Metodický list k didaktickému materiálu

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-215
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Textový soubor – seznámení
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 17–18 let
Úroveň žáků	Mírně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Textový a binární soubor, kódování, dialogová okna
Anotace	Studenti chápou rozdíly mezi soubory a problém kódování, učí se vytvořit jednoduchý textový soubor, také pomocí dialogových oken.
Použité zdroje	VYSTAVĚL, Radek. *Moderní programování: sbírka úloh k učebnici pro středně pokročilé*. 1. vyd. Ondřejov: moderníProgramování, 2008-2009, 2 sv. ISBN 978-80-903951-3-8. VYSTAVĚL, Radek. *Moderní programování: učebnice pro středně pokročilé*. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1.
Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.	Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod. Při společné práci je vhodné nejprve obtížnější úlohy rozebrat, potom společně se studenty implementovat na počítači. (Rozbor nejlépe na tabuli, synchronní řešení s promítáním) Prezentace obsahuje stručné shrnutí poznatků potřebných pro řešení příkladů. V pracovním listu je zadání cvičení – většinou se jedná o úlohy, které by měli studenti naprogramovat samostatně. Není nutné, aby všichni zpracovali všechno, vhodné je diferencovat podle jejich zájmu a schopností. Obtížnější úlohy jsou označeny hvězdičkou. Součástí materiálu je zdrojový kód těchto příkladů. Návrh způsobu hodnocení: ohodnocení samostatné práce během hodiny např. podle volby a počtu úloh a elaborace řešení (efektivnost, komentáře…).

Prohlášení autora

Tento materiál je originálním autorským dílem. K vytvoření tohoto didaktického materiálu nebyly použity žádné externí zdroje s výjimkou zdrojů citovaných v metodickém listu.

Obrázky (schémata a snímky obrazovek) pocházejí od autora.

215. Textové soubory – úvod

Prakticky nevystačíme se zpracováním údajů v operační paměti – data je třeba také ukládat dlouhodobě – na disky v podobě souborů.

Soubory binární – informace v nich se ukládají podle jejich vnitřní reprezentace v počítači, pro člověka jsou nečitelné bez programu, který s nimi umí pracovat. Patří sem jednak **soubory veřejné** (např. obrázky BMP nebo JPG), jednak **proprietární.** (DOC). Tyto soubory jsou vždy strukturované a aby se s nimi dalo manipulovat, je třeba dopředu vědět, co je ve kterém bajtu. Pro veřejné soubory je tento **formát** znám – může s nimi manipulovat kdokoliv, proprietární formáty znají pouze jejich tvůrci.

Textové soubory – obsahují informace v čitelné textové formě. Jsou buď nestrukturované – TXT nebo strukturované. (INI, CSV, XML)

Seznámíme se s tvorbou a čtením textových souborů.

Tvorba textového souboru

Vytvoříme soubor pokus.txt, do kterého zapíšeme dva řádky: (Líbezná první věta obsahuje veškerou českou diakritiku)

Příliš žluťoučký kůň úpěl ďábelské ódy.

Uvidíme, jak to dopadne s kódováním.

Abychom mohli pracovat se soubory, musíme k našemu projektu připojit příslušnou jednotku (jmenný prostor). Proto připojíme na začátek programu klauzuli:

```
using System.IO;

…

private void buttonTS1_Click(object sender, EventArgs e)
        {
            //vytvoření prvního pokusného textového souboru
            string Jmeno = "Pokus.txt";
            StreamWriter soubor = new StreamWriter(Jmeno);
            //otevření souboru
```

```
            soubor.Write("Příliš žluťoučký kůň úpěl ");//zápis
            soubor.WriteLine("ďábelské ódy.");
            soubor.WriteLine("Uvidíme jak to dopadne s kódováním.");
            //zápis s odřádkováním
            soubor.Close();
            //Zavření souboru
            MessageBox.Show("Soubor vytvořen");
        }
```

Otevření souboru

vysílá požadavek na operační systém a zpřístupňuje ho našemu programu a zamyká programům jiným. (Přitom se např. kontrolují práva na manipulaci se souborem)

Zápis se provádí pomocí instance třídy **StreamWriter,** odkaz na ni se uloží do proměnné soubor a následně voláme její metody:

Write – zápis bez odřádkování

WriteLine – zápis s odřádkováním

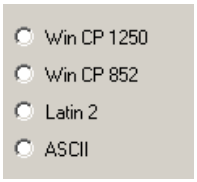
Parametrem obou metod je text, který může být také uložen v proměnné.

Close – zavření souboru, odemčení zámku a uvolnění vyrovnávací paměti. (Při vynechání tohoto příkazu bychom mohli ztratit některá data)

Stream – proud zde znamená proud bajtů, který může být souborem, ale také tokem dat v síti, apod.

Kódování

Přidáme na formulář 4 radioButtonky podle obrázku, ze kterých si vybereme kódování. Jméno souboru tentokrát zapíšeme do textového pole, abychom mohli vytvořené soubory s různým kódováním porovnat.




```
    string Jmeno = textBoxJmeno.Text;
            Encoding kod;
            if (radioButtonL2.Checked)
                kod = Encoding.GetEncoding(28592);
            else
                if (radioButtonW1250.Checked)
                    kod = Encoding.GetEncoding(1250);
                else
```

```
            if (radioButtonW852.Checked)
                kod = Encoding.GetEncoding(852);
            else
                kod = Encoding.ASCII;
        StreamWriter soubor = new StreamWriter(Jmeno,false,kod);
```

Našeptávač nenabízí všechny kódy – proto jsme použili metodu GetEncoding, která má jako parametr číslo požadované stránky. StreamWriter zde má tři parametry, prostřední – false nebo true rozhoduje o tom, zda se v případě existujícího souboru má soubor přepsat (false) nebo nový obsah přidat na jeho konec. (true)

Dialogové okno pro uložení souboru

všechna standardní dialogová okna, která používáme ve Windows, nalezneme v Toolbar – kategorie **Dialogs.** Pro práci se soubory budeme používat **SaveFileDialog** pro uložení a **OpenFileDialog** pro otevření souboru.

Komponentu přetáhneme na formulář, pojmenujeme, případně můžeme nastavit některé její vlastnosti. (Například vlastnost Filter ve tvaru: Texty|*.txt|Všechno|*.*)

Dialogy se spouštějí metodou **JménoOkna.ShowDialog()**. Tato metoda má návratovou hodnotu výčtového typu – **DialogResult**, která závisí na tlačítku, kterým byl dialog zavřen. **OK** vrací **DialogResult.OK**, **Storno** pak **DialogResult.Cancel**.

Příklad

Uložte do textového souboru tabulku prvních deseti sudých přirozených čísel ve tvaru:

1. číslo: 2

2. číslo: 4

…

10:číslo: 20

```
private void buttonSuda_Click(object sender, EventArgs e)
        {
            //Uloží do souboru, který si vybere uživatel prvních dvacet sudých
přirozených čísel.
            DialogResult odp = saveFileDialogUloz.ShowDialog();
            if (odp == DialogResult.OK)
                textBoxJmeno.Text = saveFileDialogUloz.FileName;
            StreamWriter soubor = new StreamWriter(textBoxJmeno.Text, false,
Encoding.Default);
```

```
            for (int i = 1; i < 11; i++)
            {
                int sud=2*i;
                string radek = i.ToString() + ". číslo " + sud.ToString();
                soubor.WriteLine(radek);
            }
            soubor.Close();
            MessageBox.Show("Soubor vytvořen");
        }
    }
```

Důležité

Zápis do textového souboru:

StreamWriter soubor = new StreamWriter(Jmeno, false, Encoding.Default);

soubor.Write – zápis bez odřádkování

soubor.WriteLine – zápis s odřádkováním

soubor.Close – zavření souboru

Práce se standardními dialogy:

```
DialogResult odp = saveFileDialogUloz.ShowDialog();
        if (odp == DialogResult.OK)
           textBoxJmeno.Text = saveFileDialogUloz.FileName;
```

Pracovní list

Cvičení

Připravte si nový projekt se třemi tlačítky a textovým polem.

1. Na stisknutí prvního tlačítka se do textového souboru uloží text z textového pole

2. Na stisknutí druhého tlačítka se to textového souboru uloží tabulka hodnot druhých mocnin:

1 1

2 4

…

10 100

3.(*) Na stisknutí třetího tlačítka se to textového souboru uloží tabulka hodnot faktoriálu:

0! 1

1! 1

2! 2

…

10! 3628800

Umožněte uložení souboru pomocí standardního dialogu.

Řešení

1.

```
//uložení textového pole do souboru
            DialogResult odp = saveFileDialogUloz.ShowDialog();
            if (odp == DialogResult.OK)
                textBoxJmeno.Text = saveFileDialogUloz.FileName;
            StreamWriter soubor = new StreamWriter(textBoxJmeno.Text, false,
Encoding.Default);
            soubor.Write(textBox1.Text);
            soubor.Close();
```

2.

```
private void button2Moc_Click(object sender, EventArgs e)
        {
            //uložení tabulky druhých mocnin do souboru
            DialogResult odp = saveFileDialogUloz.ShowDialog();
            if (odp == DialogResult.OK)
                textBoxJmeno.Text = saveFileDialogUloz.FileName;
            StreamWriter soubor = new StreamWriter(textBoxJmeno.Text, false,
Encoding.Default);
            for (int i = 0; i < 11; i++)
            {
                int moc2 = i * i;
                string radek = i.ToString() + "  " + moc2.ToString();
                soubor.WriteLine(radek);
            }
            soubor.Close();
            MessageBox.Show("Soubor vytvořen");

        }
```

3.

…

```
            for (int i = 0; i < 11; i++)
            {
                int fakt = 1; //výpočet faktoriálu čísla i

                for (int j = 1; j < i+1; j++)
                    fakt*=j;
                string radek = i.ToString() + "! =  " + fakt.ToString();
                soubor.WriteLine(radek);
            }
```

…

Faktoriály ovšem není nutné (také to není efektivní) počítat vnořeným cyklem (pokud si jej nechcete procvičit). Stačí si uvědomit, že každý další výsledek je právě i* větší než předchozí.

```
         int fakt = 1;
            for (int i = 1; i < 11; i++)
            {
                fakt *= i;
                string radek = i.ToString() + "! =  " + fakt.ToString();
                soubor.WriteLine(radek);
            }
```