

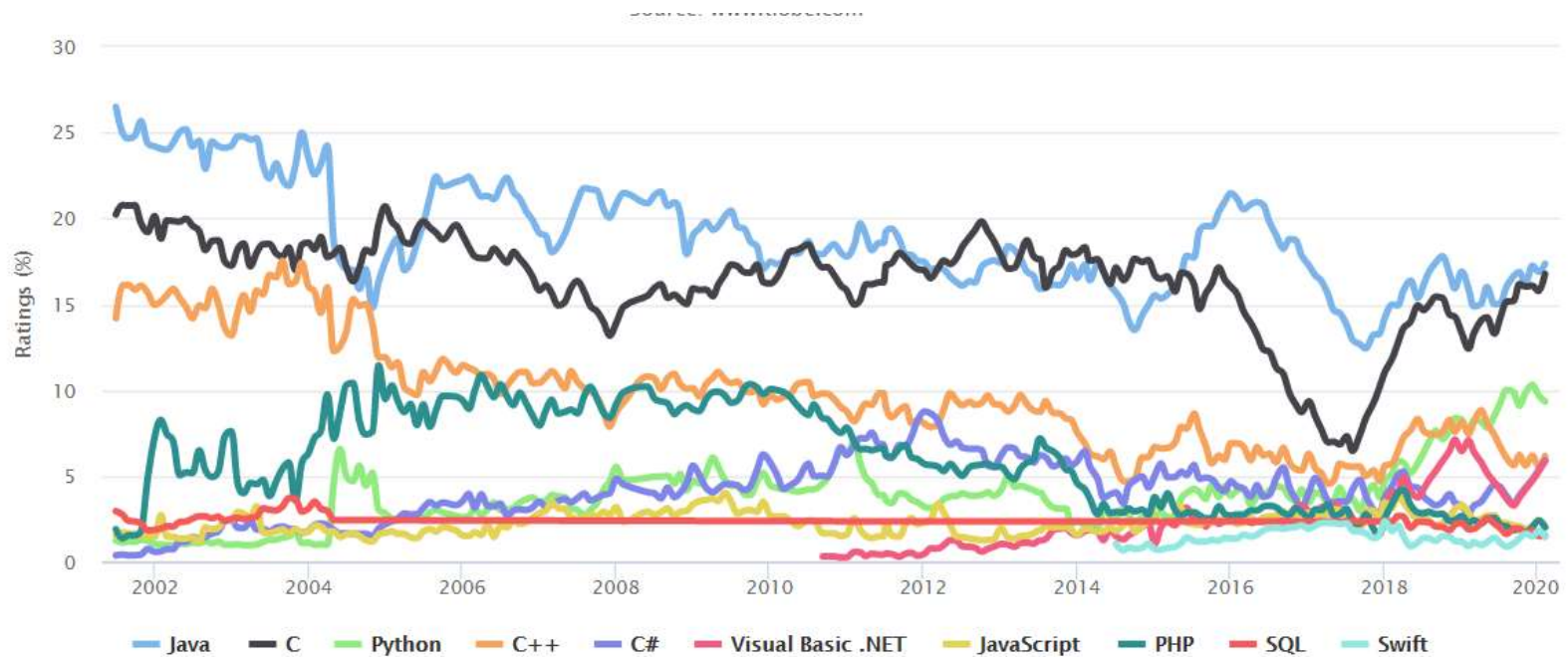
Introdução a Linguagem C/C++

Dr. Almir Rogério Camolesi
camolesi@fema.edu.br

Roteiro

- História da linguagem C
- Características da linguagem C
- Sistemas de tipos
- Variáveis e operadores
- Estruturas de controle
- Funções
- Entrada e Saída

Pesquisa TIOBE – Fevereiro 2021



<https://www.tiobe.com/tiobe-index/>

Pesquisa TIOBE – Fevereiro 2020

Feb 2020	Feb 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.358%	+1.48%
2	2		C	16.766%	+4.34%
3	3		Python	9.345%	+1.77%
4	4		C++	6.164%	-1.28%
5	7	▲	C#	5.927%	+3.08%
6	5	▼	Visual Basic .NET	5.862%	-1.23%
7	6	▼	JavaScript	2.060%	-0.79%
8	8		PHP	2.018%	-0.25%
9	9		SQL	1.526%	-0.37%
10	20	▲	Swift	1.460%	+0.54%
11	18	▲	Go	1.131%	+0.17%
12	11	▼	Assembly language	1.111%	-0.27%
13	15	▲	R	1.005%	-0.04%
14	23	▲	D	0.917%	+0.28%
15	16	▲	Ruby	0.844%	-0.19%
16	12	▼	MATLAB	0.794%	-0.40%
17	21	▲	PL/SQL	0.764%	-0.05%
18	14	▼	Delphi/Object Pascal	0.748%	-0.32%
19	13	▼	Perl	0.697%	-0.40%
20	10	▼	Objective-C	0.688%	-0.76%

<https://www.tiobe.com/tiobe-index/>

Origem

Linguagens de Programação

- <https://camo.githubusercontent.com/12085df272db0468fcdc183de32d10d47db06ca0ace8029276cb3b05eb29ab43/68747470733a2f2f7261776769746875622e636f6d2f73746572656f626f6f737465722f70726f6772616d6d696e672d6c616e6775616765732d67656e65616c6f676963616c2d747265652f67682d70616765732f696d672f6469616772616d2e737667>
- https://edisciplinas.usp.br/pluginfile.php/3439710/mod_resource/content/1/Aula08-HistoriaLinguagensProgramacao.pdf

História da Linguagem C

- Surgiu no início dos anos 70
- Criada inicialmente para o UNIX
- Criadores:
 - Dennis Ritchie (direita)
 - Kenneth Thompson (esquerda)
- Baseada na Linguagem B
- Versão inicial bastante simples



História da Linguagem C

- Ampla popularização nos anos 80
- Muitas arquiteturas e compiladores
- Problemas com a incompatibilidade
- Padronização de 82 a 89 (C ANSI)
- Até hoje existem problemas entre os diversos compiladores e sistemas operacionais

Características

- Paradigma Procedural
- Flexível
- Alta performance
- Poucas restrições
- Ótima interação com:
 - Sistemas Operacionais
 - Dispositivos de Hardware
 - Outras Linguagens

Palavras Reservadas

*auto, break, case, char, const,
continue,
default, do, double, else, enum,
extern, float,
for, goto, if, int, long, register,
return,
short, signed, sizeof, static, struct,
switch,
typedef, union, unsigned, void,
volatile, while*

Obs.: C/C++ é case sensitive

Tipos

- São as formas que utilizamos para representar dados
- C/C++ possui 5 tipos básicos:
 - `char`, `int`, `float`, `double` e `void`
- E 4 modificadores básicos:
 - `signed`, `unsigned`, `long` e `short`
 - Os 4 podem ser aplicados ao `int`
 - `long` pode ser aplicado ao `double`
 - `signed` e `unsigned` aplicados ao `char`

Tipos

- O tamanho do inteiro depende da arquitetura do sistema:
 - Sistemas de 32 bits \Rightarrow inteiro de 32 bits
 - Sistemas de 64 bits \Rightarrow inteiro de 64 bits
- Restrições:
 - `short int` e `int` devem ter pelo menos 16 bits
 - `long int` com no mínimo 32 bits
 - `short int` \leq `int` \leq `long int`

Variáveis

- Declaração:
 - *tipo nome = inicialização;*
- Escopo da variáveis:
 - globais: podem ser usadas em qualquer lugar do programa
 - locais: podem ser usadas apenas na função onde foi declarada

Variáveis

- Restrições

- O nome das variáveis deve começar com uma letra ou um sublinhado “_”
- Os demais caracteres podem ser letras, números ou sublinhado
- O nome da variável não pode ser igual a uma palavra reservada e aos nomes das funções
- Tamanho máximo para o nome de uma variável:
 - 32 caracteres

Constantes

- São valores que são mantidos fixos pelo compilador
- Também podem ser:
 - Octais - 0NUMERO_OCTAL
 - Hexadecimais - 0xNUMERO_HEXADECIMAL
- Exemplos:
 - `'\n'` (caractere), `"C++"` (string), **10** (inteiro), **15.0** (float), **0xEF** (239 em decimal), **03212** (1674 em decimal)

Operadores Relacionais

Operador	Relação
>	Maior que
>=	Maior que ou igual a
<	Menor que
<=	Menor que ou igual a
==	Igual a
!=	Diferente de

Operadores Lógicos

Operador	Função
&&	AND
 	OR
!	NOT

Tabela de Precedências

,	=	?		&		^	&	=	<	<	+	*	!	(
	+			&				=	<	<	-	/	~)
	=							!	<	<		%	+	[
	-							=	=	>			+]
	=								>	>			-	->
	*								=				-	
	=												.	
	/												-	
	=												*	
													&	



-

Precedência

+



Casts

- Sintaxe:
 - *(tipo) expressão*

- Exemplo:

```
long a = 10, b = 4;
```

```
// f = 2.5
```

```
double f = (double) a/b;
```

Controladores de Fluxo

- C possui 7 controladores de fluxo básicos:
 - `if`
 - `?:`
 - `switch`
 - `for`
 - `while`
 - `do-while`
 - `goto`

if

```
if (condição) { declaração }
```

```
if (condição) { declaração1 }  
else { declaração2 }
```

```
if (condição1) { declaração1 }  
else if (condição2) { declaração2 }  
else { declaração3 }
```

Exemplo

```
int a = 2;  
if (2) {  
    a = 4;  
} else if (0) {  
    a = 0;  
}  
if (a) {  
    a = -a;  
}
```

?: - Operador Ternário

- condição ? expressão1 : expressão2;
- Equivalente a:

```
if (condição) { expressão1 }  
else { expressão2 }
```
- O operador ? é restrito, mas pode simplificar expressões grandes
- Uma expressão é diferente de uma declaração

Exemplo

```
int a = 9, b = -4, c;
```

```
b = (a > b) ? a : b;
```

```
c = (a > b) ? b : a;
```


Funções

- Sintaxe:

```
tipoDeRetorno  
  nomeDaFunção(declaraçãoDosParâmetros)  
{  
  corpoDaFunção;  
}  
declaraçãoDosParâmetros = tipo1 nome1,...,  
                           tipoN nomeN
```

- Funções que não possuem retorno são **void**
- O retorno de uma função é feito através do comando **return**

Exemplos

```
int dobro(int a) {  
    return 2 * a;  
}
```

```
int main(int a) {  
    int c;  
  
    c = 8;  
    c = dobro(c);  
    return 0;  
}
```

Entrada e Saída de Dados

- A função `scanf` é utilizada para a leitura dados do dispositivo de entrada padrão
- A função `printf` é utilizada para a escrita de dados do dispositivo de saída padrão

Exemplos

```
int a;  
scanf("%d", &a); // lê a  
printf("%d", a); // escreve a
```

```
char ch;  
scanf("%c", &ch); // lê c  
printf("%c", ch); // escreve c
```

```
float num;  
scanf("%f", &num); // lê f  
printf("%f", num); // escreve f
```

Obs.: Cuidado para não esquecer do & no uso da função scanf

Tabela de Formato para E/S

Tipo	Formato para escrita e leitura
[signed unsigned] char	%c
[signed] int	%i ou %d
unsigned int	%u
[signed] short int	%hi
unsigned short int	%hu
[signed] long int	%li
unsigned long int	%lu
float	%f
double	%lf
long double	%Lf

Exercícios

- 4) Escreva um programa que lê um inteiro do teclado responde 'sim' se ele é um múltiplo de 3 e 'não' caso contrário

Tome como base a seguinte estrutura de programa:

```
#include <stdio.h>

int main() {
    return 0;
}
```

- 5) Implemente um programa que lê 10 inteiros do teclado ou para quando dois inteiros consecutivos sejam digitados. Após a leitura o programa deverá imprimir o maior, o menor, a média e a soma dos números digitados.
- 6) Modifique a sua resposta para o exercício anterior trocando os loops (**for**, **while**, **do-while**) para **goto**

Referências

- Matos, P. A. & Carvalho, G. H. P. - A Linguagem de Programação C
- The C Programming Language
 - <http://www.engin.umd.umich.edu/CIS/course.dcs/cis400/c/c.html>
- Curso de C da UFMG
 - <http://www.mtm.ufsc.br/~azeredo/cursoC/aulas-main.html>
- Lammert Bies, ASCII character map
 - <http://www.lammertbies.nl/comm/info/ascii-characters.html>
- Slides de Gustavo Henrique Porto do curso de C/C++