



Posudek oponenta závěrečné práce

Student: Jiří Borský
Oponent práce: Ing. Jakub Průša
Název práce: Generátor vstupních dat pro validaci Bash skriptů
Obor: Webové a softwarové inženýrství

Datum vytvoření: 10. 2. 2019

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – následující škálou 1 až 4:</i>
1. Splnění zadání	1=zadání splněno, 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno
<i>Popis kritéria:</i> Posuďte, 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, posuďte 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.	
<i>Komentář:</i> Zadání práce bylo splněno, nicméně k práci jako takové mám výhrady, které jsou pospané dále v tomto posudku.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):</i>
2. Písemná část práce	60 (D)
<i>Popis kritéria:</i> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále 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. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, 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.	

Komentář:

U obrázku 2.1. chybí zdroj. Navíc je odkazován z textu, který pojednává o příkazové řádce a operačních systémech na uživatelských počítačích, avšak je v něm iOS, který příkazovou řádku nemá a ani není určen pro počítače, ale pro mobilní telefony. Pokud má zobrazovat i operační systémy mobilních zařízení, navrhol bych zahrnout i Android. Zároveň by bylo lepší obrázek překreslit do křivek pro lepší čitelnost. Dále je v textu uvedeno následující: "Linux/Unix má velké využití na serverech nebo malých zařízeních". Nevím, co si mám představit pod "malým zařízením" a navíc jsem tvrzení nenalezl v uvedeném zdroji.

U obrázku 2.2. bych navrhol jeho překreslení a přejmenování poslední možnosti "other (tell us in the comments)". Navíc je obrázek anglicky v jinak českém textu (stejně jako obrázek 2.1.).

U specifikace požadavků 3.1.1 až 3.1.8 by bylo vhodné doplnit ukázky vygenerovaného obsahu pro lepší přehlednost.

V kapitole 4 se v úvodu pracuje s pojmem "fuzzing" před tím, než je zaveden. Také mi v této kapitole chybí rozsáhlejší analýza předchozího generátoru. Například jeho rozhraní, rychlost, funkce. V práci je předchozí generátor popsán jako: "funguje docela dobře". Jeho celková analýza je na 4 řádky psaného textu.

V kapitole 5.2 Architektonické návrhové vzory bych uvítal i další návrhové vzory. Například MVVM.

V části 5.2.2 MVC je věta: "Toto rozdělení pomáhá minimalizovat duplikaci kódu.", t.j. rozdělení na Model-View-Controller. S tímto tvrzením nesouhlasím. Hlavním důvodem je rozdělení zodpovědností. Zároveň není v kapitole uveden zdroj pro čerpaní informací ohledně MVC.

V části 5.4.1 Zasazení do projektu je uvedeno následující: "Model v tomto případě reprezentuje třída DatabaseManager. Tato třída zajišťuje přístup k databázi a validaci dat." Opět s tvrzením nemohu souhlasit. O přístup do databáze by se měl starat controller a aktualizovat model. Řešení, kdy má model DatabaseManager instanci, mi nepřijde vhodné.

Sekvenční diagram na obrázku 5.3 obsahuje komunikaci mezi třídami. Bohužel se jedná pouze o třídy navržené autorem, jak by mohl portál Learnshell integrovat generátor. Nejedná se o sekvenční diagram pro generátor a generování dat. Tímto neuduhem trpí celá kapitola. Věnuje se celku jakožto LearnShell portálu a ne generátoru jakožto tématu závěrečné práce.

V kapitole bych rád viděl diagram tříd/třídy a nadefinované její rozhraní.

Na straně 33 a 34 je v kapitole 6.3.1 Základní rozdělení generátoru 2x velmi podobný odstavec:

"Dále můžeme vidět proměnnou (`__NON_PRINTABLE`), která uchovává netisknutelné znaky. Tyto znaky získávám převedením dekadického čísla na znak, který odpovídá danému číslu v ASCII tabulce."

a

"Dále je vidět proměnná (`__NON_PRINTABLE`), která uchovává netisknutelné znaky. Tyto znaky získávám převedením dekadického čísla na znak, který odpovídá danému číslu v ASCII tabulce."

Zřejmě se jedná nepozornost ze strany autora při finalizaci práce.

V sekci 7.4.1 Využití v projektu se autor odkazuje na obrázek 7.6. Ten v práci vůbec není. Zřejmě jde o obrázek 7.1. V závěru kapitoly na straně 45 končí text: "kde jsou testy spouštěny pomocí CI. Příklad jak může vypadat takový konfigurační soubor pro CI jsem předvedl na obrázku." a zcela chybí odkazovaný obrázek.

Kapitola o testování detailně popisuje různé důvody proč a jak testovat. Ale jen velmi málo popisuje, jak autor svoji implementaci testoval.

V závěru práce bych uvítal popis možností navázání na tuto práci.

Autor použil 44 zdrojů. U některých chybí informace, zda jsou dostupné online nebo knižně (chybějící ISBN).

Hodnotící kritérium:

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

3. Nepísemná část, přílohy

60 (D)

Popis kritéria:

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

Komentář:

Po implementační stránce je práce relativně krátká. Pro představu počet řádku ve všech souborech s implementací:

```
81 ./Task.py
24 ./Controller.py
25 ./core.py
97 ./Generator.py
9 ./Script.py
236 total
```

Pokud by práce obsahovala vše potřebné, tak s takovým rozsahem nemám problém. Práce má ve třídě Generator.py metody pro generování slov "lower_ascii" a "upper_ascii", nicméně vygenerovat slovo jde pouze z malých znaků, protože metoda "_simple_word" volá ve své implementaci pouze "lower_ascii". Práce obsahuje ve své implementaci několik "#TODO" poznámek. S nimi souhlasím a uvítal bych jejich implementaci již do této verze.

Dále implementace funkce "_single_letter_parameter" a "_multi_letter_parameter" je úplně identická a liší se pouze výchozí hodnotou v její definici.

U testování shell skriptů bych ocenil i generování dlouhých přepínačů pomocí dvou pomlček. Toto současná implementace neumožňuje.

U funkcí bych očekával jejich dokumentaci a popis rozhraní. To bohužel ve zdrojových souborech úplně chybí.

Autor v práci zmiňuje standard PEP 8 pro psaní čitelného kódu. Nicméně se zdá, že ho autor nepoužívá. Protože po jeho pouštění je v projektu odhaleno 23 varování. Jako vhodné řešení bych viděl integrace PEP 8 do CI projektu.

Hodnotící kritérium: *Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):*

4. Hodnocení výsledků, jejich využitelnost

65 (D)

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

Komentář:

Práce se zdá použitelná v systému Learnshell, nicméně vidím v ní ještě velké množství práce, než bude nasazená v produkčním prostředí.

Hodnotící kritérium: *Způsob hodnocení – nehodnotí se*

5. Otázky k obhajobě

Popis kritéria:
Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

Otázky:

1. Nezvažoval jste využití TDD (Test Driven Development) při vývoji práce? U tohoto typu práce se to přímo nabízí.
2. Nezvažoval jste použití Cythonu pro zrychlení celého generátoru a zachování tak kompatibility s Pythonem?
3. Neplánujete práci zveřejnit jako open-source pro možnost využití na dalších projektech?

Hodnotící kritérium: *Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):*

6. Celkové hodnocení

60 (D)

Popis kritéria:
Shrňte stránku 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.

Text hodnocení:

Práci doporučuji k obhajobě. Nicméně její písemná forma obsahuje faktické i typografické chyby. Její implementace je opravdu velmi minimální, nicméně splňuje zadání s výhradami.

Podpis oponenta práce: