

# IVS

## Praktické aspekty vývoje software

### Projekt 1

#### Definice testů a profiling

xtotha01  
Adrián Tóth

#### Správa o profilování:

Vytvoril som si program pomocou ktorého som vykonal profiling.

```
1 require_relative 'fib-sequence'
2
3 @seq = FibonacciSequence.new
4 @seq.[](50)
5 @seq.[](100)
6 @seq.[](500)
7 @seq.[](1000)
```

```
1 require_relative 'fib-sequence_slow'
2
3 @seq = FibonacciSequence.new
4 @seq.[](50)
5 @seq.[](100)
6 @seq.[](500)
7 @seq.[](1000)
```

Najskôr som vykonal profiling na fib-sequence.rb a následne potom na fib-sequence\_slow.rb.

#### Výsledné hodnoty:

##### fib-sequence.rb

```
xtotha01@ivs:~/proj1$ time ruby -rprofile fib-prof.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
71.43 0.05 0.05 1001 0.05 0.05 FibonacciSequence#next
28.57 0.07 0.02 4 5.00 17.50 FibonacciSequence#[]
0.00 0.07 0.00 3005 0.00 0.00 Array#[]
0.00 0.07 0.00 1 0.00 0.00 Kernel.require_relative
0.00 0.07 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 0.07 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 0.07 0.00 1 0.00 0.00 Class#new
0.00 0.07 0.00 1009 0.00 0.00 Fixnum#<
0.00 0.07 0.00 4 0.00 0.00 Fixnum#<=
0.00 0.07 0.00 1091 0.00 0.00 Fixnum#+
0.00 0.07 0.00 6 0.00 0.00 Module#method_added
0.00 0.07 0.00 2 0.00 0.00 Kernel.nil?
0.00 0.07 0.00 4 0.00 0.00 IO#set_encoding
0.00 0.07 0.00 999 0.00 0.00 NilClass#nil?
0.00 0.07 0.00 1998 0.00 0.00 Fixnum#-
0.00 0.07 0.00 999 0.00 0.00 Array#[]=
0.00 0.07 0.00 4 0.00 0.00 Fixnum#>=
0.00 0.07 0.00 4 0.00 0.00 FibonacciSequence#current
0.00 0.07 0.00 1 0.00 0.00 Class#inherited
0.00 0.07 0.00 909 0.00 0.00 Bignum#+
0.00 0.07 0.00 1 0.00 70.00 #toplevel

real 0m0.093s
user 0m0.085s
sys 0m0.009s
xtotha01@ivs:~/proj1$
```

##### fib-sequence\_slow.rb

```
xtotha01@ivs:~/proj1$ time ruby -rprofile fib-prof.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
77.12 3.54 3.54 999 3.54 4.52 Range#each
17.65 4.35 0.81 412686 0.00 0.00 Bignum#+
3.70 4.52 0.17 87815 0.00 0.00 Bignum#+
1.31 4.58 0.06 1001 0.06 4.58 FibonacciSequence#next
0.22 4.59 0.01 4 2.50 1147.50 FibonacciSequence#[]
0.00 4.59 0.00 1007 0.00 0.00 Array#[]
0.00 4.59 0.00 1 0.00 0.00 Class#new
0.00 4.59 0.00 1009 0.00 0.00 Fixnum#<
0.00 4.59 0.00 4 0.00 0.00 Fixnum#<=
0.00 4.59 0.00 1 0.00 0.00 FibonacciSequence#reset
0.00 4.59 0.00 1 0.00 0.00 FibonacciSequence#initialize
0.00 4.59 0.00 2 0.00 0.00 Kernel.nil?
0.00 4.59 0.00 1 0.00 0.00 Kernel.require_relative
0.00 4.59 0.00 999 0.00 0.00 NilClass#nil?
0.00 4.59 0.00 999 0.00 0.00 Fixnum#-
0.00 4.59 0.00 6 0.00 0.00 Module#method_added
0.00 4.59 0.00 999 0.00 0.00 Array#[]=
0.00 4.59 0.00 4 0.00 0.00 Fixnum#>=
0.00 4.59 0.00 4 0.00 0.00 FibonacciSequence#current
0.00 4.59 0.00 1 0.00 0.00 Class#inherited
0.00 4.59 0.00 4 0.00 0.00 IO#set_encoding
0.00 4.59 0.00 1 0.00 4590.00 #toplevel

real 0m4.781s
user 0m4.606s
sys 0m0.176s
xtotha01@ivs:~/proj1$
```

Z hore uvedených hodnôt z profilingu vyplýva že fib-sequence.rb je rýchlejší pretože skončil o 4,688s skôr ako fib-sequence\_slow.rb. Časový rozdiel môžeme zbadat' aj pri funkcii [](n), ktorý sa líši o 1130 ms/call. Funkcia next sa líši o 4,53ms/call. Čas trvania funkcie v fib-sequence\_slow.rb bol približne 4,59s pričom u fib-sequence.rb bol 0,07s. Z týchto hodnôt môžeme usúdiť, že čím väčší člen Fibonacciho postupnosti chceme vypočítať tým väčší čas sa spotrebuje pri oboch implementáciách, pričom pri fib-sequence\_slow.rb omnoho viac ako pri fib-sequence.rb.