

Posudek oponenta bakalářské práce

Autor/autorka práce: Michal Linha

Název práce: Nástroj pro ruční vytváření komplexních vstupních dat pro testování

Obsah práce

Práce slouží jako rozšíření unit testů, které by mělo usnadnit uživatelům zadávání vstupních dat v jednotkových testech. Text práce pokrývá všechny body zadání, bohužel ale na sebe jednotlivé části příliš nenavazují. V první části práce je popsáno několik různých automatických algoritmů pro generování testových dat, bez hlubšího rozboru jejich využitelnosti. Druhá část je zaměřená na možnost ručního vkládání dat, bohužel ale bez toho, že by jakkoliv využila teoretické poznatky popsané v první části (například jako náповědu pro testera). Také analýza problému v praktické části je provedena jen velmi povrchně, zejména chybí jakákoliv diskuse týkající se ukládání vytvořených dat, možnosti jejich zpětné kontroly a opakovatelnosti testů. Text práce je výrazně zaměřen na datové typy v jazyce, bez toho aby se zabýval tím, proč tester bude chtít data nastavovat nebo jak budou výsledné testy použity. Knihovna umožní libovolné nastavení atributů generovaných tříd, ale nijak nepracuje s následnými aserty, takže výsledné testy jsou bez další práce jen obtížně použitelné. Je také škoda, že se autor nepodíval na další knihovny zaměřené na generování formulářů pro vstup dat, jako je např. FXForm2, tyto nástroje by mohly posloužit jako základ nebo alespoň inspirace pro implementaci požadované funkcionality. Popis implementace je v podstatě jen výčet tříd a metod, nejsou zde žádné informace o struktuře knihovny nebo datovém modelu, kterým jsou popisovány generované instance, v této podobě tedy aplikace může sloužit spíše jako demonstrátor ukazující možnosti ruční tvorby dat.

Kvalita řešení a dosažených výsledků

Vytvořená knihovna je spustitelná podle návodu na CD a po explicitním vyvolání umožňuje do jednotkových testů vkládat vlastní vstupní data. Vytvořené rozhraní je ale krajně nepřehledné a jen obtížně použitelné. Hlavním problémem je nemožnost kontroly už vytvořených dat (nemluvě o nemožnosti jejich uložení pro opakování testů), takže při neúspěchu unit testu tester neví, jestli je chyba v zadávaných datech nebo v programu. Není možné se jakkoliv vrátit a zkontrolovat vytvořená data. Tvorba instancí je také matoucí, zejména není příliš jasné, co se děje, pokud uživatel vytvoří několik různých instancí – použije se první, poslední? Pro tvorbu dat slouží 2 režimy, základní vytváří instance na základě existujících veřejných konstruktorů, ale pro složitější objekty je potřeba nástroj přepnout do jiného režimu, kdy jsou atributy zjišťovány ne z konstruktoru, ale přímo z deskriptorů třídy. To je minimálně matoucí, při přepnutí do detailního režimu najednou uživatel vidí jinak strukturovaná okna a jiné atributy (v závislosti na podobě konstruktorů). Problémy, které mohou nastat, jsou řešeny prostým výpisem výjimky do dialogového okna s velmi generickým označením, alespoň nejběžnější chyby (např. snaha vytvořit instanci neexistující nebo abstraktní třídy) by mohly mít vlastní chybové hlášení, které by testerovi usnadnilo hledání problémů.

Výsledná knihovna je spustitelná, ale bez další práce jen velmi obtížně využitelná jako zdroj testovacích dat.

Formální úroveň

Text práce je přehledný a dobře strukturovaný, bohužel ale místy zvláště formulovaný; naštěstí nikdy na úkor pochopitelnosti vět. Zdrojové texty jsou důkladně komentované.

Práce s literaturou

V práci je odkázáno 35 relevantních zdrojů, vesměs zaměřených na použité technologie nebo na výzkum týkající se generování dat. Chybí mi cokoliiv, co by se týkalo automatické tvorby formulářů pro datové objekty nebo literatura zaměřená na potřeby testerů během testování. Veškerá odkazovaná literatura je relevantní.

Splnění zadání

Zadání považuji za splněné, s tím že testování bylo zaměřeno jen na správnost generování objektů, ale ne na použitelnost vytvářeného rozhraní. V analytické části také chybí větší důraz na problematiku ručního zadávání dat, soustředí se jen na automatické metody, které nejsou ve finále použity.

Dotazy k práci

1. Jak složité by bylo realizovat ukládání testových dat, jaké formáty nebo způsoby jejich uložení byste použil?
2. Jak jste prováděl testování s využitím TestFX, respektive proč bylo potřeba zadávat nějaké hodnoty ručně? Neztrácí pak použití TestFX frameworku smysl?

Navrhuji hodnocení známkou **dobře** a práci doporučuji k obhajobě.

V Plzni 12. 8. 2019

Ing. Richard Lípka, Ph.D.