

3. Introdução

Segundo alguns especialistas, qualquer programa de computador pode ser escrito utilizando apenas três estruturas básicas: Seqüência, Condição e Repetição. Isto significa que, na construção de um algoritmo, deve ser criado um fluxo de ações a ser executado seqüencialmente, ou seja, de cima para baixo e da esquerda para a direita.

Antes, porém, é necessário fazer as seguintes perguntas: quais dados serão utilizados para solucionar o problema? De que tipo eles são (inteiro, real, literal ou lógico)? Qual o tamanho desses dados?

Portanto, nos itens seguintes são abordados descrição e exemplos a respeito de tipos de dados e demais assuntos relacionados. São também apresentados exemplos de utilização das estruturas básicas e uma lista de exercícios para fixação dos conceitos.

3.1. Tipos de dados, variáveis e operadores

Uma relação dos tipos de dados, dos operadores de atribuição, aritméticos, lógicos e relacionais que podem ser utilizados pelo computador bem como o conceito de variável são mostrados a seguir.

3.1.2. Tipos de dados

Uma informação manipulada pelo computador pode ser classificada como instruções ou dados. Sendo que, uma instrução comanda o funcionamento da máquina e determina como devem ser tratados os dados. E os dados, correspondem à porção das informações a serem processadas pelo computador. Os tipos de dados manipulados pelo computador são três:

- Dados numéricos
- Dados literais
- Dados lógicos

Recordando:

Conjunto dos números naturais (N)

$$\mathbf{N = \{ 1, 2, 3, 4, ... \}}$$

Conjunto dos números inteiros (Z)

$$\mathbf{Z = \{ ..., -3, -2, -1, 0, 1, 2, 3, ... \}}$$

Conjunto dos números fracionários (Q)

$$\mathbf{Q = \{ p/q \mid p, q \text{ pertencem a } Z \}}$$

Conjunto dos números reais (R) é formado pela união do conjunto Q.

3.1.2. Dados numéricos inteiros

Este tipo de dados são números pertencentes ao conjunto dos números inteiros (Z).

Podendo assumir valores negativos, nulo e positivos. Exemplos:

- 24 – número inteiro positivo
- 0 – número inteiro
- 12 – número inteiro negativo

2.1.1 Dados numéricos reais

Este tipo de dados são números pertencentes ao conjunto dos números fracionários (Q). Podendo assumir valores negativos, nulos e positivos. Exemplos:

- 24.01 – número real positivo com duas casas decimais.
- 144 – número real positivo com zero casas decimais.
- 13.3 – número real negativo com uma casa decimal.
- 0 – número real com zero casas decimais.

2.1.2 Dados literais

Este tipo de dados são caracteres tais como: letras, dígitos e/ou símbolos especiais. Podem ser chamados também como: dados alfanuméricos, cadeias de caracteres ou string. No algoritmo, estes dados são delimitados pelo apostrofo ('cadeia de caracteres'). Exemplos:

- 'qual ?' – literal de comprimento 6.

' '	- literal de comprimento 1.
'1-2+3='	- literal de comprimento 6.
'AbCdeF'	- literal de comprimento 7.
'0'	- literal de comprimento 1.

2.1.3 Dados lógicos

Estes tipos de dados são chamados de booleanos. Eles representam dados lógicos com

Verdadeiro e falso. Nos algoritmo seus valores são delimitados pelo ponto (.V.). Exemplos:

.V.	– valor lógico verdadeiro
.F.	– valor lógico falso

3.2. Variáveis

Variável representa um endereço da memória RAM. Nela somos capazes de armazenar informações de vários tipos: numéricos, literais e lógicos. Quando declaramos uma variável estamos atribuindo um nome simbólico a um endereço da memória RAM. Dentro de nosso programa utilizaremos este nome para manipular a informação contida no endereço da memória relacionada à variável.

Variável é um nome (rótulo) dado a uma parte da memória que pode ser alterado durante a execução do programa. Embora seja possível alterar o valor armazenado na memória (variável), este rótulo só armazena um valor de cada vez. Sendo assim, quando guardamos um valor em uma variável, o valor antigo que estiver lá será perdido. A variável é composta de dois elementos básicos:

- a) conteúdo – valor atual da variável.
- b) identificador – nome dado à variável.

Regras para criar um identificador:

- 1 – Não deve começar com um número;**
- 2 – Não pode ser igual a uma palavra reservada da linguagem;**

3 - Não pode conter espaço;

4 – Pode conter letras, números e underline (_).

Identificadores permitidos	A	Nome	N1	S2N	Nota_1
Identificadores não permitidos	2N	A –B	N[1]	A*2	média

Exemplos:

$A \leftarrow 5$ (O valor 5 é armazenado na variável A)

$B \leftarrow 6$ (O valor 6 é armazenado na variável B)

$C \leftarrow A+B$ (O valor que está em A é somado ao valor em B, e o resultado é armazenado na variável C.)

Atenção: Em Fluxograma não há declaração de variáveis.

Obs: Em Pseudocódigo todas as variáveis usadas no algoritmo devem ser declaradas antes de serem usadas. Isto faz necessário para permitir a reserva de espaço na memória.

Declaração de variáveis:

Var

<nome_da_variável> : <tipo_de_dados>;

Regras:

- A palavra Var deve aparecer uma única vez;
- Para um mesmo tipo de dado, podem ser listadas várias variáveis (seus nomes devem ser separados por vírgula);
- Variáveis de tipos diferentes devem ser declaradas em linhas diferentes.

Exemplos:

Var

nome,endereço: literal;

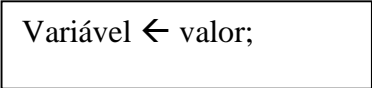
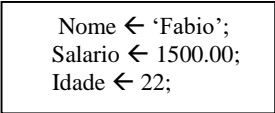
idade:inteiro;

salário:real;

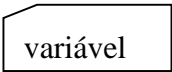
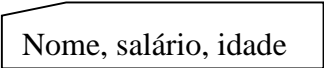
tem_filhos:lógico;

Atribuição dos dados (programador)

Pseudocódigo	Fluxograma
--------------	------------

Variável ← valor;	
Exemplo: Nome ← 'Fabio'; Salario ← 1500.00; Idade ← 22;	

Entrada de dados via teclado (usuário).

Pseudocódigo	Fluxograma
Leia(variável)	
Exemplo: Leia(nome); Leia(salario); Leia(idade);	

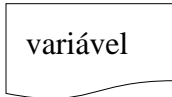
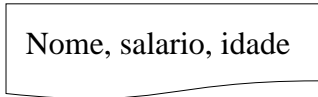
Memória



Como visualizar os dados armazenados na memória?

R: Usando o comando de saída de dados

Saída de dados via monitor

Pseudocódigo	Fluxograma
Escreva(variável)	
Exemplo: Escreva(nome); Escreva(salario); Escreva(idade);	

3.3 Tipos de operadores

- Operador de atribuição
- Operadores aritméticos

- Operadores relacionais
- Operadores lógicos

3.3.1 Operador de atribuição (\leftarrow)

Usamos o operador de atribuição para fornecer um valor a uma variável. Por exemplo:

$A \leftarrow 5$ (Estamos atribuindo o valor 5 à variável A)

$B \leftarrow A$ (Estamos atribuindo o valor da variável A, valor igual a 5, à variável B)

$A \leftarrow B + C$ (Estamos atribuindo o valor da soma dos valores das variáveis B + C – expressão- à variável C)

3.3.2. Operadores Aritméticos

São símbolos que indicam as operações aritméticas:

+	→	Adição
-	→	Subtração
*	→	Multiplicação
/	→	Divisão Real
DIV	→	Divisão Inteira
MOD	→	Resto da Divisão Inteira

Exemplos:

$2 + 5$	→ o resultado é 7
$2 + 3 * 3$	→ o resultado é 11
$(2 + 3) * 3$	→ o resultado é 15
$11 / 2$	→ o resultado é 5.5
$11 \text{ DIV } 2$	→ o resultado é 5 (Quociente de divisão de 11 por 2)
$11 \text{ MOD } 2$	→ o resultado é 1 (Resto da divisão de 11 por 2)

PRIORIDADE NA AVALIAÇÃO DE EXPRESSÕES

- 1º Parênteses ();
- 2º *, /, div e Mod;
(resolvidos da esquerda para a direita)
- 3º + e -

EXERCÍCIOS

1) Relacione as colunas:

- | | |
|----------------------------|--|
| a) Variáveis | <input type="checkbox"/> Tipo de dado numérico. |
| b) Inteiro | <input type="checkbox"/> Classificação dos dados que a variável receberá segundo uma linguagem de programação. |
| c) Declaração de variáveis | <input type="checkbox"/> Utilizados para fornecer dados a uma variável. |
| d) Tipo de dados | <input type="checkbox"/> é um endereço de memória representada por um nome cujo conteúdo pode ser alterado no decorrer do algoritmo. |
| e) Operador de atribuição | <input type="checkbox"/> Indicação do tipo de dados que as variáveis irão receber em um algoritmo. |

2) Na hora de criar o nome da variável (identificador ou rótulo) é necessário seguir regras. Marque com X apenas as regras verdadeiras.

- | | |
|--|---|
| <input type="checkbox"/> não pode usar símbolos especiais (% , & , * , - , / , etc). | <input type="checkbox"/> pode usar números para compor o restante do nome após iniciar com letra. |
| <input type="checkbox"/> pode haver espaços. | <input type="checkbox"/> Não pode haver espaços em branco. |
| <input type="checkbox"/> não pode iniciar com números. | <input type="checkbox"/> pode ter acento agudo ou crase. |
| <input type="checkbox"/> deve iniciar com uma letra ou <i>underline</i> | <input type="checkbox"/> O único símbolo especial permitido é o (_) <i>underline</i> |
| <input type="checkbox"/> pode iniciar com número. | |

3) Relacione as colunas criando uma correspondência entre dados e seus tipos.

- | | |
|------------------------|---|
| a) 4, 123.54 , 0, 1.03 | <input type="checkbox"/> literal com 1 caractere. |
| b) A, *, V, 2, f | <input type="checkbox"/> inteiro |
| c) Claudia, sapato. | <input type="checkbox"/> lógico |
| d) 1000, 0, 8, 34 | <input type="checkbox"/> literal com até 255 caracteres |
| e) .V., .Falso., .F. | <input type="checkbox"/> real |

4) Sendo P, Q, R e S, variáveis numéricas, cujos conteúdos são iguais a 2, 3, 12 e 4.5, respectivamente, quais os valores fornecidos por cada uma das expressões abaixo.

- a) $100 * Q \text{ DIV } P + R$
- b) $P * R \text{ MOD } 5 - Q/2$
- c) $Q * Q - R/4 * P - 3$
- d) $S \text{ MOD } (P+1) - Q * R$
- e) $S + R * (3 - 2 * P)/5 + 5 * Q$

5) Analise os trechos dos algoritmos abaixo e faça a declaração de variáveis e responda quais as saídas de dados após a execução dos mesmos:

a)

$A \leftarrow 10;$

$B \leftarrow 20$

Escreva(B);

$B \leftarrow 5;$

Escreva (A, B);

A	B	Saída

b)

$A \leftarrow 30;$

$B \leftarrow 20;$

$C \leftarrow A + B;$

Escreva(C);

$B \leftarrow 10;$

Escreva(B, C);

$C \leftarrow A + B;$

Escreva (A, B, C);

A	B	C	Saída

c)

$A \leftarrow 10;$

$B \leftarrow 20;$

$C \leftarrow A;$

$A \leftarrow B;$

$B \leftarrow C;$

A	B	C	Saída

d)

$A \leftarrow 5;$

$B \leftarrow A + 1;$

$A \leftarrow B + 1;$

$B \leftarrow A + 1;$

Escreva (A);

$A \leftarrow B + 1;$

Escreva (A,B);

A	B	Saída

f)

$X \leftarrow 1;$

$Y \leftarrow 2;$

$Z \leftarrow Y - X;$

Escreva (Z);

$X \leftarrow 5;$

$Y \leftarrow X + Z;$

Escreva (X< Y< Z);

X	Y	Z	Saída

e)

$i \leftarrow 10;$

$j \leftarrow 5;$

$k \leftarrow i + j;$

$j \leftarrow 20;$

$i \leftarrow 10$

Escreva (i, j, k);

i	j	k	Saída