C# - Grafika

Centrum pro virtuální a moderní metody a formy vzdělávání na Obchodní akademii T.G. Masaryka, Kostelec nad Orlicí

Co je za programováním grafiky ?

- Matematika, matematika, matematika
- OpenGL OpenGL seriál pro vývojáře
- DirectX Direct X seriál pro vývojáře
- Tvar a zrakové vnímání (světlo, stíny, odraz) objektu lze popsat pomocí matematickým vyjádřením (např. objekt má své parametrické vyjádření) světlo je vektor, a průnik světla s objektem lze vypočítat

Knihovna System.Drawing

- Umožňuje přístup k základním grafickým funkcím GDI+
- Rozšířené funkce poskytují:
  - System.Drawing.Drawing2D
  - System.Drawing.Imaging
  - System.Drawing.Text
- Seznam tříd, struktur, delegátů a výčtových typů této knihovny naleznete zde

Color (struktura)

- Slouží k uchování informací o barvě
- Reprezentuje ARGB (alpha, red, green, blue) barvu

```
private void ShowPropertiesOfSlateBlue(PaintEventArgs e)
{
    Color slateBlue = Color.FromName("SlateBlue");
    byte g = slateBlue.G;
    byte b = slateBlue.B;
    byte r = slateBlue.R;
    byte a = slateBlue.A;
    string text = String.Format("Slate Blue has these ARGB values: Alpha:{0}, " +
        "red:{1}, green: {2}, blue {3}", new object[]{a, r, g, b});
    e.Graphics.DrawString(text,
        new Font(this.Font, FontStyle.Italic),
        new SolidBrush(slateBlue),
        new RectangleF(new PointF(0.0F, 0.0F), this.Size));
}
```

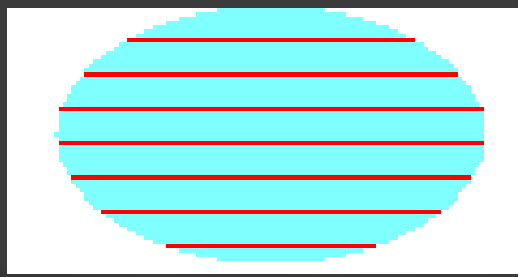
Pen (pero)

- Slouží k vykreslení čar a obrysů objektů (většinou pomocí metody Draw)
- Uchovává informace o barvě a tloušťce pera

```
private void ShowLineJoin(PaintEventArgs e)
{

    // Create a new pen.
    Pen skyBluePen = new Pen(Brushes.DeepSkyBlue);

    // Set the pen's width.
    skyBluePen.Width = 8.0F;

    // Set the LineJoin property.
    skyBluePen.LineJoin = System.Drawing.Drawing2D.LineJoin.Bevel;

    // Draw a rectangle.
    e.Graphics.DrawRectangle(skyBluePen,
        new Rectangle(40, 40, 150, 200));

    //Dispose of the pen.
    skyBluePen.Dispose();

}
```

Brush (struktura)

- Slouží k uchování informací o štětci
- Definuje objekty používané k vyplňování různých grafických objektů (elipsy, čtverce, atd.)
- Typy štětců:
  - HatchBrush – šrafovaný
  - LinearGradientBrush – štětec s lineárním sklonem
  - PathGradientBrush – vyplňuje oblast objektu, jež vznikl uzavřeným spojením několika křivek
  - SolidBrush – jednobarevný štětec
  - TextureBrush – štětec, který pro výplň používá nějaký obrázek, texturu



```
HatchBrush hBrush = new HatchBrush(
    HatchStyle.Horizontal,
    Color.Red,
    Color.FromArgb(255, 128, 255, 255));
e.Graphics.FillEllipse(hBrush, 0, 0, 100, 60);
```

Struktury

- V našich ukázkách budeme používat následující struktury:
  - Point
  - Size
  - Rectangle
- Jak již víte, struktury nejsou referenční datové typy – (viz. Úvodní přednáška o datových typech) – to znamená že jsou uloženy na zásobníku

Point

```
private void btnCreatePoint_Click(object sender, EventArgs e)
{
    Point myPoint = new System.Drawing.Point(100, 100);
}
```

System.Drawing.Point.Point(int x, int y) (+ 3 overload(s))
Initializes a new instance of the System.Drawing.Point class with the specified coordinates.

- Point ukládá dvě souřadnice v dvojrozměrném prostoru x,y
- Několik typů konstruktorů:
  - ***Point(int x,int y)*** – pomocí dvou souřadnic
  - **Point(int dw)** – pomocí jednoho čísla – nižších 16bitů představuje souřadnici x, vyšších 16bitů představuje souřadnici y
- Metoda ***Offset(int dx, int dy)*** slouží k posunu bodu o určitý vektor na novou pozici

- Použití viz. Ukázková aplikace : WinForm_BasicGraphicApp

DrawLine

- Metoda objektu Graphics
- Kreslí spojnici dvou bodů
- Typy volání najdete zde

```
public void DrawLinePoint(PaintEventArgs e)
{

    // Create pen.
    Pen blackPen = new Pen(Color.Black, 3);

    // Create points that define line.
    Point point1 = new Point(100, 100);
    Point point2 = new Point(500, 100);

    // Draw line to screen.
    e.Graphics.DrawLine(blackPen, point1, point2);
}
```

- Využívá objekt Pen (pero) sloužící k vykreslení čar a křivek:
- Property objektu Pen naleznete zde
- Vytvoření pera:

```
Pen skyBluePen = new Pen(Brushes.DeepSkyBlue);
skyBluePen.Width= 8.0F;
```

Rectangle

- Struktura pravoúhelník je dána levým horním rohem a svými rozměry (výška, šířka)
- Možné konstruktory naleznete zde
- V naší aplikaci využijeme konstruktor který využije strukturu Point a Size

```
//vytvořím bod pomocí konstruktoru, kde je zadána jeh x-ové a y-ové
//souřadnice
Point myPoint = new System.Drawing.Point(100, 100);
//vytvořím rozměr pomocí konstruktoru, kde je udána jeho x-ová a y-ová velikost
Size  mySize = new System.Drawing.Size(200, 300);

//vytvořím pravoúhelník pomocí struktur Bod a Rozměr
Rectangle myRectangle = new Rectangle(myPoint , mySize );
```

- Využijeme metody ***DrawRectangle*** k vykreslení pravoúhelníku pomocí pera a ***FillRectangle*** k jeho vyplnění zvolenou barvou

```
//vytvoříme pravoúhelník pomocí struktur Bod a Rozměr
Rectangle myRectangle = new Rectangle(myPoint , mySize );
//pro vykreslení budeme potřebovat objekt pero
//v konstruktoru je nastavena barva a tloušťka pera
System.Drawing.Pen myPen;
myPen = new System.Drawing.Pen(System.Drawing.Color.Red, 2);

//pro vykreslení je použit objekt typu Graphics a jeho metoda
//DrawLine
System.Drawing.Graphics formGraphics = this.CreateGraphics();
formGraphics.DrawRectangle(myPen, myRectangle);

//vybarvíme ho
formGraphics.FillRectangle(Brushes.SeaGreen, myRectangle );
```

Další metody Rectangle

- Inflate – mění rozměry pravouhelníka
- Offset – posunuje pravoúhelník

```
public void RectangleInflateTest2(PaintEventArgs e)
{

    // Create a rectangle.
    Rectangle rect = new Rectangle(100, 100, 50, 50);

    // Draw the uninflated rectangle to screen.
    e.Graphics.DrawRectangle(Pens.Black, rect);

    // Set up the inflate size.
    Size inflateSize = new Size(50, 50);

    // Call Inflate.
    rect.Inflate(inflateSize);

    // Draw the inflated rectangle to screen.
    e.Graphics.DrawRectangle(Pens.Red, rect);
}
```

Křivky

- Několik možných způsobů zadání:
  - Předdefinované metody vykreslení určitých geometrických útvarů

```
public void DrawLinePoint(PaintEventArgs e)
{

    // Create pen.
    Pen blackPen = new Pen(Color.Black, 3);

    // Create points that define line.
    Point point1 = new Point(100, 100);
    Point point2 = new Point(500, 100);

    // Draw line to screen.
    e.Graphics.DrawLine(blackPen, point1, point2);
}
```

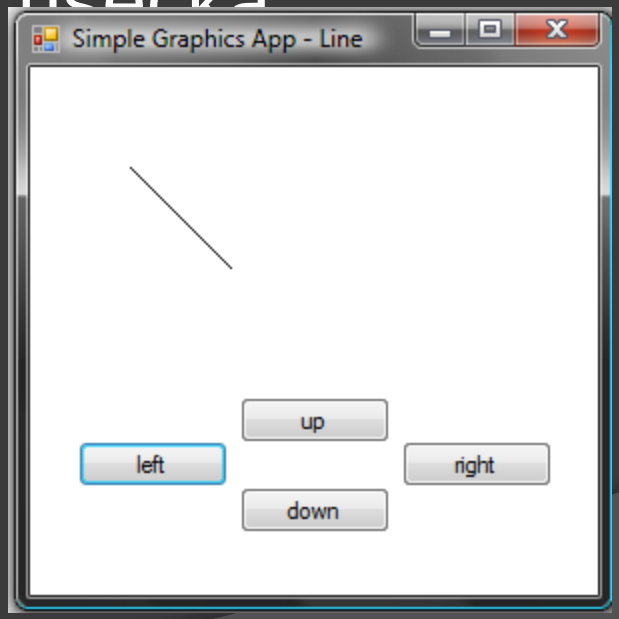
  - Použití parametrických rovnic geometrických útvarů

Využití parametrického vyjádření geometrických útvarů

- Teorii parametrického vyjádření můžete nalézt zde

Úkol 1

- Vytvořte formulářovou aplikaci dle vzoru. Tlačítka budou posouvat úsečku po formuláři. Úsečku se všemi metodami vytvořte jako objekt usecka



Pomoc při plnění úkolu

- Projekt nazvěte :
  - *GraphicsApp_Moveable_Line*
- Vytvořte objekt usecka s těmito atributy:
  - *private Point a, b; //krajní body přímky*
  - *private Color barva; //barva*
  - *private Pen pero; //pero pro vykreslení přímky*
- V objektu usecka přidejte do sekce *using* knihovnu *System.Drawing*
- Vytvořte konstruktor, který bude mít dva parametry typu Point a defaultně nastaví barvu na černou a vytvoří Pero s touto barvou
- Přidejte atribut Barva s metodami get a set – set nenastaví pouze property barva ale zároveň změní i barvu pera
- Přidejte metody, které provedou posun souřadnic krajních bodů (doprava, doleva, nahoru, dolů) vždy o **10 pixelů**
- Přidejte metodu, která se bude jmenovat *nakresli_usecku* a bude mít jediný parametr (*System.Windows.Forms.Form **myForm***)

  tato metoda vytvoří grafický objekt na formuláři předaném parametrem a zavolá metodu DrawLine k vykreslení úsečky

  Uvolněte vytvořený grafický objekt pomocí metody Dispose

- Vytvořte formulář dle obrázku a pojmenujte všechny jeho Controls
- Na formuláři vytvořte globální proměnnou typu objekt usecka
- Vytvořte na formuláři metodu CreateLine() ve které vytvoříte vytvoříte instanci objektu usecka s tím že při vytvoření nastavíte body ohraničující úsečku na souřadnice 50,50 a 100,100
- Přidejte na formulář událost Shown() a zavolejte v ní událost CreateLine()
- Dodefinujte kódy, které budou provedeny po stisku jednotlivých tlačítek (překreslit existující úsečku barvou pozadí formuláře, posunout souřadnice, nakreslit novou úsečku)
- Řešení: ***GraphicsApp_Moveable_Line***

Úkol 2

- Doplňte předchozí projekt tak, aby posun úsečky reagoval na klik myši na formuláři