

Warning

This page is located in archive. Go to the latest version of this [course pages](#). Go the latest version of [this page](#).

Unit testing

Během práce na předchozích úlohách jste jistě zjistili, že testování toho, co naprogramujete, je nedílnou součástí tvorby softwaru. Vlastně každá funkce, metoda, nebo třída by měla mít svůj testovací kód, který vám jako programátorům pomůže zajistit, že váš kód dělá to, co po něm chcete.

Nebylo by pěkné, kdybyste pak mohli přijít ke svému třeba i poměrně rozsáhlému projektu a říct: “Hej, projekte, otestuj se?” A jakési kouzlo by pak prošlo, spustilo a zkontrolovalo výsledky všech testů, které jste pro svůj kód připravili. To je jeden z účelů metodiky, které se říká unit testing.

[https://en.wikipedia.org/wiki/unit_testing]

Co to má společného s námi?

Úloha Spam je trochu rozsáhlejší. Abyste si mohli být jistí, že jdete po správné cestě, připravili jsme pro vás sadu unit testů, které by měly zkontrolovat všechny důležité aspekty vašeho kódu. Kontrola v upload systému kontroluje hlavně váš finální produkt - spam filtr. Unit testy zkontrolují dílčí funkce a třídy, které budete muset vytvořit. Pokud porozumíte jejich výstupům, měly by vám dát poměrně dobrá vodítka k tomu, kde se ve vašem kódu vyskytuje chyba či jiný problém.

U každého dílčího kroku na úloze Spam, který po vás požadujeme, proto najdete jeden či více souborů s unit testy. Ty si ukládejte a spouštějte ve stejném adresáři, kde budete vyvíjet svůj spam filtr a jeho podpůrné třídy či funkce.

Unit testy, které od nás dostanete, budou v modulech, jejichž názvy mají tvar `test_XXXXX.py`. Dostanete od nás také soubory, jejichž jména budou ve tvaru `tst_XXXXX.py`. Tyto moduly neobsahují unit testy, ale např. definice nějakých konstant či podpůrných funkcí, a jsou importovány do modulů s unit testy.

Váš kód v žádném případě nesmí importovat ani moduly `test_XXXXX`, ani `tst_XXXXX`. Můžete testovací kódy studovat a inspirovat se v nich, ale nesmíte je přímo využívat jako součást vašeho řešení úlohy, protože tyto moduly nebudou při finální kontrole kódu k dispozici!

Jak je to udělané?

Existuje několik balíků či knihoven umožňujících provádět unit testing. V základní distribuci Pythonu najdete dva:

- `doctest` [<http://docs.python.org/py3k/library/doctest.html>] (který je použitý např. v [Python tutoriálu](http://howto.py.cz/english/ch05.xhtml#auto7) [<http://howto.py.cz/english/ch05.xhtml#auto7>]) a
- `unittest` [<http://docs.python.org/py3k/library/unittest.html>].

Pro implementaci našich testů jsme vybrali druhou možnost, balík `unittest`, protože je Pythonovskou implementací standardně používaného testovacího frameworku `xUnit` [<https://en.wikipedia.org/wiki/xUnit>].

K otestování jedné dílčí funkce, metody nebo třídy stačí spustit příslušný unit test. V pozdější fázi řešení úlohy budete občas chtít spustit všechny unit testy. Doporučujeme vám vytvořit si dávkový soubor, po jehož spuštění modul `unittest` vyhledá a spustí všechny testy v aktuálním adresáři:

```
c:\python34\python -m unittest discover .
```

Řetězec `c:\python34\` je cesta k interpretu Pythonu a vy si ji samozřejmě správně upravte podle vlastní instalace. Tečka za mezerou na konci řádku označuje aktuální adresář a musí tam být. Funkce `discover` je k dispozici až od verze 3.2. Vyvíjíte-li v Pythonu 3.1, udělejte si dávku s výčtem všech unit testů.

Viz též [Spouštění testů v PyCharm](#)

[/b211/courses/b4b33rph/tutorialy/python/pycharm#spousteni_testu_v_pycharm].

Prohřešky v unit testech

Existuje celá řada doporučení, jak unit testy psát, co v nich dělat a co naopak nedělat. Např. se v praxi nedoporučuje v unit testech pracovat se soubory na disku, protože to přináší řadu neblahých jevů (testy jsou pomalé, dochází k souběhu, atd.).

My jsme se záměrně rozhodli některá z těchto doporučení porušit kvůli názornosti. V unit testech např. vytváříme na disku adresáře a soubory, které následně mažeme. Když budete kód procházet pomocí debuggeru, máte možnost vidět adresáře a soubory, které skutečně váš program zpracovává. Snad nám tyto prohřešky zkušenější programátoři mezi vámi odpustí.

[courses/b4b33rph/cviceni/spam/unit_testing.txt](#) · Last modified: 2018/07/17 13:25 (external edit)

Bug reports and suggestions Helpdesk CTU