



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 zároveň

A zároveň

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_454		
Název tématické oblasti (sady)	Programování		
Název materiálu	A zároveň		
Anotace	Text představuje logický operátor součinu. 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 součin.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Přečte výraz poskládaný z logických součinů.		
Klíčová slova	logický součin		
Druh výukového zdroje	Výklad	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	11.10.2013	Celková velikost	

Obsah

- [Shrnutí](#)
- [Dilema](#)
 - [Implementace problému](#)
 - [Vylepšení](#)
- [A zároveň - &&](#)

Shrnutí

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

Dilema

Žena povídá svému muži, který se živí programováním: „Běž koupit do obchodu 6 rohlíků, a když budou mít vejce, kup jich 20“.

Zřejmě každý správný programátor z výše uvedeného vyvodí, že když budou mít v obchodě alespoň jedno vejce, měl by koupit 20 rohlíků.

Implementace problému

Provedme deklaraci proměnných, které budou simulovat zboží na krámě. Zrovna tyto proměnné nastavme na kladnou hodnotu, která udává množství daného zboží.

```
int rohliky = 16;
int vejce = 21;

Console.WriteLine(" zboží   | množství ");
Console.WriteLine("-----");
Console.WriteLine("rohlíky   |      {0} ks", rohliky);
Console.WriteLine("vejce     |      {0} ks", vejce);
```

Nyní napíšeme podmínku, která podpoří programátora a pomůže mu se rozhodnout, čeho má kolik koupit.

```
if (vejce > 0) {
    Console.WriteLine("Kup 20 rohlíků.");
    rohliky -= 20;
}
else {
    Console.WriteLine("Kup 6 rohlíků.");
    rohliky -= 6;
}

Console.WriteLine("Zůstatek rohlíků: {0}", rohliky);
```

Podmínka hovoří jasně. Budou-li mít alespoň jedno vejce, kupuji 20 rohlíků. V opačném případě jich беру jen 6.

Na konci je vypsáno, kolik rohlíků zbylo. Patrně jste si všimli zápisu `rohliky -= 20`. Jde o zkrácený zápis `rohliky = rohliky - 20`.

Vylepšení

Co když ale nebudou mít skladem všech 20 rohlíků? Nebo dokonce ani těch 6?

To jsou stavy, které mohou nastat a jen by zbytečně našemu programátorovi popletly hlavu. Proto si pomocí podmínek pohlídneme, že na krámě mají dostatečné množství rohlíků.

```
if (vejce > 0) {
    if (rohliky >= 20) {
        Console.WriteLine("Kup 20 rohlíků.");
        rohliky -= 20;
    }
    else {
        Console.WriteLine("Nemají dost rohlíků.");
    }
}
```

```
}  
else {  
    if (rohliky >= 6) {  
        Console.WriteLine("Kup 6 rohlíků.");  
        rohliky -= 20;  
    }  
    else {  
        Console.WriteLine("Nemají dost rohlíků.");  
    }  
}
```

Patrně jste si všimli, že jsme "opět" museli přizvat stádo if-ů, abychom mohli problém dostatečně ošetřit. Navíc se nám v kódu opakuje ta samá hláška, totiž že není dostatek rohlíků.

A zároveň - &&

Pokud máme více podmínek, které musí být splněny zároveň (současně), jako u výše probíraného příkladu, totiž že:

- mají-li vejce ... a
- je dost rohlíků

případně

- vejce nemají ... a
- je dost rohlíků

Pak můžeme tyto podmínky spojit do jediného výrazu pomocí tzv. logického součinu AND - &&.

```
if (vejce > 0 && rohliky >= 20) {  
    Console.WriteLine("Kup 20 rohlíků.");  
    rohliky -= 20;  
}  
else if (vejce == 0 && rohliky >= 6) {  
    Console.WriteLine("Kup 6 rohlíků.");  
    rohliky -= 6;  
}  
else {  
    Console.WriteLine("Nemají dost rohlíků.");  
}
```

Výsledek je, že pokud mají alespoň jedno vejce `vejce > 0`, a zároveň `&&` mají dostatek rohlíků `rohliku >= 20`, pak je podmínka splněna, a je tedy možné koupit 20 rohlíků.

Nebo jestliže `else if`, pokud nemají vejce `vejce == 0`, a zároveň mají dostatek rohlíků `rohliku >= 6`, pak je podmínka splněna, a je tedy možné koupit 6 rohlíků.