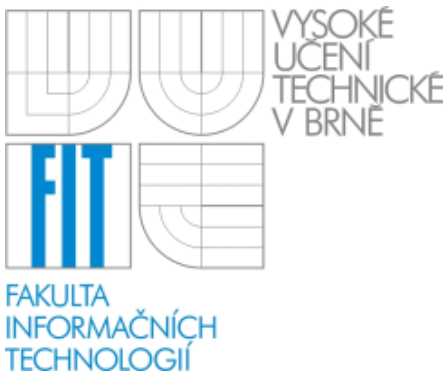
Projekt 1 – Testy a profilování

Petr Škoda

Ústav počítačové grafiky a multimédií
Fakulta informačních technologií

Vysoké učení technické v Brně



Motivace a zadání

Vyzkoušejte si psaní testů a také psaní kódu, pro který jsou testy již napsány (*Test-Driven Development*), dále vytvořte neefektivní implementaci Fibonacciho řady a proveďte profilování obou implementací.

Úkoly:

1. Napsat unit testy ověřující korektnost dodané implementace Fibonacciho řady
2. Napsat funkci řešící až kvadratické rovnice s jednou proměnnou v reálných číslech, která splní zadanou sadu testů
3. Vytvořit vlastní implementaci Fibonacciho řady, která bude neefektivní, ale bude splňovat vaši sadu testů
4. Provést profilování obou implementací Fibonacciho řady a shrnout výsledky

Motivace a zadání

Prostředím je jazyk Ruby a jeho testovací framework `Test::Unit`. Jako referenční bude bráno Ruby na serveru *ivs.fit.vutbr.cz* – **verze 1.9.3**.

Za projekt je možno celkem získat až 20 bodů.

Generátor Fibonacciho řady

$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$

Třída `FibonacciSequence` umožňuje generovat členy řady a získat člen s určitým indexem. Po vytvoření je aktuální člen (`current`) i index aktuálního členu (`current_idx`) nedefinovaný (`nil`) – nezačalo se generovat. Opakovaným voláním metody `next` lze získat hodnoty v posloupnosti. Generátor lze nastavit do výchozího stavu (`reset`). Generátor také může být nastaven na člen s požadovaným indexem.

Generátor Fibonacciho řady

- Třída nabízí následující metody:
  - `next` – vrátí následující člen řady
  - `current` – aktuální člen
  - `current idx` – index aktuálního členu řady
  - `reset` – nastaví řadu do výchozího stavu (`current` i `current_idx` vrací nil), volání `next` vrátí 0. člen
  - indexace (`sequence[i]`) – vrátí člen se zadaným indexem (`[i]`)
- Implementace je v souboru `fib-sequence.rb`, testy budou v souboru `fib-sequence_test.rb`.

1. úkol

- 8 bodů
- Do souboru fib-sequence_test.rb doplňte testy třídy FibonacciSequence
- Testy třídy musejí pokrývat alespoň většinu běžných stavů, testy by však neměly být příliš „nabubřelé“ a překombinované
- Přehled funkcí assert *, sloužících k ustanovení tvrzení, které mají v testech platit, najdete na http://ruby-doc.org/stdlib-1.9.3 ve třídě Test::Unit::Assertions
- http://ruby-doc.org se Vám celkově může hodit (nezapomeňte na správnou verzi)

2. úkol

5 bodů

Implementujte statickou metodu `Equation.solve_quadratic(a, b, c)`, která umí řešit až kvadratické rovnice s jednou proměnnou v reálných číslech.

Výsledek je vždy vrácen v poli (např. [3, 4] nebo [-1]).

Funkce musí splnit danou sadu testů, které v našem případě fungují jako specifikace. Testy se nacházejí v souboru `equation-test.rb`.

3. úkol

2 body

Implementujte neefektivní (pomalou) verzi třídy `FibonacciSequence`.

Funkce musí splnit vámi vytvořenou sadu testů (`fib-sequence_test.rb`).

4. úkol

5 bodů

Proveďte profilování neefektivní (`fib-sequence-slow.rb`) i vzorové (`fib-sequence.rb`) implementace Fibonacciho řady např. metodou „code instrumentation“ a výsledky profilování diskutujte v krátké textové zprávě o profilování. Profilování provádějte s velkým množstvím vstupů (různé členy Fibonacciho řady).

Textová zpráva by měla obsahovat několik vysvětlujících vět (cca 5) a data, která vaše tvrzení podporují. Textovou zprávu odevzdávejte v PDF (`profiling.pdf`).

Pokud nemáte nápad, jak začít s profilováním v Ruby, zkuste si přečíst např. http://www.synbioz.com/blog/optimize_ruby_code nebo vyhledávejte na internetu sami. Není podmínkou použít sofistikované profilovací nástroje, ale sepsat zprávu, která diskutuje profilování a podává důkazy (data z profilování) o vašich tvrzeních (kde je kód neefektivní, proč, o kolik se jaká část zlepší jinou implementací)...

Odevzdání

- Odevzdává se přes **WIS do úterý 29. března**.
- V archivu zip, jehož jméno bude váš login (např. `xskoda06.zip`), budou 4 soubory:
  - Testy pro třídu Fibonacciho řady (`fib-sequence_test.rb`)
  - Implementace funkce řešící rovnice (`equation.rb`)
  - Neefektivní implementace Fibonacciho řady (`fib-sequence_slow.rb`)
  - Zpráva o profilování v PDF (`profiling.pdf`)