

# LẬP TRÌNH GIAO DIỆN

Nguyễn Thị Mai Trang

Nguyễn Thị Mai Trang

1

1

## Chương 8

# Đồ họa

2

## MỤC TIÊU

- Trình bày được các thành phần quan trọng trong thư viện GDI+ được sử dụng để vẽ
- Sử dụng các đối tượng thuộc lớp Graphics để vẽ đường, vẽ và tô màu các đối tượng hình học, hiển thị văn bản và hình ảnh.
- Xây dựng được ứng dụng vẽ bằng chuột

## NỘI DUNG

1. Giới thiệu về GDI+
2. Một số đối tượng GDI+
3. Lớp Graphics
4. Làm việc với các đối tượng đồ họa
5. Vẽ, tô màu các đối tượng cơ bản

## 8.1 Giới thiệu về GDI+

- Được phát triển từ GDI (**G**raphics **D**evice **I**nteface), chỉ có trong .NET Framework.
- GDI+ là một thư viện cung cấp các lớp cho phép làm việc với các đối tượng đồ họa 2D như:
  - Vẽ, tô màu đối tượng
  - Hiển thị văn bản (vẽ chữ)
  - Vẽ hình ảnh, biến đổi hình ảnh.
  - Thư viện .NET Framework chứa các lớp liên quan đến thao tác vẽ trong không gian tên System.Drawing.

Nguyễn Thị Mai Trang

5

5

## Giới thiệu về GDI+ (tt)

- GDI+ cung cấp các lớp thư viện cho phép thực hiện tiến trình vẽ, bao gồm các bước:
  - Xác định phạm vi (bề mặt) để vẽ
    - Hệ thống tọa độ
    - Các cấu trúc dữ liệu như Rectangle, Point, Size
  - Tạo các công cụ để vẽ
    - Cọ tô (Brush)
    - Bút vẽ (Pen)
    - Phong chữ (Font)
  - Thực hiện thao tác vẽ, tô màu
    - Lớp Graphics

Nguyễn Thị Mai Trang

6

6

## Xác định phạm vi để vẽ

- 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)

Nguyễn Thị Mai Trang

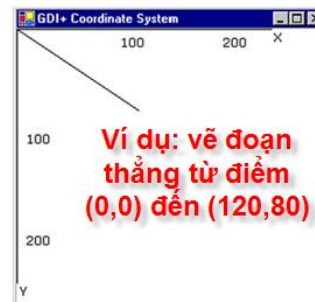
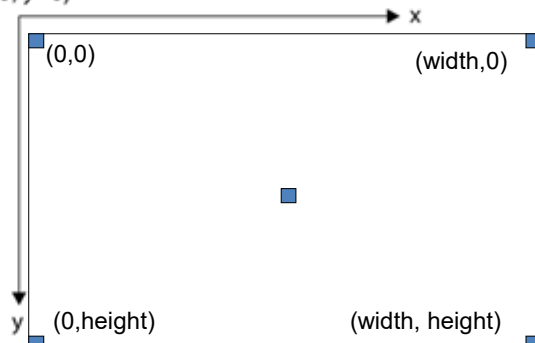
7

7

## Xác định phạm vi để vẽ

- Hệ thống tọa độ (Coordinate System)

(x=0, y=0)



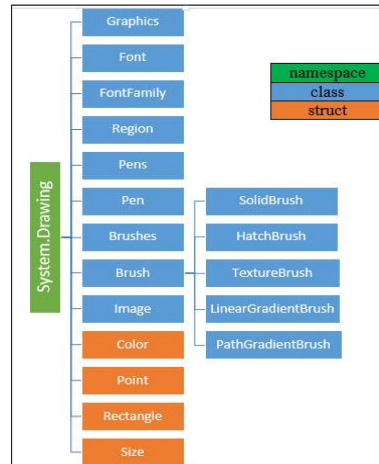
Nguyễn Thị Mai Trang

8

8

## Tạo các công cụ để vẽ

- Cọ tô (Brush)
- Bút vẽ (Pen)
- Phong chữ (Font)
- Màu sắc (cấu trúc Color)



Nguyễn Thị Mai Trang

9

9

## Thực hiện các thao tác vẽ và tô màu

- Sử dụng các phương thức thuộc lớp Graphics
  - Draw...
  - Fill...

FontConverter.FontUnitConverter Class

FontFamily Class

FontStyle Enumeration

Graphics Class

Graphics Methods

AddMetafileComment Method

BeginContainer Method

Clear Method

CopyFromScreen Method

Dispose Method

DrawArc Method

DrawBezier Method

DrawBeziers Method

DrawClosedCurve Method

DrawCurve Method

DrawEllipse Method

Graphics Class

.NET Framework (current version) | Other Versions

Encapsulates a GDI+ drawing surface. This class cannot be inherited.

Namespace: System.Drawing

Assembly: System.Drawing (in System.Drawing.dll)

Inheritance Hierarchy

System.Object

System.MarshalByRefObject

System.Drawing.Graphics

Syntax

C# C++ F# VB

public sealed class Graphics : MarshalByRefObject, IDrawingContext

Properties

Nguyễn Thị Mai Trang

10

10

## 8.2 Một số đối tượng GDI

- Color
- Point
- Rectangle
- Size

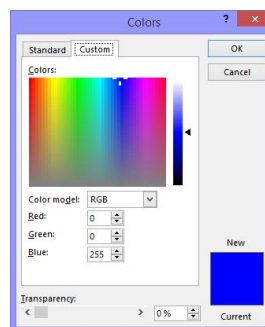
Nguyễn Thị Mai Trang

11

11

## 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





Nguyễn Thị Mai Trang

12

12

## Color

- Tạo đối tượng Color từ các giá trị ARGB: sử dụng phương thức `Color.FromArgb`
  - Ví dụ: tạo đối tượng Color màu đỏ   
`Color red = Color.FromArgb(255, 0, 0);`
    - Lưu ý: nếu ta bỏ qua giá trị A, mặc định A = 255
- Tạo đối tượng Color từ một chuỗi tên màu xác định: sử dụng phương thức `Color.FromName`
  - Ví dụ: tạo đối tượng Color màu xanh dương   
`Color blue = Color.FromName ("Blue");`

Nguyễn Thị Mai Trang

13

13

## Point

- Là cấu trúc dữ liệu xác định một điểm trong mặt phẳng với hai thuộc tính x, y
- Có thể tạo một đối tượng Point thông qua các phương thức khởi tạo sau:
  - `public Point(int);`
    - ví dụ: `Point pt1 = new Point(10);`
  - `public Point(Size);`
    - ví dụ: `Point pt2 = new Point( new Size(20, 20) );`
  - `public Point(int, int);`
    - ví dụ: `Point pt3 = new Point(30, 30);`

Nguyễn Thị Mai Trang

14

14

## Rectangle

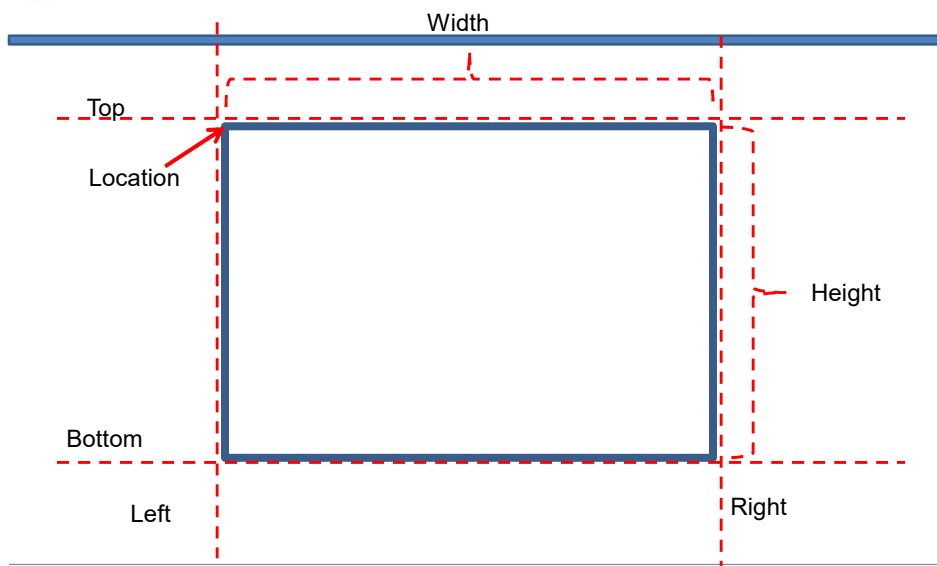
- Là một cấu trúc dữ liệu xác định vị trí, kích thước một vùng hình chữ nhật trong mặt phẳng
- Các thuộc tính của Rectangle:
  - **Left, Right**: giá trị x của tọa độ điểm bên trái và phải của HCN
  - **Top, Bottom**: giá trị y của tọa độ điểm bên trên và dưới của HCN
  - **Width, Height**: chiều rộng, cao của HCN
  - **X, Y**: giá trị x, y (int) của tọa độ điểm góc trên bên trái của HCN
  - **Location**: tọa độ điểm (Point) góc trên bên trái của HCN
  - **IsEmpty**: kiểm tra HCN rỗng (tọa độ 4 điểm đều bằng 0: True)
  - **Size**: trả về đối tượng Size cho biết kích thước của HCN

Nguyễn Thị Mai Trang

15

15

## Rectangle



Nguyễn Thị Mai Trang

16

16



## Rectangle (tt)

- Các phương thức
  - Contains: kiểm tra một điểm có nằm trong HCN?
  - Inflate: thay đổi kích thước HCN theo các chiều
  - Offset: thay đổi vị trí HCN
  - Ceiling: chuyển RectangleF → Rectangle với các giá trị tọa độ được làm tròn lên
  - Round: chuyển RectangleF → Rectangle với các giá trị tọa độ được làm tròn
  - Truncate: chuyển RectangleF → Rectangle với các giá trị tọa độ được làm tròn xuống

17

## Size

- Là một cấu trúc dữ liệu xác định kích thước một vùng hình chữ nhật, với hai thuộc tính Width, Height
- Ví dụ:
  - Size sz = new SizeF(100,80);

18

## 8.3 Lớp Graphics

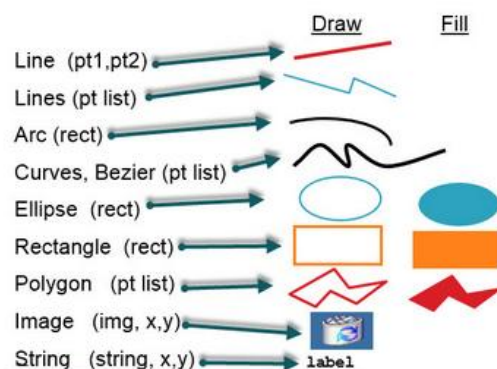
- Là thành phần chính của GDI+, cung cấp các tài nguyên và phương pháp thao tác với các đối tượng đồ họa
- **Một số thuộc tính lớp Graphics**
  - **Clip**: get/set phạm vi bản vẽ
  - **ClipBounds**: trả về cấu trúc Rectangle là phạm vi bản vẽ
  - **DpiX, DpiY**: trả về độ phân giải của đối tượng đồ họa tính theo inch
  - **PageUnit**: get/set đơn vị hệ thống tọa độ
  - **SmoothingMode**: get/set chế độ làm mịn của đối tượng đồ họa:
    - AntiAlias: chống răng cưa
    - HighQuality: chất lượng cao
    - HighSpeed: tốc độ cao

## Lớp Graphics (tt)

- **Tạo đối tượng Graphics**
  - Sử dụng thuộc tính Graphics được truyền cho **Paint ()**
  - Sử dụng phương thức **CreateGraphics()** của form, control
  - Lấy từ đối tượng dẫn xuất từ Bitmap để vẽ trên hình.
  - (Xem các ví dụ trong tài liệu học tập Lập trình giao diện)

## Lớp Graphics (tt)

### • Một số phương thức vẽ và tô màu của lớp Graphics



Nguyễn Thị Mai Trang

21

21

## 8.4 Làm việc với các đối tượng đồ họa

- Brushes
- Pens
- Font

Nguyễn Thị Mai Trang

22

22

## 8.4.1 Brushes

- Brushes là đối tượng được sử dụng để tô màu vùng bên trong của hình, tô màu văn bản, thường được kết hợp trong các phương thức **Fill...** của lớp Graphics
- Được định nghĩa trong không gian tên
  - System.Drawing: Brushes, SolidBrush
  - System.Drawing.Drawing2D:
    - HatchBrush
    - TextureBrush
    - LinearGradientBrush
    - PathGradientBrush

Nguyễn Thị Mai Trang

23

23

## Brushes (tt)

- Bao gồm các lớp:
  - Brushes
  - Brush
    - SolidBrush
    - HatchBrush
    - TextureBrush
    - LinearGradientBrush
    - PathGradientBrush

Nguyễn Thị Mai Trang

24

24

## class Brushes

- Là một lớp không được kế thừa (sealed class)

```
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; }
    }
}
```

Nguyễn Thị Mai Trang

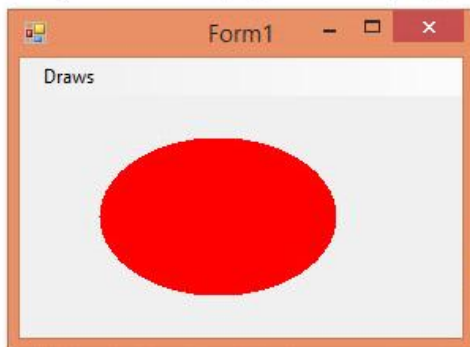
25

25

## Brushes (tt)

- Ví dụ sử dụng class Brushes: tô màu đỏ cho một ellipse

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



```
ng mode
s;
) to (100, 100
00, 100);
```

Nguyễn Thị Mai Trang

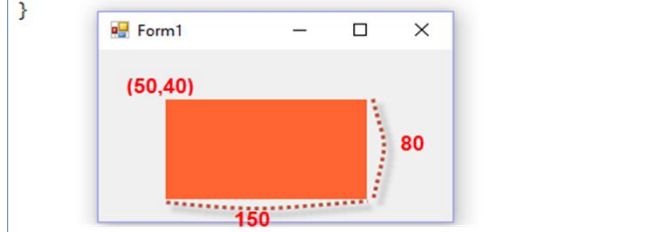
26

26

## Solid Brushes

- Là loại cọ tô một màu đồng nhất
- Tạo SolidBrush: sử dụng phương thức khởi tạo
- Ví dụ:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Color color = Color.FromArgb(255, 100, 50);
    SolidBrush brush = new SolidBrush(color);
    e.Graphics.FillRectangle(brush, 50, 40, 150, 80);
}
```



Nguyễn Thị Mai Trang

27

27

## Hatch Brushes

- Là loại cọ tô có hiệu ứng nền
- Hai phương thức khởi tạo:
  - public HatchBrush(HatchStyle, Color);
  - public HatchBrush(HatchStyle, Color, Color);
- Trong đó, HatchStyle là các kiểu nền

Nguyễn Thị Mai Trang

28

28

## Hatch Brushes (tt)

- HatchStyle:



Nguyễn Thị Mai Trang

29

29

## Texture Brushes

- Là loại cọ tô có nền là một đối tượng Image
- Lớp này có các thuộc tính:
  - Image: ảnh nền
  - WrapMode: Clamp, Tile, TileFlipX, TileFlipY, TileFlipXY

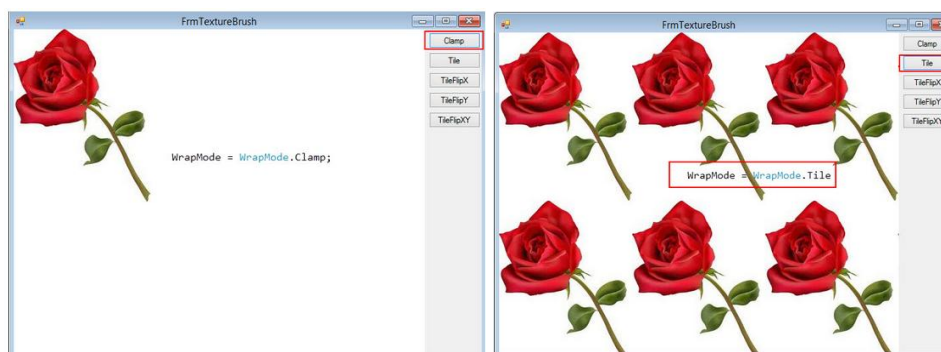
Nguyễn Thị Mai Trang

30

30

## Texture Brushes (tt)

- Các kiểu WrapMode



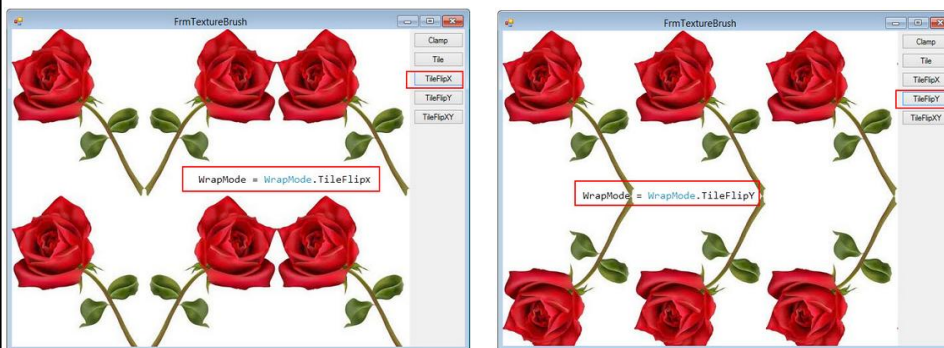
Nguyễn Thị Mai Trang

31

31

## Texture Brushes (tt)

- Các kiểu WrapMode



Nguyễn Thị Mai Trang

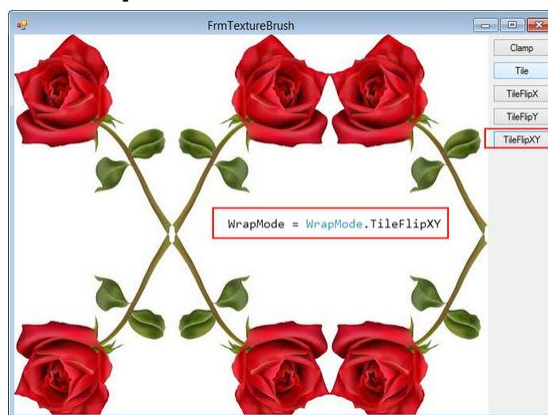
32

32



## Texture Brushes (tt)

- **Các kiểu WrapMode**



Nguyễn Thị Mai Trang

33

33

## Linear Gradient Brushes

- Là loại cọ tô pha trộn giữa hai màu
- Các phương thức khởi tạo:
  - LinearGradientBrush (Point, Point, Color, Color)
  - LinearGradientBrush (PointF, PointF, Color, Color)
  - LinearGradientBrush (Rectangle, Color, Color, LinearGradientMode)
  - LinearGradientBrush (Rectangle, Color, Color, Single)
  - LinearGradientBrush (Rectangle, Color, Color, Single, Boolean)
  - LinearGradientBrush (RectangleF, Color, Color, LinearGradientMode)
  - LinearGradientBrush (RectangleF, Color, Color, Single)
  - LinearGradientBrush (RectangleF, Color, Color, Single, Boolean)

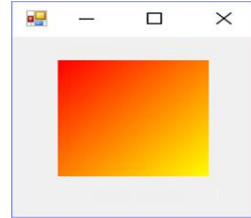
Nguyễn Thị Mai Trang

34

34

## Linear Gradient Brushes

- Trong đó:
  - LinearGradientMode là hướng pha trộn, gồm các giá trị sau:
    - Horizontal
    - Vertical
    - BackwardDiagonal
    - ForwardDiagonal
  - angle: góc pha trộn.
  - Point, PointF, Rectangle, RectangleF: kích thước của brush



Nguyễn Thị Mai Trang

35

35

## Path Gradient Brushes

- Là loại cọ tô được tạo từ một đối tượng GraphicsPath, có thể pha trộn nhiều màu



Nguyễn Thị Mai Trang

36

36

## 8.4.2 Pens

- Pen - còn gọi là bút vẽ, là đối tượng được dùng để vẽ đường thẳng, đường cong, đường viền cho các đối tượng,
- Pen thường được kết hợp trong các phương thức **Draw...** của lớp Graphics
- Có hai lớp bút vẽ: Pens và Pen.

Nguyễn Thị Mai Trang

37

37

## Pens (tt)

- Class Pens: là một lớp không cho phép kế thừa, cung cấp loại bút vẽ có độ dày bằng 1 pixel với các màu chuẩn

```
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; }
    }
}
```

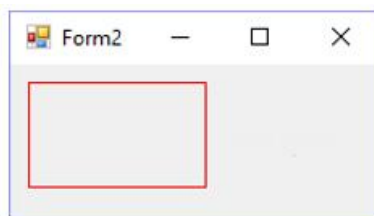
Nguyễn Thị Mai Trang

38

38

## Pens (tt)

- Class Pens:
  - Ví dụ sử dụng class Pens để vẽ một khung hình chữ nhật màu đỏ:



Nguyễn Thị Mai Trang

39

39

## Pen (tt)

- class Pen: cho phép tạo đối tượng Pen dựa vào các phương thức khởi tạo sau:
  - public Pen ([Brush](#) brush)
  - public Pen ([Color](#) color)
  - public Pen( [Brush](#) brush, float width)
  - public Pen( [Color](#) color, float width)

Nguyễn Thị Mai Trang

40

40

### 8.4.3 Fonts

- Windows hỗ trợ hai loại phông chữ: GDI fonts và device fonts.
  - **Device fonts** có nguồn gốc từ các thiết bị đầu ra như màn hình, máy in,...
  - **GDI fonts** được lưu trữ trong các tập tin trên hệ thống, thường chứa trong thư mục Windows\Fonts.
- Mỗi font có tập tin riêng, ví dụ, Arial, Arial Black, Arial Bold, Arial Italic, Italic Arial Black, Arial Bold Italic, Arial Narrow, Italic Arial Narrow Bold, và Arial Narrow Italic,...

Nguyễn Thị Mai Trang

41

41

### Fonts (tt)



Nguyễn Thị Mai Trang

42

42

## Fonts (tt)

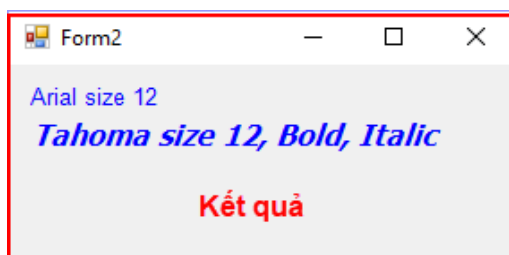
- Thư viện GDI+ cung cấp các lớp font cho phép thiết lập font chữ khi xuất văn bản là **Font** và **FontFamily**
- **class FontFamily**: cung cấp các loại font chữ được định nghĩa sẵn để sử dụng trong kết xuất văn bản, lớp này chứa các thuộc tính như
  - **Name**: tên font chữ
  - **Families**: mảng chứa tất cả các font families trong thiết bị ngữ cảnh

## Fonts (tt)

- **class Font**: là lớp cho phép tạo đối tượng font chữ với một số thuộc tính như kiểu chữ (đậm, nghiêng, gạch dưới,...), kích thước,...
  - Một số phương thức khởi tạo thường được sử dụng:
    - [Font \(FontFamily, Single\)](#)
    - [Font \(String, Single\)](#)
    - [Font \(FontFamily, Single, FontStyle\)](#)
    - [Font \(String, Single, FontStyle\)](#)

## Font (tt)

- Ví dụ sử dụng Font



Nguyễn Thị Mai Trang

45

45

## 8.5 Vẽ, tô màu các đối tượng

- Vẽ đường thẳng
  - Vẽ, tô màu các đối tượng hình học
    - Hình chữ nhật
    - Ellipse
    - Đa giác
  - Vẽ Image
  - Tô màu các đối tượng
  - Vẽ chữ (văn bản)
  - Vẽ bằng chuột
- ☞ Sử dụng các phương thức thuộc lớp Graphics

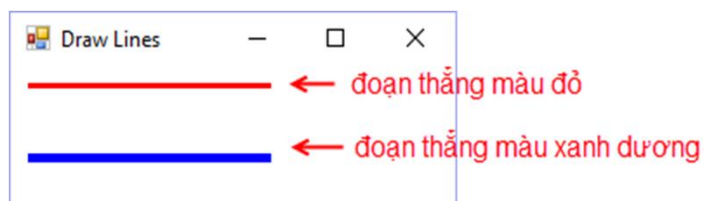
Nguyễn Thị Mai Trang

46

46

## Vẽ đường thẳng

- public static DrawLine (Pen pen, Point pStart, Point pEnd)
- public static DrawLine (Pen pen, int x1, int y1, int x2, int y2)



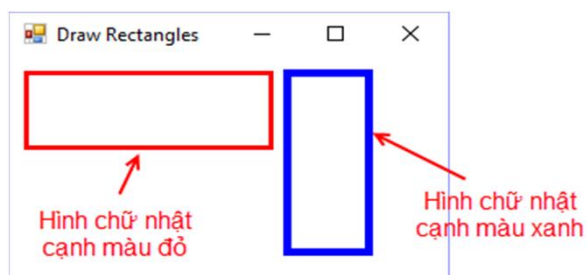
Nguyễn Thị Mai Trang

47

47

## Vẽ hình chữ nhật

- public static DrawRectangle (Pen pen, int x, int y, int width, int height)
- public static DrawRectangle (Pen pen, Rectangle rect)



Nguyễn Thị Mai Trang

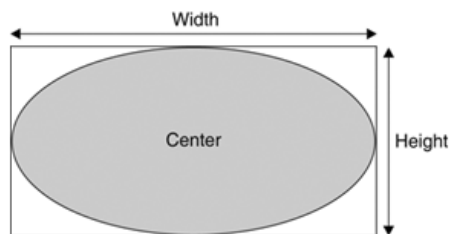
48

48



## Vẽ Ellipses và hình tròn

- Ellipse là một hình nội tiếp trong một vùng hình chữ nhật
- Để vẽ một Ellipse, cần xác định hình chữ nhật ngoại tiếp nó
- Ellipse nội tiếp trong một hình vuông là một hình tròn



Nguyễn Thị Mai Trang

49

49

## Vẽ Ellipses và hình tròn

- `public static DrawEllipse (Pen pen, int x, int y, int width, int height)`
- `public static DrawEllipse (Pen pen, Rectangle rect)`



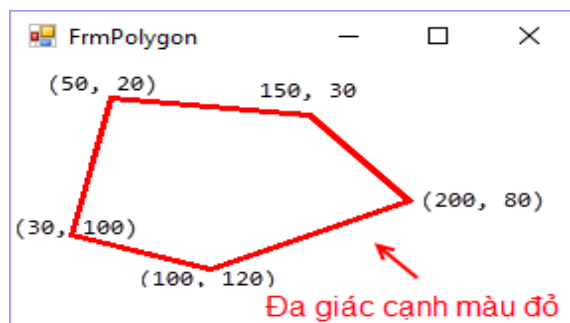
Nguyễn Thị Mai Trang

50

50

## Vẽ đa giác

- public static DrawPolygon( [Pen](#) pen, [Point](#) [ ] points )



Nguyễn Thị Mai Trang

51

51

## Vẽ Image

- Net Framework cung cấp lớp Image cho phép trình bày hình ảnh.
- Một số phương thức thông dụng của lớp Image:
  - **FromFile (string)**: phương thức static cho phép tạo ảnh từ đường dẫn đến một file (bmp, jpeg, jpg, gif, png, ico,... )
  - **Save (string, ImageFormat)**: lưu ảnh thành file
  - **RotateFlip (RotateFlipType)**: xoay, lật ảnh.
    - Lưu ý: để sử dụng **ImageFormat**, ta cần phải khai báo namespace **System.Drawing.Imaging**
- Để vẽ Image, sử dụng Phương thức **DrawImage** của lớp Graphics

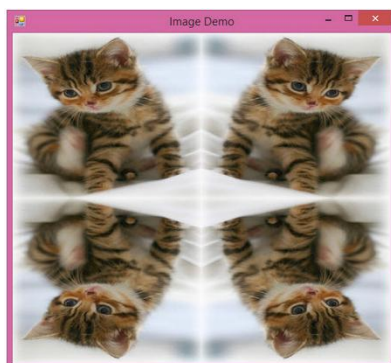
Nguyễn Thị Mai Trang

52

52

## Vẽ Image

- Ví dụ: xem tài liệu học tập Lập trình giao diện



Nguyễn Thị Mai Trang

53

53

## Tô màu các đối tượng

- Sử dụng các phương thức **Fill...** của lớp Graphics.



Nguyễn Thị Mai Trang

54

54

## Vẽ chữ

- Sử dụng phương thức DrawString
  - Vẽ chữ xuất phát tại một điểm:
    - DrawString (string, Brush, Font, Point)
    - DrawString (string, Brush, Font, Point, StringFormat)
    - Trong đó, **StringFormat** là đối tượng cho phép thiết lập các định dạng xuất của văn bản.

Nguyễn Thị Mai Trang

55

55

## Vẽ chữ (tt)

- FormatFlags : được dùng để canh chỉnh, chuyển hướng văn bản xuất.
- Giá trị của FormatFlags là một giá trị thuộc kiểu dữ liệu liệt kê **StringFormatFlags**
  - DirectionRightToLeft
  - DirectionVertical
  - NoClip
  - NoWrap



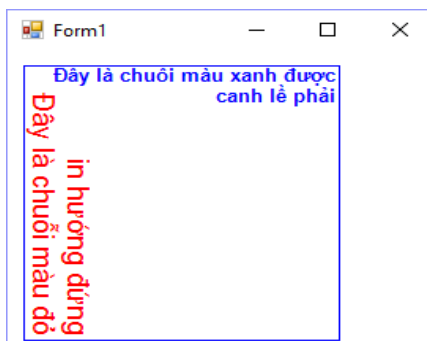
Nguyễn Thị Mai Trang

56

56

## Vẽ chữ (tt)

- Vẽ chữ trong khung chữ nhật:
  - DrawString (string, Brush, Font, Rectangle)
  - DrawString (string, Brush, Font, Rectangle, StringFormat)



Nguyễn Thị Mai Trang

57

57

## Vẽ bằng chuột

- Nắm bắt các sự kiệnMouseDown, MouseUp, MouseMove để thực hiện thao tác vẽ
  - Trong sự kiện **MouseDown**, lưu giữ vị trí điểm nhấn chuột
  - Trong sự kiện **MouseMove**, kiểm tra nếu như trong quá trình di chuyển chuột, người dùng có giữ nút trái chuột, mới thực hiện thao tác vẽ, (có thể kết hợp xử lý trong sự kiện MouseUp), sau đó, cập nhật lại vị trí điểm vẽ hiện tại

Nguyễn Thị Mai Trang

58

58