

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-212
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Posloupnosti a pole
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	Aritmetická, geometrická posloupnost, rekurentní zadání, vzorec pro n-tý člen, posloupnost jako datový typ v C#
Anotace	Studenti spojují své znalosti z matematiky a programování a zkoumají vlastnosti posloupností, prohlubují si také programování posloupností a práci s nimi
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>ODVÁRKO, Oldřich. <i>Matematika pro gymnázia: Posloupnosti a řady</i>. 1. vyd. Praha: Prometheus, 1995, 126 s. Učebnice pro střední školy (Prometheus). ISBN 80-858-4991-7.</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: sbírka úloh k učebnici pro začátečníky</i>. 2. vyd. Ondřejov: moderníProgramování, 2008, 2 sv. ISBN 978-80-903951-5-2.</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> <p>VYSTAVĚL, Radek. <i>Moderní programování:</i></p>

	<i>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.	Celý materiál lze zadat jako samostatné cvičení, s individuální dopomocí. Lze používat staré studijní texty, učebnice matematiky, internet. 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.

212. Opakování – Posloupnosti a pole

Pracovní list

Cvičení

V minulé hodině jsme se zabývali programováním několika úloh s posloupnostmi, přitom jsme vždy vlastně pracovali s jediným prvkem – celou posloupnost jsme si nepotřebovali pamatovat. Dnes si vyzkoušíme přístup, kdy budeme celou posloupnost udržovat v paměti počítače jako jedinou strukturovanou proměnnou – pole čísel.

Zopakujme si – odpovězte na otázky:

Jak se deklaruje pole pro deset celých čísel?

Jak se dá při deklaraci naplnit hodnotami?

Jak byste do prvního prvku (s indexem nula) umístili číslo v textovém poli vstup a do druhého prvku dvojnásobek tohoto čísla?

Jak by se pole čísel zobrazilo do textového pole?

Jak by se naplnilo náhodnými čísly z generátoru?

Řešte pomocí pole následující úlohy o posloupnostech:

1. Naprogramujte naplnění pole aritmetickou, případně geometrickou posloupností.
(Například: AP: 2, 4, 6, ...20; GP: 1, 2, 4, ...512)

2.(*). Rozhodněte, zda čísla v daném poli reprezentují aritmetickou posloupnost.

3.(*). Rozhodněte, zda čísla v daném poli reprezentují geometrickou posloupnost.

4. Naprogramujte plnění pole Fibonacciho posloupností.

($F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}$; 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...)

5. Vypište posloupnost pozpátku.

The screenshot shows a Windows application window titled 'Form1'. On the left side, there are four radio buttons: 'Nahoda', 'AP', 'GP', and 'Fibonacci'. The 'Fibonacci' option is selected. Below these are three buttons: 'Je AP nebo GP?', 'Od prvního d o posledního', and 'Od posledního do prvního'. On the right side, there is a list box containing the numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.

Poznámka: Úlohy 1. 4. i 5. se dají řešit bez použití pole (viz minulá hodina), výhodou daného postupu je šetření paměti.

Řešení

Jak se deklaruje pole pro deset celých čísel?

```
int[] posl = new int[10];
```

Jak se dá při deklaraci naplnit hodnotami?

```
Např. int[] posl = new int[10]{1,2,3,4,5,6,7,7,8,9};
```

Jak byste do prvního prvku (s indexem nula) umístili číslo v textovém poli vstup a do druhého prvku dvojnásobek tohoto čísla?

```
posl[0] = Convert.ToInt32(textBoxVstup.Text);  
posl[1] = 2*posl[0];
```

Jak by se pole čísel zobrazilo do textového pole?

```
for (int i = 0; i < 10; i++)  
    textBoxVystup.Text += posl[i].ToString() + Environment.NewLine;
```

Jak by se naplnilo náhodnými čísly z generátoru?

```
for (int i = 0; i < 10; i++)  
    {posl[i] = nahoda.Next(1,10);  
    }
```

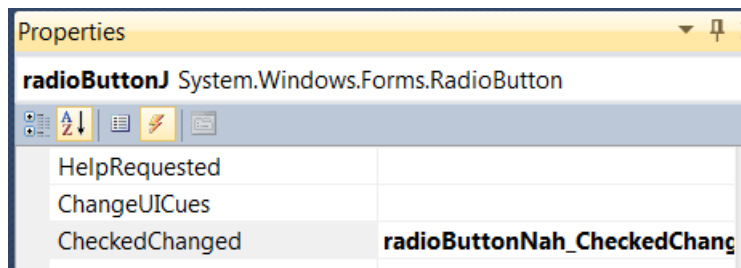
Zdrojové kódy

```
namespace Posloupnosti_a_pole
```

```
{  
    public partial class Form1 : Form  
    {  
        int[] posl = new int[10]  
        Random nahoda = new Random();  
        public Form1()  
        {  
            InitializeComponent();  
        }  
    }  
}
```

...

// Volbu plnění polí rozhodneme podle zaškrtnutí radioButtonu, a můžeme naprogramovat jednou jako společnou událost. (pro ostatní tlačítka vybereme v okně Properties



```
private void radioButtonNah_CheckedChanged(object sender, EventArgs e)  
{  
    if (radioButtonNah.Checked) //náhodná  
        for (int i = 0; i < 10; i++)  
        {  
            posl[i] = nahoda.Next(1,10);  
        }  
    if (radioButtonP1.Checked) //aritmetická (jde samozřejmě i jinak)  
    {  
        for (int i = 0; i < 10; i++)  
        {  
            posl[i] = 2 * (i+1);  
        }  
    };  
    if (radioButtonP2.Checked) //geometrická  
        for (int i = 0; i < 10; i++)  
        {  
            posl[i] = Convert.ToInt32(Math.Pow(2,i));  
        }  
    if (radioButtonP3.Checked) //Fibonacci
```

```

{
    posl[0] = 1;
    posl[1] = 1;
    for (int i = 2; i < 10; i++)
    {
        posl[i] = posl[i - 1] + posl[i - 2];
    }
}
if (radioButtonJ.Checked) //jiná posloupnost
{
    for (int i = 0; i < 9; i++)
    {
        posl[i] = 2 ;
    }
    posl[9] = 3;
};
}

private void buttonJaka_Click(object sender, EventArgs e)
{
    int d = posl[1] - posl[0]; //testujeme AP
    int i=1;
    while ((i < 9) && ((posl[i + 1] - posl[i]) == d))
        i++;
    if (i==9)
        MessageBox.Show("Je aritmetická");
    else
        MessageBox.Show("Není aritmetická");
    double q = (double)posl[1] / posl[0]; ]; //testujeme GP
    //konverze je nutná, jinak se jedná o celočíselné dělení.
    //žádná z našich posloupností nezačíná nulou, v obecnějším případě by
    //bylo nutné to rovněž ošetřit
    double p;
    i = 1;
    do
    {
        p = (double)posl[i + 1] / posl[i];
        i++;
    }
    while ((i < 9) && (p == q));
    if (p==q)
        MessageBox.Show("Je geometrická");
}

```

```

        else
            MessageBox.Show("Není geometrická");
    }

    private void buttonObr_Click(object sender, EventArgs e)
    {
        //výpis pozpátku
        textBoxVystup.Text = null;
        for (int i = 9; i >=0; i--)
            textBoxVystup.Text += posl[i].ToString() + Environment.NewLine;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //výpis
        textBoxVystup.Text = null;
        for (int i = 0; i < 10; i++)
            textBoxVystup.Text += posl[i].ToString() + Environment.NewLine;
    }
}
}

```

Poznámka: amozřejmě, jak aritmetickou posloupnost sudých čísel, tak geometrickou posloupnost mocnin 2 můžete vytvářet rekurentně, např:

```

    posl[0]=2;
    for (int i = 1; i < posl.Length; i++)
    {
        posl[i]= posl[i-1]+2;
    }

```