
LEKCE 6

Operátory

V této lekci najdete:

♦ Aritmetické operátory.....	94
♦ Porovnávací operátory.....	96
♦ Operátor řetězení.....	97
♦ Bitové logické operátory	97
♦ Další operátory	101

Operátory se v programech používají k provádění matematických operací, k porovnávání a k provádění logických operací se dvěma čísly nebo číselnými výrazy. Jednoduchými příklady operátorů jsou například znaménka plus (+) a mínus (-). S mnoha operátory jste se již v aplikaci Access jistě setkali při práci se vzorci.

Operátory mají různou prioritu a na základě své priority se v rámci jednotlivých kategorií (aritmetické, porovnávací a logické) vyhodnocují v určitém pořadí. Prioritu jednotlivých operátorů uvádí následující tabulka. Operátory jsou v tabulce řazeny podle priority odshora dolů.

Aritmetické	Porovnávací	Logické
Mocnění (^)	Je rovno (=)	Not
Negace (-)	Není rovno (<>)	And
Násobení a dělení (*, /)	Menší než (<)	Or
Celočíselné dělení (\)	Větší než (>)	Xor
Modulo (Mod)	Menší než nebo rovno (<=)	Eqv
Sčítání a odčítání (+, -)	Větší než nebo rovno (>=)	Imp
Řetězení (&)	Like / Is	

Prioritu operátorů lze ve vzorci měnit pomocí kulatých závorek.

Vzorce, které jsou vnořené nehlouběji, se vyhodnotí jako první.

Při změně priority operátorů pomocí závorek můžete snadno udělat chybu, takže je důležité, abyste přesně chápali, jak závorky s operátory fungují. Vyzkoušejte si v podrutině modulu následující příklady zdvořitého kódu:

```
MsgBox (10 + 6) / 3
```

Toto volání vrátí hodnotu 5,3333.

```
MsgBox 10 + 6 / 3
```

A toto volání hodnotu 12.

V prvním příkladu vynutily závorky vyhodnocení součtu 10 + 6 před vydělením výsledku číslem 3.

V druhém příkladu se však priorita operátorů vyhodnocuje normálně, takže se nejprve provede dělení 6/3 a poté se k výsledku přičte 10.

Aritmetické operátory

Operátory + (plus), - (mínus), * (krát) a / (děleno) se používají v aritmetických výpočtech.

Operátor *

Operátor * násobí dvě čísla. Výsledkem následující operace je číslo 18:

```
MsgBox 6 * 3
```

Násobit lze jakékoliv číselné výrazy. Datový typ výsledku odpovídá datovému typu nejpřesnějšího operandu. Výsledek může být typu **Integer** (nejméně přesný), **Long**, **Single**, **Double** a **Currency** (nejpřesnější). Další informace o datových typech naleznete v kapitole 2. Pokud jako jeden z operandů použijete hodnotu **Null**, bude výsledkem násobení hodnota **Null**.

Operátor +

Operátor + sčítá dvě čísla nebo dva výrazy. Výsledkem následujícího výpočtu je číslo 6:

```
MsgBox 4 + 2
```

Pomocí tohoto operátoru lze sčítat čísla i spojovat řetězce. Spojování řetězců (řetězení) pomocí operátoru + však může být matoucí, a proto je lepší řetězení provádět pomocí operátoru &. Příklad vlivu operátoru + na operaci řetězení naleznete na konci této části.

Sčítat lze jakékoliv číselné výrazy. Datový typ výsledku odpovídá datovému typu nejpřesnějšího operandu. Výsledek může být typu **Integer** (nejméně přesný), **Long**, **Single**, **Double** a **Currency** (nejpřesnější). Další informace o datových typech naleznete v kapitole 2. Pokud jako jeden z operandů použijete hodnotu **Null**, bude výsledkem sčítání hodnota **Null**.

Níže uvádíme některá obecná pravidla sčítání a řetězení:

- ◆ Pokud jsou oba operandy číselné, použijte operátor sčítání.
- ◆ Pokud jsou oba operandy řetězce, použijte operátor řetězení.
- ◆ Pokud je jeden z operandů číslo a druhý typu **Variant**, použijte operátor sčítání.
- ◆ Pokud je jeden z operandů řetězec a druhý je typu **Variant** (ne **Null**), použijte operátor řetězení.

Pokud je jeden z operandů číselný a druhý řetězec, dojde při pokusu o jejich sečtení k chybě **Type Mismatch** (neshoda typů):

```
MsgBox 1 + " Richard"
```

K této chybě však nedojde, použijete-li v uvedeném příkladu operátor & (řetězení):

```
MsgBox 1 & " Richard"
```

Operátor -

Operátor - odečítá jedno číslo od druhého nebo neguje číselnou hodnotu. Výsledkem následujícího výpočtu je číslo 2:

```
MsgBox 6 - 4
```

Následující volání zobrazí číslo -5:

```
MsgBox -5
```

Odčítat lze jakékoliv číselné výrazy. Datový typ výsledku odpovídá datovému typu nejpřesnějšího operandu. Výsledek může být typu **Integer** (nejméně přesný), **Long**, **Single**, **Double** a **Currency** (nejpřesnější). Další informace o datových typech naleznete v kapitole 2. Pokud jako jeden z operandů použijete hodnotu **Null**, bude výsledkem odčítání hodnota **Null**.

Operátor /

Operátor / dělí dvě čísla a vrací výsledek ve formátu čísla s plovoucí desetinnou čárkou. Výsledkem následujícího výpočtu je číslo 2:

```
MsgBox 6 / 3
```

Pokud by byl součástí výsledku zbytek, zobrazil by se ve formě desetinných míst. Datový typ výsledku odpovídá datovému typu nejpřesnějšího operandu. Výsledek může být typu **Integer** (nejméně přesný), **Long**, **Single**, **Double** a **Currency** (nejpřesnější). Další informace o datových typech naleznete v kapitole 2. Pokud jako jeden z operandů použijete hodnotu **Null**, bude výsledkem dělení hodnota **Null**.

Operátor ^

Operátor ^ mocní číslo určeným exponentem. Výsledkem následujícího výpočtu je číslo 8 (číslo 2 umocněné exponentem 3):

```
MsgBox 2 ^ 3
```

Jako operandy lze použít jakékoliv číselné výrazy.

Operátor Mod

Operátor **Mod** dělí jedno číslo druhým a vrací pouze zbytek po dělení. Výsledkem následujícího výpočtu je číslo 2, což je zbytek po dělení čísla 6 číslem 4:

```
MsgBox 6 Mod 4
```

Tento operátor se často používá, když je potřeba zjistit, zda je nějaké číslo sudé či liché. Pokud má zbytek po dělení číslem 2 hodnotu **True** (je nenulový), číslo je liché.

Porovnávací operátory

Porovnávací operátory porovnávají dva výrazy. S porovnávacími operátory jste již pracovali v kapitole 4 v rozhodovacích příkazech jazyka VBA. Následující výraz vrátí hodnotu **True**, protože číslo 3 je větší než číslo 1:

```
MsgBox 3 > 1
```

Porovnávací operátory vždy vrací Booleovskou hodnotu **True** nebo **False**. Tuto hodnotu nevrací pouze v případě, že je jeden z operandů hodnota **Null**. V takovém případě vrátí hodnotu **Null**. Následující tabulka uvádí přehled porovnávacích operátorů:

Operátor	Význam
<	Menší než
<=	Menší než nebo rovno
>	Větší než
>=	Větší než nebo rovno

Operátor	Význam
=	Rovno
<>	Není rovno

Pokud jsou oba operandy číselné, provede se číselné porovnání. Pokud jsou oba operandy řetězce, provede se porovnání řetězců. Pokud je jeden operand číselný (použitá proměnná je číselného typu a obsahuje číselnou hodnotu) a jeden řetězec (proměnná obsahující řetězec znaků), dojde k chybě Type Mismatch (nesoulad typů).

Operátor řetězení

Operátor řetězení (&) zřetězí dva operandy. Následující výraz vrátí řetězec „Richard Shepherd“:

```
MsgBox "Richard " & "Shepherd"
```

Všimněte si mezery za slovem „Richard“, která vizuálně odděluje připojovaný řetězec.

Spojovat můžete i řetězce a čísla, ale výsledná hodnota bude vždy řetězec. Následující výraz vrátí řetězec „12 Dvanáct“:

```
Msgbox 12 & " Dvanáct"
```

Ačkoliv je jazyk VBA dost chytrý na to, aby vyhodnotil, že se snažíte spojit číslo s řetězcem a převede číslo 12 na řetězec, řetězení dvou různých datových typů bez jejich převedení na stejný datový typ není, z hlediska programování, „čisté“. Následující výraz vrátí hodnotu 34, ale ve formě řetězce, a ne čísla:

```
Msgbox 3 & 4
```

Bitové logické operátory

Bitové logické operátory porovnávají dva výrazy bit po bitu. K určení výsledku používají čisté binární operace.

Operátor And

Operátor **And** vyhodnocuje, zda mají oba operandy ve výrazu hodnotu **True**. Hodnotu **True** v jazyce VBA ve skutečnosti reprezentuje hodnota **-1**. Následující výraz vrátí výsledek **False**. Výraz s operátorem **And** vrací hodnotu **True** pouze v případě, že mají hodnotu **True** oba operandy:

```
Msgbox True And False
```

Pomocí operátoru **And** lze ve výrazu také porovnávat čísla. Vyhodnocení se provádí na binárním základě. První řádek následující tabulky představuje hodnotu každého bitu čísla, zapsaného ve dvojkové soustavě, a to od bitu 7 do bitu 0. Hodnoty ve druhém a třetím řádku představují binární ekvivalenty desítkových čísel ve sloupci *n*. Hodnoty v posledním řádku představují binární a desítkový ekvivalent porovnání dvou desítkových čísel ve sloupci *n* pomocí operátoru **And**. Pro každý bitový pár se používá k určení konečného výsledku v posledním řádku operátor **And**.

128	64	32	16	8	4	2	1	N
0	1	0	1	0	1	0	0	84
1	0	0	1	0	0	0	1	145
0	0	0	1	0	0	0	0	16

V druhém, třetím a čtvrtém řádku prvního až osmého sloupce této tabulky jsou hodnoty jednoho bitu 8bitového binárního čísla. Hodnoty představované jednotlivými bity jsou zobrazeny tučně v prvním řádku. Sloupec *n* obsahuje reprezentace čísel v desítkové soustavě.

Bity jednobajtového čísla jsou očíslovány zprava doleva, počínaje bitem 0 a konče bitem 7. Všimněte si, že se hodnoty jednotlivých bitů zvyšují exponenciálně; jedná se o mocniny čísla 2. Bit 0 reprezentuje hodnota 1 a bit 7 hodnota 128. První desítkové číslo je 84, takže se používají bity 6, 4 a 2 ($64 + 16 + 4 = 84$). Druhé desítkové číslo je 145, takže se používají bity 7, 4 a 0 ($128 + 16 + 1 = 145$).

Porovnáte-li tato dvě čísla pomocí logického operátoru **And**, vrátí výraz hodnotu 16. Je tomu tak proto, že jediným bitem, který má v obou číslech hodnotu 1, je bit 4. Správnost výsledku si můžete ověřit pomocí následujícího zdrojového kódu:

```
MsgBox 84 And 145
```

Výraz vrátí číslo 16.

Tato zvláštní binární operace se obecně používá k určení bitů čísla, které mají hodnotu **True**, nebo při maskování. Při maskování se určité bity v čísle nastaví na hodnoty **True** a **False**. Proveďte se to tak, že se cílové číslo porovná s „maskovacím“ pomocí operátoru **Or** a hodnoty bitů cílového čísla se nastaví na hodnoty bitů masky. Kdybyste například potřebovali nastavit bit 7 cílového čísla na hodnotu 1, spojili byste pomocí operátoru **Or** cílové číslo s číslem 128 (hodnota reprezentovaná bitem 7) a bit 7 výsledného čísla by se nezávisle na hodnotách ostatních bitů nastavil na hodnotu **True**.

Nebo byste například mohli mít proměnnou, která používá 8 bitů k uchování určitých informací o nějaké entitě, podobně jako kdyby se jednalo o její vlastnosti – každý bit by mohl reprezentovat určité nastavení. Pokud byste chtěli zjistit, zda je bit 4, reprezentující určitou hodnotu, nastaven na hodnotu **True**, stačilo by jej pomocí operátoru **And** spojit s číslem 16, což je binární hodnota bitu 4. Pokud by byl bit nastavený na hodnotu **True**, byla by výsledkem hodnota 16. Pokud ne, výraz by vrátil hodnotu 0. Při této operaci by se vůbec nebraly v potaz hodnoty ostatních bitů.

Operátor Not

Operátor **Not** provádí logickou negaci dvou čísel nebo výrazů. Tento operátor v podstatě invertuje hodnoty bitů čísla. Pokud má bit hodnotu 0, je nastaven na hodnotu 1; pokud má hodnotu 1, je nastaven na hodnotu 0. Následující výraz vrátí hodnotu **True**:

```
MsgBox Not (2 = 3)
```

Číslo 2 se sice nerovná číslu 3 (což samo o sobě odpovídá hodnotě **False**), ale operátor **Not** poté hodnoty bitů obrátí a vrátí **True**.

Operátor Or

Operátor **Or** vyhodnocuje, zda mají oba operandy hodnotu **True** (nenulovou hodnotu), nebo jeden z operandů hodnotu **True** a druhý **False**. Následující výraz vrátí hodnotu **True**, protože jeden z operandů má hodnotu **True** (**True** a **False** jsou vestavěné konstanty jazyka VBA):

```
MsgBox True Or False
```

Následující výraz vrátí **False**, protože ani jeden z operandů ve výrazu nemá hodnotu **True**:

```
MsgBox False Or False
```

Operátor **Or** využívá při vyhodnocení výrazu binární aritmetiku – výsledkem výrazu **1 Or 1** je hodnota **1**, výsledkem výrazu **1 Or 0** je hodnota **1**, výsledkem výrazu **0 Or 1** je hodnota **1** a výsledkem výrazu **0 Or 0** je **0**.

První řádek následující tabulky reprezentuje hodnotu každého bitu binárního čísla od bitu 7 do bitu 0. Hodnoty ve druhém a třetím řádku reprezentují binární ekvivalenty desítkových čísel ve sloupci *n*. Hodnoty v posledním řádku reprezentují binární a desítkový ekvivalent porovnání dvou desítkových čísel ve sloupci *n* pomocí operátoru **Or**. Pro každý bitový pár se používá k určení konečného výsledku v posledním řádku operátor **Or**.

128	64	32	16	8	4	2	1	<i>n</i>
0	1	0	1	0	1	0	0	84
1	0	0	1	0	0	0	1	145
1	1	0	1	0	1	0	1	213

V druhém, třetím a čtvrtém řádku prvního až osmého sloupce této tabulky jsou hodnoty jednoho bitu 8bitového binárního čísla. Hodnoty reprezentované jednotlivými bity jsou zobrazeny tučně v prvním řádku. Sloupec *n* obsahuje reprezentace čísel v desítkové soustavě.

Bity jednobajtového čísla jsou očíslovány zprava doleva, počínaje bitem 0 a konče bitem 7. Všimněte si, že se hodnoty jednotlivých bitů zvyšují exponenciálně; jedná se o mocniny čísla 2. Bit 0 reprezentuje hodnota 1 a bit 7 hodnota 128. První desítkové číslo je 84, takže se používají bity 6, 4 a 2 ($64 + 16 + 4 = 84$). Druhé desítkové číslo je 145, takže se používají bity 7, 4 a 0 ($128 + 16 + 1 = 145$).

Porovnáte-li tato dvě čísla pomocí logického operátoru **Or**, výraz vrátí hodnotu 213 ($128 + 64 + 16 + 4 + 1$).

Správnost výsledku si můžete ověřit pomocí následujícího zdrojového kódu:

```
MsgBox 84 Or 145
```

Tento výraz vrátí hodnotu 213.

Operátor **Or** se často používá k maskování při práci s grafikou a také při kombinování dvou parametrů. V kapitole 5 jste při volání funkce okna se zprávou používali parametry **vbExclamation** a **vbYesNo** k zobrazení správné ikony a správných tlačítek. Pokud byste spojili parametry **vbExcla-**

mation a **vbYesNo** pomocí jednoduchého operátoru **+**, měla by požadovaná logická značka chybnou hodnotu. Správného výsledku lze v tomto případě dosáhnout pouze pomocí parametru **Or**.

Příklad použití operátoru **Or** s podmínkovými příkazy **If** jste také viděli v kapitole 4:

```
If x = 1 Or y = 1 Then
```

Operátor Xor

Operátor **Xor** má velmi podobnou funkci jako operátor **Or** až na to, že pokud mají oba operandy hodnotu **True**, vrátí hodnotu **False**. Hodnotu **True** vrátí pouze v případě, že má jeden z operandů hodnotu **True** a druhý **False**. Hodnotu **True** tedy vrací výrazy **False Xor True** a **True Xor False**. Operátor **Xor** zastává funkci vylučovacího operátoru **Or**.

Následující výraz vrátí hodnotu **True**:

```
MsgBox True Xor False
```

Tento výraz vrátí hodnotu **False**:

```
MsgBox True Xor True
```

Operátor **Xor** využívá při vyhodnocení výrazu binární aritmetiku – výsledkem výrazu **1 Xor 1** je hodnota **0**, výsledkem výrazu **1 Xor 0** je hodnota **1**, výsledkem výrazu **0 Xor 1** je hodnota **1** a výsledkem výrazu **0 Xor 0** je hodnota **0**.

První řádek následující tabulky reprezentuje hodnotu každého bitu binárního čísla od bitu 7 do bitu 0. Hodnoty ve druhém a třetím řádku reprezentují binární ekvivalenty desítkových čísel ve sloupci *n*. Hodnoty v posledním řádku reprezentují binární a desítkový ekvivalent porovnání dvou desítkových čísel ve sloupci *n* pomocí operátoru **Xor**. Pro každý bitový pár se používá k určení konečného výsledku v posledním řádku operátor **Xor**.

128	64	32	16	8	4	2	1	<i>n</i>
0	1	0	1	0	1	0	0	84
1	0	0	1	0	0	0	1	145
1	1	0	0	0	1	0	1	197

V druhém, třetím a čtvrtém řádku prvního až osmého sloupce této tabulky jsou hodnoty jednoho bitu 8bitového binárního čísla. Hodnoty reprezentované jednotlivými bity jsou zobrazeny tučně v prvním řádku. Sloupec *n* obsahuje reprezentace čísel v desítkové soustavě.

Bity jednobajtového čísla jsou očíslovány zprava doleva, počínaje bitem 0 a konče bitem 7. Všimněte si, že se hodnoty jednotlivých bitů zvyšují exponenciálně; jedná se o mocniny čísla 2. Bit 0 reprezentuje hodnota 1 a bit 7 hodnota 128. První desítkové číslo je 84, takže se používají bity 6, 4 a 2 ($64 + 16 + 4 = 84$). Druhé desítkové číslo je 145, takže se používají bity 7, 4 a 0 ($128 + 16 + 1 = 145$).

Porovnáte-li tato dvě čísla pomocí logického operátoru **Xor**, výraz vrátí hodnotu 197 ($128 + 64 + 4 + 1$).

Správnost výsledku si můžete ověřit pomocí následujícího zdrojového kódu:

```
MsgBox 84 Xor 145
```

Tento výraz vrátí hodnotu **197**. Tento výsledek má jednu zajímavou vlastnost: porovnáte-li jej pomocí operátoru **Xor** s jedním z použitých čísel, získáte druhé z čísel použitých ve výrazu. Následující výraz vrátí hodnotu **145**:

```
MsgBox 84 Xor 197
```

A tento výraz vrátí hodnotu **84**:

```
MsgBox 145 Xor 197
```

Tento operátor se často používá v jednoduchých šifrovacích rutinách spolu s řetězcem náhodných znaků. Řetězec, který se má zašifrovat, se znak po znaku spojí s řetězcem náhodných znaků a výsledkem je řetězec, který vypadá zcela náhodně a který zdánlivě není tvořen podle žádného vzoru (hackeři se při nabourávání zdrojového kódu zajímají především o vzory).

Chcete-li řetězec dešifrovat, stačí jej opět pomocí operátoru **Xor** znak po znaku spojit s původním náhodným řetězcem.

Další operátory

Kromě výše uvedených operátorů jazyk VBA pro aplikaci Access nabízí i množství dalších operátorů.

Operátor Is

Operátor **Is** porovná dvě proměnné obsahující odkaz na nějaký objekt a určí, zda mají stejný obsah. Následující výraz vrátí hodnotu **True**, protože oba výrazy odkazují na stejný objekt (list1):

```
MsgBox Worksheets(1) Is Worksheets(1)
```

Následující výraz vrátí hodnotu **False**, protože výrazy neodkazují na stejný objekt:

```
MsgBox Worksheets(1) Is Worksheets(2)
```

V tomto případě neodkazují výrazy na stejný list, protože má jiný název. Mezi oběma listy mohou být i jiné rozdíly, ale pokud jsou oba listy úplně nové, liší se pouze názvem.

Operátor Like

Operátor **Like** porovnává dva řetězcové výrazy a určuje, zda jsou si navzájem podobné. Řetězce mohou mít stejné znaky nebo je mohou tvořit pouze velká či malá písmena.

```
Option Compare Text
Sub test()
    MsgBox "RICHARD" Like "richard"
End Sub
```

Pokud příkazu **Option Compare** v sekci deklarací předáte parametr **Text**, vrátí výše uvedený příklad hodnotu **True**. Pokud příkazu předáte parametr **Binary**, vrátí hodnotu **False**. Jedná se

o podobnou funkcionalitu jako v případě parametru **Compare**, který jste používali ve funkci **Instr** v kapitole 5 – uvedeným způsobem v podstatě určíte, zda se při porovnání řetězců bude brát ohled na velikost písmen.

Operátoru také můžete předávat zástupné znaky. Zástupný znak **?** označuje jediný znak a zástupný znak ***** označuje řetězec znaků. Zástupné znaky zde fungují přesně stejně, jako když je používáte k vyhledávání souborů. Zástupné znaky, které můžete používat s operátorem **Like**, uvádí následující tabulka:

Zástupný znak	Význam
?	Jakýkoliv samostatný znak
*	Nula a více znaků
#	Jakákoliv samostatná číslice (0-9)
[seznam znaků]	Jakýkoliv samostatný znak uvedený v seznamu znaků
[!seznam znaků]	Jakýkoliv samostatný znak, který není uvedený v seznamu znaků

Oba následující výrazy vrátí hodnotu **True**:

```
MsgBox "RICHARD" Like "ri?hard"
```

```
MsgBox "RICHARD" Like "ric*"
```