

CHƯƠNG 3: GDI – Graphical Device Interface

1

NGUYỄN THỊ LƯƠNG

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC ĐÀ LẠT

NỘI DUNG

2

- Graphical Device Interface
- Device Context
- Đối tượng Graphics
- Cursor
- Màu (Color)
- Bút vẽ (Pen)
- Hàm vẽ cơ bản:
 - Hình chữ nhật và hình vuông
 - Hình tròn và hình Elip
 - Đoạn thẳng
 - Đường gấp khúc
 - Pies
 - Vẽ cung tròn
 - Vẽ đường cong
- Brush (Cọ vẽ)

GDI

3

- Microsoft Windows là một hệ điều hành giao diện đồ họa, sử dụng các hình vẽ, hình ảnh, đoạn thẳng, màu sắc và nhiều biến thể khác để vận hành một cách ấn tượng các thiết bị vật lý.
- Để hỗ trợ khả năng trình bày hình ảnh và các hiệu ứng, tính năng khác, Microsoft cung cấp một thư viện gọi là Giao tiếp thiết bị đồ họa (Graphical Device Interface – GDI).
- Để đáp ứng các yêu cầu mới về kỹ thuật khi máy tính trở nên cần thiết, Microsoft nâng cấp GDI thành GDI+. Đây là thư viện đồ họa được sử dụng trong .Net Framework.

Device Context

4

- Là tất cả những phong nền (giấy, bảng, ...) và các thiết bị để vẽ
- Bao gồm cả kích cỡ của phong nền, sự định hướng, màu sắc và nhiều phụ tùng khác để bạn thỏa sức trình diễn ý tưởng của mình.

Namespace

5

- System.Drawing
- System.Drawing.Imaging

Đối tượng Graphics

6

- Để tạo đối tượng Graphics:
 - Lấy Graphics từ một Windows Control qua phương thức CreateGraphics
`public Graphics CreateGraphics();`
 - Lấy Graphics từ một Windows Control thông qua trình điều khiển Control
`public static Graphics FromHwnd(IntPtr hwnd);`
- Ví dụ:

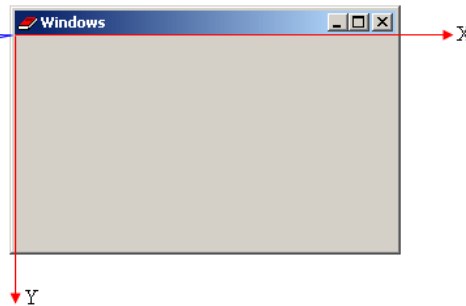

```
Graphics graph = Graphics.FromHwnd(this.Handle);
```
- Nếu đang vẽ trong hàm xử lý sự kiện Paint, sử dụng thuộc tính Graphics của đối số PaintEventArgs.

```
private void frmGDI_Paint(object sender, PaintEventArgs e)
{
    Graphics graph = e.Graphics;
    e.Graphics....
}
```

Đối tượng Graphics

7

- Cung cấp các phương thức vẽ các hình bắt đầu bằng chữ Draw....
- Đối số đưa vào các phương thức vẽ:
 - Đối tượng Pen,
 - Các đối số về kích thước của hình muốn vẽ,
 - Điểm bắt đầu vẽ (0, 0)
 - ...



Icon

8

- Sử dụng Icon:
 - Tạo Icon sử dụng phương thức:


```
Public Icon(string fileName);
```
 - Nếu đã có Icon muốn sử dụng với kích thước khác dùng:


```
Public Icon(Icon original, Size size);
```

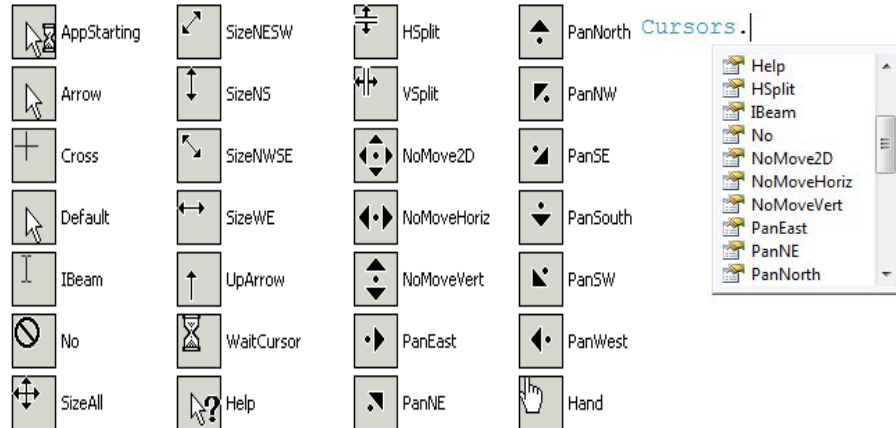
```
Public Icon(Icon original, int width, int height);
```
 - Ví dụ:


```
private void btnChangeIcon_Click(object sender, EventArgs e)
{
    Icon icon = new Icon("Icon1.ico");
    Icon icon1 = new Icon(icon, new Size(50, 50));
    this.Icon = icon1;
}
```

Cursor

9

- Sử dụng cursor có sẵn:



Cursor

10

- Tạo Cursor từ file:

- `public Cursor(string fileName);`

- Ví dụ:

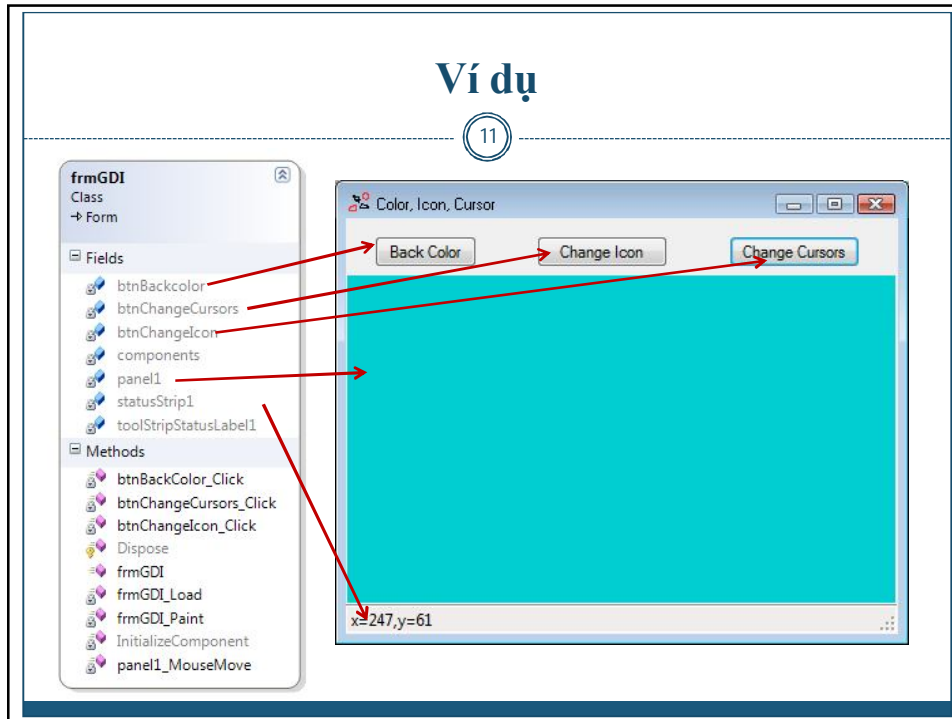
```
private void btnChangeCursors_Click(object sender, EventArgs e)
{
    Cursor cursorpaint = new Cursor("Cursor1.cur");
    panel1.Cursor = cursorpaint;
    this.Cursor = Cursors.Help;
}
```

- Hiện và ẩn cursor:

- `Cursor.Hide();`
 - `Cursor.Show();`

Ví dụ

11



```
private void btnBackColor_Click(object sender, EventArgs e)
{
    panel1.BackColor = Color.Blue;
    //Hoặc có thể dùng
    //panel1.BackColor = Color.FromArgb(26, 69, 174);
    //panel1.BackColor =
    Color.FromKnownColor(KnownColor.DarkTurquoise);
}
private void btnChangeIcon_Click(object sender, EventArgs e)
{
    Icon icon = new Icon("Icon1.ico");
    Icon icon1 = new Icon(icon, new Size(50, 50));
    this.Icon = icon1;
}
private void btnChangeCursors_Click(object sender, EventArgs e)
{
    Cursor cursorpaint = new Cursor("Cursor1.cur");
    panel1.Cursor = cursorpaint;
    this.Cursor = Cursors.Help;
    Cursor.Hide();
    Cursor.Show();
}
private void panel1_MouseMove(object sender, MouseEventArgs e)
{
    this.toolStripStatusLabel1.Text = "; x=" + Cursor.Position.X
    + ",y=" + Cursor.Position.Y;
}
```

Màu

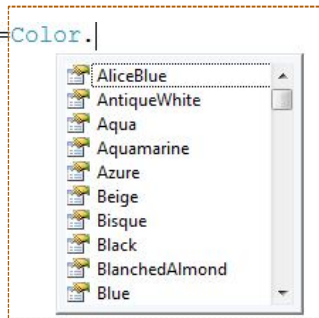
13

- Cấu trúc màu

Color	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	Blue								Green								Red							

- Sử dụng màu:

```
this.BackColor=Color.
```



Màu (Color)

14

- Sử dụng phương thức tĩnh để tạo màu:

```
public static Color FromArgb(int argb);
public static Color FromArgb(int alpha, Color baseColor);
public static Color FromArgb(int red, int green, int blue);
public static Color FromArgb(int alpha, int red, int green, int blue);
public static Color FromKnownColor(KnownColor name);
```

- Ví dụ:

```
private void btnBackColor_Click(object sender, EventArgs e)
{
    panel1.BackColor = Color.Blue;
    //Hoặc có thể dùng truyền 3 giá trị Red, Green, blue
    panel1.BackColor = Color.FromArgb(26, 69, 174);
    //Hoặc có thể dùng tên màu như sau
    panel1.BackColor = Color.FromKnownColor(KnownColor.DarkTurquoise);
}
```

Pen (bút vẽ)

15

- Sử dụng phương thức tạo đối tượng Pen

```
public Pen(Color color);  
public Pen(Color color, float width)
```

- Hoặc có thể sử dụng bút vẽ xây dựng sẵn (chỉ quan tâm đến màu)

```
Pen penBorder = Pens.Lavender;
```

Hàm vẽ cơ bản

16

- Lớp **Graphics** cung cấp 1 số phương thức vẽ cơ bản:

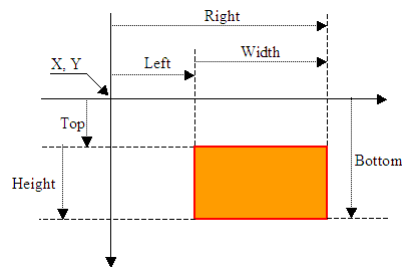
- Đoạn thẳng: DrawLine
- Đường gấp khúc: DrawLines
- Hình chữ nhật và hình vuông: DrawRectangle
- Hình tròn và hình Elip: DrawEllipse
- Hình đa giác: DrawPolygon

Hình chữ nhật và hình vuông

17

- Sử dụng phương thức `DrawRectangle()` của lớp `Graphics`, với cú pháp

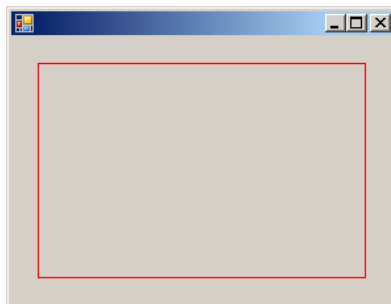
```
public void DrawRectangle(Pen pen, Rectangle rect);
public void DrawRectangle(Pen pen, RectangleF rect);
public void DrawRectangle(Pen pen, int x, int y, int width, int height);
public void DrawRectangle(Pen pen, float x, float y, float wd, float hg);
```



Ví dụ

18

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.DrawRectangle(new Pen(Color.Red),
        new Rectangle(20, 20, 248, 162));
}
```

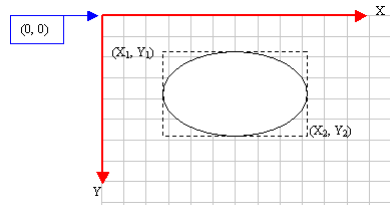


Hình tròn và hình Elip

19

- Sử dụng phương thức `DrawEllipse()` của lớp `Graphics`, với cú pháp

```
public void DrawEllipse(Pen pen, Rectangle rect);
public void DrawEllipse(Pen pen, RectangleF rect);
public void DrawEllipse(Pen pen, int x, int y, int width, int height);
public void DrawEllipse(Pen pen, float x, float y, float width, float height);
```



Ví dụ

20

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Pen penCurrent = new Pen(Color.Red);

    e.Graphics.DrawEllipse(penCurrent,
        new Rectangle(20, 20, 226, 144));
}
```

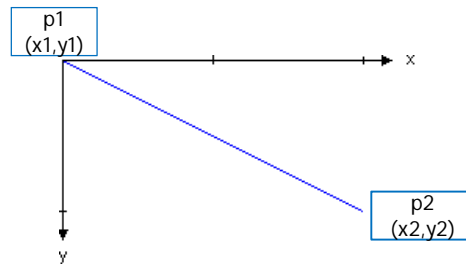


Đoạn thẳng

21

- Sử dụng phương thức DrawLine () của lớp Graphics, với cú pháp

```
public void DrawLine(Pen pen, Point pt1, Point pt2);
public void DrawLine(Pen pen, PointF pt1, PointF pt2);
public void DrawLine(Pen pen, int x1, int y1, int x2, int y2);
public void DrawLine(Pen pen, float x1, float y1, float x2, float y2);
```



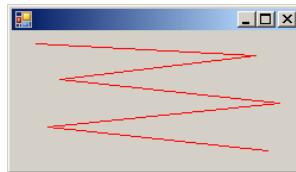
Đường gấp khúc

22

- Sử dụng phương thức DrawLines () của lớp Graphics, với cú pháp

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

- Ví dụ:

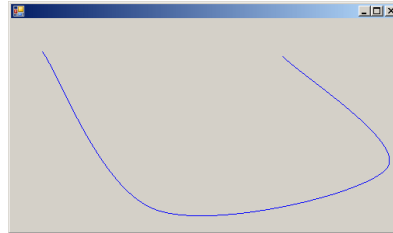


```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Point[] Coordinates = { new Point(20,10), new Point(205,20),
                           new Point(40, 40), new Point(225, 60),
                           new Point(30, 80), new Point(215, 100) };
    Pen penCurrent = new Pen(Color.Red);
    e.Graphics.DrawLines(penCurrent, Coordinates);
}
```

Vẽ các đường cong

23

- Sử dụng phương thức `DrawCurve()` của lớp `Graphics`, với cú pháp
`public void DrawCurve(Pen pen, Point[] points);`
`public void DrawCurve(Pen pen, PointF[] points);`



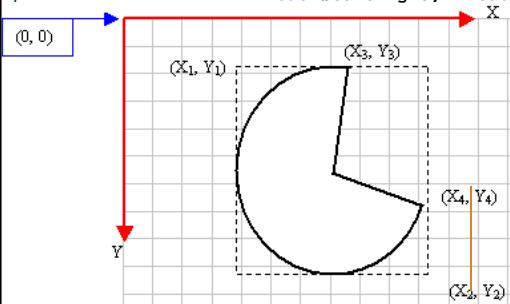
```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Pen penCurrent = new Pen(Color.Blue);
    Point[] pt = { new Point( 40, 42),new Point(188, 246),
                  new Point(484, 192),new Point(350, 48) };
    e.Graphics.DrawCurve(penCurrent, pt);
}
```

Drawing Pies

24

- Một Pie là một mảnh của Elip bị giới hạn bởi một góc bắt đầu và một số đo góc.

```
public void DrawPie(Pen pen,Rectangle rect,float startAngle,float sweepAngle);
public void DrawPie(Pen pen,RectangleF rect,float startAngle,float sweepAngle);
public void DrawPie(Pen pen,int x,int y,int width,int height,
                    int startAngle,int sweepAngle);
public void DrawPie(Pen pen,float x, float y, float width, float height,
                    float startAngle, float sweepAngle);
```

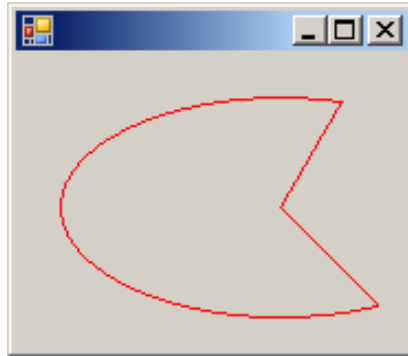


- (startAngle) – đo theo chiều kim đồng hồ từ trục x. (90 độ là chỉ 6 giờ)
 - (sweepAngle) – đo theo chiều kim đồng hồ bắt đầu từ startAngle

Ví dụ

25

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Pen penCurrent = new Pen(Color.Red);
    e.Graphics.DrawPie(penCurrent, 20, 20, 100, 100, 45, 255);
}
```

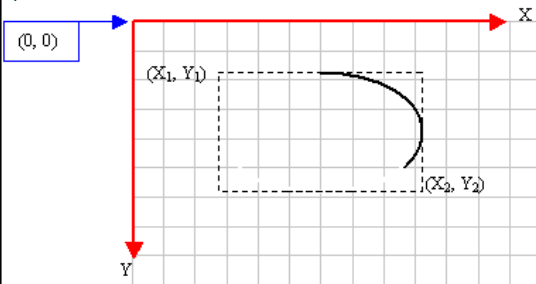


Vẽ cung tròn (Drawing Arcs)

26

- Một cung tròn là 1 phần của Elip không khép kín

```
public void DrawArc(Pen pen, Rectangle rect, float startAngle, float sweepAngle);
public void DrawArc(Pen pen, RectangleF rect, float startAngle, float sweepAngle);
public void DrawArc(Pen pen, int x, int y, int width, int height,
    int startAngle, int sweepAngle);
public void DrawArc(Pen pen, float x, float y, float width, float height,
    float startAngle, float sweepAngle);
```

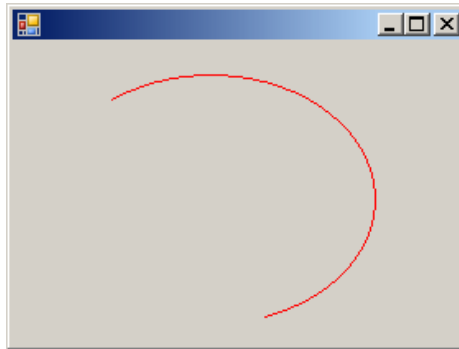


- (startAngle) – đo theo chiều kim đồng hồ từ trục x. (90 độ là chỉ 6 giờ)
 - (sweepAngle) – đo theo chiều kim đồng hồ bắt đầu từ startAngle

Ví dụ

(27)

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Pen penCurrent = new Pen(Color.Red);
    e.Graphics.DrawArc(penCurrent, 20, 20, 200, 150, 225, 200);
}
```



Brush (Cọ vẽ)

(28)

- Cọ vẽ (brush) là một đối tượng dùng để lưu trữ màu, hình ảnh hay một mẫu vẽ dùng để tô nền trong của một hình kín.
- Sử dụng Brush:
 - *Solid Brushes*
 - *Tạo Solid Brush*
 - *Hatch Brushes*

Brush (Cọ vẽ)

29

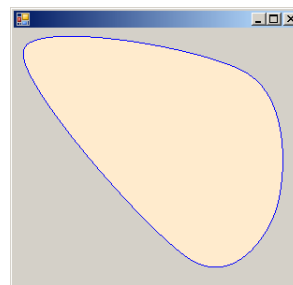
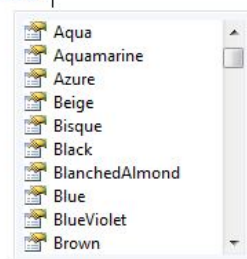
- Để tô kín sử dụng các phương thức sau trong lớp Graphics
 - FillRectangle: Sử dụng để tô phần trong của hình chữ nhật hoặc hình vuông
 - FillRectangles: Sử dụng để tô nền cho một loạt các hình chữ nhật.
 - FillEllipse: Dùng để tô màu Elip hoặc hình tròn.
 - FillPolygon: Dùng để tô màu phần trong của đa giác.
 - FillClosedCurve: Sử dụng để tô phần trong của một đường cong kín.
 - FillPie: Dùng để tô màu phần trong một Pie
 - FillPath: Dùng để tô phần trong của một Graphic Path

Solid Brushes

30

- Sử dụng các phương thức tĩnh của lớp **Brushes**:

Brushes.



```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    Point[] pt = { new Point(10, 22), new Point(188, 246),
                  new Point(280, 192), new Point(250, 48) };
    e.Graphics.FillClosedCurve(Brushes.BlanchedAlmond, pt);
    e.Graphics.DrawClosedCurve(Pens.Blue, pt);
}
```

Tạo Solid Brush

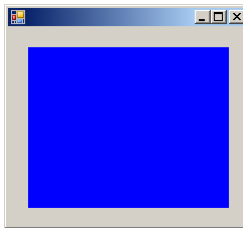
31

- Sử dụng các phương thức sau để tạo:

```
public SolidBrush(Color color);
```

- Ví dụ:

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    SolidBrush brushBlue = new SolidBrush(Color.Blue);
    e.Graphics.FillRectangle(brushBlue, 20, 20, 200, 160);
}
```



Tạo Solid Brush

32

```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    SolidBrush colorizer = new SolidBrush(Color.Lime);
    e.Graphics.FillRectangle(colorizer, 10, 10, 120, 120);
    colorizer.Color = Color.Salmon;
    e.Graphics.FillRectangle(colorizer, 140, 10, 120, 120);
    colorizer.Color = Color.Aqua;
    e.Graphics.FillRectangle(colorizer, 10, 140, 120, 120);
    colorizer.Color = Color.Navy;
    e.Graphics.FillRectangle(colorizer, 140, 140, 120, 120);
}
```

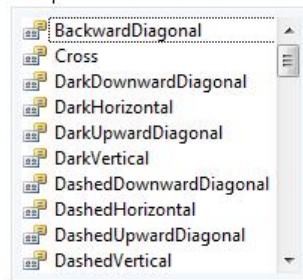


Hatch Brushes

33

- Hatch Brush dựa trên các mẫu tô đã được vẽ hay thiết kế trước để tô nền. `using System.Drawing.Drawing2D;`
- Sử dụng phương thức:

```
public HatchBrush(HatchStyle style, Color foreColor);
HatchStyle.
```

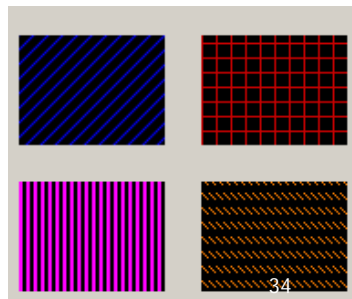


```
private void Exercise_Paint(object sender, PaintEventArgs e)
{
    HatchBrush brushBackDiag = new HatchBrush(HatchStyle.BackwardDiagonal,
        Color.FromArgb(0, 0, 255));
    e.Graphics.FillRectangle(brushBackDiag, 20, 20, 80, 60);

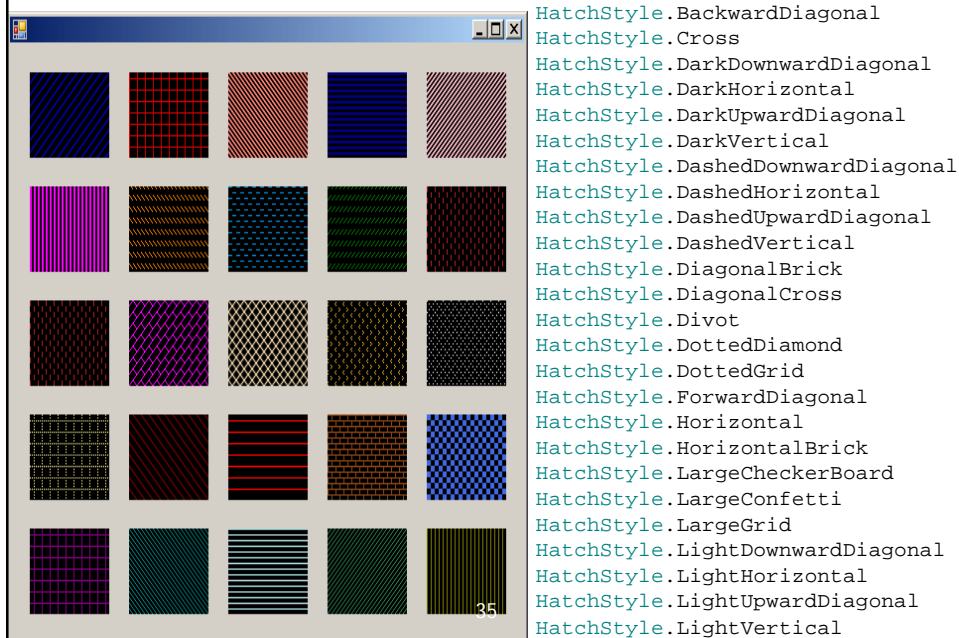
    HatchBrush brushCross=new HatchBrush(HatchStyle.Cross,
        Color.FromArgb(200,0, 0));
    e.Graphics.FillRectangle(brushCross, 120, 20, 80, 60);

    HatchBrush brushVertical = new HatchBrush(HatchStyle.DarkVertical,
        Color.FromArgb(255, 0, 255));
    e.Graphics.FillRectangle(brushVertical, 20, 100, 80, 60);

    HatchBrush brushDashDnDiag = new HatchBrush(HatchStyle.DashedDownwardDiagonal,
        Color.FromArgb(255, 128, 0));
    e.Graphics.FillRectangle(brushDashDnDiag, 120, 100, 80, 60);
}
```



Một số Hatch Style



KẾT THÚC CHƯƠNG

36

Q&A