

Zpravodaj moderní Programování 4/2013: Kreslení do obrázku

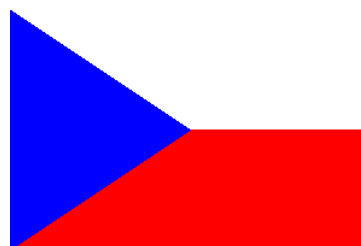
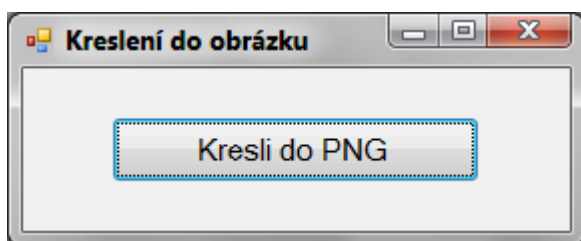
Obtížnost: začátečníci

V .NETu je pěkné, že podobné činnosti se programují podobně. V tomto Zpravodaji si ukážeme, jak všechno, co znáte z grafiky, z kreslení do okna programu, lze využít při kreslení do obrázku. Kreslení jako kreslení. Ukážeme si, jak programem vytvořit rastrový obrázek. To se může hodit např.:

- Když potřebuje nějaké pravidelné obrázky, které by se těžko dělaly ručně;
- Když potřebuje obrázek, jehož obsah je odvozen od vstupu uživatele;
- Když potřebuje náhodný obrázek (nejen při odlišování člověka od robota při různých přihlašováních).

Program

Vytvoříme program, který po stisku tlačítka **na disku** vytvoří rastrový obrázek s českou vlajkou:



Nejprve výpis, potom komentáře:

```
using System;
using System.Drawing;
using System.Windows.Forms;

using System.Drawing.Imaging; // přidáno

namespace Kreslení_do_obrázku
{
    public partial class oknoProgramu : Form
    {
        public oknoProgramu()
        {
            InitializeComponent();
        }
    }
}
```

```

private void tlačítkoKresliDoPng_Click(object sender, EventArgs e)
{
    // Zjistí jméno souboru
    if (oknoProVýběrSouboru.ShowDialog() == DialogResult.Cancel)
        return;
    string jménoSouboru = oknoProVýběrSouboru.FileName;

    // Připrav obrázek
    Bitmap obrázek = new Bitmap(300, 200, PixelFormat.Format24bppRgb);
    Graphics kp = Graphics.FromImage(obrázek);

    // Modrý klín
    kp.FillPolygon(Brushes.Blue,
        new Point[]
        {
            new Point(0, 0),
            new Point(150, 100),
            new Point(0, 200)
        });

    // Červený spodek
    kp.FillPolygon(Brushes.Red,
        new Point[]
        {
            new Point(0, 200),
            new Point(150, 100),
            new Point(300, 100),
            new Point(300, 200)
        });

    // Bílý vršek
    kp.FillPolygon(Brushes.White,
        new Point[]
        {
            new Point(0, 0),
            new Point(150, 100),
            new Point(300, 100),
            new Point(300, 0)
        });

    // Zapiš do souboru
    kp.Dispose();
    obrázek.Save(jménoSouboru, ImageFormat.Png);
    MessageBox.Show("Hotovo!");
}
}
}

```

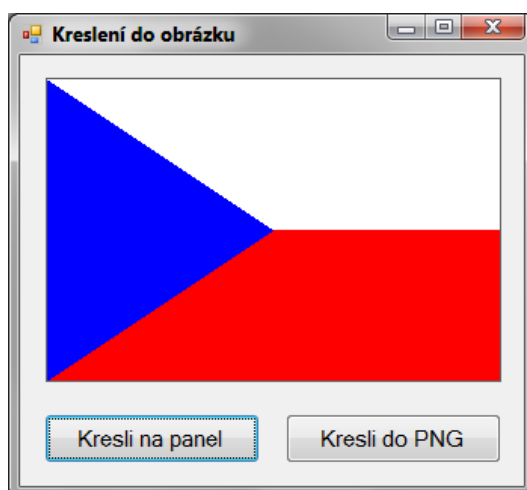
Poznámky:

- Poznáváte starou známou kreslicí plochu `kp`? Nyní to není kreslicí plocha okna ani panelu, ale kreslicí plocha rastrového obrázku. Fungují pro ni ale stejné příkazy jako pro kreslení do okna. Je to totiž proměnná přesně stejného typu, jak znáte - `Graphics`;

- `FillPolygon` kreslí vyplněný mnohoúhelník, možná jste se s touto metodou již někdo setkal;
- Pro vytvoření rastrového obrázku v paměti potřebujeme instancovat třídu `Bitmap`. Je použit tříparametrický konstruktér, který přebírá šířku a výšku obrázku a něco jako barevnou hloubku;
- Voláním metody `Graphics.FromImage` získáme z vytvořeného rastrového obrázku odkaz na jeho kreslicí plochu;
- Instance `Bitmap` má metodu `Save`, který umožňuje obrázek fyzicky zapsat na disk. V kódu je použit formát PNG, což je nejvhodnější volba pro obrázky s ostrými hranami (loga, tlačítka ap., v podstatě cokoli kromě fotek);
- Třídy `PixelFormat` a `ImageFormat` patří do jmenného prostoru `System.Drawing.Imaging`, proto je na začátek potřeba přidat příslušný `using`.

Současné kreslení do obrázku i do okna

Abychom si podobnost kreslení do obrázku a do okna ještě více demonstrovali, upravíme náš program tak, aby **jedinou metodou** kreslil jak do obrázku, tak do okna:



```
using System;
using System.Drawing;
using System.Windows.Forms;

using System.Drawing.Imaging; // přidáno

namespace Kreslení_do_obrázku
{
    public partial class oknoProgramu : Form
    {
        bool kreslitNaPanel = false;

        public oknoProgramu()
        {
            InitializeComponent();
        }
    }
}
```

```
private void tlačítkoKresliNaPanel_Click(object sender, EventArgs e)
{
    kreslitNaPanel = true;
    panel.Refresh();
}

private void panel_Paint(object sender, PaintEventArgs e)
{
    Graphics kp = e.Graphics;
    if (kreslitNaPanel)
        Kresli(kp);
}

private void tlačítkoKresliDoPng_Click(object sender, EventArgs e)
{
    // Zjistí jméno souboru
    if (oknoProVýběrSouboru.ShowDialog() == DialogResult.Cancel)
        return;
    string jménoSouboru = oknoProVýběrSouboru.FileName;

    // Připrav obrázek
    Bitmap obrázek = new Bitmap(300, 200, PixelFormat.Format24bppRgb);
    Graphics kp = Graphics.FromImage(obrázek);

    // Kresli!
    Kresli(kp);

    // Zapiš do souboru
    kp.Dispose();
    obrázek.Save(jménoSouboru, ImageFormat.Png);
    MessageBox.Show("Hotovo!");
}

private static void Kresli(Graphics kp)
{
    // Modrý klín
    kp.FillPolygon(Brushes.Blue,
        new Point[]
        {
            new Point(0, 0),
            new Point(150, 100),
            new Point(0, 200)
        });

    // Červený spodek
    kp.FillPolygon(Brushes.Red,
        new Point[]
        {
            new Point(0, 200),
            new Point(150, 100),
            new Point(300, 100),
            new Point(300, 200)
        });
};
```

```
// Bílý vršek
kp.FillPolygon(Brushes.White,
    new Point[]
    {
        new Point(0, 0),
        new Point(150, 100),
        new Point(300, 100),
        new Point(300, 0)
    });
}
```

Poznámky:

- Logika kreslení obrázku se neduplikuje, je soustředěna na jednom místě, v metodě `Kresli`, která přebírá odkaz na kreslicí plochu jako svůj parametr;
- Metoda `Kresli` je pak volána jak po stisku jednoho tlačítka, tak po stisku druhého (při kreslení na panel je volání zprostředkovaně v `Paintu`). Dostaneme tak stejný obrázek na panelu i v souboru;
- Rozčlenění do metod je přesně vzato pokryto až v učebnici pro středně pokročilé, principiálně by však tento program měl pochopit i začátečník.

Není to pěkné, ta jednota? Naučíte se jednou, použijete vícekrát...

Radek Vystavěl, 20. července 2013

Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu www.moderniProgramovani.cz.