APÊNDICE A - FUNDAMENTOS DA LINGUAGEM DE PROGRAMAÇÃO PASCAL.

1 . INTRODUÇÃO

Este apêndice tem o objetivo de fornecer as regras básicas para transformar algoritmos (em Natural) em programas (em PASCAL).

Um programa Pascal é constituído por três áreas distintas, o cabeçalho, as declarações e as instruções. A forma geral de um programa Pascal obedece o seguinte formato:

```
PROGRAM nome_do_programa ;
    <declarações de forma geral>
    <rotinas>
    BEGIN
        <comando_1>;
        <comando_2>;
        .....
        <,comando_n>;
    END.
```

2 - ITENS SINTÁTICOS EM PASCAL

2.1 - SÍMBOLOS (Capítulo 2, Item 2.2.1)

LETRAS: A B C ... U W V X Y Z (MAIÚSCULAS ou minúsculas)

DÍGITOS: 0 1 2 ... 9

Operadores :

Relacionais : < , < =, > , > = , = e <>

Aritméticos : + , - , * , / , DIV MOD

Lógicos : AND, OR, NOT

Outros : IN , + (concatenação de cadeias)

Caracteres especiais : ‘ ( ) , . : := ; [ ] { } .. < >

OBS: Os símbolos anteriormente citados não constituem uma lista exaustiva dos disponíveis na linguagem Pascal.

2.2 - Identificadores

- denotam constantes, tipos, variáveis, procedimentos, funções , unidades (unit) e programas

Os identificadores obedecem as seguintes regras de formação:

- Tamanho : até 64 caracteres alfanuméricos (reconhecidos).
- O primeiro caracter é obrigatoriamente uma letra.
- Pode ser também utilizado o caracter de quebra (_).

2.2.1 Identificador "CONSTANTES".

Pode- se utilizar constantes como habitualmente usa-se, ou seja, inserindo no corpo do programa ou declarando através de expressões que definem seu conteúdo , por exemplo :

```
CONST
    kbyte  = 1024 ;
    pi     = 3.1416;
    sim    =  TRUE;
```

As constantes assumem o tipo de seu conteúdo mas , pode serem tipadas.

```
CONST
    s : STRING[12] = 'LITERAL';
```

2.2.2 *Identificador "VARIÁVEIS"*

<u>Tipos de Variáveis Pré-definidos</u>.

As variáveis podem ser classificadas em quatro tipos : Numéricas, Alfanuméricas e Lógicas.

No pascal recebem a seguinte terminologia:

BOOLEAN, CHAR , INTEGER, LONGINT , REAL e STRING.

A declaração de variáveis tem a seguinte sintaxe :

VAR

nome_da_variável [, nome_da_variável.....] : tipo;

. . . . . .

nome_da_variável[, nome_da_variável.....]:tipo;

por exemplo:

VAR

valor_1, valor_2 : **real ; { [2.9 E –39, 1.7 E 38] }**

resp : **char ; { 1 Byte}**

quant : **integer ; {[32768 , 32767] }**

n : **longint ; { [-2147483648 , 2147483647]}**

palavra : **string[20]** ; **{ 2 a 256 Bytes }**

logica : **boolean ; { 1 Byte TRUE/FALSE}**

2.3 – COMPILAÇÃO

COMPILAÇÃO SEM ERRO:

A Tradução dos comandos dos para a linguagem de máquina é feita, verificando para cada comando se foram observadas as regras que limitam as construções dos mesmos. Caso alguma regra tenha sido desobedecida, o programa Compilador fornecerá a descrição do erro detectado e passará a traduzir o próximo comando. Ao atingir o comando END, o Compilador saberá que terminaram os comandos a traduzir.

Caso todos os comandos tenham sido traduzidos para a linguagem de máquina com sucesso, isto é, nenhum erro foi cometido, o programa estará apto a ser executado, Todos estes passos podem ser realizados através das teclas CTRL mais a tecla de função F9. A execução é começada pelo primeiro comando executável, neste caso, pelas especificações de tipo de variáveis. A execução prossegue lendo, imprimindo, calculando, conforme os comandos ordenados, até atingir o comando END quando o computador pára o programa,

COMPILAÇÃO COM ERRO

Se, ao atingir o comando END o compilador detectou algum erro, o computador devolve ao usuário uma listagem do programa com a indicação do(s) erro(s) cometido(s), para que se providencie as correções.

(Rever - ETAPAS DO PROCESSAMENTO DE UM PROGRAMA).

.

Exemplo

a) Em NATURAL

ALGORTIMO ANEL

NUMHORAS : INTEIRA

RAIOEXT, RAIOINT, SALARIO, SALBRU : REAL

ALTURA, LARGURA, COMPRIM : REAL

NOME*30, CIDADE*20 : CADEIA

b) Em PASCAL

```
PROGRAM ANEL;
VAR
  NUMHORAS: INTEGER;
  RAIOEXT,RAIOINT,SALÁRIO,SALARIOBRU:REAL;
  ALTURA,LARGURA,COMPRIMENTO:REAL;
  CIDADE:STRING[20];
  NOME:STRING[30];
```

4 - ESTRUTURA DE SEQUÊNCIA EM PASCAL

A estrutura de sequência é constituída por um conjunto de comandos de ENTRADA, SAÍDA, ATRIBUIÇÃO e COMENTÁRIO, que são executados na ordem em que são escritos.

a) Comando de ENTRADA em PASCAL .

Comandos de Entrada.

Este procedimento permite a entrada de dados oriunda de diversos dispositivos . O dispositivo padrão de entrada é o console (teclado) sua sintaxe geral é :

READLN ([var <f>: TEXT; ,v1. [,v2....,,vn.]);

exemplos:

READLN (S);

READLN (A,B);

EXEMPLOS

- LEIA, CIDADE, ESTADO, HABIT ---> READ(CIDADE, ESTADO, HABIT)

- LEIA, RAIOEXT, RAIOINT --------> READLN(RAIOEXT, RAIOINT)

b) Comando de SAÍDA em PASCAL

Procedimento que permite escrever em um dispositivo de saída, sua sintaxe geral é :

WRITE ([varf : text,] v1 [,v2....., vn]);

a princípio usaremos somente o dispositivo padrão de E/S , ou seja , a CONSOLE (vídeo + teclado). Este comando imprime e mantém o cursor onde ele parou.

Outro procedimento é ***WRITELN,*** que é idêntico ao anterior exceto que ele e imprime a linha corrente e posiciona o cursor no início da próxima linha.

EXEMPLOS

- ESCREVA, "RAIO EXTERNO .....:", RAIOEXT, "CM"

  ------> WRITELN('RAIO EXTERNO .....:', RAIOEXT, 'CM')

- ESCREVA, "DEVO ESTUDAR COMPUTACAO"

  ------> WRITELN('DEVO ESTUDAR COMPUTACAO')

Obs: Para exibir uma mensagem de texto deve-se usar apóstrofo, em lugar de aspas, no comando WRITE.

Considere o seguinte segmento de programa:

```
A := 1;
B := 2;
C:= 3;
D:=10;
E:= 2;
F:=100;
X := 42.51;
MATERIA := 'COMPUTACAO';
WRITELN (A,B,C);                 {123}
WRITELN (A,' ',B,' ',C);         {1 2 3}
WRITELN ('A',MATERIA);   {ACOMPUTACAO}
WRITELN ('A ',MATERIA);  {A COMPUTACAO}
WRITELN (D,E,F);                 {102100}
WRITELN (D:2,E:2,F:2);           {10 2100}
```

WRITELN (D:3,E:3,F:3); **{ 10 2100}**

WRITELN (D,E:2,F:4); **{10 2 100}**

WRITELN (X); **{4.2510000000E+01}**

WRITELN (X:5:3); **{42.51}**

c) Comando de ATRIBUIÇÃO em PASCAL

A maioria das operações básicas são comandos de atribuição cuja a sintaxe é:

identificador := expressão ;

Em Natural: variável := expressão

Em PASCAL: idêntico

d) Comando de COMENTÁRIO em PASCAL

Em Natural: C comentário (C na primeira coluna da linha)

Em PASCAL : (* *) ou { }

EXEMPLO DE ALGORITMO COMPLETO EM PASCAL (USANDO ESTRUTURA DE SEQUÊNCIA)

```
PROGRAM VERIFICA;
        VAR
            A,B : INTEGER;
            TESTE : BOOLEAN;
        BEGIN
            WRITE (‘Entre com dois números : );
            READLN (A,B):
            TESTE :=  A > B;
            WRITELN (‘A É MAIOR QUE B’, TESTE);
        END.
```

REGRAS DE PRECEDÊNCIA DE OPERADORES.

1) ( )

2) - (prefixo negativo)

3) NOT

4) * / DIV MOD AND

5) + - OR

Exemplo e considerações:

- ALT-F5 - Exibe a tela do usuário. (após rodar um programa mostra os resultados).

- Para elevar ao quadrado e extrair a raiz quadrada de um número x utiliza-se respectivamente (p.ex.) Q := **SQR (x)** ; e RQ := **SQRT (x)** ;

- A potenciação , N^X pode ser obtida usando poten := **EXP (X *(LN N))**;

```
PROGRAM EXPO;
        USES CRT;
        VAR
         EXPON,X,N:REAL;
        BEGIN
         CLRSCR;
         READ (N,X);
         EXPON := EXP(X* (LN(N)));
         WRITE ('EXPONENCIAL DE  ',N:5:2,' ELEVADO A ',X:5:2, ' EH
',EXPON:6:2);
        END.
```

- Exemplo de um algoritmo transformado em um programa na linguagem Pascal

```
algoritmo média;
inteira n1, n2, n3;
real resp;
início
        escreva (‘Informe 3 nros. inteiros:’);
        leia (n1,n2,n3);;
        resp := n1 + n2 + n3;
        resp := resp / 3;
        escreva (‘Media:’, resp);
fim.
```

```
program media;
var n1, n2, n3: integer;
    resp: real;
begin
        writeln(‘Informe 3 nros. inteiros:’);
        read(n1,n2,n3);
        resp := n1 + n2 + n3;
 resp := resp / 3;
 write(‘Media:’, resp);
end.
```

- Identificadores:

  - nomes que possuem um significado no programa

  - 2 tipos:

a) Palavras reservadas: nomes pré-definidos da LP Pascal (comandos, cabeçalhos, seções de declaração). Exemplos: write, program, var, read.

b) Definidos pelo programador: quaisquer outros nomes (de variáveis, constantes, nome do programa, etc.). Apresentam algumas regras:

- não podem ser declarados 2 ou mais nomes iguais
- não podem ser uma palavra reservada

- ***Organização de um programa***

a) Comentários.

- úteis na documentação de programas (melhor compreensão)
- descrito entre os caracteres: { e } ou (* e *)

b) Indentação.

- organização da edição (digitação) de um programa (melhor legibilidade)
- envolve: separação de comandos por linhas e tabulação

- **Funções pré-definidas da linguagem Pascal.**

SQR (X), SQRT(X), EXP(X), SIN(X) ,COS(X), LN(X)

TRADUÇÃO DE COMANDOS DA LINGUAGEM ALGORÍTMICA PARA A LINGUAGEM PASCAL

ALGORITMO	***PROGRAMA PASCAL***
SE *condição* **ENTÃO** *comando n ;*	**IF** *condição* **THEN** *comando n ;*
SE *condição* **ENTÃO início** *comando 1 ;* ..... *comando n;* **fim**	**IF** *condição* **THEN BEGIN** *comando 1 ;* ..... *comando n ;* **END**
SENÃO início *comando 1 ;* ..... *comando n ;* **fim**	**ELSE BEGIN** *comando 1 ;* ..... *comando n ;* **END;**
CASO *expressão* *ctt1* **:** *comando 1 / bloco* *ctt2* **:***comando 2 / bloco* **.....** *cttn* **:***comando m/ bloco* ***;*** ***[*SENÃO** *comando / bloco* **]*** **FIM**	**CASE** *expressão* **OF** *ctt1* **:** *comando 1/ bloco* **;** *ctt2* **:***comando 2/ bloco* **;** **.....** *cttn* **:***comando m / bloco* ***[ELSE** comando / bloco **;]*** **END;**
ENQUANTO *condição* **FAÇA** *comando / bloco;*	**WHILE** *condição* **DO** comando / bloco**;**

REPITA	**REPEAT**
comando 1 ;	*comando 1 ;*
.....	.....
comando n;	*comando n ;*
ATE condição	**UNTIL** *condição ;*
PARA var de vlr1 ate vlr2 **FAÇA**	**FOR** *var* **:=** *vlr1* {**TO** */* **DOWNTO**} *vlr2* **DO**
comando / bloco;	*comando / bloco;*