

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-113
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Grafika II
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	začátečníci
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Canvas, Panel, Refresh, společná obsluha událostí
Anotace	Studenti si procvičují práci s kreslicím plátnem a jeho metodami, určování souřadnic objektů, programují společnou obsluhu událostí.
Použité zdroje	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) Prezentace obsahuje stručné shrnutí poznatků potřebných pro řešení příkladů. 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.

113. Grafika II

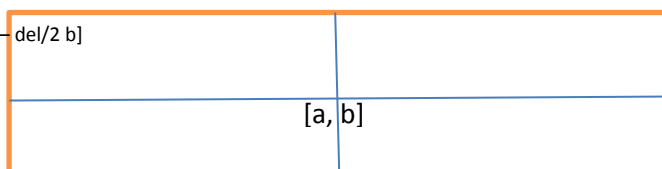
Komponenta Panel

je používána jednak jako kontejner jiných komponent nebo se také hodí pro kreslení, pokud se kreslená grafika na panel nevejde, bude oříznuta. Souřadný systém bude spojen s panelem. (jeho levý horní roh má souřadnice [0,0]) Nalezneme ho v ToolBoxu ve skupině Containers.

Příklad 1

Na formulář umístíme panel, na který budeme kreslit a čtyři textová políčka pro souřadnice středu

[a – sir/2, b – del/2 b]



elipsy, její šířku a výšku. Kreslit budeme tentokrát elipsu podle přání uživatele.

Kreslení elipsy (obdélníka) pokud známe souřadnice středu $S[a, b]$ a rozměry sir, del:

`DrawEllipse(Pero, a – sir/2, b – del/2, sir, del);`

Samo kreslení bude v události Paint panelu, ale o její vyvolání se tentokrát budeme muset postarat – použijeme jeho metodu **panel.Refresh()**

```
private void buttonKresli_Click(object sender, EventArgs e)
```

```
{
    panel1.Refresh();
}
```

```
private void panel1_Paint(object sender, PaintEventArgs e)
```

```
{
    try
    {
        int XS, YS, sirka, vyska;
        XS = Convert.ToInt32(textBoxXS.Text);
```

```

YS = Convert.ToInt32(textBoxYS.Text);

sirka = Convert.ToInt32(textBoxSirka.Text);

vyska = Convert.ToInt32(textBoxVyska.Text);

Graphics kp = e.Graphics;

kp.DrawEllipse(Pens.DarkGreen, XS - sirka / 2, YS - vyska / 2,
sirka, vyska);
}

catch
{
    MessageBox.Show("Zadejte číselné údaje");
}
}

```

Pokud nechcete, aby se varovný MessageBox objevil při spuštění programu, zadejte do textBoxů výchozí hodnoty při návrhu programu.

Příklad 2

Doplníme předchozí příklad dvěma CheckBoxy, aby si uživatel mohl vybrat, zda chce kreslit elipsu nebo obdélník, případně obojí.

Úprava kódu události Paint panelu je jednoduchá:

```

if (checkBoxEl.Checked)

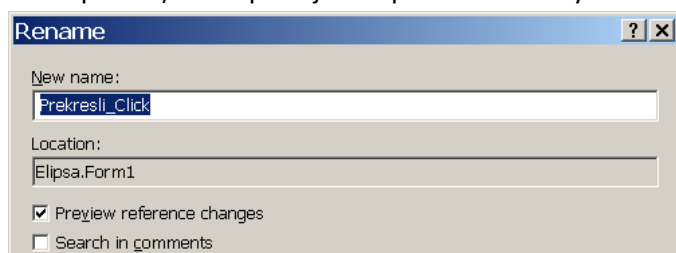
    kp.DrawEllipse(Pens.DarkGreen, XS - sirka / 2, YS - vyska / 2,
sirka, vyska);

if (checkBoxObd.Checked)

    kp.DrawRectangle(Pens.Red, XS - sirka / 2, YS - vyska / 2,
sirka, vyska);

```

Ovšem událost `panel1.Refresh()` by se teď měla objevit při zaškrtnutí CheckBoxů – tedy při jejich události **CheckedChanged**. (eventuelně Click) – je to tedy stejný kód pro tři události. Není ideální stejnou věc programovat víckrát (ani když to jde snadno zkopírovat), protože takový program se pak těžko udržuje. (Případná oprava se musí dělat na mnoha místech, což je nepřehledné a nebezpečné.) Proto použijeme společné obsluhy události – v Editoru vlastností a Událostí vybereme



CheckedChanged zaškrtačovacího políčka, rozbalíme šipku a vybereme událost `ButtonKresli.Click`. To provedeme i pro druhý checkBox. Teď by ovšem bylo dobré tuto událost přejmenovat – to nikdy

nedělejte ručně (podobně jako při mazání, mohly by se narušit vazby mezi objekty.)

Přejmenování události: Na jméno metody, kterou chceme přejmenovat, klepneme pravým tlačítkem myši a z místní nabídky vybereme Recator/Rename a v následujícím okně zdáme New Name – v našem případě nejspíš Prekresli_Click.

Důležité

Panel – se používá jako kontejner dalších komponent, má také kreslicí plochu.

Refresh() – je metoda objektů jako panel nebo formulář, vynutí jejich překreslení.

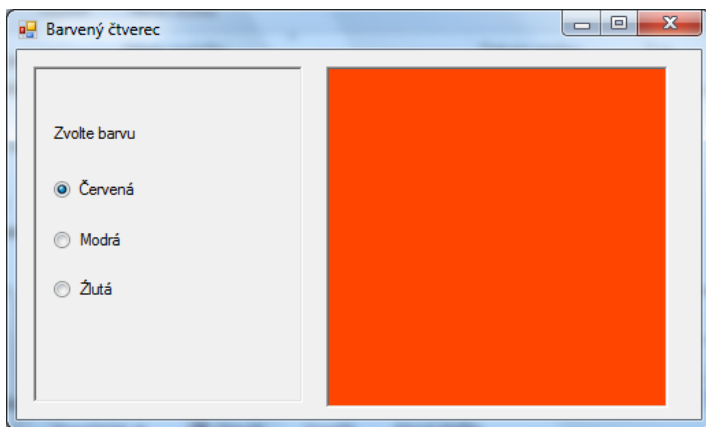
Přejmenování události: Na jméno metody, kterou chceme přejmenovat, klepneme pravým tlačítkem myši a z místní nabídky vybereme Recator/Rename

CheckChanged – je událost, která nastává při změně stavu zaškrtačacího políčka nebo radiového tlačítka.

Pracovní list

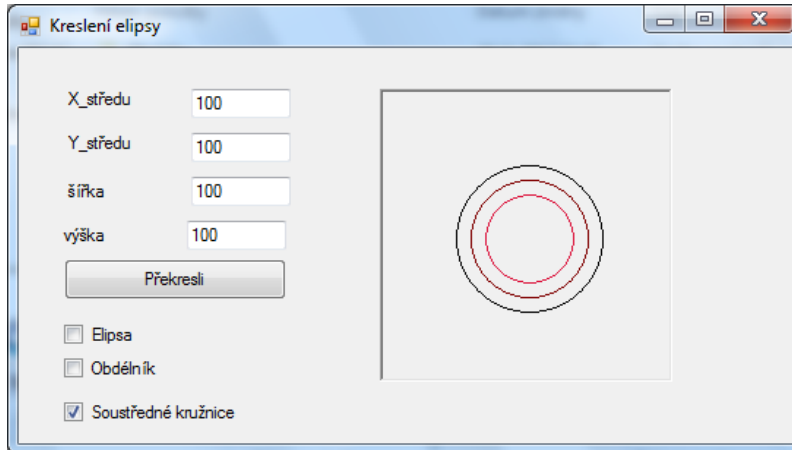
Cvičení

1. Připravte si formulář se dvěma panely – na jednom budou nejméně tři RadioButtony pro volbu barvy štětce, druhý (čtvercový) se bude vyplňovat zvolenou barvou. (FillRectangle). Pro volbu barvy využijte společné obsluhy události.

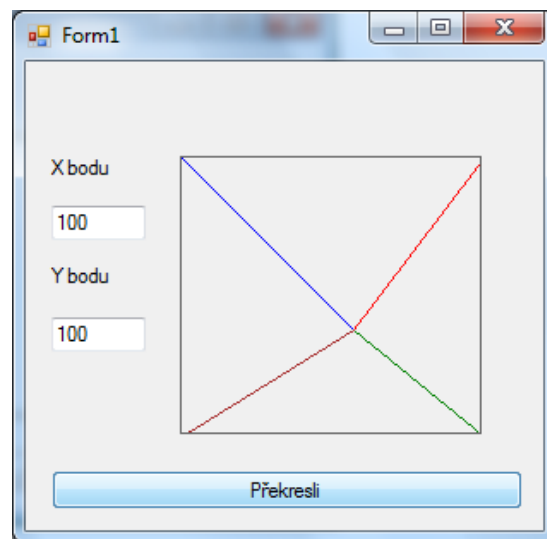


2. Doplňte příklad, kde si uživatel vybíral, zda chce kreslit obdélník nebo elipsu, zaškrtačacím políčkem Soustředné kružnice, které nakreslí tři soustředné kružnice se středem ve středu panelu. Pokud uživatel zadá různě šířku a výšku, upozorněte ho, že jako poloměr nastavíte

šířku.



3. Na formulář umístěte panel. Uživatel vždy zadá souřadnice bodu a po stisknutí tlačítka Překresli se nakreslí čáry z rohů panelu do tohoto bodu. Rozměry panelu pro kreslení: sirka = panel.Width
vyska = panel.Height;



Řešení

1.

```
private void Prekresli_Panel(object sender, EventArgs e)
{
    panelKresli.Refresh();
}

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

    if (radioButtonR.Checked)
        kp.FillRectangle(Brushes.OrangeRed, 0, 0, 250, 250);

    if (radioButtonB.Checked)
        kp.FillRectangle(Brushes.CornflowerBlue, 0, 0, 250, 250);

    if (radioButtonY.Checked)
        kp.FillRectangle(Brushes.Yellow, 0, 0, 250, 250);
}
```

2.

```
if (checkBoxSK.Checked)
{
    if (sirka!=vyska)
    {
        MessageBox.Show("pro kružnici šířka = výška, nastavuji šířku");
        vyska=sirka;
    }
    int r1 = sirka / 2;
    int r2 = r1 - 10;
    int r3 = r2 - 10;
    kp.DrawEllipse(Pens.Black, XS - r1, YS - r1 , 2*r1, 2*r1);
    kp.DrawEllipse(Pens.Maroon, XS - r2, YS - r2, 2*r2, 2*r2);
    kp.DrawEllipse(Pens.Crimson, XS - r3, YS - r3, 2*r3, 2*r3);
}
```

3.

```
private void buttonPrekresli_Click(object sender, EventArgs e)
{
    panel.Refresh();
}

private void panel_Paint(object sender, PaintEventArgs e)
{
    int x = Convert.ToInt32(textBoxX.Text);
    int y = Convert.ToInt32(textBoxY.Text);
    Graphics kp = e.Graphics;
    kp.DrawLine(Pens.Blue, 0, 0, x, y);
    kp.DrawLine(Pens.Red, panel.Width, 0, x, y);
    kp.DrawLine(Pens.Green, panel.Width, panel.Height, x, y);
    kp.DrawLine(Pens.Brown, 0, panel.Height, x, y);
}
```