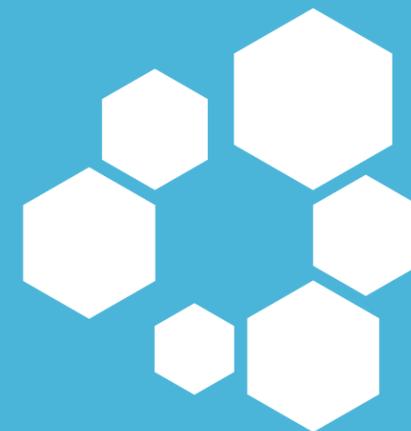


Introdução à Computação Variáveis e Entrada/Saída Continuação

Prof. Dr. Marcos Paulino Roriz Junior (marcosroriz@ufg.br)



UFG

UNIVERSIDADE
FEDERAL DE GOIÁS



ENGENHARIA DE
TRANSPORTES

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

Estrutura do programa C

```
#include <stdio.h>
```

Diretivas

```
int main() {
```

```
// bloco de comandos
```

Instruções

Corpo principal
do programa

```
return 0;
```

```
}
```



Declarações de variáveis

TIPO nome_da_variável;

Exemplo:

- `int x;`
- `float dado;`



Tipos de dados

Tipo	Faixa de valores	Tamanho (aproximado)
char	-128 a 127	8 bits
unsigned char	0 a 255	8 bits
int	-32.768 a 32.767	16 bits
unsigned int	0 a 65.535	16 bits
short int	-32.768 a 32.767	16 bits
long	-2.147.483.648 a 2.147.483.647	32 bits
unsigned long	0 a 4.294.967.295	32 bits
float	3.4×10^{-38} a 3.4×10^{38}	32 bits
double	1.7×10^{-308} a 1.7×10^{308}	64 bits
long double	3.4×10^{-4932} a 1.1×10^{4932}	80 bits



Tipos de Variáveis

```
scanf ("formato", &lista de variáveis);
```

```
printf ("formato", lista de variáveis);
```

%c	char
%d ou %i	int
%u	unsigned int
%f	float
%lf	double
%e	formato científico
%g	número real (float ou double)
%s	uma string (sequência de char)



Atribuição

Formas:

- Algoritmo: ←
- Programa: =

Exemplo:

- $x = 4;$
- $x = x + 2;$
- $y = 2.5;$
- $\text{sexo} = 'F';$



Palavras reservadas

Palavras reservadas são nomes utilizados pelo compilador para representar comandos de controle do programa, operadores e diretivas. As palavras reservadas da linguagem C/C++ são:

asm	auto	break	case	cdecl	char
class	const	continue	_cs	default	delete
do	double	_ds	else	enum	_es
export	extern	far	_fastcall	float	friend
goto	huge	for	if	inline	int
interrupt	_loadds	long	near	new	operator
pascal	private	protected	public	register	return
_saverregs	_seg	short	signed	sizeof	_ss
static	struct	switch	template	this	typedef
union	unsigned	virtual	void	volatile	while



Estrutura do programa C

```
#include <stdio.h>
```

Diretivas

```
int main() {
```

```
// bloco de comandos
```

Instruções

```
return 0;
```

Corpo principal
do programa

```
}
```



Definindo constantes

- `#define nome valor`

Exemplo

- `#define PI 3.14`
- `#define r 1444.32`



Comentários

- Comentários permite documentar o seu programa
- São ignorados pelo compilador

Comentário de linha (ignora tudo depois de //)

- `int x; // declarando x`

Comentário de bloco (pode ser mais de uma linha)

- Inicia com `/*`
- Termina com `*/`
 - `/* meu programa`
 - `feito no dia __/__/__`
 - `criei usando o codeblocks`
 - `*/`



Exemplo

```
/*
```

```
Essa variável é útil para calcular a área
```

```
*/
```

```
int a;
```

```
// outro comentário
```

```
int b; // b também é relacionado a variável a
```



Operadores

Operador	Exemplo	Comentário
=	$x = y$	O conteúdo da variável Y é atribuído à variável X (A uma variável pode ser atribuído o conteúdo de outra, um valor constante ou, ainda, o resultado de uma função).
+	$x + y$	Soma o conteúdo de X e de Y.
-	$x - y$	Subtrai o conteúdo de Y do conteúdo de X.
*	$x * y$	Multiplica o conteúdo de X pelo conteúdo de Y.
/	x / y	Obtém o quociente da divisão de X por Y. Se os operandos são inteiros, o resultado da operação será o quociente inteiro da divisão. Se os operandos são reais, o resultado da operação será a divisão. Por exemplo: int z = 5/2; → a variável z receberá o valor 2. float z = 5.0/2.0; → a variável z receberá o valor 2.5.
%	$x \% y$	Obtém o resto da divisão de X por Y.



Operadores

Operador	Exemplo	Comentário
<code>+=</code>	<code>x += y</code>	Equivale a $X = X + Y$.
<code>--</code>	<code>x -= y</code>	Equivale a $X = X - Y$.
<code>*=</code>	<code>x *= y</code>	Equivale a $X = X * Y$.
<code>/=</code>	<code>x /= y</code>	Equivale a $X = X / Y$.
<code>%=</code>	<code>x %= y</code>	Equivale a $X = X \% Y$.
<code>++</code>	<code>x++</code>	Equivale a $X = X + 1$.
<code>++</code>	<code>y = ++x</code>	Equivale a $X = X + 1$ e depois $Y = X$.
<code>++</code>	<code>y = x++</code>	Equivale a $Y = X$ e depois $X = X + 1$.
<code>---</code>	<code>x---</code>	Equivale a $X = X - 1$.
<code>--</code>	<code>y = --x</code>	Equivale a $X = X - 1$ e depois $Y = X$.
<code>--</code>	<code>y = x--</code>	Equivale a $Y = X$ e depois $X = X - 1$.



Operadores

Operador	Exemplo	Comentário
<code>==</code>	<code>x == y</code>	O conteúdo de X é igual ao conteúdo de Y.
<code>!=</code>	<code>x != y</code>	O conteúdo de X é diferente do conteúdo de Y.
<code><=</code>	<code>x <= y</code>	O conteúdo de X é menor ou igual ao conteúdo de Y.
<code>>=</code>	<code>x >= y</code>	O conteúdo de X é maior ou igual ao conteúdo de Y.
<code><</code>	<code>x < y</code>	O conteúdo de X é menor que o conteúdo de Y.
<code>></code>	<code>x > y</code>	O conteúdo de X é maior que o conteúdo de Y.



Algoritmos

1. Linguagem natural → *português*.
2. Diagramas de fluxo → gráfica.
3. **Pseudocódigo: intermediário**
4. Linguagem de programação → código.



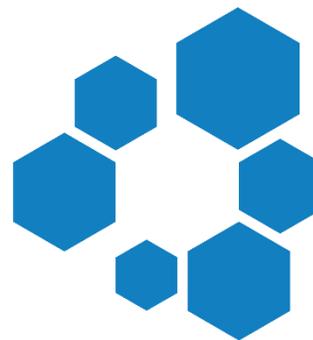
Obrigado!

Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE
TRANSPORTES**

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS



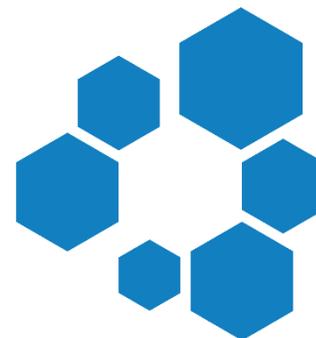
Obrigado!

Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE
TRANSPORTES**

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

