

Kreslení v prostředí C# a .NET

- prostor jmen System.Drawing

Graphics

- kreslit se dá na cokoliv (formulář, tlačítko, nejlépe ale PictureBox)
- kresba je dočasná, při překreslení okna nebo komponenty zmizí
- vytvoření kreslicí plochy: objekt třídy Graphics

```
Graphics gr = this.CreateGraphics ();
```

```
Graphics gr = pictureBox1.CreateGraphics ();
```

Jak zachovat kresbu:

- a) překreslovat ji při každém překreslení komponenty
- kreslení umístit do obsluhy události Paint příslušné komponenty

```
private void pictureBox1_Paint  
    (object sender, PaintEventArgs e)
```

- používat v ní

```
Graphics gr = e.Graphics;
```

- b) kreslit do bitmapy a tu zobrazovat do PictureBox.Image

```
pictureBox1.Image = new Bitmap(300, 500);  
Graphics gr = Graphics.FromImage(pictureBox1.Image);
```

Velikost komponenty PictureBox

```
Bitmap bmp = new Bitmap(700, 600);  
Graphics gr = Graphics.FromImage(bmp);  
pictureBox1.Image = bmp;  
pictureBox1.Size = bmp.Size;
```

Pera a štětce

- čárové útvary se kreslím perem, výplně a písma štětcem

```
Pen pero = new Pen(barva, tloušťka);
```

například:

```
Pen pero = new Pen(Color.Red, 15);
```

Předpřipravená pera tloušťky 1 zvolených standardních barev najdeme ve třídě Pens

```
Pens.Red
```

```
Brush štětec = new SolidBrush(barva);
```

například:

```
Brush štětec = new SolidBrush(Color.Blue);
```

Předpřipravené štětce zvolených standardních barev najdeme ve třídě Brushes

```
Brushes.Blue
```

Příkazy na kreslení

- metody třídy Graphics
- je jich mnoho, uvádíme jen pár příkladů

```
gr.DrawLine(pero, x1, y1, x2, y2);          čára  
gr.DrawRectangle(pero, x, y, šířka, výška);  obdélník
```

```
gr.FillRectangle(štětec, x, y, šířka, výška);  
                                                    vyplněný obdélník
```

(nemá nakreslen okraj perem, to musíme provést zvlášť,
pokud si to přejeme)

```
gr.DrawString(string, font, štětec, x, y);    nápis
```

k tomu:

```
Font font = new Font("Arial", 50, FontStyle.Bold);
```

Začátek a konec čáry

```
Pen pero = new Pen(Color.ForestGreen, 40);  
pero.StartCap =  
    System.Drawing.Drawing2D.LineCap.Round;  
pero.EndCap =  
    System.Drawing.Drawing2D.LineCap.ArrowAnchor;  
gr.DrawLine(pero, 50, 50, 200, 300);
```

Styl čáry (přerušovaná apod.)

```
pero.DashCap =  
    System.Drawing.Drawing2D.DashCap.Round;  
pero.DashStyle =  
    System.Drawing.Drawing2D.DashStyle.Dash;
```

Barvy

- třída Color

```
Color c = new Color();
```

- kódování pomocí 4 bytů ARGB

A ... „alfa“ = průsvitnost, standardně hodnota 255 (neprůsvitná)

R, G, B ...barevné složky

c.A, c.R, c.G, c.B – složky barvy (byty)

Vlastní míchání barev

- metoda Color.FromArgb()

- metoda je přetížená, lze ji volat například takto:

```
c = Color.FromArgb(50, 100, 0, 255);
```

```
//4 parametry v pořadí ARGB
```

```
c = Color.FromArgb(100, 0, 255);
```

```
//3 parametry v pořadí RGB
```

Bitmapy

- třída Bitmap – nemusí mít nic společného s komponentami Windows Forms Application, lze ji používat i v konzolové aplikaci
- lze do ní vložit předem připravený obrázek ze souboru (formáty .bmp, .jpg, .png, .gif a mnoho dalších)
- obsah lze naopak uložit do souboru ve zvoleném grafickém formátu
- lze ji zobrazovat v PictureBox.Image
- lze ji vykreslovat metodou Graphics.DrawImage()
- umožňuje práci po jednotlivých pixelech (číst a zapisovat pixely)
- lze do ní kreslit pomocí prostředků třídy Graphics

```
pictureBox1.Image = new Bitmap(300, 500);
```

```
...
```

```
pictureBox1.Refresh();
```

musíme provést, pokud jsme změnili bitmapu v pictureBox1.Image a změnu chceme vidět na obrazovce


```
Bitmap bm = new Bitmap(300, 400);
```

nová prázdná bitmapa daných rozměrů

```
Bitmap bm = new Bitmap(@"C:\Obrazky\Auto.jpg");
```

nová bitmapa vytvořená z obrázku v souboru

```
bm.Save(@"C:\Obrazky\NoveAuto.png");
```

```
bm.Save(@"C:\Obrazky\NoveAuto.jpg",
```

```
System.Drawing.Imaging.ImageFormat.Jpeg);
```

pokud neuvedeme zvolený grafický formát,

použije se standardně formát .png

(formát se neurčuje z přípony souboru!)

```
Graphics gr = Graphics.FromImage (bm) ;  
gr.DrawLine (Pens.Green, 10, 10, 50, 50) ;
```

kreslení do bitmapy pomocí jejího Graphics
použití: doplnění obrázků (doplnění textu, rámečku apod.)

```
Color c = bm.GetPixel(x, y);
```

```
bm.PutPixel(x, y, c);
```

čtení jednotlivých pixelů a jejich zápis do bitmapy
použití: úpravy obrázků (změna barev, filtry ostrosti apod.)

Inverze barev v obrázku:

```
for (int x = 0; x < bm.Width; x++)  
    for (int y = 0; y < bm.Height; y++)  
    {  
        Color c = bm.GetPixel(x, y);  
        c = Color.FromArgb(255-c.R, 255-c.G, 255-c.B);  
        bm.SetPixel(x, y, c);  
    }
```

Převedení obrázku do odstínu šedi:

```
for (int x = 0; x < bm1.Width; x++)  
{  
    for (int y = 0; y < bm1.Height; y++)  
    {  
        Color c1 = bm1.GetPixel(x, y);  
        byte b = Convert.ToByte(0.2989*c1.R +  
                                0.5866*c1.G + (1-0.2989-0.5866)*c1.B);  
        Color c2 = Color.FromArgb(b, b, b);  
        bm2.SetPixel(x, y, c2);  
    }  
}
```