

# Informační a komunikační technologie

Výrazy, operátory, priority

Ing. Kovář David  
[kowy@centrum.cz](mailto:kowy@centrum.cz)



# Návrat k matematice

- výrazy

- v lidské řeči skládáme slova do vět – **výrazů**
- v počítačové řeči používáme příkazy a podmnožinu matematických vzorců = **matematické výrazy**
- skládá se z **operandů** a **operátorů**
- **operandy** jsou **literály** nebo **proměnné**

$$3.2 * x * x + (b - 4) * x + 12 = 0$$



# Operátory

## Podle počtu operandů

- 1) unární
- 2) binární
- 3) ternární

## Podle typu

- 1) přiřazení
- 2) aritmetické
- 3) relační
- 4) logické
- 5) bitové



# Aritmetické operátory

- pro vyjádření aritmetických výrazů

Operace	Operátor	Příklad
sčítání, odčítání	+, -	$x + 10$
unární minus	-	-10
násobení	*	$x * 10$
dělení	/	$x / 10$
modulo	%	$x \% 10$
operace a přiřazení	+=, -=, *=, /=, %=	$x += 10$
inkrementace	++	$x++$
dekrementace	--	$x--$

# Aritmetické operátory 2

```
int a=17
```

```
int b=2+3
```

```
int c=a/b
```

```
int d=a%b
```

```
printf("%d/%d=%d, zb.%d", a, b, c, d)
```

```
c *= b
```

```
c += d
```

kolik je v c?



# Aritmetické operátory 3

- operátory ++ a -- mají *post* a *pre* varianty

```
int x=5, y=0;
printf ("Před %d, po %d\n", x, y)
y = x++
printf ("Před %d, po %d\n", x, y)
y = x--
printf ("Před %d, po %d\n", x, y)
y = ++x
printf ("Před %d, po %d\n", x, y)
y = --x
printf ("Před %d, po %d\n", x, y)
```



# Relační operátory

- pro porovnání 2 hodnot

Operace	Operátor	Příklad
rovno	==	x == 10
nerovno (různé)	!=	x != 10
větší než	>	x > 10
menší než	<	x < 10
větší nebo rovno	>=	x >= 10
menší nebo rovno	<=	x <= 10



# Relační operátory 2

```
int i = 10  
float j = 10.0f  
println (i == j)  
println (i != j)  
println (i > j)  
println (i <= j)
```





# Logické operátory

- spojení 2 a více relací

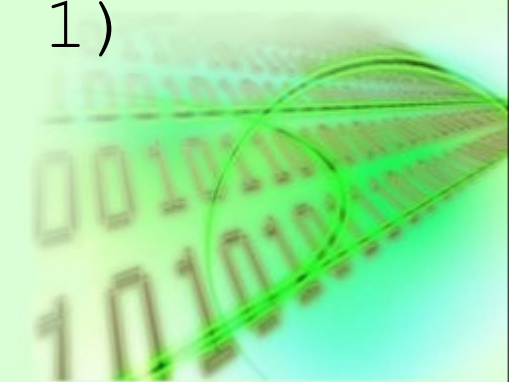
Operace	Operátor	Příklad
AND	&&	<code>a &gt; 10 &amp;&amp; a &lt; 15</code>
OR		<code>a == 10    b == 10</code>
ternární operátor	?:	<code>vek &lt; 50 ? "Mladý" : "Starý"</code>



# Logické operátory 2

```
int testy = 60, cvic = 50;
boolean prospel = testy > 50 && cvic > 50
printf ("Prospěl? %s", prospel)
printf ("Student %s",
    prospel?"Prospěl":"Neprospěl")

int znamka = 1
prospel = prospel || (++znamka == 1)
printf ("Známka: %d", znamka)
```



# Bitové operátory

AND (&)
1110
1010
1010

OR ( )
1110
1010
1110

posun vlevo (<<)
11110101
11010100

XOR (^)
1110
1010
0100

doplňěk (~)
1010
0101

posun vpravo (>>)
11110101
00111101

posun bez znaménka (>>>)
10001111
10000111



# Priority operátorů

Kolik je:

$$2 + 5 * 10$$

?



# Priority operátorů 2

Pořadí	Typ operátoru	Operátor
1	postfixové op	. (výraz) výraz++ výraz--
2	unární op	++výraz --výraz +výraz -výraz ~ !
3	typové op	new (typ)
4	multiplikativní op	* / %
5	součtové op	+ -
6	op posuvu	<< >> >>>
7	relační op	< > <= >= instanceof
8	op rovnosti	= !=
9	bitové AND	&
10	bitové XOR	^
11	bitové OR	
12	logické AND	&&
13	logické OR	
14	ternární op	?:
15	přiřazovací op	= += -= *= /= %= &= ^=

U více operátorů se stejnou prioritou,  
se vyhodnocuje zleva doprava



# Otázky a úkoly

- 1) Jaký je rozdíl mezi operátorem a operandem?
- 2) K čemu jsou vhodné závorky ve výrazu?
- 3) Co vrací operátor modulo?
- 4) Co jsou to priority operátorů?
- 5) K čemu slouží operátory = a ==?
- 6) K čemu slouží operátor || ?
- 7) Jaký je výsledek výrazu:  
 $x=11; y = 5 * (2 + 8) \% --x + 4$
- 8) Co bude v x:  
 $x=10; x *= x >= 10 ? ++x : x;$

