

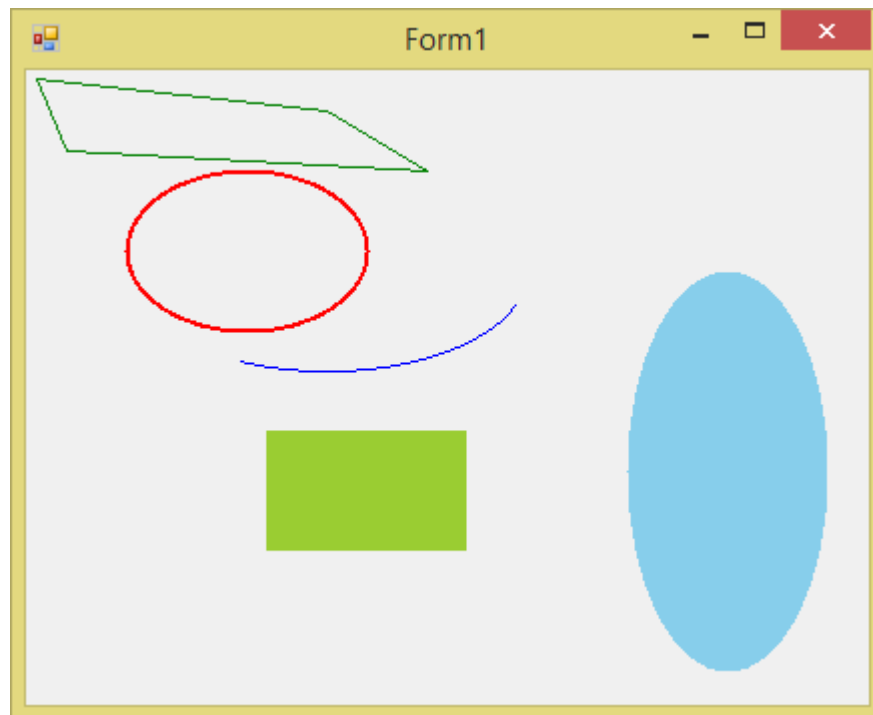
1 Bài tập GDI+

Tham khảo thêm thư viện tại MSDN:

<https://msdn.microsoft.com/en-us/library/7f4d4txw%28v=vs.110%29.aspx>

<https://msdn.microsoft.com/en-us/library/a36fascx%28v=vs.110%29.aspx>

1.1 Vẽ một số hình cơ bản lên Form



Để vẽ hình lên form có 2 cách:

1.1.1 Code cho sự kiện Paint của Form

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    //Lấy đối tượng Graphics của form
    Graphics g = e.Graphics;

    //Tạo bút vẽ
    Pen pen1 = new Pen(Color.Red, 2);

    //Vẽ ellipse
    g.DrawEllipse(pen1, 50, 50, 120, 80);

    //Vẽ cung
    Rectangle rect = new Rectangle(50, 50, 200, 100);
    g.DrawArc(new Pen(Color.Blue), rect, 10, 124);
}
```

```

//Vẽ đa giác
g.DrawPolygon(new Pen(Color.Green),
    new PointF[] {
        new PointF (20, 40), new PointF(5, 4),
        new PointF(150, 20), new PointF(200, 50)
    });

//Tạo chổi tô
SolidBrush brush1 = new SolidBrush(Color.YellowGreen);

//Vẽ và tô hình chữ nhật
e.Graphics.FillRectangle(brush1, 120, 180, 100, 60);

//Vẽ và tô ellipse
e.Graphics.FillEllipse(new SolidBrush(Color.SkyBlue),
    300, 100, 100, 200);
// ...
}

```

1.1.2 Dùng hàm CreateGraphics() của form

```

private void btnDraw_Click(object sender, EventArgs e)
{
    // Lay doi tuong graphic cua form
    Graphics g = this.CreateGraphics();

    //Tạo bút vẽ
    Pen pen1 = new Pen(Color.Red, 2);

    //Vẽ ellipse
    g.DrawEllipse(pen1, 50, 50, 120, 80);

    //Vẽ cung
    Rectangle rect = new Rectangle(50, 50, 200, 100);
    g.DrawArc(new Pen(Color.Blue), rect, 10, 124);

    // ...
}

```

1.2 Một số kiến thức:

Khái niệm cơ bản

- 1 đối tượng **Graphics** quản lý 1 bối cảnh đồ họa
- Mỗi phương thức có nguồn gốc từ **System.Windows.Forms.Form** kế thừa 1 phương thức **virtual OnPaint**.
- Các đối số trong phương thức **OnPaint** bao gồm 1 đối tượng **PaintEventArgs**.
Graphics g = e.Graphics;

FONT

Khai báo thư viện: *System.Drawing* và *System.Drawing.Text*

4 lớp cơ bản liên quan đến đối tượng FONT: FontStyle, FontFamily, GraphicsUnit, Font

a. Hàm vẽ chuỗi:

```
void DrawString(string s, Font font, Brush brush, float x, float y);
```

```
void DrawString(string s, Font font, Brush brush, Point p);
```

b. Bút vẽ:

```
Pen(Color c);
```

```
Pen(Color c, float width);
```

```
Pen(Brush b);
```

```
Pen(Brush b, float width);
```

c. Chổi tô & Font Style:

SolidBrush: Chổi tô đặc với màu đã chọn.

```
SolidBrush(Color c);
```

FontStyle:

Thành phần	Đặc tả
Bold	Chữ in đậm
Italic	Chữ in nghiêng
Regular	Chữ bình thường
StrikeOut	Chữ với đường gạch ngang thân
Underline	Chữ gạch chân

```
FontStyle style = FontStyle.Bold;
```

```
Font(tên_font, cỡ);
```

```
Font(tên_font, cỡ, style);
```

d. Các hàm vẽ

Vẽ (rỗng): Draw_____

- Vẽ đường thẳng: DrawLine
- Vẽ elip: DrawEllipse
- Vẽ hình chữ nhật: DrawRectangles

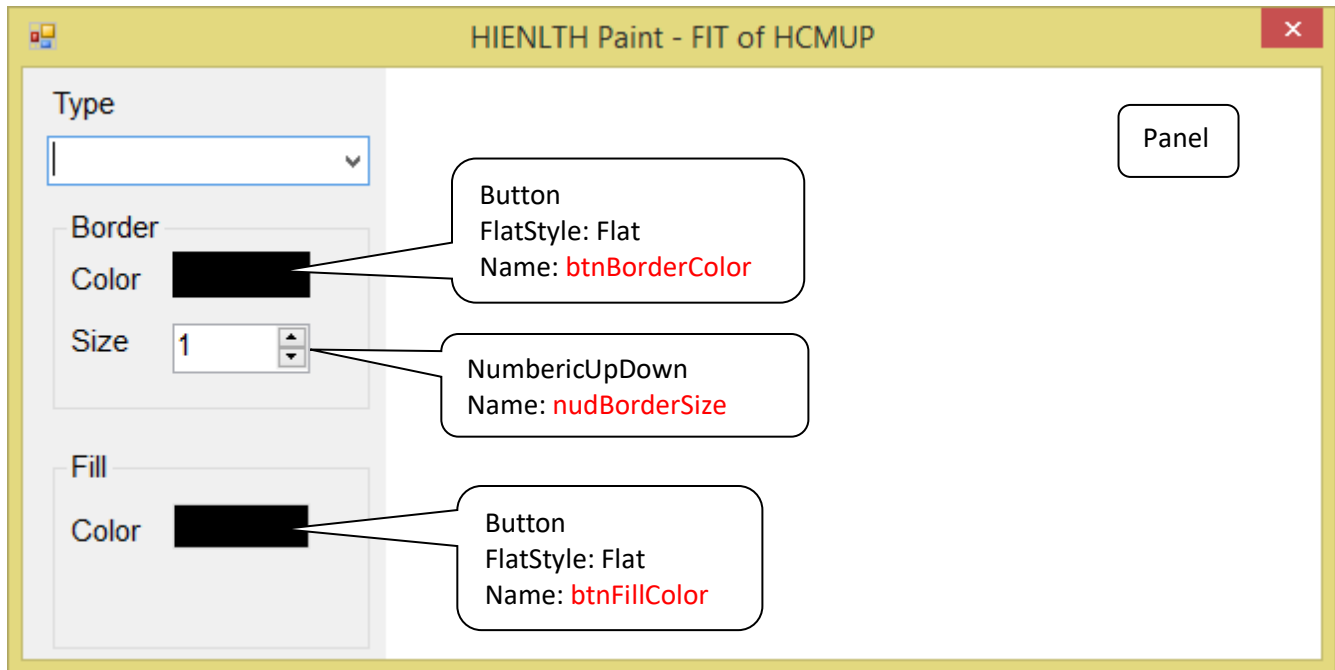
- ...

Vẽ hình đặc: Fill_____

- **Hàm tô hình chữ nhật:**
void FillRectangle(Brush brush, int x, int y, int width, int height);

1.3 Bài tập vẽ hình

Thiết kế Form theo mẫu:



Sự kiện vẽ: Ấn chuột trái để lấy điểm đầu tiên, nhả phím chuột để xác định điểm cuối cùng. Sau đó thực hiện vẽ và tô theo màu theo hình đã chọn trên khung vẽ.

Hướng dẫn:

//Các thuộc tính

```
private Color fill;           //lưu trữ màu tô
private Color border;        //lưu trữ màu nét vẽ
private bool mouse_is_down = false;
private Point start;
private Graphics g;          //đối tượng đồ họa
```

//Sự kiện Form_Load

```
g = panel1.CreateGraphics();
border = Color.Black;
btnBorderColor.BackColor = border;
fill = Color.Red;
```

```
btnFillColor.BackColor = fill;
```

//Sự kiện Click chọn màu cho Border¹

```
ColorDialog dl = new ColorDialog();  
dl.FullOpen = true;  
dl.AnyColor = true;  
if (dl.ShowDialog() == DialogResult.OK)  
{  
    btnBorderColor.BackColor = dl.Color;  
    border = dl.Color;  
}
```

//Sự kiện nhấn chuột vào vùng vẽ (Panel_MouseDown)

```
mouse_is_down = true;  
start = new Point(e.X, e.Y);
```

//Sự kiện nhả chuột vào vùng vẽ (Panel_MouseUp)

```
mouse_is_down = false;
```

//Sự kiện rê (di chuyển) chuột trong vùng vẽ (Panel_MouseMove)

```
if (mouse_is_down)  
{  
    g.Clear(panel1.BackColor);  
    Point end = new Point(e.X, e.Y);  
    switch (cboType.SelectedIndex)  
    {  
        //line  
        case 0:  
            g.DrawLine(new Pen(border, (int)nudBorderSize.Value),  
start, end);  
            break;  
        //empty ellipse.  
        case 1:  
            g.DrawEllipse(new Pen(border, (int)nudBorderSize.Value),  
start.X, start.Y, e.X - start.X, e.Y - start.Y);  
            break;  
        //filled ellipse  
        case 2:  
            {  
                g.DrawEllipse(new Pen(border,  
(int)nudBorderSize.Value), start.X, start.Y, e.X - start.X, e.Y - start.Y);  
            }  
    }  
}
```

¹ SV tự cài đặt cho sự kiện click vào nút chọn màu cho màu tô (fill)

```

        g.FillEllipse(new SolidBrush(fill), start.X, start.Y,
e.X - start.X, e.Y - start.Y);
        break;
    }
    //empty rectangle
    case 3:
        g.DrawRectangle(new Pen(border, (int)nudBorderSize.Value),
start.X, start.Y, e.X - start.X, e.Y - start.Y);
        break;
    //filled rectangle
    case 4:
        g.DrawRectangle(new Pen(border, (int)nudBorderSize.Value),
start.X, start.Y, e.X - start.X, e.Y - start.Y);
        g.FillRectangle(new SolidBrush(fill), start.X, start.Y,
e.X - start.X, e.Y - start.Y);
        break;
    default:
        break;
}

```

2 Mở rộng

- Thử xoay đối tượng vẽ?

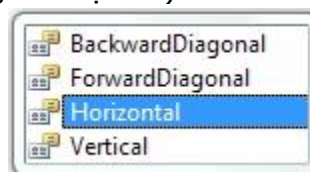
Sử dụng: `RotateTranform(góc)` & `ResetTranform()`;

3 Bài tập về nhà (Assignment 04 – My Paint)

3.1 Yêu cầu

Hoàn thành chương trình Paint thêm những yêu cầu sau:

- Tổ chức các loại hình học dạng Hướng đối tượng.
- Bổ sung thêm nhiều loại hình: Tam giác, Vuông, Bình hành, ...
- Cho phép vẽ nhiều (đối tượng) hình trên bàn vẽ.
- Cho phép chọn và đổi màu hình
- Cho phép vẽ chuỗi với Font, Size tùy chọn.
- Cho phép tô màu theo các dạng:
 - Solid,
 - LinearGradientMode (một trong các loại sau):



- PathGradientBrush
- TextureBrush

- HatchBrush

Brush Class	Description
SolidBrush	The simplest form of brush, which paints in a solid color.
HatchBrush	Similar to a SolidBrush , but it allows you to select from a large variety of preset patterns to paint with, rather than a solid color.
TextureBrush	Paints using a texture, such as an image.
LinearGradientBrush	Paints two colors blended along a gradient.
PathGradientBrush	Paints using a complex gradient of blended colors, based on a unique path defined by the developer.

- Cho phép save hình từ khung vẽ xuống file bmp, png, jpeg
- Phải có About giới thiệu chương trình, tác giả.

3.2 Hình thức & Đánh giá

- Làm theo cá nhân, đặt tên project dạng **MSSV_HoTen_Ass03_Paint**. Sai tên – 01 điểm. Trong đó MSSV là mã số sinh viên (không có dấu .), HoTen là họ tên đầy đủ tiếng Việt không dấu, không khoảng trắng.
- Ngày nộp: TBD.
- Nộp tại: TBD
- Đánh giá:
 - Giao diện: 25%
 - Tổ chức dạng hướng đối tượng, có kế thừa: 25%
 - Vẽ hình dạng dùng Brush dạng: Hatch, Gradient, Texture : 20%
 - Vẽ chữ: 10%
 - Chọn hình: 10%
 - Save file ảnh: 10%

---hết---