

Strukturovaný posudek oponenta bakalářské práce

Autor/autorka práce: Jiří Láška

Název práce: Generování magických pravidel z ozdobeného logického programu

Obsah práce:

Vynikající logická struktura, nadprůměrný obsah i rozsah;
~~Velmi dobrá logická struktura, odpovídající obsah i rozsah;~~
Vyhovující logická struktura, obsah i rozsah;
Nevyhovující

Komentář:

Práce obsahuje lepší a horší kapitoly. Pochválit lze kap. 3 a 4 kde jsou veškeré informace pro uvedení čtenáře do problému magických množin a funkcionálním programování v jazyku Java. Na druhou stranu kap. 2, 5 a 6 nemají dobrou úroveň. Některé pojmy jsou uvedeny v nelogickém pořadí a čtenáři je jejich význam vysvětlen až v dalších částech textu nebo je jejich vysvětlení zcela opomenuto. Autor práce v úvodních kapitolách popisuje zbytečně mnoho konkrétních příkladů jednotlivých pojmů, jako jsou příklady symbolů, celých čísel, desetinných čísel, řetězců a proměnných. Závěrečná kapitola popisující realizaci algoritmu tvorby magických pravidel obsahuje velmi zhuštěný popis samotné implementace algoritmu kombinovaný s vysvětlováním některých použitých ADT. Tzv. „reálná data“ byla vytvořena ručně, což neodpovídá obvyklé představě o tom, co to jsou reálná data. Závěrečná informace o době běhu aplikace zpracovávající 19 pravidel logického programu je zcela nevyhovující. Čtenář neví, na jaké sestavě bylo měření prováděno, ani nedokáže z jednoho uvedeného měření odhadnout časovou náročnost. Tuto informaci by bylo možné vzhledem k cílům práce zcela vynechat. Uživatelská příručka nejdříve popisuje, jak spustit jednotkové testy a teprve následně, jak spustit samotný program.

Kvalita řešení a dosažených výsledků:

Vynikající; ~~Velmi dobrá;~~ Vyhovující; ~~Nevyhovující~~

Komentář:

Cílem práce byla implementace algoritmů generující magická pravidla. Tento cíl by splněn bez výhrad. Kvalita implementace je na vysoké úrovni a má jasnou logickou strukturu. Kód je hojně komentován a to především i vhodně uvnitř jednotlivých metod.

Pokrytí jednotkovými testy je trochu zkrácené vzhledem k tomu, že obsahuje i takové testy, jako je kontrola stejně vytvořených instancí.

Na přiloženém CD chybí soubor .project, který je zmíněný v souboru README.txt. Přiložený soubor build.xml pro Apache Ant byl automaticky vygenerován použitým vývojovým prostředím Eclipse, a tedy obsahuje absolutní cesty k adresářům. Uživatel, který nemá instalované stejné vývojové prostředí, nedokáže spustit testy bez zásahu do konfiguračního souboru. Předklad samotného programu proběhne v pořádku.

Formální úroveň:

~~Vynikající; Velmi dobrá;~~ Vyhovující; ~~Nevyhovující~~

Komentář:

Práce obsahuje větší množství formálních a pravopisných chyb či neformálních vyjádření, např.:

- V definici 2.7 na straně 10 je použit zatím nedefinovaný pojem *literál*.
- Diagram 5.1 na straně 29 obsahuje titulek „Celkový pohled“, ačkoli se jedná o UML diagram balíčků programové realizace.
- Na straně 30 jsou v textu reference na ukázky 5.1, 5.2 a 5.3, které představují graf předávání informace, ozdobená a magická pravidla. Samotné ukázky jsou však až na stranách 33 a 34.
- Na straně 31 je použit výraz „naimplementováno“.
- Na straně 32 je použit výraz „rozkouskovat“.
- Na straně 39 citují „... bylo potřeba vytvořit dvě struktury šité na míru ...“.

Práce s literaturou:

~~Vynikající; Velmi dobrá;~~ Vyhovující; ~~Nevyhovující~~

Komentář:

Skromně citovaná literatura obsahuje pouze 2 neinternetové zdroje, a to bakalářskou práci, na kterou autor navazoval, a dále pak disertační práci vedoucího práce. Zbývající literatura obsahuje Wikipedii, stackoverflow (internetové fórum) a domovskou stránku přednášek teoretické informatiky. Chybí odkazy na použité technologie, konkrétně Stream API, Swing, JUnit 4, Eclipse, Apache Ant, EclEmma Java Code Coverage.

Splnění zadání:

Splněno bez výhrad; ~~Splněno s menšími výhradami;~~ ~~Splněno s většími výhradami;~~ ~~Nesplněno~~

Komentář:

Dotazy k práci:

- Jaký je rozdíl mezi *extenzionálním* a *vestavěným* predikátem?
- Jak by se dalo integrovat Vaše řešení do některého z existujících implementací jazyka Prolog, např. SWI-Prolog, případně do některé deduktivní databáze?

Navrhuji hodnocení známkou **dobře** a práci doporučuji k obhajobě.

V Plzni 23.5.2016


Ing. Ondřej Kaas