

# Oponentský posudek diplomové práce

Název DP: **Porovnání použitelnosti Java O/R frameworku**

Diplomant: **Martin Hlavatý**

---

## *Obsah práce:*

Předmětem DP je analýza použitelnosti nástroje Hibernate, který je Java nadstavbou pro perzistenci objektů formou objektově-relačního mapování. V první kapitole autor seznamuje s problémem propojení objektově orientovaného programování a perzistence vzniklých objektů v relační databázi. K tomuto účelu se používají nástroje ORM (object-relational mapping), které umožňují programátorovi OO jazyka přidat objektům vlastnost perzistence (v nějaké relační databázi). Autor se soustředí na nástroj Hibernate, který je nejrozšířenější v prostředí Java. Druhá kapitola pojednává o jazyku HQL, který nad datovým modelem Hibernate umožňuje dotazování podobného stylu jako klasické SQL, nicméně s řadou odlišností specifických pro objektový datový model. Autor zde rovněž dokazuje tezi, že jistá omezená relační algebra (jako formální aparát dotazování v relačních databázích) je zahrnuta v HQL (resp., že HQL je silnější). Ve třetí kapitole je diskutována použitelnost a omezení Hibernate v prostředí komplexních enterprise aplikací, čtvrtá kapitola se pak zabývá řešením běžných problémů při použití Hibernate.

## *Hodnocení:*

Diplomová práce se snaží o dva cíle. Jednak popsat vztah mezi relační algebrou (RA) a jazykem HQL, jednak zhodnotit přínosy použití nástroje Hibernate v prostředí komplexních a rozsáhlých (enterprise) aplikací. Bohužel, ani jeden z cílů nebyl zcela uspokojivě naplněn.

Už samotný úkol porovnávání RA a HQL je poměrně diskutabilní, vždyť oba dotazovací jazyky pracují nad jiným datovým modelem a tutíž autor porovnává hrušky s jablky. Ano, relační model lze triviálně zanořit do objektového modelu, který je ostře silnější a tudíž veškerá další tvrzení v kapitole 2.3 jsou triviální, pouze komplikovaně rozepsaná. Kdyby se operace RA přepisovaly do SQL, mohlo se rovnou prohlásit, že příslušné SQL a HQL konstrukce jsou ekvivalentní. Jediná zajímavost by byl důkaz tvrzení v sekci 2.3.9 (tj., že HQL žádným způsobem nepodporuje množinové operace), ten ale chybí. Výsledkem srovnání RA a HQL je pouhý fakt, že RA bez množinových operací lze emulovat pomocí HQL, což je poněkud neužitečná informace.

Další cíl práce, totiž evaluace Hibernate v prostředí enterprise aplikací, na mě působí poměrně nekonceptně a neúplně. Autor si na začátku kapitoly 3 zvolil 3 systémy, z nichž 2 používají Hibernate, poslední (starší) nikoliv. Poté naprosto ad-hoc způsobem komentuje omezení Hibernate vzhledem k dílčím problémům v rámci těchto systémů, a jejich řešení. Není rigorózně stanovena metodika, problémy nejsou vztaženy ke konkrétním systémům, prostě se systémem “co mně zrovna zajímavého napadlo” popisují různé vlastnosti Hibernate k různým dílčím problémům. Nejvíce mně zklamala kapitola 3.3 – performance. Očekával jsem, že autor objektivně srovná výkonnost nástroje Hibernate s tradičním nízkourovňovým JDBC přístupem pro tutéž aplikaci, a to buď z hlediska výkonnosti (rychlosti) samotné aplikace, nebo efektivnosti (opět rychlosti) implementace systému, anebo chybovosti ve vývoji.

To se ovšem nestalo, neboť všechny tři ukázkové systémy jsou pouze ve verzi s nebo bez Hibernate – nelze tudíž provést jakékoliv vyhodnocení. Autor pouze háže všelijakými konstantami, např. 40 sekund start-time reference implementation něčeho (není ani řečeno čeho!). V závěru práce autor mimoděk naznačuje, že původním předmětem práce byla právě implementace Hibernate do stávajícího enterprise systému, což by byl smyslupný cíl této práce. **Ostatně o tomto cíli se hovoří i v zadání DP.** To se ale nestalo a autor daný problém přesunul do “Future work”, což je z hlediska zadání diplomové práce k ničemu.

Kapitola 4 se nese v podobném duchu jako kapitola 3, totiž ad-hoc vymezení 2 problémů (logování a správa temporálních dat). Z celé práce hlavně není zřejmé, co je vlastně autorův původní přínos – jeho vlastní zkušenosti (autor evidentně v této oblasti pracuje pro společnost Profinit) – a co jsou známá fakta a doporučení z bohaté množiny publikací o Hibernate, které autor uvádí (několik knih, webové zdroje).

Práce je psána ve srozumitelné angličtině, i když na mnoha místech jsou školácké chyby (např., neurčitý člen u množného čísla, čárkami nevhodně oddělované věty, další syntaktické chyby). Po formální stránce práce splňuje požadavky na strukturu textu.

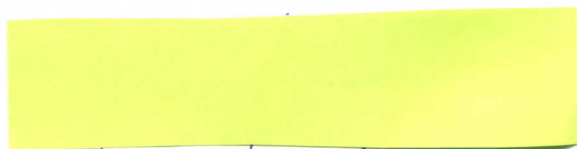
*Podrobnější připomínky, poznámky, dotazy:*

- 1) V kapitole 2 používáte nevhodnou relační terminologii. Místo relace-instance relace-relační hodnota se používá relační schéma-relace-prvek relace. Vaše terminologie není konzistentní s matematickým pojmem relace a je tudíž velmi zavádějící.
- 2) Sekce 2.3.3 – nepřehledná a nevysvětlená notace formalismů
- 3) Používáte nedefinované pojmy: CRUD, POJO, QA (resp., jsou vysvětleny na konci práce, kdy už je se čtením konec. Je lépe je vysvětlit přímo v místě prvního užití.)
- 4) V sekci Performance není vysvětleno co to je Build-time, Start-time, Run-time.

*Závěr:*

Celá práce působí dojmem, že autor je asi odborník v použití Hibernate a ve vývoji enterprise aplikací, nicméně obsah této DP je sporný – neobsahuje jasně specifikovaný přínos. Technologie Hibernate nebyla podle zadání implementována do existující rozsáhlé enterprise aplikace a tím nebylo umožněno objektivní srovnání. Práci doporučuji k obhajobě s váháním, neboť nesplnila bezezbytku zadání.

V Praze dne 15. září 2008



Doc. RNDr. Tomáš Skopal, Ph.D.  
oponent