

Zpravodaj moderní Programování 6/2013: Potřebujeme typ decimal? (pokračování)

Obtížnost: středně pokročilí

V minulém čísle Zpravodaje jsem se zabýval typem `decimal`. Připravili jsme si program, pomocí něžž můžeme rozdíl mezi `doublem` a `decimalem` blíže prozkoumat. Pust'me se do toho.

Sólo hodnoty

Zadáním nulového exponentu pro počet řekneme, že chceme „sečíst“ jedinou hodnotu ($10^0 = 1$). Postupně zadejte do pole pro exponent horní meze:

- 12 - stamiliony → `decimal` i `double` OK;
- 13 - miliardy → `decimal` i `double` OK;
- 14 - desítky miliard → `decimal` stále OK, `double` už má problémy v haléřích.

Výsledek se shoduje s dříve uvedenými teoretickými poznatky. 13 cifer koruny + 2 cifry haléře = 15 platných cifer čísla `double`! Na šestnáctou cifru se už u `double` nedá spolehnout.

Součty

Postupně budu zvyšovat počet sčítaných hodnot a zjišťovat, při jaké horní mezi nastanou v `double` problémy. `Decimal` problémy nemá nikdy. Výsledky experimentu shrnuji v tabulce:

Exponent počtu	Problémový exponent horní meze čísel	Součet exponentů
0	14	14
1	13	14
2	12	14
3	10	13
4	9	13
5	7	12
6	6	12
7	5	12
8	3	11

Výsledek se opět shoduje s teoretickou předpovědí z minulého čísla Zpravodaje. Pokud by peněžní částky byly v typu `double` reprezentovány přesně, součet exponentů by byl vždy 14. Když budeme mít 10^2 čísel, musejí být nejvýše 10^{11}

korun velká, neboť pak je součet necelých 10^{13} a se dvěma ciframi na haléře dostáváme 15 platných cifer `double`.

Vzhledem k nepřesné reprezentaci čísla `double` se ale s velkým počtem sčítanců kumuluje zaokrouhlovací chyba, takže při součtu sto milionu čísel mohou být tato čísla nejvýše do stovky korun, abychom se mohli spolehnout na haléře.

Dílčí závěr

Pro součty mnoha čísel je problém v typu `double` udržet přesnost na haléře. **Proto je doporučen na finanční výpočty typ `decimal`.** Na druhou stranu je však třeba si všimnout, že **problém se týká především obrovských firem** typu bank.

Uvažujme totiž nad malou nebo střední firmou. Co ona sčítá? Dejme tomu 10 tisíc transakcí v řádu desetitisíců Kč? Exponenty 4, 5 jsou v pohodě. Když se k tomu přidá tisíc položek po milionu? Exponenty 3, 7 jsou také v pohodě. Myslíte, že jsem odhady poddimenzoval? Stejně je tam velká rezerva. Malá nebo střední firma problémy s `doublem` mít nebude.

Měření rychlosti

Náš program upravíme, abychom mohli výpočty v obou typech porovnat z hlediska rychlosti. Nejdříve naplníme velké pole `doubleů` a velké pole `decimalů` stejnými čísly a poté změříme, za jak dlouho se sečte pole `doubleů` a za jak dlouho pole `decimalů`. Zkrácený výpis následuje:

```
public partial class OknoProgramu : Form
{
    // Konstruktork a metoda Mocnina10 beze změny

    private void tlačítkoSpočti_Click(object sender, EventArgs e)
    {
        // Vstupy ... beze změny

        // Příprava polí
        long početHodnot = Mocnina10(exponentPočtu);
        double[] číslaDouble = new double[početHodnot];
        decimal[] číslaDecimal = new decimal[početHodnot];

        // Příprava náhody - částku vytvoříme ze tří čísel - beze změny

        // Generování náhodných částek
        for (long index = 0; index < početHodnot; index++)
        {
            // Tři čísla ... beze změny

            // Náhodná částka v haléřích, resp. korunách ... beze změny

            // Do polí
            číslaDouble[index] = částkaDouble;
            číslaDecimal[index] = částkaDecimal;
        }
    }
}
```

```

// Sčítání
DateTime čas1 = DateTime.Now;
double součetDouble = 0;
for (int index = 0; index < početHodnot; index++)
{
    součetDouble += číslaDouble[index];
}
DateTime čas2 = DateTime.Now;
decimal součetDecimal = 0;
for (int index = 0; index < početHodnot; index++)
{
    součetDecimal += číslaDecimal[index];
}
DateTime čas3 = DateTime.Now;

// Výsledky
TimeSpan dobaVýpočtuDouble = čas2 - čas1;
TimeSpan dobaVýpočtuDecimal = čas3 - čas2;

// Zobrazení výsledků
string zpráva =
    "Doba výpočtu double (sekundy): " +
    dobaVýpočtuDouble.TotalSeconds.ToString("F2") +
    Environment.NewLine +
    "Doba výpočtu decimal (sekundy): " +
    dobaVýpočtuDecimal.TotalSeconds.ToString("F2");
MessageBox.Show(zpráva);
}
}

```

Výsledky na mém počítači shrnuje tabulka:

Počet hodnot	Doba pro double	Doba pro decimal
10^6	0,02 s (a méně)	0,06 s
10^7	0,06 s	0,62 s
$5 \cdot 10^7$	0,31 s	3,17 s

Abych dostal poslední řádek, musel jsem do programu ještě přidat násobení počtu pětkou. Do 10^8 jsem už nechtěl jít, dle grafů *Správce úloh* (obsazení paměti) jsem měl pocit, že pak už by počítač musel swapovat na disk a změřil bych kdovíco.

Samozřejmě čísla nemají absolutní platnost, zvláště měření malých intervalů může být problematické, závěry však, mám za to, vyvodit můžeme. Myslím, že **experiment jednoznačně potvrzuje výrazně vyšší výpočetní výkon v typu double**, v tomto případě cca 10× vyšší oproti decimalu.

Rozdíl ve výkonnosti se ale nějak projevil až v těch desítkách milionů hodnot. Pro tak velký počet už ale hodnoty samotné nesmějí být veliké, pokud lpíme na haléřích, viz předchozí dílčí závěr. V tomto experimentu na mém počítači procesor nezatěžovalo nic jiného. Na vytíženém serveru může být situace jiná, tam se rozdíl může znatelně projevit i při menším počtu hodnot.

Zaokrouhlování

Jsou i další rozdíly mezi `doublem` a `decimalem`. Třeba zaokrouhlování. Nepřesný `double` by nás mohl někdy při zaokrouhlování překvapit (kdyby zobrazoval `XXX,5`, ale uvnitř měl `XXX,49999999...`). Ale zaokrouhlování je třeba pokaždé dobře promyslet, pokud na něm záleží, a to bez ohledu na datový typ.

Závěry

Pro finanční výpočty, pro případy, kdy záleží na haléřích i u obrovských částek, je **nepsaným standardem použití datového typu `decimal`**. V obou posledních Zpravodajích jsme si podrobně rozebrali důvody. Přestože tento standard pochází zřejmě z půdy velikých firem typu banky, není příliš důvodů se jím neřídit obecně.

Viděli jsme ale také, že v typickém případě **malých a středních firem si, když chcete nebo potřebujete**, nakonec bez problémů **vystačíte s obyčejným `doublem`**. Rovněž bych příliš nespěchal se zaváděním typu `decimal` ve školní výuce - studenti obvykle mají dost, aby zvládli dva číselné typy. No a k typu `double` se můžete zkusit obrátit také tehdy, když zoufale honíte zlomky sekund v odezvě aplikace. Pro posouzení, kdy je situace bezpečná, máte výše uvedenou tabulku.

Radek Vystavěl, 8. srpna 2013

Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu www.moderniProgramovani.cz.