

# Zpravodaj moderní Programování 2/2014: Jak správně počítat faktury (a jiné finanční výkazy)

*Obtížnost: začátečníci*

Dnes se ve Zpravodaji budu věnovat jednomu detailu, se kterým se setkáte při práci s čísly a nejspíš při výpočtu nějakých faktur či jiných finančních výkazů.

## Zadání

Problém si nastíníme na aplikaci s následujícím uživatelským rozhraním:

Aplikace ilustruje jádro výpočetního problému, se kterým se setkáte při výpočtu položky na faktuře. Uživatel zadá cenu zboží a procentní slevu, program vypočte cenu po slevě a cenu za deset kusů.

## První řešení

Úloha není obtížná, kód následuje:

```
private void tlačítkoŠpatněDouble_Click(object sender, EventArgs e)
{
    // Vstupy
    double základníCena = Convert.ToDouble(poleZákladníCenaZaKus.Text);
    double slevaProcent = Convert.ToDouble(poleSlevaProcent.Text);

    // Výpočet
    double cenaZaKus = základníCena * (1 - slevaProcent / 100d);
    double cenaZa10Kusů = 10 * cenaZaKus;

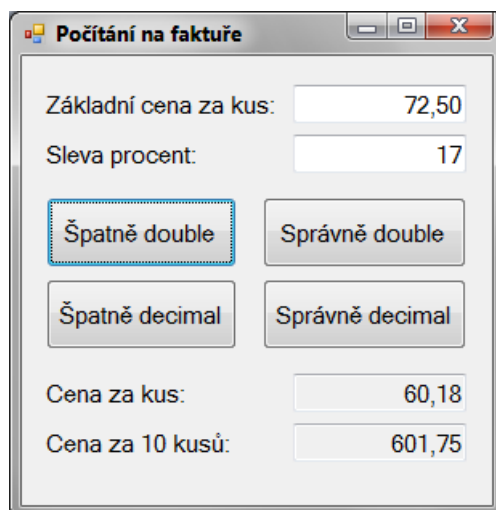
    // Výstupy
    poleCenaZaKus.Text = cenaZaKus.ToString("N2");
    poleCenaZa10Kusů.Text = cenaZa10Kusů.ToString("N2");
}
```

Na první pohled tam asi není co řešit, zarazí vás možná jen znak `d` za tou stovkou. Jde o explicitní zápis, že se jedná o hodnotu typu `double`. Zde bychom ji

uvádět nemuseli, se stovkou v `intu` by to fungovalo stejně, jelikož `slevaProcent` je `double`, takže si nenaběhneme s celočíselným dělením. Vzhledem k dalšímu pokračování programu jsem ji tam ale dal.

## Problém

Kde je tedy jaký problém? Problém je vidět až na pečlivý druhý, nebo spíš třetí pohled. **Pokud základní cena nebude v celých korunách nebo sleva nebude v celých procentech**, nebude už většinou jedno procento základní ceny v celých haléřích a výsledky se mohou podobat těmto:



Pokud na fakturu vytisknete oba výsledky, může se zákazník oprávněně divit, jaktože cena za deset kusů není desetnásobkem ceny za kus. Výpočet ceny za kus jednoduše vyšel na více než dvě desetinná místa a tato místa po vynásobení (v našem případě deseti) vylezla.

Stejnému problému budete čelit kdykoli, když peníze nebudete pouze sčítat a odčítat. Problém je trochu zálužný v tom, že na první pohled si jej člověk nemusí všimnout. Vzniká jen někdy, navíc jen málokdy násobíte pěkně kulatou desítkou a i při kontrole kalkulačkou si to nemusíte uvědomit.

Řeknete si drobnost? Jistě, ale účetnictví musí sedět na haléře, o tom již byla dříve řeč. Když nesedí, objevuje se nedůvěra.

## Typ decimal?

Říkáte si, jestli by nepomohl typ `decimal`? Tomu jsem se věnoval ve Zpravodajích 5 a 6/2013, kde jsme si říkali, že je určena na finanční výpočty. Odpovědí je ne, typ `decimal` řeší **jiný problém**. Pokud nevěříte, můžete si to přeprogramovat do `decimalu` (kód na další straně).

Stejně výsledky. **Řešením je**, jak již teď zřejmě tušíte, první výsledek (cenu za kus) zaokrouhlit na haléře a **v dalším výpočtu pracovat s touto zaokrouhlenou hodnotou**. Tak byste měli postupovat vždycky. Pokud uživateli předložíte nějaký výsledek a pracujete s ním dál, měli byste dílčí výsledek zaokrouhlit, aby si uživatel vaše výpočty mohl kdykoli ověřit. Jakmile do sestavy vypíšete 60,18, je třeba dále počítat s hodnotou 60,18 a ne s nějakými tajnými dalšími ciframi.

```
private void tlačítkoŠpatněDecimal_Click(object sender, EventArgs e)
{
    // Vstupy
    decimal základníCena = Convert.ToDecimal(poleZákladníCenaZaKus.Text);
    decimal slevaProcent = Convert.ToDecimal(poleSlevaProcent.Text);
    // Výpočet
    decimal cenaZaKus = základníCena * (1 - slevaProcent / 100m);
    decimal cenaZa10Kusů = 10 * cenaZaKus;
    // Výstupy
    poleCenaZaKus.Text = cenaZaKus.ToString("N2");
    poleCenaZa10Kusů.Text = cenaZa10Kusů.ToString("N2");
}
```

**Poznámka (jeden detail):** Za stovkou vidíte m - označení čísla typu decimal. Opět by zde mohla být stovka bez ničeho , tj. v intu, nemohla by tam ale být stovka v double - opět viz Zpravodaj 5/2013, kde jsme si ukazovali, že decimal se s doublem „nesnese“. Chtěl jsem na toto ještě jednou upozornit.

### Správné řešení

Správné řešení je stejné pro double i pro decimal, za výpočet hodnoty cenaZaKus se vsune jediný řádek, který provede zaokrouhlení na haléře **a uložení do proměnné mezivýsledku**:

```
cenaZaKus = Math.Round(cenaZaKus, 2);
```

### Další využití

Řešení diskutované otázky můžeme aplikovat také na průzkum sčítání v typu double, který jsme prováděli ve Zpravodaji 6/2013. Za výpočet

```
součetDouble += částkaDouble;
```

vložte jediný řádek:

```
součetDouble = Math.Round(součetDouble, 2);
```

Program vyzkoušejte pro různé hodnoty. Uvidíte, že součet z tabulky již bude pokaždé 14! **Tímto způsobem lze tedy odstranit kumulování zaokrouhlovací chyby!** Jednoduše součet pokaždé opravíte, aby byl stále přesný na haléře a neměl žádné tajné cifry, které pak v haléře přerostou.

Pokud by to zpomalovalo, lze tuto „čisticí“ operaci provést třeba po sto hodnotách, efekt bude téměř stejný.

### Závěr

Cílem tohoto Zpravodaje bylo upozornit na problém skrytých cifer vzniklých při násobení, dělení, anebo při ještě komplexnějších výpočtech. Podrobně jsme si rozebrali řešení, abyste při finančních výpočtech měli jistotu, že vás počítač nezaskočí špatným počítáním, a abyste viděli, co si můžete dovolit.

*Radek Vystavěl, 25. února 2014*

*Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu [www.moderniProgramovani.cz](http://www.moderniProgramovani.cz).*