



Programování

Grafika a animace v C#

Pro připomenutí

- Třída obsahující práci s grafikou **System.Drawings**
- Pro práci s grafikou je třeba definovat nový objekt typu **Graphics**
- Grafika je vykreslována skrze obslužnou metodu **komponenta_Paint**
- Jako plátno pro kreslení obvykle používáme komponentu **Form** nebo **Panel**
- V závislosti zda chceme vykreslit obrys nebo obrazec využíváme třídy:
 - **Brushes** – výplně obrazců
 - **Pens** – obrysy, čáry, ...
- Máme možnost vytvářet si i vlastní objekty typu Brush, Pen

Třída Graphics – vybrané atributy

- Clip – udává region (třída Region), na který můžeme vkládat grafiku
- DpiX/DpiY – rozlišení tvořené grafiky
- Transform – matice transformace objektu (rotace, zkosení, škálování, ...)
- CompositingQuality – udává kvalitu pro renderování grafiky

```
// Set the Clip property to a new region.  
e.Graphics.Clip = new Region(new Rectangle(10, 10, 100, 200));
```



Třída Graphics – vybrané metody

- `Clear()` – vymazání obsahu plátna, umožňuje vybrat barvu pozadí plátna
- `DrawArc()` – nakreslení oblouku
- `DrawBezier()` – vykreslení Bezierovy křivky
- `DrawCurve()` – vykreslení křivky procházející konkrétními body
- `DrawEllipse()` – vykreslení elipsy případně kružnice
- `DrawImage()` – vykreslení obrázku ze zvoleného souboru
- `DrawLine()` – vykreslení křivky mezi určenými body
- `DrawPolygon()` – vykreslení polygonu určeného body
- `DrawRectangle()` – vykreslení obdélníku případně čtverce
- `DrawString()` – grafické zobrazení řetězce

Třída Graphics – vybrané metody

- `FillEllipse()` – vykreslení oválu nebo kruhu
- `FillPie()` – vykreslení výseče v kruhu
- `FillPolygon()` – vyplnění polygonu
- `FillRectangle()` – vyplnění obdélníku nebo čtverce

- `ResetTransform()` – resetování transformační matice na jednotkovou
- `RotateTransform()` – vložení matice rotace
- `ScaleTransform()` – vložení matice škálování
- `TranslateTransform()` – vložení matice posunu

Ukázka použití funkcí třídy Graphics

- Vykreslení červeného kruhu

```
Graphics g = e.Graphics;  
g.FillEllipse(Brushes.Red, 0, 0, 50, 50);
```

- Vstupními parametry pro vykreslení obrazce jsou:
- Štětec výplně, počátek vykreslování x, y a velikost ve směru x a y

- Vykreslení zelené kružnice

```
Graphics g = e.Graphics;  
g.DrawEllipse(Pens.Green, 0, 0, 50, 50);
```

- Vstupními parametry pro vykreslení kružnice jsou:
- Obrysové pero, počátek vykreslování x, y a velikost ve směru x a y

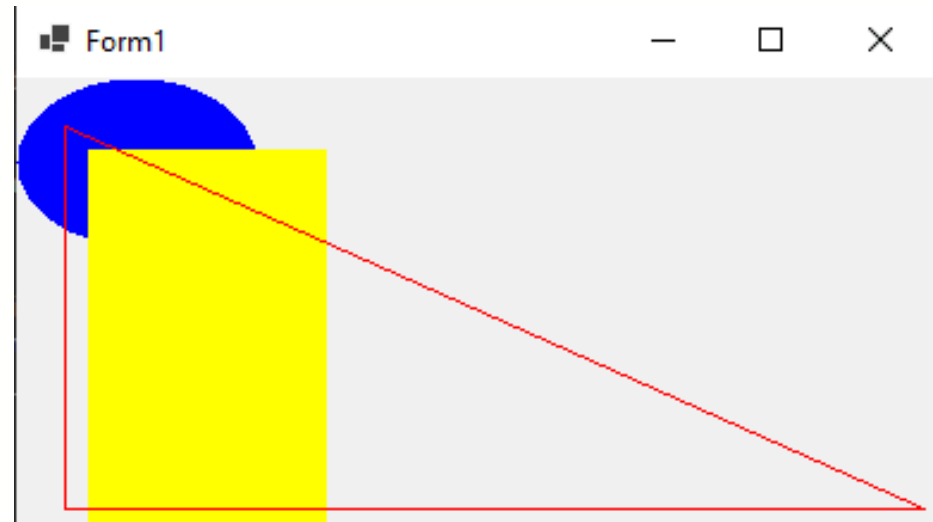
Jednoduchá grafika v C#

- Při vykreslování grafiky si je třeba uvědomit, že záleží na pořadí vykreslování obrazců
- Na problematiku lze pohlížet jako na řešení malířova algoritmu
- **Malířův algoritmus:** Jako první se vykreslují objekty v pozadí, které jsou postupně zakrývány objekty ležící před nimi

```
Počet odkazů: 1
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;

    g.FillEllipse(Brushes.Blue, 0, 0, 100, 70);
    g.FillRectangle(Brushes.Yellow, 30, 30, 100, 220);

    Point[] triangle = {
        new Point(20, 20),
        new Point(380, 180),
        new Point(20, 180),
        new Point(20, 20)
    };
    g.DrawLines(Pens.Red, triangle);
}
```





Animace v C#

- Pro tvorbu animace využijeme komponentu časovače Timer, která nám generuje v určitém časovém intervalu tzv. Tick
- Pokaždé, když uběhne daný čas vyvoláme překreslení všech objektů s upravenými parametry
- Časovač je ve výchozím stavu zakázán – Enabled = false
- Interval je udáván v milisekundách a výchozí hodnotou je 100ms
- Příkladem může být změna koordinát pro vykreslovaný objekt

Pohyb modrého čtverce po diagonále

- Do nového formuláře si vložíme komponentu panel a komponentu časovač
- Vykreslovaný čtverec má na začátku programu souřadnice 0,0
- S každým uplynulým intervalem zvedneme hodnoty x,y
- Panel obnovíme pomocí funkce Refresh()

```
public partial class Form1 : Form
{
    int x = 0;
    int y = 0;

    Počet odkazů: 1
    private void panel1_Paint(object sender, PaintEventArgs e)
    {
        e.Graphics.FillRectangle(Brushes.Blue, new Rectangle(x, y, 10, 10));
    }
    Počet odkazů: 1
    private void timer1_Tick(object sender, EventArgs e)
    {
        x++;
        y++;
        panel1.Refresh();
    }
}
```