

Apostila de Portugol

TIPOS PRIMITIVOS DE DADOS

INTEIRO	ADMITE SOMENTE NÚMEROS INTEIROS. GERALMENTE É UTILIZADO PARA REPRESENTAR UMA CONTAGEM (QUANTIDADE).
REAL	ADMITE NÚMEROS REAIS (COM OU SEM CASAS DECIMAIS). GERALMENTE É UTILIZADO PARA REPRESENTAR UMA MEDIÇÃO.
CARACTERE	ADMITE CARACTERES ALFANUMÉRICOS. OS NÚMEROS QUANDO DECLARADOS COMO CARACTERES TORNAM SE REPRESENTATIVOS E PERDEM A ATRIBUIÇÃO DE VALOR.
LÓGICO	ADMITE SOMENTE VALORES LÓGICOS(VERDADEIRO/FALSO).

COMANDOS DE I/O (INPUT/OUTPUT)

LER → Comando de entrada que permite a leitura de Variáveis de Entrada.

ESCREVER → Comando de saída que exibe uma informação na tela do monitor.

IMPRIMIR → Comando de saída que envia uma informação para a impressora.

SINAL DE ATRIBUIÇÃO

Uma Variável nunca é eternamente igual a um valor, seu conteúdo pode ser alterado a qualquer momento. Portanto para atribuir valores a variáveis devemos usar o sinal de ":=".

Exemplos:

A := 2;

B := 3;

C := A + B;

SINAL DE IGUALDADE

As constantes são eternamente iguais a determinados valores, portanto usamos o sinal de "=".

Exemplos:

PI = 3.1416;

Empresa = 'Colégio de Informática L.T.D.A.'

V = Verdadeiro

CORPO GERAL DE UM PROGRAMA

PROGRAMA <<identificador>>;

CONST

<<identificador>> = <<dado>>

VAR

<<identificador>> : <<tipo>>;

ÍNICIO

{

COMANDOS DE ENTRADA, PROCESSAMENTO E SAÍDA

<<comando1>>;

<<comandoN>>

}

FIM.

ESTRUTURAS SEQUÊNCIAIS

Como pode ser analisado no tópico anterior, todo programa possui uma estrutura seqüencial determinada por um ÍNICIO e FIM.

; PONTO E VÍRGULA ;

O sinal de ponto e vírgula ";" indica a existência de um próximo comando (passa para o próximo).

Na estrutura ÍNICIO e no comando que antecede a estrutura FIM não se usa ";".

PRIMEIRO ALGORITMO

Segue um Algoritmo que lê o nome e as 4 notas bimestrais de um aluno. Em seguida o Algoritmo calcula e escreve a média obtida.

```
PROGRAMA MEDIA_FINAL;
```

```
VAR
```

```
    NOTA1, NOTA2, NOTA3, NOTA4, MEDIA: INTEIRO;
```

```
    NOME : CARACTERE [35]
```

```
INICIO
```

```
    LER (NOME);
```

```
    LER (NOTA1, NOTA2, NOTA3, NOTA4);
```

```
    MEDIA := (NOTA1 + NOTA2 + NOTA3 + NOTA4) / 4;
```

```
    ESCREVER (NOME, MEDIA)
```

```
FIM.
```

SEGUNDO ALGORITMO

Segue um Algoritmo que lê o raio de uma circunferência e calcula sua área.

```
PROGRAMA AREA_CIRCUNFERENCIA;  
  
CONST PI = 3.1416;  
  
VAR RAIIO, AREA : REAL;  
  
INICIO  
  
    LER (RAIO); {PROCESSAMENTO}  
  
    AREA := PI * SQR(RAIO); {ENTRADA}  
  
    ESCREVER ('AREA =', AREA) {SAÍDA}  
  
FIM.
```

'ASPAS SIMPLES'

Quando queremos exibir uma mensagem para a tela ou impressora ela deve estar contida entre aspas simples, caso contrário, o computador irá identificar a mensagem como Variável Indefinida.

Exemplo:

```
ESCREVER ('AREA OBTIDA =', AREA) {COMANDO DE SAÍDA}  
  
AREA OBTIDA = X.XX {RESULTADO GERADO NA TELA}
```

ESTRUTURAS DE DECISÃO

Executa uma seqüência de comandos de acordo com o resultado de um teste.

A estrutura de decisão pode ser Simples ou Composta, baseada em um resultado lógico.

Simples:

```
SE <<CONDIÇÃO>>  
  
    ENTÃO <<COMANDO1>>
```

Composta 1:

```
SE <<CONDIÇÃO>>  
  
    ENTÃO <<COMANDO1>>  
  
    SENÃO <<COMANDO1>>
```

Composta 2:

```
SE <<CONDIÇÃO>>  
  
    ENTÃO INICIO  
  
        <<COMANDO1>>;  
  
        <<COMANDON>>  
  
    FIM;  
  
    SENÃO INICIO  
  
        <<COMANDO1>>; <<COMANDON>>  
  
    FIM;
```

ALGORITMO TRÊS

Segue um Algoritmo que lê 2 números e escreve o maior.

```
PROGRAMA ACHA_MAIOR;
```

```
VAR A, B : INTEIRO;
```

```
INICIO
```

```
    LER (A, B);
```

```
    SE A>B
```

```
        ENTÃO ESCREVER (A)
```

```
    SENÃO ESCREVER (B)
```

```
FIM.
```

ALGORITMO QUATRO

Segue um Algoritmo que lê o nome e as 4 notas bimestrais de um aluno. Em seguida o Algoritmo calcula e escreve a média obtida pelo aluno escrevendo também se o aluno foi aprovado ou reprovado.

Média para aprovação = 6

```
PROGRAMA MEDIA_FINAL;
```


VAR

NOTA1, NOTA2, NOTA3, NOTA4, MEDIA: REAL;

NOME : CARACTERE [35]

INICIO

LER (NOME);

LER (NOTA1, NOTA2, NOTA3, NOTA4);

MEDIA := (NOTA1 + NOTA2 + NOTA3 + NOTA4) / 4;

SE MEDIA >= 6

ENTÃO ESCREVER ('APROVADO')

SENÃO ESCREVER ('REPROVADO')

ESCREVER (NOME, MEDIA)

FIM.

NINHOS DE SE

Usados para tomadas de decisões para mais de 2 opções.

Forma Geral:

SE <<CONDIÇÃO>>

ENTÃO <<COMANDO1>>

SENÃO SE <<CONDIÇÃO>>

ENTÃO <<COMANDO1>>

SENÃO <<COMANDO1>>

ALGORITMO CINCO

Segue um Algoritmo que lê 3 números e escreve o maior.

PROGRAMA ACHA_MAIOR;

VAR A, B, C : INTEIRO;

INICIO

LER (A, B, C);

SE (A>B) E (A>C)

ENTÃO ESCREVER (A)

SENÃO SE (B>A) E (B>C)

ENTÃO ESCREVER (B)

SENÃO ESCREVER (C)

FIM.

ESTRUTURAS DE CONDIÇÃO

A estrutura de condição equivale a um ninho de SE'S.

Forma Geral:

FACA CASO

CASO <<CONDIÇÃO1>>

<<COMANDO1>>;

CASO <<CONDIÇÃO2>>

<<COMANDO1>>;

OUTROS CASOS

<<COMANDO1>>;

FIM DE CASO

ALGORITMO SEIS

Segue um Algoritmo que lê 3 números e escreve o maior.

PROGRAMA ACHA_MAIOR;

VAR A, B, C : INTEIRO;

INICIO

LER (A, B, C);

FACA CASO

CASO (A>B) E (A>C)

ESCREVER (A);

CASO (B>A) E (B>C)

ESCREVER (B);

OUTROS CASOS

ESCREVER (C);

FIM DE CASO

FIM.

ESTRUTURA DE REPETIÇÃO DETERMINADA

Quando uma seqüência de comandos deve ser executada repetidas vezes, tem-se uma estrutura de repetição.

A estrutura de repetição, assim como a de decisão, envolve sempre a avaliação de uma condição.

Na repetição determinada o algoritmo apresenta previamente a quantidade de repetições.

Forma Geral 1:

PARA <<VARIÁVEL DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE
<<VALOR FINAL>> FAÇA

<<COMANDO1>>;

Forma Geral 2:

PARA <<VARIÁVEL DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE
<<VALOR FINAL>> FAÇA

ÍNICIO

<<COMANDO1>>;

<<COMANDON>>

FIM;

A repetição por padrão determina o passo do valor inicial até o valor final como sendo 1. Determinadas linguagens possuem passo -1 ou permitem que o programador defina o passo.

ALGORITMO SETE

Segue um algoritmo que escreve 10 vezes a frase "VASCO
DA GAMA"

PROGRAMA REPETICAO;

VAR I:INTEIRO

INICIO

PARA I :=1 ATE 10 FACA

ESCREVER ('VASCO DA GAMA')

FIM.

VARIÁVEL IMPLEMENTADA DE



ALGORITMO OITO

Segue um algoritmo que escreve os 100 primeiros números pares.

PROGRAMA PARES;

VAR I,PAR: INTEGER;

INICIO

PAR:=0;

PARA I:=1 ATE 100 FACA

INICIO

ESCREVER (PAR);

PAR := PAR+2

FIM

FIM.

ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO

INICIAL

É usada para repetir N vezes uma ou mais instruções. Tendo como vantagem o fato de não ser necessário o conhecimento prévio do número de repetições.

Forma Geral 1:

ENQUANTO <<CONDIÇÃO>> FACA

<<COMANDO1>>;

VALIDAÇÃO



Forma Geral 2:

ENQUANTO <<CONDIÇÃO>> FACA

ÍNICIO

<<COMANDO1>>;

<<COMANDON>>

FIM;

ALGORITMO NOVE

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

```
PROGRAMA SOMA_SALARIOS;  
  
VAR SOMA, SALARIO : REAL;  
  
INICIO  
  
    SOMA:=0;  
  
    SALARIO:=1;  
  
    ENQUANTO SALARIO>=0  
  
        INICIO  
  
            LER (SALARIO);  
  
            SOMA:=SOMA+SALARIO  
  
        FIM;  
  
    ESCREVER (SOMA)  
  
FIM.
```

**TODAS AS VARIÁVEIS QUE ACUMULAM VALORES DEVEM
RECEBER UM VALOR INICIAL.**

ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO

FINAL

Assim como a estrutura ENQUANTO É usada para repetir N vezes uma ou mais instruções.

Sua validação é final fazendo com que a repetição seja executada pelo menos uma vez.

Forma Geral;

REPITA

<<COMANDO1>>;

<<COMANDON>>

ATE <<CONDIÇÃO>>

ALGORITMO DEZ

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

PROGRAMA SOMA_SALARIOS;

VAR

SOMA, SALARIO : REAL;

```
INICIO

    SOMA:=0;

    REPITA

        LER (SALARIO);

        SOMA:=SOMA+SALARIO

    ATE SALARIO<0;

    ESCREVER (SOMA)

FIM.
```

ALGORITMO ONZE

Segue um algoritmo que escreve os 100 primeiros números pares.

```
PROGRAMA PARES_2;

VAR I, PAR, CONTADOR : INTEIRO;

INICIO

    CONTADOR := 0;

    PAR := 0;

    REPITA

        ESCREVER (PAR);
```

```
PAR := PAR+2;  
  
CONTADOR := CONTADOR+1;  
  
ATE CONTADOR=100  
  
FIM.
```

Programas Equivalentes

O algoritmo onze poderia ter sido criado com qualquer estrutura de repetição. Portanto podemos ter algoritmos que são escritos de maneiras diferentes, mas, funcionam realizando o mesmo objetivo.