

塗りつぶす

閉じた図形

前回、基本的な論理ブラシについて解説しました

ブラシが理解できれば、閉じた図形を自由に塗りつぶすことができます

前は長方形を塗りつぶす `FillRectangle()` を使ったので、今回はこれ以外の
まだ説明していない、塗りつぶし用の `Graphics` クラスのメソッドを紹介します

円弧を塗りつぶすには **`Graphics.FillEllipse()`** メソッドを使います

やはり、ペンの代わりにブラシを使うこと以外は `DrawEllipse()` と同じです

```
public void FillEllipse(Brush brush, Rectangle rect);  
public void FillEllipse(Brush brush, RectangleF rect);  
public void FillEllipse(Brush brush, int x, int y, int width, int height);  
public void FillEllipse(Brush brush, float x, float y, float width, float height);
```

brush には塗りつぶしに用いるブラシの `Brush` オブジェクトを指定します

rect は円弧が外接する長方形の情報を格納した構造体を指定します

x と y には長方形の左上角を示す座標、width と height は幅と高さを指定します

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        g.FillEllipse(  
            new SolidBrush(Color.FromArgb(0xAA , 0xAA , 0xFF)) ,  
            10 , 10 , 200 , 100  
        );  
    }  
}
```



欠けた円、すなわち特定の角度のみを描画する `DrawPie()` の塗りつぶし

`Graphics.FillPie()` メソッドを用いて行うことができます

```
public void FillPie(  
    Brush brush, Rectangle rect ,  
    float startAngle, float sweepAngle  
);  
public void FillPie(  
    Brush brush ,  
    int x, int y, int width, int height ,  
    int startAngle, int sweepAngle  
);  
public void FillPie(  
    Brush brush ,  
    float x, float y, float width, float height ,  
    float startAngle, float sweepAngle  
);
```

brush には塗りつぶしに使う `Brush` オブジェクトを指定します

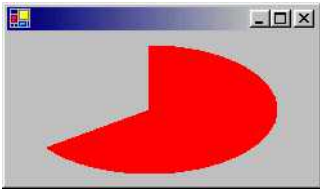
rect には、円が外接する長方形の大きさを格納する構造体を指定します

x と y には、長方形の左上角を表す座標を、width と height には幅と高さを指定します

startAngle には描画の開始角度を、sweepAngle には終了角度を指定します

角度は、時計回りに計測されます

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        g.FillPie(  
            new SolidBrush(Color.FromArgb(0xFF , 0 , 0)) ,  
            10 , 10 , 200 , 100 , -90 , 250  
        );  
    }  
}
```



これも DrawPie() メソッドとほとんど同じなので説明不要でしょう
また、長方形の塗りつぶしには複数の長方形を同時に塗りつぶすことができる
Graphics.FillRectangles() というメソッドも用意されています

```
public void FillRectangles(Brush brush , Rectangle[] rects);  
public void FillRectangles(Brush brush , RectangleF[] rects);
```

brush には塗りつぶす時の Brush オブジェクトを
rects には塗りつぶす長方形の情報を持った配列へのポインタを指定します

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        Rectangle[] rect = {  
            new Rectangle(10 , 10 , 200 , 100) ,  
            new Rectangle(20 , 20 , 200 , 100)  
        };  
        g.FillRectangles(new SolidBrush(Color.FromArgb(0xFF , 0 , 0)) , rect);  
    }  
}
```



このプログラムは、二つの長方形を同時に描画しています
頻繁に使うことはないと思いますが、場合によっては便利なメソッドです

複数の頂点の配列から構成されるポリゴンの面を塗りつぶすには
Graphics.FillPolygon() メソッドを利用します

```
public void FillPolygon(Brush brush , Point[] points);  
public void FillPolygon(Brush brush , PointF[] points);  
public void FillPolygon(Brush brush , Point[] points , FillMode fillMode);  
public void FillPolygon(Brush brush , PointF[] points , FillMode fillMode);
```

brush には、用いるブラシの Brush オブジェクトを指定します
points は頂点の座標が格納された構造体の配列を
FillMode には塗りつぶしモードを指定します

最後の引数は **System.Drawing.Drawing2D.FillMode** 列挙型です
この列挙型は、多角形モードのメンバを次のように定義しています

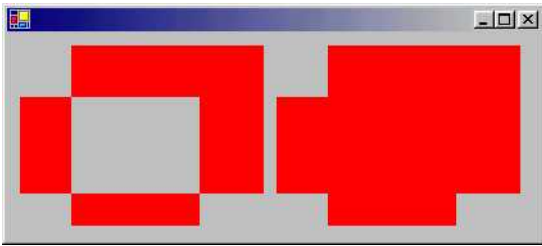
メンバ	解説
Alternate	これは多角形の内部で隣接しない部分を塗りつぶす 他の境界線と交差している内面は塗りつぶされない デフォルトです
Winding	交差する線の方が逆の場合は塗りつぶされない

```
using System.Windows.Forms;  
using System.Drawing;  
using System.Drawing.Drawing2D;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        Point[] pt = {  
            new Point(10 , 50) , new Point(150 , 50) ,  
            new Point(150 , 150) , new Point(50 , 150) ,  
            new Point(50 , 10) , new Point(200 , 10) ,  
            new Point(200 , 125) , new Point(10 , 125)  
        };  
        g.FillPolygon(new SolidBrush(Color.FromArgb(0xFF , 0 , 0)) , pt);  
    }  
}
```

```

for (int i = 0 ; i < pt.Length ; i++) pt[i].X += 200;
g.FillPolygon(
    new SolidBrush(Color.FromArgb(0xFF , 0 , 0)) ,
    pt , FillMode.Winding
);
}
}

```



このプログラムは、ポリゴンの塗りつぶしと塗りつぶしモードの違いについて理解できるように作ったサンプルです
最初の FillPolygon() は左の図形はデフォルトの Alternate を
右側の図形は明示的に Winding を指定しています

また、以前紹介したスプライン曲線にも閉じた図形が存在します
これには **Graphics.FillClosedCurve()** メソッドを使用します

```

public void FillClosedCurve(Brush brush , Point[] points);
public void FillClosedCurve(Brush brush , PointF[] points);

```

```

public void FillClosedCurve(
    Brush brush , Point[] points ,
    FillMode fillmode
);
public void FillClosedCurve(
    Brush brush , PointF[] points ,
    FillMode fillmode
);
public void FillClosedCurve(
    Brush brush , Point[] points,
    FillMode fillmode , float tension
);
public void FillClosedCurve(
    Brush brush , PointF[] points ,
    FillMode fillmode , float tension
);

```

brush には塗りつぶしに用いる Brush オブジェクトを
points には、曲線を定義する各頂点を示す構造体の配列を指定します
配列 points は、最低でも4つ以上の頂点を含んでいなければなりません
fillmode には塗りつぶしモードを、tension には、カーブの強さを指定します

塗りつぶしモードの解説がまだだったため、説明していませんでしたが
同様に、**Graphics.DrawClosedCurve()** メソッドを用いることによって
閉じたスプライン曲線の輪郭を描くことが可能です

```

public void DrawClosedCurve(Pen pen , Point[] points);
public void DrawClosedCurve(Pen pen , PointF[] points);

```

```

public void DrawClosedCurve(
    Pen pen , Point[] points ,
    float tension , FillMode fillmode
);
public void DrawClosedCurve(
    Pen pen , PointF[] points ,
    float tension , FillMode fillmode
);

```

pen には輪郭の描画に用いる Pen オブジェクトを
points には曲線を定義する各頂点を示す構造体の配列を指定します
tension にはカーブの強さ、fillmode には塗りつぶしモードを指定します

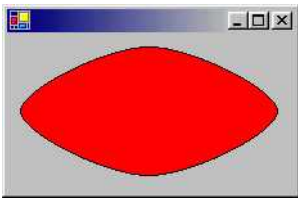
```

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

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        Point[] pt = {
            new Point(10 , 60) , new Point(110 , 10) ,
            new Point(210 , 60) , new Point(110 , 110)
        };
        Brush myBrush = new SolidBrush(Color.FromArgb(0xFF , 0 , 0));
        Pen myPen = new Pen(Color.FromArgb(0 , 0 , 0));

        g.FillClosedCurve(myBrush , pt);
        g.DrawClosedCurve(myPen , pt);
    }
}

```



このプログラムは、スプライト曲線の輪郭を描画し内部を塗りつぶしています
DrawCurve() メソッドと違うのは、最後の頂点の後、線が閉じられるということです

[前のページへ](#)

[戻る](#)

[次のページへ](#)