

Lab 03:

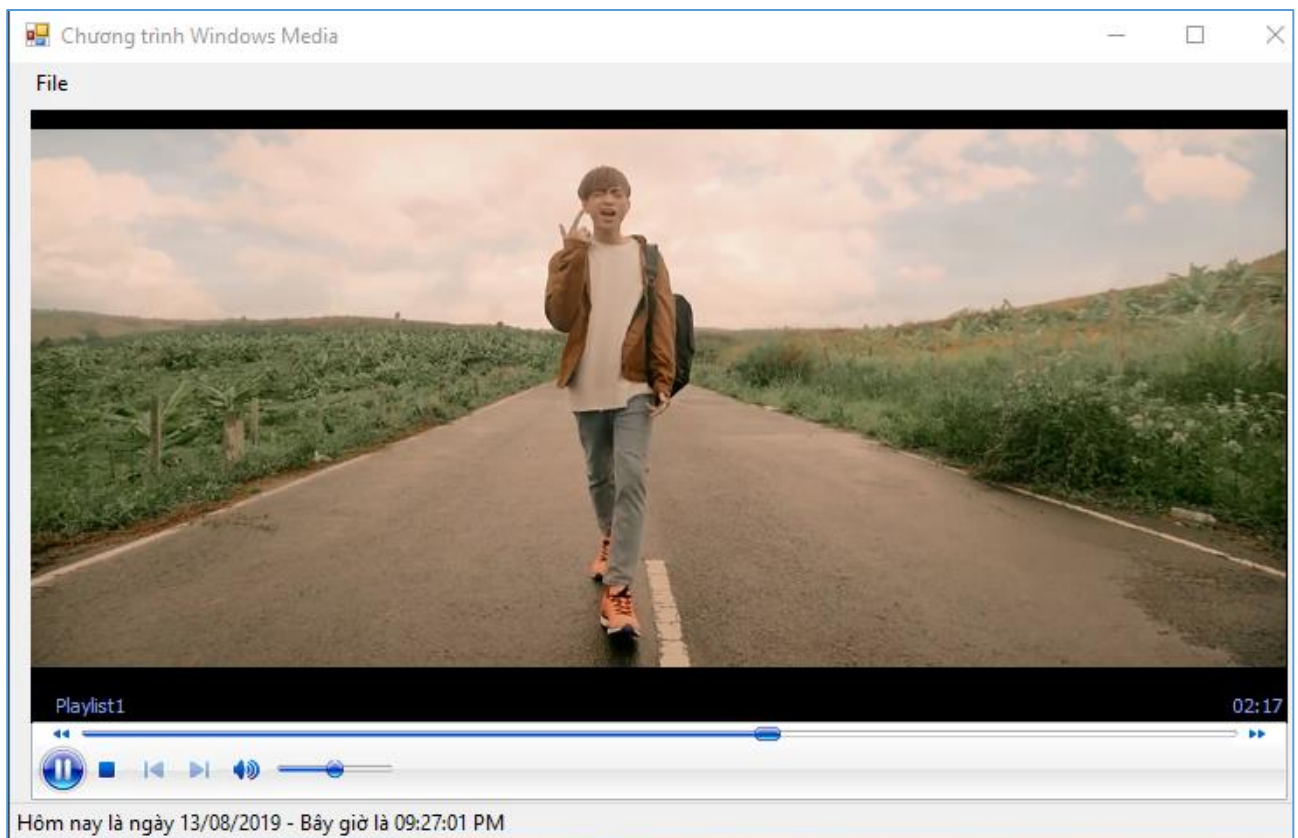
LẬP TRÌNH WINDOWS FORM VỚI GIAO DIỆN Multiple Document Interface (MDI)

A. MỤC TIÊU:

- ✓ Sử dụng Visual Studio .NET 2015/2017/2019 tạo ứng dụng dạng Windows Forms MDI
- ✓ Làm quen với các control
 - *MenuStrip*
 - *ToolStrip*
 - *StatusStrip*
 - *Timer*
 - *Media*
 - *ProgressBar*

B. NỘI DUNG:

Bài Tập 1: Tạo Project **Lab03** mô phỏng điều khiển Windows media Player để phát các tập tin Audio và Video.



Hình 1: Thiết kế giao diện có windows media

Yêu cầu:

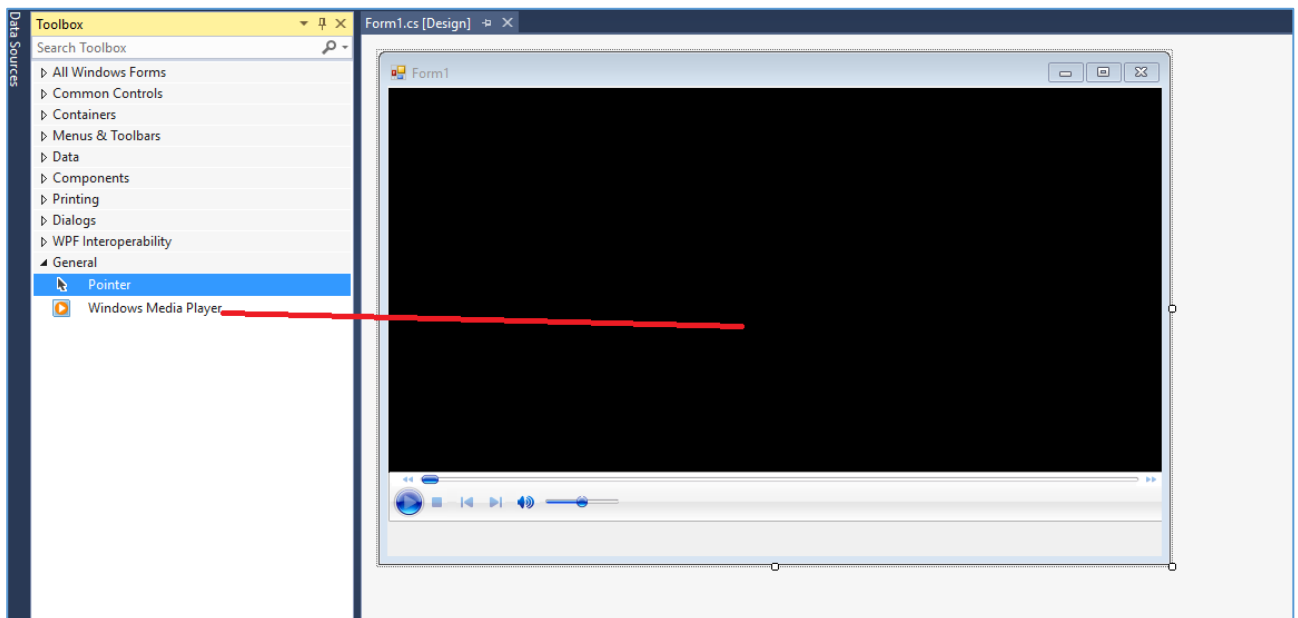
- Ứng dụng chứa Windows Media Player control cho phép Play các file video/sound theo nhiều dạng format:(*.avi *.mpeg *.wav *.midi *.mp4 *.mp3)
- Menu Bar có 1 Menu **File** và 2 Sub Menu **Open** và **Exit**

- Khi chọn Sub Menu **Exit** sẽ thoát ứng dụng
- Khi chọn Sub Menu **Open** sẽ mở hộp thoại Open File để chọn mở file Media - Control Multi Media sẽ phát file Media đã chọn.
- **StatusStrip** hiển thị: Ngày giờ hệ thống và thay đổi giờ theo mỗi giây.

Hướng dẫn:

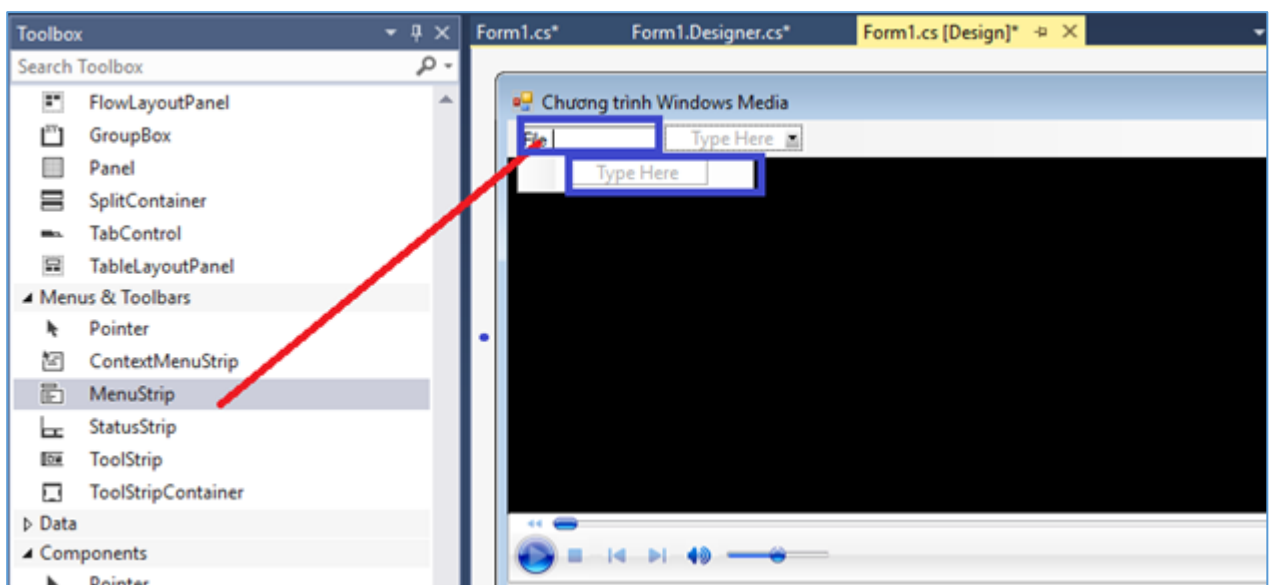
Bước 1: Thiết kế giao diện

- ✓ Để Sử dụng điều khiển Windows Media Player để phát các tập tin nhạc và phim. Để sử dụng được điều khiển Windows Media Player click phải lên ToolBox và chọn Choose Items, trong cửa sổ Choose Toolbox Item, chọn thẻ COM Components và chọn thư viện Windows Media Player.



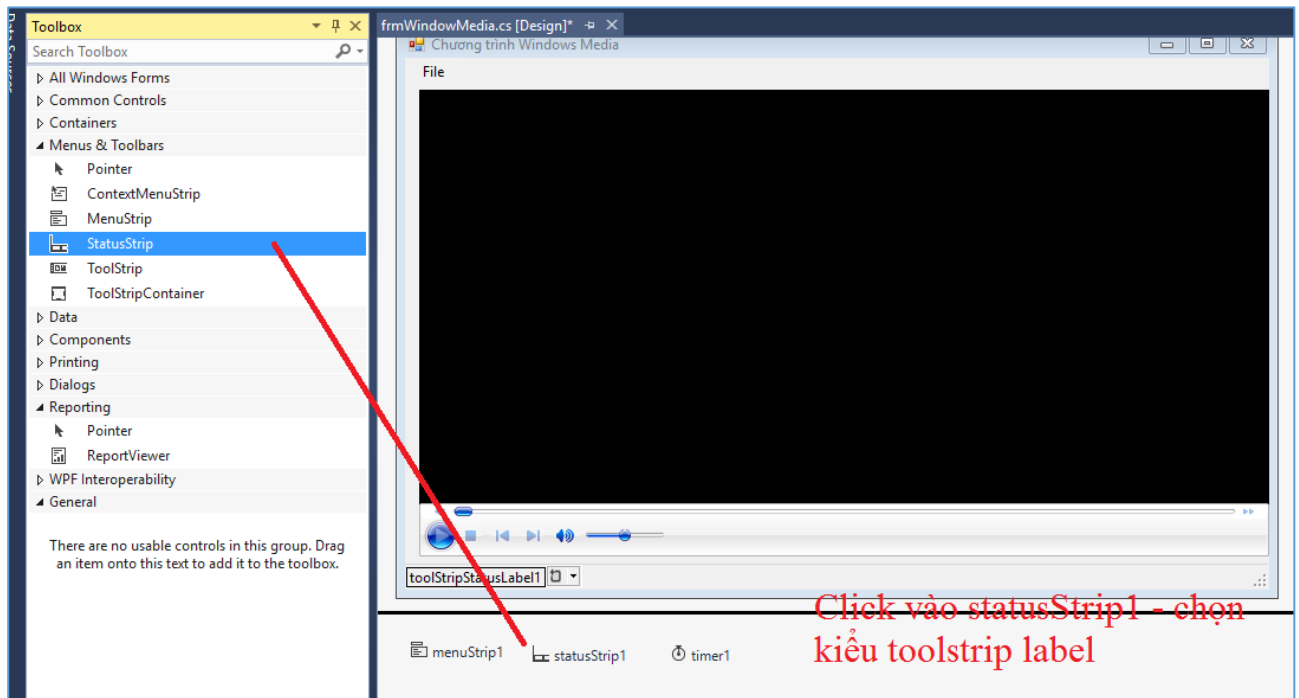
Hình 2: Kéo thả control Windows Media Player vào Giao diện Form

- ✓ Để tạo Menu **File**: Kéo thả MenuStrip trên toolbar (Menu & ToolBars)



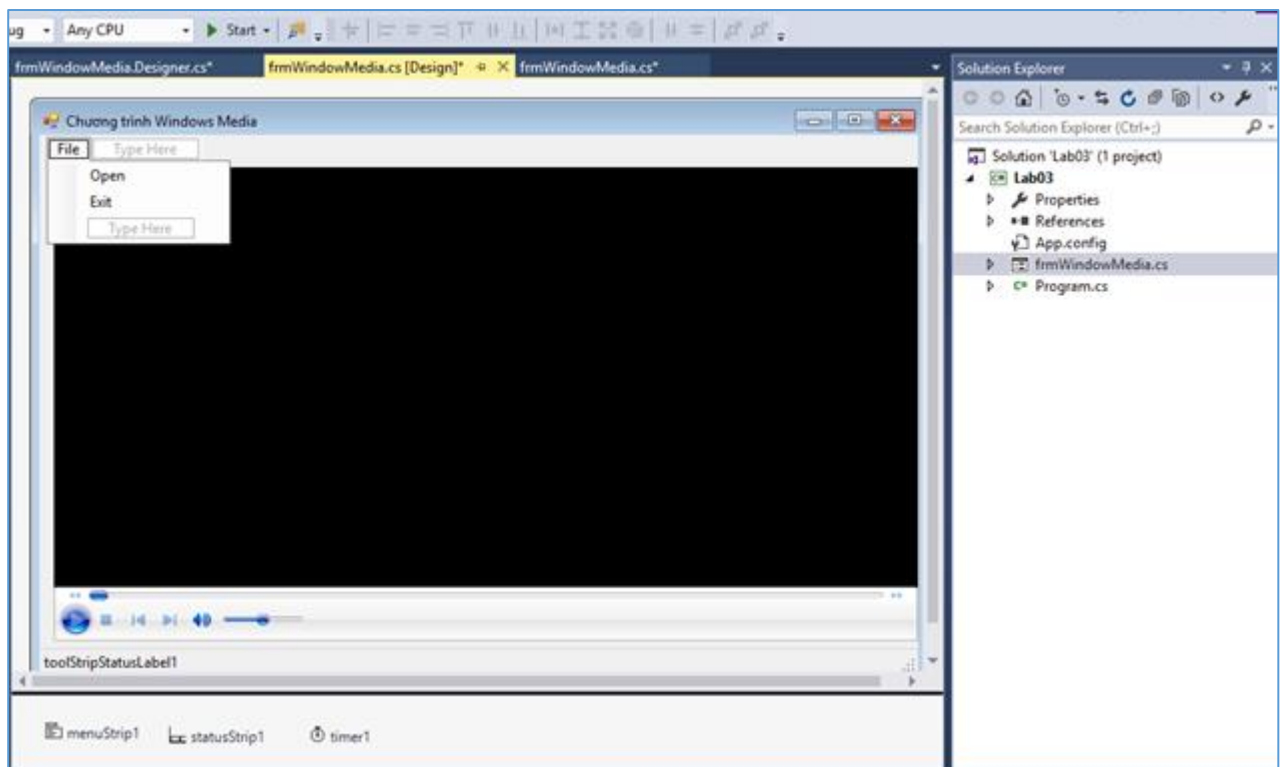
Hình 3: Kéo MenuStrip để tạo Menu

✓ Để tạo Status: Kéo thả StatusStrip



Hình 4: Sử dụng StatusStrip

- ✓ Để thực hiện các sự kiện liên quan đến thời gian: Kéo **Timer** vào forms
- ✓ Giao diện sau thiết kế



Hình 5: Giao diện sau khi thiết kế

Bước 2: Viết các lệnh xử lý

✓ Viết lệnh xử lý cho sự kiện *Timer*

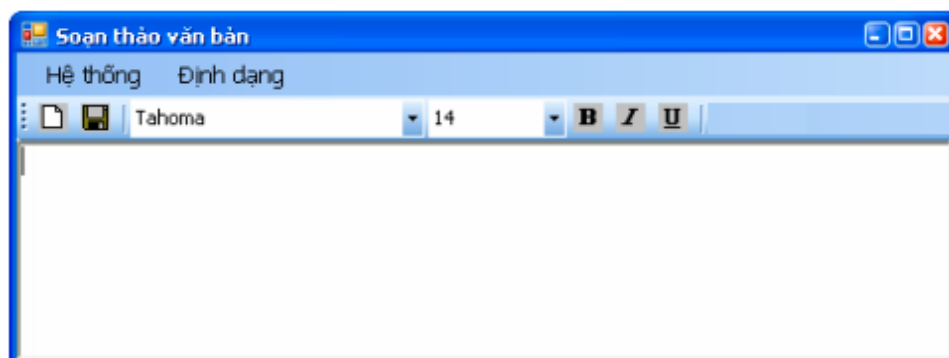
- Thiết lập thuộc tính của Timer **Interval:1000** (Đơn vị ms, cứ mỗi 1 giây sự kiện tick sẽ xảy ra)
- Chọn thuộc tính Enable = True để Timer được hoạt động
- Viết lệnh xử lý cho sự kiện **Timer1_Tick()**

```
private void timer1_Tick(object sender, EventArgs e)
{
    this.toolStripStatusLabel1.Text = string.Format("Hôm nay là ngày
{0} - Bây giờ là {1}", DateTime.Now.ToString("dd/MM/yyyy"),
DateTime.Now.ToString("hh:mm:ss tt"));
}

✓ Xử lý menu Open Sử dụng OpenFileDialog
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Tạo hộp thoại mở file
    OpenFileDialog dlg = new OpenFileDialog();
    //lọc hiện thị các loại file
    dlg.Filter = "AVI file| *.avi | MPEG File | *.mpeg | Wav File |
*.Wav | Midi File | *.midi | Mp4 File | *.mp4 | MP3 | *.mp3";
    //hien thi openFileDialog
    if (dlg.ShowDialog() == DialogResult.OK)
        axWindowsMediaPlayer1.URL = dlg.FileName; //Lấy tên file
        cần mở
}
```

Bài tập 2: Soạn thảo văn bản

Yêu cầu: Thêm project **Lab03-02** vào Solution Lab03 có giao diện như sau

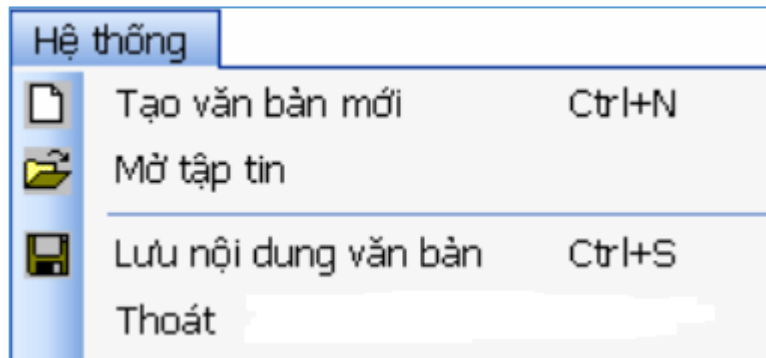


Sử dụng **RichTextBox** để thiết kế điều khiển hiển thị và nhập nội dung văn bản:

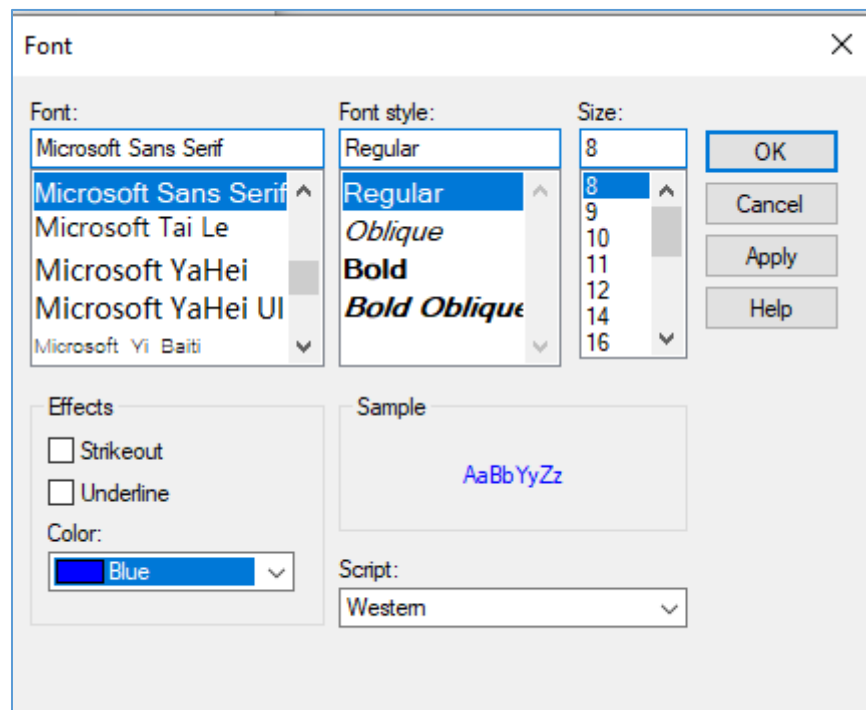
Sử dụng công cụ **MenuStrip** để tạo Hệ Thống và Định Dạng

Sử dụng công cụ **ToolStrip** để tạo thanh công cụ chứa các button và image như trên

✓ Trong menu Hệ thống thiết kế như sau



✓ Khi click vào menu định dạng gọi **FontDialog** có sẵn của windows




Hướng Dẫn cách gọi


```
FontDialog fontDlg = new FontDialog();
fontDlg.ShowColor = true;
fontDlg.ShowApply = true;
fontDlg.ShowEffects = true;
fontDlg.ShowHelp = true;
if (fontDlg.ShowDialog() != DialogResult.Cancel)
{
    richText.ForeColor= fontDlg.Color;
    richText.Font = fontDlg.Font;
}
```


✓ Cài đặt xử lý cho các chức năng

2.1 Khi mới mở Form, thực hiện:

- Tạo dữ liệu cho ComboBox Font: chứa tất cả các Font chữ của hệ thống.
- Tạo dữ liệu cho ComboBox Size: chứa các giá trị từ 8 → 72.
(8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72)
- Tạo giá trị mặc định là Font **Tahoma**, Size **14**

2.2 Khi chọn Tạo văn bản mới (hoặc nhấn nút ): Xóa nội dung hiện có trên RichTextBox và khởi tạo các giá trị mặc định như Font, Size, ...

2.3 Khi chọn Mở tập tin (hoặc nhấn nút ): Hiện thị hộp thoại mở tập tin (OpenFileDialog) cho phép người dùng chọn tập tin văn bản (*.txt hoặc *.rtf) để mở.

2.4 Khi chọn Lưu nội dung văn bản (hoặc nhấn nút ): Lưu nội dung văn bản trên RichTextBox xuống tập tin. Nếu là văn bản mới và trước đó chưa lưu lần nào thì hiện thị hộp thoại lưu tập tin (SaveFileDialog) cho phép người dùng chọn thư mục cần lưu tập tin với kiểu tập tin cần lưu là *.rtf.

Nếu là văn bản đã được mở trước đó thì thông báo cho người dùng lưu văn bản thành công.

2.5 Nút **B**: Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được in đậm hay không

Nút **I**: Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được in nghiêng hay không

Nút **U**: Khi chọn, tùy thuộc vào trạng thái của nút để xử lý nội dung của vùng văn bản đang được chọn có được gạch dưới hay không.

Hướng Dẫn

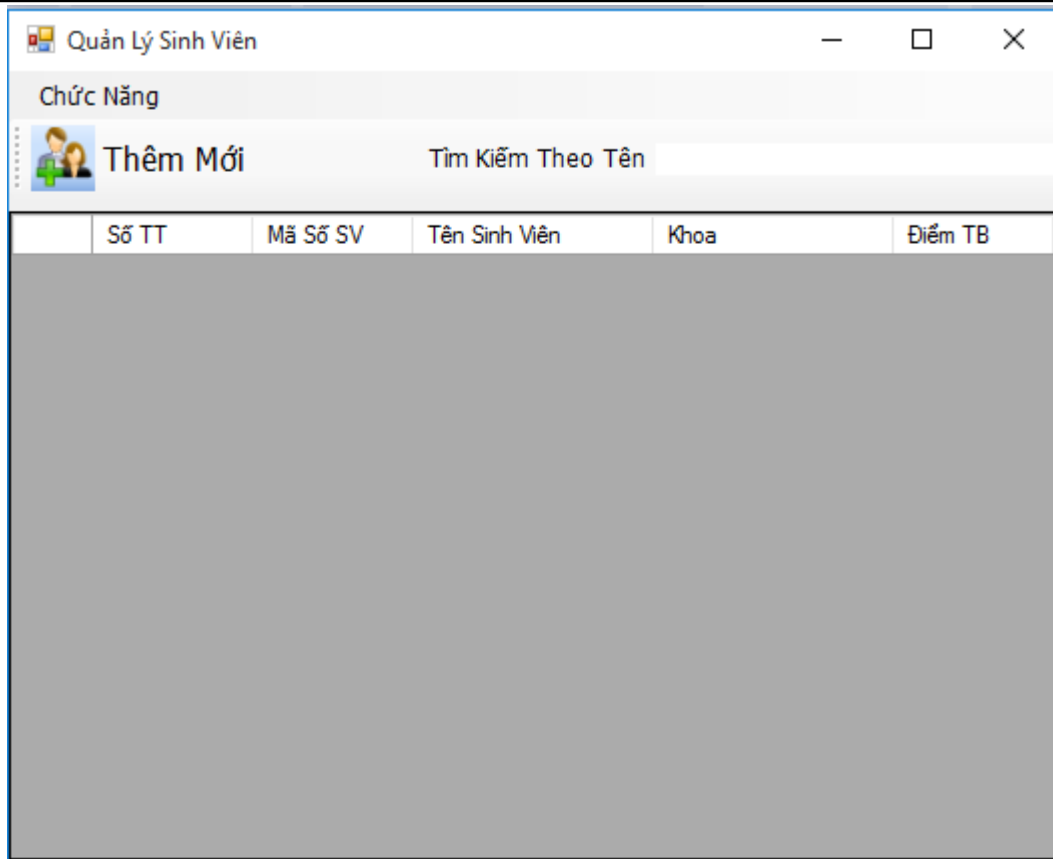
- Lấy tất các fonts từ hệ thống đưa vào combobox Fonts
- ```
foreach (FontFamily font in new InstalledFontCollection().Families)
{
 cmbFonts.Items.Add(font.Name);
}
```
- Để load file sử dụng richText.LoadFile(...);
  - Để Save file sử dụng richText.SaveFile(...);

### **Bài tập 3: Viết Chương Trình Nhập Liệu Sinh Viên Đơn giản sau**

- Form Quản lý sinh viên có Menu Chức năng gồm 2 SubMenu là **Thêm mới** và **Thoát**

Sử dụng MenuStrip để tạo menu

Sử dụng ToolStrip để tạo hình và nút bấm Thêm mới, Tìm kiếm theo tên



- Khi click vào menu thêm mới (Ctrl + N) hoặc ở icon hình thêm mới trên toolStrip thì gọi Form nhập liệu sinh viên như sau

Có 3 khoa được hiện thị trong Combobox (Style là DropDownList) gồm “Công nghệ thông tin”, “Ngôn ngữ Anh” và “Quản trị kinh doanh”

Khi click vào Nút Thêm Mới

- ✓ Thông báo cho người dùng các thông tin bắt buộc phải nhập liệu (Mã số, Tên Sinh Viên, Điểm)

- ✓ Thông báo cho người dùng thông tin mã số sinh viên nếu bị trùng trong DataGridView ở Form trước đó
- ✓ Thông báo cho người dùng dữ liệu nhập điểm trong phạm vi từ (0 -10)
- ✓ Trở về Form chính và đưa dữ liệu vào GridView hiển thị

Ở phần tìm kiếm theo tên sinh viên, khi Textbox tìm kiếm thay đổi thì luôn luôn tìm lại dữ liệu chứa tên tìm kiếm (không phân biệt hoa thường)



|   | Sđ TT | Mã Số SV | Tên Sinh Viên  | Khoa                | Điểm TB |
|---|-------|----------|----------------|---------------------|---------|
| ▶ | 1     | 1        | Nguyễn văn Bảo | Công nghệ thông tin | 5.6     |
|   | 2     | BH030343 | Phạm Chí Bình  | Công nghệ thông tin | 8.9     |

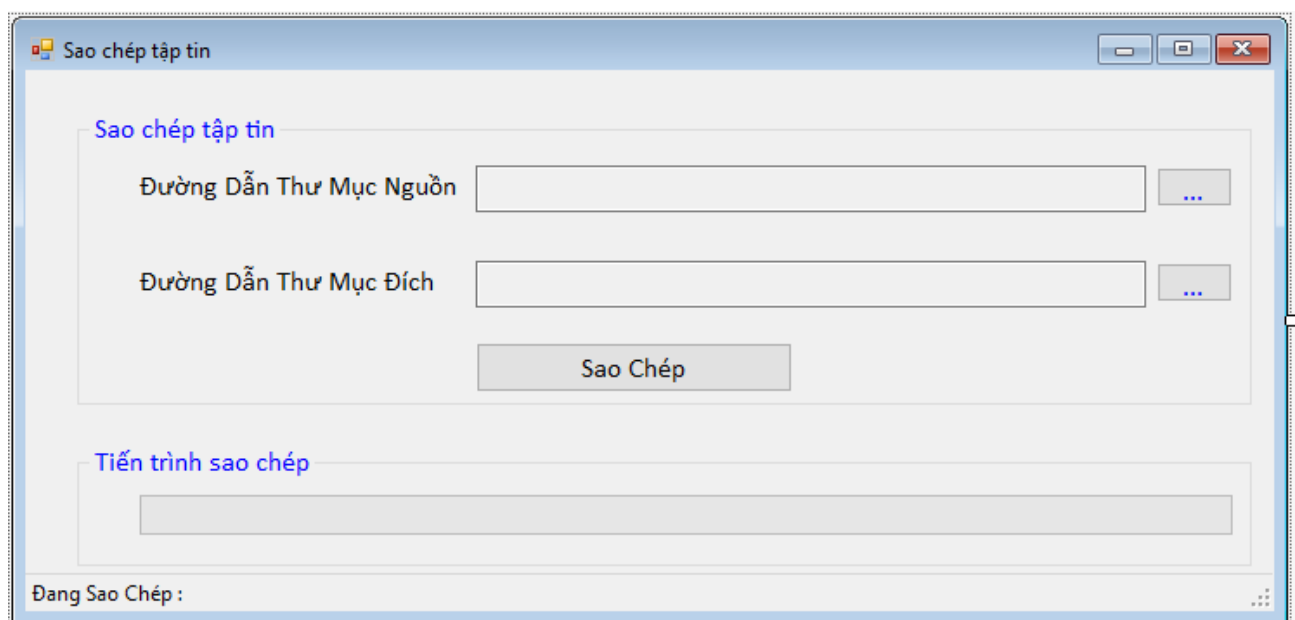
#### Hướng Dẫn:

- Thêm 1 lớp `Student` giúp cho việc lưu trữ dữ liệu ở form nhập liệu
- Tạo 1 List lưu trữ dữ liệu là biến toàn cục `static`

```
Public static List<Student> listStudent = new List<Student>();
```
- Xử lý trên `listStudent` toàn cục vừa tạo

#### Bài tập 4: Tiến Trình sao chép tập tin ( Sử dụng điều khiển ProgressBar và Tooltip)

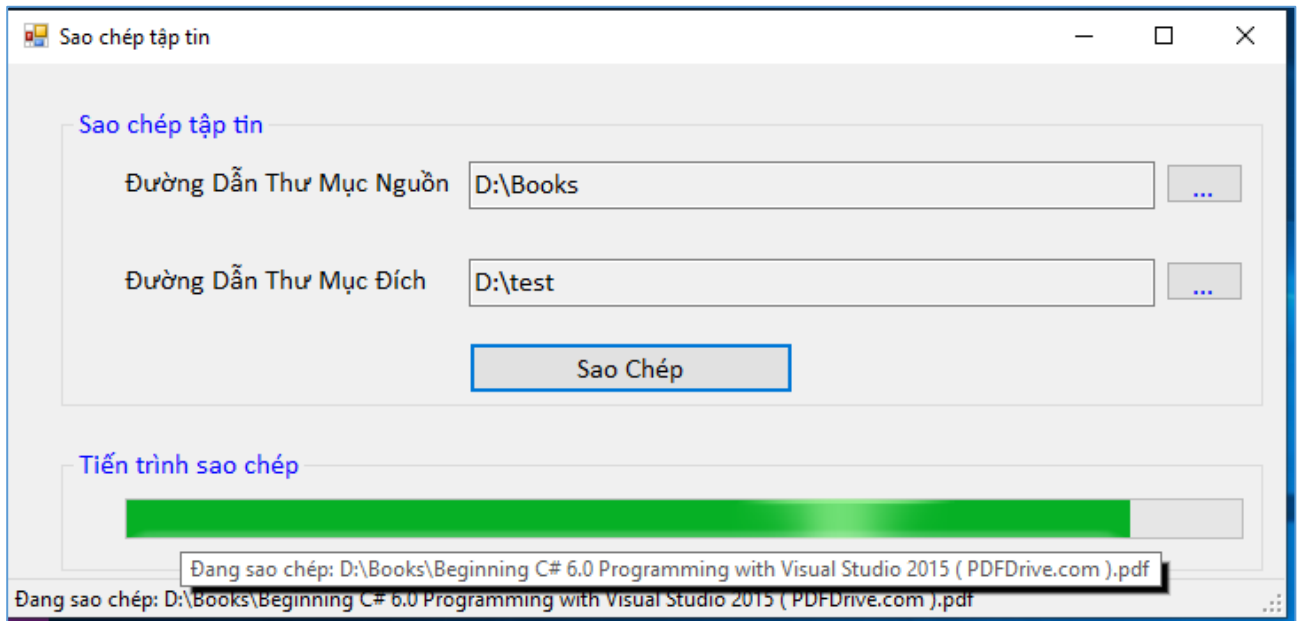
##### Yêu Cầu



Hình 6: Giao diện thiết kế chương trình sao chép file



- ✓ 2 textbox nguồn và đích có ReadOnly = True, Người dùng chọn từ nút ... để chọn đường dẫn folder đưa vào
- ✓ Sử dụng điều khiển **ProgressBar** để hiển thị tiến trình sao chép tập tin
- ✓ Sử dụng điều khiển **ToolTip** để hiển thị câu chú thích ý nghĩ cho các điều khiển khi rê con trỏ chuột lên điều khiển, và tương tự hiện **StatusStrip** ở phía dưới



- ✓ Khi hoàn tất quá trình thông báo thành công và tổng số file được sao chép cho người sử dụng
- ✓ Khi nhấn nút Sao chép tập tin, thực hiện:
  - Kiểm tra xem đường dẫn thư mục nguồn và thư mục đích có hợp lệ hay không. Nếu không hợp lệ xuất thông báo cụ thể cho người dùng biết.
  - Nếu đường dẫn hợp lệ, thực hiện sao chép tất cả các tập tin có trong thư mục nguồn sang thư mục đích và thể hiện tiến trình sao chép tập tin như hình trên.

## Hướng Dẫn

- ✓ Sử dụng OpenFileDialog để chọn folder

```
private void btnAdd_Click(object sender, EventArgs e)
{
 OpenFileDialog folderBrowser = new OpenFileDialog();
 folderBrowser.ValidateNames = false;
 folderBrowser.CheckFileExists = false;
 folderBrowser.CheckPathExists = true;
 // Always default to Folder Selection.
 folderBrowser.FileName = "Folder Selection";
 if (folderBrowser.ShowDialog() == DialogResult.OK)
 {
 string folderPath = Path.GetDirectoryName(folderBrowser.FileName);
 txtSource.Text = folderPath; //đưa txt nguồn = folderPath chọn
 }
}
private void btnCopy_Click(object sender, EventArgs e)
{
 string msg = "Đang sao chép: {0}";
 if (Directory.Exists(txtSource.Text) && Directory.Exists(txtDes.Text))
```

```

 {
 string[] files = Directory.GetFiles(txtSource.Text);
 int step = 100 / files.Count();
 foreach(string f in files)
 {
 string fileName = Path.GetFileName(f);
 string desFile = Path.Combine(txtDes.Text, fileName);
 File.Copy(f, desFile);
 progressBar1.Value += step;
 msg = string.Format(msg, Path.Combine(txtSource.Text,
fileName));
 LoadStatusStrip(msg);
 }
 }
 }

 void LoadStatusStrip(string status)
 {
 this.toolStripStatusLabel1.Text = status;
 ToolTip tooltip = new ToolTip();
 tooltip.SetToolTip(progressBar1, status);
 }

```

- ✓ Một số hàm sử dụng trong xử lý file( các class sử dụng File, Directory, Path):

`File.Copy(srcFile, desFile)` : Copy từ file nguồn sang đích

`Directory.GetFiles(txtSource.Text)` : Lấy tên file trong đường dẫn nguồn

`Path.Combine(txtSource.Text, fileName)`: Nối đường dẫn file

`Directory.Exists(txtSource.Text)` : Kiểm tra đường dẫn thư mục có tồn tại hay không

`Directory.GetFiles(txtSource.Text)`: Lấy tất cả đường dẫn file trong thư mục

-----Hết Lab 03 -----