

**Roteiro de
Estudos**



**Algoritmos e Técnicas de
Programação**

Este roteiro orientará a sua aprendizagem por meio da leitura de livros e artigos que cabem na sua rotina de estudos. Experimente esse recurso e aumente a sua habilidade de relacionar a teoria à prática profissional.

No seu caminho de aprendizagem, você encontrará os seguintes tópicos:

- ✓ Texto de apresentação de cada leitura indicada;
- ✓ Links para acesso às referências bibliográficas.

É importante ressaltar: o seu esforço individual é fundamental para a sua aprendizagem, mas você não estará sozinho nessa!

UNIDADE 2

Constantes, variáveis e operações

Constantes e variáveis com tipos de dados primitivos

Os programas de computadores, precisam acessar os dados que foram calculados ou inseridos pelo usuário, chamamos de variáveis os locais onde os dados são armazenados. Deitel e Deitel (2011, p. 43) nos trazem a seguinte definição: “Uma variável é uma posição na memória onde um valor pode ser armazenado para ser utilizado por um programa.”

Quando uma variável é criada em um programa, um espaço na memória é utilizado para armazenar o que poderá ser utilizado a qualquer momento enquanto o programa estiver sendo executado. Deitel e Deitel (2011, p. 43) nos trazem a seguinte definição: “Uma variável é uma posição na memória onde um valor pode ser armazenado para ser utilizado por um programa.” Soffner (2013, p. 33) incrementa dizendo que “variáveis são endereços de memória de trabalho que guardam, temporariamente, um valor utilizado pelo programa.”

Na linguagem C de programação, cada tipo de variável usa um especificador de formato para fazer a impressão do valor que está guardado naquele espaço da memória (PEREIRA, 2017). Veja a seguir alguns tipos de variáveis.

INT - Numérico inteiro: Corresponde ao conjunto dos números naturais, ou seja, valores inteiros que podem ser positivos, negativos ou zero.

FLOAT - Numérico de ponto flutuante: esse tipo é utilizado para armazenar valores com casas decimais. São exemplos de variáveis desse grupo as que armazenam temperatura, velocidade, dinheiro, etc.

BOOL - Usado para validações, apenas armazenam dois valores: verdadeiro ou falso. Na programação pode ser usado para verificar se o usuário acessou ou não alguma função, se uma opção de uma lista foi selecionada ou não, etc.

CHAR – Esse tipo de variável é utilizado para armazenar letras, como o nome do cliente, a placa do automóvel, etc.

Muitas vezes, o programa precisa do valor inalterado durante todo o processamento, esse tipo de armazenamento de dados, é denominado **constante**, por exemplo, um coeficiente de atrito, uma velocidade específica, etc. Quando for nomear as variáveis, atenção em algumas regras como, não poder ser usados acentos nem caracteres especiais, como a exclamação, arroba, asterisco, etc. A maioria das linguagens de programação são case sensitive, o que significa que letras maiúsculas e minúsculas são tratadas com distinção. Então, a variável “media” é diferente da variável “Media”.

Na linguagem C, quando for definida uma constante, deve-se usar a diretiva `#define`, Segundo Manzano (2015), um ponto interessante é que a diretiva não utiliza espaço em memória, ela apenas cria um rótulo associado a um valor. Existe uma outra maneira de declarar uma constante, basta inserir o comando `const` definir o nome. Por exemplo, `#define pi 3.14`, `const float gravidade = 9.80`.

Referências e Link do material na Biblioteca Virtual e artigo

DEITEL, P.; DEITEL, H. **C Como Programar**. 6. ed. São Paulo: Pearson, 2011.

MANZANO, J. A. N. G. **Algoritmos: técnicas de programação**. 2. ed. São Paulo: Érica, 2015.

RICARTE, I.L.M. **Estrutura de Dados**. <http://www.dca.fee.unicamp.br/~ricarte/>

SCHIEFFER, Vanessa Cadan. ARTERO, Marcio Aparecido. **Algoritmos e técnicas de programação** – Londrina: Editora e Distribuidora Educacional S.A., 2018.

Variáveis com tipos de dados compostos e ponteiros

As **variáveis compostas** são formadas a partir dos tipos primitivos e podem ser classificadas em homogêneas e heterogêneas. Além disso, podem ser unidimensionais ou multidimensionais, sendo a bidimensional mais comumente usada (MANZANO, 2015). Quando armazenam valores do mesmo tipo primitivo, são homogêneas, mas quando armazenam valores de diferentes tipos são heterogêneas. A essa composição de dados, chamamos de vetor ou array.

Cada vetor tem um índice que referência o conteúdo, assim é possível acessar qualquer informação. Um vetor não deixa de ser uma variável que tem a capacidade de armazenar vários dados ao mesmo tempo. A criação de um vetor é similar a uma variável primitiva, tendo que acrescentar apenas um número entre colchetes indicando qual será o tamanho desse vetor. Na programação a definição ficaria assim: `int idade[3]; float altura[3] = {1,1.5,1.7}`.

Uma característica dos vetores são os ponteiros, ou seja, as posições onde as informações são armazenadas. E consequentemente as informações são facilmente recuperadas ou processadas. Os ponteiros são fundamentais para a Linguagem C; são parte do poder que essa linguagem tem em relação as demais. É por meio dos ponteiros que podemos manipular variáveis e outros recursos pelo endereço de memória, o que gera amplas capacidades e possibilidades.

Referências e Link do material na Biblioteca Virtual e artigo

MANZANO, J. A. N. G. **Linguagem C: acompanhada de uma xícara de café**. São Paulo: Érica, 2015. 480 p.

SCHEFFER, Vanessa Cadan. ARTERO, Marcio Aparecido. **Algoritmos e técnicas de programação** – Londrina: Editora e Distribuidora Educacional S.A., 2018.

Operações e Expressões

Caro aluno, após armazenar os dados os processamentos podem ser realizados e gerar as suas respectivas soluções, dentre esse processamento, serão realizadas comparações e cálculos matemáticos.

Operador	Descrição	Exemplo	Resultado
+	Soma	$4 + 2$	6
-	Subtração	$4 - 2$	2
*	Multiplicação	$4 * 2$	8
/	Divisão	$4 / 2$	2
=	Atribuição	$x = 4$	$x = 4$
%	Módulo	$4 \% 2$	0

Segundo Soffner (2013), Os operadores aritméticos possuem a seguinte ordem de execução:

- 1° parênteses;
- 2° potenciação e radiciação;
- 3° multiplicação, divisão e módulo;
- 4° soma e subtração;

Fonte: adaptado de Manzano (2015, p. 43).

Em programação, para comparar valores os operadores relacionais serão utilizados, quadro abaixo apresenta os operadores usados nas linguagens de programação (DEITEL; DEITEL, 2011).

>	maior que
>=	maior que ou igual a
<	menor que
<=	menor que ou igual a
==	igual a
!=	diferente

Note que o operador de igualdade é “==”, e não simplesmente “=”. Esta é uma causa comum de erros para programadores que estão acostumados com outras linguagens onde = é um operador relacional. Valores booleanos podem ser combinados com os operadores booleanos:

Além dos operadores relacionais, outro importante recurso para o processamento é a utilização de operadores lógicos, que possuem como fundamento a lógica matemática clássica e a lógica Booleana (GERSTING, 2017).

	OU lógico
&&	E lógico
!	Negação

Estes operadores OU e E devem ser sempre usados entre duas expressões relacionais O operador && (E) resulta verdadeiro quando os dois valores de seus operandos são verdadeiros. O operador || (OU) resulta verdadeiro quando pelo menos um de seus dois operandos é verdadeiro.

((2 > 1) (3 < 7))	resultado VERDADEIRO
((3 < 2) && (2 == 2))	resultado FALSO
((5 != 0) (1 < 2))	resultado VERDADEIRO

O operador de Negação deve ser usado antes de uma expressão relacional, como por exemplo:

!(2 > 1)	resultado FALSO
!(1 < 0)	resultado VERDADEIRO

O símbolo de exclamação (!) na expressão “ $!(2 > 1)$ ” está negando o que está entre os parênteses deixando-a falsa, já na expressão “ $!(1 < 0)$ ” a negação(!) transforma o que está entre parênteses verdadeiro.

Siga em frente e ótimos estudos!

Referências e Link do material na Biblioteca Virtual e artigo

GERSTING, J. L. **Fundamentos matemáticos para a ciência da computação: matemática discreta e suas aplicações**. Rio de Janeiro: LTC, 2017.

SCHEFFER, Vanessa Cadán. ARTERO, Marcio Aparecido. **Algoritmos e técnicas de programação** – Londrina: Editora e Distribuidora Educacional S.A., 2018.