



# Posudek oponenta závěrečné práce

Oponent práce:	Ing. Jakub Jirůtka
Student:	Adam Plodek
Název práce:	Návrh šablonovacího jazyka a implementace překladače
Obor / specializace:	Teoretická informatika
Vytvořeno dne:	16. června 2022

## Hodnotící kritéria

### 1. Splnění zadání

[1] zadání splněno

► [2] zadání splněno s menšími výhradami

[3] zadání splněno s většími výhradami

[4] zadání nesplněno

Zadání stanovuje následující cíle:

1. Rešerše existujících šablonovacích jazyků – Tímto bodem se zabývá třetí kapitola, která na pouhých třech stránkách textu obsahuje velmi letmé představení pěti šablonovacích jazyků pro HTML. Jde o klasické, běžně používané šablonovací jazyky pro HTML, které jsou ovšem velmi vzdálené, a tedy těžko porovnatelné, jazyku, který student v této práci navrhl. Navržený jazyk má blíže spíše k DSL pro tvorbu HTML v rámci obecného funkcionálního programovacího jazyka, jakým je např. projekt Lucid pro Haskell. Tento bod zadání tedy považuji za nedostatečně naplněný.

2. Návrh šablonovacího jazyka se zaměřením na funkcionální principy a podporu HTML – Tento bod pokrývá kapitola čtvrtá. Navržený jazyk je velmi inspirovaný jazykem Haskell. Definuje jednoduchý, funkcionálně orientovaný typový systém, ve kterém nechybí podpora funkcí, vlastních složených typů (struktury a pole), algebraických typů a typových aliasů. Podporuje funkce vyššího řádu i jednoduchý „pattern matching“. Navržený jazyk je obecný; podpora HTML je řešena pomocí standardní knihovny, která poskytuje pomocné funkce pro tvorbu HTML a také Markdownu.

3. Popis jazyka formální gramatikou – Formální popis jazyka nalezneme na konci čtvrté kapitoly, která staví na rozsáhlé definici pojmů pro tvorbu gramatik a LL(1) parserů v kapitole druhé.

4. Implementace překladače – Součástí práce je funkční překladač a interpret navrženého jazyka implementovaný v jazyce Haskell.

5. Testování – Tomuto bodu je věnováno pouze několik slov na konci páté kapitoly. Rozsah a kvalitu implementovaných testů považuji za velmi slabou.

Zadání hodnotím jako průměrně náročné.

## 2. Písemná část práce

40 /100 (F)

Celkový rozsah písemné části práce je na hraně doporučeného minima. Čítá celkem 37 stran, obsahuje však velké množství rozsáhlých ukázek kódu a jiného obsahu, který by byl vhodnější spíše do příloh. Když odečteme prázdné stránky a tyto rozsáhlé ukázky kódu, zbyde cca 25 stran textu.

V práci se nedozvíme, co je vlastně šablonovací jazyk, tedy jaká má specifika a čím se liší od běžného programovacího jazyka. Ve třetí kapitole (Šablonovací jazyky) jsou pouze letmo zmíněny některé z typických rysů šablonovacích jazyků.

Zcela chybí jakákoli analýza nebo alespoň diskuze nad tím, proč byl jazyk navržen tak, jak byl navržen, jaké konkrétnější cíle sledoval. Nenajdeme zde ani rozbor vlastností navrženého jazyka a jeho typového systému z pohledu teorie programovacích jazyků a typových systémů.

V kapitole páté, Implementace překladače, není popsáno, jak se provádí evaluace výrazů.

Dále v práci postrádám instalační a uživatelský manuál.

Seznam zkratk čítá pouze tři položky, přičemž v textu je použito mnohem více zkratk.

Jazykovou stránku práce hodnotím jako velmi, velmi špatnou. Obsahuje velké množství pravopisných, gramatických a stylistických chyb všeho druhu, překlepů, nevhodných (velmi neformálních) formulací a místy i typografické chyby. Mohlo by se až zdát, že čeština není studentův rodný jazyk, potom by tyto nedostatky bylo možné prominout, ale není tomu tak.

Bibliografie obsahuje pouze devět zdrojů, všechny relevantní k danému tématu a korektně citovány.

## 3. Nepísemná část, přílohy

75 /100 (C)

Přílohu práce tvoří překladač a interpret navrženého jazyka implementovaný v jazyce Haskell a dále knihovna poskytující DSL pro tvorbu HTML a základních elementů Markdownu.

S jazykem Haskell mám bohužel pouze minimální zkušenosti a nikdy jsem v něm nepsal ani neviděl napsaný překladač. Hodnocení implementační části jsem proto konzultoval s kolegou, který tyto zkušenosti má.

Aplikace není architektonicky úplně dobře navržená. Není zde jasně ohraničený modul „type checkeru“; kontrola typů se provádí v rámci druhého průchodu parseru a působí, že vznikla spíš živelně než prvoplánově. Dále vyhodnocení výrazů (evaluace) se provádí přímo v generátoru AST.

Kód neobsahuje téměř žádné vysvětlující komentáře, některé části jsou zakomentované a celkově působí jako šitý horkou jehlou. Stejně tak obsažené testy. Nicméně, dle historie repositáře i vedoucího na něm student pracoval průběžně, nikoli na poslední chvíli.

Student si sám zvolil jazyk Haskell, se kterým sice již měl předchozí zkušenost, ale je zjevné, že v něm není nejsilnější. Jeho vedoucí Haskell nezná, takže mu v tomto směru nemohl moc pomoci. Musel si tedy sám poradit s poměrně nelehkým úkolem. Na druhou stranu, mohl požádat o rady některého z kolegů na fakultě, kteří se implementací překladačů v Haskellu zabývají nebo si sám najít vhodné výukové materiály a příklady, což mi potvrdil, že neudělal.

Implementace je funkční, avšak kvalitativně na úrovni hrubého proof-of-concept – nedosahuje kvalit produktu. Nicméně, pro obor Teoretická informatika je tato úroveň zcela dostačující.

#### **4. Hodnocení výsledků, jejich využitelnost**

75 /100 (C)

V navrženém jazyku ani jeho implementaci, v současném stavu, nevidím velký potenciál pro praktické využití, nicméně to ani nebylo cílem této práce. Student vytvořil funkční prototyp, na kterém ověřil použitelnost navrženého jazyka a principů.

#### **Celkové hodnocení**

60 /100 (D)

Student v rámci práce navrhl jednoduchý, avšak víceméně obecný, funkcionální programovací jazyk včetně typového systému. Překladač pro tento jazyk napsal v čistě funkcionálním jazyce Haskell, který je běžně používaným programovacím jazykům velmi vzdálený. Šlo tedy o nelehkou výzvu, se kterou si dokázal samostatně poradit, ač ne úplně ideálně.

Největším nedostatkem práce je bezesporu její písemná část.

Celkově práci hodnotím známkou D.

## **Instrukce**

### **Splnění zadání**

Posudte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posudte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

### **Písemná část práce**

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posudte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti.

Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 52/2021, článek 3.

Posudte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

### **Nepísemná část, přílohy**

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

### **Hodnocení výsledků, jejich využitelnost**

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

### **Celkové hodnocení**

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.