Chương 4: Giao diện người dùng (User Interface)

Gv: Đặng Hữu Nghị

Nội dung

4.7. Điều khiển dialog và phương thức message

4.7.1. Lớp MessageBox

4.7.2. Điều khiển ColorDiaLog

4.7.1. Lớp MessageBox

- Lớp *MessageBox* giúp hiển thị một hộp thoại trên màn hình.
- Khi hộp thoại xuất hiện thì người dùng bắt buộc phải thao tác trên hộp thoại trước thì mới có tiếp tục thực hiện công việc trên ứng dụng.
- Việc hiển thị như thế hữu ích khi muốn cảnh báo một lỗi hoặc hướng dẫn cho người dùng.
- Sử dụng lớp *MessageBox* thì cần khai báo không gian tên: System.Windows.Forms

4.7.1. Lớp MessageBox

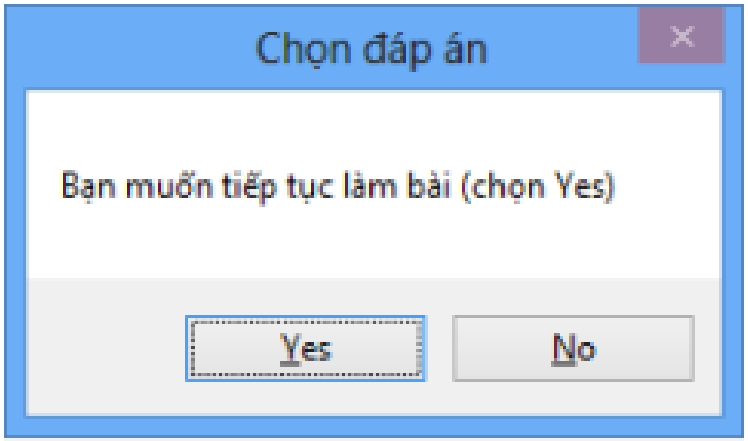
- Để hiển thị hộp thoại, người dùng chỉ cần sử dụng phương thức tĩnh Show() của lớp *MessageBox*

```
DialogResult Show(string text, string caption,
MessageBoxButtons buttons, MessageBoxIcon icon,
MessageBoxDefaultButton defaultButton,
MessageBoxOptions options);
```

4.7.1. Lớp MessageBox

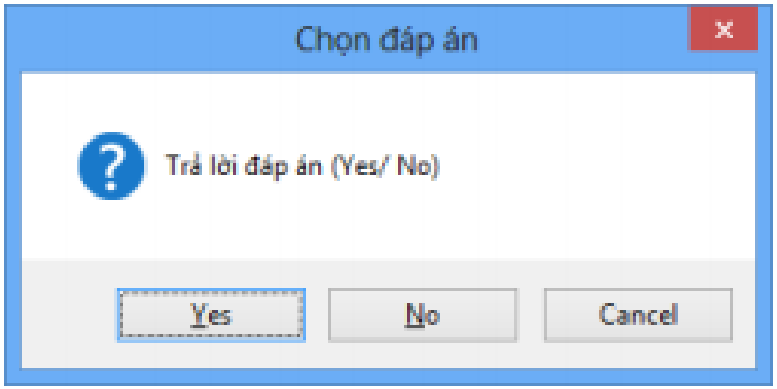
- Trong đó:
  - ✓ *Text*: Chuỗi hiển trên trên hộp thoại *MessageBox*
  - ✓ *Caption*: Chuỗi hiển thị trên thanh titlebar của hộp thoại *MessageBox*
  - ✓ Kiểu liệt kê *MessageBoxIcon* gồm các giá trị xác định biểu tượng hiển thị trên *MessageBox*
  - ✓ Kiểu liệt kê *MessageBoxDefaulButtons* gồm các giá trị xác định nút được *Focus* trên *MessageBox*
  - ✓ Kiểu liệt kê *MessageBoxButtons* gồm các giá trị xác định nút hiển thị trên *MessageBox*
  - ✓ Kiểu liệt kê *MessageBoxOption* gồm các giá trị xác định nút hiển thị bên phải hoặc bên trái của *MessageBox*

4.7.1. Lớp MessageBox



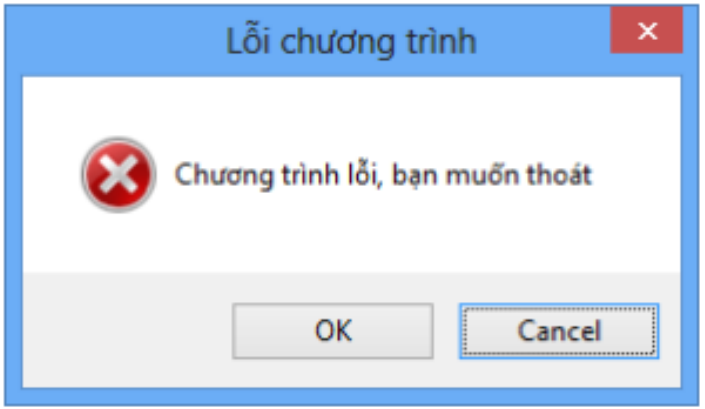

```
DialogResult r = MessageBox.Show("Bạn muốn tiếp tục làm
bài”, "Chọn đáp án", MessageBoxButtons.YesNo);
if (r == DialogResult.Yes)
        MessageBox.Show("Bạn đã chọn Yes");
else
        MessageBox.Show("Bạn đã chọn No");
```

4.7.1. Lớp MessageBox




```
DialogResult r = MessageBox.Show("Trả lời đáp án (Yes/ No)",
"Chọn đáp án", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question);
if (r == DialogResult.Yes)
        MessageBox.Show("Bạn đã chọn Yes");
else
        if(r==DialogResult.No)
                MessageBox.Show("Bạn đã chọn No");
        else
                MessageBox.Show("Bạn đã thoát chương trình");
```

4.7.1. Lớp MessageBox




```
DialogResult r = MessageBox.Show("Chương trình lỗi, bạn muốn
thoát", "Lỗi chương trình", MessageBoxButtons.OKCancel,
MessageBoxIcon.Error, MessageBoxDefaultButton.Button2);
if (r == DialogResult.OK)
        MessageBox.Show("Bạn đã chọn OK, thoát chương trình");
else
        if(r==DialogResult.No)
        MessageBox.Show("Bạn đã chọn Cancel, tiếp tục chạy
chương trình");
```

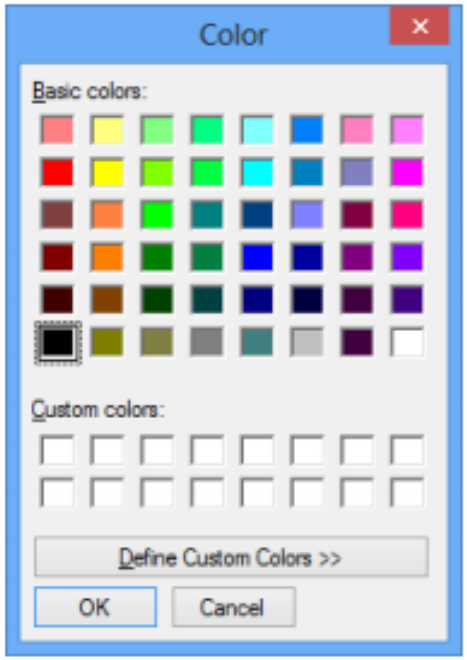
4.7.2. Điều khiển ColorDiaLog

- *ColorDialog Box* chứa một danh sách các màu sắc được xác định cho hệ thống hiển thị. Người dùng có thể lựa chọn hoặc tạo ra một màu sắc đặc biệt từ danh sách, sau đó áp dụng khi thoát khỏi hộp thoại.
- *ColorDialog* nằm trong nhóm Dialog của cửa sổ Toolbox

4.7.2. Điều khiển ColorDiaLog

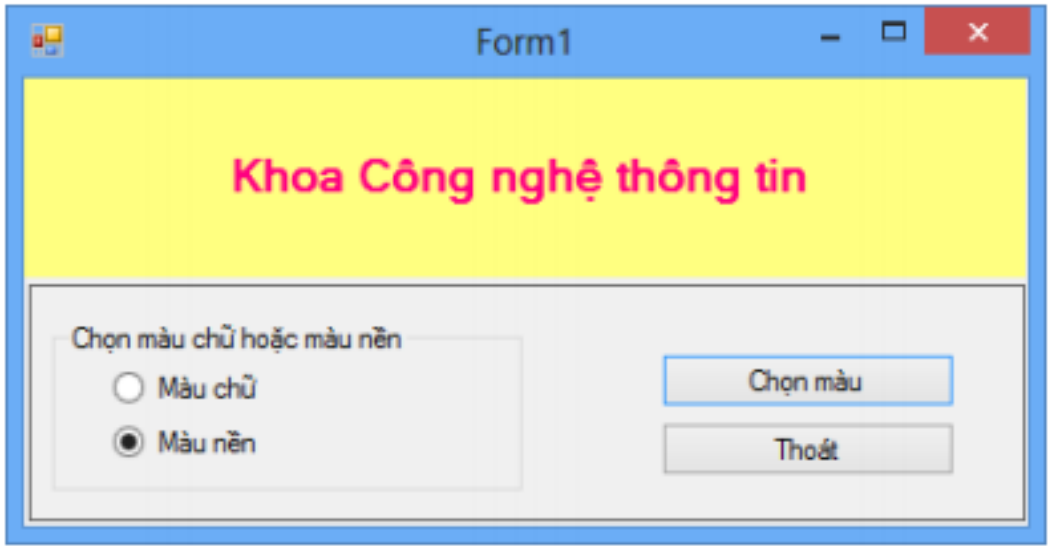
- Ta có thể tạo ra đối tượng *ColorDialog* bằng cách kéo điều khiển vào form từ cửa sổ Toolbox hoặc sử dụng mã lệnh tạo đối tượng và hiển thị *ColorDialog Box* bằng cách gọi phương thức ShowDialog() để hiển thị hộp thoại như hình

```
ColorDialog clbox= new ColorDialog();
clbox.ShowDialog();
```



4.7.2. Điều khiển ColorDiaLog

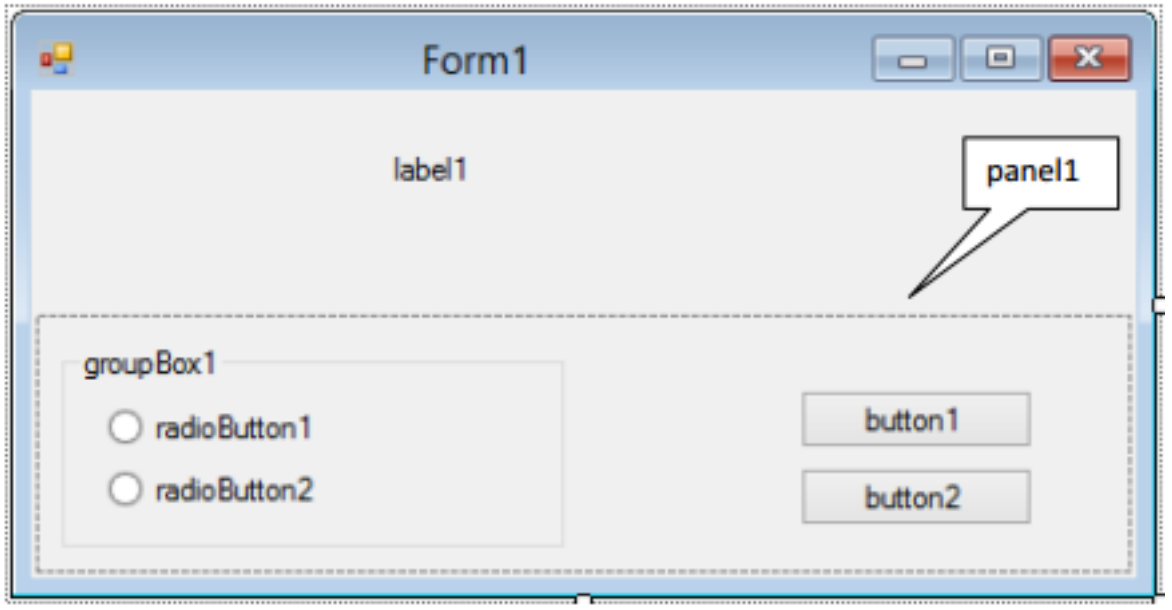
- Ví dụ: Viết chương trình chọn màu chữ và màu nền cho Label có giao diện như hình



- Yêu cầu: Người dùng chọn thay đổi màu nền hoặc màu chữ của Label có dòng chữ "Khoa Công nghệ thông tin" bằng cách chọn một trong hai *RadioButton*: "Màu chữ", "Màu nền". Sau khi đã có lựa chọn, người dùng nhấn vào nút "Chọn màu" để hiện thị *ColoDialog Box* và chọn màu muốn thay đổi cho *Label*.

4.7.2. Điều khiển ColorDiaLog

- Bước 1: Thiết kế giao diện ban đầu: Thêm các điều khiển *Label*, *Button*, *Panel*,*GroupBox* và *RadioButton* từ cửa sổ Toolbox vào form như hình



4.7.2. Điều khiển ColorDiaLog

- Bước 2: Thiết lập giá trị thuộc tính cho điều khiển trong cửa sổ Properties
  - ✓ label1:
    - Thuộc tính *Text*: “Khoa Công nghệ thông tin”
    - Thuộc tính *Font Size*: 14
    - Thuộc tính *Font style*: Bold
    - Thuộc tính *AutoSize*: False
    - Thuộc tính *Size*: 428, 86
    - Thuộc tính *TextAlign*: MiddleCenter
  - ✓ panel1:
    - Thuộc tính *BorderStyle*: FixedSingle
  - ✓ groupBox1:
    - Thuộc tính *Text*: “Chọn màu chữ hoặc màu nền”

4.7.2. Điều khiển ColorDiaLog

- ✓radioButton1:
  - Thuộc tính *Text*: “Màu chữ”
  - Thuộc tính *Name*: radMauChu
- ✓radioButton2:
  - Thuộc tính *Text*: “Màu nền”
  - Thuộc tính *Name*: radMauNen
- ✓button1:
  - Thuộc tính *Text*: “Chọn màu”
  - Thuộc tính *Name*: btnChonMau
- ✓button2:
  - Thuộc tính *Text*: “Thoát”
  - Thuộc tính *Name*: btnThoat

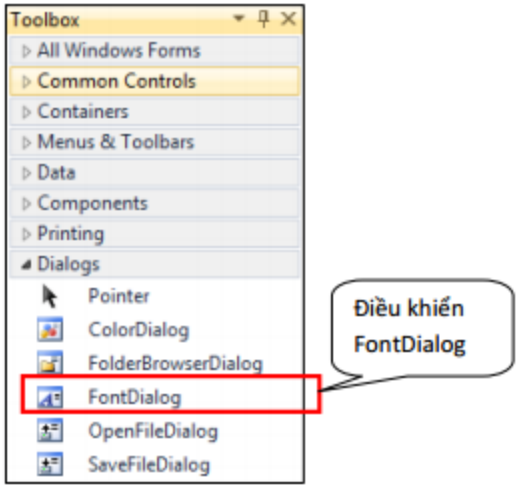
4.7.2. Điều khiển ColorDiaLog

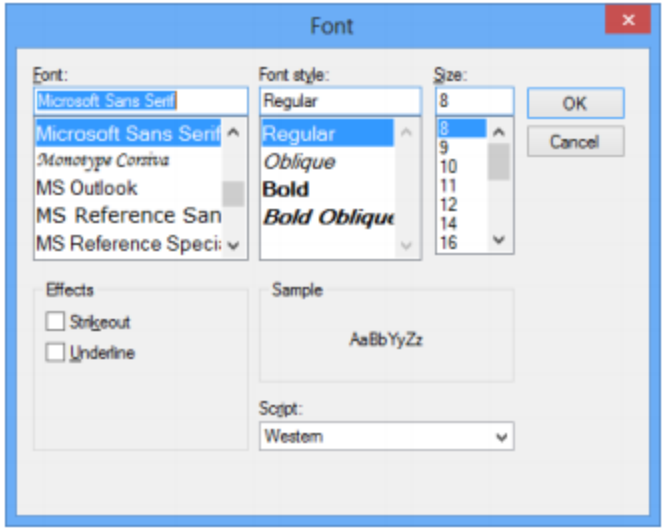
- Sự kiện *Click* nút btnChonMau:

```
private void btnChonMau_Click(object sender, EventArgs e)
{
        ColorDialog clDLog = new ColorDialog();
        DialogResult rs = clDLog.ShowDialog();
        if (rs == DialogResult.OK)
        {
                if (radMauChu.Checked == true)
                {
                        lblHienThi.ForeColor = clDLog.Color;
                }
                if (radMauNen.Checked == true)
                {
                        lblHienThi.BackColor = clDLog.Color;
                }
        }
}
```

4.7.3. Điều khiển FontDiaLog

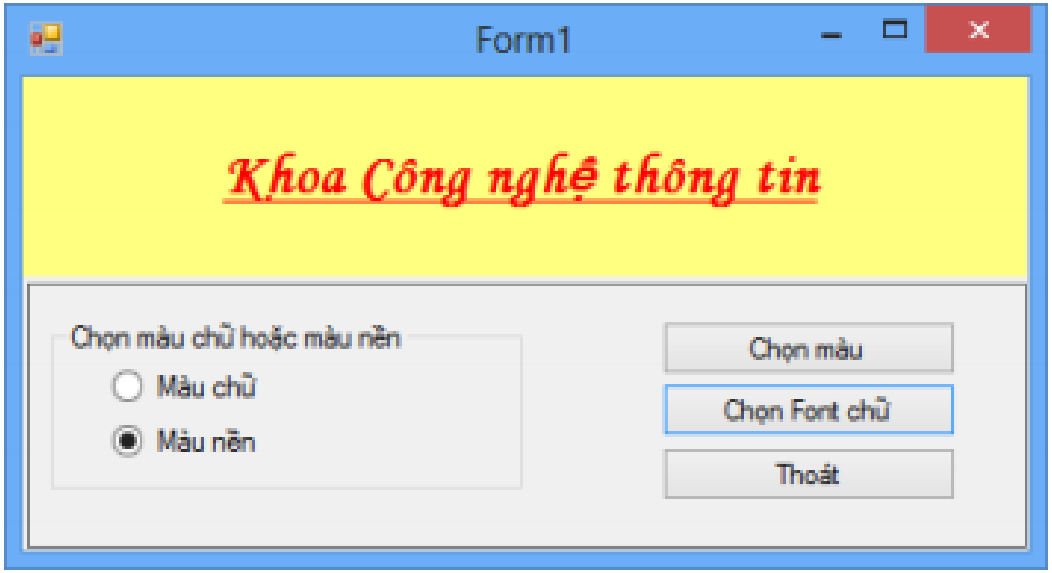
- Tương tự như *ColorDialog Box*, để thiết lập font chữ hiển thị, C# cũng hỗ trợ hộp thoại gọi là *FontDialog Box*. *FontDialog Box* sẽ hiển thị một danh sách các font chữ hiện đang được cài đặt trên hệ thống; Cho phép người dùng lựa chọn các thuộc tính cho một font chữ hợp lý, như kiểu font chữ, kích cỡ chữ, hiệu ứng,..





4.7.3. Điều khiển FontDiaLog

- Ví dụ: Dựa vào ví dụ trên. Thêm nút lệnh “Chọn Font chữ” như hình



4.7.3. Điều khiển FontDiaLog

- Khai báo biến fontbox có kiểu *FontDialog*:

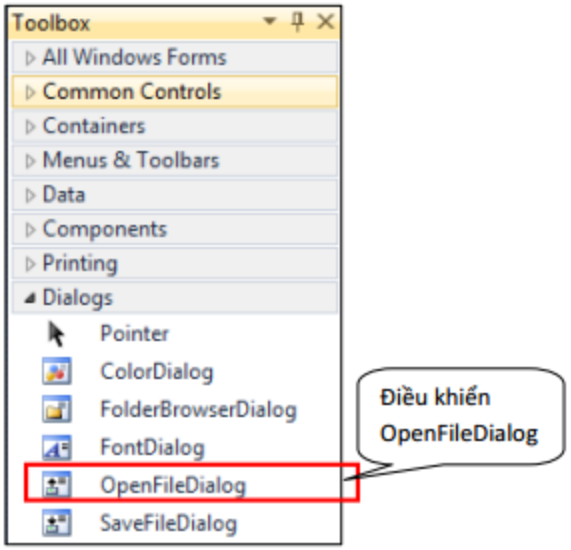
```
FontDialog fontbox = new FontDialog();
```

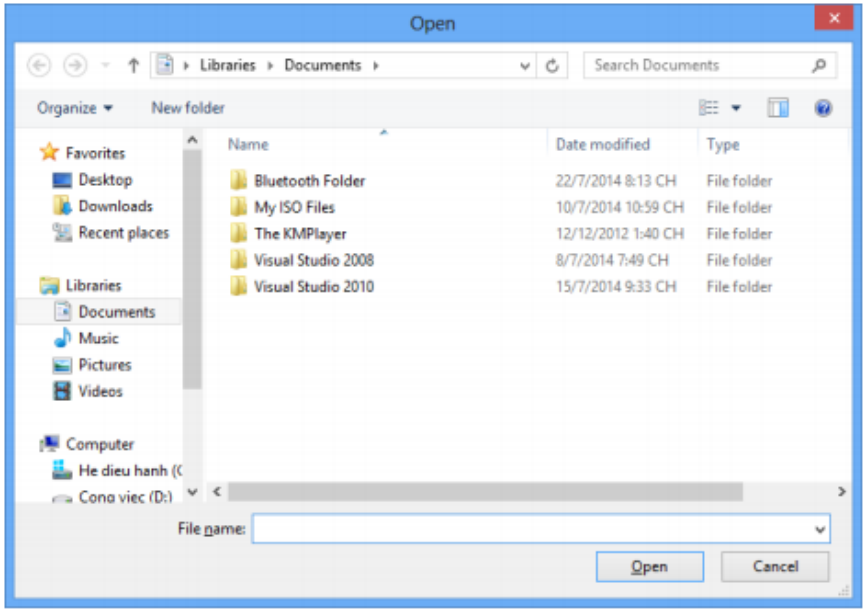
- Sự kiện *Click* nút btnChonFont

```
private void btnChonFont_Click(object sender, EventArgs e)
{
        DialogResult rs = fontbox.ShowDialog();
        if (rs == DialogResult.OK)
        {
                lblHienThi.Font = fontbox.Font;
        }
}
```

4.7.4. Điều khiển OpenFileDiaLog

- Điều khiển *OpenFileDialog* cho phép hiển thị hộp thoại *OpenFileDialog* để lấy về ổ đĩa, tên thư mục, tên tập tin đối với một tập tin hay thư mục đang tồn tại. Thông thường các ứng dụng Windows Forms sử dụng hộp thoại *OpenFileDialog* để mở một tập tin nào đó lưu trên bộ nhớ máy tính.





4.7.4. Điều khiển OpenFileDiaLog

Thuộc tính	Mô tả
Tilte	Thiết lập chuỗi hiển thị trên titlebar của hộp thoại OpenFileDialog
Filter	Thuộc tính nhận giá trị là một chuỗi. Chuỗi này chứa các đuôi mở rộng của tập tin, giúp lọc ra trên hộp thoại OpenFileDialog chỉ hiển thị đúng loại đuôi có trong chuỗi. Ví dụ: Thiết lập chuỗi chỉ hiển thị file hình ảnh có đuôi JPG và đuôi GIF: "jpg images|*.jpg|gif images|*.gif" Thiết lập chuỗi hiển thị tất cả những gì có trong thư mục trên hộp thoại: "All|*.*"
FileName	Thuộc tính trả về đường dẫn cùng tên tập tin được chọn
FileNames	Thuộc tính trả về một mảng các đường dẫn cùng tên tập tin được chọn. Lưu ý: Thuộc tính thường được sử dụng khi thuộc tính MultiSelect được thiết lập là True

4.7.4. Điều khiển OpenFileDiaLog

Thuộc tính	Mô tả
FilterIndex	Thuộc tính sử dụng cùng với thuốc tính Filter. Có ý nghĩa sẽ ưu tiên lọc loại tập tin nào trước trong chuỗi Filter. Ví dụ: Chuỗi Filter lọc hai loại tập tin JPG và GIF: "jpg images|*.jpg|gif images|*.gif" Nếu muốn hộp thoại khi hiển thị lọc hình có đuôi JPG đầu tiên thiết lập thuộc tính FilterIndex có giá trị 1. Nếu muốn hộp thoại khi hiển thị lọc hình có đuôi GIF đầu tiên thiết lập thuộc tính FilterIndex có giá trị 2.
InitialDirectory	Giá trị thuộc tính là một đường dẫn đến một thư mục hoặc ổ đĩa. Khi hộp thoại OpenFileDialog được mở lên sẽ hiển thị nội dung của đường dẫn này đầu tiên.
RestoreDirectory	Mang hai giá trị True hoặc False, cho biết hộp thoại có trả lại đường dẫn thư mục vừa mở trước đó. Lưu ý: Thuộc tính chỉ có hiệu lực khi thuộc tính InitialDirectory mang giá trị rỗng

4.7.4. Điều khiển OpenFileDiaLog

Thuộc tính	Mô tả
MultiSelect	Mang giá trị True hoặc False. - Nếu là True: Cho phép người dùng chọn nhiều tập tin hoặc thư mục - Nếu là False: Chỉ được chọn 1 tập tin hoặc một thư mục
CheckFileExists	Mang hai giá trị True hoặc False. Nếu là True sẽ cho phép hộp thoại hiển thị một cảnh báo nếu người dùng chỉ định một tập tin không tồn tại và nhấn nút Open.
ReadOnlyChecked	Mang giá trị True hoặc False. - Nếu là True: Hiển thị một CheckBox với tiêu đề Open as Read Only phía dưới ComboBox Files Of Types - Nếu là False: Không hiển thị CheckBox. Lưu ý: Thuộc tính chỉ có hiệu lực khi thuộc tính ShowReadOnly mang giá trị True

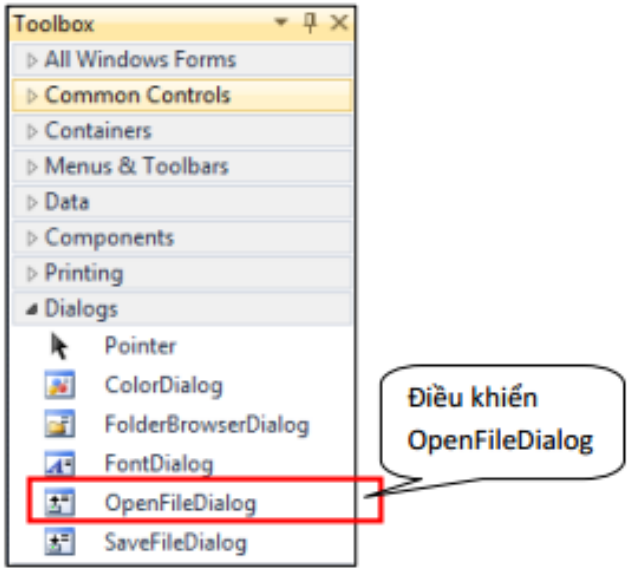
4.7.4. Điều khiển OpenFileDiaLog

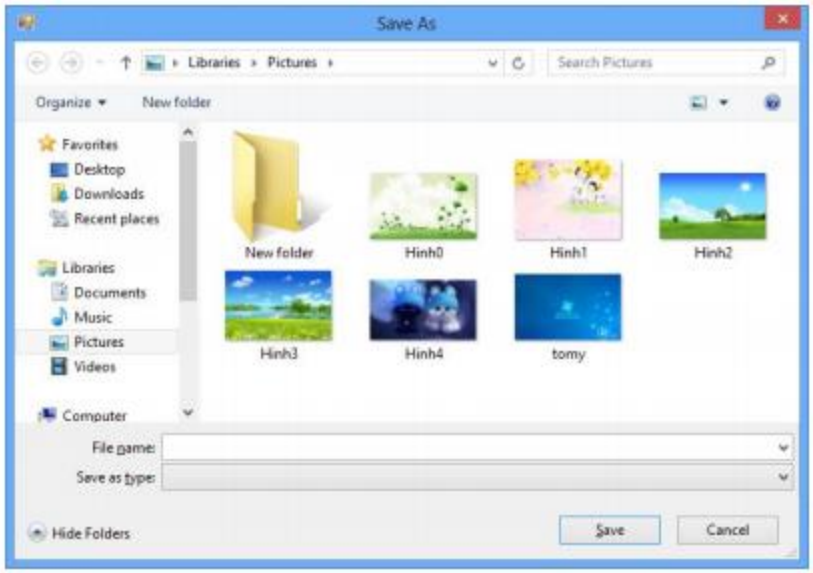
Thuộc tính	Mô tả
AddExtension	Mang hai giá trị True hoặc False. - Nếu là True sẽ cho phép thêm vào tên mở rộng (jpg, gif, ..) vào tập tin. - Nếu là False: Không cho phép
DefaultExt	Thêm tên mở rộng (.jpg, .gif, …) cho tập tin nếu người dùng không cung cấp tên mở rộng.

Phương thức	Mô tả
Reset()	Thiết lập giá trị các thuộc tính trở lại giá trị mặc định
ShowDialog()	Hiển thị hộp thoai, bắt buộc người dùng phải thao tác và đóng hộp thoại lại mới có thể thực hiện công việc khác.

4.7.5. Điều khiển SaveFileDialog

- Điều khiển *SaveFileDialog* cho phép hiển thị hộp thoại *SaveFileDialog* để người dùng lựa chọn đường dẫn đến một thư mục hoặc ổ đĩa trên hệ thống để ghi một tập tin mới hoặc ghi đè tập tin đã có.





4.7.5. Điều khiển SaveFileDialog

- *SaveFileDialog* và *OpenFileDialog* đều thừa kế từ lớp cơ sở là *FileDialog* do đó *SaveFileDialog* có nhiều thuộc tính giống như *OpenFileDialog*.
- Một số thuộc tính *OpenFileDialog* có nhưng *SaveFileDialog* không có như: *Multiselect*, *ReadOnlyChecked*, *ShowReadOnly*.
- Do là hộp thoại giúp lưu tập tin do đó *SaveFileDialog* hỗ trợ hai thuộc tính mới là: *CreatePrompt* và *OverwritePrompt.*

4.7.5. Điều khiển SaveFileDialog

Thuộc tính	Mô tả
Tilte	Thiết lập chuỗi hiển thị trên titlebar của hộp thoại OpenFileDialog
Filter	Thuộc tính nhận giá trị là một chuỗi. Chuỗi này chứa các đuôi mở rộng của tập tin, giúp lọc ra trên hộp thoại OpenFileDialog chỉ hiển thị đúng loại đuôi có trong chuỗi. Ví dụ: Thiết lập chuỗi chỉ hiển thị file hình ảnh có đuôi JPG và đuôi GIF: "jpg images|*.jpg|gif images|*.gif" Thiết lập chuỗi hiển thị tất cả những gì có trong thư mục trên hộp thoại: "All|*.*"
FileName	Thuộc tính trả về đường dẫn cùng tên tập tin được chọn

4.7.5. Điều khiển SaveFileDialog

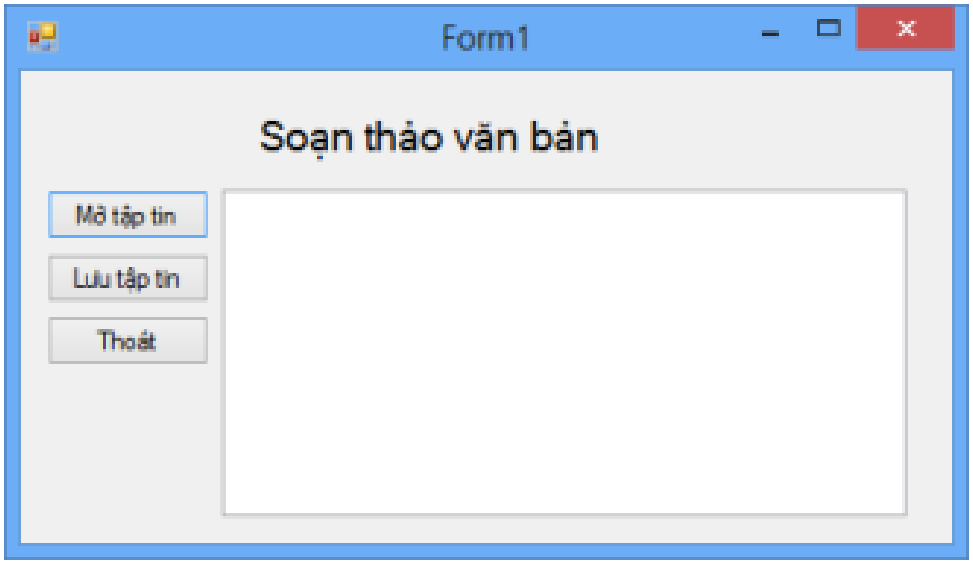
Thuộc tính	Mô tả
FilterIndex	Thuộc tính sử dụng cùng với thuốc tính Filter. Có ý nghĩa sẽ ưu tiên lọc loại tập tin nào trước trong chuỗi Filter. Ví dụ: Chuỗi Filter lọc hai loại tập tin JPG và GIF: "jpg images|*.jpg|gif images|*.gif" Nếu muốn hộp thoại khi hiển thị lọc hình có đuôi JPG đầu tiên thiết lập thuộc tính FilterIndex có giá trị 1. Nếu muốn hộp thoại khi hiển thị lọc hình có đuôi GIF đầu tiên thiết lập thuộc tính FilterIndex có giá trị 2.
InitialDirectory	Giá trị thuộc tính là một đường dẫn đến một thư mục hoặc ổ đĩa. Khi hộp thoại OpenFileDialog được mở lên sẽ hiển thị nội dung của đường dẫn này đầu tiên.
RestoreDirectory	Mang hai giá trị True hoặc False, cho biết hộp thoại có trả lại đường dẫn thư mục vừa mở trước đó. Lưu ý: Thuộc tính chỉ có hiệu lực khi thuộc tính InitialDirectory mang giá trị rỗng

4.7.5. Điều khiển SaveFileDialog

Thuộc tính	Mô tả
CheckFileExists	Mang hai giá trị True hoặc False. Nếu là True sẽ cho phép hộp thoại hiển thị một cảnh báo nếu người dùng chỉ định một tập tin không tồn tại và nhấn nút Open.
CheckPathExists	Mang hai giá trị True hoặc False. Nếu là True sẽ kiểm tra đường dẫn tới tập tin có hợp lệ hay không.
AddExtension	Mang hai giá trị True hoặc False. - Nếu là True sẽ cho phép thêm vào tên mở rộng (jpg, gif, ..) vào tập tin. - Nếu là False: Không cho phép
DefaultExt	Thêm tên mở rộng (.jpg, .gif, …) cho tập tin nếu người dùng không cung cấp tên mở rộng
OverwritePrompt	Mang hai giá trị True và False. Nếu là True sẽ hiện cảnh báo khi người dùng ghi đè vào một tập tin đã tồn tại
CreatePrompt	Mang hai giá trị True và False. Nếu là True sẽ hiện thông báo khi người dùng lưu một tập tin mới

4.7.5. Điều khiển SaveFileDialog

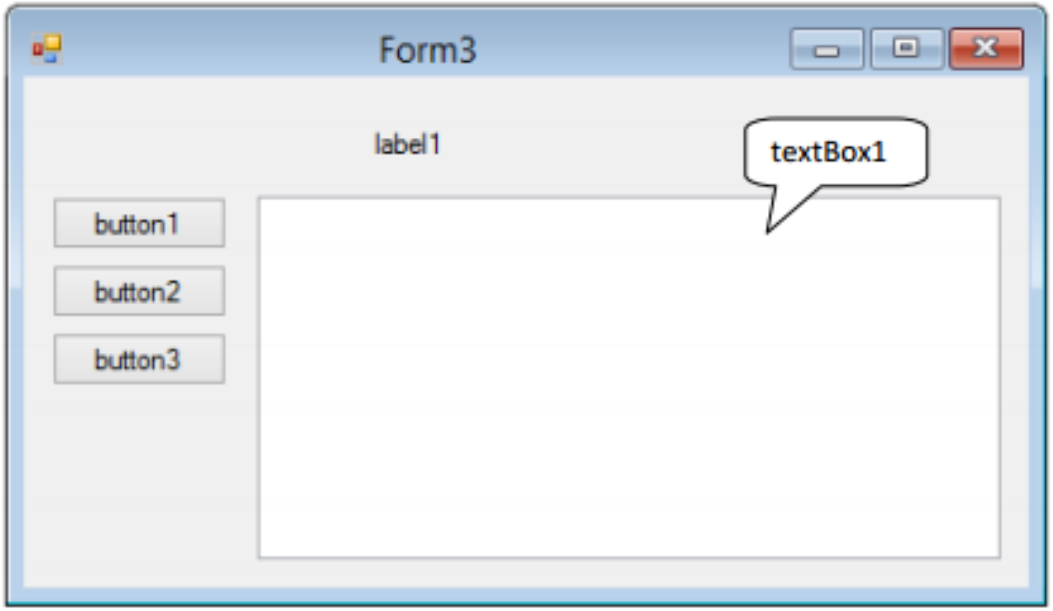
- Ví dụ: Viết chương trình soạn thảo văn bản đơn giản sử dụng *OpenFileDialog* để mở tập tin và *SaveFileDialog* để lưu tập tin.



Yêu cầu: Người dùng nhấn nút "Mở tập tin" để mở một tập tin có định dạng .txt. Nội dung của tập tin này sẽ hiển thị trên *TextBox* (chỉ cho phép hiển thị tập tin .txt) Người dùng nhấn nút "Lưu tập tin" để lưu nội dung vừa soạn thảo trong *TextBox* vào một thư mục nào đó trên hệ thống.

4.7.5. Điều khiển SaveFileDialog

- Bước 1: Thiết kế giao diện chương trình. Lập trình viên thêm các điều khiển: *Label*, *TextBox*, *Button*, *OpenFileDialog* và *SaveFileDialog* từ cửa sổ Toolbox vào form như hình



4.7.5. Điều khiển SaveFileDialog

- Bước 2: Thiết lập giá trị thuộc tính cho điều khiển trong cửa sổ Properties
  - ✓ label1:
    - Thuộc tính *Text*: “Soạn thảo văn bản”
    - Thuộc tính *Font Size*: 14
  - ✓ button1:
    - Thuộc tính *Name*: btnMoTapTin
    - Thuộc tính *Text*: “Mở tập tin”
  - ✓ button2:
    - Thuộc tính *Name*: btnLuuTapTin
    - Thuộc tính *Text*: “Lưu tập tin”
  - ✓ button3:
    - Thuộc tính *Name*: btnThoat
    - Thuộc tính *Text*: “Thoát”

4.7.5. Điều khiển SaveFileDialog

- ✓ textbox1:
  - Thuộc tính *Name*: txtNoiDung
  - Thuộc tính *MultiLine*: True
  - Thuộc tính *Size*: 93, 55
- ✓ openFileDialog1:
  - Thuộc tính *RestoreDirectory*: True
  - Thuộc tính *Title*: “Mở tập tin Text để soạn thảo”
  - Thuộc tính *Filter*: “File Text|*.txt”
- ✓ saveFileDialog1:
  - Thuộc tính *RestoreDirectory*: True
  - Thuộc tính *Title*: “Lưu tập tin Text”
  - Thuộc tính *Filter*: “File Text|*.txt”
  - Thuộc tính *OverwritePrompt*: True

4.7.5. Điều khiển SaveFileDialog

- Sự kiện *Click* của nút btnMoTapTin:

```
private void btnMoTapTin_Click(object sender, EventArgs e)
{
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
string tentaptin = openFileDialog1.FileName;
StreamReader rd = new StreamReader(tentaptin);
txtNoiDung.Text = rd.ReadToEnd();
rd.Close();
}}
```

- Sự kiện *Click* của nút btnLuuTapTin:

```
private void btnLuuTapTin_Click(object sender, EventArgs e)
{
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
string noidung = txtNoiDung.Text;
string tentaptin = saveFileDialog1.FileName;
StreamWriter wt = new StreamWriter(tentaptin);
wt.Write(noidung);
wt.Close();
}}
```