Graphics

By Pakita Shamoi

System.Drawing Namespace

- Enables you to create graphics from scratch or modify existing images.
- Classes
  - Bitmap
  - Brush (used to fill shapes)
    - SolidBrash – single color
    - TextureBrash - uses an image to fill the interior of a shape
  - Font (size, face, etc.)
  - Graphics – GDI+ drawing surface
  - Pen – used to draw curves, arrows, lines
  - Region – an interior of a shape

System.Drawing Namespace

- Structures
  - Color
  - Point
  - Rectangle - stores a set of four integers that represent the location and size of a rectangle
  - Size – width, height

Location, Size, Color

- button1.Location = new Point(10, 10);
- Or: button1.Left = 10;
  button1.Top = 10;

- button1.Size = new Size(30, 30);

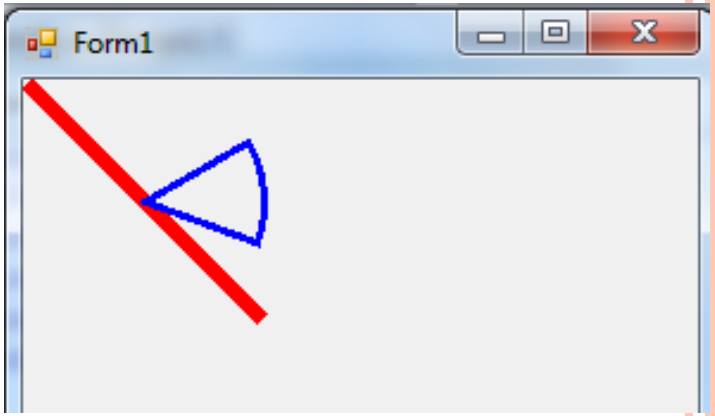
- button1.ForeColor = Color.Red;
  button1.BackColor = Color.Blue;
  Or: button1.ForeColor = Color.FromArgb(10, 200, 200);

Drawing Lines and Shapes

- Methods of Graphics class
  - Clear()
  - DrawEllipse()
  - DrawLine()
  - DrawRectangle()
  - DrawString()
  - DrawPolygon()
  - DrawPie()

Example

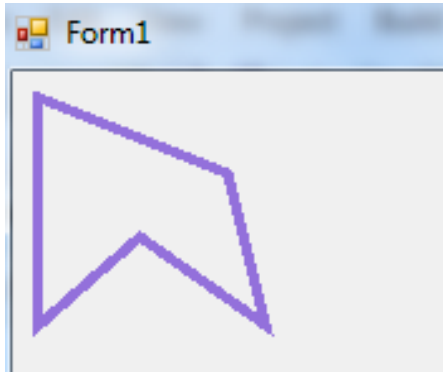
- Add the code to a method run during the form's *Paint event.*




```
// Create a graphics object from the form
            Graphics g = this.CreateGraphics();

// Create a pen object with which to draw

            Pen p = new Pen(Color.Red, 7);

            // Draw the line
            g.DrawLine(p, 1, 1, 100, 100);


            Pen p2 = new Pen(Color.Blue, 3);
            g.DrawPie(p2, 1, 1, 100, 100, -30, 50);
```

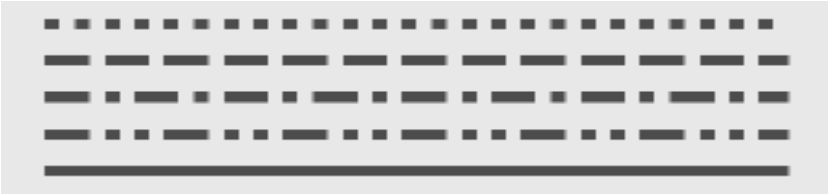
Example

```
Pen purple = new Pen(Color.MediumPurple, 4);
     // Create an array of points
Point[] points = new Point[]{new Point(10, 10),
                             new Point(10, 100),
                             new Point(50, 65),
                             new Point(100, 100),
                             new Point(85, 40)};
  // Draw a shape defined by the array of points
     g.DrawPolygon(purple, points);
```



Customizing pen

- Besides controlling the **color** and **size** of a pen, which are specified in the *Pen constructor,* you can also control the **pattern** and **endcaps**
- To draw a dotted line, create an instance of the *Pen* class, and then set the *Pen.DashStyle* property to one of these values: *DashStyle.Dash, DashStyle.DashDot, DashStyle.DashDotDot, DashStyle.Dot, or DashStyle.Solid*
- ***Dot, Dash, DashDot, DashDotDot, Solid***



Customizing Pen

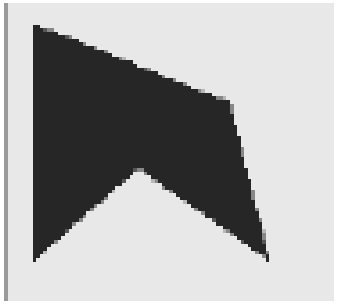
- To control the endcaps and create arrows or callouts, modify the *Pen.StartCap and Pen.EndCap* properties using the *LineCap* enumeration.

```
Pen p = new Pen(Color.Red, 10);
p.StartCap = LineCap.ArrowAnchor;
p.EndCap = LineCap.DiamondAnchor;
g.DrawLine(p, 50, 25, 400, 25);
p.StartCap = LineCap.SquareAnchor;
p.EndCap = LineCap.Triangle;
g.DrawLine(p, 50, 50, 400, 50);
p.StartCap = LineCap.Flat;
p.EndCap = LineCap.Round;
g.DrawLine(p, 50, 75, 400, 75);
p.StartCap = LineCap.RoundAnchor;
p.EndCap = LineCap.Square;
g.DrawLine(p, 50, 100, 400, 100);
```

Filling Shapes

- The Graphics class also has Fill methods that draw a shape and fill in the contents.
- These methods work exactly like the Draw methods, except they require an instance of the *Brush* class instead of the *Pen* class.

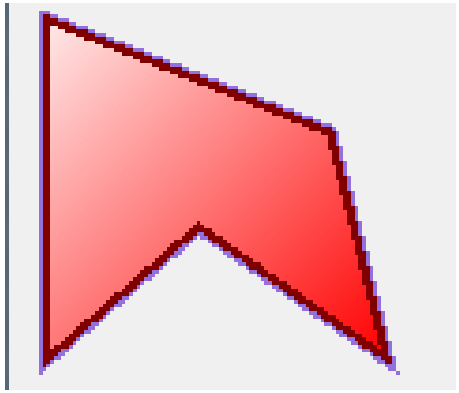
```
Graphics g = this.CreateGraphics();
Brush b = new SolidBrush(Color.Maroon);
Point[] points = new Point[]
{new Point(10, 10),
new Point(10, 100),
new Point(50, 65),
new Point(100, 100),
new Point(85, 40)};
g.FillPolygon(b, points);
```



Filling shapes

- You can draw filled objects with an outline by first calling the *Graphics class Fill* method, and then calling the *Graphics class Draw method.*

```
Graphics g = this.CreateGraphics();
Pen p = new Pen(Color.Maroon, 2);
Brush b = new LinearGradientBrush(new Point(1,1), new
Point(100,100),Color.White, Color.Red);
Point[] points = new Point[]{
        new Point(10, 10),
        new Point(10, 100),
        new Point(50, 65),
        new Point(100, 100),
        new Point(85, 40)};
g.FillPolygon(b, points);
g.DrawPolygon(p, points);
```



Summary

- The *System.Drawing namespace provides tools for drawing graphics and editing*

  existing images. The most useful classes are *Graphics and Bitmap.*
- Use the *Point and Size classes to specify the location and size of controls.*
- The *System.Drawing.Color structure provides predefined properties for common*

  colors.
- To draw lines and shapes, create an instance of the *Graphics class, create a Pen*

  object, and then call one of the *Graphics member methods to draw a line or a*

  shape using the *Pen instance.*
- Pens can be customized by adding endcaps or changing the line pattern to

  various combinations of dots and dashes.
- To draw solid shapes, create an instance of the *Graphics class, create a Brush*

  object, and then call one of the *Graphics member methods to draw the shape*

  using the *Brush instance.*