

Tipos de variáveis

Inteira - admite apenas números inteiros, positivos ou negativos.

Real – Números decimais, positivos ou negativos

Caracter - Aceita apenas 1 caracter (letra, dígito ou símbolo), veja a tabela ASCII.

Booleano - aceita apenas Verdadeiro(!=0) ou Falso (0)

Cadeia de caracteres - Aceita de 1 a vários caracteres, o comprimento da cadeia é o total de caracteres + 1 que indica o fim da cadeia.

Vetor – Um ou mais posições do mesmo tipo da variável

Constante

Contante - É uma variável especial que assume apenas um valor, quando é criada, e não pode ser alterada.

Representação de variáveis:

Podemos pensar que cada variável é uma caixa, e que a memória do computador é uma sequência de caixas, como podemos ver abaixo um exemplo de um trecho de memória do computador:

[illegible]

```
Inteiro      I
cadeia de caracteres C[5]
vetor Inteiro P3D[3]
Booleano b
```

[illegible]

Escopo de variáveis

Global - É reconhecida em todo o programa/contexto. A vantagem é não precisa ser passada como parâmetro. A desvantagem é que pode ter seu valor alterado em qualquer parte do programa, sem que seja correto.

Local - É reconhecida apenas dentro do bloco onde foi declarada. A vantagem de ser local, é que apenas naquele bloco ela pode ter seu conteúdo alterado. A desvantagem é que para passar o seu valor para outro lugar do programa é necessário que ela seja alocada como resultado da execução do bloco em questão e devolvida para o programa principal.

Declaração de variável

Para os algoritmos e fluxograma deveremos colocar a palavra ***global*** a frente do nome da variável. Toda a variável que for declarada deste modo, será uma variável global. Caso não seja utilizado, a variável tem o escopo local.

Para que a variável seja reconhecida como constante, deve ser precedida na sua declaração com a palavra ***constante***.

Exemplos de declaração:

Inteiro constante fator \leftarrow 10
Inteiro p1
Real carteira
Cadeia de caracteres nome
Vetor Ponto3D[3]

Teste de Mesa:

É uma maneira de executar o nosso algoritmo, fluxograma ou até mesmo um programa, para identificar se está funcionando corretamente, ou identificar o que ele faz.

Exemplo:

```
1 DECLARAR VARIÁVEIS
2 INTEIRO V1, V2, V3
3 V1  $\leftarrow$  10
4 V3  $\leftarrow$  3
5 V2  $\leftarrow$  3
6 V1  $\leftarrow$  V2 * V2 + V3
7 MOSTRAR V1
```

8 FIM

#passo	Comando	Variáveis		
		V1	V2	v3
3	V1 ← 10	10		
4	V3 ← 3	10		3
5	V2 ← 3	10	3	3
6	V1 ← V2 * V2 + V3	12	3	3

cadeia de caracteres nome ← “Fulan**o** de tal”
nome ← “”

c ← nome[5] => c = “**o**”

vetor inteiro ponto3D[**3**]

ponto3D		
0	1	2
Inteiro	Inteiro	inteiro

inteiro contador

ponto3D			contador
0	1	2	0
Inteiro	Inteiro	inteiro	inteiro

caracter letra

ponto3D			contador	letra
0	1	2	0	0
Inteiro	Inteiro	inteiro	inteiro	caracter

vetor inteiro histograma[256]

ponto3D			contador	letra	histograma		
0	1	2	0	0	0	...	255
Inteiro	Inteiro	inteiro	inteiro	caracter	inteiro	...	inteiro

ponto3D ← { 10, 5, -1}

ponto3D			contador	letra	histograma		
0	1	2	0	0	0	...	255
10	5	-1	inteiro	caracter	inteiro	...	inteiro

vetor inteiro data[3]

ponto3D			contador	letra	histograma			data		
0	1	2	0	0	0	...	255	0	1	2
10	5	-1	inteiro	caracter	inteiro	...	inteiro	Inteiro	Inteiro	inteiro

data ← { 20, 10, 2020}

ponto3D			contador	letra	histograma			data		
0	1	2	0	0	0	...	255	0	1	2
10	5	-1	inteiro	caracter	inteiro	...	inteiro	20	10	2020

Algoritmo de Calculo de média

```
1 Declarar variáveis  media, c, quantidade  <= Globais
2 Declarar vetor entrada[100]
3 Pedir quantidade de valores para obter a média
4 se quantidade < 1 OU quantidade > 100
    4.1 Voltar ao passo 3
5 fim se
6 Faça c ← 1 até quantidade
    6.1 Pedir entrada[c]
7 FIM Faça
8 media ← calculemedia(entrada, quantidade)
9 mostrar media
10 fim
```

Calculemedia(vetor, inteiro)

```
1 Declarar variavel c,r  <= variáveis locais
2 se quantidade < 1 ou quantidade > 100
    2.1 r ← -1
3 caso contrário
    3.1 Faça c ← 1 até quantidade
        3.1.1 r ← r + entrada[c]
    3.1 FIM Faça
    3.2 r ← r / quantidade
4 Fim Se
5 Retorne r
```