

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-209
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Pole čísel, aplikace
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	Pole, aritmetický průměr, odchylka
Anotace	Studenti prakticky ověřují práci společně na výpočtu odchylek od aritmetického průměru, procvičují si zpracování pole na dalších příkladech
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>KUKAL, Jaromír. <i>Myšlením k algoritmům</i>. Praha: Grada, 1992, 131 s. ISBN 80-854-2447-9.</p> <p>TÖPFER, Pavel. <i>Algoritmy a programovací techniky</i>. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6.</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: moderníProgramování s.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.	Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod. Při společné práci je vhodné nejprve obtížnější

	<p>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 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.

209. Pole čísel, aplikace

Příklad 1

Vygenerujte pole 20 náhodných celých čísel mezi 10 a 20 (simulace výsledků měření nějaké veličiny) Vypočítejte aritmetický průměr a odchylky od průměru (na dvě desetinná místa). Náhodná čísla i odchylky zobrazte do dvou textových polí.

Průměr je třeba nejprve vypočítat a pak postupně odčítat od jednotlivých hodnot – proto je třeba, abychom si hodnoty pamatovali v poli čísel.

```
public partial class Form1 : Form
{
    Random nahoda=new Random();

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

```

}

private void buttonAP_Click(object sender, EventArgs e)
{
    double[] cisla = new double[20];
    double prumer, x, odch; //x - náhodné číslo
    double suma = 0;
    for (int i = 0; i < 20; i++)
    {
        //generování, zobrazení čísel a výpočet součtu
        x = nahoda.Next(10,21);
        cisla[i]=x;
        textBoxVysl1.Text += x.ToString("F2") + Environment.NewLine;
        suma += x;
    }
    prumer = suma / 20;
    MessageBox.Show(prumer.ToString("F2"));
    for (int i = 0; i < 20; i++)
    {
        //výpočet a zobrazení odchylek
        odch = cisla[i] - prumer;
        textBoxVysl2.Text += odch.ToString() + Environment.NewLine;
    }
}
}

```

Příklad 2

Simulujte házení kostkou pomocí generátoru náhodných čísel. Zobrazte hody do textBoxu a zjistěte, kolikrát co padlo.

Pro četnosti hodů bychom mohli používat 6 veličin, ale elegantněji (a pro větší množství hodnot mnohem jednodušeji) se to dá udělat pomocí pole četnost[x], kde x je hodnota, jejíž četnost padla, takže například padne-li dvojka, zvýšíme četnost[2] o jedničku...

```

private void buttonKostka_Click(object sender, EventArgs e)
{
    //20 vrhů kostkou a kolikrát co padlo
    int[] cetnost = new int[7];
    //pro jednoduchost cetnost[0] nebudeme používat
    int hod;
    for (int i = 1; i < 7; i++)
        cetnost[i] = 0;
    //vynulování pole četností
    for (int i = 0; i < 20; i++)
    {
        hod = nahoda.Next(1, 7);
        cetnost[hod]++;
        textBoxVysl1.Text += hod.ToString() + Environment.NewLine;
    }

    for (int i = 1; i < 7; i++)
        textBoxVysl2.Text += i.ToString() + ": " + cetnost[i].ToString() +
Environment.NewLine;
}

```

Pracovní list

Cvičení

Vygenerujte členské pole 10 náhodných celých čísel mezi 10 a 20 při vytvoření formuláře, zobrazte ho a s ním pak pracujte. Zobrazte původní pole a vždy pole upravené.

1. Zjistěte, kolikrát se v poli vyskytuje poslední číslo.
2. Zvětšete všechna lichá čísla v poli o 100 .
3. Vyhledejte nejmenší číslo a vyměňte ho s prvním číslem v poli.

4. (*) Naprogramujte průchod celým polem, přičemž budete vždy porovnávat dvojice sousedních prvků. Pokud je větší číslo před menším, vyměňte je. (Probublávání) Kam se přitom dostane největší číslo?

Řešení

Tisk upraveného pole:

```
for (int i = 0; i < 10; i++)
{
    textBoxVysl2.Text += cisla[i].ToString() + Environment.NewLine;
}
```

1.

```
//počet výskytů posledního čísla
int pocet = 0;
for (int i = 0; i < 10; i++)
{
    if (cisla[i] == cisla[9])
        pocet++;
}
MessageBox.Show(pocet.ToString());
```

2.

```
//Zvětšení lichých o 10
for (int i = 0; i < 10; i++)
{
    if (cisla[i] % 2 != 0)
        cisla[i] += 100;
}
```

3.

```
//Nejmenší na začátek
//Abychom mohli vyměnit minimum s 1. prvkem, musíme si pamatovat,
```

```
//kde se v původním poli vyskytovalo.  
  
int imin,pom; //index minima, pomocné číslo na výměnu.  
  
imin = 0;  
  
for (int i = 1; i < 10; i++)  
{  
  
    if (cisla[imin]>cisla[i])  
  
        imin=i;  
  
}  
  
pom = cisla[imin]; cisla[imin] = cisla[0]; cisla[0] = pom;
```

4.

//Proublávání

```
int pom;  
  
for (int i = 0; i < 9; i++)//pro i+1=9 i může být nejvýš 8  
{  
  
    if (cisla[i] > cisla[i + 1])  
  
    {  
  
        pom = cisla[i]; cisla[i] = cisla[i+1]; cisla[i+1] = pom;  
  
    }  
  
}
```