Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-320
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech III
Téma didaktického materiálu	Opakovací cvičení k objektovému programování
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 17–18 let
Úroveň žáků	Středně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Třídy, metoda, vlastnost, virtuální a předefinované metody, grafika
Anotace	Studenti spojují své znalosti objektů a grafiky k tvorbě většího systému grafických objektů.
Použité zdroje	OCHRANOVÁ, Renata a Michal KOZUBEK. *Objektově orientované programování v Turbo Pascalu*. 1. vyd. Brno: Masarykova univerzita, 1993, 117 s. ISBN 80-210-0659-5. VYSTAVĚL, Radek. *Moderní programování: sbírka úloh k učebnici pro středně pokročilé*. 1. vyd. Ondřejov: moderníProgramování, 2008-2009, 2 sv. ISBN 978-80-903951-3-8. VYSTAVĚL, Radek. *Moderní programování: sbírka úloh k učebnici pro začátečníky*. 2. vyd. Ondřejov: moderníProgramování, 2008, 2 sv. ISBN 978-80-903951-5-2. VYSTAVĚL, Radek. *Moderní programování: učebnice pro pokročilé*. Ondřejov: moderníProgramování, 2011, 149 s. ISBN 978-80-903951-7-6. VYSTAVĚL, Radek. *Moderní programování: učebnice pro středně pokročilé*. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1. VYSTAVĚL, Radek. *Moderní programování: učebnice pro začátečníky*. 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.	Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod.

	Při společné práci je vhodné nejprve obtížnější úlohy rozebrat, potom společně se studenty implementovat na počítači. (Rozbor nejlépe na tabuli, synchronní řešení s promítáním) 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. 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.

320. Opakovací cvičení

Pro začátek si znovu nadefinujeme třídu Kruh – ale tentokrát bude určená pro kreslení. Otestujeme si ji v programu, kde při stisknutí levého tlačítka myši nakreslíme kružnici na pozici kurzoru, při současném stisknutí klávesy Shift a kliknutí do libovolného kruhu, se daný kruh zvětší. Při stisknutí tlačítka Tlustá červená se všechny kružnice zbarví červeně a tloušťka kreslicího pera bude 10.

Vzhledem k požadavkům úlohy bude třída Kruh mít vlastnosti střed, poloměr a obrys (Pero). Připojíme konstruktor s inicializací hodnot, metodu pro kreslení a pro zvětšení poloměru. Parametrem metody kreslení bude kreslicí plocha.

V programu pak budeme pracovat se seznamem Kruhů, do kterého budeme podle požadavků uživatele přidávat další kruhy nebo ho opakovaně vykreslovat.

Abychom mohli určit, ve které kružnici je kurzor, nadefinujeme logickou funkci JeTam s parametrem bod. Pokud je vzdálenost bodu od středu menší nebo rovna poloměru, bod se v kruhu nachází.

Abychom mohli pracovat s grafikou, je nutné do klauzulí using přidat: `using System.Drawing;`

```
class Kruh
    {
        Point s; //střed
```

```
        public Point S
        {
            get
            {
                return s;
            }
            set
            {
                s = value;
            }
        }
            Pen obrys;//obrysové pero
            public Pen Obrys
            {
                get
                {
                    return obrys;
                }
                set
                {
                    obrys = value;
                }

            }
            int r; //poloměr
            public int R
            {
                get
                {
                    return r;
                }
                set
                {
                    r = value;
                }

            }
public Kruh(Point stred, Pen pero,int polomer )
         {
            S = stred;
            Obrys = pero;
            R = polomer;
         }


public virtual void Kresli(Graphics kp)//kreslení kruhu
            {
                kp.DrawEllipse(Obrys, S.X - R, S.Y - R, 2 * R, 2 * R);
            }
            public virtual bool JeTam(Point B)//vrací true, je-li bod B uvnitř kruhu,
jinak false
            {
                double d=Math.Sqrt((B.X-S.X)*(B.X-S.X)+(B.Y-S.Y)*(B.Y-S.Y));
```

```
            return (d<=R);
        }
        public void zvetsi(int d)//zvětšení poloměru
        {
            R += d;
        }
```

Metody pro kreslení a zjišťování bodu v kruhu je vhodné nadeklarovat jako virtuální, aby je potomci kruhu, které budete vyrábět ve cvičení, mohly vhodně zdědit.

Použití třídy v programu:

```
public partial class Form1 : Form
    {
       List<Kruh> Kola=new List<Kruh>();//seznam kreslených objektů
       Random nahoda = new Random();
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
           //při stisknutí levého tlačítka myši do seznamu přibude kruh,
            //při současném stisknutí klávesy shift se vybraný objekt zvětší
            Pen pero1=new Pen(Color.Green,5);
            Point stred = new Point(e.X, e.Y);
            int r = 50;
            if (Control.ModifierKeys == Keys.Shift)// stisknuto Shift
                foreach (Kruh k in Kola)
                {
                    if (k.JeTam(stred))
                    {
                        k.zvetsi(10);
                        Refresh();
                    }
                }
            else
            {
                if (e.Button == MouseButtons.Left)
                {
                    Kruh k = new Kruh(stred, pero1, r);
                    Kola.Add(k);
```

```
                Refresh();
            }
        }
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        //seznam kreslených objektů se vykresluje metodou paint
        Graphics plocha = e.Graphics;
        foreach (Kruh k in Kola)
            k.Kresli(plocha);
    }
private void buttonTC_Click(object sender, EventArgs e)
    {
        //abychom mohli měnit obrys prvků seznamu, je třeba
        //ho zpracovávat for cyklem
        for (int i = 0; i < Kola.Count; i++)
        {
                        Pen pero = new Pen(Color.Red, 10);
            Kola[i].Obrys = pero;
        }
        Refresh();
    }
```

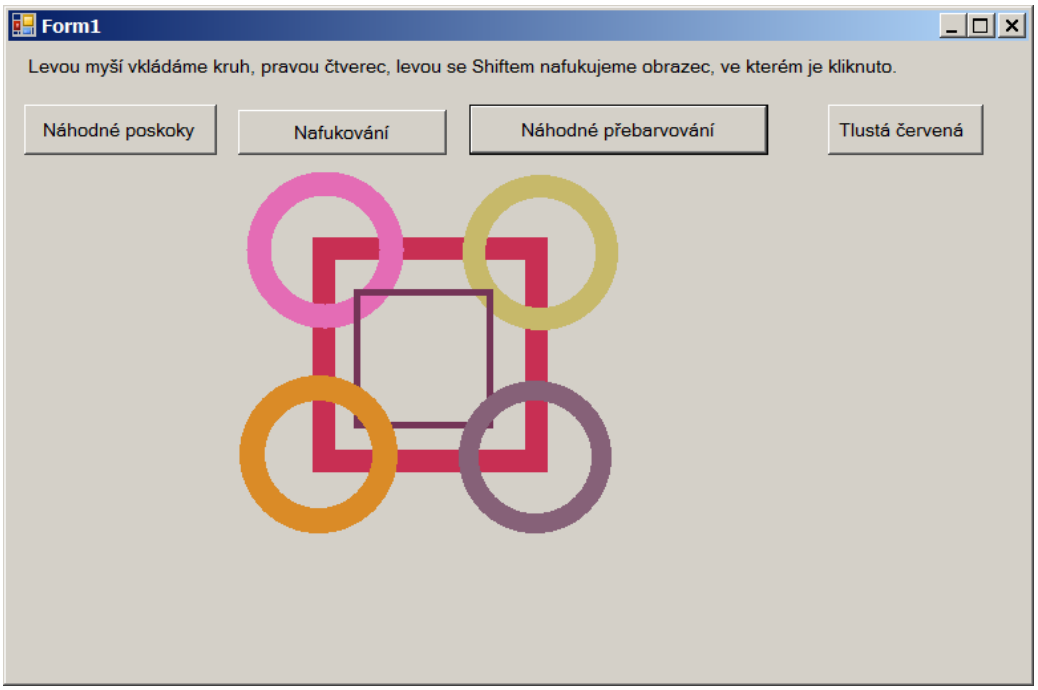
Pracovní list

Cvičení

Přidejte kruhu další vlastnosti a metody. (Barvu výplně, posunutí o daný vektor apod.) Nadefinujte další třídy odvozené od Kruhu (čtverec, obdélník, elipsu…) Nastavte v programu kresleným objektům nějakou jednoduchou animaci…

Řešení

Individuální, pro začátek a inspiraci následující rozšíření úlohy ze společné práce:




```
class Ctverec:Kruh //potomek kruhu, liší se jen metodami
    {
        public override void Kresli(Graphics kp)//kreslení čtverce
        {
            kp.DrawRectangle(Obrys, S.X - R, S.Y - R, 2 * R, 2 * R);
        }
        public override bool JeTam(Point B)//zjištění, zda se bod nachází ve čverci
        {

            return ((S.X - R <= B.X) && (S.X + R >= B.X) && (S.Y - R <= B.Y) && (S.Y +
R >= B.Y));
        }
         public Ctverec(Point stred, Pen cara, int r):base(stred,cara,r)//konstruktor
            {

            }
    }

namespace objektGrafik
{
    public partial class Form1 : Form
    {
       List<Kruh> Kola=new List<Kruh>();//seznam kreslených objektů
       Random nahoda = new Random();
        public Form1()
        {
```

```
    InitializeComponent();
}
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
   //při stisknutí levého tlačítka myši do seznamu přibude kruh,
    //při stisknutí pravého tlačítka čtverec.
    //při současném stisknutí klávesy shift se vybraný objekt zvětší
    Pen pero1=new Pen(Color.Green,5);
    Pen pero2=new Pen(Color.Blue,5);
    Point stred = new Point(e.X, e.Y);
    int r = 50;
    if (Control.ModifierKeys == Keys.Shift)// stisknuto Shift
        foreach (Kruh k in Kola)
        {
            if (k.JeTam(stred))
            {
                k.zvetsi(10);
                Refresh();
            }

        }
    else
    {
        if (e.Button == MouseButtons.Left)
        {
            Kruh k = new Kruh(stred, pero1, r);
            Kola.Add(k);
            Refresh();
        }
        else
        {
            Ctverec c = new Ctverec(stred, pero2, r);
            Kola.Add(c);
            Refresh();
        }
    }
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    //seznam kreslených objektů se vykresluje metodou paint
    Graphics plocha = e.Graphics;
```

```
    foreach (Kruh k in Kola)
        k.Kresli(plocha);
}

private void buttonHop_Click(object sender, EventArgs e)
{
    timer1.Enabled = !timer1.Enabled;
}

private void timer1_Tick(object sender, EventArgs e)
{
    int x=nahoda.Next(20)-10;
    int y=nahoda.Next(20)-10;
    Point A=new Point(x,y);
    foreach (Kruh k in Kola)
        k.posun(A);
    Refresh();
 }

private void buttonNafuka_Click(object sender, EventArgs e)
{
    int d = nahoda.Next(20);
    foreach (Kruh k in Kola)
        k.zvetsi(d);
    Refresh();
    }

private void timer2_Tick(object sender, EventArgs e)
{
    //abychom mohli měnit obrys prvků seznamu, je třeba
    //ho zpracovávat for cyklem
    for (int i=0;i<Kola.Count;i++)
    {
        int r = nahoda.Next(255);
        int g = nahoda.Next(255);
        int b = nahoda.Next(255);
        int d=nahoda.Next(1,20);
        Color barva = Color.FromArgb(r, g, b);
        Pen pero=new Pen (barva,d);
        Kola[i].Obrys = pero;
    }
    Refresh();
```

```
            }

        private void button1_Click(object sender, EventArgs e)
        {
            timer2.Enabled = !timer2.Enabled;
        }

        private void buttonTC_Click(object sender, EventArgs e)
        {
            //abychom mohli měnit obrys prvků seznamu, je třeba
            //ho zpracovávat for cyklem
            for (int i = 0; i < Kola.Count; i++)
            {

                Pen pero = new Pen(Color.Red, 10);
                Kola[i].Obrys = pero;
            }
            Refresh();
        }
    }
}
```