

Comparação de desempenho entre linguagens de programação

Mattheus Medina Ayala

Cadeira de Paradigmas de Linguagens de programação
Centro Universitário Ritter dos Reis (UniRitter)
91240-261 – Porto Alegre – RS – Brazil

mataus94@outlook.com

1. Introdução

Primeiramente este artigo visa fazer uma análise das características funcionais e de desempenho das seguintes linguagens de programação: C, C# e Java. Para isso começaremos verificando a precisão numérica de seus tipos e em seguida realizaremos testes de performance para demonstrar a velocidade de cada uma diante da resolução de um mesmo teste.

2. Análise dos diferentes tipos e de suas precisão numérica

Análise dos diferentes tipos e a precisão numérica com short int, float e double na realização de operações de multiplicação de matrizes.

Para gerar as matrizes, criei uma classe Matriz que possui uma função que gera matrizes aleatórias com valores de 0 a 99.

Link para os códigos usados: [Trabalho-de-paradigmas-de-linguagens-de-programação](#)

C

Matrizes short

Matriz 1:

3	14	88
76	1	8
94	63	80

Matriz 2:

52	9	63
79	44	32
82	59	22

Matriz resultante:

8478 5835 2573
4687 1200 4996
16425 8338 9698

Matrizes float

Matriz 1:

18.924528 18.512527 65.144814
35.758537 32.364880 84.307388
83.178200 84.954376 85.213783

Matriz 2:

32.358776 64.738914 55.586414
94.268623 85.164955 41.380047
32.145145 54.954678 11.542101

Matriz resultante:

4451.614746 6381.784180 2569.903809
6918.168457 9704.408203 4300.033691
13439.287109 17302.898438 9122.540039

Matrizes double

Matriz 1:

57.5884273812066994 16.7668691061128570 70.0827051606799500
88.9187292092654218 5.5696279793694874 68.4682760093997018
34.1319009979552632 65.0105288857692187 22.1655934324167596

Matriz 2:

67.2017578661458117 68.8222907193212734 23.4626300851466425
57.5518051698355038 91.9980468153935362 15.6376842555009610
18.2287057100131236 25.4982146671956578 85.1466414380321623

Matriz resultante:

6112.5241446027685015 7292.8705609144735718 7580.6779411890320262
7544.1251082708786271 8377.8043272035047266 8003.2070820928784087
6439.2471183649286104 8895.0603527764033061 3704.7641274678462651

matrizes short:

matriz 1:

26 29 24

57 32 67

67 54 96

matriz 2:

30 35 20

40 43 47

11 35 0

matriz resultante short:

2204 2997 1883

3727 5716 2644

5226 8027 3878

matrizes float:

matriz 1:

16,967916 31,948849 33,480732

8,251378 11,749776 85,27158

68,11375 64,03911 63,766613

matriz 2:

13,763884 20,245045 43,297615

77,15929 47,523853 31,02057

64,83069 70,62601 64,44923

matriz resultante float:

4869,2734 4226,459 3883,5493

6548,3906 6747,836 6217,4375

10012,755 8925,932 9045,401

matrizes double:

matriz 1:

4,4607484026166215 35,55940728645892 24,196455835973307

24,033968799286054 60,29895525916598 58,41696461693121

27,219575051954735 88,8078923354012 13,192411613585708

matriz 2:

30,90970104830103 49,17364204808631 51,44696345116052
92,37771712129785 29,150477004200482 46,93169463911483
83,0715820753733 9,512501713642664 19,51653561386021

matriz resultante double:

5432,815133797288 1486,0937572075375 2370,586196402214
11165,952293000735 3495,272563414959 5206,503640083976
10141,133787946175 4052,770901701781 5825,73953857172

JAVA

matrizes short:

matriz 1:

62 88 31
89 35 23
16 97 3

matriz 2:

31 74 93
85 81 56
91 89 32

matriz resultante short:

12223 14475 11686
7827 11468 10973
9014 9308 7016

matrizes float:

matriz 1:

27,329762 83,448189 23,345131
78,989410 13,053399 82,135529
42,939293 64,316246 18,594730

matriz 2:

65,265343 53,949268 20,403177
37,928104 53,302868 39,512085
90,634850 53,940834 72,361664

matriz resultante float:

7064,600098 7181,704102 5544,118164
13094,703125 9387,664063 8070,865234
6927,171387 6747,799316 4762,913086

matrizes double:

matriz 1:

93.88774989926463 16.153752666965314 49.09925647773824
41.57040631901878 10.954917259416785 52.62313050420524
22.61689187040793 40.27061890260847 11.679460854695357

matriz 2:

33.83501980155982 36.38303007545273 38.60324194534434
41.774946298716934 80.92910070154834 89.0995321003533
43.232755827927114 17.636401562441996 61.050586936461414

matriz resultante double:

5974.212193785572 5589.163708252993 8061.201755466509
4139.21955317136 3327.1116064630755 5793.50345862721
2952.4812056904307 4287.91969128342 5174.216590819683

3. Realize experimentos comparando os tempos de execução das linguagens

O teste foi realizado a partir da execução da rotina de criação de matrizes 10 vezes. Ao fim da execução era apresentado o tempo total de execução e a média de tempo para cada repetição.

Vale ressaltar que cada programa possui funções próprias de geração de números aleatórios e que para as linguagens Java e C#, utilizei o paradigma da Orientação a Objetos enquanto que para a linguagem C o paradigma Estruturado. Além do que a forma de implementação de cada programa também usufruiu de algumas diferenças entre si para o funcionamento adequado dos mesmos.

Os resultados obtidos no console foram:

C

O tempo total de execucao da rotina 10x foi de: 1.000000 segundos

O tempo medio de execucao de cada rotinas foi de: 0.1000 segundos

C#

O tempo de execução de 10 rotinas foi de: 210 milisegundos

O tempo médio de execução de cada rotina foi de: 21 milisegundos

JAVA

O tempo de execução de 10 rotinas foi de: 87 milisegundos

O tempo médio de execução de cada rotina foi de: 8 milisegundos

4. Relato sobre os dados encontrados

Apesar de ser conhecidamente a linguagem mais rápida, o C nos testes realizados ficou em último lugar. O motivo do ocorrido não ficou bem definido, mas acredito que tenha a ver com a forma de implementação de cada programa visto que ao tirar funções de menos importância para o mesmo, como as de print das matrizes, o resultado melhorou consideravelmente (70% mais rápido). Ainda assim o C foi eliminado com o último lugar nos testes.

O Java ficou com a primeira posição entre as três linguagens sendo 2.4x mais rápido do que o segundo lugar C#. Sendo assim o mesmo foi eliminado com o primeiro lugar no ranking de velocidade.

Tendo em vista que, os testes foram analisados cuidadosamente, o principal limitador percebido durante os mesmos foi a velocidade de escrita no console. Uma vez que o C que deveria ser o mais rápido mostrava sinais claros de fadiga intervalos rítmicos de escrita no prompt (isto é de tempos em tempos era perceptível uma engasgada na execução do programa).

Por fim vale ressaltar que durante a execução do programa escrito em JAVA a escrita no prompt foi quase instantânea e que o mesmo também conseguiu os melhores resultados nos testes. Por isso acredito que esta característica deve ter sido um dos fatores de maior influência para o teste.

3. Algoritmo para a demonstração da Sequência de Fibonacci, utilizando recursividade e não utilizando recursividade

JAVA

Fibonacci de 35

Fibonacci iterativo: 5702887

O tempo de execução do fibonacci iterativo foi de: 0 milisegundos

Fibonacci Recursivo: 5702887

O tempo de execução do fibonacci recursivo foi de: 23 milisegundos

O tempo total de execução da rotina de 100 repetições foi de: 2440 milisegundos

C

Fibonacci de 35

Fibonacci iterativo: 5702887

Tempo de Execucao em segundos: 0.000000

Fibonacci Recursivo: 5702887

Tempo de Execucao em segundos: 0.000000

O tempo total de execucao da rotina de 100 repeticoes foi de: 5 milisegundos

C#

Fibonacci de 35

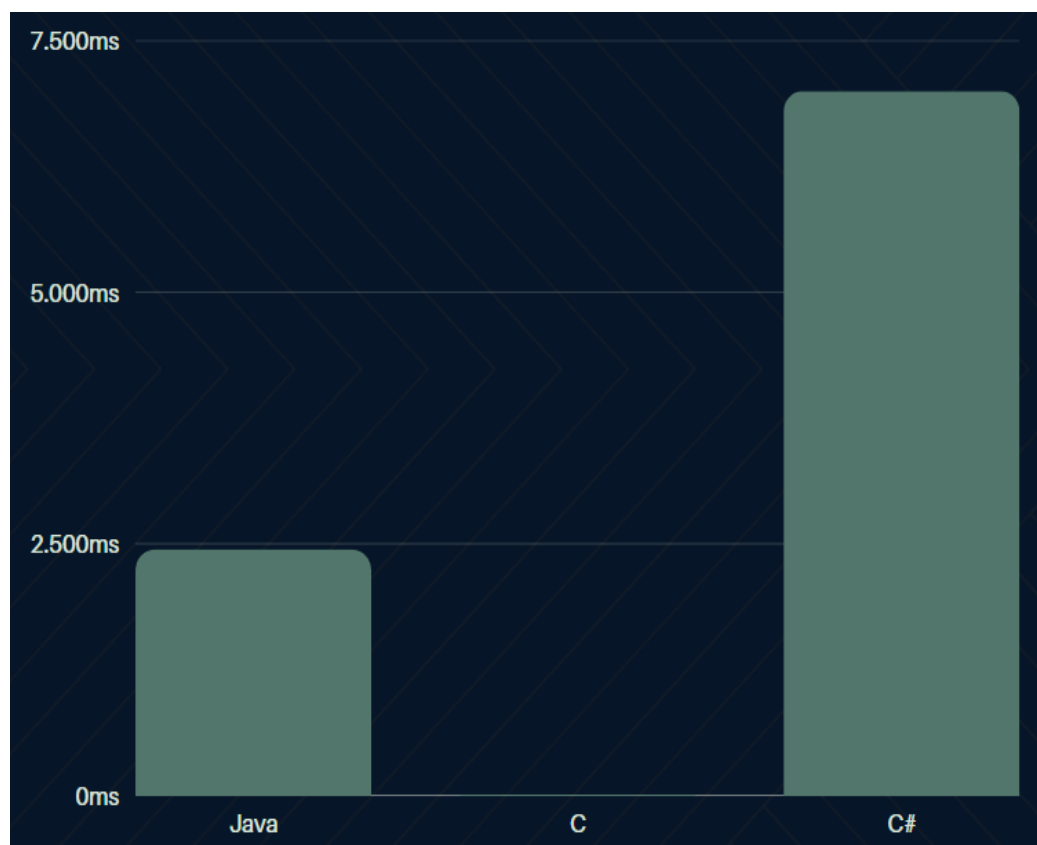
Fibonacci iterativo: 5702887

Tempo de execução em segundos: 72 milisegundos

Fibonacci Recursivo: 5702887

Tempo de execução em segundos: 66 milisegundos

O tempo total de execução de 100 repetições da rotina foi de: 6991 milisegundos



Códigos desenvolvidos e usados no trabalho:

Linguagem C

[Trabalho-de-paradigmas-de-linguagens-de-programa-o/A1/C at main · Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o \(github.com\)](https://github.com/Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o/tree/main/A1/C)

Linguagem C#

[Trabalho-de-paradigmas-de-linguagens-de-programa-o/A1/CSHARP at main · Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o \(github.com\)](https://github.com/Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o/tree/main/A1/CSHARP)

Linguagem Java

[Trabalho-de-paradigmas-de-linguagens-de-programa-o/A1/Java at main · Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o \(github.com\)](https://github.com/Matthsh/Trabalho-de-paradigmas-de-linguagens-de-programa-o/tree/main/A1/Java)