Zadanie 1

Prosty program testujący działanie profilera.

In [2]:
```
function function1()
    for  i = 1:100000
        string("Test1")
    end
end

function function2()
    for  i = 1:10000
        string("Test2")
    end
end

function prog()
    for j = 1:500
        function1()
        function2()
    end
end
```

Wynik dla wywołania **Profile.print(format=:flat)**

```
julia> Profile.print(format=:flat)
 Count File                          Line Function
   422 ./<missing>                     -1 anonymous
   422 ./REPL.jl                       66 eval_user_input(::Any, ::Base.REPL....
   422 ./REPL.jl                       97 macro expansion
   422 ./boot.jl                      235 eval(::Module, ::Any)
   422 ./event.jl                      73 (::Base.REPL.##1#2{Base.REPL.REPLBa...
   422 ./profile.jl                    23 macro expansion
    52 ./strings/string.jl            320 string(::String, ::Vararg{String,N}...
   389 ...ulia/lab03/simpleFun.jl       3 function1()
    33 ...ulia/lab03/simpleFun.jl       9 function2()
   389 ...ulia/lab03/simpleFun.jl      15 prog()
    33 ...ulia/lab03/simpleFun.jl      16 prog()
```

Wynik dla **ProfileView.view()**

Czas spędzony w poszczególnych funkcjach przestaje odpowiadać rzeczywistym wartościom dla **delay** > ~0.08

Zadanie 2

Początkowe wyniki testów z wykorzystaniem BenchmarkTools

```
BenchmarkTools.Trial:
  memory estimate:  6.53 GiB
  allocs estimate:  121828690
  --------------
  minimum time:     9.861 s (11.16% GC)
  median time:      9.861 s (11.16% GC)
  mean time:        9.861 s (11.16% GC)
  maximum time:     9.861 s (11.16% GC)
  --------------
  samples:          1
  evals/sample:     1

julia>
```

Po wykonywaniu modyfikacji w kolejnych etapach otrzymywałem poniższe wyniki:

- zamiana zmiennych globalnych na **const ... = Int64 (...)**
- deklaracja typów pól dla typów złożonych oraz argumentów funkcji

```
BenchmarkTools.Trial:
  memory estimate:  3.79 GiB
  allocs estimate:  5556767
  --------------
  minimum time:     1.887 s (11.10% GC)
  median time:      1.899 s (11.30% GC)
  mean time:        1.917 s (12.29% GC)
  maximum time:     1.964 s (14.38% GC)
  --------------
  samples:          3
  evals/sample:     1
```

- podział funkcji **graph_to_str** na mniejsze funkcje – eliminacja nieoptymalnej **isa()** na dopasowania

```
BenchmarkTools.Trial:
  memory estimate:  2.63 GiB
  allocs estimate:  5367045
  --------------
  minimum time:     1.895 s (10.52% GC)
  median time:      1.907 s (10.74% GC)
  mean time:        1.910 s (10.65% GC)
  maximum time:     1.929 s (10.62% GC)
  --------------
  samples:          3
  evals/sample:     1

julia> @benchmark Graphs.test_graph()
```

- zamiana indeksowania na „column major” w funkcji **convert_to_graph**

```
BenchmarkTools.Trial:
  memory estimate:  2.64 GiB
  allocs estimate:  5366116
  --------------
  minimum time:     1.580 s (15.35% GC)
  median time:      1.596 s (15.28% GC)
  mean time:        1.598 s (15.36% GC)
  maximum time:     1.620 s (15.52% GC)
  --------------
  samples:          4
  evals/sample:     1

julia>
```

- dodanie klauzul **@simd** oraz **@inbound** przed wywołaniami funckji **for**

```
BenchmarkTools.Trial:
  memory estimate:  2.10 GiB
  allocs estimate:  3630184
  --------------
  minimum time:     1.361 s (18.07% GC)
  median time:      1.378 s (17.80% GC)
  mean time:        1.376 s (17.91% GC)
  maximum time:     1.385 s (17.69% GC)
  --------------
  samples:          4
  evals/sample:     1

julia> @benchmark Graphs.test_graph()
```

- wywołanie **rand()**, **randString()** bezpośrednio z **generate_random_nodes**
- zmiana sposobu przekazywania wartości pomiędzy **graph_to_string** a **node_to_str**
- zamiana tablicy reprezentującej macierz **A** (N x N) na tablice bitów, ponieważ przyjmuje ona i tak wartości 0 lub 1, oraz wykorzystanie związanej z nią funkcji **falses()**.

  Końcowy wynik modyfikacji:

```
BenchmarkTools.Trial:
  memory estimate:  1.06 GiB
  allocs estimate:  3704981
  --------------
  minimum time:     985.232 ms (17.17% GC)
  median time:      995.207 ms (17.62% GC)
  mean time:        993.267 ms (17.51% GC)
  maximum time:     996.294 ms (17.75% GC)
  --------------
  samples:          6
  evals/sample:     1

julia>
```