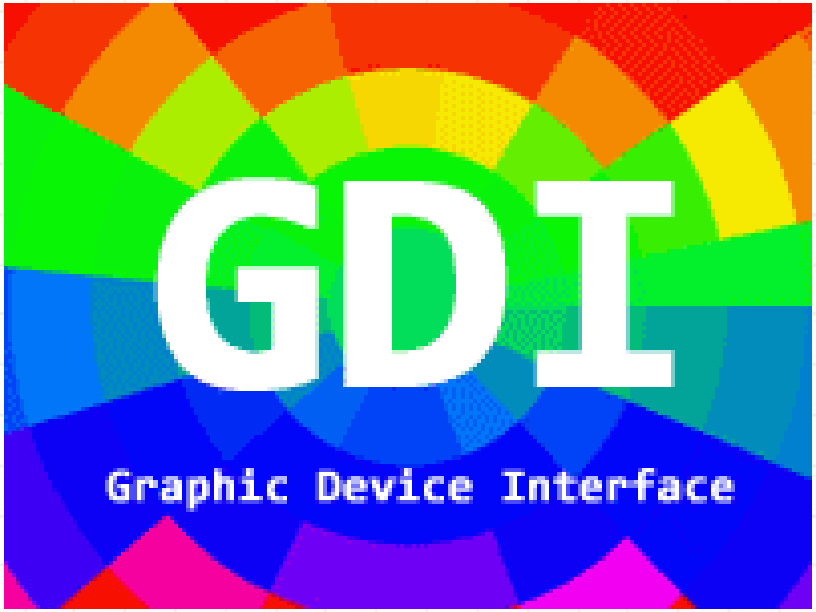
LẬP TRÌNH GDI+ # PHẦN 2

CÁC CÔNG CỤ NÂNG CAO



Nội dung – Phần 2

1	Sự kiện Paint của Form
2	Các công cụ vẽ nâng cao

Sự kiện Paint của Form

Khi nào sự kiện Paint xảy ra?

1) Lần đầu Form được hiển thị

Khi chạy ứng dụng và Form xuất hiện, hệ thống gọi OnPaint để vẽ giao diện lần đầu

2) Khi Form (hoặc control) bị che rồi hiển thị lại

Vd: khi mở cửa sổ khác che Form, sau đó đóng cửa sổ kia -> vùng bị che cần vẽ lại

3) Khi Form bị thay đổi kích thước (resize)

Vùng hiển thị bị thay đổi -> cần vẽ lại

4) Khi gọi phương thức Invalidate() hoặc Refresh()

- Invalidate() báo cho hệ thống rằng vùng client cần vẽ lại → Paint sẽ được gọi.
- Refresh() = Invalidate() + Update() (tức gọi vẽ lại ngay lập tức)

5) Khi nội dung Form bị thay đổi, cần vẽ lại

Vd: vẽ lại hình, thay đổi màu nền, di chuyển control bên trong

Sự kiện Paint của Form

Cách xử lý sự kiện Paint:

(1) Gắn event handler:

```
public Form1()
{
    InitializeComponent();
    this.Paint += new PaintEventHandler(Form1_Paint);
}

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawLine(Pens.Black, 10, 10, 200, 200);
    g.DrawString("Hello Paint Event", new Font("Arial", 14),
Brushes.Red, 10, 50);
}
```

Sự kiện Paint của Form

Cách xử lý sự kiện Paint:

(2) Ghi đè phương thức OnPaint:

```
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    Graphics g = e.Graphics;
    g.FillEllipse(Brushes.Blue, 50, 50, 100, 70);
}
```

Sự kiện Paint của Form

Lưu ý quan trọng:

- Không nên gọi OnPaint trực tiếp. Muốn vẽ lại thì gọi Invalidate() hoặc Refresh().
- Vẽ trong sự kiện Paint để đảm bảo tính **bền vững** (persistent):
- Nếu bạn chỉ vẽ trong sự kiện click chẳng hạn, khi form bị che rồi hiện lại thì hình sẽ biến mất (vì không được lưu trạng thái).
- Trong Paint, hệ thống **tự động vẽ lại mọi lần cần thiết**

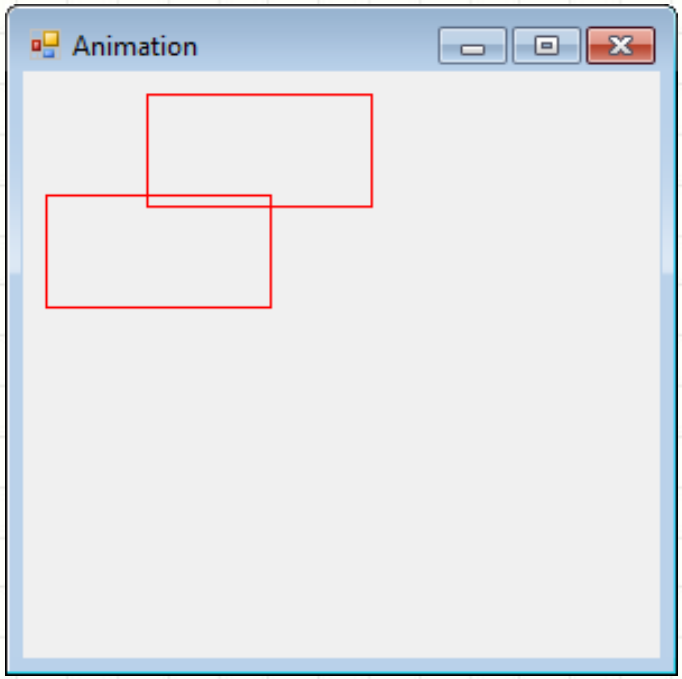
Sự kiện Paint của Form

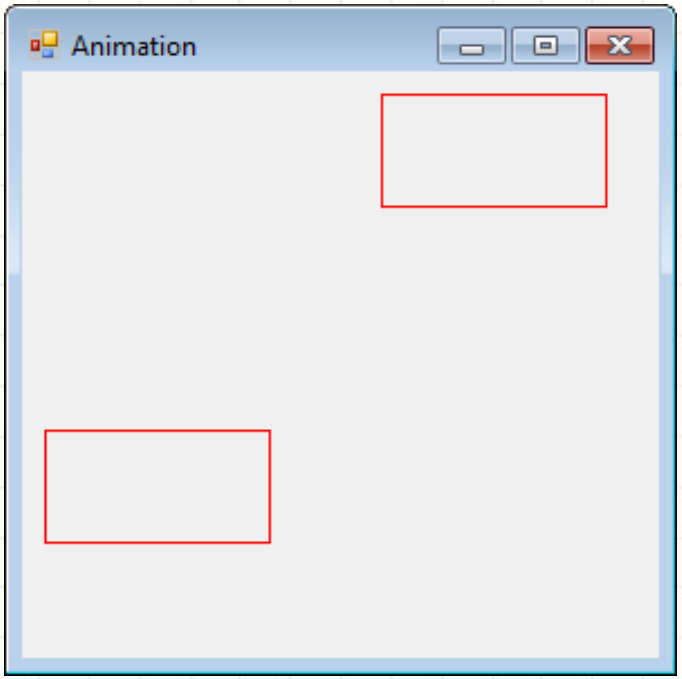
Ứng dụng trong animation:

- Không nên: "xóa cũ vẽ mới"
- Nên: vẽ lại form theo tốc độ nhất định, kiểm soát bằng các biến trạng thái

```
protected int x=0;
protected int y=0;
private void Form1_Paint(object sender, PaintEventArgs e) {
        Graphics g = e.Graphics;
        Pen pen = new Pen(Color.Red);
        g.DrawRectangle(pen, x, 10, 100, 50);
        g.DrawRectangle(pen, 10, y, 100, 50);
}
private void timer1_Tick(object sender, EventArgs e) {
        x = (x + 1) % 200; y = (y+1) % 200;
        Refresh();
}
```

Sự kiện Paint của Form





Các công cụ vẽ nâng cao

Các lớp kế thừa từ Brush

❖ **SolidBrush**

Là một brush dùng để tô một vùng với một màu đơn.

❖ **LinearGradientBrush**

Là một brush dùng để tô một vùng với cách trộn nhiều màu lại với nhau.

❖ **TextureBrush**

Cho phép sử dụng ảnh như là một brush để tô các đối tượng.

❖ **HatchBrush**

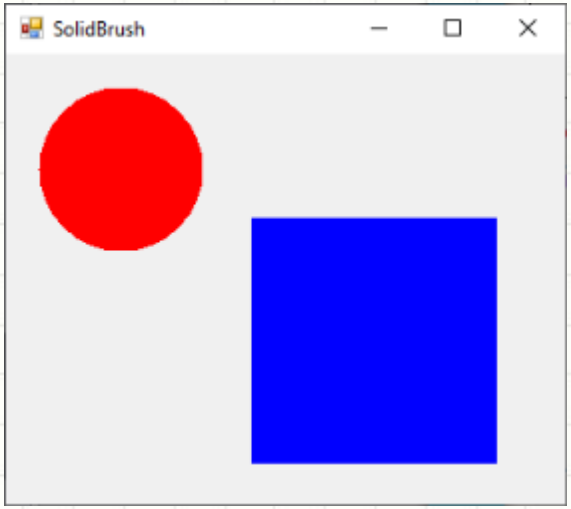
Hatch brush là brush được sử dụng dựa trên các kiểu Hatch Style, màu nền và màu đường

Các công cụ vẽ nâng cao

Các lớp kế thừa từ Brush

❖ SolidBrush

```
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    SolidBrush redBrush = new SolidBrush(Color.Red);
    SolidBrush blueBrush = new SolidBrush(Color.Blue);
    g.FillEllipse(redBrush, 20, 20, 100, 100);
    Rectangle rect = new Rectangle(150, 100, 150, 150);
    g.FillRectangle(blueBrush, rect);
}
```



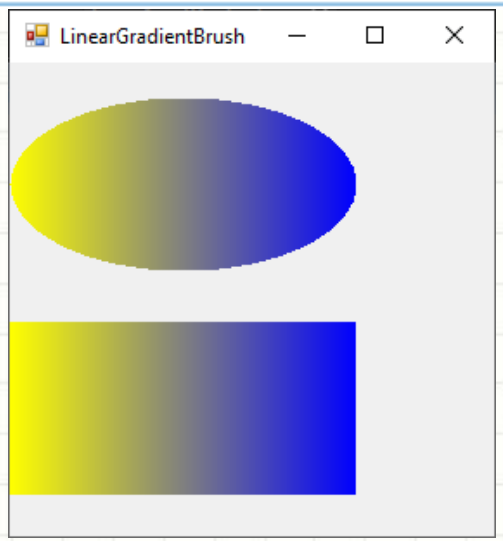
Các công cụ vẽ nâng cao

Các lớp kế thừa từ Brush

❖ LinearGradientBrush

```
protected override void OnPaint(PaintEventArgs e)
{
    Point pt1 = new Point(0, 10);
    Point pt2 = new Point(200, 10);
    LinearGradientBrush brush = new LinearGradientBrush(pt1,
                                                        pt2,
                                                        Color.Yellow,
                                                        Color.Blue);

    e.Graphics.FillEllipse(brush, 0, 20, 200, 100);
    e.Graphics.FillRectangle(brush, 0, 150, 200, 100);
}
```

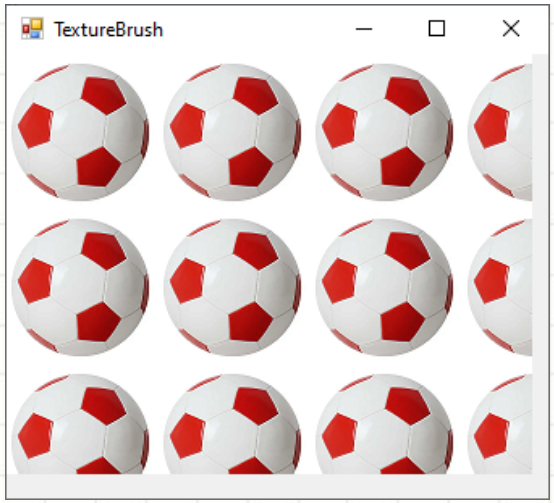


Các công cụ vẽ nâng cao

Các lớp kế thừa từ Brush

❖ TextureBrush

```
protected override void OnPaint(PaintEventArgs e)
{
    Image img = new Bitmap("D:\\ball.jpg");
    TextureBrush brush = new TextureBrush(img);
    Graphics g = e.Graphics;
    Rectangle rect = new Rectangle(0, 0, 325, 260);
    g.FillRectangle(brush, rect);
}
```



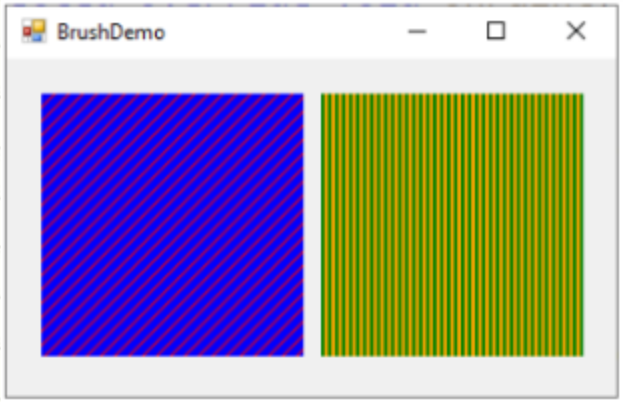
Các công cụ vẽ nâng cao

Các lớp kế thừa từ Brush

❖ HatchBrush

```
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    HatchStyle style = HatchStyle.BackwardDiagonal;
    Color foreColor = Color.Red;
    Color bgColor = Color.Blue;
    HatchBrush brush = new HatchBrush(style, foreColor, bgColor);
    g.FillRectangle(brush, 20, 20, 200, 200);

    style = HatchStyle.DarkVertical;
    foreColor = Color.Green;
    bgColor = Color.Orange;
    brush = new HatchBrush(style, foreColor, bgColor);
    g.FillRectangle(brush, 240, 20, 200, 200);
}
```



Các công cụ vẽ nâng cao

Pen nâng cao

- Thuộc tính **LineCap** của Pen được dùng để xác định kiểu điểm đầu (***StartCap***) và điểm cuối (***EndCap***) của các dòng vẽ bởi Pen.

  GetLineCap(): Trả về một enum LineCap.

  SetLineCap(): Áp dụng một LineCap cho Pen.

- Một số kiểu **LineCap**:

  ArrowAnchor

  RoundAnchor

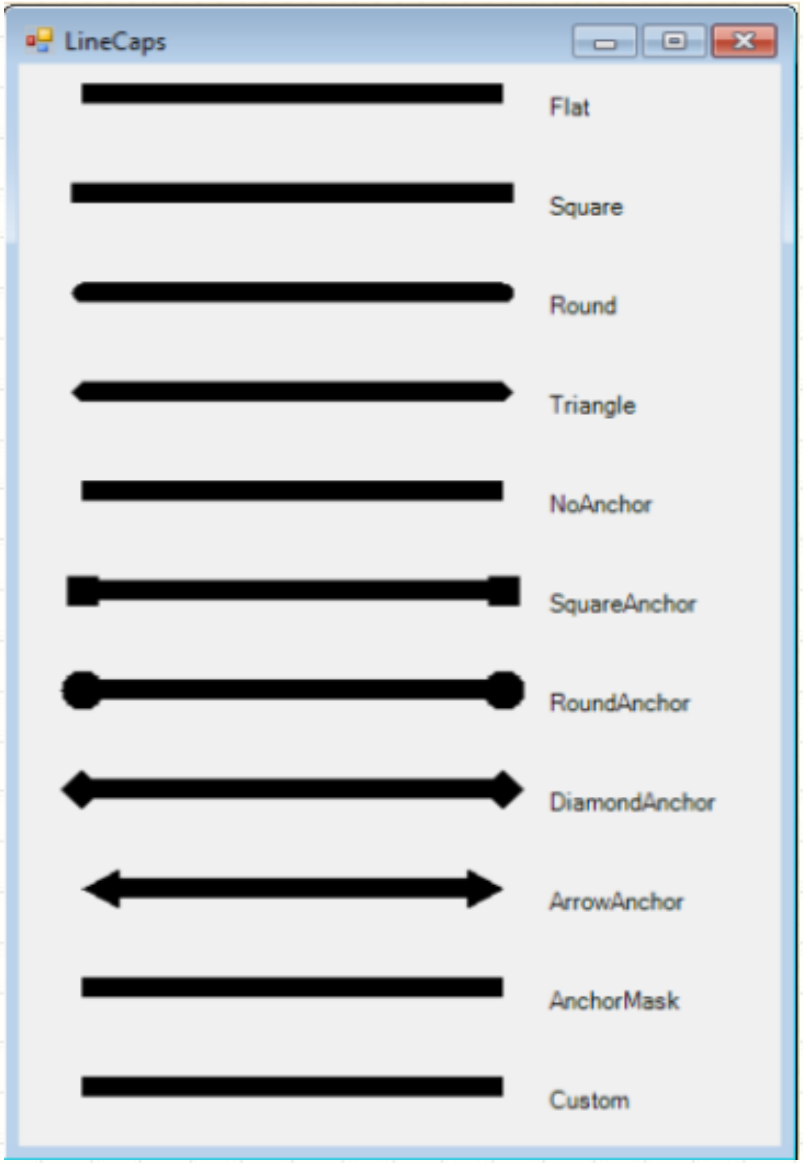
  SquareAnchor

  Triangle

  …

Các công cụ vẽ nâng cao

Pen nâng cao



Các công cụ vẽ nâng cao

Pen nâng cao

- Thuộc tính **DashStyle**: Các kiểu đường đứt khúc
  - Custom
  - Dash
  - DashDot
  - DashDotDot
  - Dot
  - Solid

Các công cụ vẽ nâng cao

Pen nâng cao

❑ Thuộc tính **DashCap**: Kết hợp một vài thuộc tính của **LineCap** và **DashStyle**

- Flat
- Round
- Triangle

Các công cụ vẽ nâng cao

Image

Cho phép vẽ các hình ảnh

- Tạo các hình ảnh thông qua class Image (Bitmap, Metafile, Icon,...)
- Class Bitmap hỗ trợ các định dạng chuẩn GIF, JPG, BMP, PNG, TIFF
- Sử dụng hàm **DrawImage**, **DrawIcon**, **DrawIconUnstretched**, **DrawImageUnscaled** của lớp Graphics

```
protected override void OnPaint(PaintEventArgs e)
{
    string fileName = "D:\\Demo.jpg";
    Graphics g = e.Graphics;
    Image curImage = Image.FromFile(fileName);
    Rectangle rect = new Rectangle(20, 20, 100, 100);
    g.DrawImage(curImage, rect);
}
```

Các công cụ vẽ nâng cao

Image

- Lớp Image còn cung cấp phương thức **RotateFlip** dùng để quay và flip (lật) ảnh.
- **RotateFlipType** xác định kiểu quay của đối tượng.

- RotateFlipType.**Rotate90FlipNone**: Quay 90 độ - Không lật
- RotateFlipType.**Rotate180FlipNone**: Quay 180 độ - Không lật
- RotateFlipType.**RotateNoneFlipX**: Lật theo chiều X - Không xoay
- RotateFlipType.**RotateNoneFlipY**: Lật theo chiều Y - Không xoay
- RotateFlipType.**Rotate90FlipX**: Quay 90 độ - Lật theo chiều X
- RotateFlipType.**Rotate180FlipY**: Quay 180 độ - Lật theo chiều Y
- RotateFlipType.**Rotate270FlipXY**: Quay 270 độ - Lật theo chiều X và Y

Bài tập trên lớp

Vẽ hình ngôi sao và tô màu vàng

Vẽ quốc kỳ Việt Nam