# IVS – Profiling

*BigyTeam*

14. 4. 2019

## 1. Output of the Profiler

Using profiler module for Visual Studio 2019, method Instrumentation.

### 1.1. Output for 10 inputs

| Název funkce | Počet volání | % uplynulého celkového č... | % uplynulého výhradního času | Průměrný uplynulý celkový čas | Průměrný uplynulý výhrad... | Název modulu |
|---|---|---|---|---|---|---|
| CustomMath.MathFunctions.Add(float64,float64) | 20 | 1,09 % | 1,09 % | 62 825,00 | 62 825,00 | CustomMath.dll |
| CustomMath.MathFunctions.Divide(float64,float64) | 2 | 0,94 % | 0,94 % | 538 916,00 | 538 916,00 | CustomMath.dll |
| CustomMath.MathFunctions.Multiply(float64,float64) | 3 | 0,73 % | 0,73 % | 281 376,00 | 281 376,00 | CustomMath.dll |
| CustomMath.MathFunctions.Power(float64,int32) | 11 | 1,03 % | 1,03 % | 108 185,00 | 108 185,00 | CustomMath.dll |
| CustomMath.MathFunctions.Root(float64,float64) | 1 | 1,44 % | 1,44 % | 1 665 133,00 | 1 665 133,00 | CustomMath.dll |
| CustomMath.MathFunctions.Subtract(float64,float64) | 2 | 0,74 % | 0,74 % | 423 772,00 | 423 772,00 | CustomMath.dll |
| get_Current() | 20 | 0,00 % | 0,00 % | 71,00 | 71,00 | Profiling.exe |
| MoveNext() | 22 | 0,01 % | 0,01 % | 411,00 | 411,00 | Profiling.exe |
| Profiling.Profiling.Main(string[]) | 1 | 100,00 % | 0,22 % | 115 265 678,00 | 258 862,00 | Profiling.exe |
| System.Collections.Generic.List`1..ctor() | 1 | 0,08 % | 0,08 % | 95 320,00 | 95 320,00 | mscorlib.dll |
| System.Collections.Generic.List`1.Add(!0) | 10 | 0,01 % | 0,01 % | 1 249,00 | 1 249,00 | mscorlib.dll |
| System.Collections.Generic.List`1.GetEnumerator() | 2 | 0,00 % | 0,00 % | 506,00 | 506,00 | mscorlib.dll |
| System.Console.WriteLine(float64) | 1 | 2,20 % | 2,20 % | 2 535 786,00 | 2 535 786,00 | mscorlib.dll |
| System.Double.Parse(string) | 10 | 0,20 % | 0,20 % | 23 609,00 | 23 609,00 | mscorlib.dll |
| System.IDisposable.Dispose() | 2 | 0,00 % | 0,00 % | 141,00 | 141,00 | mscorlib.dll |
| System.IO.File.ReadAllLines(string) | 1 | 88,06 % | 88,06 % | 101 508 003,00 | 101 508 003,00 | mscorlib.dll |
| System.Linq.Enumerable.Count(class System.Collections... | 3 | 3,23 % | 3,23 % | 1 242 059,00 | 1 242 059,00 | System.Core.dll |

### 1.2. Output for 100 inputs

| Název funkce | Počet volání | % uplynulého celkového č... | % uplynulého výhradního času | Průměrný uplynulý celkový čas | Průměrný uplynulý výhrad... | Název modulu |
|---|---|---|---|---|---|---|
| CustomMath.MathFunctions.Add(float64,float64) | 200 | 9,66 % | 9,66 % | 11 403,00 | 11 403,00 | CustomMath.dll |
| CustomMath.MathFunctions.Divide(float64,float64) | 2 | 5,06 % | 5,06 % | 597 879,00 | 597 879,00 | CustomMath.dll |
| CustomMath.MathFunctions.Multiply(float64,float64) | 3 | 3,59 % | 3,59 % | 282 652,00 | 282 652,00 | CustomMath.dll |
| CustomMath.MathFunctions.Power(float64,int32) | 101 | 5,11 % | 5,11 % | 11 936,00 | 11 936,00 | CustomMath.dll |
| CustomMath.MathFunctions.Root(float64,float64) | 1 | 8,75 % | 8,75 % | 2 066 087,00 | 2 066 087,00 | CustomMath.dll |
| CustomMath.MathFunctions.Subtract(float64,float64) | 2 | 3,55 % | 3,55 % | 418 810,00 | 418 810,00 | CustomMath.dll |
| get_Current() | 200 | 0,00 % | 0,00 % | 4,00 | 4,00 | Profiling.exe |
| MoveNext() | 202 | 0,05 % | 0,05 % | 59,00 | 59,00 | Profiling.exe |
| Profiling.Profiling.Main(string[]) | 1 | 100,00 % | 1,17 % | 23 615 253,00 | 277 043,00 | Profiling.exe |
| System.Collections.Generic.List`1..ctor() | 1 | 0,43 % | 0,43 % | 101 094,00 | 101 094,00 | mscorlib.dll |
| System.Collections.Generic.List`1.Add(!0) | 100 | 0,17 % | 0,17 % | 408,00 | 408,00 | mscorlib.dll |
| System.Collections.Generic.List`1.GetEnumerator() | 2 | 0,02 % | 0,02 % | 2 024,00 | 2 024,00 | mscorlib.dll |
| System.Console.WriteLine(float64) | 1 | 17,58 % | 17,58 % | 4 150 770,00 | 4 150 770,00 | mscorlib.dll |
| System.Double.Parse(string) | 100 | 1,75 % | 1,75 % | 4 142,00 | 4 142,00 | mscorlib.dll |
| System.IDisposable.Dispose() | 2 | 0,00 % | 0,00 % | 87,00 | 87,00 | mscorlib.dll |
| System.IO.File.ReadAllLines(string) | 1 | 23,12 % | 23,12 % | 5 460 240,00 | 5 460 240,00 | mscorlib.dll |
| System.Linq.Enumerable.Count(class System.Collections.Generic.I... | 3 | 19,99 % | 19,99 % | 1 573 353,00 | 1 573 353,00 | System.Core.dll |

### 1.3. Output for 1000 inputs

| Název funkce | Počet volání | % uplynulého celkového č... | % uplynulého výhradního... | Průměrný uplynulý celkový... | Průměrný uplynulý výhrad... | Název modulu |
|---|---|---|---|---|---|---|
| CustomMath.MathFunctions.Add(float64,float64) | 2 000 | 6,67 % | 6,67 % | 632,00 | 632,00 | CustomMath.dll |
| CustomMath.MathFunctions.Divide(float64,float64) | 2 | 7,86 % | 7,86 % | 746 202,00 | 746 202,00 | CustomMath.dll |
| CustomMath.MathFunctions.Multiply(float64,float64) | 3 | 4,50 % | 4,50 % | 285 071,00 | 285 071,00 | CustomMath.dll |
| CustomMath.MathFunctions.Power(float64,int32) | 1 001 | 6,15 % | 6,15 % | 1 167,00 | 1 167,00 | CustomMath.dll |
| CustomMath.MathFunctions.Root(float64,float64) | 1 | 8,86 % | 8,86 % | 1 682 539,00 | 1 682 539,00 | CustomMath.dll |
| CustomMath.MathFunctions.Subtract(float64,float64) | 2 | 4,29 % | 4,29 % | 407 533,00 | 407 533,00 | CustomMath.dll |
| get_Current() | 2 000 | 0,28 % | 0,28 % | 26,00 | 26,00 | Profiling.exe |
| MoveNext() | 2 002 | 0,35 % | 0,35 % | 32,00 | 32,00 | Profiling.exe |
| Profiling.Profiling.Main(string[]) | 1 | 95,50 % | 0,00 % | 18 134 835,00 | 0,00 | Profiling.exe |
| System.Collections.Generic.List`1..ctor() | 1 | 0,51 % | 0,51 % | 96 124,00 | 96 124,00 | mscorlib.dll |
| System.Collections.Generic.List`1.Add(!0) | 1 000 | 0,41 % | 0,41 % | 77,00 | 77,00 | mscorlib.dll |
| System.Collections.Generic.List`1.GetEnumerator() | 2 | 0,01 % | 0,01 % | 1 013,00 | 1 013,00 | mscorlib.dll |
| System.Console.WriteLine(float64) | 1 | 15,12 % | 15,12 % | 2 871 329,00 | 2 871 329,00 | mscorlib.dll |
| System.Double.Parse(string) | 1 000 | 5,36 % | 5,36 % | 1 017,00 | 1 017,00 | mscorlib.dll |
| System.IDisposable.Dispose() | 2 | 0,00 % | 0,00 % | 2,00 | 2,00 | mscorlib.dll |
| System.IO.File.ReadAllLines(string) | 1 | 18,04 % | 18,04 % | 3 426 037,00 | 3 426 037,00 | mscorlib.dll |
| System.Linq.Enumerable.Count(class System.Collections.Generic.IEnume... | 3 | 21,60 % | 21,60 % | 1 367 080,00 | 1 367 080,00 | System.Core.dll |

## 2. Summary

Seeing as the mostly used functions during the calculation of a *Standart deviation* are **addition** and **power**, the main focus during refactorization and streamlining of CustomMath library should be these two functions.

A lot of time is needed for correct parsing of numbers from input. Unfortunately, C#, the language we used, doesn't support number (float, integer) loading, it parses everything as a string when reading input. So, if we counted *Standart deviation* on slower computers, or with much bigger inputs, perhaps it would be more efficient to write the program in a different language (for exapmle C, as a fast and efficient language).

On the other hand, dynamic memory management in C# is simple and quick (supported by the output of the profiler, as storing a 1000 numbers took only 0,41% of processing time). The biggest advantage is simple dynamic memory manipulation (compared to C), saving a lot of time/work for developers and with almost zero risk of memory leaks.