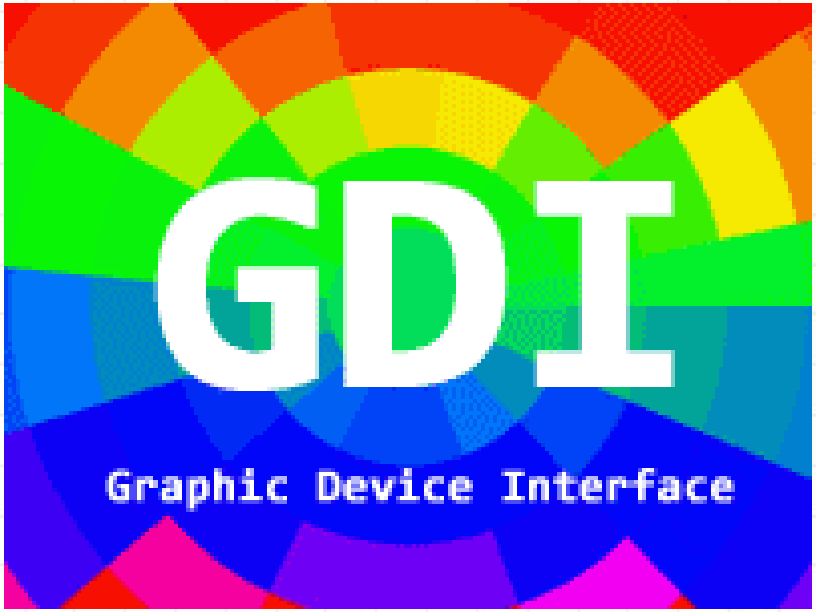
LẬP TRÌNH GDI+ # PHẦN 1

GIỚI THIỆU CƠ BẢN VỀ GDI+



Nội dung – Phần 1

1	Giới thiệu tổng quan về GDI+
2	Các bước thực hiện khi vẽ hình
3	Các lớp và cấu trúc cơ bản trong GDI+
4	Các công cụ vẽ cơ bản
5	Các hàm vẽ cơ bản

Giới thiệu tổng quan về GDI+

GDI (Graphic Device Interface) là gì?

GDI là một giao diện lập trình ứng dụng (API) của Windows, dùng để vẽ đồ họa (graphics), xử lý hiển thị (rendering) và tương tác với các thiết bị đầu ra: màn hình, máy in, file

GDI+ là gì?

GDI+ là phiên bản nâng cấp của GDI giúp giảm độ phức tạp và tăng tính linh hoạt trong việc vẽ các đối tượng của GDI

Các công cụ được cung cấp bởi GDI+ được đóng gói trong đâu?

namespace System.Drawing

Giới thiệu tổng quan về GDI+

Những đặc tính mới của GDI+ so với GDI?

- Hỗ trợ các tọa độ số thực (PointF, SizeF, RectangleF)
- Phối màu với giá trị alpha (Alpha Blending)
- Cung cấp tính trong suốt cho hình ảnh (image transparency)
- Làm mịn lề (antialiasing)
- Cung cấp những phép biến đổi hình
- Các loại cọ vẽ (brush) texture và gradient

Giới thiệu tổng quan về GDI+

Lịch sử phát triển của GDI:

- **1985 - 2000**: GDI là cốt lõi đồ họa Windows (API hàm C), nhanh nhưng đơn giản
- **2001 - 2006**: GDI+ ra mắt, hỗ trợ nhiều tính năng đồ họa đẹp hơn, tích hợp trong .NET, nhưng không tận dụng GPU
- **2006 - nay**: Microsoft chuyển sang Direct2D, DirectWrite, WPF, UWP, WinUI, SkiaSharp…

GDI+ chỉ còn dùng cho ứng dụng cũ, chủ yếu duy trì tương thích

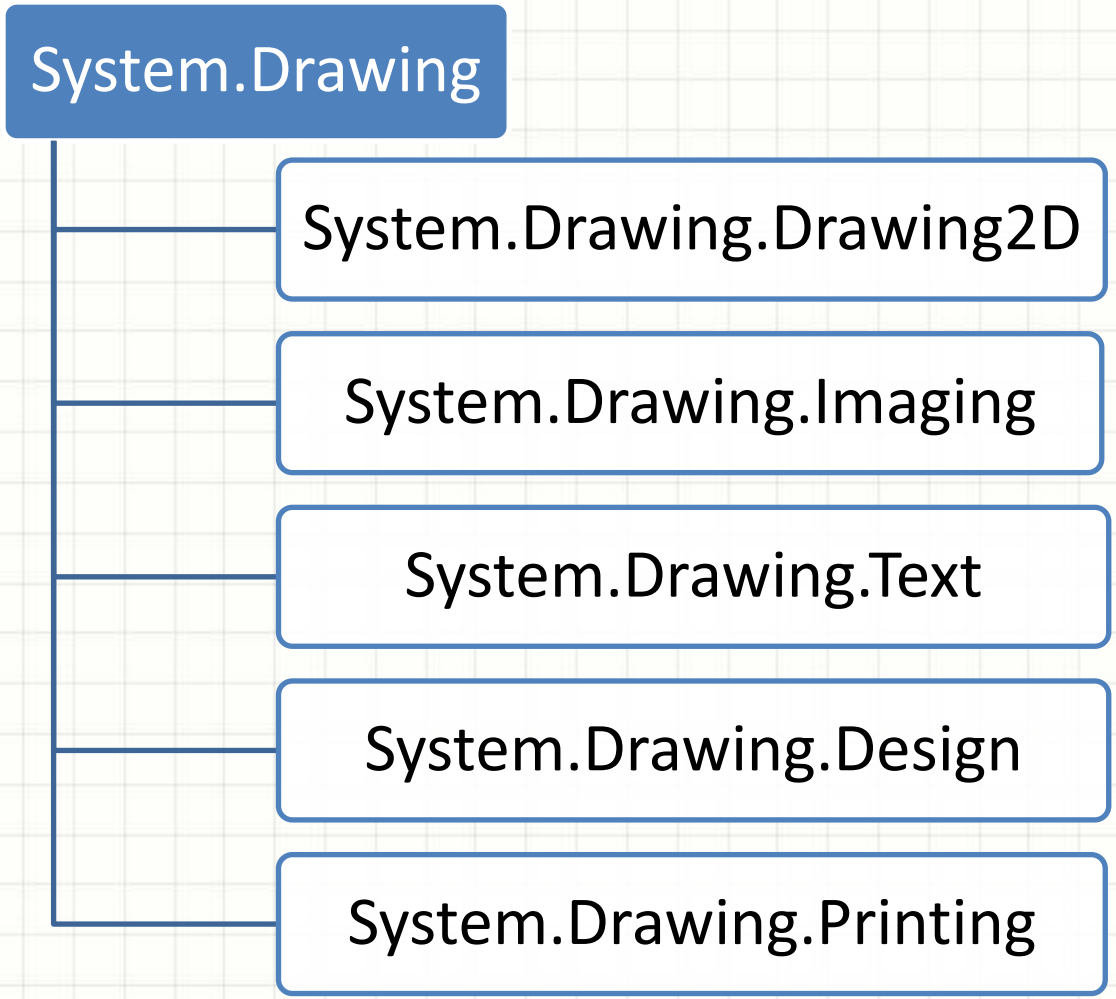
Giới thiệu tổng quan về GDI+

GDI+ cung cấp 3 nhóm dịch vụ chính:

- 2D vector graphics: cho phép tạo hình từ các hình cơ bản (primitive): đường thẳng, tròn, ellipse, đường cong,…
- Imaging: liên quan đến xem và xử lý hình ảnh
- Typography: vẽ chữ

Giới thiệu tổng quan về GDI+

Các namespaces trong GDI+:



Giới thiệu tổng quan về GDI+

- **System.Drawing**

Là cốt lõi của GDI+, gồm: Font, Pen, Brush,… trong đó Graphics là thành phần quan trọng nhất.

- **System.Drawing.Drawing2D**

Cung cấp các đối tượng đồ họa 2 chiều và các phép biến đổi hình học

- **System.Drawing.Imaging**

Xử lý hình ảnh đồ họa như thay đổi bảng màu, trích xuất siêu dữ liệu hình ảnh,…

- **System.Drawing.Text**

Trang trí văn bản, sử dụng tập hợp phông chữ,…

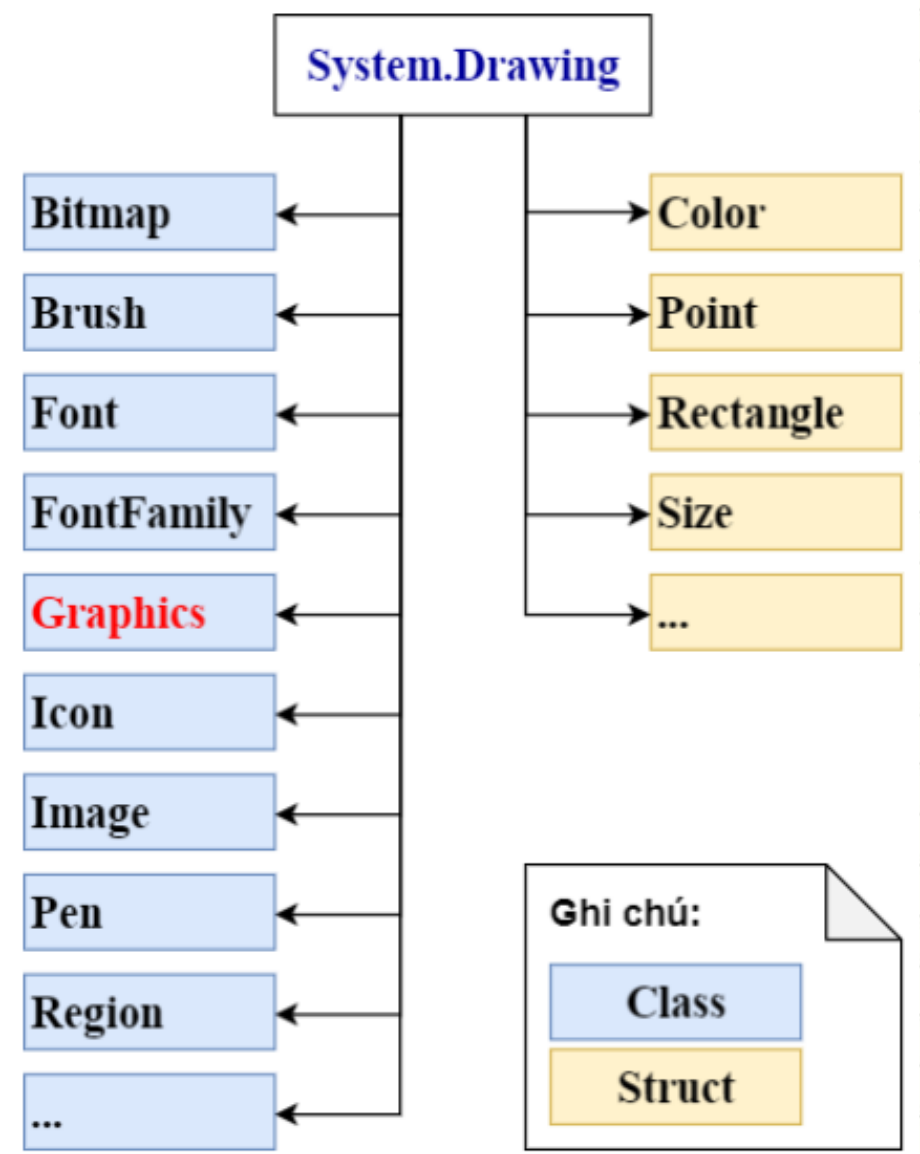
- **System.Drawing.Design**

Cung cấp chức năng bổ sung để phát triển các controls: toolbox items, graphics editors.

- **System.Drawing.Printing**

Hiển thị hình ảnh cho trang in, tương tác với máy in và định dạng giao diện tổng thể của lệnh in

Giới thiệu tổng quan về GDI+



Giới thiệu tổng quan về GDI+

System.Drawing

Bitmap: Làm việc với hình ảnh với dữ liệu pixel.

Brush: Các mẫu tô nền.

Font: Định dạng font chữ, kích thước, kiểu font.

FontFamily: Tập hợp các dạng font chữ.

Graphics: Các đối tượng đồ họa cơ sở của GDI+.

Icon: Ảnh bitmap nhỏ đại diện cho một đối tượng.

Image: Lớp abstract cho Bitmap và Metafile.

Pen: Sử dụng để vẽ đường thẳng và đường cong.

Region: “Phần ruột” của các khuôn hình học.

Các bước thực hiện khi vẽ hình

1. Xác định bề mặt, phạm vi muốn vẽ (**canvas**)
2. Chọn công cụ để vẽ (**pen** hoặc **brush**)
3. Tiến hành vẽ

Các bước thực hiện khi vẽ hình

Xác định canvas:

- Chiều rộng, chiều cao: xác định vị trí, kích thước bề mặt để vẽ
- Độ phân giải: số điểm ảnh theo chiều ngang và chiều dọc của màn hình
- Độ sâu màu: số lượng màu sắc được sử dụng cho mỗi điểm ảnh
- Điểm ảnh (pixel): là đơn vị nhỏ nhất tham gia vào quá trình hiển thị đối tượng, gồm 3 thành phần đỏ, xanh lá, xanh dương (RGB)
- 3 canvas thông dụng: form, printer, bitmap

Các bước thực hiện khi vẽ hình

Chọn công cụ vẽ:

- Bút vẽ (Pen)
- Cọ vẽ (Brush)
- Phông chữ (Font)
- Màu sắc (Color)

Các bước thực hiện khi vẽ hình

Tiến hành vẽ:

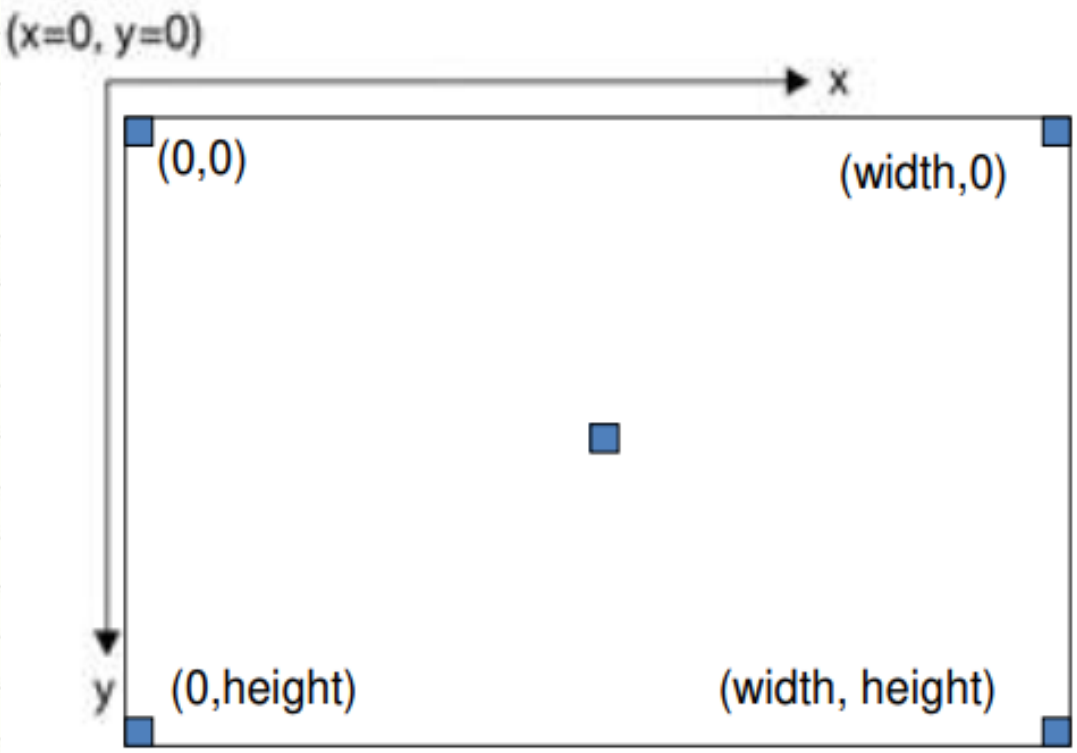
- Phải tạo ra hoặc sử dụng đối tượng **Graphics** có sẵn
- Gọi các phương thức thích hợp trên đối tượng đó

  Draw…

  Fill…

Các bước thực hiện khi vẽ hình

Hệ tọa độ:



Các lớp và cấu trúc cơ bản trong GDI+

Color

❖ Là một cấu trúc dữ liệu thể hiện màu sắc, là sự kết hợp giữa 4 giá trị:

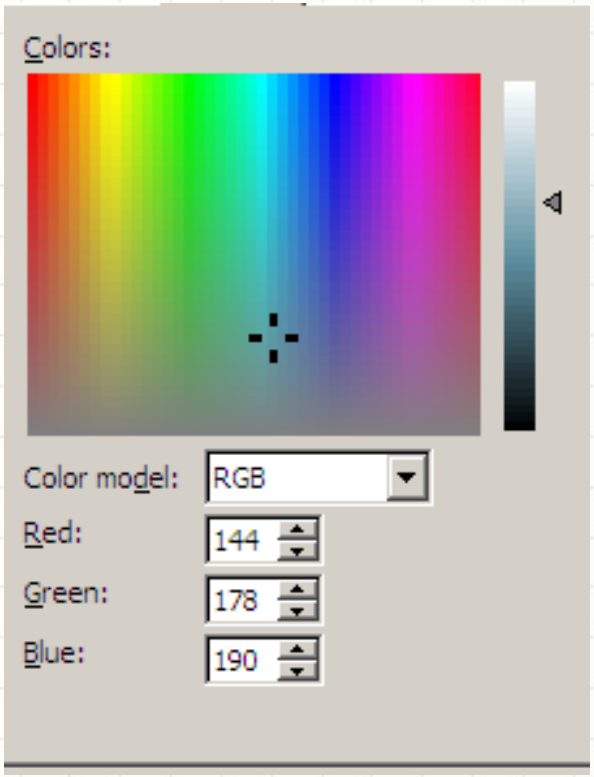
– R: Red

– G: Green

– B: Blue

– A: Alpha: độ trong suốt của màu

❖ Giá trị mỗi thành phần từ 0-255



Các lớp và cấu trúc cơ bản trong GDI+

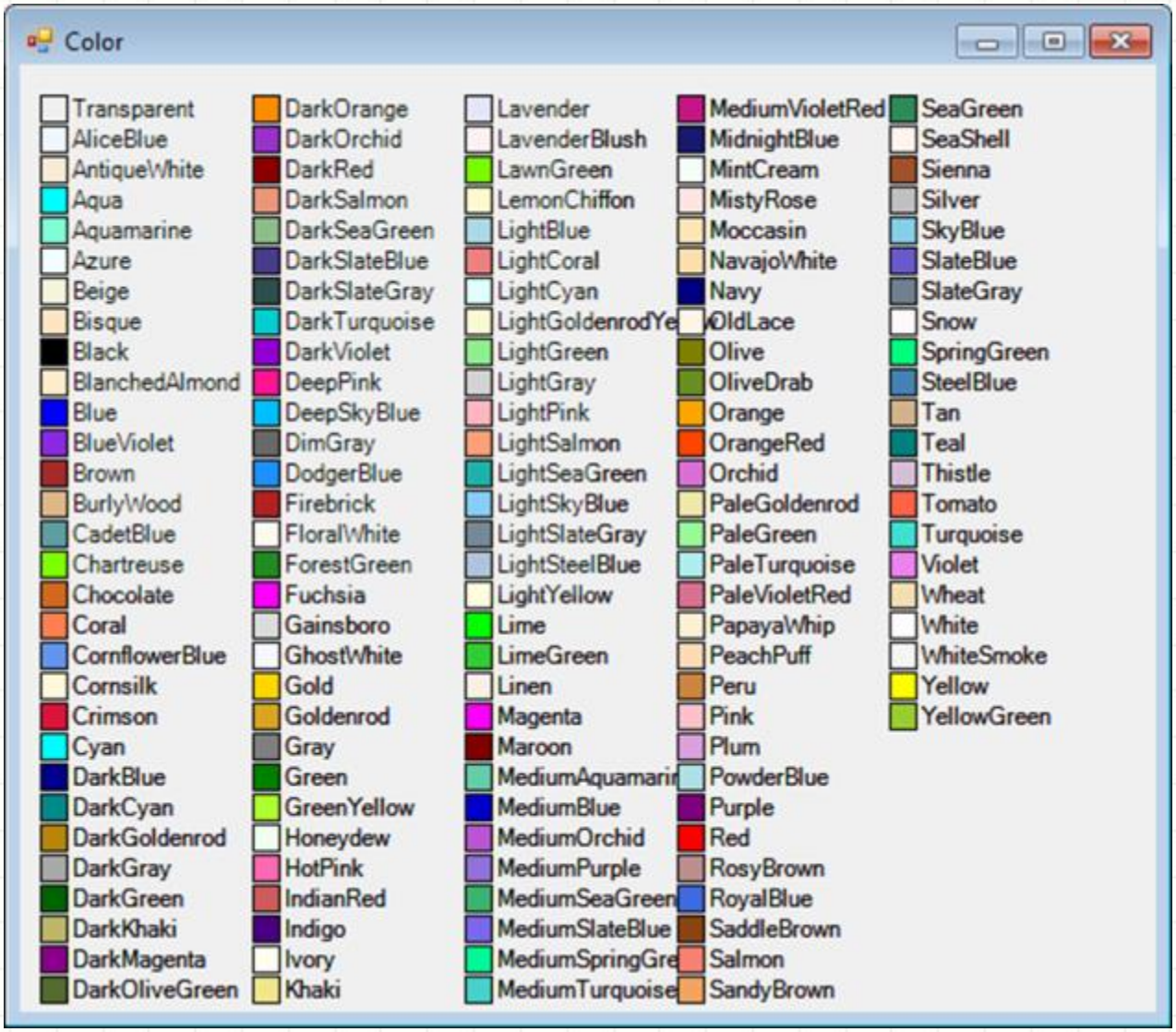
Color

Thuộc tính	Miêu tả
A	Trả về giá trị alpha của màu
R	Trả về giá trị của sắc màu đỏ
G	Trả về giá trị của sắc màu xanh lá cây
B	Trả về giá trị của sắc màu xanh dương
IsEmpty	Xác dịnh xem màu có được tạo
IsKnownColor	Xác dịnh xem màu có được xác định trước

Phương Thức	Mô tả
FromArgb	Tạo màu sắc từ các giá trị 8bit alpha, red, green, blue
GetBrightness	Trả về giá trị độ sáng của cấu trúc Color
GetHue	Trả về giá trị Hue của cấu trúc Color
GetSaturation	Trả về giá trị Saturation
ToArgb	Trả về giá trị 32bit của cấu trúc Color

Các lớp và cấu trúc cơ bản trong GDI+

Color



Các lớp và cấu trúc cơ bản trong GDI+

Point, Rectangle, Size, Region

Point, PointF	X,Y +, -, ==, !=, IsEmpty
Rectangle, RectangleF	X,Y Top, Left, Botton, Right Height, Width Inflate(), IntersSec,() Union() Contain()
Size, SizeF	+, -, ==, != Height, Width
Region	“phần ruột” của khuôn hình học Rectangle rect=new Rectangle(0,0,100,100) Region rgn= new Region(rect)

Các lớp và cấu trúc cơ bản trong GDI+

Graphics

- Đây là lớp quan trọng nhất của GDI+.
- Mọi thao tác vẽ đều thực hiện trên đối tượng Graphics của lớp này.
- Bất kỳ control nào cũng đều có thuộc tính Graphics dùng để vẽ chính nó.
- Không thể tạo đối tượng Graphics bằng toán tử **new**.
- Có thể lấy đối tượng Graphics từ tham số **PaintEventArgs** của sự kiện **Paint** của form hoặc từ phương thức **OnPaint** của form.
- Hoặc có thể lấy đối tượng Graphics thông qua hàm **CreateGraphics()**. Ảnh vẽ sẽ mất đi khi form được Reload.
- Hoặc có thể lấy đối tượng Graphics từ các hàm tĩnh (**static**) của lớp Graphics như **FromImage**

Các lớp và cấu trúc cơ bản trong GDI+

Graphics

Lấy đối tượng Graphics từ tham số PaintEventArgs của sự kiện Paint của form

```
private void Form1_Paint(object sender,
    PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Pen pen = new Pen(Color.Red);
    g.DrawLine(pen, 0, 0, 200, 200);
}
```

Các lớp và cấu trúc cơ bản trong GDI+

Graphics

Lấy đối tượng Graphics từ tham số PaintEventArgs từ phương thức OnPaint của form

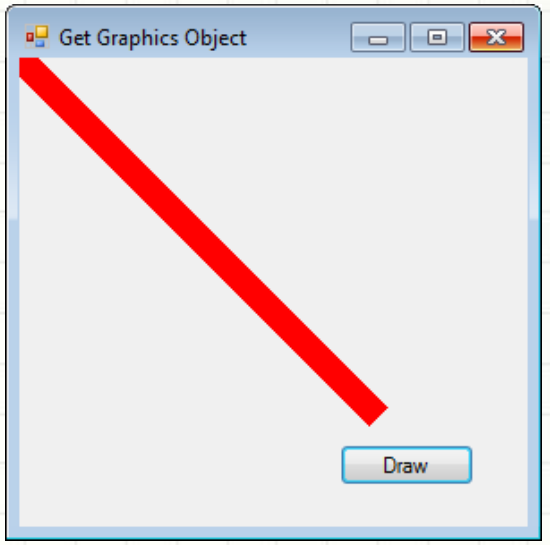
```
// Override OnPaint
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g  = e.Graphics;
    Pen pen = new Pen(Color.Red);
    g.DrawLine(pen,0,0,100,100);
}
```

Các lớp và cấu trúc cơ bản trong GDI+

Graphics

Lấy đối tượng Graphics thông qua hàm CreateGraphics(), hàm này trả về một đối tượng Graphics. Ảnh vẽ sẽ mất đi khi Form được Reload.

```
private void button1_Click(object sender, EventArgs e)
{
    Graphics g = this.CreateGraphics();
    Pen pen = new Pen(Color.Red, 15);
    g.DrawLine(pen, 0, 0, 200, 200);
    g.Dispose();
}
```



Khi tạo bằng CreateGraphics, nhớ Dispose

Các công cụ vẽ cơ bản

Brushes, Brush

- Được sử dụng để tô màu phần bên trong một hình hay để tô màu văn bản
- Thường được dùng kết hợp với các phương thức Fill…
- Lớp Brushes là lớp không được kế thừa, cung cấp hơn 141 thuộc tính tô màu chuẩn

Các công cụ vẽ cơ bản

Brushes, Brush

Có thể tạo một brush qua Brushes từ 141 màu sắc thông thường

Brush b1 = Brushes.Brown;

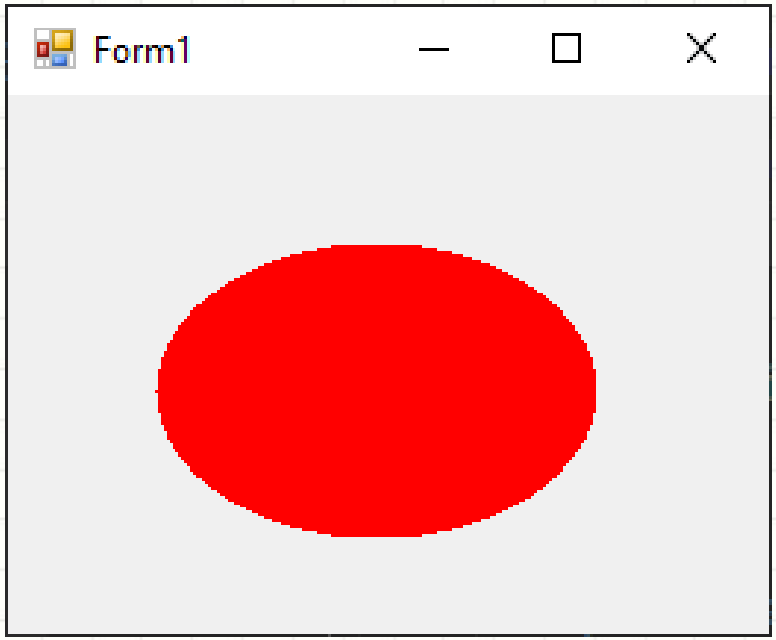
Brush b2 = Brushes.Acqua;

```
namespace System.Drawing
{
    public sealed class Brushes
    {
        public static Brush AliceBlue { get; }
        public static Brush AntiqueWhite { get; }
        public static Brush Aqua { get; }
        public static Brush Aquamarine { get; }
        public static Brush Azure { get; }
        public static Brush Beige { get; }
        public static Brush Bisque { get; }
        public static Brush Black { get; }
```

Các công cụ vẽ cơ bản
Brushes, Brush

Ví dụ dùng Brushes để tô một hình ellipse màu đỏ

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.FillEllipse(Brushes.Red, 50, 50, 150, 100);
}
```



Các công cụ vẽ cơ bản

Pens, Pen

- Được sử dụng để vẽ đường thẳng, đường cong, đường viền cho các đối tượng
- Thường được dùng kết hợp với các phương thức Draw…
- Lớp Pens là lớp không được kế thừa

Các công cụ vẽ cơ bản

Pens, Pen

Có thể tạo một pen qua Pens từ 141 màu sắc thông thường và độ dày 1 pixel

Pen p1 = Pens.AliceBlue;

Pen b2 = Pens.Bisque;

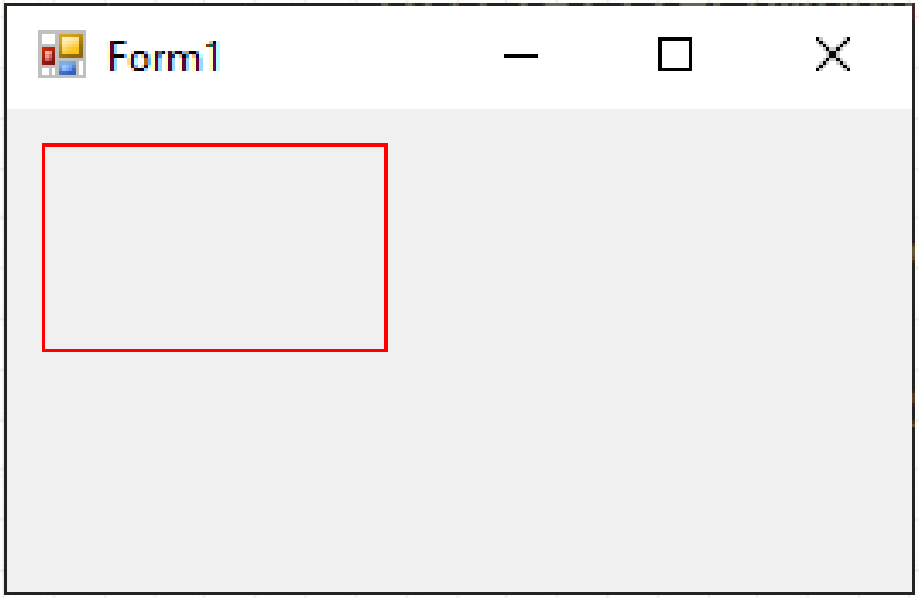
```
namespace System.Drawing
{
    public sealed class Pens
    {
        public static Pen AliceBlue { get; }
        public static Pen AntiqueWhite { get; }
        public static Pen Aqua { get; }
        public static Pen Aquamarine { get; }
        public static Pen Azure { get; }
        public static Pen Beige { get; }
        public static Pen Bisque { get; }
        public static Pen Black { get; }
        public static Pen BlanchedAlmond { get; }
```

Các công cụ vẽ cơ bản

Pens, Pen

Ví dụ dùng Pens để vẽ một khung hình màu đỏ

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.DrawRectangle(Pens.Red, 10, 10, 100, 60);
}
```



Các công cụ vẽ cơ bản

Pens, Pen

Đối tương Pen có thể được khởi tạo dựa vào các phương thức khởi tạo (constructor) sau:

- Pen (Brush brush) : tạo đối tượng Pen từ đối tượng Brush, với độ dày nét vẽ là 1 pixel
- Pen (Color color): tạo đối tượng Pen từ đối tượng Color, với độ dày nét vẽ là 1 pixel
- Pen(Brush brush, float width): tạo đối tượng Pen từ đối tượng Brush, với độ dày nét vẽ là width
- Pen(Color color, float width): tạo đối tượng Pen từ đối tượng Color, với độ dày nét vẽ là width

Các hàm vẽ cơ bản

- Vẽ đường thẳng qua 2 điểm:

DrawLine(Pen, Điểm1, Điểm2);

DrawLine(Pen, Điểm1.X, Điểm1.Y, Điểm2.X, Điểm2.Y);

- Vẽ đường gấp khúc qua n điểm:

DrawLines(Pen, Mảng chứa các điểm);

- Vẽ đường Polygon:

DrawPolygon(Pen, Mảng chứa các điểm);

- Vẽ đường Ellipse:

DrawEllipse(Pen, Hình chữ nhật mà Elip nội tiếp);

DrawEllipse(Pen, X, Y, Chiều rộng, Chiều cao);

Các hàm vẽ cơ bản

- Vẽ hình quạt:

DrawPie(Pen, Hình chữ nhật mà đường tròn nội tiếp, Góc bắt đầu, Góc quét);

- Vẽ cung tròn:

DrawArc(Pen, Hình chữ nhật mà đường tròn nội tiếp, Góc bắt đầu, Góc quét);

- Vẽ hình chữ nhật:

DrawRectangle(Pen, Hình chữ nhật);

DrawRectangle(Pen, X, Y, Chiều rộng, Chiều cao);

- Vẽ đường cong chính tắc bất kỳ:

DrawCurve(Pen, Mảng các điểm);

DrawCurve(Pen, Mảng các điểm, Offset, Số phân đoạn, Độ căng);

Bài tập trên lớp