

Metodický list k didaktickému materiálu

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-203
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Cykly v grafice
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	Mírně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Cyklus, grafika
Anotace	Studenti spojují a prohlubují poznatky z grafiky a programování cyklů k opakovanému vykreslování útvarů na určené pozice.
Použité zdroje	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. 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. VYSTAVĚL, Radek. <i>Moderní programování: 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.	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...).

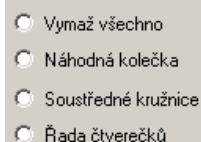
Prohlášení autora

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.

203. Cykly v grafice

Cykly se nám budou hodit při vykreslování více podobných obrazců, ať už na pozici náhodné nebo nějak pravidelné.



Kreslení jednoho z následujících příkladů si vybereme zaškrtnutím políčka RadioButton. Větvení se bude odehrávat v události **Form1.Paint**, událost **CheckedChanged** všech políček bude mít společnou obsluhu **radioButton_Prekresli**.

Políčko **Vymaž všechno** se zaškrtně při vytvoření formuláře.

Příklad 1.

Nakreslete na pozadí okna 500 náhodných červených koleček o průměru 6 px.

V cyklu for budeme pokaždé generovat náhodné souřadnice, tak, aby se kolečko vešlo do formuláře a vykreslovat kružnici. Vlastní kreslení bude samozřejmě v události Paint formuláře.

Příklad 2.

Nakreslete řadu soustředných kružnic, střed bude ve středu formuláře, poloměry se budou lišit o 5 px.

Budeme tedy kreslit kružnice se společným středem, `kp.DrawEllipse(Pero, x - poloměr, y - poloměr, průměr, průměr)` a poloměr se bude při každém průchodu cyklem zvětšovat: `kp.DrawEllipse(Pens.Blue, x - 5 * i, y - 5 * i, 10 * i * i);`

Příklad 3.

Nakreslete řadu čtverečků vedle sebe. Čtverečky budou mít rozměry 10*10 px a jejich středy budou vzdáleny 20 px.

Polohu čtverečků tentokrát budeme při průchodu cyklem posouvat o konstantní hodnotu.

```
kp.DrawRectangle(Pero, x - 5, y - 5, 10, 10);  
  
x += posun;  
  
namespace Cykly_v_grafice  
{  
  
    public partial class Form1 : Form
```

```

{
    Random Nahoda = new Random();

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Graphics kp = e.Graphics;

        int maxX = ClientSize.Width; //šířka formuláře
        int maxY = ClientSize.Height; //výška formuláře

        if (radioButtonKolecka.Checked)
        {
            for (int i = 0; i < 500; i++)
            {
                int x = Nahoda.Next(2, maxX - 2);

                int y = Nahoda.Next(2, maxY - 2); //střed kolečka

                kp.DrawEllipse(Pens.Red, x - 2, y - 2, 4, 4);
            }
        }

        if (radioButtonSoustredy.Checked)
        {
            int x = maxX / 2;

            int y = maxY / 2; //střed

            for (int i = 0; i < 30; i++)
            {
                kp.DrawEllipse(Pens.Blue, x - 5 * i, y - 5 * i, 10 * i * i);
            }
        }

        if (radioButtonRadaCtvercu.Checked)
        {

```

```

int posun = 20;
//každý další čtvereček bude posunutý o 20 px doprava

int x = 10; //x a y středu prvního čtverečku

int y = maxY / 2;

for (int i = 0; i < 30; i++)
{
    kp.DrawRectangle(Pens.Green, x - 5, y - 5 , 10 , 10) ;

    x += posun;
}

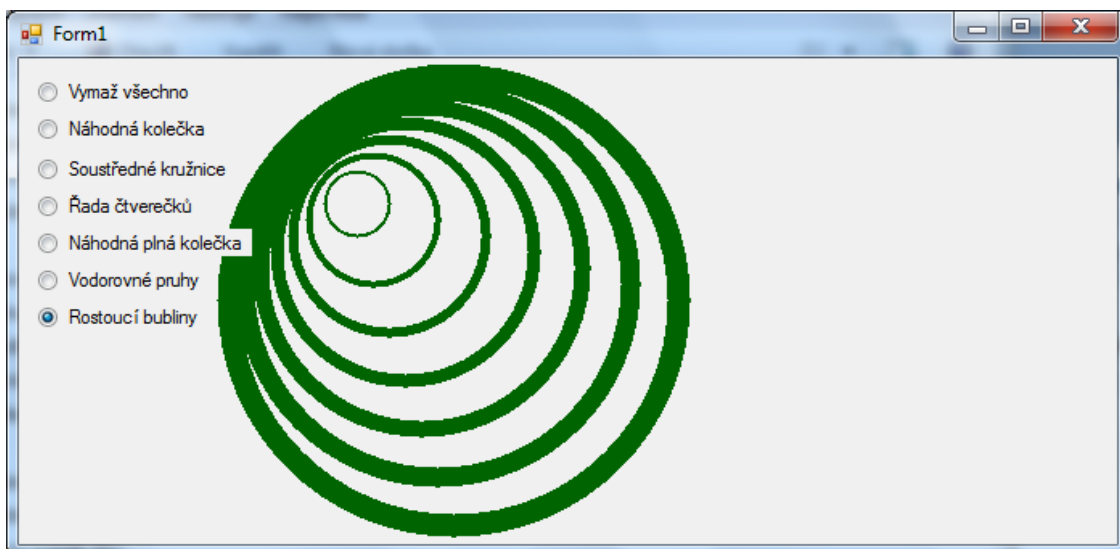
private void radioButton_Prekresli(object sender, EventArgs e)
{
    Refresh();
}

private void Form1_Load(object sender, EventArgs e)
{
    radioButtonVymaz.Checked=true;
}
}
}

```

Pracovní list

Cvičení



1.

Modifikujte příklad 1 tak, že rozměry koleček budou rovněž náhodné a kolečka budou plná.

2.

Nakreslete řadu vodorovných čar přes celý formulář, tak, aby se jich na formulář vešlo 30. Definujte pero s tloušťkou 3 px libovolné barvy.

3.

Vymyslete si vlastní kreslení řady pravidelných obrazců.

Řešení

1.

```
if (radioButtonPlna.Checked)
{
    for (int i = 0; i < 100; i++)
    {
        int x = Nahoda.Next(2, maxX - 2);
        int y = Nahoda.Next(2, maxY - 2); //střed kolečka
        int r = Nahoda.Next(1, maxY / 10); //poloměr kolečka
        kp.FillEllipse(Brushes.Aqua, x - r, y - r, 2*r, 2*r);
    }
}
}
```

2.

```
if (radioButtonPruhy.Checked)
{
    Pen pero = new Pen(Color.DeepSkyBlue,3);
    for (int i = 1; i < 31; i++)
    {
        int y = maxY / 30; //třicetina výšky okna se bude násobit i
        kp.DrawLine(pero,0,y*i,maxX,y*i);
    }
}
```

3. Individuální řešení

Příklad pro inspiraci:

```
if (radioButtonBubl.Checked) ;//rostoucí nesoustředné bubliny
{
    Pen pero = new Pen(Color.DarkGreen, 1);
    for (int i = 1; i < 8; i++)
    {
        int r = 20 * i;
        int x = 200+10 * i;
        int y = 80+10 * i;
        pero.Width =2* i;
        kp.DrawEllipse(pero, x-r, y-r, 2*r, 2*r);
    }
}
```