

Variáveis

João Pedro Oliveira Batisteli

Fevereiro, 2025

Sempre que queremos resolver um problema no computador precisamos, antes de tudo, descrevê-lo de forma clara e precisa. Ou seja, precisamos escrever o seu algoritmo.

O que é um Algoritmo?

O que é um Algoritmo?

Ingredientes

Gelatina

- 50 gramas de gelatina em pó do sabor da sua preferência
- 500 mililitros de água quente
- 500 mililitros de água gelada

Modo de Preparo

Gelatina

- 1 Em uma tigela, coloque 50 gramas de gelatina do sabor de sua preferência.
- 2 Adicione 500 mililitros de água quente e dissolva bem a gelatina com um garfo.
- 3 Despeje a água gelada sobre a água quente com gelatina dissolvida e volte a misturar bem.
- 4 Cubra com um plástico-filme e leve à geladeira até endurecer.
- 5 Sua receita de gelatina simples está pronta para ser servida. Bom apetite!

O que é um Algoritmo?

- "*Um algoritmo corresponde a uma descrição de um padrão de comportamento expresso em termos de um conjunto finito de ações*" - Dijkstra (1971).

O que é um Algoritmo?

- "*Um algoritmo corresponde a uma descrição de um padrão de comportamento expresso em termos de um conjunto finito de ações*" - Dijkstra (1971).
- É uma receita para resolver um problema.

Dado -

Dado - toda e qualquer tipo de informação manipulada por um programa

Dado - toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado -

Dado - toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado - representação computacional da informação. Define como armazená-la e como interpretá-la.

Dado - toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado - representação computacional da informação. Define como armazená-la e como interpretá-la.

Em computação, uma estrutura de dados é uma forma de armazenar e organizar os dados de modo que eles possam ser usados de forma eficiente.

Dado - toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado - representação computacional da informação. Define como armazená-la e como interpretá-la.

Em computação, uma estrutura de dados é uma forma de armazenar e organizar os dados de modo que eles possam ser usados de forma eficiente.

Tipos Básicos -

Dado - toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado - representação computacional da informação. Define como armazená-la e como interpretá-la.

Em computação, uma estrutura de dados é uma forma de armazenar e organizar os dados de modo que eles possam ser usados de forma eficiente.

Tipos Básicos - mais utilizados. Podem ser numéricos, lógicos, literais ou caracteres.

Tipos Básicos de Datos

- Numérico: 2, 10, 381, 584, -15, -78

Tipos Básicos de Datos

- Numérico: 2, 10, 381, 584, -15, -78
- Numérico: 3.141592653, 2.71828

Tipos Básicos de Datos

- Numérico: 2, 10, 381, 584, -15, -78
- Numérico: 3.141592653, 2.71828
- Lógico: True, False

Tipos Básicos de Datos

- Numérico: 2, 10, 381, 584, -15, -78
- Numérico: 3.141592653, 2.71828
- Lógico: True, False
- Caracteres: 'z', 'f', '1'

Tipos Básicos de Dados

- Numérico: 2, 10, 381, 584, -15, -78
- Numérico: 3.141592653, 2.71828
- Lógico: True, False
- Caracteres: 'z', 'f', '1'
- Literais: "Exemplo", "Eu amo AEDs I"

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Variável -

Variável - local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

Variável - local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

Toda variável tem um nome, um tipo e um valor.

Variável - local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

Toda variável tem um nome, um tipo e um valor.

O nome será escolhido pelo desenvolvedor/programador e deve atender a algumas regras. Sugere-se que comece com minúscula ou '_' e todas as palavras que compõe o nome com maiúsculas (CamelCase).

Variável - local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

Toda variável tem um nome, um tipo e um valor.

O nome será escolhido pelo desenvolvedor/programador e deve atender a algumas regras. Sugere-se que comece com minúscula ou '_' e todas as palavras que compõe o nome com maiúsculas (CamelCase).

Ex: YouTube, PowerPoint, MinhaVariavel, NumeroMuitoGrande

1. Diferente de palavras reservadas;

Nomenclatura de Variáveis - Regras

1. Diferente de palavras reservadas;
2. Não pode ter símbolos, exceto o *underline* '_';

Nomenclatura de Variáveis - Regras

1. Diferente de palavras reservadas;
2. Não pode ter símbolos, exceto o *underline* '_';
3. Deve começar por uma letra ou pelo *underline* '_';

Nomenclatura de Variáveis - Regras

1. Diferente de palavras reservadas;
2. Não pode ter símbolos, exceto o *underline* '_';
3. Deve começar por uma letra ou pelo *underline* '_';
4. Não podem existir nomes repetidos em um mesmo escopo;

Nomenclatura de Variáveis - Regras

1. Diferente de palavras reservadas;
2. Não pode ter símbolos, exceto o *underline* '_';
3. Deve começar por uma letra ou pelo *underline* '_';
4. Não podem existir nomes repetidos em um mesmo escopo;
5. Minúsculas e maiúsculas são consideradas diferentes (*case sensitive*);

Nomenclatura de Variáveis - Regras

1. Diferente de palavras reservadas;
2. Não pode ter símbolos, exceto o *underline* '_';
3. Deve começar por uma letra ou pelo *underline* '_';
4. Não podem existir nomes repetidos em um mesmo escopo;
5. Minúsculas e maiúsculas são consideradas diferentes (*case sensitive*);
6. Não podem ser acentuados ou utilizar o cedilha 'ç'.

Declaração de Variável

Declaração de Variável

Declaração de Variável - o comando que efetivamente “cria” a variável, reservando-a em memória e associando nome e tipo a ela. O local onde a variável foi armazenada é identificado internamente no computador pelo endereço de memória.

Declaração de Variável

Declaração de Variável - o comando que efetivamente “cria” a variável, reservando-a em memória e associando nome e tipo a ela. O local onde a variável foi armazenada é identificado internamente no computador pelo endereço de memória.

Inicialmente não vamos precisar nos preocupar com a alocação de memória, ela será feita de maneira “automática”.

Declaração de Variável

Declaração de Variável - o comando que efetivamente “cria” a variável, reservando-a em memória e associando nome e tipo a ela. O local onde a variável foi armazenada é identificado internamente no computador pelo endereço de memória.

Inicialmente não vamos precisar nos preocupar com a alocação de memória, ela será feita de maneira “automática”.

Atribuição

Declaração de Variável

Declaração de Variável - o comando que efetivamente “cria” a variável, reservando-a em memória e associando nome e tipo a ela. O local onde a variável foi armazenada é identificado internamente no computador pelo endereço de memória.

Inicialmente não vamos precisar nos preocupar com a alocação de memória, ela será feita de maneira “automática”.

Atribuição - armazena dado na variável substituindo conteúdo anterior pelo indicado.

- `int` - inteiro, sem casa decimal (ex: -2, 10, 65902);

- `int` - inteiro, sem casa decimal (ex: -2, 10, 65902);
- `Float/Double` - ponto flutuante, números com casa decimal, (ex: -3.167, 1.786, 29.33);

- `int` - inteiro, sem casa decimal (ex: -2, 10, 65902);
- `Float/Double` - ponto flutuante, números com casa decimal, (ex: -3.167, 1.786, 29.33);
- `bool` - lógicos (ex: `True` e `False`);

- `int` - inteiro, sem casa decimal (ex: -2, 10, 65902);
- `Float/Double` - ponto flutuante, números com casa decimal, (ex: -3.167, 1.786, 29.33);
- `bool` - lógicos (ex: True e False);
- `char` - caracter (ex: 'a', 'b', 'c', '5').

Tipos Básicos em C/C++

Tipo	Bits	Bytes	Escala
char	8	1	128 a 127
int	32	4	-2.147.483.648 a 2.147.483.647 (ambientes de 32 bits)
short	16	2	-32.765 a 32.767
long	32	4	-2.147.483.648 a 2.147.483.647
unsigned char	8	1	0 a 255
unsigned	32	4	0 a 4.294.967.295 (ambientes de 32 bits)
unsigned long	32	4	0 a 4.294.967.295
unsigned short	16	2	0 a 65.535
float	32	4	$3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
double	64	8	$1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
long double	80	10	$3,4 \times 10^{-4932}$ a $3,4 \times 10^{4932}$
void	0	0	nenhum valor

Declaração de Variáveis

Formato:

```
<tipo> <nome_da_variavel> = dadoInicial;  
                                (opcional)
```

Exemplos:

- 1) Declare a variável x e atribua a ela o valor 5.

- 1) Declare a variável x e atribua a ela o valor 5.
- 2) Declare a variável b e atribua a ela o valor 7.14.

- 1) Declare a variável x e atribua a ela o valor 5.
- 2) Declare a variável b e atribua a ela o valor 7.14.
- 3) Declare a variável c e atribua a ela o valor 'c'.

- 1) Declare a variável x e atribua a ela o valor 5.
- 2) Declare a variável b e atribua a ela o valor 7.14.
- 3) Declare a variável c e atribua a ela o valor 'c'.

Conversão de tipos

Conversão de tipos - é a mudança da forma de representação de um tipo para outro

Conversão de tipos - é a mudança da forma de representação de um tipo para outro

Ex:

$5.0 \longrightarrow 5$ ou $5 \longrightarrow 5.0$

Conversão de tipos - é a mudança da forma de representação de um tipo para outro

Ex:

$$5.0 \longrightarrow 5 \quad \text{ou} \quad 5 \longrightarrow 5.0$$

Conversão implícita – ocorre automaticamente para alguns tipos. Normalmente, refere-se a um subconjunto atribuído a uma variável com tipo que o engloba (geralmente ocorre automaticamente durante a execução do programa).

Conversão de Variáveis

Conversão de tipos - é a mudança da forma de representação de um tipo para outro

Ex:

$$5.0 \longrightarrow 5 \quad \text{ou} \quad 5 \longrightarrow 5.0$$

Conversão implícita – ocorre automaticamente para alguns tipos. Normalmente, refere-se a um subconjunto atribuído a uma variável com tipo que o engloba (geralmente ocorre automaticamente durante a execução do programa).

Conversão explícita – necessária quando existe o risco de perder informação (precisão) na conversão.

Constante - informação armazenada em um primeiro momento não pode ser alterada posteriormente.

Constante - informação armazenada em um primeiro momento não pode ser alterada posteriormente.

Constantes são usadas para armazenar valores que **NÃO** podem ser modificadas durante a execução de um programa.

Constante - informação armazenada em um primeiro momento não pode ser alterada posteriormente.

Constantes são usadas para armazenar valores que **NÃO** podem ser modificadas durante a execução de um programa.

O nome segue as regras dos identificadores das variáveis. Sugere-se o uso de maiúsculas apenas.

Constante - informação armazenada em um primeiro momento não pode ser alterada posteriormente.

Constantes são usadas para armazenar valores que **NÃO** podem ser modificadas durante a execução de um programa.

O nome segue as regras dos identificadores das variáveis. Sugere-se o uso de maiúsculas apenas.

Uma constante precisa ser declarada, e para tanto usamos a diretiva de pré-processador `#define`.

Exemplo de Constantes

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265
. . .
```

Dúvidas?