

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Kateřina Průšová
Název práce Faker - generátor pseudonáhodného obsahu objektů v C#
Rok odevzdání 2021
Studijní program Informatika
Studijní obor Programování a softwarové systémy
Autor posudku Adam Dingle
Pracoviště KSVI
Role Oponent

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

K celé práci

	lepší	O K	horší	nevyhovuje
Obtížnost zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Splnění zadání	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rozsah práce ... textová i implementační část, zohlednění náročnosti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Komentář This bachelor's thesis presents an implementation of a faker, i.e. a software system that can generate pseudorandom objects of arbitrary data types in C# in a flexible way. One part of the thesis explains the weaknesses of the built-in C# random number generator, then surveys a variety of better methods of generating pseudorandom numbers, some of which were invented only in recent years (and which I personally was unaware of). A random number generator of such high quality is arguably beyond the needs of most faker users, but it's still nice that the thesis explores this topic in detail. The thesis also examines a number of other faker implementations in great detail and presents a clear design for a new implementation that meets a set of desirable requirements, using various C# language features such as generics and reflection as necessary.</p> <p>Overall I believe that the thesis very successfully achieves its goals.</p>				

Textová část práce

	lepší	O K	horš í	nevyhovuj e
Formální úprava ... <i>jazyková úroveň, typografická úroveň, citace</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Struktura textu ... <i>kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analýza	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vývojová dokumentace	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uživatelská dokumentace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Komentář The entire thesis text is exceptionally clearly written. The author describes both existing implementations and her own design decisions with great attention to small details, and thoroughly explains why she chose certain design paths over others. She tries hard to build an efficient implementation, presenting various benchmarks with low-level performance measurements. Once again, this attention to performance may even be beyond what many faker users will actually require, but it is still nice to see.</p> <p>I like the tutorial format of the user documentation, which is quite easy to follow. However it would be nice if the author also provided API reference documentation that listed exactly which classes and methods are available in her system.</p>				

Implementační část práce

	lepší	O K	horš í	nevyhovuj e
Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stabilita implementace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Komentář Overall I think the author made a fine series of design decisions and constructed a system that seems easy to use. She's made good use of inheritance in the BaseFaker/AutoFaker/StrictFaker classes, and her system of "fluent helper" objects is a clever way to use generics to enforce type safety. It's great that she thought about multithreading thoroughly.</p> <p>I will say that the system's constructs for conditional rules (When / Otherwise) add a fair amount of complexity to the design; for example, the inheritance diagram in 3.11 seems pretty complicated. To some degree this seems to duplicate programming language constructs (if / else) at the meta level. I don't know whether it would be possible to modify the design so that a user could specify these sorts of conditions in ordinary code, e.g. by providing a C# method that contains the if/then logic and calls Faker methods when necessary. But it might be nice if the thesis at least explored this possibility (or explained why it would be unrealistic).</p> <p>I have not studied every line of the implementation, however it seems to be well structured and commented, and it's great that there is a thorough suite of unit tests. I was able to build the system and run the unit tests successfully.</p>				

Celkové hodnocení Výborně**Práci navrhuji na zvláštní ocenění** Ne**Datum** 9.8.2021**Podpis**
