

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Lukáš Rozsypal
Název práce Kampa: an experimental programming language
Rok odevzdání 2020
Studijní program Informatika
Studijní obor Programování a softwarové systémy

Autor posudku Lubomír Bulej Vedoucí
Pracoviště Katedra distribuovaných a spolehlivých systémů

K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání	X			
Splnění zadání		X		
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>	X	X		

Cílem práce byl návrh a prototypová implementace experimentálního programovacího jazyka. Experimentální charakter jazyka není dán množstvím vlastností, které by nebylo možné najít v jiných jazycích, ale spíše snahou o poskytnutí takových vlastností v bezpečném, staticky typovaném jazyce bez nutnosti řešit uvnitř překladače speciální případy, které syntax jazyka v principu nepodporuje, ale programátoři je očekávají (např. spojování řetězců pomocí operátoru `+` v jazyce Java).

Vedle dnes běžných vlastností klade jazyk zvýšený důraz na bezpečnost při práci s pamětí a za tím účelem zavádí rozšířenou podporu pro hodnotové typy a jejich použití v různých kontextech. Pro bezpečnější práci s poli zavádí jazyk omezenou podporu pro závislé typy.

Prototypová implementace jazyka využívá framework Truffle, který je určen (ve spojení s just-in-time překladačem Graal) pro psaní efektivních interpretrů (převážně dynamických) programovacích jazyků. Přestože u staticky typovaných jazyků nejspíše nemá Truffle tak velký výkonnostní potenciál (není kde využít agresivní typovou spekulaci a specializaci), výsledkem je funkční interpreter jazyka Kampa, který běží na dnes již běžně dostupné Graal VM.

Čistě subjektivně musím konstatovat, že považuji syntaxi jazyka za poměrně těžko čitelnou, ale s ohledem na experimentální povahu jazyka za akceptovatelnou. Vytvořit jazyk, který bude zároveň výrazově silný, typově bezpečný a lehce čitelný je evidentně netriviální úkol—návrhu jazyka by prospělo testování na složitějších příkladech a rozsáhlejších projektech, ale to v rámci jedné bakalářské práce není dost dobře možné.

I v současném stavu, kdy má implementace řadu (autorem deklarovaných) omezení, považuji práci rozsahem a složitostí bližší spíše diplomové práci. Autor se zadání práce (i přes jeho velkou otevřenost) snažil naplnit v maximální možné míře, pracoval samostatně a nepochybně prokázal schopnost řešit netriviální problém. Práci proto považuji za úspěšnou.

Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>	X	X		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>		X		
Analýza		X		
Vývojová dokumentace		X	X	

Uživatelská dokumentace		X	X	
<p>Text práce je psán dobře srozumitelnou angličtinou a s určitou lehkostí, která není u anglicky psaných bakalářských prací úplně běžná. K typografii nemám výhrady.</p> <p>Část věnovaná analýze a návrhu jazyka je spojena do větší kapitoly, což nepovažuji za problém, protože cílem bylo návrh vlastností jazyka opřít o rozbor podobných vlastností v jiných jazycích. Co si myslím, že analytické části chybí, je větší důraz na příklady, které by porovnávaly způsob modelování různých konceptů v existujících jazycích a v Kampě.</p> <p>Druhá větší kapitola je věnovaná prototypové implementaci jazyka v rámci frameworku Truffle. Autor zde popisuje reprezentaci hodnot a funkcí, hlavní typy uzlů syntaktického stromu, který poté Truffle vyhodnocuje, rozhraní pro psaní knihoven a implementaci závislých typů. Myslím si, že této části chybí těsnější provázanost popisu implementace s popisem souvisejících konceptů Truffle. Rovněž se zde poměrně překvapivě objevuje popis maker, které bych považoval spíše za vlastnost jazyka než jeho implementace. I proto, že makra jsou vlastně použita k vytvoření některých běžných konstrukcí jazyka místo toho, aby je rozpoznával překladač.</p> <p>V práci není explicitní programátorská nebo uživatelská dokumentace, ale obojí do jisté míry zastupuje popis syntaxe jazyka a popis implementace hlavních aspektů v rámci Truffle. S ohledem na experimentální povahu jazyka bych neočekával, že vnitřní rozhraní implementace budou příliš stabilní.</p> <p>Co bych v textu také vytknul je malá pozornost věnovaná prezentaci výsledku ve formě delších kusů kódu. To sice z části supluje úvodní stránka projektu v souboru README.md, ale v samotném textu práce jsou větší příklady a testy zmíněny až v Appendixu, což je přinejmenším škoda. Vhodnější by bylo větší příklady prezentovat explicitně v rámci vyhodnocení.</p>				

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu ... architektura, struktury a algoritmy, použité technologie		X		
Kvalita zpracování ... jmenné konvence, formátování, komentáře, testování		X		
Stabilita implementace		X		
<p>K implementační části práce nemám vážné výhrady. Implementace je přehledně rozdělena na frontend, runtime, integraci do Truffle. Rozsah kódu je zhruba 10k LOC (necelých 14k včetně komentářů a prázdných řádků), build systém je automatizován pomocí nástroje Maven. Kód je solidně pokrytý testy, poměrně systematicky komentovaný na úrovni tříd, sporadičtěji pak na úrovni metod.</p>				

Celkové hodnocení Výborně
Práci navrhuji na zvláštní ocenění Ne

Datum 30. června 2020

Podpis