



Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Programování - Programování C#

Ivan Pomykacz



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

Programování

Programování C#

Obsah

- Logické operátory

Logické operátory

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_23_PRG_456		
Název tématické oblasti (sady)	Programování		
Název materiálu	Logické operátory		
Anotace	Text shrnuje poznatky o logických operátorech. Přidává operátor logické negace. Dále předkládá cvičení na procvičení nasazení logických operátorů.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Rozhodne o použití logického operátoru. Používá logické operátory v kódu.		
Klíčová slova	logický součet, logický součin, logická negace, výraz, operátor		
Druh výukového zdroje	Pracovní list	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	21.10.2013	Celková velikost	

And, Or, Not

Obsah

- Shrnutí
- Pravdivostní tabulky
 - Logický součin &&
 - Logický součet ||
 - Logická negace !
- Cvičení
 - Řešení

Shrnutí

Logické operátory umožňují tvořit komplexnější podmínky, které pak dokážou vlastně jednodušeji popsat daný problém. O čemž jste se mohli přesvědčit v **A zároveň** a **A nebo**.

Pravdivostní tabulky

V souvislosti s logickými operátory se uvádějí tzv. pravdivostní tabulky, ve kterých vystupují logický součin, součet a logická negace.

V tabulkách níže se objevuje `x`, za které lze dosadit libovolný výraz, jehož výsledkem je buď `true` nebo `false`. V tabulce jsou pak uvedeny všechny kombinace, které by mohly nastat včetně výsledku `y`.

Logický součin `&&`

Provedeme-li se všemi `x` logický součin, pak výsledkem je `y`.

AND - <code>&&</code>		
<code>x₁</code>	<code>x₂</code>	<code>y</code>
<code>false</code>	<code>false</code>	<code>false</code>
<code>false</code>	<code>true</code>	<code>false</code>
<code>true</code>	<code>false</code>	<code>false</code>
<code>true</code>	<code>true</code>	<code>true</code>

U logického součinu tedy platí, že **právě všechna** `x` musí být `true`, aby i výsledek byl `true`.

V příkladu jsou pouze `x1 .. x2`, nicméně tento počet není omezen.

Logický součet `||`

Provedeme-li se všemi `x` logický součet, pak výsledkem je `y`.

OR - <code> </code>		
<code>x₁</code>	<code>x₂</code>	<code>y</code>
<code>false</code>	<code>false</code>	<code>false</code>
<code>false</code>	<code>true</code>	<code>true</code>
<code>true</code>	<code>false</code>	<code>true</code>
<code>true</code>	<code>true</code>	<code>true</code>

U logického součtu platí, že stačí aby **alespoň jedno** `x` bylo `true`, a pak i výsledek je `true`.

Logická negace `!`

Logická negace tzv. neguje (převrátí) pravdivostní hodnotu na její opak. Tedy z `true` udělá `false`, a obráceně z `false` bude `true`.

NOT - <code>!</code>	
<code>x</code>	<code>y</code>
<code>false</code>	<code>true</code>
<code>true</code>	<code>false</code>

Oproti logickému součtu nebo součinu má negace právě jeden argument (max jedno `x`). Pokud ovšem použijeme závorky, můžeme negovat výsledek takové závorky, např. `!(true || false)`. Výsledkem bude vždy `false`, protože závorka vždy vrátí `true`.

Cvičení

Napište program, který bude simulovat "logickou kalkulačku".

Nechť uživatel zadá po sobě dvě čísla 0 nebo 1.

Program pak vyhodnotí logický součin, součet a negaci těchto čísel.

Řešení

```
Console.WriteLine ("Program: and-or-not v1.0");

string a;
string b;

Console.Write("Zadej log. 1 nebo 0: ");
a = Console.ReadLine();
Console.Write("Zadej ještě jednu log. 1 nebo 0: ");
b = Console.ReadLine();

Console.WriteLine("Logický součin - &&");
if (a == "1" && b == "1") {
    Console.WriteLine("{0} && {1} = 1", a, b);
}
else {
    Console.WriteLine("{0} && {1} = 0", a, b);
}

Console.WriteLine("Logický součet - ||");
if (a == "0" && b == "0") {
    Console.WriteLine("{0} || {1} = 0", a, b);
}
else {
    Console.WriteLine("{0} || {1} = 1", a, b);
}

Console.WriteLine("Logická negace - !");
if (a == "1") {
    Console.WriteLine("{0} ! 0", a);
}
else {
    Console.WriteLine("{0} ! 1", a);
}
if (b == "1") {
    Console.WriteLine("{0} ! 0", b);
}
```



```
else {  
    Console.WriteLine("{0} ! 1", b);  
}
```

Cvičení IX

Obsah

- Zadání
 - Postup
 - Verze 1.1
- Řešení

Zadání

Vytvořte aplikaci, která bude vyžadovat přihlášení uživatele. *Náš program bude značně omezený, ale jen do té doby, než se seznámíme s dalšími metodami a technikami programování.*

Postup

- Program vyzve uživatele k zadání uživatelského jména a hesla. Použijte metodu `Console.ReadLine()` pro získání uvedených dat.
- Proveďte ověření totožnosti uživatele. V naší aplikaci bude veden pouze jeden uživatelský účet. Pro úspěšné přihlášení je potřeba aby přihlašovací jméno a heslo byly stejné jako ty, co jsou uloženy v programu. Můžete použít logický součin k vyřešení autentizace.
- Vypište informaci o úspěchu či neúspěchu přihlášení. Pozor! Z bezpečnostního hlediska byste neměli uživatele informovat o tom, zda-li zadal chybné jméno nebo heslo. Útočník zpravidla nezná ani jméno ani heslo. A pokud mu napovíte tím, že uhádl heslo chybovou hláškou "Špatně zadané heslo", je na půl cesty k tomu, aby získal i heslo.

Verze 1.1

Schovíme heslo před zraky náhodných čumilů.

Pomocí vlastností `Console.ForegroundColor` a `Console.BackgroundColor` můžeme dočasně dosadit barvu pozadí do barvy popřeni a tím vlastně zneviditelnit psaný text.

Deklarujeme novou proměnnou `barva` typu `ConsoleColor`, do které uložíme aktuální barvu popřeni (písma).

```
ConsoleColor barva;  
barva = Console.ForegroundColor;
```

Následně můžeme změnit barvu popřeni na barvu pozadí.

```
Console.ForegroundColor = Console.BackgroundColor;
```

Po zadání hesla to ovšem musíme vrátit zpět :)

```
Console.ForegroundColor = barva;
```

Řešení

```
Console.WriteLine("Program: Login v1.1\n");

string secretName = "root";
string secretPass = "toor";

string loginName;
string loginPass;

Console.WriteLine("Přihlášení");
Console.Write("login: ");
loginName = Console.ReadLine();

// příprava na změnu barvy
ConsoleColor barva;
barva = Console.ForegroundColor;

Console.Write("heslo: ");
// záměna barvy
Console.ForegroundColor = Console.BackgroundColor;
// načtení hesla
loginPass = Console.ReadLine();
// barva zpět
Console.ForegroundColor = barva;

// autentizace
if (secretName == loginName && secretPass == loginPass) {
    Console.WriteLine("\nPřihlášení proběhlo úspěšně");
}
else {
    Console.WriteLine("\nChybně zadané jméno nebo heslo!");
}
```

Mini příklady III

Obsah

- Úloha 1 - Hádankář
- Úloha 2 - Mysli si číslo

Úloha 1 - Hádankář

Program nabídne uživateli hádanku.

Pokud uživatel odpoví správně, položí program další hádanku. Celkem budou v programu 3 hádanky.

Pokud uživatel zodpoví všechny hádanky správně, program vypíše informaci o úspěšném řešení.

Pokud uživatel odpoví špatně na alespoň jednu hádanku, tak program okamžitě skončí.

Program by měl tolerovat, zda-li bude odpověď napsána různě velkými písmeny. Případně by mu nemělo vadit, zda-li před nebo za slovem bude mezera.

[demo](#)

Úloha 2 - Mysli si číslo

Program vyzve uživatele, aby si myslel číslo od 1-5. Následně se program bude ptát uživatele na otázky, pomocí kterých zjistí, jaké si uživatel myslel číslo. Otázky budou typu: Je tvé číslo větší nebo menší jak ...

[demo](#)
