...)
- ovladače jsou k dispozici pro většinu databázových systémů

inspirováno rozhraním ODBC:

- objektové rozhraní
- strukturovanější a přehlednější
- možnost spolupráce s ODBC



JDBC ovladač

- zprostředkovává komunikaci aplikace s konkrétním typem databáze
- implementován obvykle výrobcem databáze
- dotazovací jazyk – SQL
 - předá se databázi
 - ovladač vyhodnotí přímo
- reprezentován specifickou třídou
 - `sun.jdbc.odbc.JdbcOdbcDriver`
 - `com.mysql.jdbc.Driver`

Typy JDBC ovladačů

Typ 1:

- využívá ODBC (pres JDBC-ODBC bridge)

- Obtížně konfigurovatelné

Typ 2:

- komunikace s nativním ovladačem nainstalovaným na počítači

Typ 3:

- komunikuje s centrálním serverem (Network Server) síťovým protokolem
- pro rozsáhlé heterogenní systémy, velmi efektivní i díky poolingů připojení

Typ 4:

- založen ciste na jazyce Java
- přímý přístup do databáze

Java Hibernate

Hibernate je framework napsaný v jazyce Java, který umožňuje tzv. ORM – Objektově-Relační mapování. Uspadňuje řešení otázky zachování dat z objektů i po ukončení běhu aplikace. Provádí podobné věci jako např. JPA – Java Persistence API.

Co dělá hibernate

Hibernate poskytuje způsob, pomocí něž je možné zachovat stav objektů mezi dvěma spuštěními aplikacemi. Říkáme tedy, že udržuje data persistentní. Dosahuje toho pomocí ORM, což znamená, že mapuje Javovské objekty na entity v relační databázi. K tomu používá tzv. mapovací soubory, ve kterých je popsáno, jakým způsobem se mají data z objektu transformovat do databáze a naopak, jakým způsobem se z databázových tabulek mají vytvořit objekty. Druhý způsob, jak mapovat objekty, je použít anotace místo mapovacích souborů. V Hibernate tedy pracujete se svými normálními business objekty, pouze pro každý atribut přidáte get/set metody a metody hashCode() a equals(). Nutno podotknout, že nelze použít EJB(viz.Java Bean), ale pouze tzv. POJO(Plain Old Java Object). Poté, co máte objekty uložené v databázi se na ně můžete dotazovat jazykem HQL (Hibernate Query Language), který je odvozen z SQL a je mu tedy velice podobný.

Objektovou struktura mého programu předhodím Hibernatu a základě tohoto objektového modelu (a označením objektů které tam požaduju) si Hibernate vytvoří vlastní schéma, aby věděl kde jsou data uložena.

Výhody používání Hibernate

Hibernate, framework pro perzistentní vrstvu, usnadňuje programátorovi práci tím, že nemusí transformovat objekty do relací ručně, ale přenechá to perzistentní vrstvě. Zároveň jsou tím odstíněna specifika jednotlivých databází – programátor používá API Hibernate.

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>

Objektově relační mapování

S relačními databázemi přichází i potřeba podpory **objektově relačního mapování**, což je proces překladu objektů na tabulkovou reprezentaci a překlad vazeb a vztahů mezi těmito objekty do dodatečných tabulek

Vloženo z <<http://www.cs.vsb.cz/behalek/frvs/2005/java/hibernate/hibernate.html>>