



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

- A nebo

A nebo

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_455		
Název tématické oblasti (sady)	Programování		
Název materiálu	A nebo		
Anotace	Text představuje logický operátor součtu. Na příkladu demonstruje jeho nasazení, výsledkem čehož je přehlednější a čitelnější kód. Jsou ukázána dvě řešení úlohy – pomocí dosavadních znalostí, jejichž výsledkem je neefektivní a nepřehledný kód. Druhé řešení používá logický součet.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Přečte výraz poskládaný z logických součtů. Chápe významu logický součet - nebo.		
Klíčová slova	logický součet		
Druh výukového zdroje	Výklad	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	18.10.2013	Celková velikost	

Obsah

- [Shrnutí](#)
- [Dilema](#)
 - [Implementace problému](#)
 - [Vylepšení](#)
- [A nebo - |||](#)

Shrnutí

Podmínky je možné spojovat (nikoli řetězit jako v případě `if-else-if`) pomocí logických operátorů AND a OR. To ve výsledku usnadňuje i zpřehledňuje zápis kódu.

V praxi je možné se setkat ještě dalším typem spojení podmínek. Totiž chceme-li, aby **alespoň jedna** z podmínek byla splněna (nikoli všechny současně).

Dilema

Ocitli jsme se na klinice v městské nemocnici. Stojíme v chodbě, kde na jejím konci jsou dvoje dveře. Jedny dveře vedou doprava a druhé doleva.

Kterými dveřmi se vydáte?

Implementace problému

Rozepíšme program, který alespoň trošičku bude simulovat *old school* textovou hru.

```
Console.WriteLine("Nacházíš se v nemocnici. Stojíš v chodbě na jejímž konci jsou dvoje dveře.");
```

```
Console.WriteLine("Chceš se vydat dveřmi vlevo nebo vpravo? ");
```

```
string volba = Console.ReadLine();
```

Necháme uživatele aby zapsal svou odpověď z klávesnice `Console.ReadLine()`. A nyní se rozhodneme, co se bude dít dál.

```
if (volba == "vlevo") {
    Console.WriteLine("Vstupuješ dveřmi nalevo do obrovské haly.");
}
else if (volba == "vpravo") {
    Console.WriteLine("Dveřmi vpravo sestupuješ po táhlém schodišti někam dolů do sklepení.");
}
else {
    Console.WriteLine("Neznámá volba ..");
}
```

Vylepšení

Co když bychom chtěli uživateli ulehčit psaní? To aby nemusel pracně vypisovat `"vlevo"`, resp. `"vpravo"`. Mohli bychom použít třeba anglická počáteční písmena `"l"` -> left a `"r"` -> right.

Chceme ale zachovat i původní chování - uživatel může uvést celou odpověď, ale zároveň může zadat jako odpověď jen jedno písmenko.

```
if (volba == "vlevo") {
    Console.WriteLine("Vstupuješ dveřmi nalevo do obrovské haly.");
}
else if (volba == "l") {
    Console.WriteLine("Vstupuješ dveřmi nalevo do obrovské haly.");
}
else if (volba == "vpravo") {
    Console.WriteLine("Dveřmi vpravo sestupuješ po táhlém schodišti někam dolů do sklepení.");
}
else if (volba == "r") {
    Console.WriteLine("Dveřmi vpravo sestupuješ po táhlém schodišti někam
```

```
dolů do sklepení.");  
}  
else {  
    Console.WriteLine("Neznámá volba ..");  
}
```

Ano, nezbude nám nic jiného, než připsat další if-else do stáda. To by ale tak nevadilo jako spíš to, že musíme některé části kódu celé zopakovat. V programování však opakování kódu vede k záhubě programátora. Toto řešení je naprosto nepřijatelné - a to jak záhuba, tak opakování se.

A nebo - ||

Pokud máme více podmínek, u kterých nám stačí alespoň u jedné aby výsledek byl `true`, jako u výše probíraného příkladu, totiž že:

- uživatel napsal "vlevo" ... a nebo
- uživatel napsal "l"

případně

- uživatel napsal "vpravo" ... a nebo
- uživatel napsal "r"

Pak můžeme tyto podmínky spojit do jediného výrazu pomocí tzv. logického součtu OR - ||

```
if (volba == "vlevo" || volba == "l") {  
    Console.WriteLine("Vstupuješ dveřmi nalevo do obrovské haly.");  
}  
else if (volba == "vpravo" || volba == "r") {  
    Console.WriteLine("Dveřmi vpravo sestupuješ po táhlém schodišti někam  
dolů do sklepení.");  
}  
else {  
    Console.WriteLine("Neznámá volba ..");  
}
```

Výsledek je, že pokud řetězec `volba == "vpravo"` **nebo** `||` řetězec `volba == "r"`, pak je podmínka splněna.

Nebo jestliže `else if`, pokud řetězec `volba == "vlevo"` **nebo** `||` řetězec `volba == "l"`, pak je podmínka splněna.