Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-304
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech III
Téma didaktického materiálu	Seznamy a soubory
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 17–18 let
Úroveň žáků	Středně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Indexové seznamy, rozvírací seznam, textový soubor, cykly for, foreach
Anotace	Studenti zpracovávají soubory s využitím rozvíracích seznamů, rozlišují zpracování dat cykly for a foreach
Použité zdroje	DRÓZD, Januš a Rudolf KRYL. *Začínáme s programováním*. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X. ELLER, Frank. *C# - začínáme programovat: podrobný průvodce začínajícího uživatele*. 1. vyd. Praha: Grada, 2002, 240 s. ISBN 80-247-0324-6. LIBICHER, Ivan a Pavel TÖPFER. *Od problému k algoritmu a programu: sbírka řešených úloh z programování*. 1. vyd. Praha: Grada, 1992, 119 s. Educa '99. ISBN 80-854-2482-7. TÖPFER, Pavel. *Algoritmy a programovací techniky*. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6. TÖPFEROVÁ, Dana a Pavel TÖPFER. *Sbírka úloh z programování*. Vyd. 1. Praha: Grada, 1992, 98 s. Educa '99. ISBN 80-854-2499-1. VYSTAVĚL, Radek. *Moderní programování: učebnice pro pokročilé*. Ondřejov: moderníProgramování, 2011, 149 s. ISBN 978-80-903951-7-6. VYSTAVĚL, Radek. *Moderní programování: učebnice pro středně pokročilé*. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1. TIŠER, Robert a Zdeněk NOVOTNÝ. *MS POWERPOINT a ACCESS v příkladech* [CD ROM]. Firma Pachner
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) 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…).

Metodický list k didaktickému materiálu

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.

304. Indexové seznamy a soubory

Indexované seznamy se ideálně hodí pro zpracování souborů – protože nejsme vázáni jejich předem deklarovanou délkou.

Pro dalším příklady bychom mohli používat vlastní seznam (List<>), ale protože stejně budeme pracovat s listBoxy, můžeme použít jejich seznam – **listBox.Items**

Příklad 1.

Načteme seznam jmen z textového souboru a zobrazíme ho do listBoxu. Soubor Jmena.txt obsahuje seznam jmen a příjmení oddělených mezerami, abychom ho nemuseli vyhledávat, umístíme ho do adresáře **bin\Debug** projektu.

```
using System.IO;

//kvůli práci se soubory

…

private void buttonNacti_Click(object sender, EventArgs e)
        {
            //načti vstupní soubor
            StreamReader vstup = new StreamReader("Jmena.txt");
            string radek;
            while ((radek = vstup.ReadLine()) != null)
```

```
            {
                listBoxJmena.Items.Add(radek);
            }
            vstup.Close();
        }
```

Příklad 2.

Určíme „přibližný“ počet žen v seznamu – jména končící na znak ‘á’. Na řetězec obsahující jméno se přitom můžeme dívat jako na pole znaků, poslední znak má index odpovídající délce řetězce.

```
private void buttonPocZen_Click(object sender, EventArgs e)
        {
            //počet žen
            int pocet =0;
            for (int i = 0; i < listBoxJmena.Items.Count; i++)
            {
                string jmeno=listBoxJmena.Items[i].ToString();
                int delka = jmeno.Length;
                if (jmeno[delka - 1] == 'á')//poslední písmeno
                {
                    pocet++;
                }

            }
            MessageBox.Show(pocet.ToString());
        }
```

Poznámka: K procházení seznamu listBoxJmena.Items můžeme použít cyklus Foreach:

foreach (string jmeno in listBoxJmena.Items)…

Příklad 3.

Umožníme nějaké úpravy – např. odstranění vybrané položky v listBoxu.

```
        private void buttonSmaz_Click(object sender, EventArgs e)
        {
            //Smaž vybranou
            int i = listBoxJmena.SelectedIndex;
            if (i == -1)
                MessageBox.Show("Není nic vybráno");
            else
                listBoxJmena.Items.RemoveAt(i);

        }
```

Příklad 4.

Upravený soubor uložíme.

```
private void buttonUloz_Click(object sender, EventArgs e)
        {
           //Ulož soubor
            StreamWriter vystup = new StreamWriter("Jmena2.txt");
            foreach (string jmeno in listBoxJmena.Items)
                vystup.WriteLine(jmeno);
            vystup.Close();
        }
```

Důležité

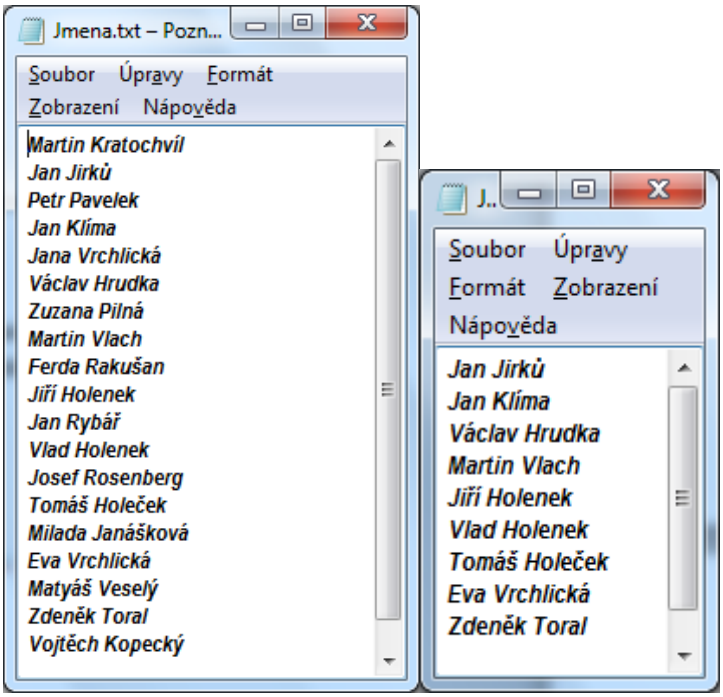
Rozlišujte, kdy je možné používat cyklus **Foreach** (jednoduché, ale nemáme přístup k indexu a vstupní seznam nelze měnit) a cyklus **For**.

Pracovní list

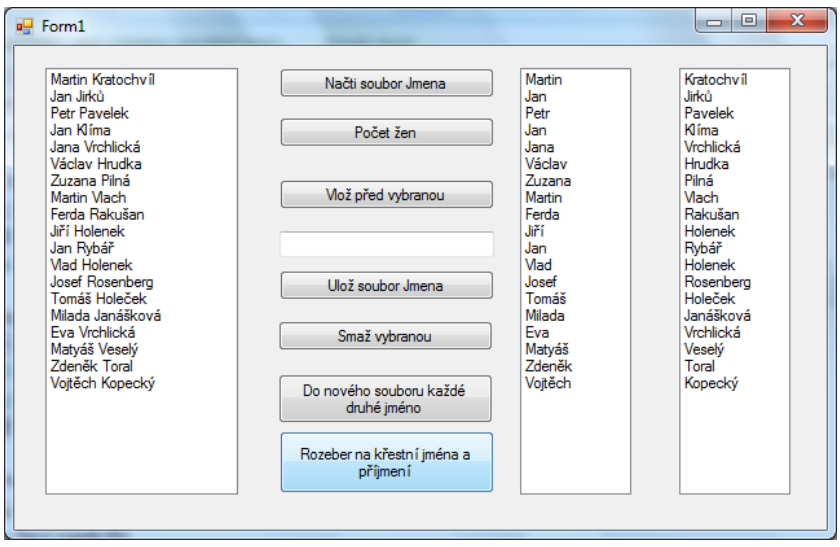
Cvičení

1. Přidejte tlačítko pro vložení do seznamu před danou položku.

2. Naprogramujte převedení seznamu z listBoxu do dalšího textového souboru, přitom ale budeme ukládat jen každé druhé jméno.



3. (*)Naprogramujte rozdělení seznamu do dvou dalších listBoxů tak, že do prvního umístíme pouze jména, do druhého příjmení. (Metoda řetězců split(oddělovací znak))



4. Co by měl dělat následující příkaz? Bude fungovat správně?

```
          foreach (string jmeno in listBoxJmena.Items)
            {
                if (jmeno[1] == 'X')
                    jmeno[1] = 'x';
```

Řešení

1.

```
  private void buttonVlozPred_Click(object sender, EventArgs e)

        {
            //Vlož před vybranou
            int i=listBoxJmena.SelectedIndex;
            if (i == -1)
                listBoxJmena.Items.Add(textBoxVstup.Text);
            else
                listBoxJmena.Items.Insert(i, textBoxVstup.Text);
        }
```

2.

```
private void buttonKDJ_Click(object sender, EventArgs e)

        {
            //Do nového souboru každé druhé jméno
            StreamWriter vystup = new StreamWriter("Jmena3.txt");
            for(int i=0;i<listBoxJmena.Items.Count;i++)
                if (i%2==1)
                    vystup.WriteLine(listBoxJmena.Items[i].ToString());
            vystup.Close();
        }
```

3.

```
  private void buttonJP_Click(object sender, EventArgs e)
        {
            //Rozdělení jmen a příjmení do dalších dvou seznamů
            foreach (string jmeno in listBoxJmena.Items)
            {
                string[] data = jmeno.Split(' ');//Jméno a příjmení je odděleno
mezerou
                listBoxKrest.Items.Add(data[0]);
                listBoxPrijm.Items.Add(data[1]);
            }

        }
```

4.

Příkaz by měl v listBoxu všechna slova, která začínají velkým X převést na slova, začínající malým x. Chyba, kvůli které nebude fungovat, se objeví už při překladu – cyklus foreach neumožňuje měnit vlastnosti seznamu, slouží pouze ke čtení.