|  |  |
| --- | --- |
|  | Escola Secundária de Sebastião da Gama |
| Sistemas Digitais e Arquitetura de Computadores – Módulo 6 |
|  |
| Prof. Carlos Pereira |

**Operadores**

*Resumo*

***Operadores, Atribuição e Variáveis***

[**www.ticmania.net**](http://www.ticmania.net)

**Palavras Chave: Operador; Variável; Atribuir; Concatenar**

Operadores com o mesmo nível de precedência são avaliados da esquerda para a direita. As expressões dentro dos parêntesis são avaliadas primeiro, independente da ordem de precedência

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operadores aritméticos reais** | |  | **Operadores aritméticos inteiros** | | |
| adição | **+** |  | adição | **+** |  |
| subtração | **-** |  | subtração | **-** |  |
| divisão | **/** |  | divisão | **Div** |  |
| multiplicação | **\*** |  | multiplicação | **\*** | exemplo |
|  |  |  | Resto da divisão | mod | 8 mod 2=0  7 mod 2=1 |

|  |  |
| --- | --- |
| OPERADOR | DESCRIÇÃO |
| \* | Multiplicação. Ex: 5\*4 = 20 |
| + | Soma. Ex: 5+4 = 9 |
| - | Subtração. Ex: 5-4 = 1 |
| / | Divisão. Ex: 2/4 = 0.5 |
| := | Atribuição. Ex: a:=4 |
| < | Menor que. Ex: if salario < 2000 then |
| <= | Menor ou igual que. Ex: if b <= 34 then |
| <> | Diferente de. Ex: if nome <> 'MSX then |
| = | Igual, em teste lógico. Ex: if a=5 then |
| > | Maior que. repeat ... until j>15 |
| >= | Maior ou igual que. Ex: while k >= 4 do |
| and | "E" lógico. Ex: if (a>4) and (b<5) |
| div | Divisão inteira. Ex: 5 div 4 |
| in | Inclusão. Ver estrutura SET. |
| mod | Resto da divisão. 4 mod 3 |
| not | Negação. if not valido then |
| or | "OU" lógico. Ex: if (a>4) or (b<5) |
| shl | Shift left (deslocamento de bits à esquerda) |
| shr | Shift right (deslocamento de bits à direita) |
| xor | "Ou exclusivo". Ex. if a xor b then |
|  |  |
| ARITIMÉTICOS | DESCRIÇÃO |
| abs | Elimina sinal de número. Ex: abs(-5) traz 5 |
| arctan | Arco tangente. Ex: arctan(0.5) traz 0.4636 |
| cos | Cosseno em radianos. Ex: cos(1.4) 0.16997 |
| exp | Exponencial neperiano (e elevado a n). Ex: exp(4) |
| frac | Retorna a parte fracionária. Ex: frac(14.58) traz 0.58 |
| int | Retorna parte inteira de número fracionário. |
| ln | Logaritmo neperiano. Ex: ln(5) |
| sin | Seno. Ex: sin(0.5) |
| sqr | Quadrado de um número. Ex: srq(4) traz 16 |
| sqrt | Raiz quadrada. Ex: sqrt(9) traz 3 |

## Atribuição

A atribuição de valor é feita utilizando os dois pontos e o igual ":=". Os dois pontos são utilizados para diferenciar a igualdade de atribuição da igualdade do teste lógico, no qual é utilizado apenas o símbolo de igual "=".

|  |
| --- |
| Exemplo |
| var nota : integer;  begin  nota := 7;  if nota = 7 then  writeln('Aprovado no limite!');  end. |

## Variáveis

|  |  |
| --- | --- |
| Variável | Descrição |
| Boolean | Variável que admite somente 2 valores distintos: FALSO ou VERDADEIRO. |
| Byte |  |
| Char | Variável utilizada para manipular caracteres. Contém o código ASCII de uma letra.  Não serve para armazenar valores, pois este tipo de variável não permite realizar operações.  A função ord() retorna o código ASCII da variável, enquanto a função chr() retorna a letra correspondente a um código ASCII. |
| Integer | Variável utilizada para manipular números inteiros |
| Real | Variável utilizada para manipular números reais.  Principais funções do real:  trunc() - Trunca um valor real.  round() - Arredonda um valor real.  int() - Pega a parte inteira de um valor real. Retorna inteiro.  frac() - Pega a parte fracionária de um real.  abs() - Retorna valor absoluto (elimina o sinal) de um inteiro ou real.  sin(), cos() - Calcula o seno ou cosseno de um ângulo em radianos.  sqr() - Calcula o quadrado de um número, equivalente a num \* num.  sqrt() - Calcula a raiz quadrada de um número |
| String | Variável que manipula cadeia de caracteres. Utilizada para armazenar palavras. |
| Text |  |

Uma variável é um espaço de memória reservado pelo programa para armazenar um determinado tipo de dado. Uma variável é composta por:

**Um identificador** - um nome que irá identificar o dado, ou seja, irá fazer referência a esse dado na memória.

**Descrição do tipo de dado** - indica qual é o tipo de dado armazenado.

No Pascal devemos declarar uma variável antes de utilizá-la. Também devemos indicar qual é o tipo de dado que essa variável irá comportar.

A palavra reservada "var" é utilizada para indicar o escopo da declaração de variáveis.

|  |
| --- |
| Sintaxe: |
| var <identificador> : <tipo de dado>;  Exemplo: Declara uma variável com o identificador "idade" e tipo de dado inteiro.  var idade : integer; |

|  |
| --- |
| Exemplo |
| Para a declaração de mais variáveis, utiliza-se um "var" apenas. Além disso, a declaração de variáveis deverá sempre estar antes do "begin" do corpo principal, função ou procedimento correspondente. Ex:  var nome : string[40];  idade : integer;  altura : real; |

|  |  |
| --- | --- |
| **Boolean - Exemplo** | **char - Exemplo** |
| var tuga : boolean;  begin  tuga := false;  if tuga then  writeln('Portuga')  else  writeln('Estrangeiro');  end.  **Saída: 'Estrangeiro'**  **Obs: para variáveis do tipo booleano, o teste do "if":**  **if tuga then**  **equivale a:**  **if tuga = true then** | var letra1, letra2 : char;  begin  letra1 := 'A';  letra2 := #65;  writeln('A letra 1 é: ', letra1);  writeln('A letra 2 é: ', letra2);  writeln('O codigo ASCII da letra 1 é: ',ord(letra1));  letra2 := chr(65); { Também pode }  end.  **Saída:**  **A letra 1 é: A**  **A letra 2 é: A**  **O código ASCII da letra 1 é: 65** |
|  |  |
| **integer - Exemplo** | **real - Exemplo** |
| var idade : integer;  begin  idade:=20;  writeln('Ela tem ', idade, ' anos.');  writeln(' anos.');  end.  **Saída:**  **Ela tem 20 anos.** | var a : real;  begin  a := 7.6;  writeln('Valor de a: ', a);  writeln('Valor truncado de a: ', trunc(a));  writeln('Valor arredondado de a: ', round(a));  end.  **Saída:**  **Valor de a: 7.6**  **Valor truncado de a: 7**  **Valor arredondado de a: 8** |
|  |  |
| **STRING - Exemplo** | |
| Aquando a declaração, podemos indicar o tamanho da palavra. Caso o tamanho não seja declarado, será atribuído o máximo valor possível, ou seja, 256 caracteres.  As principais funções de manipulação de strings são:   |  | | --- | | Sintaxe: | | copy(string,inicio,tamanho) – Copia uma string carater de início, quantidade de caracteres  length(string) - Retorna o tamanho da string. |   var nome : string[10];  begin  nome:='Marcelo';  writeln(copy(nome,1,3)+'TIC');  end.  **Saida:**  **MarTIC**    **Concatenação** é a união de strings. No pascal utiliza-se o sinal de soma ("+").  var frase1, frase2, frase3 : string;  begin  frase1 := 'TIC';  frase2 := frase1 + ' e ';  frase3 := frase2 + frase1;  writeln(frase1);  writeln(frase2);  writeln(frase3);  end.  **Saída:**  **TIC**  **TIC e**  **TIC e TIC** | |

|  |  |  |
| --- | --- | --- |
| Converte em Código pascal os fluxogramas abaixo. | | |
|  |  |  |