



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání  
pro konkurenceschopnost**

INVESTICE  
DO ROZVOJE  
VZDĚLÁVÁNÍ

**Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1**

**Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT**

**Název: Základy programování a algoritmizace úloh**

**Téma: Priorita operátorů**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_25–10**

**Anotace:** *Materiál popisuje je zaměřen na specifickou oblast související s výrazy a operátory v jazyce Pascal, kterou je priorita operátorů. Popisuje rozdělení operátorů podle priority vyhodnocování a uvádí typické chyby, které se při tvorbě výrazů objevují.*

*Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: srpen 2012.*

# 10. Priorita operátorů

- výrazy obsahující standardní operátory jazyka Pascal se vyhodnocují vždy zleva doprava s přihlédnutím k prioritě použitých operátorů
- proces vyhodnocování výrazů můžeme ovlivnit také použitím závorek ve výrazech
- rozlišujeme 4 úrovně priority

<b>priorita</b>	<b>skupina</b>	<b>operátory</b>
nejvyšší	unární	@ not
vyšší	multiplikativní	* / div mod and shl shr
nižší	aditivní	+ - or xor
nejnižší	relační	= < > <= >= <> in

# Vyhodnocování výrazů

- pokud výraz obsahuje více operátorů, ale pouze z jedné skupiny dle priority, nezáleží na tom, v jakém pořadí jsou dílčí výrazy vyhodnocovány

výraz  $a+b-c$  bude mít stejný výsledek jako  $b-c+a$

- vyhodnocovány budou takto:

$$\underline{a+b} - c$$

$$\underline{x - c}$$

z

$$\underline{b-c} + a$$

$$\underline{y + a}$$

z

- použitím závorek také nic nezměníme
- výrazy  $(a+b) - c$  i  $a + (b-c)$  budou mít stejné výsledky

$$\underline{(a+b)} - c$$

$$\underline{x - c}$$

z

$$a + \underline{(b-c)}$$

$$\underline{a + y}$$

z

# Vyhodnocování výrazů

- problém nastane ve chvíli, kdy máme v jednom výrazu operátory z více skupin

výraz  $a+b*c$  bude vyhodnocován takto:

$$a + \underline{b*c}$$

$$a + \underline{x}$$

Z

- pokud bychom chtěli nejprve vyhodnotit dílčí výraz  $a+b$ , musíme použít závorky  $(a+b)*c$
- vyhodnocení pak bude vypadat takto:

$$\underline{(a+b)} * c$$

$$\underline{x} * c$$

Z

# Časté chyby při vytváření výrazů

- nesprávným zápisem výrazů, kdy opomeneme vzít v úvahu prioritu (zvláště u méně obvyklých operátorů) mohou vzniknout dva druhy chyb:
- výraz nebude možné vyhodnotit kvůli konfliktu typů dílčích výsledků s typem informací zpracovatelných použitým operátorem (takovou chybu nalezne překladač a upozorní nás na ni)
- výraz vyhodnotit půjde, vyhodnocen ale bude jinak, než jsme si mysleli
- druhý případ je mnohem záladnější než první, neboť obdržíme výsledek, ale ten vznikne z jiného výrazu a může tak být chybný, čehož si nemusíme vůbec všimnout

# Časté chyby při vytváření výrazů

- ukázkou prvního typu chyb je výraz  $a > 2 \text{ and } b < 5$
- výraz tvořený dvěma dílčími relačními výrazy s jejichž výsledky chceme provést logický součin bude kvůli prioritám operátorů vyhodnocen takto:

$a > \underline{2 \text{ and } b} < 5$  bitový součin!

$\underline{a > x} < 5$

$\underline{y} < 5$  porovnání logické hodnoty a čísla

chyba

- správně zapsaný je tento výraz  $(a > 2) \text{ and } (b < 5)$

$\underline{(a > 2) \text{ and } (b < 5)}$  správné vyhodnocení výrazu jsme

$\underline{x \text{ and } y}$

si vynutili použitím závorek

# Časté chyby při vytváření výrazů

- druhý typ chyb reprezentuje výraz `a=1 or 4 or 6`
- záměrem tohoto výrazu bylo zjistit, zda je v proměnné `a` obsažena některá z hodnot 1, 4 nebo 6
- zápis výrazu však vychází z lidského chápání úlohy

`a=1 or 4 or 6` bitový součet!

`a= 5 or 6` bitový součet!

`a= 7`

v

- dostali jsme tak odpověď na něco jiného, než byl náš dotaz
- správný zápis vypadá takto: `(a=1) or (a=4) or (a=6)`

`(a=1) or (a=4) or (a=6)`

`x or y or z`

`w or z`

v