

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-208
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Pole, vstup dat
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	Mírně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Pole, generování náhodných čísel, pole řetězců
Anotace	Studenti programují jednak plnění pole náhodnými čísly, jednak zadávání hodnot od uživatele a dále si procvičují práci s ním
Použité zdroje	<p>DRÓZD, Januš a Rudolf KRYL. <i>Začínáme s programováním</i>. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X.</p> <p>ELLER, Frank. <i>C# - začínáme programovat: podrobný průvodce začínajícího uživatele</i>. 1. vyd. Praha: Grada, 2002, 240 s. ISBN 80-247-0324-6.</p> <p>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.</p> <p>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.</p> <p>VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro středně pokročilé</i>. Ondřejov: moderníProgramovánís.r.o, 2008. ISBN 978-80-903951-2-1.</p>
Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.	<p>Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod.</p> <p>Při společné práci je vhodné nejprve obtížnější úlohy rozebrat, potom společně se studenty implementovat na počítači. (Rozbor nejlépe na tabuli, synchronní řešení s promítáním)</p> <p>V pracovním listu je zadání cvičení – většinou se</p>

	<p>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.</p> <p>Návrh způsobu hodnocení: ohodnocení samostatné práce během hodiny např. podle volby a počtu úloh a elaborace řešení (efektivnost, komentáře...).</p>
--	---

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.

208. Generování prvků pole, vstup dat od uživatele, příklady

Když jsme minule pracovali s polem, zadali jsme jeho prvky přímo – ať dosazením nebo inicializací v deklaraci. Při řešení většiny praktických úloh ovšem potřebujeme umožnit vstup od uživatele nebo z vhodného datového souboru. Pro účely modelování můžeme pole také naplnit z generátoru náhodných čísel:

Příklad 1

```
publicpartialclassForm1 : Form
{
    int[] a = newint[20]; //pole čísel pro další zpracování
    Randomnahoda=newRandom();
}
```

Takto vytvořené pole můžeme současně naplnit a zobrazit:

```
privatevoidbuttonNapln_Click(objectsender, EventArgs e)
{
    //naplnění pole a z generátoru náhodných čísel
}
```

```

for (int i = 0; i < a.Length; i++)
    {
        a[i] = nahoda.Next(10, 20);
        textBoxVypis1.Text += a[i].ToString() + Environment.NewLine;
    }
}

```

Příklad 2

Budeme pracovat s polem slov – tedy textových řetězců. Uživatel nejprve zadá počet slov, pak je bude postupně přidávat. Protože délka pole je omezena deklarací, budeme uživatelem požadovaný počet prvků uchovávat v proměnné, která musí být samozřejmě menší, než deklarovaná délka pole.

```

public partial class Form1 : Form
{
    string []b = newstring[20]; //pole slov, které bude zadávat uživatel
    int n = 0; //skutečná délka pole slov, kterou zvolí uživatel
    int p=0; //p-tá položka, kterou uživatel zadává

```

Zadání počtu prvků:

```

private void buttonPocet_Click(object sender, EventArgs e)
{
    //načtení uživatelem zadané skutečné délky pole b
    //nesmí přesáhnout deklarované
    try
    {
        n = Convert.ToInt32(textBoxPocet.Text);

        if (n >= 20)
            throw new Exception();
    }
    catch

```

```

        {
    MessageBox.Show("Přirozené číslo menší než 20");
    return;

    throw;
        }
    }

```

Vložení prvku do pole z textového políčka:

```

private void buttonAdd_Click(object sender, EventArgs e)
    {
    //přidání slova do pole b - pokud je v něm ještě místo
    if (p < n)
        {
            b[p] = textBoxPrvekPole.Text;
            p++;
        }

    else
    MessageBox.Show("Pole už je plné.");
    textBoxPrvekPole.Text = null;
    textBoxPrvekPole.Focus();
    }

```

Výpis pole slov a výpis pole slov pozpátku

```

private void buttonVypis1_Click(object sender, EventArgs e)
    {
    //výpis pole slov
    for (int i = 0; i < n; i++)
        {
            textBoxVypis1.Text += b[i] + Environment.NewLine;
        }
    }

```

```
        textBoxVypis1.Text += "*****" + Environment.NewLine;
    }
```

```
private void buttonOtoc_Click(object sender, EventArgs e)
{
    //výpis pole slov pozpátku
    for (int i = n-1; i >=0; i--)
    {
        textBoxVypis1.Text += b[i] + Environment.NewLine;
    }
}
```

Pracovní list

Cvičení

Pracujte s polem čísel generovaným v předchozích příkladech.

1. Určete počet sudých čísel v poli a zobrazte je.
2. Zvětšete každou hodnotu v poli desetkrát.
3. Nalezněte největší číslo a zjistěte jeho index. (Kolikáté je, číslováno od nuly, pokud je maxim více, stačí najít index prvního z nich.)
Například: 2, 8, 7, 6, 5, 8, 9 Výsledek je 8, index 1
4. (*) Řešte předchozí úlohu, ale vypište do zvláštního textBoxu indexy všech maxim.
Například: 2, 8, 7, 6, 5, 8, 9 Výsledek je 8, index 1, 5

Řešení

1.

```
private void buttonSud_Click(object sender, EventArgs e)
{
    //počet sudých čísel a jejich výpis
    int pocsud = 0;
    for (int i = 0; i < a.Length; i++)
    {
```

```

if (a[i] % 2 == 0)
    {
pocsud++;
        textBoxVypis2.Text += a[i].ToString() + Environment.NewLine; ;
    }
}
MessageBox.Show("Sudých je "+pocsud.ToString());
}

```

2.

```

privatevoid button10_Click(object sender, EventArgs e)
    {
//vynásobení všech čísel 10
for (int i = 0; i <a.Length; i++)
    {
        a[i]*= 10;
        textBoxVypis2.Text += a[i].ToString() + Environment.NewLine;
    }
}

```

3.

```

privatevoid buttonMax_Click(object sender, EventArgs e)
    {
//nalezení maxima a určení jeho indexu (pokud je jich víc,
//půjde o index prvního výskytu
intmax = -1; intimax=-1;
for (int i = 0; i <a.Length; i++)
    {
if (a[i] >max)
        {
max = a[i];
imax = i;
        }
}
}

```

```

    }

    MessageBox.Show("Maximum: " + max.ToString() + Environment.NewLine +
        "Jeho 1. výskyt: " + imax.ToString());

    }

}

```

4.

```

private void button1_Click(object sender, EventArgs e)
{
    //V nejjednodušším případě vyhledáme maximum jako v minulém příkladu
    //a při druhém průchodu polem vypsát, na kterých pozicích
    // se vyskytuje
    int max = -1;
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] > max)
            max = a[i];
    }
    MessageBox.Show("Maximum: " + max.ToString());
    for (int i = 0; i < a.Length; i++)
    {
        if (a[i] == max)
            textBoxVypis2.Text += i.ToString() + Environment.NewLine;
    }
}

```