

# Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Jakub Kolšovský  
**Název práce** Metajazyk generující zdrojové kódy  
**Rok odevzdání** 2020  
**Studijní program** Informatika      **Studijní obor** Programování a softwarové systémy

**Autor posudku** Mgr. Jakub Gemrot, Ph.D.      **Role** Oponent  
**Pracoviště** KSVI

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

## K celé práci

	lepší	OK	horší	nevyhovuje
Obtížnost zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Splnění zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rozsah práce ... <i>textová i implementační část, zohlednění náročnosti</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p><b>Komentář</b> Student zpracoval tematiku návrhových zdrojů ve formě jím navrženého jazyka MetaDPT, který je založen na syntaxi jazyků Java a C#, přičemž je obohacen o klíčová slova návrhových vzorů, pomocí nichž může programátor deklarovat jejich použití v rámci deklarací tříd. Výsledný kód v jazyce MetaDPT pak studentův překladač umí přeložit do jazyků Java a C#, přičemž dochází k implementaci jednotlivých návrhových vzorů. Překladač tak umožňuje programátorovy generovat boilerplate kód návrhových vzorů.</p> <p>Primární motivace práce je seznámit čtenáře a uživatele MetaDPT jazyka s návrhovými vzory a prezentovat jejich implementace v jazyce Java a C#. Což práce splňuje, protože skrz experimenty s jazykem MetaDPT má uživatel možnost nahlédnout, jak vypadá praktická implementace všech většiny z 23 návrhových vzorů popisovaných v klasické literatuře (Gamma, E. (1995). Design patterns: elements of reusable object-oriented software. Pearson Education India.).</p> <p>Za slabinu práce lze požadovat absenci ukázkového programu implementovaného v jazyce MetaDPT. Při pokusu prakticky jazyk použít však narazíme na různé problémy, např. u návrhového vzoru dekorátor překladač nevloží do výsledného přeloženého kódu v C# či Javě kód pozměněných (dekorovaných) metod a tento musí být do vygenerovaných tříd vložen ručně. Je to trochu škoda, protože MetaDPT jazyk obsahuje hezkou myšlenku, kdy lze do zdrojového kódu vkládat libovolný text, který programátor vloží mezi dvojité křížky (## kód ##).</p>				

**Textová část práce**

	lepší	OK	horší	nevyhovuje
Formální úprava ... <i>jazyková úroveň, typografická úroveň, citace</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Struktura textu ... <i>kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analýza	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vývojová dokumentace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uživatelská dokumentace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Komentář**

Text má vhodnou strukturu, dobře se čte. V analýze student prochází všech 23 návrhových vzorů a zdůvodňuje svou volbu jejich podpory ve svém jazyce, diskutuje volbu jazyka jako základu pro MetaDPT, stejně jako zdůvodňuje implementační rozhodnutí (volba ANTLR pro implementaci překladače, volba C# backendu, atp.). V textu student příliš neseznamuje s návrhovými vzory na abstraktní úrovni, jako spíš předpokládá, že je čtenář zná. Uvítal bych také, kdyby technickou sekci o volbě frameworku pro implementaci překladače předcházela analýza způsobu implementace návrhových vzorů do studentem navrhovaného MetaDPT jazyka, což by logicky mělo předcházet před volbou implementačních nástrojů.

Práce je na bakalářskou práci poměrně rozsáhlá, bez parseru (generovaného skrz ANTLR) obsahuje 4318 řádků kódu v C# (dle metriky MS VS 2019), definice lexeru a parseru pro ANTLR má pak 33 kB textu (1774 LoC).

Vývojová a uživatelská dokumentace je přiměřeného rozsahu a obsahu.

**Implementační část práce**

	lepší	OK	horší	nevyhovuje
Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stabilita implementace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Komentář**

Implementace překladače byla stabilní, překladač poskytoval poměrně vhodné chybové hlášení v případě nevalidního vstupu. Překladač generuje kód vhodně zalomený s přihlédnutím ke konvencím obou cílových jazyků zvlášť.

MS VS projekt překladače je přehledně strukturovaný, jednotlivé funkcionality jsou vhodně rozděleny do samostatných jmenných prostorů, které kopírují adresářovou strukturu. Kód je dokumentovaný, byť jednotlivé komentáře jsou poměrně krátké.

**Celkové hodnocení** Výborně Choose an item.

**Práci navrhuji na zvláštní ocenění** Ne

**Datum** 31. srpna 2020

**Podpis**