

# Posudek oponenta závěrečné práce

České vysoké učení technické v Praze

Fakulta informačních technologií

**Student:** Martin Kuzma  
**Oponent práce:** Mgr. Jan Starý, Ph.D.  
**Název práce:** Julia rozhraní pro knihovnu Triangle  
**Obor:** Softwarové inženýrství

**Datum vytvoření:** 31. 1. 2017

<b>Hodnotící kritérium:</b>	<b>Způsob hodnocení - následující škálou 1 až 5:</b>
<b>1. Náročnost a další komentář k zadání</b>	<b>1=mimořádně náročné zadání, 2=náročnější zadání, 3=průměrně náročné zadání, 4=lehčí, ale ještě dostatečně náročné zadání, 5=nedostatečně náročné zadání</b>
<b>Popis kritéria:</b> Podrobněji charakterizujte diplomovou (bakalářskou) práci a její případné návaznosti na předchozí nebo běžící projekty. Dále posuďte, čím je zadání této ZP náročné. (U obtížnější ZP lze dále tolerovat některé nedostatky, které by u ZP standardní obtížnosti tolerovány nebyly; a naopak u jednoduché ZP mohou být zjištěné nedostatky hodnoceny přísněji.)	
<b>Komentář:</b> Jedná se o zpřístupnění standardní knihovny pro triangulaci prostřednictvím balíčku pro jazyk Julia.	
<b>Hodnotící kritérium:</b>	<b>Způsob hodnocení - následující škálou 1 až 4:</b>
<b>2. Splnění zadání</b>	<b>1=zadání splněno, 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno</b>
<b>Popis kritéria:</b> Posuďte, zda předložená ZP splňuje zadání. V komentáři uveďte body zadání, které nebyly zcela splněny, případně rozšíření ZP oproti původnímu zadání. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.	
<b>Komentář:</b> Funkcionalita knihovny Triangle je zpřístupněna prostřednictvím Julia balíčku. Tím je zadání splněno, ale některé implementační detaily jsou na pováženou (viz dále).	
<b>Hodnotící kritérium:</b>	<b>Způsob hodnocení - následující škálou 1 až 4:</b>
<b>3. Rozsah písemné zprávy</b>	<b>1=splňuje požadavky, 2=splňuje požadavky s menšími výhradami, 3=splňuje požadavky s většími výhradami, 4=nesplňuje požadavky</b>
<b>Popis kritéria:</b> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části.	
<b>Komentář:</b> Rozsah práce je odpovídající. Z neznámého důvodu se čísluje už od prvního listu, takže samotný text začíná na stránce 21 a končí na straně 83.  Samotný <code>_text_</code> je relevantní, ale práce obsahuje značné množství zcela zbytečných obrázků. Předně "diagramy" pro chování přepínačů: to co by se dalo na pár řádcích říci slovy, vyjádří se půlstránkovým obrázkem. Stejně tak strukturované diagramy samozřejmých iterátorů působí zcela samoúčelně. Ukázky převodu Vectoru na jednorozměrné pole a zpět, obalovací omáčka soubotu <code>deps.jl</code> , a automaticky generovaný (!) soubor <code>deps.jl</code> k pochopení také nijak nepřispívají, Ukázka kódu 3.1 nepředvádí "funkce pro triangulaci", ale jen jejich interface, podobně některé další.	
<b>Hodnotící kritérium:</b>	<b>Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):</b>
<b>4. Věcná a logická úroveň práce</b>	<b>69 (D)</b>
<b>Popis kritéria:</b> Posuďte, 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.	

#### Komentář:

Práce vůbec nezmiňuje existující řešení, například balíček JuliaGeometry. Geometrický úvod je místy zmatený, některé definice nedávají smysl. Například už v samotné definici triangulace je daná množina `_prvkem_ roviny`, a zároveň je triangulace `T` rovna trojúhelníku `t_i`; v definici optimální triangulace se bez další zmínky zmiňuje `T`, zřejmě každá jiná triangulace; podle definice Delaunayho triangulace je opsaná kružnice prázdná, atd. Triangulační algoritmy uvedené na straně 29 a 30 neztriangulují ani jednotkový čtverec, jsou buďto chybně opsány nebo chybně přeloženy. Na straně 46 je obecný popis volání `ccall()` v Julia, ale chybí ukázka toho, jak konkrétně budeme volat funkce z `Triangle`. Podobně na straně 52 se popisuje obecný postup generování grafů, ale žádnou ukázkou takového grafu nevidíme.

V Návrhu se místo přímočaré kopie struktury `triangulateIO`, která v knihovně `Triangle` funguje jako univerzální struktura pro všechny vstup i výstup, zavedou rozličné datové typy pro všechny její součásti, přičemž skoro všechny mají zbytečná "nová" jména, a následuje výpis pování mezi jmény ve struktuře `triangulateIO` a nově zavedenými strukturami; například `"AreaFileSection"` je nová struktura pro `"arealist"` atd, na dvou stránkách. Zcela dadaisticky působí zavedení speciálních datových typů pro neexistující (!) sekce, například `NoSegmentFileSection` jako datový typ pro neexistující popis segmentů.

Pro účely parsování `commandline options` se zavede speciální datový typ `CommandLine` a metody `createCommand`, `parseOptions` atd; místo obyčejného `getopt` (to není přímo v jazyce?) zavádíme další zbytečnou strukturu. Místo cesty k souboru jakožto jména (stringu) se rovněž zavádí zbytečný abstraktní typ s metodami `getName` atd.

Kapitola realizace dosti přehledně popisuje samotnou implementaci; obsahuje ale balastní pasáže o tom, že datové typy musíme nejdříve definovat, než je předáme funkcím, a že generování grafu si máme představit jako "posloupnost úkonů, které se provádějí po sobě" (zřejmě narozdíl od ostatního kódu).

V ukázce kódu 4.6 je zřejmě chyba: část výstupu se píše na `stdout`, část do `fileStream`.

Kapitola Testování popisuje jen formální rámec takového testování. Je zmíněn server `TravisCL`, který "umožní automatizovat spouštění testů", ale žádný takový test nevidíme. Tato kapitola neobsahuje žádnou zmínku o tom, co a jak jsme přesně testovali a jak tyto testy dopadly. Není uveden jediný příklad nějaké konkrétní triangulace a jejího výsledku(!).

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

### 5. Formální úroveň práce

70 (C)

Popis kritéria:

Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 14/2015, článek 3.

#### Komentář:

Oba abstrakty by se bez problémů vešly na jednu stránku, a jméno `Jonathana Schewchuka` zřejmě nepatří do klíčových slov. Podobně Úvod o dva řádky přetéká na jinak prázdnou stránku. V sazbě se nijak neošetřují nepřízvučné předložky na konci řádků. Na straně 26 začíná řádek tečkou. V textu se často odkazuje na "funkci `triangulate 2.1`", "strukturu 2.2", "ukázkou 2.2." "objekt typu 2.6" apod, místo odkázů jménem, což znamená, že čtenář musí zbytečně listovat buďto k obrázku či ukázce kódu. V seznamu zkratk zbytečně přetékají dvě písmena.

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

### 6. Práce se zdroji

70 (C)

Popis kritéria:

Vyjádríte se k aktivitě studenta při získávání a využívání studijních materiálů k řešení ZP. Charakterizujte výběr studijních pramenů. Posuďte, zda student využil všechny relevantní zdroje nebo zda se pokoušel řešit již vyřešené problémy. 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.

#### Komentář:

Zadání obsahuje standardní učebnici výpočetní geometrie [3], ale pro každý nový pojem obsahuje práce další náhodný odkaz, zjevně první relevantní z google search. Tak například definice konvexního obalu odkazuje na náhodné slajdy z MFF UK, definice triangulace na dvoje náhodné slajdy z FEL a z ETH Zurich, a tak dále. Všechny tyto pojmy jsou samozřejmě ve standardní učebnici zavedeny. Seznam literatury je tak zaplevelen mnoha zbytečnými odkazy na web. Stejně tak odkaz na Delaunayho horolezecké vzpomínky by se dal beze ztráty vypustit. Odkazy na různé části Julia dokumentace jsou rozesety po celém seznamu. Odkazy [20] a [21] jsou stejné.

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):</i>
<b>7. Hodnocení výsledků, publikační výstupy a ocenění</b>	<b>75 (C)</b>
<i>Popis kritéria:</i> Vyjádřete se k úrovni dosažených hlavních výsledků ZP, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, apod. Případně také zhodnoťte, zda software nebo zdrojové texty, které nevytvořil sám student, byly v ZP použity v souladu s licenčními podmínkami a autorským právem. Popište případnou publikační činnost a získaná ocenění související s řešením této ZP.	
<i>Komentář:</i> Hlavním výsledkem je funkční balíček pro jazyk Julia, který dokáže triangulovat konečné množiny bodů v rovině prostřednictvím C knihovny Triangle.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení - nehodnotí se</i>
<b>8. Komentář o využitelnosti výsledků</b>	
<i>Popis kritéria:</i> Uvedte, zda hlavní výsledky ZP rozšiřují již publikované známé výsledky a/nebo přinášející zcela nové poznatky. Uveďte možnosti využití výsledků ZP v praxi.	
<i>Komentář:</i> Software je prokazatelně použitelný.  Od uživatele se zřejmě očekává zběhlost ve vytváření Julia balíčků - dostat se na spuštěné Julia konsoli k použití implementovaných funkcí stojí několik málo konkrétních Julia příkazů, které nejsou podrobně popsány.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení - nehodnotí se</i>
<b>9. Otázky k obhajobě</b>	
<i>Popis kritéria:</i> Uvedte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).	
<i>Otázky:</i> Proč zavádíte speciální datové typy pro popis neexistujících sekcí? Odkud přesně pochází omezení implementace na Julia 0.5 a na operační systém Linux? V čem je vytvořený software lepší a v čem horší než existující geometrické balíčky napsané přímo v Julia?	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):</i>
<b>10. Celkové hodnocení</b>	<b>70 (C)</b>
<i>Popis kritéria:</i> Shrňte stránky ZP studenta, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení <b>nemusí</b> být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích 1 až 9.	
<i>Text hodnocení:</i> Vytvořený software je funkční, ale samotná implementace není lepší než průměrná.	

Podpis oponenta práce: