

Tomasz Gańko

# **Windows 8**

# **Rysowanie 2D**

# Wprowadzenie

Przestrzenie nazw:

- Windows.UI.Xaml.Shapes
- Windows.UI;

Obiekty:

- Grid
- Canvas

# Podstawowe kształty

- Linia
- Prostokąt
- Elipsa (koło)
- Polilinia
- Poligon
- Obiekt tekstowy

# Linia - Line

```
void DrawLine(Canvas p, Point start, Point end, SolidColorBrush color, double thickness)
{
    Line line = new Line();
    line.X1 = start.X;
    line.Y1 = start.Y;
    line.X2 = end.X;
    line.Y2 = end.Y;
    line.Stroke = color;
    line.StrokeThickness = thickness;

    p.Children.Add(line);
}
```

# Prostokąt - Rectangle

```
void DrawRectangle(Canvas p, Thickness position, double width, double height, SolidColorBrush fill,
                  SolidColorBrush stroke, double thickness)
{
    Rectangle rect = new Rectangle();
    rect.HorizontalAlignment = HorizontalAlignment.Left;
    rect.VerticalAlignment = VerticalAlignment.Top;
    rect.Margin = position;
    rect.Width = Math.Abs(width);
    rect.Height = Math.Abs(height);
    rect.Stroke = stroke;
    rect.Fill = fill;
    rect.StrokeThickness = thickness;

    p.Children.Add(rect);
}
```

# Elipsa / koło - ellipse

```
void DrawEllipse(Canvas p, Thickness position, double width, double height,  
                SolidColorBrush fill, SolidColorBrush stroke, double thickness)  
{  
    Ellipse el = new Ellipse();  
    el.Fill = fill;  
    el.StrokeThickness = thickness;  
    el.Stroke = stroke;  
    el.Width = Math.Abs(width);  
    el.Height = Math.Abs(height);  
    el.Margin = position;  
  
    p.Children.Add(el);  
}
```

# Polilinia – Polyline

```
void DrawPolyline(Canvas p, SolidColorBrush stroke, double thickness,  
                  int numberOfpoints, List<Point> pts)  
{  
    Polyline polyline = new Polyline();  
    polyline.Stroke = stroke;  
    polyline.StrokeThickness = thickness;  
    for (int i = 0; i < numberOfpoints; i++)  
    {  
        polyline.Points.Add(pts[i]);  
    }  
  
    p.Children.Add(polyline);  
}
```

# Poligon - Polygon

```
void DrawPolygon(Canvas p, SolidColorBrush fill, SolidColorBrush stroke, double thickness,
                 int numberOfpoints, List<Point> pts)
{
    Polygon polygon = new Polygon();
    polygon.Stroke = stroke;
    polygon.StrokeThickness = thickness;
    polygon.Fill = fill;
    for (int i = 0; i < numberOfpoints; i++)
    {
        polygon.Points.Add(pts[i]);
    }

    p.Children.Add(polygon);
}
```



# TextBlock

```
void DrawTextBlock(Canvas p, double height, Thickness thickness,  
                  SolidColorBrush col, string text,int FSize)  
{  
    TextBlock textB = new TextBlock();  
    textB.Height = height;  
    textB.HorizontalAlignment = HorizontalAlignment.Left;  
    textB.Margin = thickness;  
    textB.Text = text;  
    textB.FontSize = FSize;  
    textB.Foreground = col;  
    p.Children.Add(textB);  
}
```

# Podstawy rysowania

Aby dodać na ekran jakikolwiek kształt trzeba:

- Podać jego kolor
- Podać współrzędne
- Podać wysokość i długość

współrzędne (0, 0)



# Operacje na kształtach

- Aby dodać kształt do danego elementu typu Grid, lub Canvas należy użyć:

```
canvas.Children.Add(kształt);
```

- Aby poznać liczbę dodanych już elementów:

```
canvas.Children.Count();
```

- Aby usunąć konkretny kształt:

```
canvas.Children.RemoveAt(numer);
```

Czas na pokaz :)

# Demo

# Aplikacja użytkowa

To mógłby być koniec...

ale...

# Paint-Shape

- Założeniem było napisanie prostego programu umożliwiającego rysowanie kształtów za pomocą dotyku/myszy.
- Stworzenie jednolitego interfejsu użytkownika.
- Dodawanie do rysunku tekstu.
- Możliwość cofania dokonanych zmian.

# Możliwości Paint-Shape

# Demo

## Windows 8 – Rysowanie 2D

Dziękuję za uwagę :)