

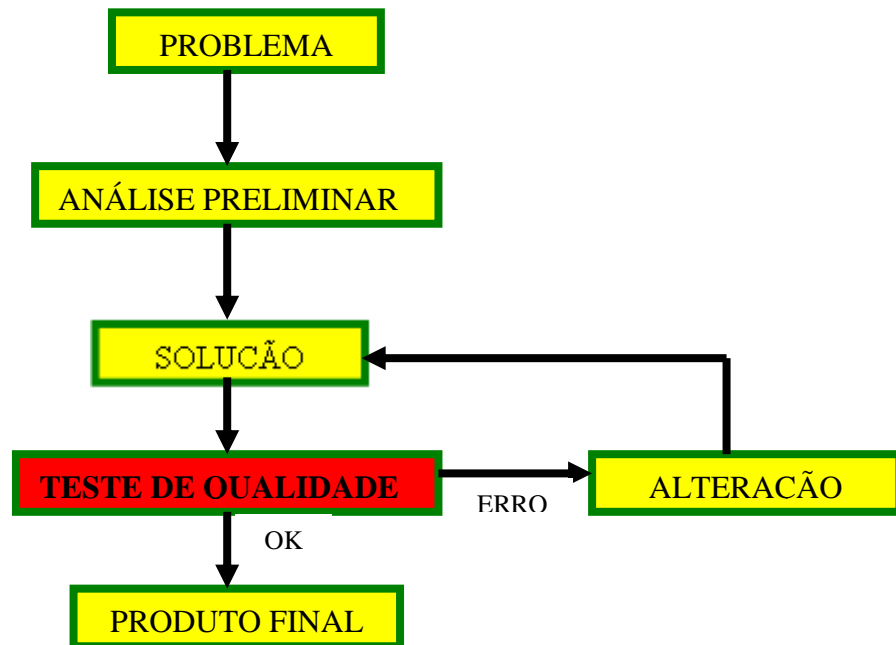
CONSTRUÇÃO DE ALGORITMOS

2.1 CONCEITOS

ALGORITMO é uma seqüência finita de ações/instruções que descrevem como um problema deve ser resolvido. Quando as ações de um algoritmo obedecem à sintaxe de uma linguagem de programação passamos a chamá-lo de PROGRAMA.

INSTRUÇÕES (ordens) são frases que indicam ações a serem executadas. São compostas por um verbo no imperativo mais um complemento.

2.2 COMO CONSTRUIR ALGORITMOS



2.3 TIPOS DE PROCESSAMENTO

1. Processamento Seqüencial: as instruções em um algoritmo são executadas uma após a outra, sem que haja desvio na seqüência das instruções.
2. Processamento com Repetição: conjunto de instruções (ou uma só) que é executado um determinado número de vezes.
3. Processamento Condicional: o conjunto de instruções (ou uma só) é executado ou não. A sua execução depende de uma condição.

2.4 CONCEITO DE VARIÁVEL

Um algoritmo para ser executado possui três fases

1. Coleta de informações necessárias,
2. Processamento,
3. Apresentação dos resultados.

Para que haja o processamento, é necessário que o algoritmo e as informações estejam na Memória Principal. (Se eu sei como fazer, mas não tenho as informações ou tenho as informações, mas não sei como fazer, não haverá processamento).

Memória Principal como uma matriz (MP)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	POSIÇÕES DA MEMÓRIA OCUPADAS PELO ALGORITMO(PROGRAMA)																							
2			90																					
3																								
4																								
5																								
6																								
7																								
8															40									
9																								
10																								
11																								
12																								

O programa ocupa os endereços de 1,1 até 2,2, temos informações nos endereços 2,3 (90) e 8,15 (40). Para acessar as informações que estão na MP o programa usa o ENDEREÇO. Ficaria muito complicado escrever programas tendo que informar o ENDEREÇO das informações. Para facilitar, utilizamos um NOME para referenciar a informação. O NOME é o meio de levar e ou de trazer a informação da MP. A esse nome dado à informação chamamos de **VARIÁVEL**. Quando damos um nome para informação (ões), variável, ou variáveis, devemos levar em conta que essa variável deve sempre, via nome, dar uma dica sobre a informação que ela está levando ou trazendo da MP.

Para associar uma informação a uma variável usaremos a ATRIBUIÇÃO:

NOTA := 80

NOFI := NOTA

Que deve ser lido: Leve a quantidade 80 para a MP associando à NOTA, ou NOTA recebe 80, ou atribua 80 à NOTA; no segundo exemplo, retire da MP o conteúdo associado à NOTA e leve para MP associando a NOFI, ou NOFI recebe NOTA, atribua NOTA à NOFI.

Ou, se for necessário atribuir um valor genérico qualquer para a variável NOTA, usaremos a instrução de leitura: (**OBSERVAÇÃO:USAREMOS AS INSTRUÇÕES DO PASCALZIM**)

READLN (NOTA);

READLN (NOME);

Que deve ser lido, para um mais fácil entendimento: no 1º caso, algum valor será digitado via teclado e esse valor atribuído à variável NOTA. No 2º caso (**ATENÇÃO a instrução de leitura READLN, interrompe o processamento. O processamento só continua após uma intervenção externa ao programa**)

Regras para atribuir nome as variáveis:

1. Deve sempre iniciar com uma letra;
2. Só pode utilizar os caracteres especiais: ponto (.) e underline (_);
3. Pode possuir os caracteres numéricos;
4. A quantidade de caracteres do nome depende da linguagem de programação;
5. A diferenciação entre maiúsculas e minúsculas, depende da linguagem de programação, no PASCALZIM não há diferenciação, logo, **NOTA** se refere ao mesmo endereço se for escrita **nota**.

2.5 TIPOS DE INFORMAÇÃO

1. Números

Representam quantidades. Basicamente, existem dois tipos de informação numérica:

Números Inteiros: 1 827 1235;

Números Reais: 1.35 827. -83.420 -5.2E7 4.8E-5

2. Caracteres

Representam as letras, os dígitos numéricos e os caracteres especiais. Constantes do tipo caracter são representadas entre aspas ‘ simples. Exemplos: ‘JUCA’, ‘FIM’, ‘837’

3. Valores lógicos (Booleanos)

Um valor lógico pode estar em apenas um dos estados: **Verdadeiro** TRUE (**V**) ou **Falso** FALSE (**F**).

Operações:

NÃO - NOT - Troca o estado

OU - OR - Se algum estado for verdadeiro, então verdadeiro

E - AND - Se algum estado for falso, então falso

A	B	A OU B	A E B	NÃO A	NÃO B	NÃO (A OU B)	NÃO (A E B)
V	V	V	V	F	F	F	F
V	F	V	F	F	V	F	V
F	V	V	F	V	F	F	V
F	F	F	F	V	V	V	V

2.6 EXPRESSÕES

Uma característica importante nos algoritmos e nos programas de computador é a capacidade de resolver expressões, que basicamente são de três tipos: **Aritmética**, **Relacional** e **Lógica**.

Aritmética: operações com informações do tipo numérico e que tem como resultado um valor numérico. Operadores: **+** (**soma**), **-** (**subtração**), ***** (**multiplicação**), **/** (**divisão – resultado tipo REAL**), **DIV** (**divisão – resultado tipo INTEIRO**) e **MOD** (**resto de uma divisão**)

Relacional: operações de comparação **entre dois valores de mesmo tipo** e que tem como resultado um **valor lógico**.

Operadores:

= : igual
> : maior
< : menor
<> : diferente
>= : maior ou igual
<= : menor ou igual.

Lógica: operações lógicas entre valores lógicos, e que tem como resultado um valor lógico. Operadores: **E - AND**, **OU - OR** e **NÃO - NOT**.

2.7 ENTRADA/SAÍDA

Todo algoritmo/programa para computador possui instruções para levar e trazer informações da M.P.

1. **Leitura**: Transporta informações de um periférico (teclado, mouse, etc.) para a M.P. (via variável). **A instrução READLN interrompe o processamento - não esqueça....**

READLN (NOTA, CUR);

2. **Apresentação**: Transporta a informação da M.P. (via variável) para um periférico (impressora, monitor, etc.)

WRITELN (‘PRIMEIRA NOTA:’, NOTA, ‘CURSO:’, CUR);

ATENÇÃO GENTE...

Apesar de parecer pouco material, **todos, eu escrevi todos** os conceitos tem que ser compreendidos.

ATENÇÃO 2...

Não deixe passar dúvidas.

Professor Luiz Antonio