UniFCV

Centro Universitário Cidade Verde

Algoritmos e Lógica de Programação

Renan Hagiwara

Expressões Lógicas

Operadores Relacionais

Operadores relacionais são utilizados para comparar valores, o resultado de uma expressão relacional é um valor booleano (TRUE ou FALSE) e estão listados abaixo:

=	Igual
<>	Diferente
>	Maior
<	Menor
>=	Maior ou Igual
<=	Menor ou Igual

Expressões Lógicas

```
01. Program exemplo_op_relacionais;
02. Var
03.     r1, r2, r3: boolean;
04. Begin
05.     r1 := 1 > 2;
06.     write(r1);
07.
08.     r2 := 3 <> 5;
09.     write(r2);
10.
11.     r3 := 10 >= 15;
12.     write(r3);
13.     readkey;
14. End.
```

Expressões Lógicas

Operadores Lógicos

São usados para representar situações lógicas que não podem ser representadas por operadores aritméticos. Também são chamados conectivos lógicos por unirem duas expressões simples numa composta.

AND, NAND, OR, XOR e NOT são os principais operadores lógicos, base para a construção de sistemas digitais e da Lógica proposicional, e também muito usado em linguagem de programação. Os operadores AND, NAND, OR e XOR são operadores binários, ou seja, necessitam de dois elementos, enquanto o NOT é unário.

Expressões Lógicas

Na computação, esses elementos são normalmente variáveis binárias, cujos possíveis valores atribuídos são 0 ou 1. Porém, a lógica empregada para essas variáveis serve também para sentenças (frases) da linguagem humana, onde se esta for verdade corresponde ao valor 1 (TRUE), e se for falsa corresponde ao valor 0 (FALSE).

Expressões Lógicas

AND

Operador lógico no qual a resposta da operação é verdade (TRUE) se ambas as variáveis de entrada forem verdade.

X1	X2	X1 AND X2
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

Expressões Lógicas

```
01. Program exemplo_and;
02. Var
03.     t1, t2, t3:boolean;
04. Begin
05.     t1 := TRUE AND TRUE;
06.     writeln('Teste 1: ', t1);
07.
08.     t2 := TRUE AND FALSE;
09.     writeln('Teste 2: ', t2);
10.
11.     t3 := (5 > 3) AND (2 < 4);
12.     writeln('Teste 3: ', t3);
13.   readkey;
14. End.
```

Expressões Lógicas

OR

Operador lógico no qual a resposta da operação é verdade (TRUE) se pelo menos uma das variáveis de entrada for verdade.

X1	X2	X1 OR X2
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

Expressões Lógicas

```
01.  Program exemplo_or;
02.  Var
03.      t1, t2, t3:boolean;
04.  Begin
05.      t1 := FALSE OR FALSE;
06.      writeln('Teste 1: ', t1);
07.
08.      t2 := TRUE OR FALSE;
09.      writeln('Teste 2: ', t2);
10.
11.      t3 := (5 < 3) OR (2 < 4);
12.      writeln('Teste 3: ', t3);
13.    readkey;
14.  End.
```

Expressões Lógicas

XOR

Operador lógico no qual a resposta da operação é verdade (TRUE) quando as variáveis assumem valores diferentes entre si.

X1	X2	X1 XOR X2
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

Expressões Lógicas

```
01.  Program exemplo_xor;
02.  Var
03.      t1, t2, t3:boolean;
04.  Begin
05.      t1 := FALSE XOR FALSE;
06.      writeln('Teste 1: ', t1);
07.
08.      t2 := TRUE XOR FALSE;
09.      writeln('Teste 2: ', t2);
10.
11.      t3 := (5 < 3) XOR (2 < 4);
12.      writeln('Teste 3: ', t3);
13.    readkey;
14.  End.
```

Expressões Lógicas

NOT

Operador lógico que representa a negação (inverso) da variável atual. Se ela for verdade, torna-se falsa, e vice-versa.

X1	NOT X1
FALSE	TRUE
TRUE	FALSE

Expressões Lógicas

```
01. Program exemplo_not;
02. Var
03.     t1, t2:boolean;
04. Begin
05.     t1 := NOT FALSE;
06.     writeln('Teste 1: ', t1);
07.
08.     t2 := NOT TRUE;
09.     writeln('Teste 2: ', t2);
10.   readkey;
11. End.
```