

# Programação Estruturada

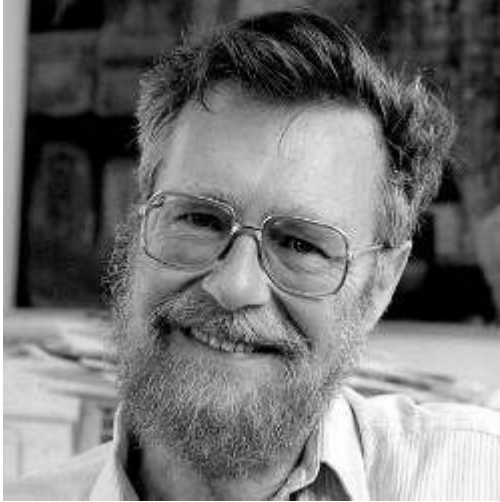
Prof. Daniel M. Martin (UFABC)

(baseado nos slides do Prof. Jesús P. M. Chalco)



O que é Ciência da Computação?





Computer Science is no  
more about computers than  
astronomy is about  
telescopes.

-- Edsger W. Dijkstra



# Algoritmo

# Algoritmo

Uma sequência finita de instruções bem definidas que devem ser seguidas para se resolver um problema.

# Elementos de um algoritmo

**Finitude** - a descrição de um algoritmo sempre deve ser finita.

**Desambiguidade** - nenhum passo pode ser ambíguo ou duvidoso.

**Entrada** - dados que o algoritmo recebe.

**Saída** - dados que o algoritmo produz.

# Problema x Instância

# Exemplo de problema computacional:

Encontrar o máximo divisor comum de dois números naturais  $a$  e  $b$ .



# Exemplo de problema computacional:

Encontrar o máximo divisor comum de dois números naturais  $a$  e  $b$ .

Exemplo de instância desse problema:  $a = 18$ ,  $b = 24$

# Exemplo de problema computacional:

Encontrar o máximo divisor comum de dois números naturais  $a$  e  $b$ .

Exemplo de instância desse problema:  $a = 18, \quad b = 24$

Outra instância desse problema:  $a = 703, \quad b = 779$

# Algoritmo de Euclides ( $\pm 300$ BC)

MDC( $a$ ,  $b$ )

**enquanto**  $a \neq b$  **faça:**

**se**  $a > b$  **então**

$a \leftarrow a - b$

**senão**

$b \leftarrow b - a$

**resposta:**  $a$





# Computação

# Computação

É o processo de se executar um determinado algoritmo com entradas determinadas e obter a saída correspondente.

# Computação

É o processo de se executar um determinado algoritmo com entradas determinadas e obter a saída correspondente.

Pode ser feito manualmente, com lápis e papel.

# Computação

É o processo de se executar um determinado algoritmo com entradas determinadas e obter a saída correspondente.

Pode ser feito manualmente, com lápis e papel.

Pode ser feito pelo computador.

# Computação

Para que um computador possa executar um algoritmo (fazer uma computação) é necessário codificar esse algoritmo em alguma linguagem que o computador (processador) entenda.





# Linguagem de programação



# Linguagem de programação

Consiste de conjuntos finitos de:

- **símbolos:** comandos, palavras-chave, caracteres, etc; e
- **regras de sintaxe:** descrevem de maneira precisa como combinar os símbolos de forma válida (i.e. descrevem a estrutura da linguagem).

# Linguagem de máquina

# Linguagem de máquina

É o conjunto de instruções que são compreendidas pelo processador.

# Linguagem de máquina

É o conjunto de instruções que são compreendidas pelo processador.

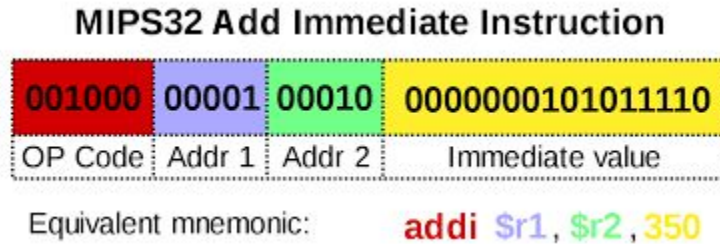
Varia conforme o processador.

# Linguagem de máquina

É o conjunto de instruções que são compreendidas pelo processador.

Varia conforme o processador.

Exemplo de instrução:



# Linguagem assembly

É um conjunto de instruções mnemônicas que se traduzem quase que diretamente em linguagem de máquina.

# Linguagem assembly

É um conjunto de instruções mnemônicas que se traduzem quase que diretamente em linguagem de máquina.

É uma linguagem de programação baixo nível (bem próxima à linguagem de máquina).



# Linguagem assembly

É um conjunto de instruções mnemônicas que se traduzem quase que diretamente em linguagem de máquina.

É uma linguagem de programação baixo nível (bem próxima à linguagem de máquina).

Depende do processador.

# Algoritmo de Euclides programado em Assembly

```
# encontra o MDC de 68 e 119
```

```
mov esi, 68
```

```
mov ebx, 119
```

```
jmp .L2
```

```
.L3:
```

```
mov ebx, edx
```

```
.L2:
```

```
mov eax, ebx
```

```
idiv esi
```

```
mov esi, ebx
```

```
test edx, edx
```

```
jne .L3
```



Linguagem de programação (alto nível)



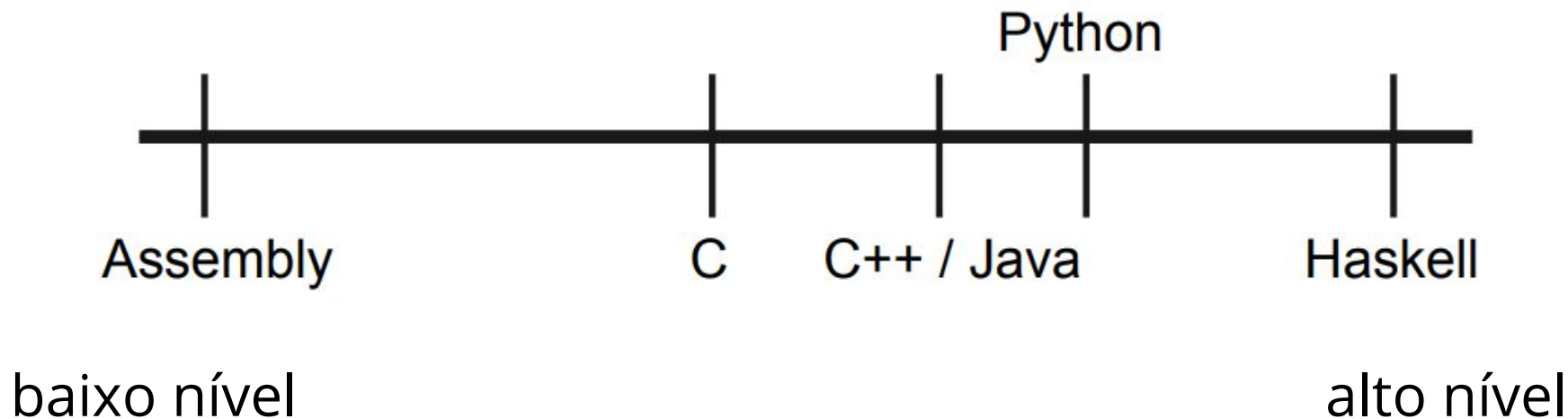
# Linguagem de programação (alto nível)

Os objetivos de uma linguagem de alto nível são:

- ser mais próxima da linguagem natural;
- ter número pequeno de instruções;
- dar o máximo de controle sobre o computador, sem a necessidade de entendê-lo completamente.

# Linguagem de programação (alto nível)

Não existe uma definição formal de “o quanto” uma linguagem é de alto nível, mas...



# Linguagens de programação se classificam em

- **Compiladas:** o código-fonte é traduzido para código de máquina e escrito em um arquivo executável.
- **Interpretadas:** o código-fonte é traduzido para instruções de máquina em tempo de execução.



# Programação Estruturada



# Programação Estruturada

“ **Structured programming** is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures, for and while loops — in contrast to using simple tests and jumps such as the `goto` statement, which could lead to ‘spaghetti code’. ”



# *Structured Programming Theorem*

**Teorema de Böhm-Jacopini: (informal)** qualquer função computável pode ser computada combinando-se blocos de código, recursivamente, de três maneiras possíveis: em sequência, por seleção (**if**) e em laços (**for** ou **while**).

# Linguagem C

# Linguagem C

É uma linguagem estruturada, desenvolvida nos Laboratórios BELL (1969-1972), por Dennis Ritchie.



Kenneth Thompson (ling. B)  
Dennis Ritchie (ling. C)

# The National Medal of Technology and Innovation 1998 Laureates



Fonte: <https://www.nationalmedals.org/laureates/dennis-ritchie#>

# Linguagem C

**Influenciada por:** ALGOL 68, Assembly, B, BCPL, CPL, Fortran, PL/I

# Linguagem C

**Influenciou:** AWK, BitC, C++, C#, C Shell, D, Euphoria, Go, Java, JavaScript, Limbo, Logic Basic, Objective-C, Perl, PHP, Python, ...

# Por que estudar a linguagem C?

- O *kernel* dos principais sistemas operacionais (Linux, MacOS, Windows) são escritos em C, e por isso
- os compiladores de C foram muito otimizados!
- É uma linguagem de nível intermediário: saber C nos ajuda a entender como o computador funciona.
- É uma linguagem que ainda é popular!

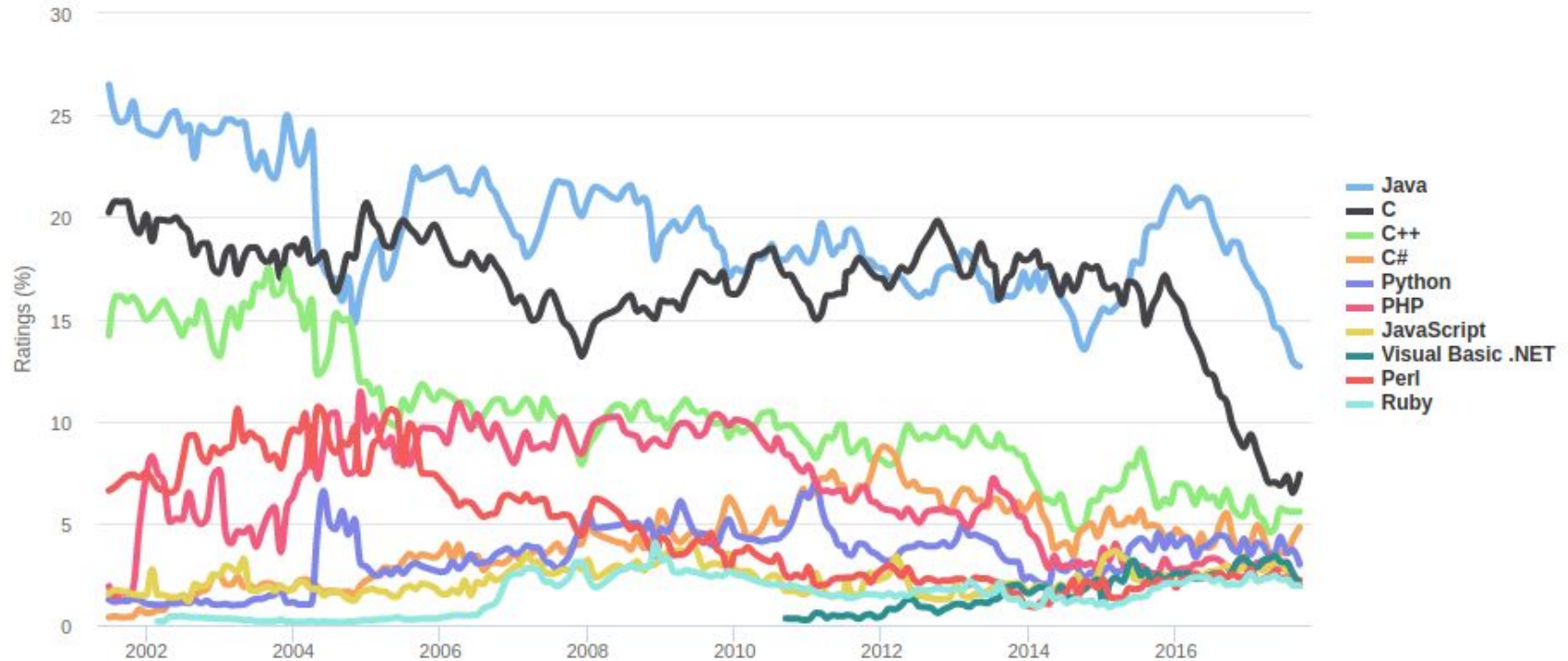
# TIOBE Programming Community Index

Sep 2017	Sep 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.687%	-5.55%
2	2		C	7.382%	-3.57%
3	3		C++	5.565%	-1.09%
4	4		C#	4.779%	-0.71%
5	5		Python	2.983%	-1.32%
6	7	▲	PHP	2.210%	-0.64%
7	6	▼	JavaScript	2.017%	-0.91%



# TIOBE Programming Community Index

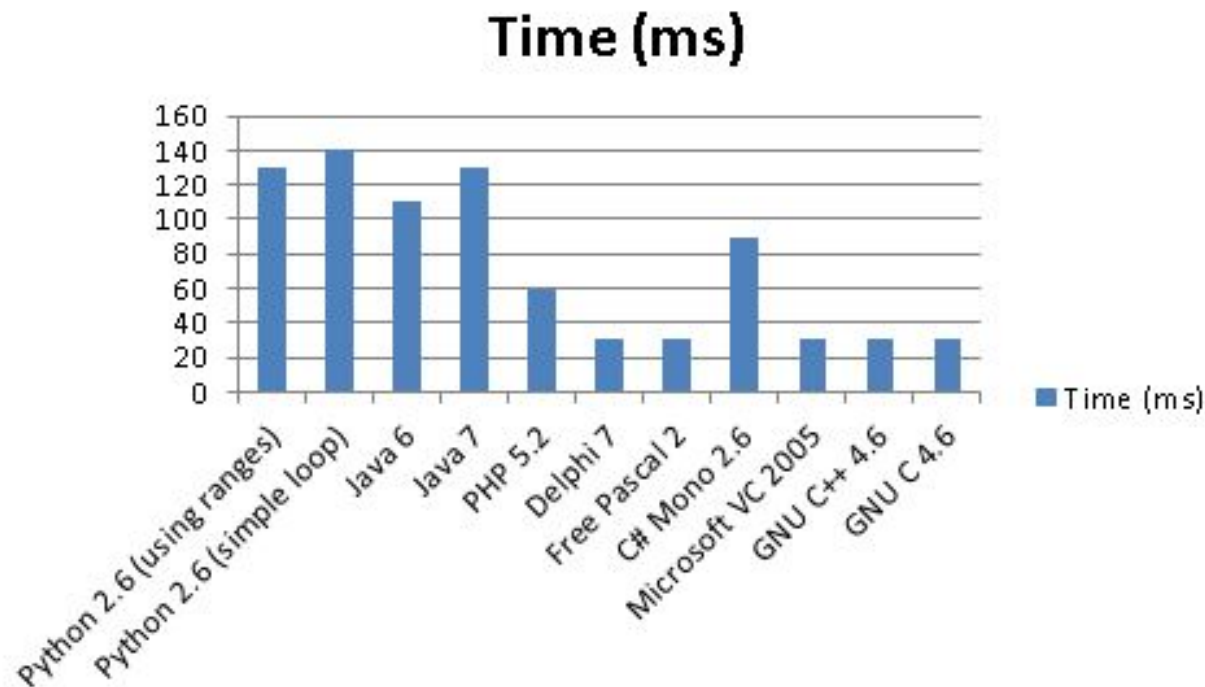
Source: [www.tiobe.com](http://www.tiobe.com)



# C versus Java

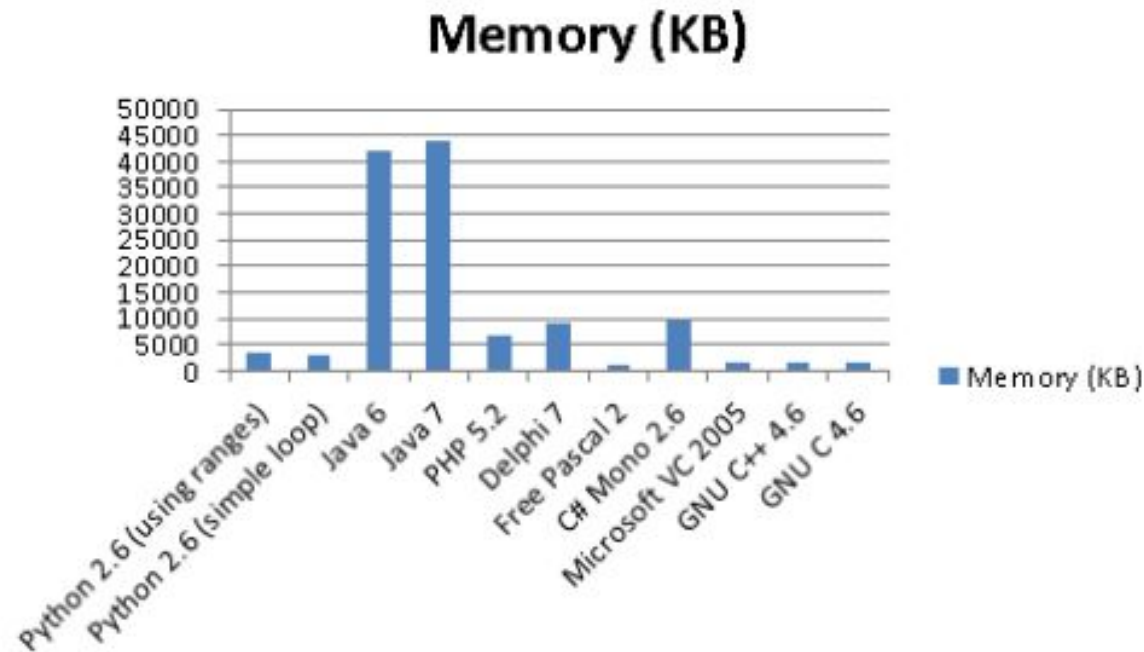
- Linguagem de nível intermediário:
  - controle mais direto do hardware, porém
  - também suporta estruturas complexas.
- Gerenciamento de memória explícito.
- Detecção de erro explícita (sem try/catch).
- Maior performance do programa final.
- Maior dificuldade de manutenção.

# Comparação (tempo de execução)



Fonte: <https://helloacm.com/a-quick-performance-comparison-on-languages-at-codeforces/>

# Comparação (consumo de memória)



Fonte: <https://helloacm.com/a-quick-performance-comparison-on-languages-at-codeforces/>