

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-313
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Zlomek jako vlastní třída s metodami
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 17–18 let
Úroveň žáků	Středně pokročilí
Časový rozsah	1 vyučovací hodina
Klíčová slova	Zlomek, třída, objekt, metoda
Anotace	Studenti programují třídu zlomek s metodami pro práci s touto třídou
Použité zdroje	<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>OCHRANOVÁ, Renata a Michal KOZUBEK. <i>Objektově orientované programování v Turbo Pascalu</i>. 1. vyd. Brno: Masarykova univerzita, 1993, 117 s. ISBN 80-210-0659-5.</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 pokročilé</i>. Ondřejov: moderníProgramování, 2011, 149 s. ISBN 978-80-903951-7-6.</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.	Příklad je určen k samostatnému procvičování programování vlastních tříd a jejich metod. Je možné nechat nejlepší studenty pracovat zcela samostatně a s ostatními alespoň zavést třídu zlomek a naprogramovat ukázkovou metodu. Není nutné, aby všichni zpracovali

	všechno, vhodné je diferencovat podle jejich zájmu a schopností. 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.

313. Zlomek jako vlastní třída s metodami

Pracovní list

Cvičení

Definujte třídu Zlomek s celočíselnými proměnnými cit a jmen. (čitatel, jmenovatel). Doplňte těla následujících metod.

Můžete nadefinovat parametrický i bezparametrický konstruktor nebo nechat vytvoření bezparametrického konstruktoru na systému.

Vyzkoušejte svou třídu v programu, kde uživatel bude zadávat čitatele a jmenovatele zlomku podle vzoru.

```
class Zlomek
{
    public int cit;
    public int jmen;
    public Zlomek()
    {
        //Bezparametrický konstruktor
    }
    public Zlomek(int c, int j)
    {
```

```

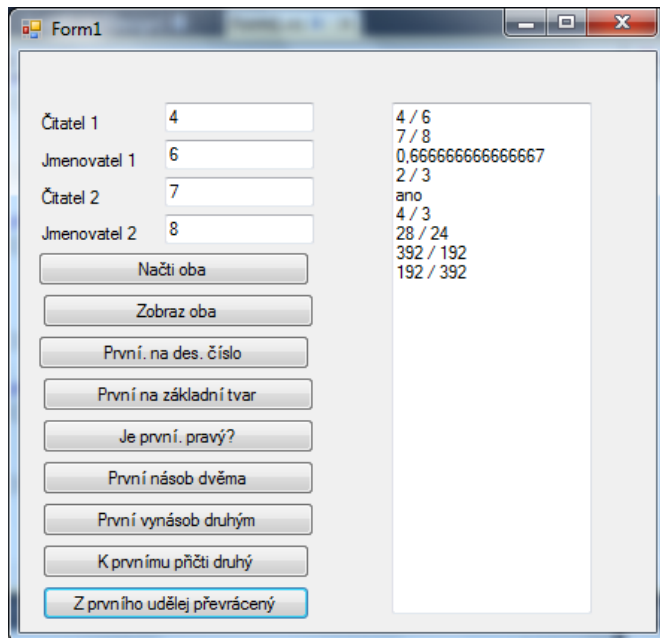
        //Konstruktor s inicializací
        cit = c; jmen = j;
    }
    public string NaRet()
    {
        //Převede zlomek na řetězec ve tvaru čítelel/jmenovatel
    }
    public bool Pravy()
    {
        //vrací True je-li čítelel menší než jmenovatel, jinak False
    }
    public double NaDes()
    {
        //převádí zlomek na desetinné číslo
    }
    public void nasob(int k)
    {
        //vynásobí zlomek celým číslem k
    }
    public void PrictiZlomek(Zlomek a)
    {
        //přičte ke zlomku zlomek a
    }
    public void NasobZlomkem(Zlomek b)
    {
        //vynásobí zlomek zlomkem b
    }
    public void ZakTvar()
    {
        //Zkrátí zlomek na základní tvar
    }

```

```

public void Prevrat()
{
    //Převrátí zlomek - vymění čitatele a jmenovatele
}

```



Řešení

```

class Zlomek

```

```

{
    public int cit;
    public int jmen;
    public Zlomek()
    {
    }
    public Zlomek(int c,int j)
    {
        cit = c; jmen = j;
    }
    public string NaRet()
    {
        //Převede zlomek na řetězec ve tvaru čítatel/jmenovatel
        return cit.ToString() + " / " + jmen.ToString();
    }
}

```

```

public bool Pravy()
{
    //vrací True je-li čitatel menší než jmenovatel, jinak False
    return (Math.Abs(cit) < Math.Abs(jmen));
}

public double NaDes()
{
    //převádí zlomek na desetinné číslo
    return Convert.ToDouble(cit)/Convert.ToDouble(jmen);
}

public void nasob(int k)
{
    //vynásobí zlomek celým číslem k
    cit*=k;
}

public void PrictiZlomek(Zlomek a)
{
    //přičte ke zlomku zlomek a
    int pomcit = a.jmen * this.cit + this.jmen * a.cit;
    int pomjmen = a.jmen * this.jmen;
    //při přímém přiřazení se změní hodnoty
    this.cit = pomcit;
    this.jmen = pomjmen;
}

public void NasobZlomkem(Zlomek b)
{
    //vynásobí zlomek zlomkem b
    this.jmen = this.jmen * b.jmen;
    this.cit = b.cit *this.cit;
}

```

```

public void ZakTvar()
{
    //Zkrátí zlomek na základní tvar
    int nsd;//nejmenší společný dělitel čitatele a jmenovatele
    int c, j;//aby se neměnily výchozí hodnoty
    c = cit;
    j = jmen;
    while (c != j)//Euklidův algoritmus
    {
        if (c > j) c = c - j;
        else
            j = j - c;
    }
    nsd=c;
    cit/=nsd;
    jmen/=nsd;
}

public void Prevrat()
{
    //Převrátí zlomek - vymění čitatele a jmenovatele
    int pom;//pro výměnu
    pom = cit;
    cit = jmen;
    jmen = pom;
}
namespace Zlomky
{
    public partial class Form1 : Form
    {
        Zlomek X, Y;

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

```

```

private void buttonNacti_Click(object sender, EventArgs e)
{
    try
    {
        X = new Zlomek();
        X.cit = Convert.ToInt32(textBoxC1.Text);
        X.jmen = Convert.ToInt32(textBoxJ1.Text);
        Y = new Zlomek(Convert.ToInt32(textBoxC2.Text),
            Convert.ToInt32(textBoxJ2.Text));
    }
    catch
    {
        MessageBox.Show("Vstup celá čísla");
    }
}

private void buttonNaText_Click(object sender, EventArgs e)
{
    textBoxVys.Text+=X.NaRet()+Environment.NewLine;
    textBoxVys.Text += Y.NaRet() + Environment.NewLine;
}

private void buttonNaDes_Click(object sender, EventArgs e)
{
    textBoxVys.Text += X.NaDes().ToString() + Environment.NewLine; ;
}

private void buttonZakTvar_Click(object sender, EventArgs e)
{
    X.ZakTvar();
    textBoxVys.Text += X.NaRet() + Environment.NewLine;
}

```

```

}

private void buttonPrav_Click(object sender, EventArgs e)
{
    if (X.Pravy())
        textBoxVys.Text += "ano" + Environment.NewLine;
    else
        textBoxVys.Text += "ne" + Environment.NewLine;
}

private void buttonkrat2_Click(object sender, EventArgs e)
{
    X.nasob(2);
    textBoxVys.Text += X.NaRet() + Environment.NewLine;
}

private void buttonNasob_Click(object sender, EventArgs e)
{
    X.NasobZlomkem(Y);
    textBoxVys.Text += X.NaRet() + Environment.NewLine;
}

private void buttonSum_Click(object sender, EventArgs e)
{
    X.PrictiZlomkem(Y);
    textBoxVys.Text += X.NaRet() + Environment.NewLine;
}

private void buttonOtoc_Click(object sender, EventArgs e)
{
    X.Prevrat();
    textBoxVys.Text += X.NaRet() + Environment.NewLine;
}
}
}

```