

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-220
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Další příkazové konstrukce
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	Cyklus foreach, přepínač switch
Anotace	Studenti se seznamují s dalšími příkazovými konstrukcemi a na příkladech se učí je vhodně používat,
Použité zdroje	ELLER, Frank. <i>C# - začínáme programovat: podrobný průvodce začínajícího uživatele</i>. 1. vyd. Praha: Grada, 2002, 240 s. ISBN 80-247-0324-6. KUKAL, Jaromír. <i>Myšlením k algoritmům</i>. Praha: Grada, 1992, 131 s. ISBN 80-854-2447-9. LIBICHER, Ivan a Pavel TÖPFER. <i>Od problému k algoritmu a programu: sbírka řešených úloh z programování</i>. 1. vyd. Praha: Grada, 1992, 119 s. Educa '99. ISBN 80-854-2482-7. TÖPFEROVÁ, Dana a Pavel TÖPFER. <i>Sbírka úloh z programování</i>. Vyd. 1. Praha: Grada, 1992, 98 s. Educa '99. ISBN 80-854-2499-1. VYSTAVĚL, Radek. <i>Moderní programování: sbírka úloh k učebnici pro středně pokročilé</i>. 1. vyd. Ondřejov: moderníProgramování, 2008-2009, 2 sv. ISBN 978-80-903951-3-8. VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro středně pokročilé</i>. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1. VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro začátečníky</i>. Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7.

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...).
--	---

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.

220. Další příkazové konstrukce

Cyklus Foreach

je určen k procházení polí a jiných datových kolekcí položku za položkou v tom pořadí, jak byly do datové struktury uloženy, aniž by bylo třeba řešit inicializaci, inkrementaci, ani správnou podmínku provádění cyklu.

Má tvar:

foreach (deklarace proměnné, která zastupuje položky **in** kolekce)

```
{
```

Příkazy

```
}
```

Příklad 1.

Určení počtu výskytů znaku "a" v textovém řetězci

```

private void buttonPocA_Click(object sender, EventArgs e)
{
    //Počet áček v daném textu

    string text = textBoxVstup.Text;

    int pocA = 0;

    foreach (char x in text)
        if ((x == 'a') || (x == 'A'))
            pocA++;

    MessageBox.Show(pocA.ToString());
}

```

Pokud zpracováváme složitější struktury a typ složek není zřejmý, můžeme použít následující zápis:

foreach (var proměnná in kolekce)

```

{
    Příkazy
}

```

Typ proměnné si přitom systém zjistí automaticky.

Příklad 2

Výpis obsahu pole řetězců:

```

string[] slova = { "ahoj", "pes", "les" };

foreach(string s in slova)
    textBox1.Text+=s+Environment.NewLine;

```

Omezení tohoto typu cyklu:

- Není v něm přístupná aktuální hodnota indexu
- Prvky kolekce (zde pole) není možné měnit

Přepínač switch

Tento příkaz větvi program podle hodnoty zadané proměnné – pomůže zjednodušit zápis, který by se komplikoval vícenásobným vnořováním příkazu if.

Může mít tvar:

```
switch (proměnná)
{
    case hodnota 1: Příkazy 1;
        break;
    case hodnota 2: Příkazy 1;
        break;
    case hodnota 3: Příkazy ;
        break;
    ...
    default:
        Další příkazy;
        break;
}
```

Nabude-li proměnná hodnoty 1, provedou se Příkazy 1... pokud nenabude žádné z uvedených hodnot, provedou se Další příkazy za návěštím default. (Tato část může být prázdná.)

Příklad 3.

Napište program, který pro daný měsíc (zadaný pořadovým číslem) určí počet dnů v něm.

```
private void buttonDny_Click(object sender, EventArgs e)
{
    //počet dnů v měsíci
    int mesic = Convert.ToInt32(textBoxVstup.Text);
    int dny=0;
    switch (mesic)
    {
        case 2: dny=28;
            break;
    }
```

```

        case 4:dny=30;

            break;

        case 6:dny=30;

            break;

        case 9:dny=30;

            break;

        case 11:dny=30;

            break;

        default:

            dny=31;

            break;

    }

    MessageBox.Show(dny.ToString());

}

```

Důležité

Cyklus foreach slouží k procházení kolekcí prvek po prvku, aniž by bylo možno tyto prvky měnit.

foreach (deklarace proměnné, která zastupuje položky **in** kolekce)

```

{

Příkazy

}

```

Větvení switch

```

switch (proměnná)

{

    case hodnota 1: Příkazy 1;

        break;

    case hodnota 2: Příkazy 1;

        break;

    ...

    default:

        Další příkazy;

}

```

```
        break;
    }
}
```

Pracovní list

Cvičení

1. Vytvořte k vstupnímu řetězci slovo, kde budou všechny znaky *a* (A) nahrazeny hvězdičkou.

Příklad: Abraka – *br*k*

2. Vytvořte k vstupnímu řetězci slovo, které bude mít tytéž znaky v opačném pořadí.

Příklad: Abraka – akarbA

3.(*) Nadeklarujte a naplňte pole deseti čísel a otočte ho postupnými výměnami posledního čísla za první, předposledního za druhé – až doprostřed. Uvědomte si, kdy při zpracování můžete použít cyklus for a kdy foreach.

Příklad: 1 2 3 4 5 6 –6 5 4 3 2 1

4. Procvičte si větvení switch na jednoduchém příkladu: vstupem bude známka číslem, výstupem hodnocení slovem. (1 – výborný atd.)

Řešení

1.

```
private void buttonAnaHv_Click(object sender, EventArgs e)
{
    //Převod áček na hvězdičky, musíme použít výstupní řetězec vystup
    string text = textBoxVstup.Text;
    string vystup = null;
    foreach (char x in text)
    {
        if ((x == 'a') || (x == 'A'))
            vystup += "*";
        else
            vystup += x;
    }
    textBoxVystup.Text = vystup;
}
```

2.

```
private void buttonPozp_Click(object sender, EventArgs e)
{
    //Otočení řetězce s pomocným řetězcem
```

```

        string text = textBoxVstup.Text;

        string vystup = null;

        foreach (char x in text)

            vystup = x + vystup;

        textBoxVystup.Text = vystup;
    }

```

3.

```

private void buttonPozp2_Click(object sender, EventArgs e)
{
    //Otočení pole postupným vyměňováním čísel

    int[] a = new int[10]{1,2,3,4,2,1,5,8,9,7};

    foreach (int x in a)

        textBoxVstup.Text += x.ToString() + " ";

    int n = a.Length;

    int p=n/2;

    int pom = 0;

    //Chceme-li změnit výchozí pole, nemůžeme použít cyklus foreach.

    for (int i = 0; i <= p; i++)
    {
        pom = a[i];

        a[i] = a[n - i - 1];

        a[n - i - 1] = pom;
    }

    foreach (int x in a)

        textBoxVystup.Text += x.ToString() + " ";
}

```

4.

```

private void buttonZnámka_Click(object sender, EventArgs e)
{
    //z známku číslem převedeme na známku slovem.(znamka)

    int z = Convert.ToInt32(textBoxVstup.Text);
}

```

```
string znamka = null;

switch (z)
{
    case 1: znamka = "výborně";

        break;

    case 2: znamka = "chvalitebně";

        break;

    case 3: znamka = "dobře";

        break;

    case 4: znamka = "dostatečně";

        break;

    case 5: znamka = "nedostatečně";

        break;

    default:

        znamka = "taková známka u nás není";

        break;

}

MessageBox.Show(znamka);

}
```