

Zpravodaj moderní Programování 2/2013: Soubory XML

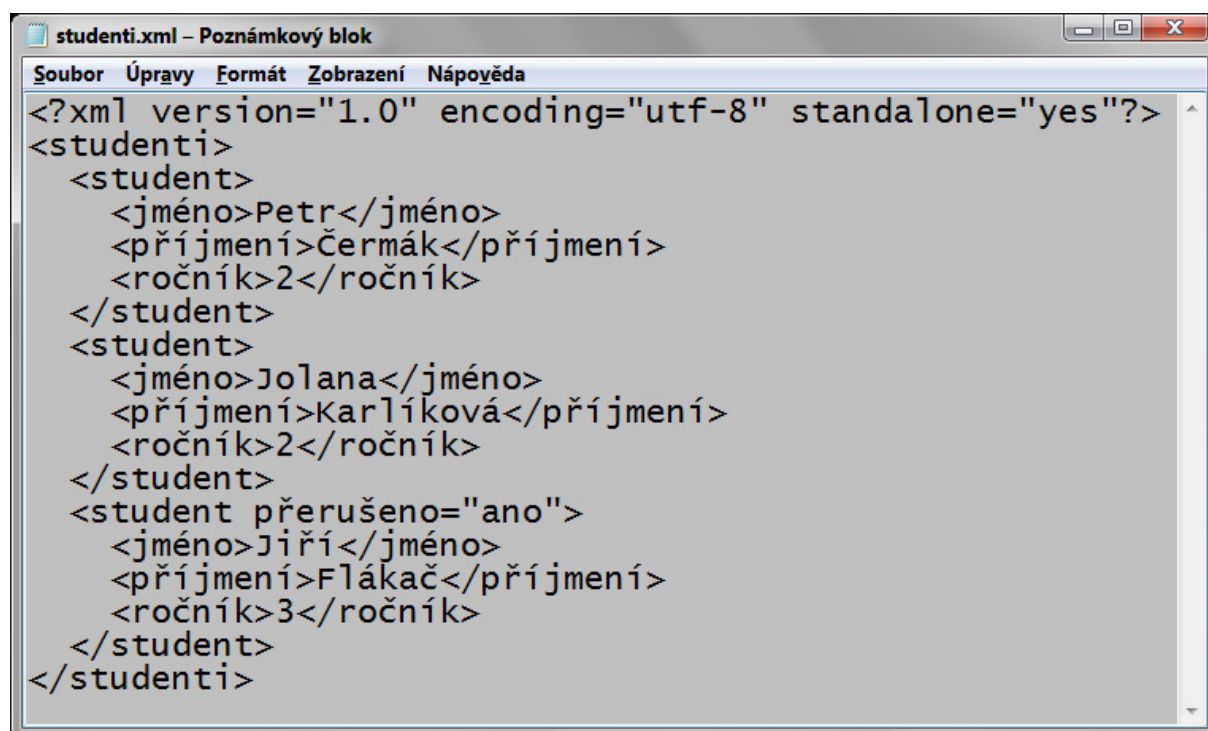
Obtížnost: středně pokročilí

Strukturovaná data a soubory XML

Chcete-li do textového souboru uložit strukturovaná data, můžete použít např. formát CSV, kde co řádek, to jeden záznam, a kde se jednotlivé atributy záznamu oddělují obvykle středníkem. Práce s formátem CSV je popsána v sedmé kapitole učebnice pro středně pokročilé.

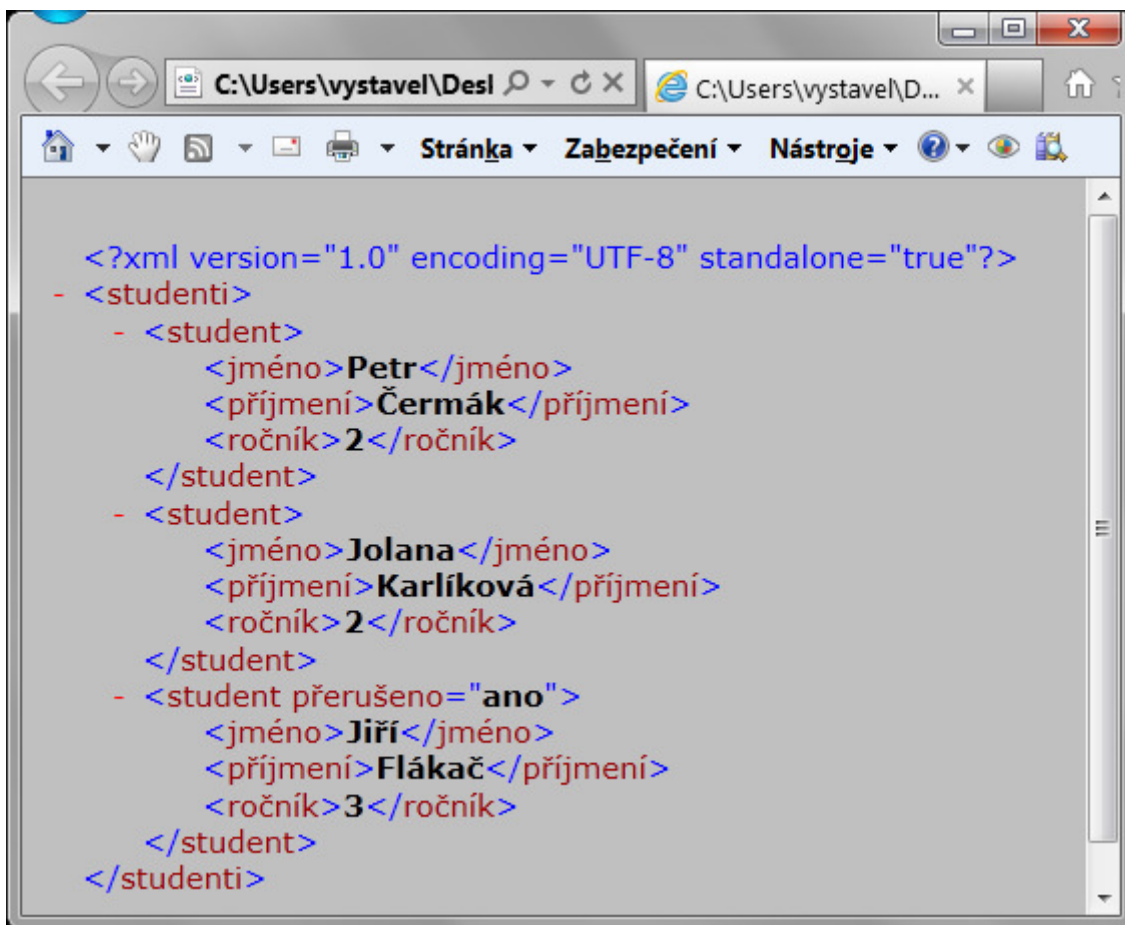
„Profesionálnější“ formátem strukturovaných dat je XML. Stejně jako s CSV se s ním setkáte např. v exportech z různých informačních systémů, mj. z internetového bankovníctví. Formát XML používá také Visual Studio pro své definiční soubory projektů (*.csproj), ve webových aplikacích ASP.NET existuje `web.config` formátovaný podle XML atd.

Pokud se budete chtít s formátem XML seznámit podrobněji, určitě mohou doporučit pěknou knihu paní Grusové. Pro tuto chvíli jen malá ukázka:



```
studenti.xml - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<studenti>
  <student>
    <jméno>Petr</jméno>
    <příjmení>Čermák</příjmení>
    <ročník>2</ročník>
  </student>
  <student>
    <jméno>Jolana</jméno>
    <příjmení>Karlíková</příjmení>
    <ročník>2</ročník>
  </student>
  <student přerušeno="ano">
    <jméno>Jiří</jméno>
    <příjmení>Flákač</příjmení>
    <ročník>3</ročník>
  </student>
</studenti>
```

To byl náhled v Poznámkovém bloku (XML je, podobně jako CSV, soubor prostého textu). XML můžete zobrazit také ve webovém prohlížeči, který umožňuje jednotlivé uzly sbalit nebo rozbalit a navíc barevně odlišuje značky od obsahu:



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <studenti>
  - <student>
    <jméno>Petr</jméno>
    <příjmení>Čermák</příjmení>
    <ročník>2</ročník>
  </student>
  - <student>
    <jméno>Jolana</jméno>
    <příjmení>Karlíková</příjmení>
    <ročník>2</ročník>
  </student>
  - <student přerušeno="ano">
    <jméno>Jiří</jméno>
    <příjmení>Flákač</příjmení>
    <ročník>3</ročník>
  </student>
</studenti>
```

Jak je vidět, XML soubor kromě **úvodní deklarace** obsahuje **elementy** vymezené otevírací a uzavírací značkou podobně jako v HTML. XML soubor musí mít vždy jeden **kořenový element**, v našem případě element `studenti`. Do každého elementu je možno vkládat **elementy vnořené** a výsledkem je stromová struktura. Element může mít také **atributy**, které se zapisují do jeho otevírací značky. V naší ukázce má jeden z elementů `student` atribut `přerušeno`.

XML a .NET

Jelikož jsou XML soubory tvořeny prostým textem, v principu je můžete zpracovávat pomocí tříd `StreamReader` a `StreamWriter` ze jmenného prostoru `System.IO` stejně jako jakékoli jiné textové soubory. Je to však pracné, a tak existují na platformě .NET šikovnější třídy zaměřené přímo na XML.

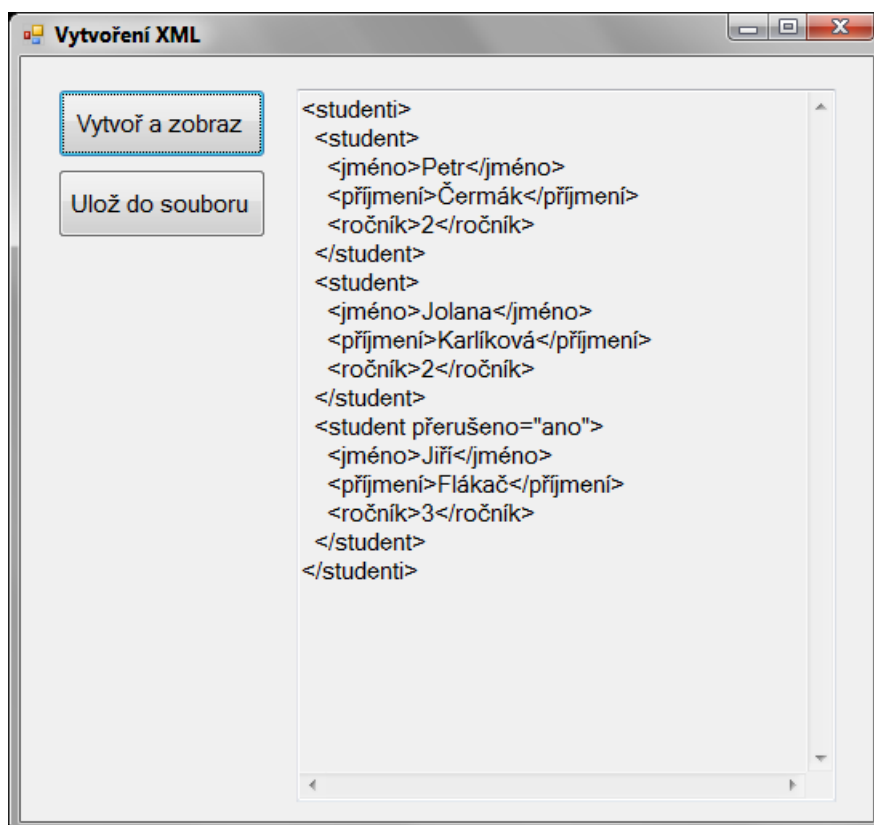
Od počátků .NETu existuje jmenný prostor `System.Xml` se spoustou tříd, jejichž jména začínají na `Xml` a s nimiž se můžete setkat obvykle ve starších, nemoderních :) učebnicích .NET/C#.

Spolu s uvedením pěkné technologie LINQ v roce 2007 ale Microsoft uvedl dílčí technologii LINQ to XML, která nabízí nové, šikovnější API pro práci s XML soubory. Jádro LINQ to XML je soustředěno ve jmenném prostoru `System.Xml.Linq` a tvoří ho řada tříd, jejichž jména začínají na `X`. A právě tyto třídy budeme využívat my.

Ukázka vytvoření XML souboru

Naše studium práce se soubory XML začneme na programu, který vytvoří XML soubor na základě napevno zadaných dat. Takový soubor bychom si samozřejmě mohli snáze vytvořit „ručně“, nyní však jde o to seznámit se se základními postupy XML-programování na co možná nejjednodušším příkladu.

Připravíme program, který nejprve stromový dokument XML vytvoří v paměti a následně jej zapíše na disk.



Následuje kód, který si na závěr blíže vysvětlíme:

```
using System;
using System.Windows.Forms;

using System.Xml.Linq; // přidáno

namespace Vytvoření_XML
{
    public partial class Form1 : Form
    {
        XDocument dokument;

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

private void tlačítkoVytvoř_Click(object sender, EventArgs e)
{
    // Vytvoř dokument
    dokument = new XDocument(new XDeclaration("1.0", "utf-8", "yes"));
    dokument.Add(
        new XElement("studenti",
            new XElement("student",
                new XElement("jméno", "Petr"),
                new XElement("příjmení", "Čermák"),
                new XElement("ročník", "2")
            ),
            new XElement("student",
                new XElement("jméno", "Jolana"),
                new XElement("příjmení", "Karlíková"),
                new XElement("ročník", "2")
            ),
            new XElement("student",
                new XAttribute("přerušeno", "ano"),
                new XElement("jméno", "Jiří"),
                new XElement("příjmení", "Flákač"),
                new XElement("ročník", "3")
            )
        )
    );

    // Zobraz dokument
    poleZobrazení.Text = dokument.ToString();

    // Aktivuj tlačítko Ulož
    tlačítkoUlož.Enabled = true;
}

private void tlačítkoUlož_Click(object sender, EventArgs e)
{
    // Zjistí jméno souboru
    if (oknoVýběruSouboru.ShowDialog() !=
        System.Windows.Forms.DialogResult.OK)
        return;
    string jménoSouboru = oknoVýběruSouboru.FileName;

    // Ulož
    dokument.Save(jménoSouboru);
    MessageBox.Show("Hotovo!");
}
}
}

```

Vysvětlení programu

- Nahoře (`using`) samozřejmě potřebujeme odkaz na příslušný jmenný prostor;
- Pro XML dokument vytvářený v paměti programu slouží třída `XDocument`. Pro její instanci máme připravenou členskou proměnnou okna programu nazvanou `dokument`;
- Pro instancování `XDocumentu` existuje několik variant konstruktoru, my použijeme tu, která přebírá jeden parametr - úvodní deklaraci reprezentovanou instancí `XDeclaration`;
- Voláním metody `Add` přidáme kořenový element, v našem případě element `studenti`;
- Elementy jsou instancemi třídy `XElement`, pro instancování existuje opět několik variant konstruktoru. V kódu se výhradně využívá varianta, která jako první parametr převezme název elementu a za ním následuje libovolný počet vnořených elementů a případně atributy;
- Právě ten proměnlivý počet parametrů a to vnořování je na celém programu to nejzajímavější. Podrobně prostudujte závorky! Snažil jsem se je naformátovat co nejpřehledněji;
- Zápis do souboru na disk se provádí voláním metody `Save` dokumentu.

Závěr

Seznámili jsme se se základní strukturou souborů formátu XML, ukázali jsme si příklad použití tříd z LINQ to XML. Ještě zajímavější určitě bude udělat si ukázkou na „normálních“ datech, tj. ne na datech přímo uvedených v programu. Na tu se můžete těšit do příštího Zpravodaje!

Radek Vystavěl, 5. března 2013

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