

TIPOLOGIA E VARIÁVEIS

A função de um computador é processar as informações que nós passamos para ele. E essas informações são compostas por dois tipos de conteúdos:

- Dados (numéricos, caracteres e lógicos).
- Instruções.

Lembre-se: quando desejar utilizar os dados dentro de um programa, você vai utilizar as variáveis.

VARIÁVEL é um tipo de estrutura mutável, inconstante. Ela recebe um valor que ela não saberá qual é, mas saberá o tipo de dado.

A variável pode assumir qualquer um dos valores de um determinado conjunto de valores.

$$A - B = D \text{ e } A + B = C$$

REGRAS PARA NOMEAR UMA VARIÁVEL:

- Atribuir um ou mais caracteres (ex: matrícula, CPF).
- Sem espaços em branco.
- Buscar utilizar nomes que queiram dizer alguma coisa.

PAPEIS DE UMA VARIÁVEL: ação ou controle.

O QUE É UMA VARIÁVEL CONSTANTE?

Tudo aquilo que é fixo ou estável. Alguns tipos de variáveis constantes: π (3,14) ou *resultado = receber * 0.10*

Ou seja, a constante é a fórmula matemática ou o valor numérico que não vai ser alterado.

INSTRUÇÕES PRIMITIVAS

Instruções determinam as ações que iremos executar em cima dos nossos dados. Geralmente são cálculos matemáticos onde utilizamos operadores. Esses operadores podem ser binários ou unários.

ESTRUTURAS CONDICIONAIS E OPERADORES

A Estrutura Condicional possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas. A Estrutura Condicional pode ser Simples, Composta ou Encadeada.

Ou seja, dado o estado de uma pessoa ou coisa existe uma condição para aquilo acontecer.

Portanto, expressa uma condição ou suposição. Ou seja, a uma condição que se satisfeita executa uma determinada instrução.

OPERADORES RELACIONAIS:

= ➡ Igual a
<> ➡ Diferente de
> ➡ Maior que
< ➡ Menor que
>= ➡ Maior ou igual a
<= ➡ Menor ou igual a

CONDIÇÃO SIMPLES:

Se (<condição>) então
<instruções para condição verdadeira>
Fim_se

CONDIÇÃO COMPOSTA:

Se (<condição>) então
<Instruções para condição verdadeira>
Senão
<Instruções para condição falsa>
Fim_se

CONDIÇÃO ENCADEADO:

Se (<condição>) então
<Instruções para condição verdadeira>
Senão
Se (<condição 2>) então
<Instruções para condição 2 verdadeira e condição 1 falsa>
Senão
<Instruções para condição 1 e 2 falsas>
Fim_se

OPERADORES LÓGICOS

São eles AND, OR e NOT. Devem ser utilizados quando você precisa de uma resposta simplificada como Verdadeiro ou Falso.

OPERADOR AND: ele verifica as entradas que devem ser satisfeitas. E ele é sempre falso se for UM OU OUTRO. Porém é verdadeiro quando os dois estão satisfeitos. Está associado à interseção.

Exemplo: Curso de inglês

Se (gramatica.e.conversacao) então

Escreva "aprovado"

Senão

Escreva "reprovado"

OPERADOR OR: Precisa que apenas uma das condições seja verdadeira para o resultado ser verdadeiro. Somente se ambas forem falsas, o resultado será falso. Está associado à união.

Exemplo: Chuva

Se (choveu .ou. grama_molhada) então

Escreva "Plantas regadas"

Senão

Escreva "Regar plantas"

OPERADOR NOT: Inversão do resultado lógico é o resultado da operação. É um operador de negação. Portanto, se tem uma condição que é verdadeira, ela se tornará falsa. E vice-versa.

Exemplo:

Not B -> tudo que não está em B

Ou seja, tudo que está em A e não pertence a B.

ESTRUTURAS DE REPETIÇÃO

Quando você precisar que um pedaço do código seja repetido mais de uma vez. Sendo as vantagens:

- Redução de linhas.
- Compreensão facilitada
- Redução de erro.

Exemplo Repita: procurar um artigo ➡ virar a página ➡ analisar conteúdo E se repetir o final: virar a página ➡ analisar o conteúdo.

Portanto, esse é um ciclo que vai se repetir até você encontrar o que quer.

O Teste lógico é no final e o número de repetições é indefinida.

Exemplo Enquanto: Grama alta ➡ Aparar a grama ➡ Analisar a grama

OBS: Se nessa análise notar que tiver grama ainda, vc repete o ciclo: aparar a grama ➡ analisar a grama

O teste lógico é no início e o número de repetições indefinidas.

Exemplo Para ... De... Até:

Somatorio = 0

Para inicio = 1 até 10

Somatorio= somatorio + inicio

Escreva somatorio

Fim para

O teste lógico será no início e o número de repetições é definido.

VETORES E MATRIZES

Vetores é uma variável que possui uma sequência, ou seja, um vetor é caracterizado por uma variável dimensionada com tamanho pré-fixado.

Matriz é uma tabela organizada em linhas e colunas no formato de $m \times n$, onde m representa o nº de linhas (horizontal) e n o número de colunas (vertical).

O QUE SÃO FUNÇÕES?

Também chamada de subalgoritmo, subprograma, bloco, método...

Vai implicar que um elemento A esteja conectado a um elemento de um conjunto B.

Portanto, as funções são blocos de instruções que realizam tarefas específicas. Ou seja, então são trechos de códigos com tarefas/objetivos específicos que podem ser chamados dentro do seu código principal.

Vem da modularização do algoritmo.

BENEFÍCIOS DA MODULARIZAÇÃO DO PROBLEMA?

- ☐ O Código fica mais claro e conciso.
- ☐ Consegue fazer a reutilização de instruções ou códigos.
- ☐ Ao invés de copiar e colar o código você utiliza uma função onde o trecho vai ficar ali. Assim, você não precisará alterar vários pontos no seu código. Apenas no primeiro.

As funções na programação são os blocos de instruções (código), identificados por nomes e parâmetros.

A assinatura da função é definida pelo nome da função e pelos parâmetros que ela possui.

Exemplo determinar a média escolar através de uma função:

```
funcao mediaescolar(nota1,nota2)
```

```
    resultado = 0
```

```
    resultado = (Nota1 + Noata2)/2
```

```
    retome resultado
```

```
fim funcao
```