

Teljesítmény Mérés

Tóth Zsolt

Miskolci Egyetem

2013

1 Bevezetés

- ## 2 Visual Studio
- Kód metrikák
 - Performance Explorer

- Elméletileg megoldhatóság
- Gyakorlati problémák
- Algoritmusok osztályozás
- Véges erőforrások
 - ▶ Tár
 - ▶ Idő

Ordo-jelölés:

Legyen $f, g : N \rightarrow C$ függvények, ekkor a $f = O(g)$ jelöli, hogy $\exists c > 0$ konstans és $n_0 \in Z_+$ küszöb úgy, hogy $\forall n > n_0$ esetén $|f(n)| \leq c |g(n)|$.

Tár

- Memória
- Virtuális Memória Kezelés
- "Véges"
- Gyakorlatilag végtelennek tekinthető

Idő

- Kritikus erőforrás
- Felhasználók türelmetlenek
- Számítás igényes feladatok
- Hatékonyság!
- Valós időben megoldható ?

Algoritmusok komplexitása

- Lineáris idejű algoritmusok $O(n)$ (Lineáris keresés, ...)
- Polinomiális idejű algoritmusok $O(n^2)$, $O(n^3)$, ... (Buborék rendezés, Gauss elimináció, ...)
- Logaritmikus idejű algoritmusok $O(\log n)$ (Logaritmikus keresés, keresés fában)

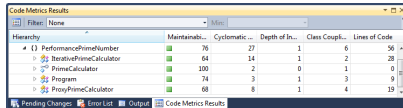
1 Bevezetés

2 Visual Studio

- Kód metrikák
- Performance Explorer

Code Metrics

- Szoftver mérési eszköz
- Forráskód elemzés
- Refactoring
- Kockázat feltárás
- Projekt aktuális állapota



The screenshot shows a 'Code Metrics Results' window with a table of metrics. The table has columns for Hierarchy, Maintainability, Cyclomatic complexity, Depth of Inheritance, Class Coupling, and Lines of Code. The data is as follows:

Hierarchy	Maintainability	Cyclomatic...	Depth of In...	Class Coup...	Lines of Code
PerformancePrimeNumber	76	27	1	6	56
IterativePrimeCalculator	64	14	1	2	28
PrimeCalculator	100	2	0	1	0
Program	74	3	1	3	9
ProxyPrimeCalculator	68	8	1	4	19

Mérőszámok

- Maintainability Index
- Cyclomatic Complexity
- Depth of Inheritance
- Class Coupling
- Lines of Code

Maintainability Index

- 0–100 közötti érték
- 100 ✓
- Relatív érték
- Kód karbantarthatósága

Depth of Inheritance

- Öröklődés mélysége
- Mélyebb hierarchia → bonyolultabb

Lines of Code

- Kódsorok száma
- Hosszabb kód
 - ▶ Bonyolultság
 - ▶ Tervezési hiba
 - ▶ Felbontható
 - ▶ Nehéz karbantartani

Class Coupling

- Osztályok összehasonlítása
 - ▶ Paraméterek
 - ▶ Lokális változók
 - ▶ Visszatérési értékek
 - ▶ Metódus hívások
 - ▶ Ősosztály
 - ▶ Interface
- Alacsony érték ✓
- Magas érték → tervezési hiba

Cyclomatic Complexity

- Strukturális komplexitás
- Alacsony érték ✓
- Vezérlés mint irányított gráf
- Bonyolultabb vezérlés
 - ▶ Bonyolultabb tesztelés
 - ▶ Nehezebb karbantarthatóság

$$M = E - N + 2P$$

M bonyolultság

E Élek száma

N Csomópontok száma

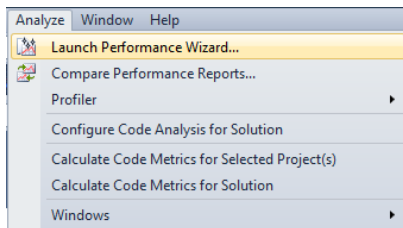
P Összefüggő
komponensek száma

1 Bevezetés

2 Visual Studio

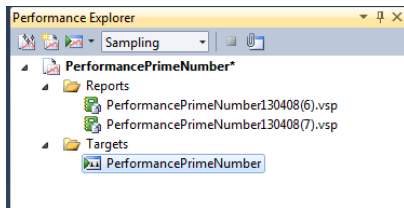
- Kód metrikák
- Performance Explorer

- Visual Studio része
- Alkalmazások vizsgálata
- Mérések
- Futásidő
- Metódus hívások
- Report készítés

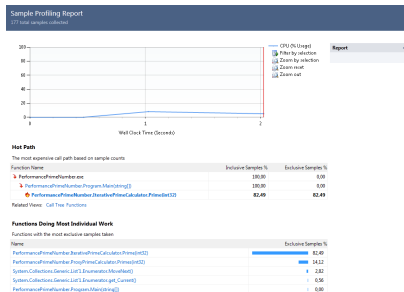


Performance Explorer

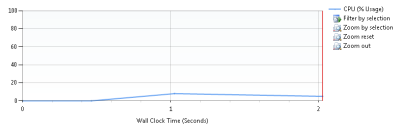
- Mérések futtatása
- Elemzés
- Varázsló
- Performance Session
- Release mód



- Mérési eredmények
- Részei
 - 1 Összegző idővonal
 - 2 Hot Path
 - 3 Függvények



- Összefoglaló eredmények
- CPU használat
- Időbeli változás
- Számítás igényes szakaszok
- I/O igényes szakaszok



Hot Path

- A legköltségesebb út
- A legaktívabb rész
- Inclusive
- Tartalmazhat System függvényeket

Hot Path

The most expensive call path based on sample counts

Function Name	Inclusive Samples %	Exclusive Samples %
PerformancePrimeNumber.exe	100,00	0,00
PerformancePrimeNumber.Program.Main<string>()	100,00	0,00
PerformancePrimeNumber.IterativePrimeCalculator.PrimeGet(132)	82,49	82,49

Related Views: [Call Tree Functions](#)

- Részletező eredmények
- Metódusonként

Functions Doing Most Individual Work

Functions with the most exclusive samples taken

Name	Exclusive Samples %
Performance\PrimeNumber.IterativePrimeCalculator.Prime(int32)	82.49
Performance\PrimeNumber.ProsyPrimeCalculator.Primes(int32)	14.12
System.Collections.Generic.List<T>.Enumerator.MoveNext()	2.82
System.Collections.Generic.List<T>.Enumerator.GetCurrent()	0.56
Performance\PrimeNumber.Program.Main(string[])	0.00

Inclusive, Exclusive

Inclusive

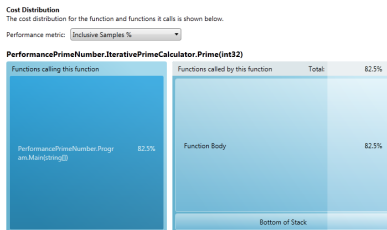
- Metódudusban töltött idő, a benne történő metódudushívások idejével együtt.

Exclusive

- Metódudusban töltött idő, a benne történő metódudushívások ideje nélkül.

Következmény: Inclusive \geq Exclusive

- Metódusok egyenkénti vizsgálata
- Részei
 - ▶ Költség eloszlás
 - ▶ Kód nézet
- Költség
- Honnan hívták ←
- Miket hívott →



Metódus elemzés

- Forrás kód
- Egyes műveletek költségei
- Tippek

Function Code View

C:\Users\01902010\Documents\Visual Studio 2010\Projects\Performance\FiveNumber\Performance\FiveNumber\DerivativeFiveCalculator.cs

```
24 |
25 |     9-8 N
26 |     7-9 N
27 |     33-3 N
28 |
29 |
30 |
31 |
32 |     9-8 N
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
```

```
bool isPrime = true;
for (int i = 2; i < result; i++)
{
    if (result % i == 0)
    {
        isPrime = false;
        break;
    }
}
if (isPrime)
{
    index--;
    if (index == 0)
    {
        break;
    }
}
result++;
}
return result;
}
```