

## Logické hodnoty a operácie s nimi

Logické hodnoty sú len dve: pravda (true) a nepravda (false). Na uloženie logických hodnôt používame premenné typu **boolean**. Veľkosť premennej logického typu je 1B.

### Operácie nad logickým údajovým typom

- Ø Ø logická negácia: **not** – počet operandov: 1
- Ø Ø logický súčin: **and** – počet operandov: 2
- Ø Ø logický súčet: **or** – počet operandov: 2
- Ø Ø logický výhradný súčet (nonekvivalencia): **xor** – počet operandov: 2

| Hodnoty operandov |          | Výsledky operácií |               |                |              |
|-------------------|----------|-------------------|---------------|----------------|--------------|
| <i>p</i>          | <i>q</i> | <i>p and q</i>    | <i>p or q</i> | <i>p xor q</i> | <b>not p</b> |
| false             | false    | false             | false         | false          | true         |
| false             | true     | false             | true          | true           | true         |
| true              | false    | false             | true          | true           | false        |
| true              | true     | true              | true          | false          | false        |

Vyššiu prioritu ako ostatné logické operácie má operácia **not**, preto sa vždy vykonáva ako prvá. Operácie **and**, **or** a **xor** sú čo do priority rovnocenné. Ak potrebujete zmeniť prioritu vykonávania operácií, použite zátvorky!

POZOR! Turbo Pascal nedokáže načítať hodnotu premennej typu boolean. Takže v programe môžeme uložiť do premennej typu boolean hodnotu len priradením.

#### Príklad 1:

Akú pravdivostnú hodnotu budú mať logické výrazy pre a=true, b=false a c=true?

- a) a) **a and b**
- b) b) **a or b or c**
- c) c) **not b xor c**
- d) d) **a xor not c**

#### Riešenie:

- a) a) **a and b = true and false = false**
- b) b) **a or b or c = true or false or true = true or true = true**
- c) c) **not b xor c = not false xor true = true xor true = false**
- d) d) **a xor not c = true xor not true = true xor false = true**

### Príklad 2:

Akým logickým výrazom zapíšeme skutočnosť, že

- a) a) logický výraz je pravdivý vtedy, ak je hodnota logickej premennej A false, inak je nepravdivý
- b) b) logický výraz je nepravdivý vtedy, ak sú hodnoty logických premenných A, B false, inak je pravdivý
- c) c) logický výraz je pravdivý vtedy, ak sú hodnoty logických premenných A, B true, inak je nepravdivý
- d) d) logický výraz je nepravdivý vtedy, ak sú hodnoty logických premenných A, B rôzne, inak je pravdivý

### Riešenie:

Vychádzajme z tabuľky logických operácií

- a) a) ak A=false, tak logický výraz je true, ak A=true, tak logický výraz je false, to je negácia, takže hľadaný logický výraz bude not A
- b) b) ak A=false, B=false, tak logický výraz je false, ak je A=true, B=false, tak logický výraz je true, pre A=false, B=true je logický výraz true a hodnotu true bude mať aj pre A=true, B=true. Pozrime sa do tabuľky a hľadáme vhodnú operáciu. Našli sme operáciu OR. Takže výsledný logický výraz bude A or B
- c) c) ak A=true, B=true, tak logický výraz je true, ak je A=true, B=false, tak logický výraz je false, pre A=false, B=true je logický výraz false a hodnotu false bude mať aj pre A=false, B=false. Pozrime sa do tabuľky a hľadáme vhodnú operáciu. Našli sme operáciu AND. Takže výsledný logický výraz bude A and B
- d) d) ak A=true, B=false, tak logický výraz je false, ak je A=false, B=true, tak logický výraz je false, pre A=true, B=true je logický výraz true a hodnotu true bude mať aj pre A=false, B=false. Pozrime sa do tabuľky a hľadáme vhodnú operáciu. To je negácia operácie XOR. Takže výsledný logický výraz bude not (A xor B)

### Príklad 3:

Zostavte program, ktorý vypíše výsledok operácie not(true and false).

### Riešenie:

```
uses Crt;  
var a,b,h:boolean;  
begin  
  a:=true;  
  b:=false;  
  h:=not(a and b);  
  writeln('not (' ,a, ' and ' ,b, ') = ' ,h);  
  readln;  
end.
```