

Zpravodaj 01/2011: Konzolové aplikace

Ke studiu potřeba: 5. kapitola učebnice pro začátečníky

Konzolové aplikace - Aplikace, které s uživatelem komunikují prostřednictvím textového uživatelského rozhraní („černá obrazovka“, Příkazový řádek).

Použití:

- Některé nástroje pro správu (populární zejména na Linuxu);
- Nástroje používané v příkazových dávkách;
- Často v knihách.

Vytváření je jednoduché:

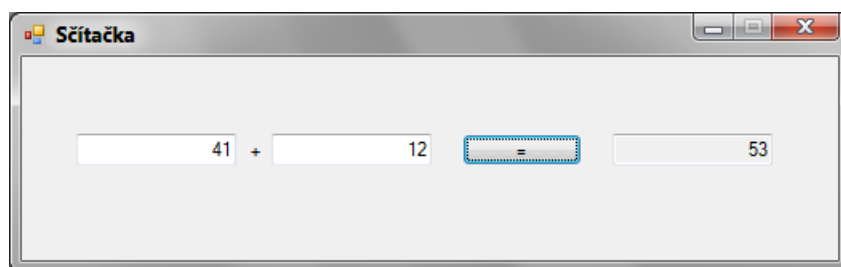
- U nového projektu zvolíme šablonu „Console Application“.
- Vstup dat pomocí metody `Console.ReadLine`. Čeká na zadání řádku textu ukončeného Enterem, zadaný text vrací jako svůj výsledek.
- Výstup dat pomocí metody `Console.Write`, resp. `Console.WriteLine`. V parametru předáváme, co chceme vypsát. `WriteLine` po zápisu odřádkuje.
- Datové typy, syntaxe, třídy, to všechno je stejné jako u aplikací s grafickým uživatelským rozhraním (Windows Forms).

Rozdíly oproti GUI aplikacím:

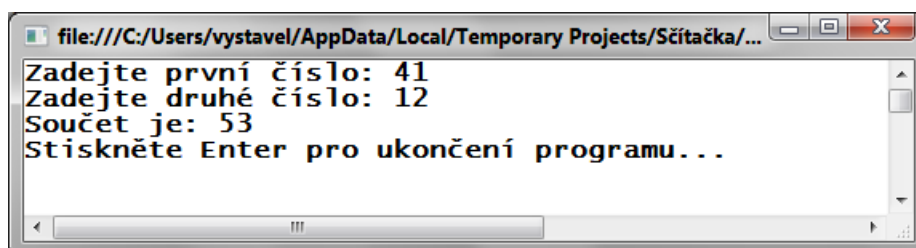
- Jinak řešený vstup a výstup dat.
- Konzolové aplikace nejsou řízeny událostmi. Prostě se provedou příkazy metody `Main` v pořadí, jak jsou zapsány.

Zadání úlohy, kterou budeme řešit:

V 5. kap. učebnice pro začátečníky je program *Sčítačka* řešící součet dvou zadaných čísel:



Analogický program si nyní vytvoříme jako konzolovou aplikaci:



Řešení:

Připomenu kód obsluhy tlačítka z původního programu:

```
private void tlačítkoRovnáSe_Click(object sender, EventArgs e)
{
    try
    {
        // Hodnoty Text obou vstupních polí převést na čísla
        int číslo1 = Convert.ToInt32(poleČíslo1.Text);
        int číslo2 = Convert.ToInt32(poleČíslo2.Text);

        // Čísla sečíst
        int výsledek = číslo1 + číslo2;

        // Výsledek převést na řetězec a zapsat jako hodnotu
        //   Text výstupního pole
        poleVýsledek.Text = Convert.ToString(výsledek);
    }
    catch
    {
        // Při nekorektním zadání informujeme uživatele
        MessageBox.Show("Byl zadán chybný údaj.");
    }
}
```

V konzolové aplikaci budou odlišně řešeny vstupy, jinak je řešení velmi podobné. Vytvořte nový projekt ze šablony „Console Application“ a jeho jediný zdrojový text upravte do následující podoby:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Sčítačka
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // Přečteme vstupní údaje
                Console.Write("Zadejte první číslo: ");
                string vstup1 = Console.ReadLine();
                Console.Write("Zadejte druhé číslo: ");
                string vstup2 = Console.ReadLine();

                // Vstupní hodnoty převedeme na čísla
                int číslo1 = Convert.ToInt32(vstup1);
                int číslo2 = Convert.ToInt32(vstup2);

                // Čísla sečteme
                int výsledek = číslo1 + číslo2;
            }
        }
    }
}
```

```
        // Výsledek převedeme na řetězec a zobrazíme jej
        Console.WriteLine("Součet je: " + výsledek.ToString());
    }
    catch
    {
        // Při nekorektním zadání informujeme uživatele
        Console.WriteLine("Byl zadán chybný údaj.");
    }

    // Aby se okno hned nezavřelo...
    Console.Write("Stiskněte Enter pro ukončení programu...");
    Console.ReadLine(); // pouze čekáme ná Enter
}
}
```

Vidíte? int, Convert, try-catch - to je stejné, jak to znáte z GUI aplikací...

Na závěr:

Zatímco původní GUI aplikace bez speciálního úsilí umožňuje opakovanou práci, vytvořená konzolová aplikace funguje jednorázově. Pokud byste chtěli umožnit opakování, museli byste výslovně programovat cyklus, např. do-while ze 4. kapitoly učebnice pro středně pokročilé.

Radek Vystavěl, 4. prosince 2011

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