INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Linguagem Pascal

Prof. Jonatas Bastos

Email: jonatasfbastos@gmail.com

Site: http://jonatasfbastos.wordpress.com/

Linguagem Pascal

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Foi desenvolvida pelo prof. Niklaus Wirth como uma linguagem simples e poderosa para ser usada em ambientes educacionais.
- Para alguns autores, por ter sido desenvolvida para fins educacionais, Pascal oferece recursos de entrada e saída limitados;
- Turbo Pascal Versão 7.0 corresponde ao ambiente de desenvolvimento de programas (IDE) que será utilizado na disciplina.
- Esse ambiente de desenvolvimento de programas foi criado pela empresa Borland International, Inc. e durante muito tempo foi considerado um dos melhores ambientes para desenvolvimento de programas Pascal.

Linguagem Pascal

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Integrated Design Environment - IDE
- Corresponde ao Ambiente de Desenvolvimento de Programas do Turbo Pascal 7.0 e permite: escrever, editar, compilar, rodar “run”, ligar “link” e depurar “debug” todos os programas desenvolvidos;

Estrutura Principal

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- A estrutura principal de um programa em pascal se apresenta da seguinte forma:
- **program** cabeçalho;

  declarações

  begin

  comandos

  end.
- As palavras em negrito são palavras reservadas ou palavras chave da linguagem, que não podem ser declaradas como identificadores;
- Os comentários são identificados por estarem delimitados pelos caracteres { e }, ou pelo caracteres (* e *). Comentários são ignorados pelo compilador na fase de tradução do programa;

Cabecalho

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Corresponde à primeira linha de um programa e sempre começa com a palavra reservada PROGRAM seguida de um nome e um ponto e vírgula (;), exemplo:

```
Program Oi;
Begin
    Writeln ( 'Oi' );
End.
```

Declarações

- Principais tipos pré-definidos da linguagem Pascal:
  - integer
  - char
  - boolean (TRUE, FALSE)
  - String
  - Real;
- Sintaxe:
- **var**
  lista-de-identificadores : tipo;
  lista-de-identificadores : tipo;
- Exemplo:
- **var**
  nota, A12 : real;
  i, codigo : integer;
  flag, Sim : boolean;
  letra1, letra2: char;

Operadores Aritméticos

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- A tabela seguinte mostra as operações aritméticas básicas da linguagem com sua prioridade de execução quando agrupadas em um expressão aritmética.

Prioridade	Operadores
1	* / div mod
2	+ -

Exemplos:

a =1, b = 2, c = 3
a + b * c = 7
c / b * a = 0.5
c div b = 1
c mod b 1

Expressões e OperadoresLógicos

- A tabela seguinte mostra as operações aritméticas básicas da linguagem com sua prioridade de execução quando agrupadas em um expressão aritmética.

=	igual
<>	diferente
<=	menor ou igual
<	menor
>=	maior ou igual
>	maior
IN	contido em (conjunto)

- Operadores Lógicos

and	conjunção
or	disjunção
not	negação

Comando de Atribuição

- Um comando de atribuição altera o conteúdo de um identificador (lado esquerdo) pelo valor resultante da expressão (lado direito). A variável e a expressão devem ser do mesmo tipo, exceto no caso em que a variável é do tipo real e o resultado da expressão é do tipo inteiro,
- identificador := expressão
- Exemplos:
- **var**
- a, c, n, soma, x , y : integer;
  k, media, total: real;
  cod, sim,teste : boolean;
  cor : string;

  k := 1;
  cor := ´verde´;
  teste := FALSE;
  a := b;
  media := soma/n;

Comandos de Entrada e Saída

- read(lista-de-identificadores);
- readln(lista-de-identificadores);
- write(lista-de-identificadores e/ou constantes e/ou expressões) ;
- writeln(lista-de-identificadores e/ou constantes e/ou expressões);
- Exemplos:
- read(x);
  readln(nome, n ,y);
  write(k, soma);
  writeln(21, ´Nome´, n);
  write(´Tabela de Preços´);
  writeln(n, sqrt(n));

Exemplo

```
program exe1;
var a, b : real;
  k : char;
begin
  a := 3.2;
  b := 5.81;
  k := ´*´;
  writeln('resultado: ', a:5, ' + ', b:5, ' = ', a+b:5, k);
  writeln('resultado: ',a:5:1, ' + ', b:5:1, ' = ', a+b:5:1, k:1);
  writeln('resultado: ',a:5:2, ' + ', b:5:2, ' = ', a+b:5:2, k:2);
  writeln('resultado: ',a:5:3, ' + ', b:5:3, ' = ', a+b:5:3, k:3);
end.
```

Saída:

resultado: 3.2e+00 + 5.8e+00 = 9.0e+00*

resultado: 3.2 + 5.8 = 9.0*

resultado: 3.20 + 5.81 = 9.01 *

resultado: 3.200 + 5.810 = 9.010 *

Estrutura de Seleção

- **if** condição **then**
  comando1
- Neste caso, comando1 só será executado se a condição for verdadeira;
- Um comando composto é formado por diversos comandos (simples ou compostos), delimitados pelas palavras BEGIN e END, além das estruturas de controle (condicional e de repetição).
- **if** condição **then**
  comando1
  else comando2
- Neste caso, se a condição for verdadeira, será executado o comando1. Caso contrário será executado o comando2.

Exemplo Estrutura de Seleção

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

```
program condicional;
        var i, j: integer;
begin
        i := 1;
        j := 3;
        if i < j then
                writeln('i menor do que j');
        else
                writeln('j maior do que i');
        if (i+j) > 0 then
                writeln('soma de i com j é maior do que 0');
end.
```

Estrutura Caso

- CASE seletor OF
  alvo1 : BEGIN
  ... instruções ...
  END;

  alvo2 : comando2;

  alvo3 : BEGIN
  ... instruções ...
  END;
  ELSE comando4;
  END;

```
Uses Crt;

  Var X : Integer;

  Begin
    Readln (X);
    Case X Of
        1 : Writeln ('Ola Mundo'); { E o valor de X for igual a 1, irá executar essa linha }
        2 : Writeln ('GNOIA');     { X = 2, essa linha será executada }
        3 : Writeln ('Software Livre'); { X = 3 - essa linha será executada }
    End;
  End.
```

Comandos de Repetição

- While (enquanto);
- Sintaxe

```
while <condição> do
 begin
   <comandos>;
   <comandos>;
 end;
```

Exemplo While

```
program  media_notas;
 var
 NOME: string;
 N1, N2, N3, MEDIA: real;
 CONT: integer;
 begin
      CONT:=0;
      while CONT<=50 do
        begin
           CONT:=CONT+1;
           read(NOME,N1,N2,N3);
           if (N1>=0) and (N2>=0) and (N3>=0) then
              begin
                    MEDIA:=(N1+N2+N3)/3;
                    writeln('O aluno de nome ',NOME,' tem a média ',MEDIA,' em suas notas ');
              end
           else
              writeln('Não são aceitas notas negativas ');
        end;
 end
```

Comandos de Repetição

- Repeat until (repita até)
- Sintaxe

repeat

<comando1>;

<comando2>;

<comando3>;

until <condição>;

Exemplo

```
program tabuada;
 uses CRT;
 var
  num:real;
  cont:integer;
 begin
  repeat
    clrscr;
    write('Digite um número para ver sua tabuada de multiplicação ');
    read(num);
    cont:=0;
      while cont<10 do  {temos aqui uma repetição dentro de outra}
         begin
           writeln(num:4:2,' x ',cont,' = ',(num*cont):6:2);
           cont:=cont+1; {a cada repetição o cont aumenta +1}
         end;
    readkey; {parada para ver o resultado até ser teclado algo}
  until num=0; {condição para parar a repetição principal}
 clrscr;
 write('Foi digitado o número 0! programa encerrado!');
 readkey;
end
```

Comandos de Repetição

- for to (para de até)
- Sintaxe

```
for contador:=ValorInicial to ValorFinal do
  begin
   comandos
  end;
```

Exemplo

```
program  media_notas;
 var
 NOME:string;
 N1,N2,N3,MEDIA:real;
 CONT:integer;
 begin
   FOR CONT:=1 to 10 do    {para cont de 1 a 10 faça}
    begin
      write('Digite o nome e as 3 notas do ',cont,'o  aluno ');
      read(NOME,N1,N2,N3);
      if (N1>=0) and (N2>=0) and (N3>=0) and (N1<=10 00) and (N2<=10 00) and (N3<=10 00)
then
        begin
          MEDIA:=(N1+N2+N3)/3;
          writeln('O aluno de nome ',NOME,' tem a média ',MEDIA,' em suas notas ');
        end
      else
        begin
          writeln('Notas invalidas!');
        end;
    end;
 end
```

Vetores

- Sintaxe

<nome array>: **array** [dimensão] **of** <tipo>;

- Declaração e exemplo:
- Var

```
    vet:array[1..6] of integer;
Begin
    vet[2]:=90;
    vet[4]:=45;
    vet[5]:=30;
end.
```

Exemplo Vetor

```
var
   v: array [1..5] of integer;
   soma,cont:integer;
Begin
     soma:=0;
     writeln('digite os 5 valores');
     for cont:=1 to 5 do
         readln(v[cont]);
     for cont:=1 to 5 do
         soma:=soma + v[cont];
     writeln('Soma=',soma);
end.
```

Matrizes

- Sintaxe:

<nome array>: **array** [dimensões] **of** <tipo>;

- Declaração e exemplo:

```
Var
    mat:array[1..5,1..3] of integer; {matriz 19x3}
Begin
    mat[1,2]:=10;
    mat[3,1]:=11;
    mat[5,3]:=12;
    mat[2,2]:=78;
end.
```

Matrizes

```
program matriz;
var
   m:array[1..2,1..3] of integer;
   soma,contl,contc:integer;
Begin
     writeln('Digite os 6 valores');
     soma:=0;
     for contl:=1 to 2 do
         for contc:=1 to 3 do
             readln(m[contl,contc]);
     for contl:=1 to 2 do
         for contc:=1 to 3 do
             soma:=soma + m[contl,contc];
     writeln('Soma=',soma);
end.
```