

I. IDENTIFIKAČNÍ ÚDAJE

Název práce:	Static Metamodel Generator for JOPA
Jméno autora:	Benjamin Rodr
Typ práce:	bakalářská
Fakulta/ústav:	Fakulta elektrotechnická (FEL)
Katedra/ústav:	Katedra počítačů
Oponent práce:	Ing. Petr Aubrecht, Ph.D.
Pracoviště oponenta práce:	Payara Services, LTD.

II. HODNOCENÍ JEDNOTLIVÝCH KRITÉRIÍ

Zadání	průměrně náročné
<i>Hodnocení náročnosti zadání závěrečné práce.</i>	
Zadáním je vytvořit modul do Mavenu, který bude generovat kód na základě anotací podobně, jako to dělá JPA.	

Splnění zadání	splněno
<i>Posudte, zda předložená závěrečná práce splňuje zadání. V komentáři případně uveďte body zadání, které nebyly zcela splněny, nebo zda je práce oproti zadání rozšířena. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.</i>	
Pokud jsem z textu dobře vyrozuměl, zadání bylo splněno a testy to potvrzují.	

Zvolený postup řešení	správný
<i>Posudte, zda student zvolil správný postup nebo metody řešení.</i>	
Text obsahuje popis různých možností i důvody výběru použité metody. S argumentací i rozhodnutím souhlasím.	

Odborná úroveň	C - dobře
<i>Posudte úroveň odbornosti závěrečné práce, využití znalostí získaných studiem a z odborné literatury, využití podkladů a dat získaných z praxe.</i>	
Program zřejmě dělá, co má, to musí posoudit vedoucí práce.	
Kód mě moc nenadchl: Kód prochází všechny soubory bez ohledu na příponu, očekával bych zpracování pouze souborů s příponou „java“. Nevidím důvod použití <code>File.createNewFile()</code> , když hned vzápětí daný soubor vytvoříte <code>FileWriterem</code> .	
Přečtení celého souboru pomocí <code>Scanneru</code> je zcela určitě pomalé. <code>new Scanner(file).useDelimiter("\\Z").next()</code> Určitě bych doporučil <code>Files.readAllBytes</code> : <code>new String(Files.readAllBytes(Paths.get(FILENAME)))</code>	
Kód obsahuje zbytečné příkazy jako přiřazení do proměnné, která se dále nepoužije: <code>String outputPath = outputDirectory.getAbsolutePath();</code> Ale koukal jsem, že to už vedoucí práce opravil.	
Zbytečný try-catch, kde v catch části je <code>throw ex</code> . Javadoc u tříd je opravdu minimální. Chybí závorky u if-u s jedním příkazem.	

Formální a jazyková úroveň, rozsah práce

E - dostatečně

Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku.

Nejsem odborník na angličtinu, ale úroveň předložené práce je špatná. Formulace jsou často velmi kostrbaté a těžko pochopitelné, např.:

In the event that the supertype of the class is not processed, it will be processed first.

The processor has to eventually be used in the project that is supposed to be processed.

Text práce je zmatený. Musel jsem ho mnohokrát procházet tam a zpět, abych pochopil, co chtěl student sdělit. Například varianty parseru jsou rozebírány odděleně v kapitolách 3.2.3 a 3.6.1, pokaždé z jiného úhlu.

Kapitola 3.1 se jmenuje „Existing solutions“ – ve dvou odstavcích jsou zmíněny Hibernate a EclipseLink, bez citace a hlubší analýzy, jak fungují. To považuji za velký nedostatek.

Kapitola „Discovering classes to create a metamodel“ řeší možnosti parsování zdrojových kódů. Trvalo mi dost dlouho pochopit, že „Annotation Processing“ znamená použití kompilátoru dostupného přímo v JDK. Možnosti výstupu už jsou popsány jasně, byť volba jediného, špatně dokumentovaného generátoru asi není reprezentativní.

Vůbec jsem nepochopil, o čem je kapitola „(Im)Possible workaround“. Slovo „workaround“ se vyskytuje jen v této kapitole.

The metamodel does not contain methods of the class or fields that are not annotated with the OWL field annotations mentioned in 2.6 – znamená to, že metamodel obsahuje jen třídy a fieldy, které jsou anotované?

Formátování by si zasloužilo také pozornost:

Listing 2.9 – na nové stránce je jenom zavírací závorka

Kód v Javě se typicky tiskne v nějakém monospace fontem.

Špatná kapitalizace u nadpisů je už jenom drobnost.

Výběr zdrojů, korektnost citací

C - dobře

Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení závěrečné práce. Charakterizujte výběr pramenů. Posudte, zda student využil všechny relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.

Většina zdrojů a citací je v pořádku. Nicméně mám několik výtek:

Seznam build nástrojů nebo projekt na githubu mi nepřipadá jako dobrý zdroj.

Text vysvětlující důležitost metamodelu asi nemá referencovat generátor v JPA.

Další komentáře a hodnocení

Vyjádřete se k úrovni dosažených hlavních výsledků závěrečné práce, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, publikačním výstupům, experimentální zručnosti apod.

III. CELKOVÉ HODNOCENÍ, OTÁZKY K OBHAJOBĚ, NÁVRH KLASIFIKACE

Shrňte aspekty závěrečné práce, které nejvíce ovlivnily Vaše celkové hodnocení. Uvedte případné otázky, které by měl student zodpovědět při obhajobě závěrečné práce před komisí.

Text práce určitě potřebuje hodně vylepšit angličtinu a zapracovat na jasném vyjadřování. Také návyky při psaní kódu v Javě potřebují víc praxe. Nicméně zadání bylo splněno a věřím, že diplomová práce bude na vyšší úrovni.

Otázky:

- 1) JPA generuje metamodel stejně jako předložená práce. Víte, jak to dělá (například EclipseLink)?
- 2) Zkoušel jste i jiné generátory kódu než CodeModel?
- 3) Co je myšleno workarouDEM v kapitole 3.6.3?

Předloženou závěrečnou práci hodnotím klasifikačním stupněm **C - dobře**.

Datum: 10.6.2023

Podpis: Ing. Petr Aubrecht, Ph.D.