

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-211
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Aplikace posloupností
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
Anotace	Studenti spojují své znalosti z matematiky a programování a zkoumají vlastnosti posloupností
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: učebnice pro začátečníky</i>. Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7.</p>

<p>Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.</p>	<p>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...).</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.

211. Opakování – aplikační úloha: práce s posloupnostmi

Pracovní list

Cvičení

Informace do paměti počítače ukládáme pomocí proměnných.

Co je to proměnná?

Jak systém ví, jak má s proměnnými pracovat (například když mezi ně vložíme znaménko +)

Kde jste se setkali s typem objektů kromě programování?

Jak byste deklarovali celočíselnou proměnnou x , reálnou y a řetězcovou z ? Jak se říká názvům proměnných x , y , z .

Jaká bude hodnota celočíselné proměnné x po provedení následujících příkazů?

1.

`Y=Math.PI;`

`If (y==22/7)`

`X = 10;`

Else

X = 0;

2.

X = 0;

For (int p=1; p<10; p++)

X=p;

3.

X = 0;

For (int p=1; p<10; p++)

X+=p;

Vysvětlete vlastními slovy: (případně uveďte příklady)

Co je to v matematice posloupnost?

Co znamená zadat posloupnost rekurentně?

Co je to aritmetická posloupnost? Jak vypadá zadání vzorcem pro n-tý člen?

Co je to geometrická posloupnost? Jak vypadá zadání vzorcem pro n-tý člen?

Jak byste počítali součet prvních n členů libovolné posloupnosti?

4. Pracujte s posloupnostmi podle obrázku: (Vypracujte podobný vlastní projekt)

Posloupnosti

1. člen: 1

Kvocient či diference: 2

Počet členů: 10

Od n-tého do prvního

Prvních n členů a součet AP

Prvních n členů a součet GP

Zadáno vzorcem $a_n = (n-1) \cdot n$

Rekurentně $a_1 = 1. a_n = a_{n-1} \cdot n$

512
256
128
64
32
16
8
4
2
1
součet: 1023,00

Tedy:

Naprogramujte generování prvních n členů AP a GP, když uživatel zadá 1. člen, počet (n) a diferenci nebo kvocient. Členy posloupnosti zobrazte a vypočítejte jejich součet. (Průběžným sčítáním i vzorci)

*Vypište prvních n členů daných vzorcem pro n -tý člen

$$a_n = (n-1)n$$

rekurentně.

$$a_1 = 1; a_n = n a_{n-1}$$

Řešení

Co je to proměnná?

Pojmenované místo v paměti počítače, kde je uložena její hodnota.

Jak systém ví, jak má s proměnnými pracovat (například když mezi ně vložíme znaménko +)

Každý objekt je nutné před použitím deklarovat.

Kde jste se setkali s typem objektů kromě programování?

Např. Excel, Access

Jak byste deklarovali celočíselnou proměnnou x , reálnou y a řetězcovou z ? Jak se říká názvům proměnných x , y , z .

`int x, double y, string z.`

Jméno proměnné = identifikátor

Jaká bude hodnota celočíselné proměnné x po provedení následujících příkazů?

1.

```
Y=Math.PI;
```

```
If (y==22/7)
```

```
X = 10;
```

```
Else
```

```
X = 0;
```

0

2.

```
X = 0;
```

For (int p=1; p<10; p++)

X=p;

9

3.

X = 0;

For (int p=1; p<10; p++)

X+=p;

45

Vysvětlete vlastními slovy: (případně uveďte příklady)

Co je to v matematice posloupnost?

Co znamená zadat posloupnost rekurentně?

Co je to aritmetická posloupnost? Jak vypadá zadání vzorcem pro n-tý člen?

Co je to geometrická posloupnost? Jak vypadá zadání vzorcem pro n-tý člen?

Jak byste počítali součet prvních n členů libovolné posloupnosti?

Posloupnost je řada čísel, každé má své pořadové číslo.

V aritmetické posloupnosti se sousední členy liší o konstantu – diference.

$$a_n = a_1 + (n-1)d$$

V geometrické posloupnosti je podíl sousedních členů konstantní – kvocient.

$$a_n = a_1 q^{n-1}$$

Zadáme postup jak vypočítat další člen pomocí předchozích.

(např. $a_1 = 3$; $a_n = a_{n-1} + d$ – AP; diference d)

(např. $a_1 = 3$; $a_n = a_{n-1} \cdot q$ – GP; kvocient q)

Postupným sčítáním, pro AP a GP jsou vzorce

$$\text{Součet prvních } n \text{ členů AP: } s_n = \frac{n}{2}(a_1 + a_n)$$

$$\text{Součet prvních } n \text{ členů GP: } s_n = a_1 \frac{q^n - 1}{q - 1}$$

Zdrojové kódy

```
namespace Posloupnosti
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void buttonAP_Click(object sender, EventArgs e)
        {
            //prvních n členů a součet AP
            double a1 = Convert.ToDouble(textBoxA1.Text);
            double an; //n-tý člen
            double d = Convert.ToDouble(textBoxDQ.Text);
            int n = Convert.ToInt32(textBoxN.Text);
            double sum = 0;
            textBoxVystup.Text = null;
            if (!checkBoxZZ.Checked)
            {
                an = a1;
                for (int i = 0; i < n; i++)
                {
                    sum += an;
                    textBoxVystup.Text += an.ToString() + Environment.NewLine;
                    an += d;
                }
                an = an - d;
            }
            else
            {
                an = a1 + (n - 1) * d;
                //schovávám si poslední člen -
                for (int i = n; i > 0; i--)
                {
                    sum += an;
                    textBoxVystup.Text += an.ToString() + Environment.NewLine;
                    an -= d;
                }
            }
        }
    }
}
```

```

        }
an = an + d;
    }
textBoxVystup.Text += "součet: " + sum.ToString("F2");
//jinak nepočítat sum průběžně, ale podle vzorce

    sum = (an + a1) * n / 2;
MessageBox.Show("podle vzorečku: " +sum.ToString("F2"));
    }

privatevoidbuttonGP_Click(objectsender, EventArgs e)
    {
        //prvních n členů a součet GP
double a1 = Convert.ToDouble(textBoxA1.Text);
doublean;//n-tý člen
double q = Convert.ToDouble(textBoxDQ.Text);
int n = Convert.ToInt32(textBoxN.Text);
double sum = 0;
textBoxVystup.Text = null;
if (!checkBoxZZ.Checked)
    {
        an = a1;
for (int i = 0; i < n; i++)
    {
        sum += an;
textBoxVystup.Text += an.ToString() + Environment.NewLine;
an *= q;
    }
an = an /q;
    }
else
    {

an = a1 * Math.Pow(q,n-1);
doubleann = an;//schovávám si poslední člen
for (int i = n; i > 0; i--)
    {
        sum += an;
textBoxVystup.Text += an.ToString() + Environment.NewLine;
an /= q;
    }
an = an *q;
    }
}

```

```

textBoxVystup.Text += "součet: " + sum.ToString("F2");
//jinak nepočítat sum průběžně, ale podle vzorce

        sum = a1 * (Math.Pow(q,n)-1)/(q-1);
MessageBox.Show("podle vzorečku: " +sum.ToString("F2"));
    }

privatevoidbuttonPN_Click(objectsender, EventArgs e)
    {
        //posloupnost vzorcem pro n-tý člen
textBoxVystup.Text = null;
int n = Convert.ToInt32(textBoxN.Text);
int sum = 0;
for (int i = 0; i < n; i++)
    {
intai = i * (i - 1);
        sum += ai;
textBoxVystup.Text += ai.ToString() + Environment.NewLine;
    }
textBoxVystup.Text += "součet: " + sum.ToString("F2");
    }

privatevoidbuttonRekur_Click(objectsender, EventArgs e)
    {
        //rekurentně zadaná posloupnost
int a = 1;
textBoxVystup.Text = null;
int n = Convert.ToInt32(textBoxN.Text);
int sum = 0;
for (int i = 1; i < n+1; i++)
    {
intai = a*i;
        a = ai;
        sum += ai;
textBoxVystup.Text += ai.ToString() + Environment.NewLine;
    }
textBoxVystup.Text += "součet: " + sum.ToString("F2");
    }
}

```