

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG KỸ THUẬT LÝ TỰ TRỌNG

-----o0o-----

ĐỀ TÀI SÁNG KIẾN KINH NGHIỆM
NĂM HỌC 2015 - 2016

BÀI TẬP HỌC PHẦN LẬP TRÌNH WINDOWS

BIÊN SOẠN: Nguyễn Duy Đạt

Lưu hành nội bộ

LỜI GIỚI THIỆU

Môn học Lập trình Windows thuộc về khối kiến thức chuyên ngành, dành cho mọi sinh viên thuộc ngành Công nghệ phần mềm, môn học này đem đến cho sinh viên những kiến thức quan trọng để tham gia hoạt động của công ty sau này. Bên cạnh đó, đây còn là khối kiến thức bắt buộc phải biết mà còn phải nắm vững để vận dụng linh hoạt trong môi trường công tác sau khi tốt nghiệp cũng như thích ứng với thị trường lao động sáng tạo, nhiều thay đổi sau này.

Hệ thống bài tập lập trình Windows này dành cho sinh viên bậc cao đẳng, hướng tới những kiến thức liên quan đến ngôn ngữ lập trình C# từ những kiến thức cơ bản cho tới xây dựng dự án lập trình. Bộ tài liệu này trình bày các bài luyện tập theo từng chủ đề và thứ tự đề cập giống như trong giáo trình lý thuyết đã học. Mỗi chương sẽ trình bày một vấn đề lớn, bao gồm nhiều bài nhỏ. Mỗi chủ đề sẽ được thể hiện theo thứ tự chính như sau: Tóm tắt lý thuyết của chương, hướng dẫn giải bài tập cơ bản và gợi ý một số bài tập nâng cao. Bài tập trong từng phần của chương đều có ghi rõ: Yêu cầu, hình minh họa, gợi ý và mã lập trình mẫu, một số phần còn có kết quả thực tế khi thực thi chương trình. Từ những bài tập gợi ý này, sinh viên sẽ có tài liệu để tham khảo, giúp cho sinh viên sẽ tổng hợp và nâng cao kiến thức và những kỹ năng quan trọng khi lập trình như kỹ năng tư duy giải quyết vấn đề, phương pháp tìm hiểu một vấn đề mới, phân bổ tài nguyên hợp lý, thiết kế được mô hình để lập trình.

Bộ bài tập này ngoài chức năng dùng cho mục đích giảng dạy, củng cố kiến thức cho sinh viên mà còn hiệu quả khi dùng làm tài liệu tham khảo chính cho những môn gần với môn học này. Những giáo viên dạy môn này ở nhiều bậc đào tạo khác nhau sẽ dựa vào chương trình khung của môn học để từ đó chọn một số phần phù hợp trong tài liệu này nhằm hướng dẫn sinh viên thực hành với mục tiêu củng cố kiến thức, cải thiện kỹ năng cần thiết trong chương trình học.

Tác giả trân trọng giới thiệu tới bạn đọc Bộ bài tập lập trình Windows này với hy vọng đem lại lợi ích không những trong công tác giảng dạy mà còn củng cố thêm hiểu biết, kỹ năng lập trình của sinh viên khi xây dựng ứng dụng trên hệ điều hành Windows sử dụng ngôn ngữ C#.

Với những lý do trên, tác giả mạn phép thực hiện bộ bài tập này. Trong quá trình thực hiện bộ bài tập, tác giả đã nhận được những ý kiến bổ sung vô cùng quý giá từ thầy cô cùng khoa, vì vậy tác giả gửi lời cảm ơn chân thành đến tất cả thầy cô đang công tác tại Khoa công nghệ thông tin trường Cao Đẳng Kỹ Thuật Lý Tự Trọng Thành phố Hồ

Chí Minh. Xuyên suốt khoảng thời gian vừa qua, những ý kiến đóng góp vô cùng giá trị của quý thầy cô đã giúp tác giả hoàn thiện nội dung của bộ tài liệu lập trình Windows cả về chuyên môn lẫn hình thức trình bày.

Thành phố Hồ Chí Minh, ngày 21 tháng 03 năm 2016

Nguyễn Duy Đạt

Chương 1: NHỮNG ĐỐI TƯỢNG ĐIỀU KHIỂN CƠ BẢN (BASIC CONTROLS)

A. Tóm tắt lý thuyết

1. Giới thiệu

- Ngôn ngữ C# được tạo ra bởi công ty Microsoft, năm 2000.
- Ngôn ngữ này được xây dựng từ sự tiếp thu ưu điểm của ngôn ngữ C++, Java.
- Đem lại sự tiện lợi, hiệu quả, tốc độ khi lập trình cho lập trình viên.

2. Những điều khiển cơ bản

- Nhãn (Label): dùng để thể hiện nội dung của một button.
- Nhãn liên kết (Link Label): cho phép người dùng click và mở một trang web dựa vào địa chỉ cho sẵn.
- Hộp văn bản (TextBox): một đối tượng rất hay sử dụng khi lập trình, cho phép người dùng nhập vào thông tin, giá trị cần tính toán hay cũng dùng để thông báo cho người dùng biết kết quả.
- Hộp văn bản mặt nạ (Masked TextBox): cho phép người dùng nhập liệu với định dạng đã được chỉ định bởi người lập trình.
- Hộp kiểm (CheckBox): được tạo ra để người dùng chọn một hay vài lựa chọn trong một tập lựa chọn mà họ muốn.
- Nút chức năng (RadioButton): người dùng phải chọn một trong nhiều sự chọn lựa do người lập trình đưa ra.
- Hộp hình ảnh (PictureBox): thể hiện hình ảnh có sẵn hay được chỉ định bởi người lập trình hoặc người dùng chọn ảnh mà họ muốn.
- Thuộc tính: đây là những tính chất của một đối tượng lập trình.
- Phương thức: những hàm hoạt động của một đối tượng.
- Tình huống: những sự kiện mà đối tượng này có thể bắt và xử lý.

B. Bài tập cơ bản

Bài 1.1: Tạo form có giao diện như sau:



Yêu cầu:

- Khi form load tạo Link: lttc.edu.vn, như trên.
- Khi Click chuột vào Link: lttc.edu.vn mở trình duyệt web và truy xuất đến trang web lttc.edu.vn.

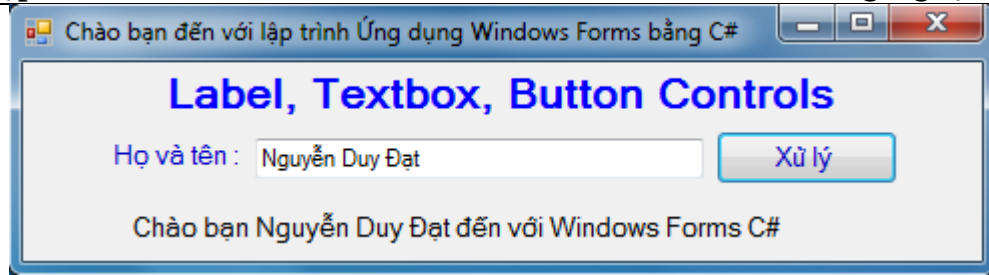
Hướng dẫn:

- Tạo mới project
- Nhấp phải trên tên project\Add\Windows Form, ...
- Nhập FrmDemoLinkLabel vào hộp name\Add
 - + Thuộc tính Text của Form có nội dung “LinkLabel Control”.
- Đặt các điều khiển lên form như hình trên.
 - + Điều khiển trên Form gồm 1 Label và 1 LinkLabel.
 - + Label.Text = “LinkLabel Control”, Font có Size là 16, Màu chữ màu xanh.
 - + linkLabel1.Text = “Web site Trường CD Kỹ Thuật Lý Tự Trọng Tp.HCM
lttc.edu.vn”
- Viết Code cho các sự kiện như sau.

```
private void FrmDemoLinkLabel_Load(object sender, EventArgs e)
{
    linkLabel1.Links.Clear();
    linkLabel1.Links.Add(47, 11, "http://lttc.edu.vn");
}
```

```
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("http://lttc.edu.vn");
}
```

Bài 1.2: Tạo form có giao diện như sau:



Yêu cầu:

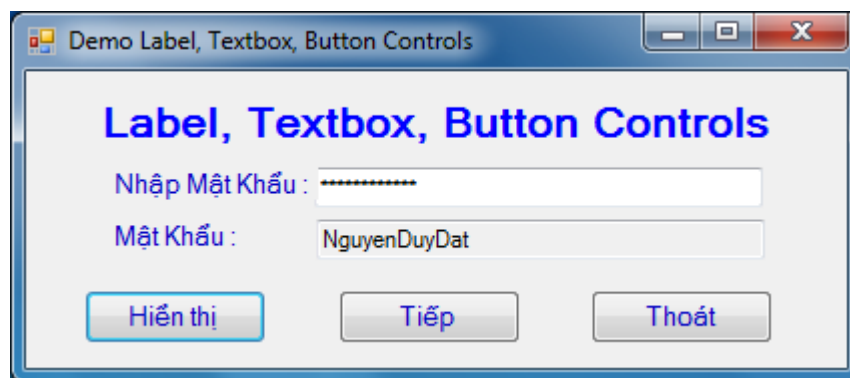
- Thiết kế form như trên.
- Khi nhập thông tin Họ và tên người dùng và Click chuột vào nút lệnh “Xử lý”. Xuất hiện thông tin Chào bạn + Tên người dùng vừa nhập + đến với Windows Forms C#.

Hướng dẫn:

- Nhấp phải trên tên project\Add\Windows Form, ...
- Nhập FrmDemoLabelTextboxButton vào hộp name\Add
- Đặt các điều khiển lên form như hình trên.
- Viết Code cho các sự kiện như sau.

```
private void btn_xuly_Click(object sender, EventArgs e)
{
    lbl_chao.Text = "Chào bạn "+txt_hoveten.Text+" đến với
    Windows Forms C#";
}
```

Bài 1.3: Tạo form có giao diện như sau:



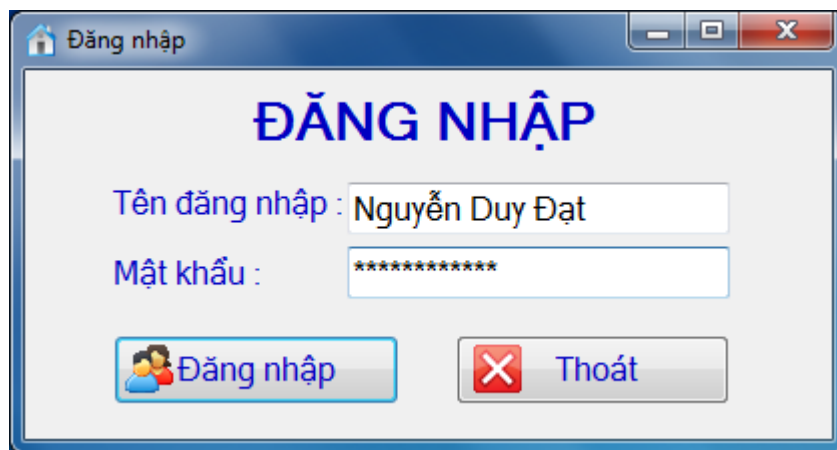
Yêu cầu:

- Khi người dùng nhập mật khẩu thì hiển thị các dấu “*”.
- Nhấn button hiển thị thì hiển thị mật khẩu trong Textbox mật khẩu.
- Nhấn button tiếp xóa hết dữ liệu trên các Textbox.
- Nhấn button thoát thì đóng chương trình.

Hướng dẫn:

- Nhấp phải trên tên project\Add\Windows Form, ...
- Nhập Frm_DemoLabelTextboxButton2 vào hộp name\Add
- Đặt các điều khiển lên form như hình trên.
- Textbox nhập mật khẩu đặt thuộc tính PasswordChar="*".
- Textbox mật khẩu đặt thuộc tính Read Only = True.

Bài 1.4: Tạo form đăng nhập có giao diện như sau:

**Yêu cầu:**

- Khi nhấn Enter thì kích hoạt sự kiện click nút **Đăng nhập** (AcceptButton = btnDangNhap)
- Khi nhấn esc thì kích hoạt sự kiện click nút **Thoát** (CancelButton=btnThoat)
- Khi nhấp nút **Đăng nhập** thì kiểm tra
 - Nếu không nhập *Tên đăng nhập* thì thông báo cho người dùng nhập vào với nội dung thông báo là **“Bạn chưa nhập tên đăng nhập. Xin hãy nhập vào”**
 - Nếu *Tên đăng nhập*=“admin” và *Mật khẩu* = “123456” thì xuất ra màn hình câu thông báo **“Đăng nhập thành công”** ngược lại xuất ra màn hình câu thông báo **“Đăng nhập thất bại”**
- Khi nhấp nút **Thoát** thì đóng form đăng nhập lại.
- Mở rộng:** Nếu đăng nhập 3 lần thất bại thực hiện đóng chương trình.

Hướng dẫn:

- Nhấp phải trên tên project\Add\Windows Form, ...
- Nhập frmLogin vào hộp name\Add
- Đặt các điều khiển lên form như hình trên.

- + Chọn thuộc tính AcceptButton là btnOK cho form frmLogin.
- + Chọn thuộc tính CancelButton là btnCancel form frmLogin.
- Viết Code cho các sự kiện như sau.
- + Double_click nút **btnDangNhap** rồi viết đoạn code sau

```
private void btnDangNhap_Click(object sender, EventArgs e)
{
    if (txtTenDN.Text.Equals(string.Empty))
        MessageBox.Show("Bạn chưa nhập tên đăng nhập", "thông báo");
    else
        if ((txtTenDN.Text.Equals("admin")) &&
            (txtMatKhau.Text.Equals("123456")))
        {
            MessageBox.Show("Đăng nhập thành công", "thông báo");
        }
        else
        {
            MessageBox.Show("Đăng nhập thất bại", "Thông báo");
        }
}
```

- a. Double_click nút btnThoat rồi viết đoạn code sau

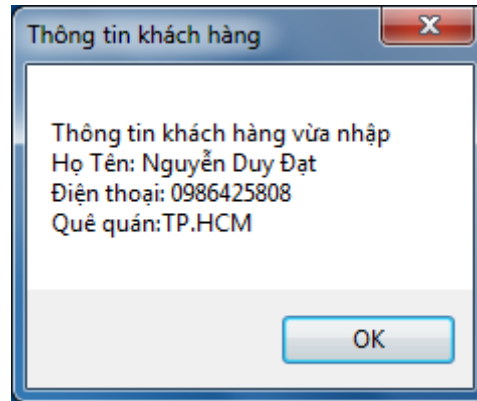
```
private void btnThoat_Click(object sender, EventArgs e)
{
    Close();
}
```

Bài 1.5: Tạo form có giao diện như sau:

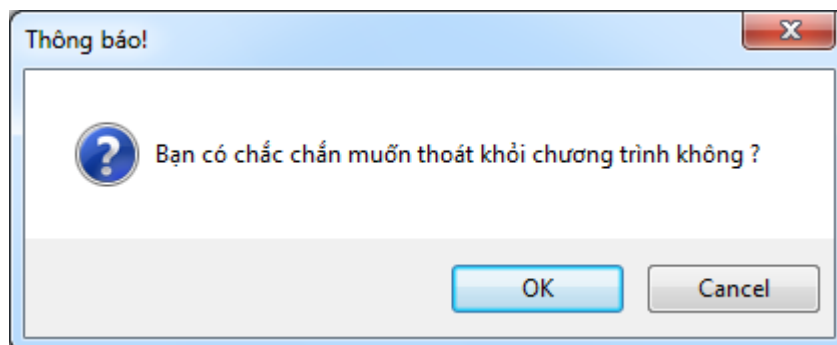
Yêu cầu:

- a. Khi nhấn Enter thì kích hoạt sự kiện click nút **Nhập**.
- b. Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.
- c. Khi người sử dụng nhập số điện thoại khách hàng, chương trình hiện danh sách tùy chọn ở textbox gồm các đầu số: 090, 091, 092, 093, 094, 095, 096, 098, 0911, 0912.

- d. Khi người sử dụng nhập quê khách hàng (tỉnh thành), chương trình hiện danh sách tùy chọn ở textbox gồm các tỉnh thành đề nghị gồm: Hà Nội, Thái Bình, Huế, Đà Nẵng, TP.HCM, Bình Dương, Bến Tre, Đồng Nai, Vũng Tàu, Tiền Giang, Trà Vinh, An Giang.
- e. Khi người sử dụng hoàn tất nhập dữ liệu nhấn Button Nhập thì chương trình xuất hiện thông tin khách hàng vừa nhập trên MessageBox như sau:



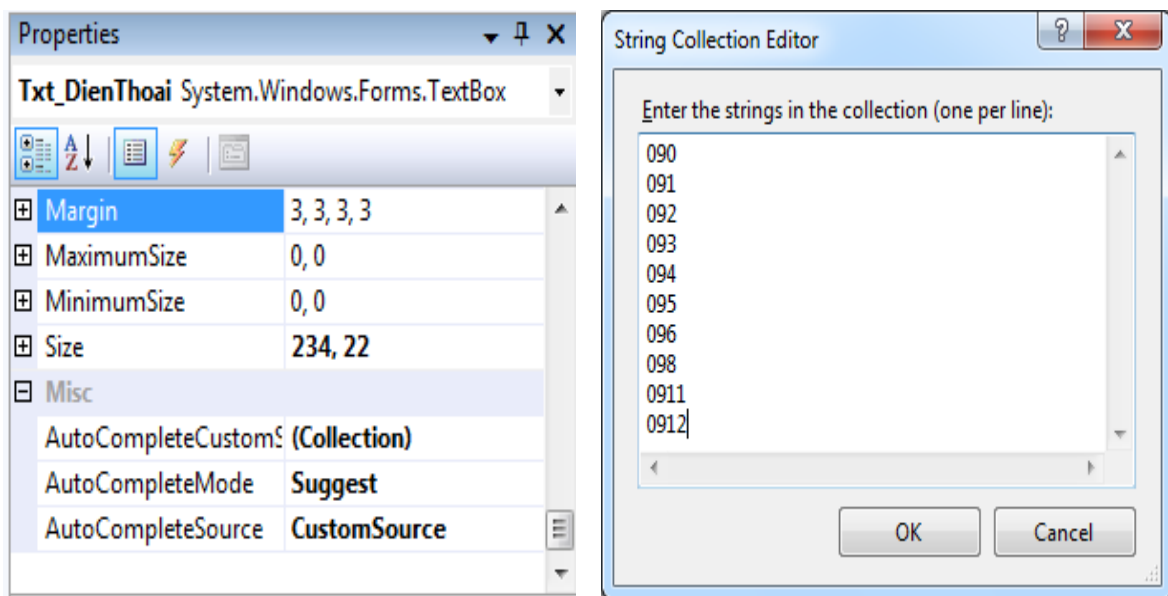
- f. Khi người sử dụng nhấn Button Thoát thì chương trình xuất hiện thông báo “Bạn có chắc chắn muốn thoát khỏi chương trình không?”. Nếu chọn OK thoát khỏi chương trình ngược lại không thoát chương trình.



- g. Thiết lập thuộc tính Anchor: Top, Left chạy chương trình và thay đổi kích thước của Form, quan sát sự thay đổi kích thước của các Controls trên form.
- h. Thiết lập thuộc tính Anchor: Top, Bottom, Left, Right chạy chương trình và thay đổi kích thước của Form, quan sát sự thay đổi kích thước của các Controls trên form.
- i. Thiết lập TabIndex theo thứ tự nhập liệu gồm: Họ tên, Điện thoại, Quê quán, Button Nhập, Button Thoát.
- j. Khi người sử dụng nhấn phím Enter tại TextBox Họ tên thì chuyển con trỏ qua TextBox Điện thoại, nhấn phím Enter tại TextBox Điện thoại thì chuyển con trỏ qua TextBox Quê quán, nhấn phím Enter tại TextBox Quê quán thì chuyển con trỏ qua Button Nhập.

Hướng dẫn:

- Nhấp phải trên tên project\Add\Windows Form, ...
- Nhập frm_ThongTinKH vào hộp name\Add
- Đặt các điều khiển lên form như hình trên.
 - + Chọn thuộc tính AcceptButton là btnNhap cho form frm_ThongTinKH.
 - + Chọn thuộc tính CancelButton là btnThoat cho form frm_ThongTinKH.
 - + Thiết lập danh sách dữ liệu hiển thị cho TextBox Điện thoại và Quê quán:
 - Thuộc tính **AutoCompleteCustomSource**-> Click (...) nhập dữ liệu.
 - Thuộc tính **AutoCompleteMode=SuggestAppend**
 - Thuộc tính **AutoCompleteSource=CustomSource**



- Viết Code cho các sự kiện như sau.
 - + Double_click nút **btnNhap** rồi viết đoạn code sau


```
private void Btn_Nhap_Click(object sender, EventArgs e)
{
    string hoten = Txt_HoTen.Text+" \n";
    string dienthoai = Txt_DienThoai.Text+" \n";
    string quequan = Txt_QueQuan.Text;
    MessageBox.Show("Thông tin khách hàng vừa nhập \n"+"Họ
    Tên:      "+hoten+"Điện      thoại:      "+dienthoai+"Quê
    quán:"+quequan,"Thông tin khách hàng");
}
```
 - + Double_click nút **btnThoat** rồi viết đoạn code sau


```
private void Btn_Thoat_Click(object sender, EventArgs e)
{
    DialogResult kq = MessageBox.Show("Bạn có chắc chắn
    muốn thoát khỏi chương trình không ?", "Thông
    báo!", MessageBoxButtons.OKCancel, MessageBoxIcon.Questi
    on);
```

```

        if (kq == DialogResult.OK)
            this.Close();
    }

```

- + Chọn sự kiện **KeyDown** hoặc **KeyPress** trong cửa sổ Property của lần lượt các TextBox Họ tên, Điện thoại, Quê quán, Double_click vào sự kiện **KeyDown** hoặc **KeyPress** rồi viết đoạn code sau:

```

private void Txt_HoTen_KeyDown(object sender, KeyEventArgs
e)
{
    if (e.KeyCode == Keys.Enter)
        Txt_DienThoai.Focus();
}

```

Hoặc

```

private void Txt_HoTen_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
        Txt_DienThoai.Focus();
}

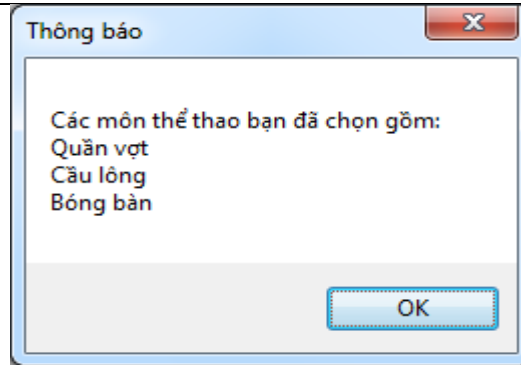
```

Bài 1.6: Tạo form có giao diện như sau:



Yêu cầu:

- Thiết kế form như hình trên.
- Khi nhấn Button Thoát kết thúc chương trình.
- Khi nhấn Button Chọn xuất hiện thông báo danh sách các môn thể thao yêu thích của bạn như sau:



Bài 1.7: Tạo form có giao diện như sau:



Yêu cầu:

- d. Khi ứng dụng khởi động lên hiển thị đèn tắt.
- e. Khi nhấn Button Mở đèn hiển thị Picture đèn sáng.
- f. Khi nhấn Button Tắt đèn hiển thị Picture đèn tắt.
- g. Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.
- h. Khi nhấn Button Thoát đóng chương trình.

Hướng dẫn:

- Nhập phải trên tên project\Add\Windows Form, ...
- Nhập frm_PictureBox vào hộp name\Add
- Đặt các điều khiển lên form như hình trên.
 - + Chọn thuộc tính CancelButton là btnThoat cho form frm_PictureBox.

+ Chọn thuộc tính SizeMode=StretchImage

- Viết Code cho các sự kiện như sau.

- //Sự kiện Click Button Mở đèn

```
private void Btn_MoDen_Click(object sender, EventArgs e)
{
    PtB_Den.Image =
        Image.FromFile("E:\\Lab06\\images\\DenSang.png");
}
```

//Sự kiện Click Button Tắt đèn

```
private void Btn_TatDen_Click(object sender, EventArgs e)
{
    PtB_Den.Image =
        Image.FromFile("E:\\Lab06\\images\\Dentat.png");
}
```

//Sự kiện Form Load

```
private void Frm_PictureBox_Load(object sender, EventArgs e)
{
    PtB_Den.Image =
        Image.FromFile("E:\\Lab06\\images\\Dentat.png");
}
```

C. Bài tập nâng cao

Bài 1.8: Tạo form có giao diện như sau:



Yêu cầu:

- Khi ứng dụng khởi động lên mặc định thuộc tính SizeMode của PictureBox là StretchImage.
- Khi chọn một trong các giá trị trong Size Mode thì hình trên PictureBox thay đổi theo Size Mode tương ứng vừa chọn.

c. Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.

d. Khi nhấn Button Thoát đóng chương trình.

--- HẾT ---

Chương 2: KỸ NĂNG THIẾT KẾ ỨNG DỤNG VÀ GỠ RỜI CHƯƠNG TRÌNH

A. Tóm tắt lý thuyết

1. Giới thiệu

- Đây là kỹ năng quan trọng dành cho người học lập trình, đặc biệt là khâu thiết kế giao diện cho một chương trình.
- Những mục tiêu cần đạt được đối với giao diện của một ứng dụng là
 - + Giao diện dễ dùng.
 - + Trình bày rõ ràng.
 - + Tính thẩm mỹ cao.
- Phần gỡ rối chương trình cần thiết cho người phát triển ứng dụng, tránh được những lỗi thường gặp, bảo mật ứng dụng. Những kỹ năng cần có gồm:
 - + Bẫy lỗi: dự đoán trước những chỗ có lỗi để bắt.
 - + Thông báo lỗi: báo cho người dùng biết về lỗi vừa xảy ra.
 - + Sửa lỗi: người lập trình sẽ thử chạy chương trình từng bước để tìm ra chỗ có lỗi và khắc phục.

2. Những kiến thức cơ bản

- Kỹ thuật cắt / dán: kỹ thuật này cơ bản nhưng rất hiệu quả khi ta thực hiện trên nhiều đối tượng giao diện.
- Canh đều tự động bằng toolbar: dùng thanh công cụ của Visual Studio để canh chỉnh vị trí cho bằng nhau.
- Cú pháp try catch: cú pháp bẫy lỗi cơ bản. Code thực thi sẽ được điền vào thân của hàm try catch, còn phần bắt lỗi, thông báo sẽ được lập trình trong thân phần catch.
- MessageBox: Hộp thoại thông báo, dùng để hiển thị những thông báo những thông tin quan trọng cho người dùng.

B. Bài tập cơ bản

Bài 2.1: Tạo form có giao diện như sau:

Yêu cầu:

- Khi nhấn Enter trên TextBox nhập số A thì chuyển con trỏ qua TextBox nhập số B, khi nhấn Enter trên TextBox nhập số B thì chuyển con trỏ qua Button Cộng, TextBox Kết quả chỉ cho phép đọc.
- Trước khi thực hiện các phép toán cần đảm bảo dữ liệu số A và Số B đều đã được nhập.
- Dữ liệu nhập chỉ chấp nhận giá trị các ký tự số.
- Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.
- Nhấn Button Thoát kết thúc chương trình.

Bài 2.2: Tạo form có giao diện như sau:

Yêu cầu:

- Khi nhấn Enter thì kích hoạt sự kiện click nút **Tính**.
- Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.
- Thực hiện các yêu cầu còn lại theo Bài 2.1.

Bài 2.3: Tạo form có giao diện như sau:

Yêu cầu:

- Khi nhấn Enter thì kích hoạt sự kiện click nút **Tính**.
- Khi nhấn Esc thì kích hoạt sự kiện click nút **Thoát**.
- Tính tiền điện theo bảng định mức như sau:

Mức	Số KW tiêu thụ	Đơn giá(VNĐ)
Khách hàng Hộ gia đình		
1	Cho kWh từ 0-50	1.484
2	Cho kWh từ 51 - 100	1.533
3	Cho kWh từ 101 -200	1.786
4	Cho kWh từ 201 -300	2.242
5	Cho kWh từ 301 -400	2.503
6	Cho kWh từ 401 trở lên	2.587
Khách hàng Doanh nghiệp		
1	Cho kWh từ 0-50	1.454
2	Cho kWh từ 51 - 100	1.502
3	Cho kWh từ 101 -200	1.750
4	Cho kWh từ 201 -300	2.197
5	Cho kWh từ 301 -400	2.453

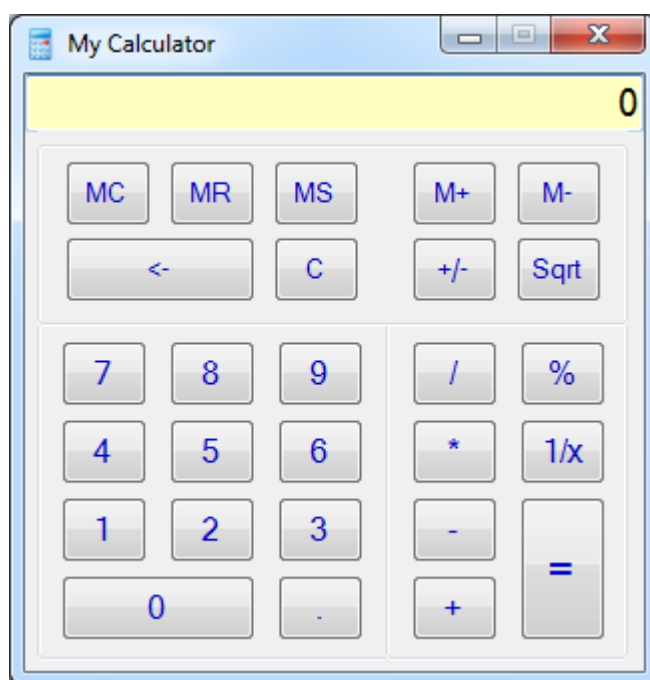
6	Cho kWh từ 401 trở lên	2.535
---	------------------------	-------

d. Thuế VAT =10%.

e. Ràng buộc nhập dữ liệu

- Chỉ số cũ, chỉ số mới chỉ chấp nhận các ký tự số.
- Chỉ số cũ, chỉ số mới, Số KW tiêu thụ ≥ 0 .
- Mặc định chọn khách hàng là hộ gia đình.
- Khi form load: Con trỏ nhập liệu ở TextBox Họ tên KH, Khi nhấn Enter tại TextBox KH con trỏ chuyển sang TextBox Chỉ số cũ, Khi nhấn Enter tại TextBox Chỉ số cũ con trỏ chuyển sang TextBox chỉ số mới, Khi nhấn Enter tại TextBox chỉ số mới chương trình tự tính số KW tiêu thụ hoặc khi Click chuột vào TextBox số KW tiêu thụ thì chương trình tự tính số KW tiêu thụ.
- TextBox số KW tiêu thụ, Thành tiền, thuế VAT, Tổng cộng chỉ cho phép đọc.

Bài 2.4: Xây dựng ứng dụng MyCalculator có giao diện như sau:



Yêu cầu:

- a. Thiết kế giao diện ứng dụng MyCalculator có giao diện như trên.
- b. Tham khảo các chứng năng của ứng dụng Calculator trong Windows để thực hiện.

Hướng dẫn:

- a. Thiết kế giao diện ứng dụng MyCalculator
 - Tạo mới một project loại Windows Application, đặt tên là MyCalculator.

- Thiết lập các thuộc tính cho Form như sau:
 - + Text: My Calculator.
 - + FormBorderStyle: FixedSingle.
 - + MaximizeBox: False.
 - + Locked: True.
- TextBox nhập số liệu và hiển thị kết quả
 - + Font: Tùy ý.
 - + FontSize: 16
 - + ReadOnly: True
 - + BackColor: Tùy ý.
 - + TextAlign: Right.
- Sử dụng điều khiển GroupBox để gom nhóm các điều khiển.
- Các Button: Thực hiện thiết lập đầy đủ các thuộc tính cho một Button sau đó thực hiện Copy và Paste. Sử dụng thanh công cụ Layout để canh, chỉnh các Button cho cân đối, phù hợp về khoảng cách, kích thước, ...

b. Phân tích xác định chức năng của các Button.

- MyCalculator là chương trình máy tính đơn giản nên chủ yếu thực hiện các phép toán 2 ngôi cơ bản. Có thể phân biệt nút bấm thuộc các nhóm:
 - + Nhóm 1: Button các chữ số 0, 1, ..., 9, ký hiệu dấu chấm thập phân, Button (+/-) đảo dấu, Button xóa trái (<-).
 - + Nhóm 2: Button phép tính +, -, x, /.
 - + Nhóm 3: Button phép tính một ngôi 1/x, %, Sqrt.
 - + Nhóm 4: Button “=” (*Bằng*) thực hiện phép tính và hiển thị kết quả.
 - + Nhóm 5: Button C thực hiện khởi tạo một phép toán mới, MC (xóa bộ nhớ máy tính), MR (đọc dữ liệu từ bộ nhớ máy tính đưa ra TextBox), MS (lưu dữ liệu trên TextBox vào bộ nhớ máy tính), M+ (Ghi thêm dữ liệu trên TextBox vào bộ nhớ máy tính), M- (Giảm dữ liệu bộ nhớ máy tính tương ứng với giá trị trên TextBox),

c. Viết Code cho Form và các Button như sau:

- + Khai báo biến toàn cục như sau:

```
string Memory, StrNumber1, StrNumber2, Clipboard;
```

```
char Operator;
```

- + Sự kiện FormLoad

```
private void Frm_MyCalculator_Load(object sender, EventArgs e)
{
    Memory = "0";
    StrNumber1="0";
    StrNumber2 = "0";
    Operator = ' ';
    Txt_Screen.Text = StrNumber1;
}
```

- + Sự kiện cho các Button phép tính +, -, x, /.

```
private void BtnOperator_Click(object sender, EventArgs e)
{
    Operator = char.Parse((sender as Button).Text);
}
```

- + Sự kiện cho các Button các chữ số 0, 1, ..., 9, ký hiệu dấu chấm thập phân, Button (+/-) đảo dấu, Button xóa trái (<-).

```
private void Btn_Number_Click(object sender, EventArgs e)
{
    string StrButton = (sender as Button).Text;

    if (Operator == ' ') //Xử lý Number1
    {
        if (StrButton==".")
        {
            if (StrNumber1.IndexOf(".") == -1)
                StrNumber1 += StrButton;
        }
        else if (StrButton=="+/-")
        {
            StrNumber1 += (double.Parse(StrNumber1) * -1).ToString();
        }
        else if (StrButton == "<-")
        {
            StrNumber1=StrNumber1.Substring(0, StrNumber1.Length - 1);
            if (StrNumber1 == "")
                StrNumber1 = "0";
        }
        else //Xử lý các Button from 0-9
        {
            if (StrNumber1 == "0")
                StrNumber1 = "";
            StrNumber1 += StrButton;
        }
        Txt_Screen.Text = StrNumber1;
    }
    else //Xử lý Number2
```

```

{
    if (StrButton == ".")
    {
        if (StrNumber2.IndexOf(".") == -1)
            StrNumber2 += StrButton;
    }
    else if (StrButton == "+/-")
    {
        StrNumber2 += (double.Parse(StrNumber2) * -1).ToString();
    }
    else if (StrButton == "<-")
    {
        StrNumber2=StrNumber2.Substring(0, StrNumber1.Length - 1);
        if (StrNumber2 == "")
            StrNumber2 = "0";
    }
    else //Xử lý các Button from 0-9
    {
        if (StrNumber2 == "0")
            StrNumber2 = "";
        StrNumber2 += StrButton;
    }

    Txt_Screen.Text = StrNumber2;
}

```

+ Sự kiện cho Button “=” (Bằng)

```

private void BtnBang_Click(object sender, EventArgs e)
{
    double Number1, Number2, Result=0;
    Number1=double.Parse(StrNumber1);
    Number2 = double.Parse(StrNumber2);

    if (Operator == '+')
        Result = Number1 + Number2;
    else if (Operator == '-')
        Result = Number1 - Number2;
    else if (Operator == '*')
        Result = Number1 * Number2;
    else if (Operator == '/')
        Result = Number1 / Number2;
    Txt_Screen.Text = Result.ToString();
}

```

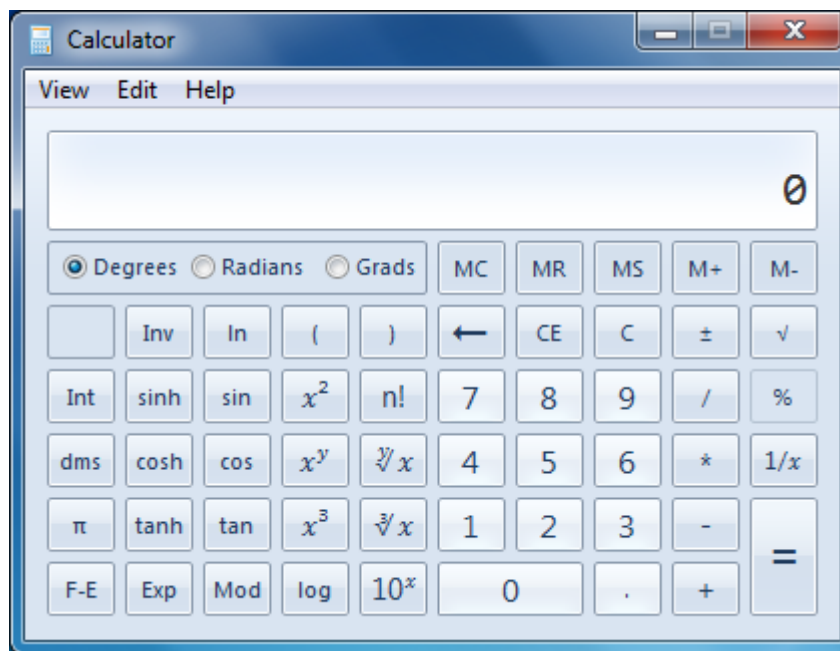
- d. Các sự kiện Button Click nhóm 3 và nhóm 5 sinh viên tự thực hiện và giảng viên hướng dẫn trên lớp.
- e. Thiết lập bổ sung một số thuộc tính cho Form như sau:
 - Thiết lập thuộc tính TopMost bằng True để trong thời gian thực thi, cửa sổ chương trình luôn luôn nổi trên các cửa sổ khác.
- f. Cài đặt thêm sự kiện Activated và Deactive cho Form như sau:
 - Hai sự kiện này được kích hoạt tương ứng khi người dùng chuyển focus từ vào ứng dụng (Activated) hay ra khỏi ứng dụng (Deactive).

```
private void Frm_MyCalculator_Activated(object sender, EventArgs e)
{
    this.Opacity=1;
}

private void Frm_MyCalculator_Deactivate(object sender, EventArgs e)
{
    this.Opacity = 0.6;
}
```

C. Bài tập nâng cao

Bài 2.5 Thiết kế ứng dụng tương đương với máy tính khoa học (Scientific Calculator) của ứng dụng Calculator trong Windows.



Yêu cầu:

- Thiết kế giao diện ứng dụng My Calculator có giao diện như trên.
- Tham khảo cách tính trong đối tượng Math.
- Thẻ hiện giao diện có nút x^2 .

--- HẾT ---

Chương 3: MENUSTRIP, CONTEXT MENUSTRIP, TOOLSTRIP, STATUSSTRIP VÀ COMMONDIALOG

A. Tóm tắt lý thuyết

1. Giới thiệu

- Trong lĩnh vực thiết kế giao diện thì thanh tiêu đề, trạng thái là gần như bắt buộc với mọi người học lập trình.

- Nhưng đối tượng này giúp người dùng dễ dàng tương tác với phần mềm bằng việc thể hiện rõ ràng những chức năng mà phần mềm có thể xử lý.

- MenuStrip, đối tượng này giống như danh sách hàng hóa mà một công ty bày bán. Những đối tượng khác tương tự như tờ mô tả thông tin chi tiết hay trạng thái của sản phẩm đang bày bán.

2. Những kiến thức cơ bản

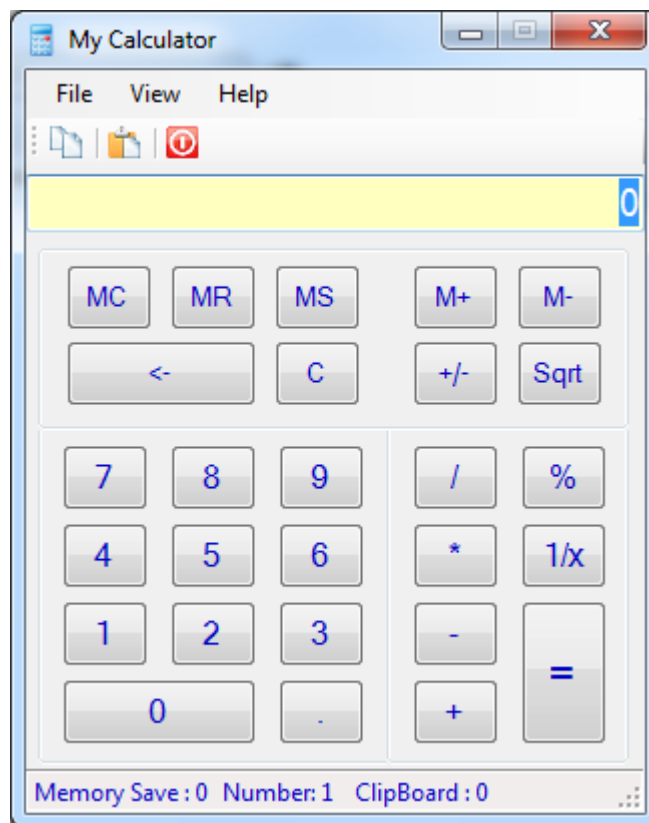
- Thanh tiêu đề (MenuStrip): đây là đối tượng vô cùng cơ bản, thường xuất hiện trong mọi phần mềm hiện hành. Chức năng chính là thể hiện những nhiệm vụ mà chương trình có thể thực hiện.

- Thanh trạng thái (StatusStrip): dùng để thông báo trạng thái của phần mềm.

- Thanh công cụ (ToolStrip): chứa những nút, lưu giữ những công việc hay làm, được thể hiện bằng hình ảnh thay vì chữ viết như thanh MenuStrip.

B. Bài tập cơ bản

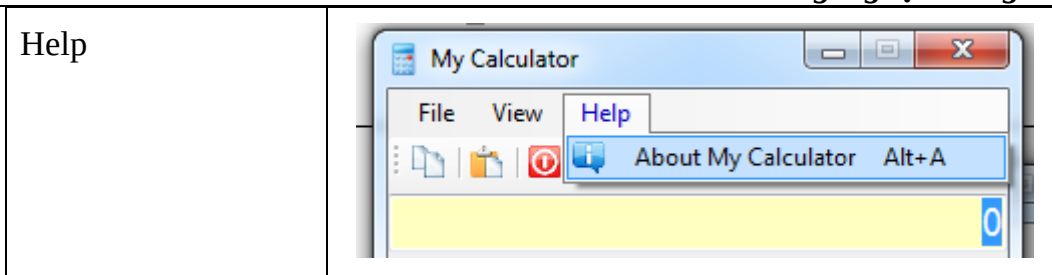
Bài 3.1: Bổ sung thêm menu, toolbar cho ứng dụng MyCalculator ở bài Bài 2.4 như sau:



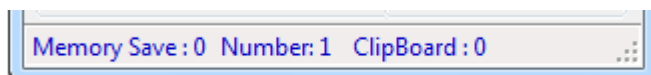
Yêu cầu:

- a. Thiết kế bổ sung giao diện ứng dụng MyCalculator gồm: Menu, Toolbar, Statusbar như trên.
 - Menu gồm:

File	
View	

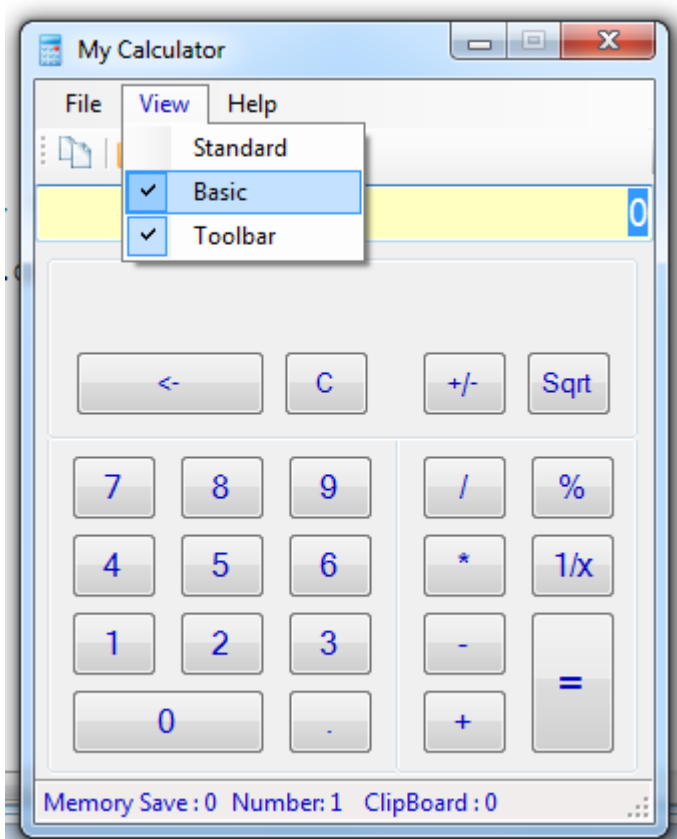


- Toolbar gồm: Copy, Separator, Paste, Separator, Exit.
- Statusbar gồm: Label trạng thái nhớ của máy tính, Trạng thái các số đang nhập, trạng thái Clipboard của chương trình.

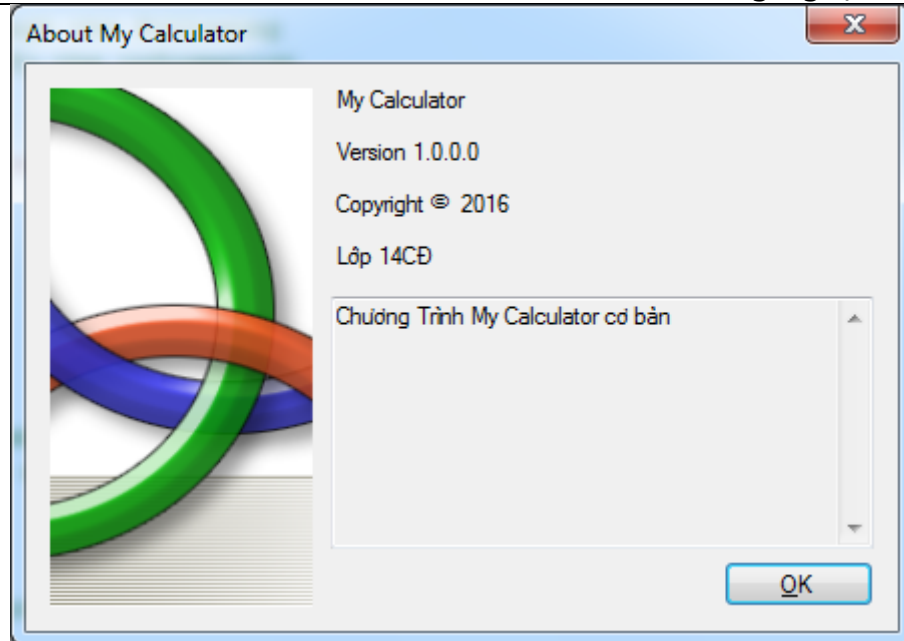


b. Thực hiện xử lý các chức năng mới thêm vào gồm:

- File -> Exit
- View -> Standard: Chế độ hiển thị này hiển thị đầy đủ các chức năng của chương trình. Chức năng này được chọn mặc định.
- View -> Basic: Chế độ hiển thị này hiển thị các chức năng cơ bản nhất của chương trình.
- **Lưu ý:** Tại một thời điểm chỉ hiển thị một trong 2 chế độ Standard hoặc chế độ Basic.

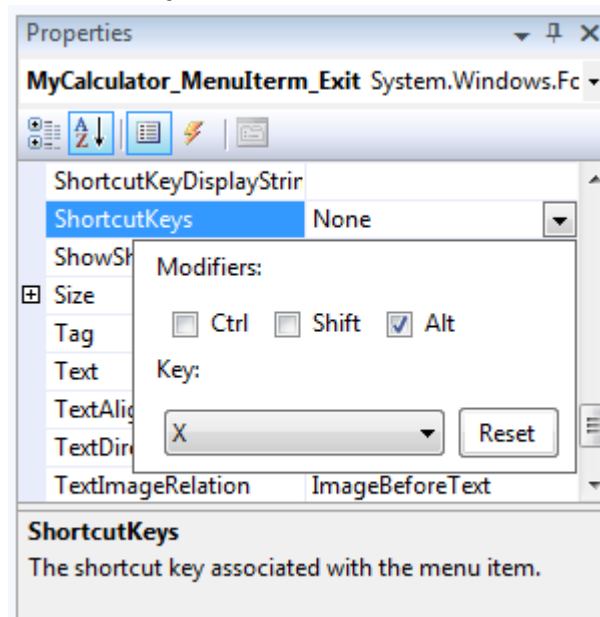


- View -> Toolbar: Cho phép hiển thị Toolbar hoặc không hiển thị.
- Help -> About My Calculator: Form Giới thiệu chương trình như sau:



Hướng dẫn:

- a. Thiết kế menu cho ứng dụng MyCalculator
 - Chọn menutrip trên ToolBox đưa vào Form MyCalculator và thiết lập các thuộc tính như sau.
 - + Name: MyCalculator_Menu
 - + Text: My Calculator Menu
 - Tạo các ToolStripMenuItem gồm:
 - + File:
 - Name: MyCalculator_Menu_Item_File
 - Text: File.
 - + Exit:
 - Name: MyCalculator_MenuItem_Exit
 - Images: Chọn icon như trong hình trên.
 - Text: Exit.
 - ShortcutKeys: Alt+X.



- + View:
 - Name: MyCalculator_Menu_Item_View
 - Text: View
- + Standard:
 - Name: MyCalculator_Menu_Item_Standard
 - Text: Standard.
 - ShortcutKeys: Alt+S.
- + Basic:
 - Name: MyCalculator_Menu_Item_Basic
 - Text: Basic.
 - ShortcutKeys: Alt+B.
- + Toolbar:
 - Name: MyCalculator_Menu_Item_Toolbar.
 - Text: Toolbar.
 - ShortcutKeys: Alt+T.
- + Help:
 - Name: MyCalculator_Menu_Item_Help
 - Text: Help
- + About My Calculator:
 - Name: MyCalculator_Menu_Item_About
 - Images: Chọn icon như trong hình trên.
 - Text: About My Calculator.
 - ShortcutKeys: Alt+A.
- Chọn ToolStrip trên ToolBox đưa vào Form MyCalculator và thiết lập các thuộc tính như sau.
 - + Name: MyCalculator_ToolBar
 - + Text: ToolBar
- Tạo ToolStripButton gồm:
 - + Copy: add Button
 - Name: ToolBar_Copy
 - Images: Chọn icon như trong hình trên.
 - Text: Copy.
 - ToolTipText: Copy.
 - + Separator: add Separator
 - + Paste:
 - Name: ToolBar_Paste
 - Images: Chọn icon như trong hình trên.
 - Text: Paste.
 - ToolTipText: Paste.
 - + Separator: add Separator
 - + Exit:
 - Name: ToolBar_Exit
 - Images: Chọn icon như trong hình trên.
 - Text: Exit.
 - ToolTipText: Exit.

- Chọn StatusStrip trên ToolBox đưa vào Form MyCalculator và thiết lập các thuộc tính như sau.
 - + Name: MyCalculator_StatusBar
 - + Text: Status My Calculator
 - + Dock: Bottom
- Tạo các StatusStrip gồm:
 - + Memory: add Label
 - Name: Status_Memory
 - Text: Memory
 - + Number: add Label
 - Name: Status_Number
 - Text: Number
 - + ClipBoard: add Label
 - Name: Status_ClipBoard
 - Text: ClipBoard

b. Tạo form About My Calculator

- Chọn project MyCalculator->Nhấn chuột phải ->Add -> new Item -> Chọn Windows Form -> Template ->AboutBox -> đặt tên Frm_About.cs
- Thiết lập lại các thông tin trong file AssemblyInfo.cs như sau:

```
// associated with an assembly.
[assembly: AssemblyTitle("My Calculator")]
[assembly: AssemblyDescription("Chương Trình My Calculator cơ bản")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("Lớp 14CB")]
[assembly: AssemblyProduct("My Calculator")]
[assembly: AssemblyCopyright("Copyright © 2016")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0")]
[assembly: AssemblyFileVersion("1.0")]
```

c. Viết Code xử lý các chức năng của Menu

- Exit:
 - + Double Click vào menu Exit nhập câu lệnh sau vào.

```
private void MyCalculator_MenuItem_Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

- View->Standard: Double Click vào menu Standard nhập câu lệnh sau vào

```
private void MyCalculator_Menu_Item_Standard_Click(object sender, EventArgs e)
{
    if (MyCalculator_Menu_Item_Standard.CheckState == CheckState.Checked)
    {
        MyCalculator_Menu_Item_Standard.Checked = false;
        MyCalculator_Menu_Item_Basic.Checked = true;
        Btn_MC.Visible = false;
        Btn_MS.Visible = false;
        Btn_MR.Visible = false;
        Btn_MPlus.Visible = false;
        Btn_MMinus.Visible = false;
    }
    else
    {
        MyCalculator_Menu_Item_Standard.Checked = true;
        MyCalculator_Menu_Item_Basic.Checked = false;
        Btn_MC.Visible = true;
        Btn_MS.Visible = true;
        Btn_MR.Visible = true;
        Btn_MPlus.Visible = true;
        Btn_MMinus.Visible = true;
    }
}
```

- View->Basic: Double Click vào menu Basic nhập câu lệnh sau vào

```
private void MyCalculator_Menu_Item_Basic_Click(object sender, EventArgs e)
{
    if (MyCalculator_Menu_Item_Basic.CheckState == CheckState.Checked)
    {
        MyCalculator_Menu_Item_Basic.Checked = false;
        MyCalculator_Menu_Item_Standard.Checked = true;
        Btn_MC.Visible = true;
        Btn_MS.Visible = true;
        Btn_MR.Visible = true;
        Btn_MPlus.Visible = true;
        Btn_MMinus.Visible = true;
    }
    else
    {
        MyCalculator_Menu_Item_Standard.Checked = false;
        MyCalculator_Menu_Item_Basic.Checked = true;
        Btn_MC.Visible = false;
        Btn_MS.Visible = false;
        Btn_MR.Visible = false;
        Btn_MPlus.Visible = false;
        Btn_MMinus.Visible = false;
    }
}
```

- View->ToolBar: Double Click vào menu ToolBar nhập câu lệnh sau vào

```
private void MyCalculator_Menu_Item_Toolbar_Click(object sender, EventArgs e)
{
    if (MyCalculator_Menu_Item_Toolbar.CheckState == CheckState.Checked)
    {
        MyCalculator_Menu_Item_Toolbar.Checked = false;
        MyCalculator_ToolBar.Visible = false;
    }
    else
    {
        MyCalculator_Menu_Item_Toolbar.Checked = true;
        MyCalculator_ToolBar.Visible = true;
    }
}
```

- Help-> About My Calculator: Double Click vào menu About My Calculator nhập câu lệnh sau vào

```
private void MyCalculator_Menu_Item_About_Click(object sender, EventArgs e)
{
    Form frmAbout = new Frm_About();
    frmAbout.Show();
}
```

- d. Viết Code xử lý các chức năng của ToolBar

- Copy: Double Click vào Button Copy trên ToolBar nhập câu lệnh sau vào

```
private void ToolBarCopy_Click(object sender, EventArgs e)
{
    Clipboard = Txt_Screen.Text;
    Status_ClipBoard.Text = "Clipboard : " + Clipboard;
}
```

- Paste: Double Click vào Button Paste trên ToolBar nhập câu lệnh sau vào

```
private void ToolBarPaste_Click(object sender, EventArgs e)
{
    Txt_Screen.Text = Clipboard;
}
```

- Exit: Double Click vào Button Exit trên ToolBar nhập câu lệnh sau vào

```
private void ToolBarButton_Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

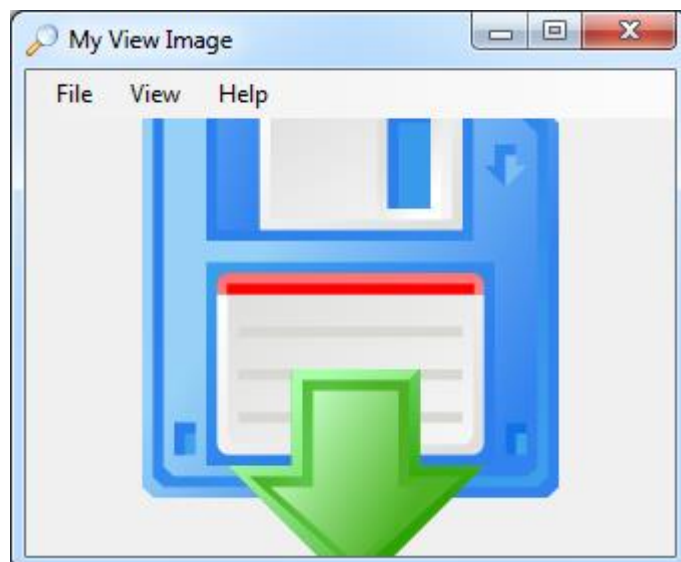
- e. Viết Code xử lý hiển thị các thông tin trên StatusBar

- Label Status_Memory:

- + Khai báo biến có phạm vi toàn cục trên forms My Calculator.
 - `string Memory;`
- + Bổ sung lệnh sau vào sự kiện Form Load
 - `Memory = "0";`
 - `Status_Memory.Text = "Memory Save : " + Memory;`

- + Bổ sung lệnh sau vào sự kiện Btn_MC_Click
 - Status_Memory.Text = "Memory Save : " + Memory;
- + Bổ sung lệnh sau vào sự kiện Btn_MS_Click
 - Status_Memory.Text = "Memory Save : " + Memory;
- + Bổ sung lệnh sau vào sự kiện Btn_MPlus_Click
 - Status_Memory.Text = "Memory Save : " + Memory;
- + Bổ sung lệnh sau vào sự kiện Btn_MMinus_Click
 - Status_Memory.Text = "Memory Save : " + Memory;
- Label Status_Number:
 - + Bổ sung lệnh sau vào sự kiện Form Load
 - Status_Number.Text = "Number: 1 ";
 - + Bổ sung lệnh sau vào sự kiện Btn_Number_Click
 - Status_Number.Text = "Number 1"; //Bổ sung ngay sau lệnh Txt_Screen.Text = StrNumber1;
 - Status_Number.Text = "Number 2"; //Bổ sung ngay sau lệnh Txt_Screen.Text = StrNumber2;
- Label Status_ClipBoard:
 - + Khai báo biến có phạm vi toàn cục trên forms My Calculator.
 - string Clipboard;
 - + Bổ sung lệnh sau vào sự kiện Form Load
 - Clipboard = "0";
 - Status_ClipBoard.Text = " Clipboard: " + Clipboard;
 - + Bổ sung lệnh sau vào sự kiện ToolBarCopy_Click
 - Status_ClipBoard.Text = "Clipboard : " + Clipboard; //Bổ sung ngay sau lệnh Clipboard = Txt_Screen.Text;

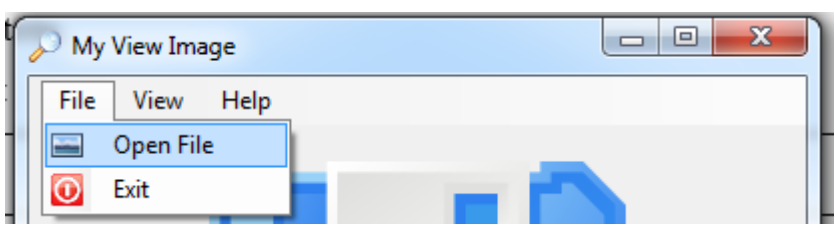
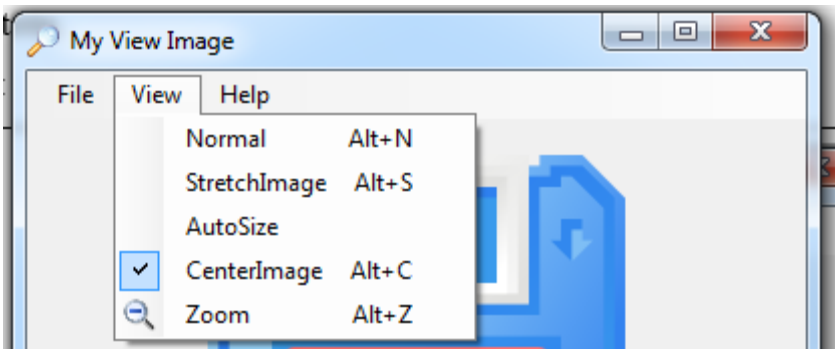
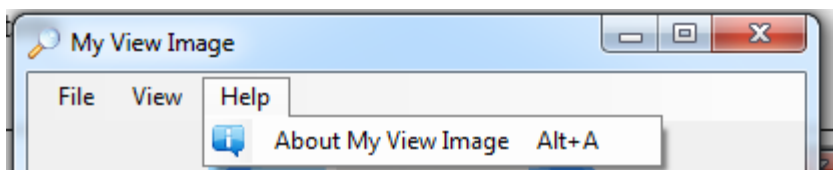
Bài 3.2: Xây dựng ứng dụng xem hình ảnh MyViewImage như sau:



Yêu cầu:

- a. Thiết kế giao diện ứng dụng MyViewImage gồm: Menu, Toolbar, Statusbar như trên.

- Menu gồm:

File	
View	
Help	

- b. Thực hiện xử lý các chức năng gồm.

- File -> Open File: Mở hộp thoại Open file chọn hình ảnh.
- File -> Exit: Đóng ứng dụng.
- View -> Normal: Chế độ hiển thị của thuộc tínhSizeMode trong điều khiển PictureBox. Chức năng này được chọn mặc định.
- View -> StretchImage: SizeMode=PictureBoxSizeMode.StretchImage.
- View -> AutoSize: SizeMode=PictureBoxSizeMode.AutoSize.
- View -> CenterImage: SizeMode=PictureBoxSizeMode.CenterImage.
- View -> Zoom: SizeMode=PictureBoxSizeMode.Zoom.
- Help -> About My View : Form Giới thiệu chương trình như sau:

Hướng dẫn:

- Viết Code cho các chức năng của ứng dụng.
- + File -> Open File:


```
private void OpenFileToolStripMenuItem_Click(object sender, EventArgs e)
{
    string strPath = "";
    openFileDialog_MyViewImage.Title = "Select a Picture file";
    openFileDialog_MyViewImage.CheckFileExists = true;
    openFileDialog_MyViewImage.CheckPathExists = true;
    openFileDialog_MyViewImage.Filter = "JPEG files (*.JPEG)|*.JPEG|PNG files (*.PNG)|*.PNG|BMP files (*.BMP)|*.BMP|GIF files (*.GIF)|*.GIF|All files (*.*)|*.*";

    if (openFileDialog_MyViewImage.ShowDialog() == DialogResult.OK)
    {
        strPath = openFileDialog_MyViewImage.FileName;
        PtB_Image.Image = Image.FromFile(strPath);
    }
    else
    {
        MessageBox.Show("Bạn chưa chọn file Image", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

+ View -> Normal và View -> StretchImage:

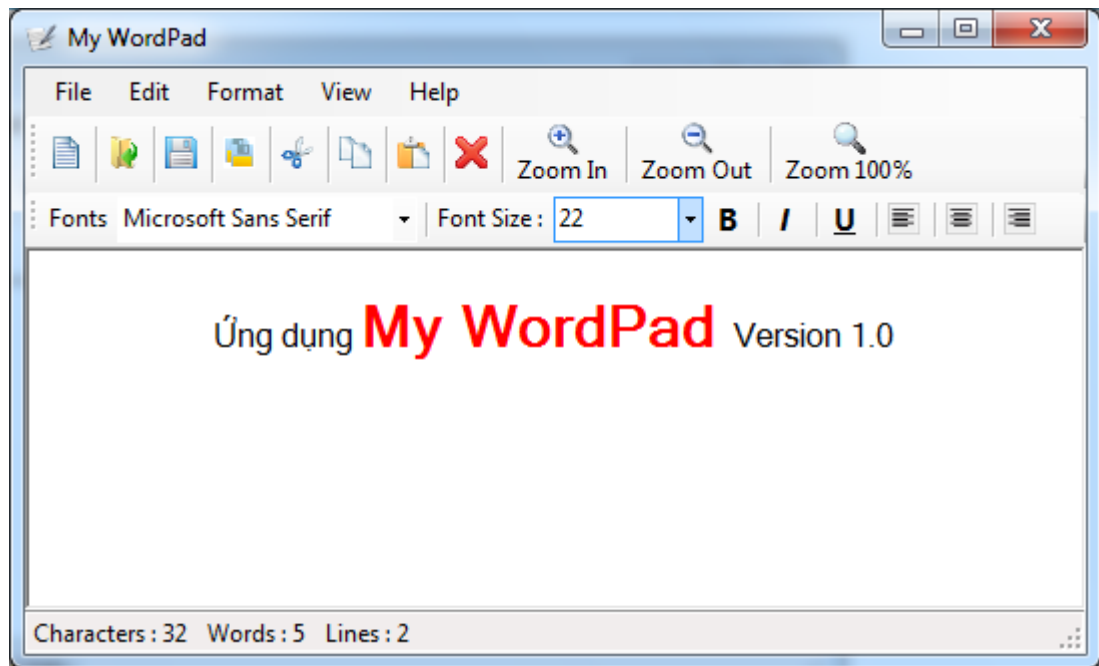
```
private void normalToolStripMenuItem_Click(object sender, EventArgs e)
{
    PtB_Image.SizeMode = PictureBoxSizeMode.Normal;
    normalToolStripMenuItem.Checked = true;
    stretchImageToolStripMenuItem.Checked = false;
    autoSizeToolStripMenuItem.Checked = false;
    centerImageToolStripMenuItem.Checked = false;
    zoomToolStripMenuItem.Checked = false;
}

private void stretchImageToolStripMenuItem_Click(object sender, EventArgs e)
{
    PtB_Image.SizeMode = PictureBoxSizeMode.StretchImage;
    normalToolStripMenuItem.Checked = false;
    stretchImageToolStripMenuItem.Checked = true;
    autoSizeToolStripMenuItem.Checked = false;
    centerImageToolStripMenuItem.Checked = false;
    zoomToolStripMenuItem.Checked = false;
}
```

+ View -> AutoSize:

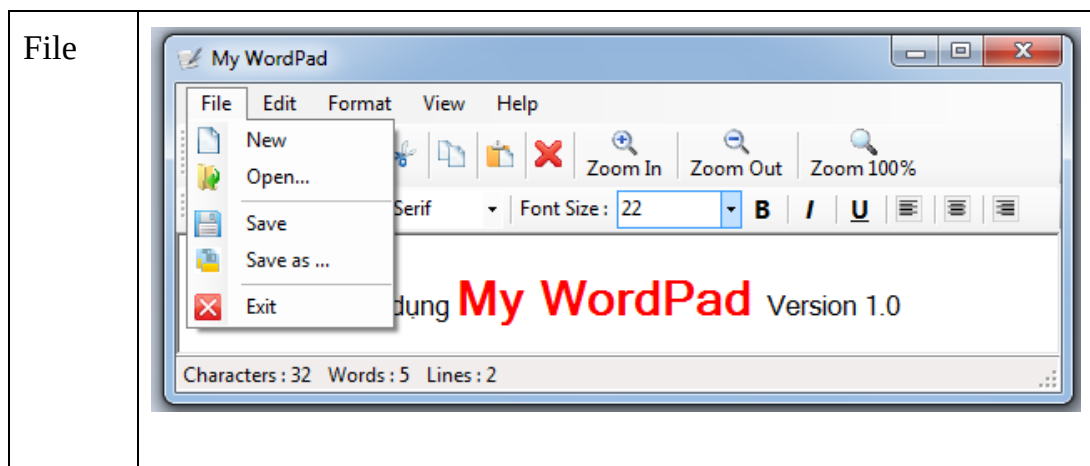
```
private void autoSizeToolStripMenuItem_Click(object sender, EventArgs e)
{
    PtB_Image.SizeMode = PictureBoxSizeMode.AutoSize;
    normalToolStripMenuItem.Checked = false;
    stretchImageToolStripMenuItem.Checked = false;
    autoSizeToolStripMenuItem.Checked = true;
    centerImageToolStripMenuItem.Checked = false;
    zoomToolStripMenuItem.Checked = false;
}
```

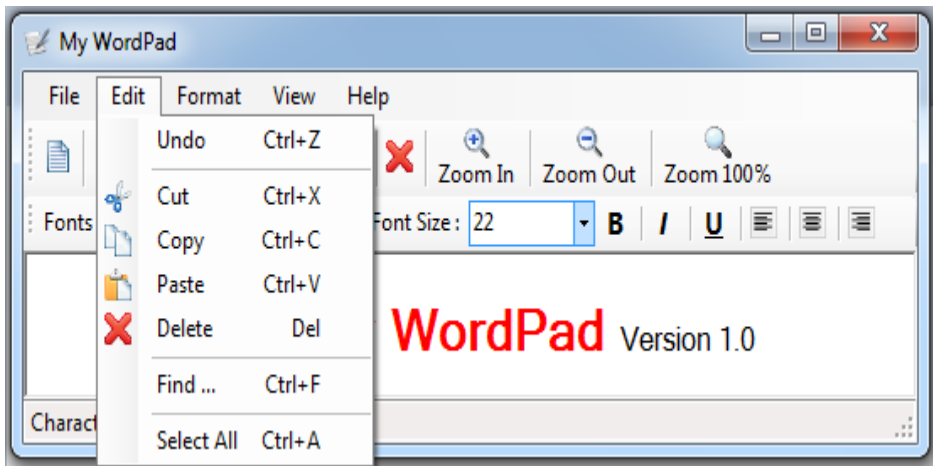
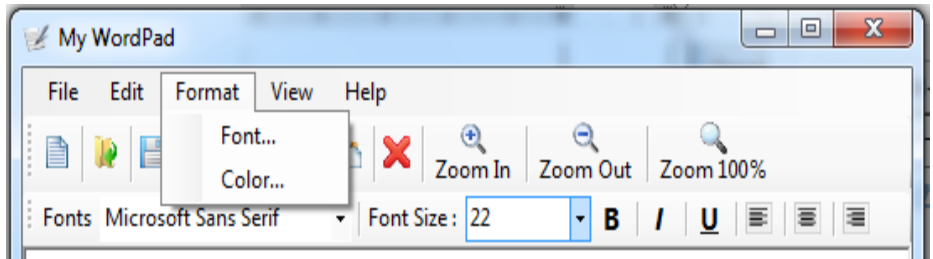
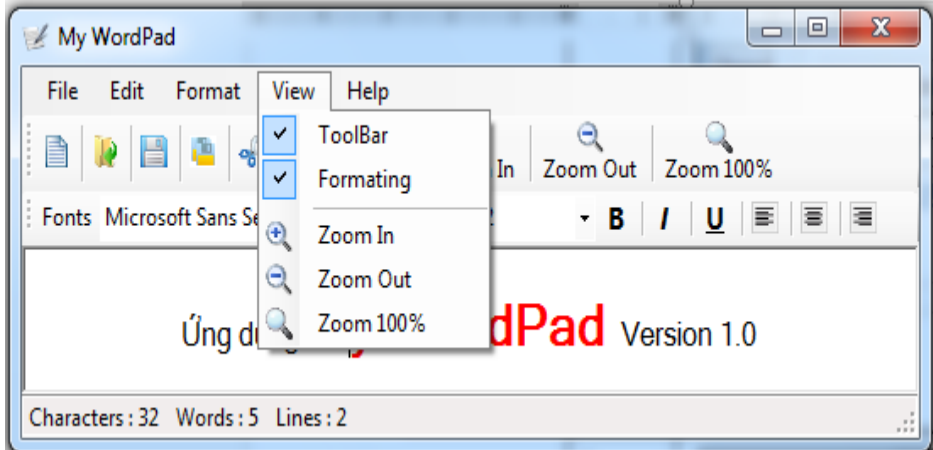
Bài 3.3: Xây dựng ứng dụng soạn thảo văn bản đơn giản như sau:



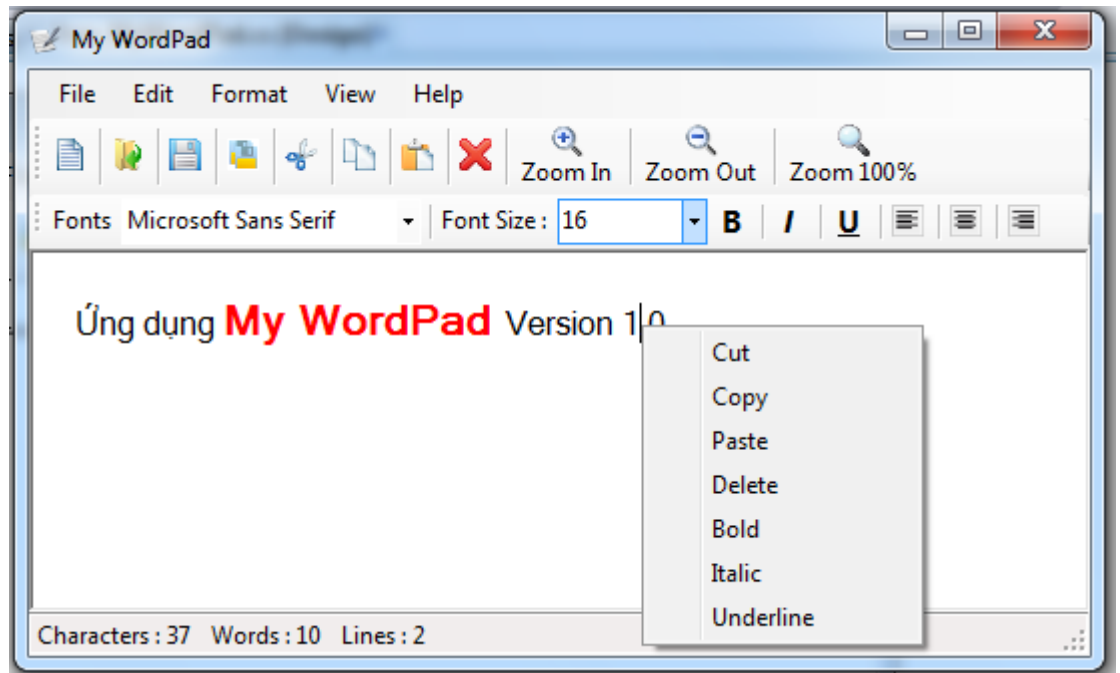
Yêu cầu:

- Ứng dụng xử lý trên tập tin *.rtf.
- Thiết kế giao diện ứng dụng MyViewImage gồm: Menu, Toolbar, Statusbar như trên.
 - Menu gồm:

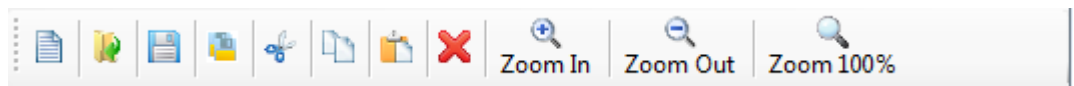


Edit	
Format	
View	
Help	About My WordPad

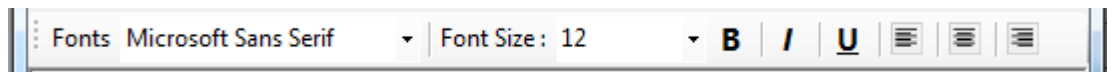
- ContextMenu:



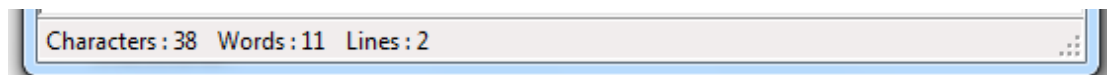
- ToolBar gồm:
 - + Standard ToolBar:



- + Formatting ToolBar:



- StatusBar:



c. Thực hiện xử lý các chức năng gồm.

- Menu -> File:
 - + New: Tạo mới một tài liệu.
 - + Open: Mở một tài liệu đã tồn tại.
 - + Save: Lưu tài liệu đã tồn tại.
 - + Save as: Lưu tài liệu chưa tồn tại hay vào một tập tin khác.
 - + Exit: Thoát khỏi ứng dụng.
- Menu -> Edit:
 - + Undo: Tạo mới một tài liệu.
 - + Cut: Cắt nội dung đã được chọn khối vào Clipboard.
 - + Copy: Chép nội dung đã được chọn khối vào Clipboard.

- + Paste: Dán nội dung trong Clipboard ra RichTextBox tại vị trí của con trỏ.
- + Delete: Xoá nội dung đã được chọn khỏi.
- + Find: Tìm một đoạn văn trong RichTextBox.
- + Select All: Chọn toàn bộ nội dung trong RichTextBox.
- Menu -> Format:
 - + Font: Hiện thị hộp thoại font cho người dùng chọn để định dạng văn bản đã chọn khỏi trước đó..
 - + Color: Hiện thị hộp thoại màu cho người dùng chọn để định dạng màu cho văn bản đã chọn khỏi trước đó.
- Menu -> Help:
 - + About My WordPad: Hiện thị form giới thiệu về My WordPad.
- ContextMenu: Cho RichTextBox gồm 4 chức năng Cut, Copy, Paste, Delete.
- Standard ToolBar:
 - + New: Tạo mới một tài liệu.
 - + Open: Mở một tài liệu đã tồn tại.
 - + Save: Lưu tài liệu đã tồn tại.
 - + Save as ...: Lưu tài liệu chưa tồn tại.
 - + Cut: Cắt nội dung đã được chọn khỏi vào Clipboard.
 - + Copy: Chép nội dung đã được chọn khỏi vào Clipboard.
 - + Paste: Dán nội dung trong Clipboard ra RichTextBox tại vị trí của con trỏ.
 - + Delete: Xoá nội dung đã được chọn khỏi.
 - + Zoom in: Phóng to nội dung trong RichTextBox mỗi lần nhấn tăng 15%.
 - + Zoom out: Thu nhỏ nội dung trong RichTextBox mỗi lần nhấn giảm 15%.
 - + Zoom 100%: Hiện thị nội dung trong RichTextBox ở chế độ mặc định.
- Formatting ToolBar:
 - + Fonts: Định dạng Font chữ trong RichTextBox được chọn.
 - + Font Size: Định dạng chữ size chữ trong RichTextBox được chọn.

- + **B**: Định dạng chữ in đậm trong RichTextBox được chọn..
- + **I**: Định dạng chữ in nghiêng trong RichTextBox được chọn..
- + **U**: Định dạng chữ có gạch chân trong RichTextBox được chọn.
- StatusBar:
 - + Characters: Hiển thị số ký tự trong RichTextBox.
 - + Words: Hiển thị số từ trong RichTextBox.
 - + Lines: Hiển thị số dòng trong RichTextBox.
- Help -> About My View : Form Giới thiệu chương trình như sau:

Hướng dẫn:

- Giảng viên hướng dẫn cụ thể trên lớp.

--- HẾT ---

Chương 4: LISTBOX VÀ COMBOBOX

A. Tóm tắt lý thuyết

1. Giới thiệu

- Việc đa dạng hóa sự chọn lựa của người dùng dưới dạng nhiều thể hiện khác nhau cũng rất quan trọng trong việc lập trình vì nó cho phép người dùng thoải mái hơn cũng như giao diện phần mềm sẽ chuyên nghiệp hơn.

- Những đối tượng này nằm sẵn trong thanh ToolBox của chương trình Visual Studio.

2. Những kiến thức cơ bản

- ListBox: đối tượng này cho phép người dùng chọn một hay vài chọn lựa khác nhau trong nhiều sự chọn lựa được hiển thị trong thân ListBox.

- ComboBox: Hộp thoại này cho phép người dùng chọn một trong nhiều sự chọn lựa đã có sẵn, dưới dạng xổ xuống.

B. Bài tập cơ bản

Bài 4.1 Viết chương trình cho sinh viên đăng ký môn học theo yêu cầu sau:

Yêu cầu:

1. Thiết kế form Đăng ký môn học như hình trên.
2. Nút thêm: Khi nhấn nút Thêm thì thực hiện thêm môn học mới vào ListBox Danh sách môn học. Kiểm tra môn học đã tồn tại trong danh sách thì thông báo môn học đã tồn tại và thực hiện chọn môn học đó trong danh sách môn học.
 - Hướng dẫn:

```

if (Txt_TenMH.TextLength == 0)
    MessageBox.Show("Tên môn học không được rỗng");
else if (Lst_DS_Monhoc.Items.IndexOf(Txt_TenMH.Text) > 0)
{
    MessageBox.Show("Tên môn học đã tồn tại trong danh sách");
    Lst_DS_Monhoc.SetSelected(Lst_DS_Monhoc.Items.IndexOf(Txt_TenMH.Text), true);
}
else
    Lst_DS_Monhoc.Items.Add(Txt_TenMH.Text);
                    
```
3. Nút Xóa: Khi nhấn nút Xóa thì thực hiện xóa môn học đang được chọn ở ListBox Danh sách môn học.
4.  Chuyển mỗi lần một môn học được chọn từ danh sách môn học qua ListBox danh sách môn học đăng ký.
 - Hướng dẫn:

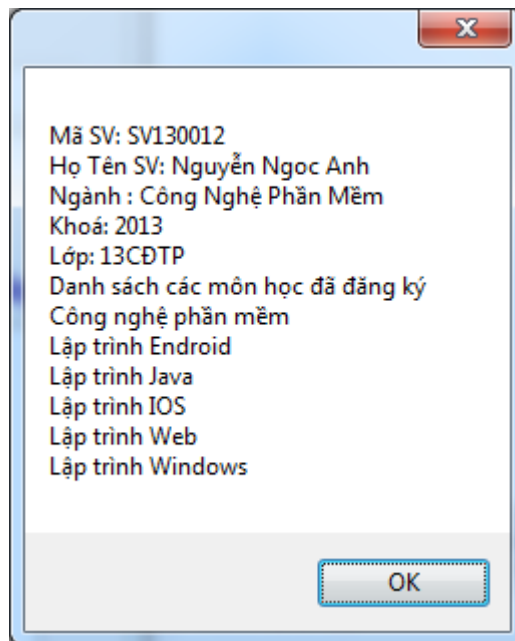
```

Lst_MonhocDangKy.Items.Add(Lst_DS_Monhoc.SelectedItem);
Lst_DS_Monhoc.Items.RemoveAt(Lst_DS_Monhoc.SelectedIndex);
                    
```
5.  Chuyển mỗi lần là tất cả các môn học được chọn từ danh sách môn học qua ListBox danh sách môn học đăng ký.
 - Hướng dẫn:

```

//Thêm các item vào ds môn học đăng ký
foreach (object itm in Lst_DS_Monhoc.SelectedItems)
{
    Lst_MonhocDangKy.Items.Add(itm);
}
//Xóa các items từ ds môn học
for (int i = Lst_DS_Monhoc.Items.Count-1; i >= 0; i--)
{
    if (Lst_DS_Monhoc.GetSelected(i) == true)
        Lst_DS_Monhoc.Items.RemoveAt(i);
}
                    
```
6.  Chuyển mỗi lần một môn học được chọn từ ListBox danh sách các môn học đăng ký qua ListBox danh sách môn học.
7.  Chuyển mỗi lần là tất cả các môn học được chọn từ ListBox danh sách các môn học đăng ký qua ListBox danh sách môn học.
8. Khi nhấn Nút lệnh đăng ký kết quả hiển thị trong MessageBox như hình trên.
9. Khi nhấn Nút lệnh thoát thì dừng chương trình.

Kết quả:



Bài 4.2 Viết ứng dụng chuyển đổi ngoại tệ theo yêu cầu sau:

1. Thiết kế giao diện theo như hình trên.

- Danh sách các ngoại tệ theo trên form.
 - ComboBox chọn ngoại tệ: Danh sách ngoại tệ theo các giá trị có trên form.
 - Khi người dùng chọn ngoại tệ thì tỷ giá tự động xuất hiện trên textbox tỷ giá.
- Bảng tỷ giá như sau:

TỶ GIÁ	
USD	21,845
JPY	179.10
AUD	16,482
CAD	17,466
GBP	34,052
SGD	16,276
EUR	24,571

1. Nút lệnh Chuyển đổi thực hiện chuyển đổi tiền VND sang các đơn vị tiền tệ khác theo hình và bảng tỷ giá theo sau.
2. Nút Làm lại thực hiện xóa các dữ liệu trong các ô textbox và chuyển con trỏ về textbox nhập tiền VNĐ.
3. Khi người dùng nhập mới một ngoại tệ không có trong danh sách thì phải nhập tỷ giá và kết quả được hiển thị trong ô TextBox cuối cùng của form.
4. Nút lệnh Thoát: Thực hiện dừng chương trình.

Bài 4.3 Viết ứng dụng tính lãi suất tham khảo cho khoản vay tại ngân hàng theo những yêu cầu sau:

BẢNG ƯỚC TÍNH THAM KHẢO CHO KHOẢN VAY

Số tiền vay : VNĐ

Thời gian vay : tháng

Hình thức lãi suất : Lãi suất : %/năm

Số tiền trả mỗi tháng đối với lãi suất dư nợ ban đầu :

Số tiền trả mỗi tháng đối với dư nợ giảm dần

Tháng 1: 1,675,000;	Trả gốc : 1,250,000	Dư nợ : 30,000,000	Lãi : 425,000
Tháng 2: 1,657,292;	Trả gốc : 1,250,000	Dư nợ : 28,750,000	Lãi : 407,292
Tháng 3: 1,639,583;	Trả gốc : 1,250,000	Dư nợ : 27,500,000	Lãi : 389,583
Tháng 4: 1,621,875;	Trả gốc : 1,250,000	Dư nợ : 26,250,000	Lãi : 371,875
Tháng 5: 1,604,167;	Trả gốc : 1,250,000	Dư nợ : 25,000,000	Lãi : 354,167
Tháng 6: 1,586,458;	Trả gốc : 1,250,000	Dư nợ : 23,750,000	Lãi : 336,458
Tháng 7: 1,568,750;	Trả gốc : 1,250,000	Dư nợ : 22,500,000	Lãi : 318,750
Tháng 8: 1,551,042;	Trả gốc : 1,250,000	Dư nợ : 21,250,000	Lãi : 301,042
Tháng 9: 1,533,333;	Trả gốc : 1,250,000	Dư nợ : 20,000,000	Lãi : 283,333
Tháng 10: 1,515,625;	Trả gốc : 1,250,000	Dư nợ : 18,750,000	Lãi : 265,625
Tháng 11: 1,497,917;	Trả gốc : 1,250,000	Dư nợ : 17,500,000	Lãi : 247,917
Tháng 12: 1,480,208;	Trả gốc : 1,250,000	Dư nợ : 16,250,000	Lãi : 230,208

2. Thiết kế giao diện theo như hình trên.
 - *Thời gian vay* : cho người dùng chọn các giá trị 12, 18, 24, 30, 36, 48, 60 (đơn vị là tháng).
 - *Hình thức trả lãi suất* : Cố định, dư nợ giảm dần.
 - *Lãi suất* : Cố định là 12% năm; dư nợ giảm dần là 18% năm. Người dùng không nhập, khi người dùng chọn hình thức trả lãi suất thì chương trình hiển thị mức lãi suất tương ứng.
3. Nút Tính : Thực hiện việc tính số tiền phải trả hàng tháng theo từng hình thức trả lãi suất tương ứng do người dùng chọn.
 - Số tiền phải trả hàng tháng theo hình thức trả lãi suất cố định = $\frac{(((((\text{Lãi suất}/12) * \text{Số tiền vay}) * \text{Số tháng vay}) + \text{Số tiền vay}) / \text{Số tháng vay})}{\text{Số tháng vay}}$.
 - Số tiền phải trả hàng tháng theo hình thức trả lãi suất dư nợ giảm dần sinh viên tự xây dựng. GV sẽ hướng dẫn trong buổi thực hành.
4. Thát: Thoát khỏi chương trình.

--- HẾT ---

Chương 5: CÁC ĐIỀU KHIỂN NÂNG CAO

A. Tóm tắt lý thuyết

1. Giới thiệu

- Khi thực hiện một dự án phần mềm, ta sẽ phải đối mặt với những yêu cầu nâng cao của khách hàng. Do đó những điều khiển nâng cao này sẽ đáp ứng tốt hơn yêu cầu khắc khe của người dùng.

- Hơn thế nữa, giao diện của phần mềm sẽ càng trở nên chuyên nghiệp hơn khi có những thành phần như đồng hồ đo thời gian, hộp chọn ngày tháng, nhiều giao diện con trong một giao diện chính, hiện thị dữ liệu dưới dạng cây...

2. Những kiến thức cơ bản

- Thanh tiến trình (ProgressBar): thể hiện phần trăm công việc đã hoàn tất. Dữ liệu đã sao chép hay quá trình đã hiện thực được bao nhiêu.

- Đồng hồ (Timer): đối tượng hiển thị thời gian của hệ thống.

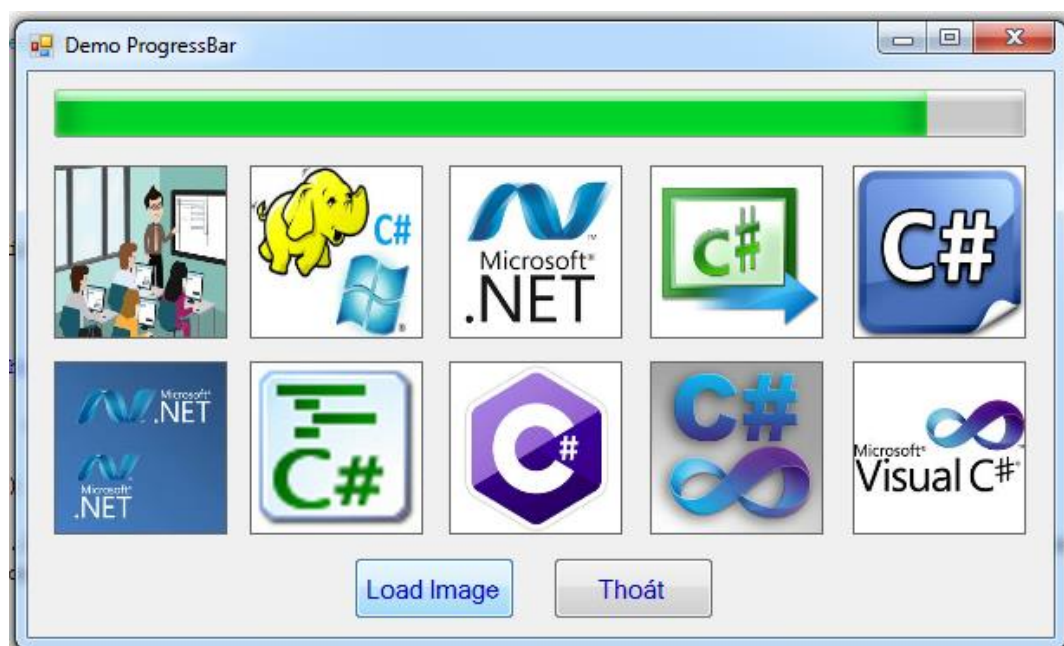
- Hộp chọn ngày (DateTimePicker): cho phép người dùng chọn thời gian từ giao diện, thường dùng để chọn ngày tháng năm sinh.

- Điều khiển dạng thẻ (TabControl): cho phép tạo ra nhiều khoảng trống khi lập trình bằng cách kết hợp nhiều giao diện vào một giao diện thông qua việc phân chia theo thẻ. Mỗi giao diện con sẽ nằm trong một thẻ.

- Hiển thị dạng cây (TreeView): đây là hiển thị cho phép người dùng theo dõi dữ liệu dưới dạng cây, như trong trường có nhiều khoa, mỗi khoa có nhiều giảng viên.

B. Bài tập cơ bản

Bài 5.1 Viết ứng dụng Demo ProgressBar có giao diện như sau:



Yêu cầu:

- Thiết kế giao diện ứng dụng Demo ProgressBar.
- Khi thực hiện Load Image quá trình hoàn thành việc Load Image cho từng khung hình được thể hiện trên ProgressBar.

Hướng dẫn:

- Giảng viên đã hướng dẫn cụ thể trên lớp.
- Viết code cho Button Load Image.

```
private void Btn_Load_Image_Click(object sender, EventArgs e)
{
    string strimagefile="";
    int i = 1;
    foreach (Control ctrl in this.Controls)
    {
        strimagefile = "image" + i.ToString() + ".jpg";

        if (ctrl is PictureBox)
        {
            ((PictureBox)ctrl).Image = Image.FromFile(
                Application.StartupPath.Substring(0, Application.StartupPath.Length - 10)
                + "\\images\\" + strimagefile);
            //progressBar_ImageLoad.Value = i * 10; hoặc
            //progressBar_ImageLoad.Increment(10); hoặc
            progressBar_ImageLoad.PerformStep();//Thuộc tính Step=10
            i += 1;
        }
        Thread.Sleep(500); //Khai báo thêm using System.Threading;
        Frm_DemoProgressBar.ActiveForm.Refresh();
    }
}
```

Bài 5.2 Viết ứng dụng Demo Timer có giao diện như sau:



Yêu cầu:

- Thiết kế giao diện ứng dụng Demo Timer.
- Khi thực hiện Load Image quá trình hoàn thành việc Load Image cho từng khung hình được thể hiện trên Timer.

Hướng dẫn:

- a. Viết code cho Button Bắt đầu.

```
private void Btn_BatDau_Click(object sender, EventArgs e)
{
    timer1.Start();
}
```

- b. Viết code cho Button dừng.

```
private void Btn_Dung_Click(object sender, EventArgs e)
{
    timer1.Stop();
}
```

- c. Viết code cho sự kiện Tick của Timer.

```
private void timer1_Tick(object sender, EventArgs e)
{
    switch (den)
    {
        case 1: Pct.Image =
            Image.FromFile(Application.StartupPath.Substring(0,
                Application.StartupPath.Length - 10) +
                "\\images\\denxanh.jpg");
            timer1.Interval = 3000;
            den += 1;
            break;

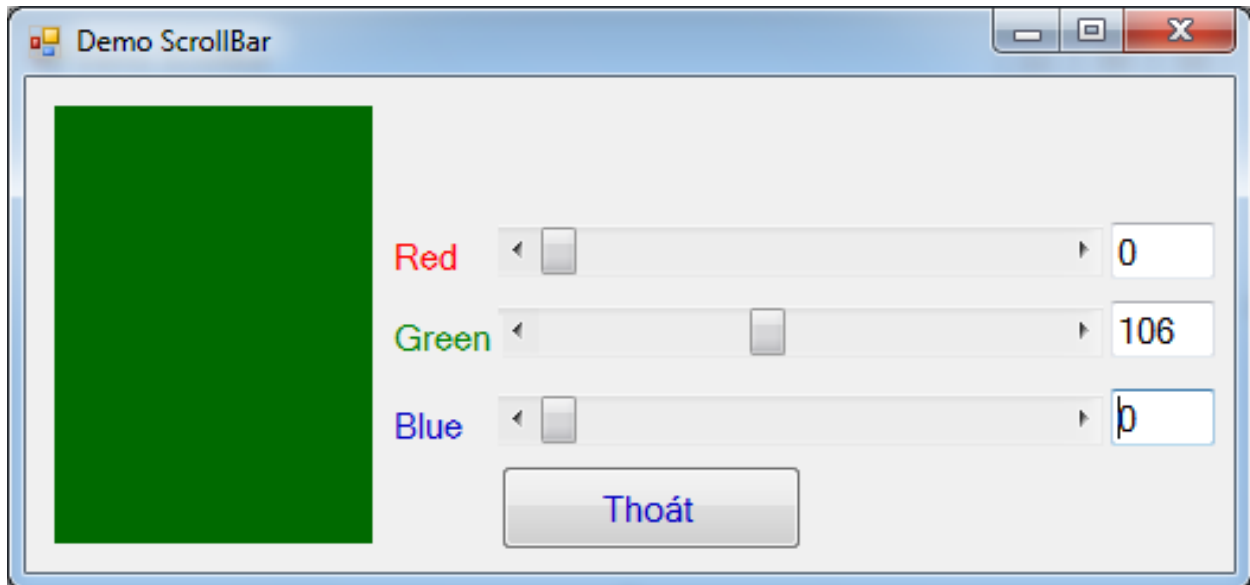
        case 2: Pct.Image =
            Image.FromFile(Application.StartupPath.Substring(0,
                Application.StartupPath.Length - 10) +
                "\\images\\denvang.jpg");
            timer1.Interval = 1000;
            den += 1;
            break;

        case 3: Pct.Image =
            Image.FromFile(Application.StartupPath.Substring(0,
                Application.StartupPath.Length - 10) +
                "\\images\\dendo.jpg");
            timer1.Interval = 3000;
            den = 1;
            break;
    }
}
```

Mở rộng:

- d. Thêm 3 TextBox cho phép người dùng nhập vào thời gian dừng giữa các đèn.

Bài 5.3 Viết ứng dụng Demo ScrollBar có giao diện như sau:



Yêu cầu:

- Thiết kế giao diện ứng dụng Demo ScrollBar.
- Khi thực hiện thay đổi vị trí trên các thanh cuộn, giá trị trên các TextBox tương ứng cũng thay đổi theo và màu sắc của hình vuông cũng thay đổi theo màu tương ứng.

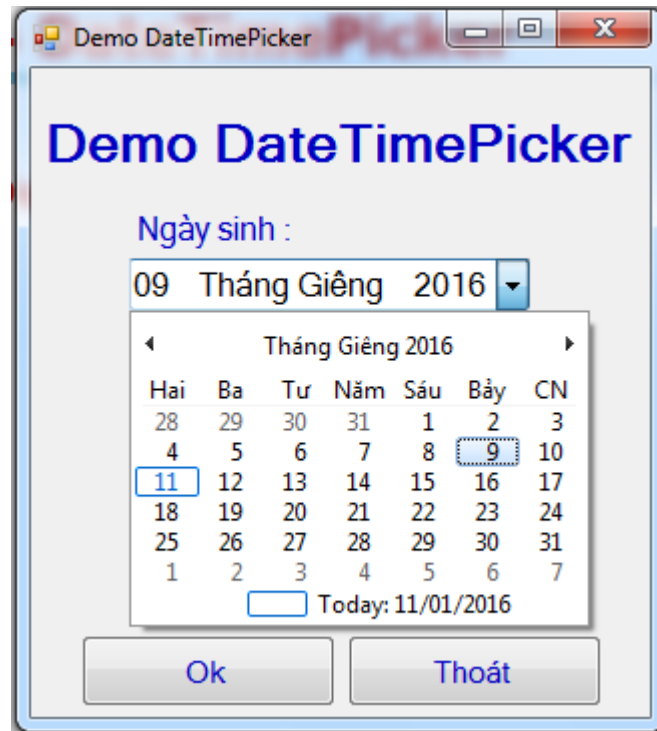
Hướng dẫn:

- Viết code cho sự kiện Scroll của ScrollBar.

```
private void hScrollBar_Red_Scroll(object sender,
ScrollEventArgs e)
{
    Txt_Red.Text = hScrollBar_Red.Value.ToString();
    Txt_Green.Text = hScrollBar_Green.Value.ToString();
    Txt_Blue.Text = hScrollBar_Blue.Value.ToString();
    Pct.BackColor =
    Color.FromArgb(Int32.Parse(Txt_Red.Text),
    Int32.Parse(Txt_Green.Text),
    Int32.Parse(Txt_Blue.Text));
}
```

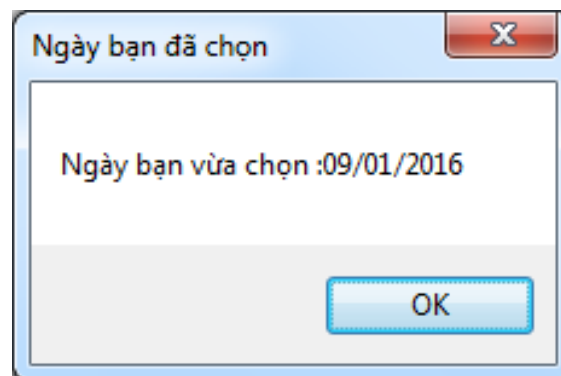
- Thiết lập sự kiện Scroll của ScrollBar Green và Blue.

Bài 5.4 Viết ứng dụng Demo DateTimePicker có giao diện như sau:



Yêu cầu:

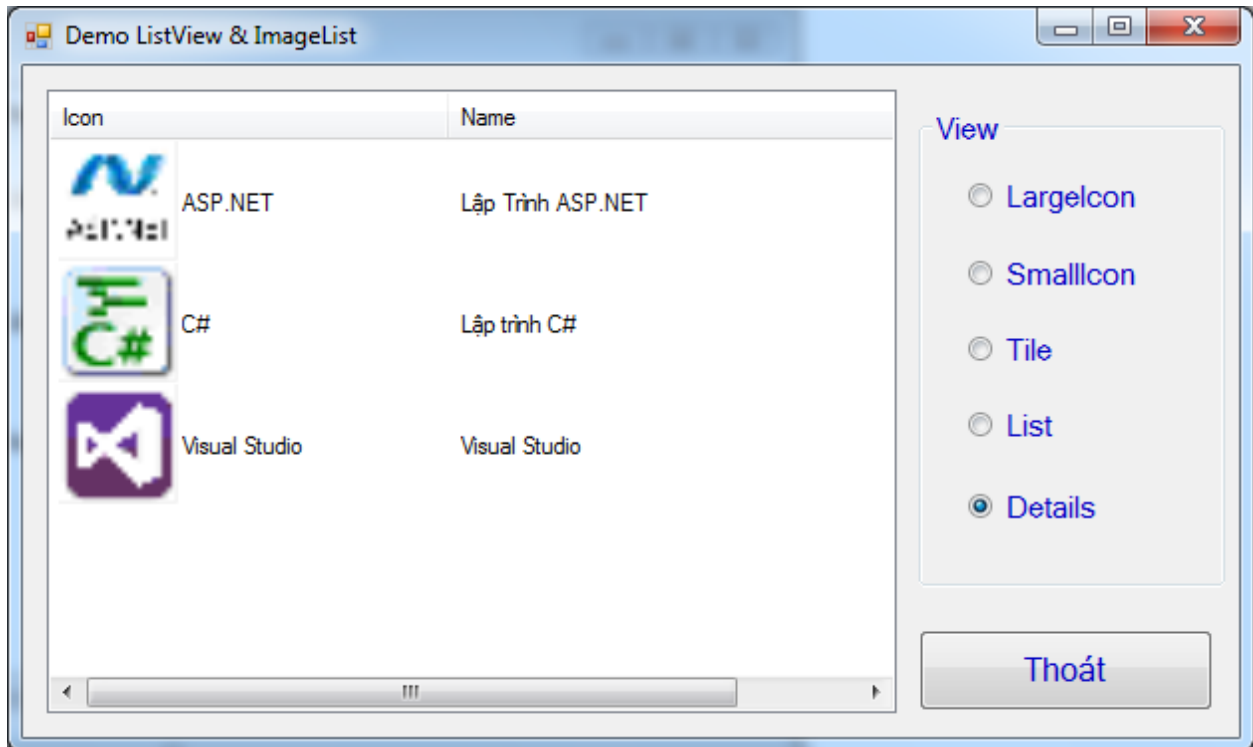
- Thiết kế giao diện ứng dụng Demo DateTimePicker.
- Khi người dùng thực hiện chọn một ngày tùy ý và nhấn OK thì chương trình xuất hiện MessageBox như sau:



Hướng dẫn:

- Viết code cho sự kiện Click của Button OK như sau.

```
private void Btn_Ok_Click(object sender, EventArgs e)
{
    MessageBox.Show("Ngày bạn vừa chọn :"+
        tp.Value.ToString("dd/MM/yyyy"), "Ngày bạn đã chọn");
}
```

**Yêu cầu:**

- Thiết kế giao diện ứng dụng Demo ListView và ImageList.
- Khi người dùng thực hiện chọn một trong các chế độ hiển thị của ListView thì trên ListView hiển thị các thông tin tương ứng.

Hướng dẫn:

- Dữ liệu trên ListView được nhập trên cửa sổ thiết kế.
- Viết code cho sự kiện Click của các RadioButton như sau.

```
private void Rdb_LargeIcon_CheckedChanged(object sender,
EventArgs e)
{
    listView_Demo.View = View.LargeIcon;
}
```

```
private void Rdb_SmallIcon_CheckedChanged(object sender,
EventArgs e)
{
    listView_Demo.View = View.SmallIcon;
}
```

```
private void Rdb_Tile_CheckedChanged_1(object sender,
EventArgs e)
{
    listView_Demo.View = View.Tile;
}
```

```
private void Rdb_List_CheckedChanged(object sender,
EventArgs e)
```

```
{
    listView_Demo.View = View.List;
}

private void Rdb_Details_CheckedChanged(object sender,
EventArgs e)
{
    listView_Demo.View = View.Details;
}
```

Bài 5.6 Viết ứng dụng Demo TabControl và ListView có giao diện như sau:

Icon	Mã Lớp	Tên Lớp	Mô Tả
Icon	01	Lập Trình C#	Lớp Tối 2,4,6
Icon	02	Lập Trình ASP.NET	Lớp Tối 3,5,7
Icon	03	Lập Trình Windows P...	Lớp Tối 3,5,7

Yêu cầu:

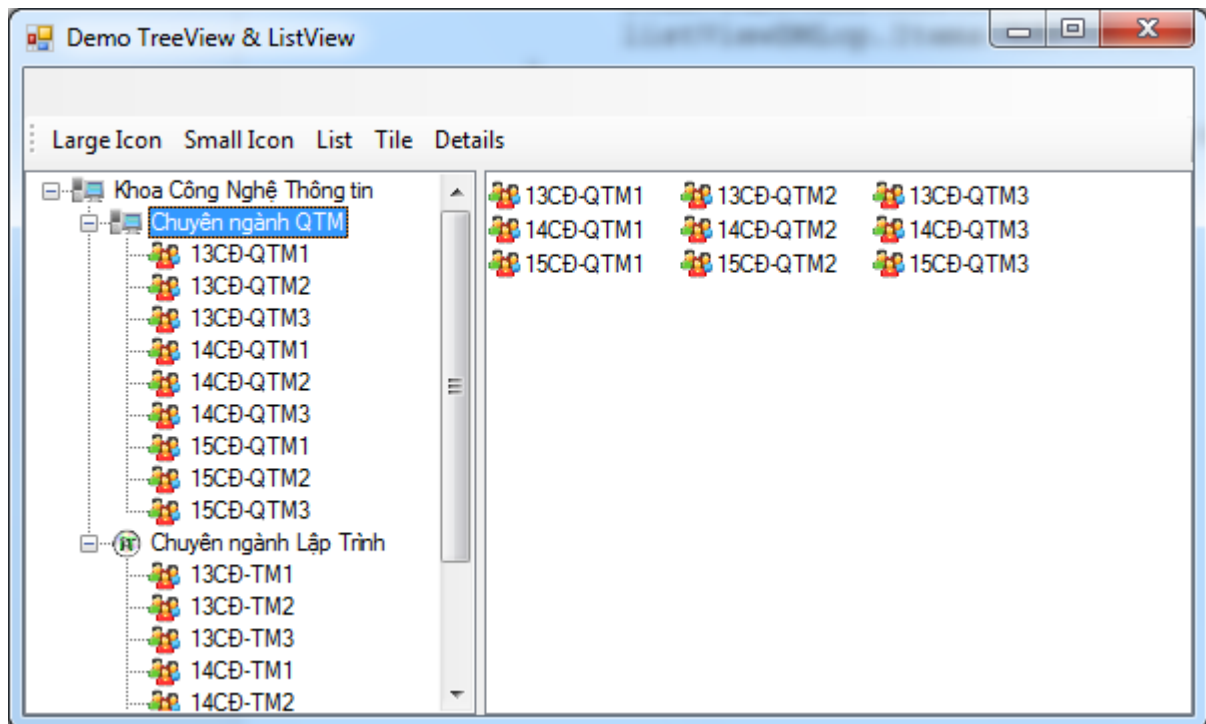
- Thiết kế giao diện ứng dụng Demo TabControl và ListView.
- Button Thêm: Xoá trắng các TextBox Mã lớp, Tên lớp, Mô tả.
- Button Xoá: Chọn một dòng trong ListView và nhấn Button xoá thì xoá khỏi ListView.
- Button Sửa: Chọn một dòng trong ListView và nhấn Button Sửa thì dữ liệu được chuyển lên các TextBox Mã lớp, Tên lớp, Mô tả cho người dùng sửa và nhấn Button lưu dữ liệu vào ListView.
- Button Lưu: Thực hiện lưu dữ liệu trên các TextBox Mã lớp, Tên lớp, Mô tả vào ListView.
- Button Thoát: Thoát khỏi ứng dụng.

Hướng dẫn:

- a. Viết code cho sự kiện Click của Button Thêm.

```
private void Btn_Them_Click(object sender, EventArgs e)
{
    int imageIndex = listViewDMLop.Items.Count;
    //Tạo icon
    ListViewItem item = new ListViewItem("Icon");
    item.ImageIndex = imageIndex;
    //Add cột Mã lớp
    ListViewItem.ListViewSubItem subItem1 = new
    ListViewItem.ListViewSubItem(item, Txt_MaLop.Text);
    item.SubItems.Add(subItem1);
    //Add cột Tên lớp
    ListViewItem.ListViewSubItem subItem2 = new
    ListViewItem.ListViewSubItem(item, Txt_TenLop.Text);
    item.SubItems.Add(subItem2);
    //Add cột Mô tả
    ListViewItem.ListViewSubItem subItem3 = new
    ListViewItem.ListViewSubItem(item, Txt_MoTa.Text);
    item.SubItems.Add(subItem3);
    listViewDMLop.Items.Add(item);
}
```

Bài 5.7 Viết ứng dụng Demo TreeView và ListView có giao diện như sau:



Yêu cầu:

- Thiết kế giao diện ứng dụng Demo TreeView và ListView.
- Dữ liệu trên TreeView được hiển thị khi form load.
- Khi người dùng chọn một Node trên TreeView (cửa sổ bên trái) thì các Node con được hiển thị ở ListView (cửa sổ bên phải).

Hướng dẫn:

a. Viết code cho sự kiện form load.

```
private void Frm_DemoTreeView_ListView_Load(object sender, EventArgs e)
{
    TreeViewCNTT.ImageList = imageList_CNTT;

    //Root Node
    TreeNode RootNode = new TreeNode("Khoa Công Nghệ Thông tin");
    RootNode.ImageIndex = 0;
    RootNode.Expand();

    //Add Node chuyên ngành QTM
    TreeNode ChildNode1 = new TreeNode("Chuyên ngành QTM");
    ChildNode1.ImageIndex = 1;
    ChildNode1.Tag = "Chuyên ngành QTM";
    ChildNode1.Expand();

    // Add các lớp thuộc chuyên ngành QTM
    string tenlop = "";
    for (int khoa = 13; khoa <= 15; khoa++)
        for (int lop = 1; lop <= 3; lop++)
        {
            tenlop = khoa.ToString() + "CD-QTM" + lop.ToString();
            TreeNode ChildNode11 = new TreeNode(tenlop);
            ChildNode11.ImageIndex = 2;
            ChildNode11.Tag = lop;
            ChildNode11.Expand();
            ChildNode1.Nodes.Add(ChildNode11);
        }
    RootNode.Nodes.Add(ChildNode1);

    //Add Node chuyên ngành Lập Trình
    TreeNode ChildNode2 = new TreeNode("Chuyên ngành Lập Trình");
    ChildNode2.ImageIndex = 1;
    ChildNode2.Tag = "Chuyên ngành Lập Trình";
    ChildNode2.Expand();

    // Add các lớp thuộc chuyên ngành Lập trình
    for (int khoa = 13; khoa <= 15; khoa++)
        for (int lop = 1; lop <= 3; lop++)
        {
            tenlop = khoa.ToString() + "CD-TM" + lop.ToString();
            TreeNode ChildNode12 = new TreeNode(tenlop);
            ChildNode12.ImageIndex = 2;
            ChildNode12.Tag = lop;
            ChildNode12.Expand();
            ChildNode2.Nodes.Add(ChildNode12);
        }
    RootNode.Nodes.Add(ChildNode2);

    //Add các Node vào TreeView
    TreeViewCNTT.Nodes.Add(RootNode);
}
```

b. Viết code cho sự kiện TreeViewCNTT_AfterSelect.

```
private void TreeViewCNTT_AfterSelect(object sender, TreeViewEventArgs e)
{
    listView_CNTT.Items.Clear();

    for (int i=0; i<TreeViewCNTT.SelectedNode.Nodes.Count; i++)
    {
```

```
        ListViewItem item = new  
        ListViewItem(TreeViewCNTT.SelectedNode.Nodes[i].Text);  
  
        item.ImageIndex=TreeViewCNTT.SelectedNode.Nodes[i].ImageInd  
        ex;  
        listView_CNTT.Items.Add(item);  
    }  
}
```

--- HẾT---

Chương 6: MẢNG VÀ CHUỖI

A. Tóm tắt lý thuyết

1. Giới thiệu

- Những đối tượng này tương tự như trong ngôn ngữ lập trình C/C++ đã được học.
Mảng và chuỗi dùng để lưu trữ giá trị của các biến trong lập trình.

2. Những kiến thức cơ bản

- Mảng: cho phép lưu nhiều giá trị kiểu số, liên tiếp nhau vào một mảng, trong đó truy xuất theo vị trí đã quy ước trước.
- Chuỗi: dùng để lưu một dãy ký tự trên bộ nhớ.

B. Bài tập cơ bản

Bài 6.1 Các thao tác nhập, xuất, sắp xếp trên mảng.



Hướng dẫn:

1. Khai báo mảng toàn cục, với kích thước mảng 5 phần tử

```
//Khai báo mảng số nguyên 5 phần tử.
int[] A= new int[5];
//Khai báo biến lưu số phần tử
int spt = 0;//Khởi tạo = 0
```

2. Viết Code xử lý cho Button Nhập

```
private void Btn_Nhap_Click(object sender, EventArgs e)
{
    if (spt == 5)
        MessageBox.Show("Mảng đã đầy !");
    else
    {
        A[spt] = Convert.ToInt32(Txt_Nhap.Text);
        spt++;
        Txt_Nhap.ResetText();
        Txt_Nhap.Focus();
    }
}
```

3. Viết Code xử lý cho Button Random

```
private void Btn_Random_Click(object sender, EventArgs e)
{
    Random r = new Random();
    for (spt = 0; spt < 5; spt++)
        A[spt] = r.Next(0, 100);
}
```

4. Viết Code xử lý cho Button Sắp xếp tăng.

```
private void Btn_SXTang_Click(object sender, EventArgs e)
{
    if (spt == 0)
        MessageBox.Show("Mảng rỗng !");
    else
    {
        Array.Sort(A, 0, spt);
        Lbl_KQ.Text = "Đã sắp xếp mảng tăng dần !";
    }
}
```

5. Viết Code xử lý cho Button Sắp xếp giảm.

```
private void Btn_SXGiam_Click(object sender, EventArgs e)
{
    if (spt == 0)
        MessageBox.Show("Mảng rỗng !");
    else
    {
        Array.Reverse(A, 0, spt);
        Lbl_KQ.Text = "Đã sắp xếp mảng giảm dần !";
    }
}
```


Bài 6.2 Các thao tác xử lý trên mảng.

Hướng dẫn:

- Khi nhấn nút “Xuất mảng ngẫu nhiên” thì chương trình xuất ra ô Mảng Gốc gồm 10 phần tử với giá trị mỗi phần tử phải thuộc [1,100].
 - Các nút còn lại khi được nhấn sẽ xuất ra ô Kết quả.
1. Khai báo mảng toàn cục, với kích thước mảng MAX_SIZE là một hằng số:

```
//Khai báo mảng
const int MAX_SIZE = 10;
int[] intMyArray;
```

2. Khởi tạo mảng trong sự kiện Load của form:

```
//khởi tạo mảng
intMyArray = new int[MAX_SIZE];
```

3. Xây dựng các hàm theo yêu cầu:

```
//Định nghĩa phương thức nhập mảng (thủ tục, không giá trị trả về)
private void NhapMang()
{
    Random r = new Random();

    for (int i = 0; i < intMyArray.Length; i++)
        intMyArray[i] = r.Next(1, 101);
}
```

```
//Định nghĩa phương thức Xuất mảng (thủ tục, không giá trị trả về)
private void XuatMang(int[] a)
{
    for (int i = 0; i < a.Length; i++)
        lblGoc.Text += "    " + a[i].ToString();
}
//Định nghĩa phương thức tính tổng mảng (hàm, giá trị trả về kiểu int)
private int TongMang(int[] a)
{
    int Kq = 0;
    for (int i = 0; i < a.Length; i++)
        Kq += a[i];

    return Kq;
}
```

4. Gọi sử dụng các phương thức tại các button.

```
private void btnXuatMang_Click(object sender, EventArgs e)
{
    this.NhapMang(); //gọi thủ tục
    this.XuatMang(intMyArray);
}

private void btnTongMang_Click(object sender, EventArgs e)
{
    //gọi hàm. lấy giá trị trả về của hàm xuất ra Label
    lblKetQua.Text = "Tổng Mảng = " + TongMang(intMyArray).ToString();
}
```

Bài 6.3 Chương trình xử lý chuỗi.



Hướng dẫn:

1. Viết Code cho Button Compare:

```
private void Btn_Compare_Click(object sender, EventArgs e)
{
    string s1 = Txt_S1.Text;
    string s2 = Txt_S2.Text;
    //So sánh s1 và s2 có phân biệt chữ hoa và chữ thường
    int kq = string.Compare(s1, s2);
    Lbl_KQ.Text = Txt_S1.Text;
    if (kq == -1)
        Lbl_KQ.Text += " < ";
    else if (kq==0)
        Lbl_KQ.Text += " = ";
    else
        Lbl_KQ.Text += " > ";
    Lbl_KQ.Text += Txt_S2.Text;
}
```

2. Viết Code cho Button Concat:

```
private void Btn_Concat_Click(object sender, EventArgs e)
{
    string s1 = Txt_S1.Text;
    string s2 = Txt_S2.Text;
    Lbl_KQ.Text = string.Concat(s1, s2);
}
```

3. Viết Code cho Button Insert:

```
private void Btn_Insert_Click(object sender, EventArgs e)
{
    string s1 = Txt_S1.Text;
    string s2 = Txt_S2.Text;
    Lbl_KQ.Text = "Chèn chuỗi 2 vào sau từ đầu tiên của chuỗi 1\n";
    Lbl_KQ.Text += s1.Insert(s1.IndexOf(" "), s2);

    Lbl_KQ.Text += "\nChèn chuỗi 2 vào trước từ cuối cùng của chuỗi 1\n";
    Lbl_KQ.Text += s1.Insert(s1.LastIndexOf(" "), s2);
}
```

4. Viết Code cho Button Xóa:

```
private void Btn_Xoa_Click(object sender, EventArgs e)
{
    Txt_S1.Clear();
    Txt_S2.Clear();
    Lbl_KQ.Text = "";
}
```

5. Viết Code cho Button COMPARE:

```
private void BtnCOMPARE_Click(object sender, EventArgs e)
{
    string s1 = Txt_S1.Text;
    string s2 = Txt_S2.Text;
    //So sánh s1 và s2 KHÔNG phân biệt chữ hoa và chữ thường
    int kq = string.Compare(s1, s2, true);
    Lbl_KQ.Text = Txt_S1.Text;
    if (kq == -1)
        Lbl_KQ.Text += " < ";
    else if (kq == 0)
        Lbl_KQ.Text += " = ";
    else
        Lbl_KQ.Text += " > ";
    Lbl_KQ.Text += Txt_S2.Text;
}
```

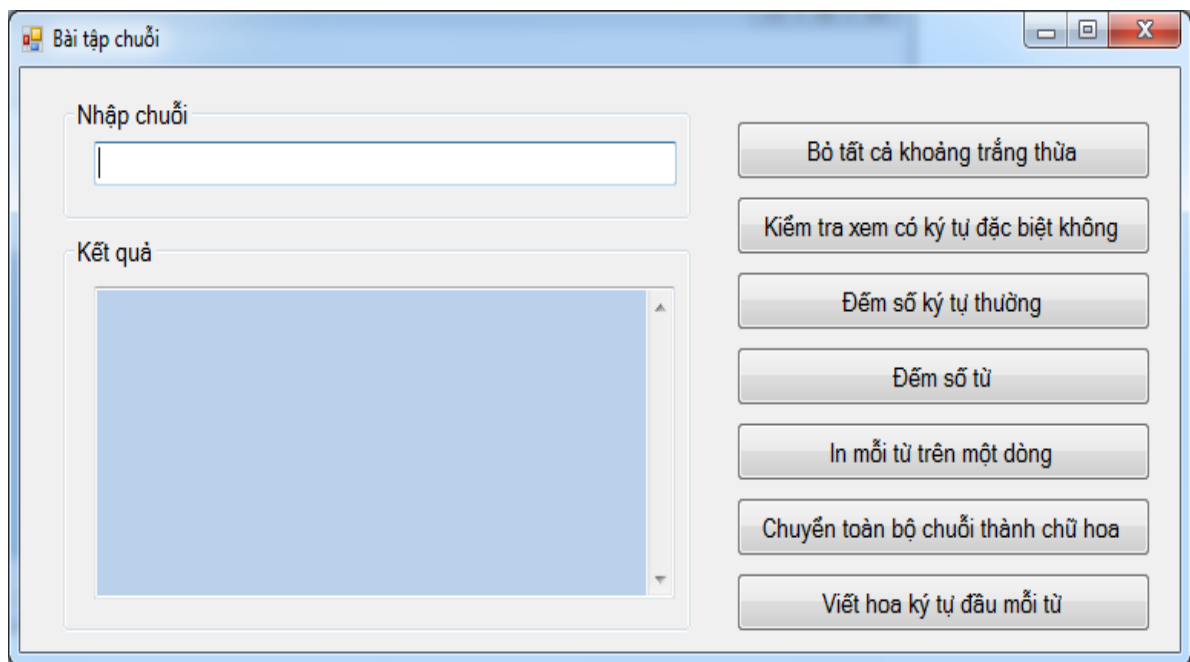
6. Viết Code cho Button IndexOf:

```
private void Btn_Indexof_Click(object sender, EventArgs e)
{
    string s1 = Txt_S1.Text;
    string s2 = Txt_S2.Text;
    //Chỉ ra vị trí xuất hiện của chuỗi 2 trong chuỗi 1
    if (s1.IndexOf(s2) >= 0)
        Lbl_KQ.Text = "Chuỗi 2 xuất hiện trong chuỗi 1 tại vị trí "
            + s1.IndexOf(s2).ToString();
    else
        Lbl_KQ.Text = "Chuỗi 2 không xuất hiện trong chuỗi 1 ";
}
```

7. Viết Code cho Button SubString:

```
private void Btn_SubString_Click(object sender, EventArgs e)
{
    //Trích chuỗi con "CHUỖI" từ Lbl chương trình xử lý chuỗi.
    string S = Lbl_TieuDe.Text;
    Lbl_KQ.Text = S.Substring(S.LastIndexOf(" "), S.Length - S.LastIndexOf(" "));
}
```

Bài 6.4 Các thao tác xử lý chuỗi.



Hướng dẫn:

- Hàm bỏ tất cả các khoảng trắng thừa:
`private string BoKhoangTrangThua(string s)`

```
{
    string re;
    re = s.Trim();
    while(re.IndexOf(" ") != -1)
        re = re.Replace(" ", "");
    return re;
}
```

- Hàm kiểm tra ký tự đặc biệt:

`private bool KiemTraKyTuDacBiet(string s)`

```
{
    string sdacbiet = "~!@$%^&*()_+0123456789-={}|[]:;<>?,./\"\\\"";
    foreach (char c in sdacbiet)
    {
        if (s.IndexOf(c) >= 0) return true;
    }
    return false;
}
```

- Hàm đếm số ký tự thường:

```
private int DemSoKyTuThuong(string s)
{
    int d = 0;
    for (int i = 0; i < s.Length; i++)
        if (char.IsLower(s[i])) d++;
    return d;
}
```

- Hàm đếm số từ:

```
private int DemSoTu(string s)
{
    s = BoKhoangTrangThua(s);
    string[] re = s.Split(new char[] { ' ' });
    return re.Length;
}
```

--- HẾT ---

Chương 7: CHƯƠNG TRÌNH CÓ NHIỀU FORM

A. Tóm tắt lý thuyết

1. Giới thiệu

- Trong một chương trình cụ thể. Ta thường thấy có nhiều giao diện cùng tồn tại trong một chương trình. Mỗi giao diện sẽ đảm nhiệm một chức năng khác nhau.

- Mục đích của việc làm này là phân chia công việc của chương trình ra nhiều việc nhỏ, mỗi tác vụ con sẽ được giao cho từng giao diện riêng biệt.

2. Những kiến thức cơ bản

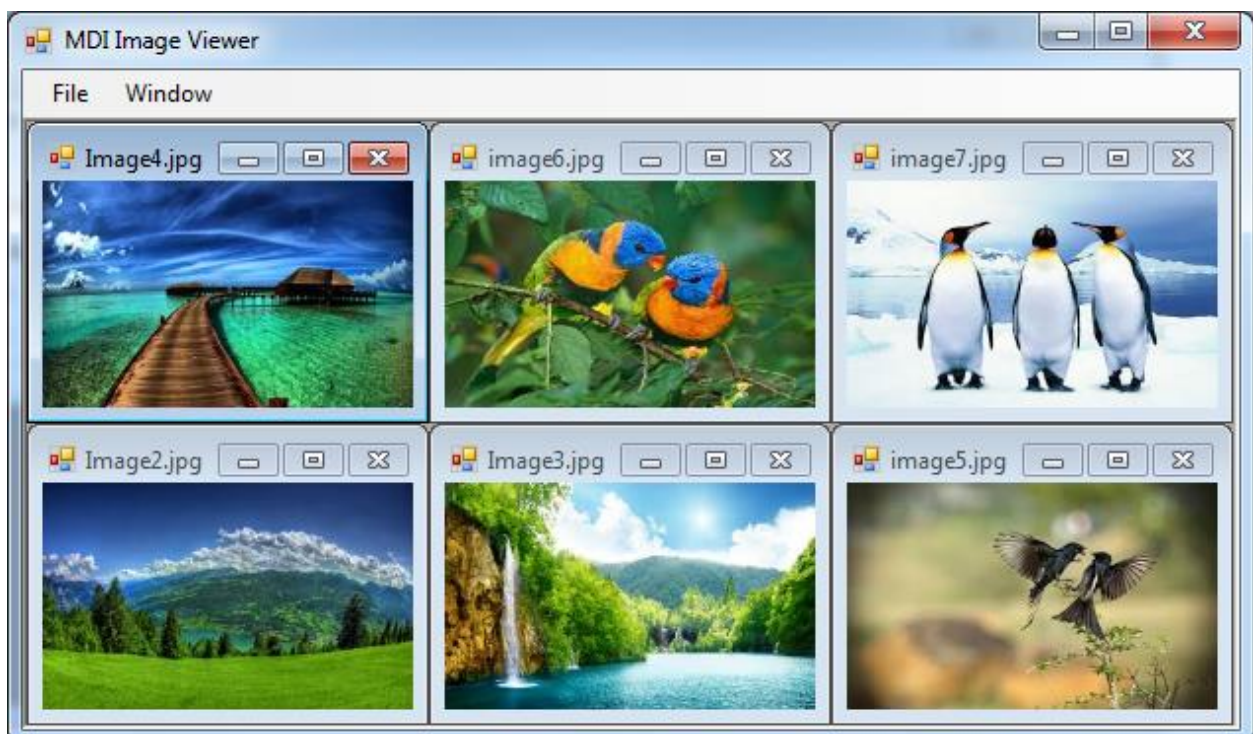
- Tạo, xóa, sửa form: đây là kiến thức cơ bản nhất khi thiết kế trong Visual Studio.

- Khai báo và truy xuất dữ liệu giữa các form: nắm được cách truyền dữ liệu giữa chúng.

- Form cha, form con: giao diện chính chứa nhiều giao diện con để làm việc tiện lợi, thay vì phải thực hiện thao tác chuyển giao diện, tốn thời gian của người dùng.

B. Bài tập cơ bản

Bài 7.1 Xây dựng ứng dụng xem hình MDI ImageViewer như hình sau:

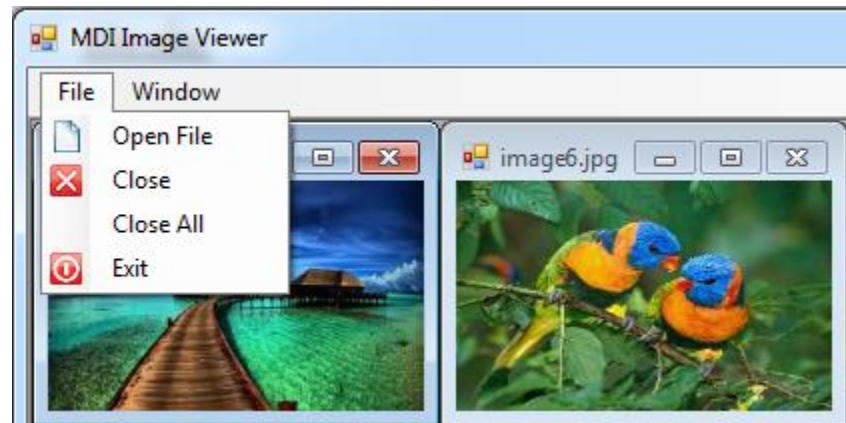


Yêu cầu:

- Xây dựng ứng dụng xem hình có thể xem nhiều hình, mỗi hình trên mỗi cửa sổ con trong ứng dụng.
- Các chức năng của chương trình gồm:

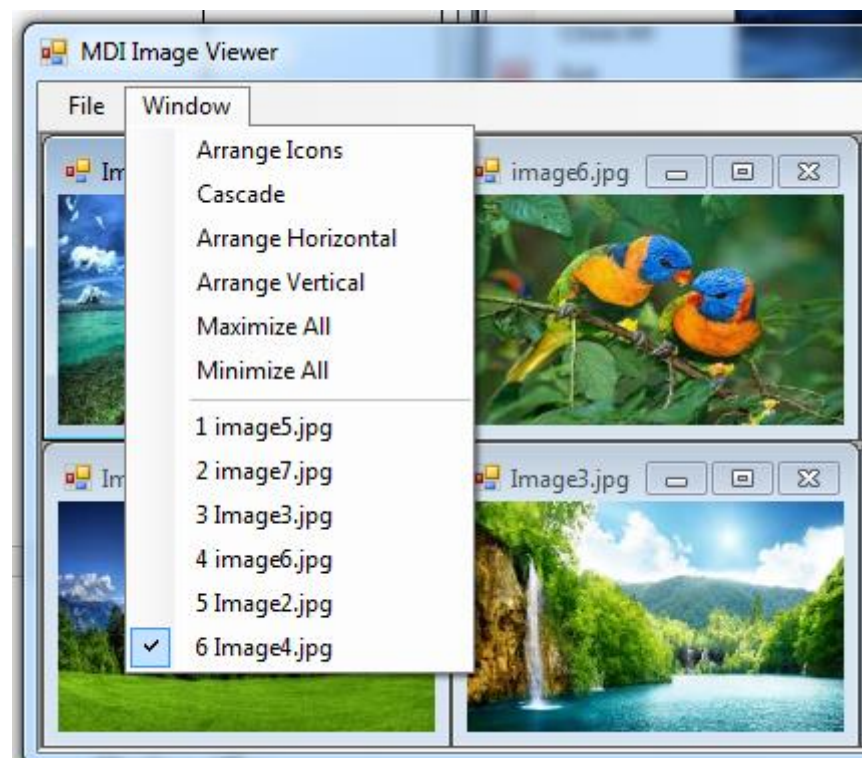
File

Open file : Mở file hình hiển thị trên một cửa sổ mới.
Close: Đóng cửa sổ đang Active.
Close All: Đóng tất cả các cửa sổ hiển thị hình ảnh.
Exit : Thoát khỏi chương trình.



Windows

Arrange Icons: Sắp xếp các cửa sổ theo Icon.
Cascade : Sắp xếp các cửa sổ theo dạng Cascade.
Arrange Horizontal: Sắp xếp các cửa sổ theo dạng Horizontal.
Arrange Vertical: Sắp xếp các cửa sổ theo dạng Vertical.
Maximize All: Phóng to tất cả các cửa sổ.
Minimize All: Thu nhỏ tất cả các cửa sổ.



Hướng dẫn:

- Tạo form chính như sau:

Thuộc tính	Giá trị
Name	Frm_Main
Text	MDI Image Viewer
isMdiContainer	True

- Tạo Menu như sau:

Thuộc tính	Giá trị
Name	menu_MDI_Image_Viewer
MdiWindowListItem	mnu_WindowItem

- Tạo form con như sau:

Thuộc tính	Giá trị
Name	Frm_Child
Text	MDI Image Viewer
isMdiContainer	True

- Tạo PictureBox trên form con như sau:

Thuộc tính	Giá trị
Name	Pic_ImageView
Dock	Fill
SizeMode	StretchImage

- Viết Code cho các chức năng trên menu của form chính như sau:
 - + File -> Open File.

```
private void Menu_File_New_Click(object sender, EventArgs e)
{
    oFileDialog.Title = "Open a Picture file";
    //oFileDialog.InitialDirectory = "C:\\\\";
    oFileDialog.CheckFileExists = true;
    oFileDialog.CheckPathExists = true;
    oFileDialog.ValidateNames = true;
    oFileDialog.Filter = "JPEG files (*.JPEG)|*.JPEG|PNG files (*.PNG)|*.PNG|BMP files (*.BMP)|*.BMP|GIF files (*.GIF)|*.GIF|All files (*.*)|*.*";
    if (oFileDialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            //Tạo thể hiện cho form con và truyền file image cần mở cho formcon
            Frm_Child childForm = new Frm_Child(oFileDialog.FileName);
            //Thiết lập form cha cho form con
            childForm.MdiParent = this;
            //Hiển thị form con
            childForm.Show();
        }
        catch
        {
            MessageBox.Show("Lỗi mở file Image", "MDI Image Viewer",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("Bạn chưa chọn file Image", "Thông báo", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
```

+ File -> Close.

```
private void Menu_File_Close_Click(object sender, EventArgs e)
{
    //Form acChild = this.ActiveMdiChild;
    //acChild.Close(); Hoặc
    if (this.ActiveMdiChild != null)
        this.ActiveMdiChild.Close();
}
```

+ File -> Close All.

```
private void Menu_File_Close_All_Click(object sender, EventArgs e)
{
    foreach (Form frm in this.MdiChildren)
        frm.Dispose();
}
```

+ File -> Exit.

```
private void Menu_File_Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

+ Window -> Arrange Icons:

```
private void arrangeIconsToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi (MdiLayout.ArrangeIcons);
}
```

+ Window -> Cascade:

```
private void cascadeToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi (MdiLayout.Cascade);
}
```

+ Window -> Arrange Horizontal:

```
private void arrangeHorizontalToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi (MdiLayout.TileHorizontal);
}
```

+ Window -> Arrange Vertical:

```
private void arrangeVerticalToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi (MdiLayout.TileVertical);
}
```

+ Window -> Maximize All:

```
private void maximizeAllToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (Form frm in this.MdiChildren)
        frm.WindowState = FormWindowState.Maximized;
}
```

+ Window -> Minimize All:

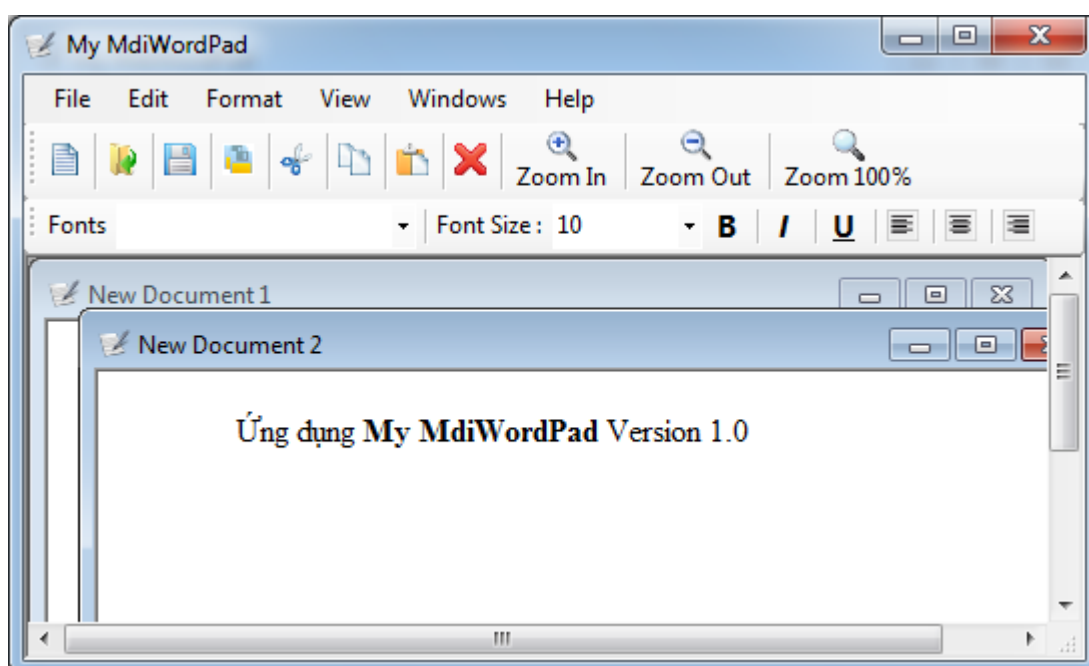
```
private void minimizeAllToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (Form frm in this.MdiChildren)
        frm.WindowState = FormWindowState.Minimized;
}
```

- Viết Code cho constructor của form con như sau:

+ Tham số là tên file cần mở.

```
public partial class Frm_Child : Form
{
    public Frm_Child(string imageFile)
    {
        InitializeComponent();
        //Hiển thị file ảnh lên PictureBox
        Pic_ImageView.Image = Image.FromFile(imageFile);
        //Hiển thị tên của file Ảnh lên Title Bar
        Text = imageFile.Substring(imageFile.LastIndexOf("\\") + 1);
    }
    public Frm_Child()
    {
        InitializeComponent();
    }
}
```

Bài 7.2 Chuyển đổi ứng dụng MyWordPad thành ứng dụng dạng MDI Form như sau:



Yêu cầu:

- Bổ sung thêm các chức năng của chương trình gồm:

File	Close: Đóng cửa sổ đang Active. Close All: Đóng tất cả các cửa sổ hiển thị hình ảnh.
Windows	Arrange Icons: Sắp xếp các cửa sổ theo Icon. Cascade : Sắp xếp các cửa sổ theo dạng Cascade. Arrange Horizontal: Sắp xếp các cửa sổ theo dạng Horizontal. Arrange Vertical: Sắp xếp các cửa sổ theo dạng Vertical. Maximize All: Phóng to tất cả các cửa sổ. Minimize All: Thu nhỏ tất cả các cửa sổ.

Hướng dẫn:

- Các bước thực hiện như sau:
 - + **Bước 1:**
 - Copy ứng dụng MyWordPad và đổi tên Folder thành My MdiWordPad
 - Đổi tên Solution thành My MdiWordPad.
 - Đổi tên Project thành My MdiWordPad.
 - + **Bước 2:**
 - Thực hiện đánh dấu Comment tất cả nội dung Code của các hàm do chúng ta tự xây dựng.
 - + **Bước 3:**
 - Tạo mới một Form con có tên là Frm_Child.
 - Thực hiện Cắt RichTextBox từ Form chính My MdiWordPad qua Frm_Child.
 - Thực hiện Cắt Status Bar từ Form chính My MdiWordPad qua Frm_Child.
 - + **Bước 4:**
 - Thiết lập thuộc tính isMdiContainer=True cho Form chính My MdiWordPad.
 - Bổ sung thêm các Menu Item cho menu trong Form chính My MdiWordPad.
 - Để hiển thị tên các cửa sổ của Frm_Child trên form chính, thiết lập thuộc tính MdiWindowListItem= mnu_WindowsItem.
 - + **Bước 5:**
 - Viết Code cho Frm_Child gồm.
 - Constructor.
 - Xử lý hiển thị dữ liệu cho Status Bar.
 - Tạo ContextMenu cho form Frm_Child.
 - + **Bước 6:**
 - Viết Code cho form chính.

--- HẾT ---

Chương 8: TRUY XUẤT DỮ LIỆU QUA TẬP TIN

A. Tóm tắt lý thuyết

1. Giới thiệu

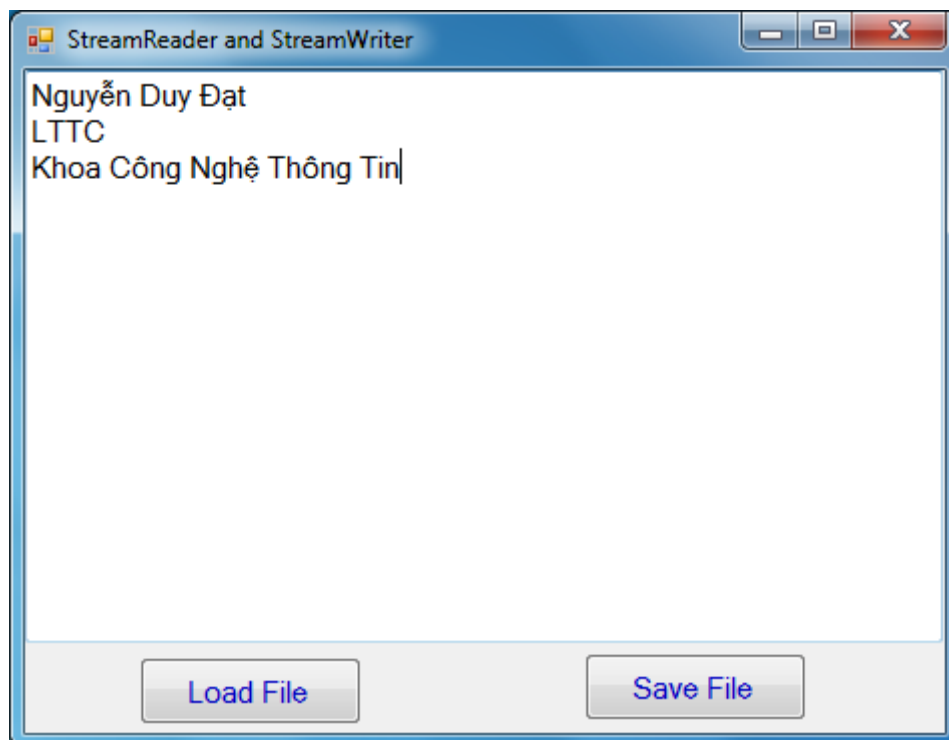
- Mọi phần mềm đều phải có dữ liệu để hoạt động. Do đó thao tác với dữ liệu là hết sức quan trọng và cần thiết.
- Dữ liệu của các phiên làm việc hay những lần làm việc đều phải lưu lại để sử dụng cho những lần sau.

2. Những kiến thức cơ bản

- StreamReader: đọc dữ liệu từ file do người dùng chọn.
- StreamWriter: ghi dữ liệu vào file được chọn.

B. Bài tập cơ bản

Bài 8.1 Xây dựng ứng dụng đọc và ghi file văn bản như hình sau:



Yêu cầu:

- Xây dựng ứng dụng đọc nội dung từ file văn bản (*.txt) lên TextBox và lưu nội dung từ TextBox ra file văn bản (*.txt).

Hướng dẫn:

- Viết code cho Button Load File.

```
private void Btn_Load_File_Click(object sender, EventArgs e)
{
    dlgOpenFile.CheckFileExists = true;
    dlgOpenFile.CheckPathExists = true;
    dlgOpenFile.Filter = "Text file (*.txt) | *.txt";
    if (dlgOpenFile.ShowDialog() == DialogResult.OK)
    {
        Txt_Doc.Text = ReadTextFile(dlgOpenFile.FileName);
    }
    else
    {
        MessageBox.Show("Bạn chưa chọn file ",
            "Thông báo lỗi ", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

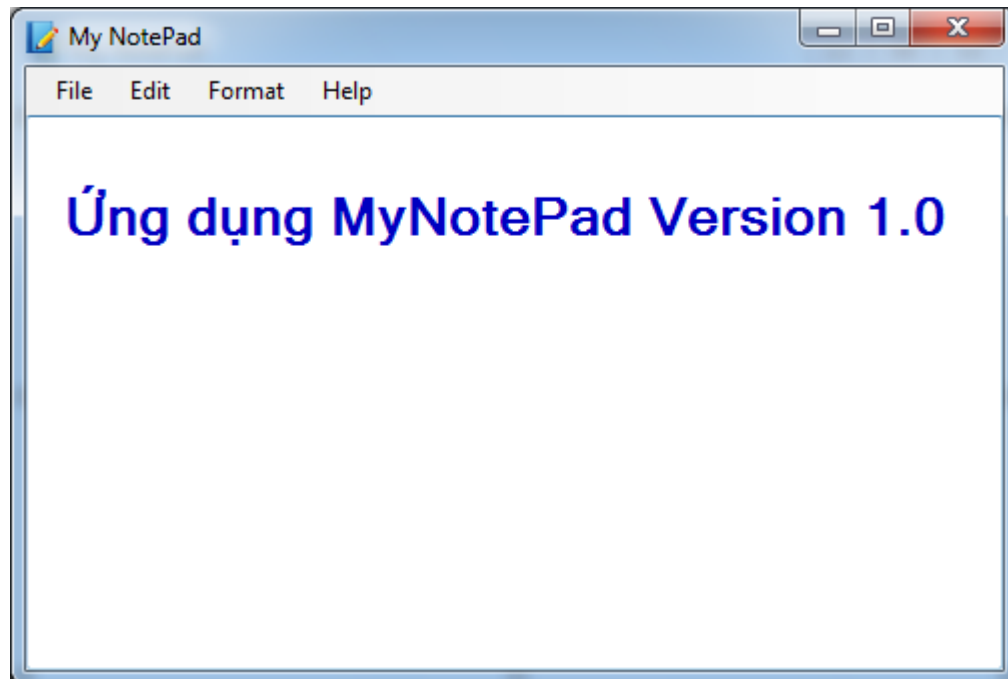
string ReadTextFile(string sfilename)
{
    StreamReader myreader = null;
    string result = string.Empty;
    try
    {
        myreader = new StreamReader(sfilename);
        result = myreader.ReadToEnd();
    }
    catch
    {
        throw;
    }
    finally
    {
        if(myreader != null) myreader.Close();
    }
    return result;
}
```

- Viết code cho Button Save File.

```
private void Btn_SaveFile_Click(object sender, EventArgs e)
{
    dlgSaveFile.CheckFileExists = false;
    dlgSaveFile.CheckPathExists = true;
    dlgSaveFile.Filter = "Text file (*.txt) | *.txt";
    if (dlgSaveFile.ShowDialog() == DialogResult.OK)
    {
        SaveTextFile(dlgSaveFile.FileName);
    }
    else
    {
        MessageBox.Show("Bạn chưa nhập file ",
            "Thông báo lỗi ", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
void SaveTextFile(string sfilename)
{
    StreamWriter sWriter = new StreamWriter(sfilename,
        false, Encoding.Unicode);
    foreach (string line in Txt_Doc.Lines)
        sWriter.WriteLine(line);
    sWriter.Close();
}
```

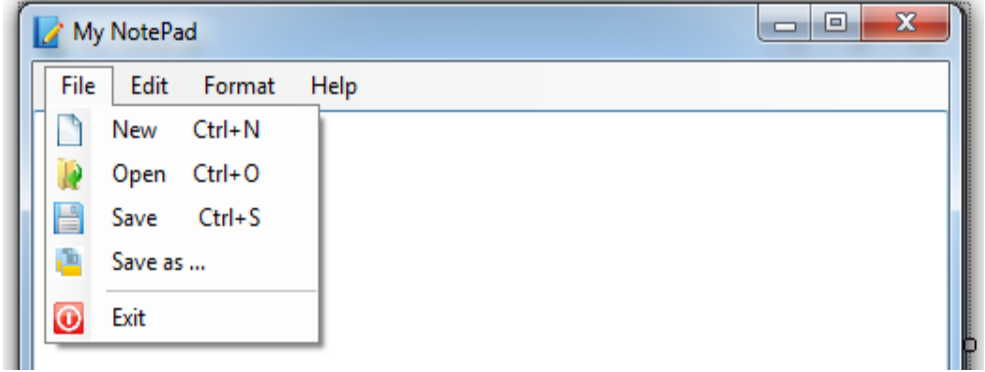
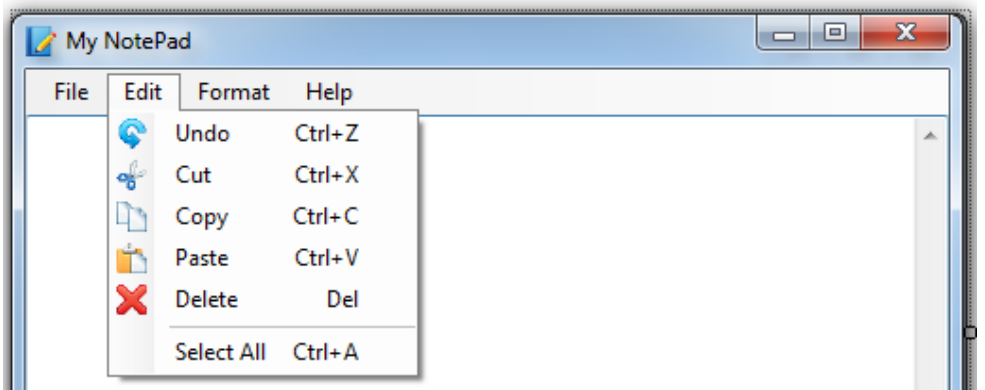
Bài 8.2 Xây dựng ứng dụng soạn thảo văn bản **MyNotePad** như hình sau:



Yêu cầu:

- Xây dựng ứng dụng soạn thảo văn bản (*.txt) có giao diện như trên và hệ thống các chức năng gồm:

File	New: Tạo mới tài liệu. Open: Mở tài liệu đã tồn tại. Save: Lưu file đã tồn tại, nếu file chưa tồn tại sẽ xuất hiện cửa sổ Save as ... Save as: Lưu file với tên file mới. Exit: Thoát khỏi chương trình.
------	--

	 A screenshot of the 'My NotePad' application window. The 'File' menu is open, showing options: New (Ctrl+N), Open (Ctrl+O), Save (Ctrl+S), Save as ..., and Exit.
Edit	Undo: Cut: Cắt nội dung đã chọn khối vào Clipboard. Copy: Sao chép nội dung đã chọn khối vào Clipboard. Paste: Chèn nội dung từ Clipboard vào văn bản tại vị trí con trỏ. Delete: Xoá nội dung đã chọn khối. Select All: Chọn khối toàn bộ tài liệu.  A screenshot of the 'My NotePad' application window. The 'Edit' menu is open, showing options: Undo (Ctrl+Z), Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Delete (Del), and Select All (Ctrl+A).
Format	Font: Cho phép chọn font.
Help	About MyNotePad

--- HẾT ---

TỔNG KẾT

1. Ý nghĩa thực tế:

- Bộ bài tập không những tóm lược lại những lý thuyết đã học ở lớp mà còn hệ thống hóa lại những bài tập này theo quy trình dễ tiếp cận nhất. Trong đó, độ khó và phức tạp được tăng dần. Sinh viên có thể học những bài tập cơ bản nhất, để sau đó những sinh viên này có thể xây dựng những chương trình đầy đủ bao gồm tất cả các khâu và quy trình trong việc xây dựng một phần mềm chuyên nghiệp.

- Trong suốt quá trình học những điều kiện cơ bản cho đến nắm bắt được quy trình sản xuất phần mềm, sinh viên sẽ rèn luyện được rất nhiều kiến thức cần thiết và kỹ năng sống còn của lập trình viên, chẳng hạn: kỹ năng tư duy, kỹ năng lập kế hoạch, kỹ năng phân phối công việc, kỹ năng thiết kế phần mềm và cuối cùng là kỹ năng lập trình chuyên nghiệp. Tất cả những điều này khi kết hợp với kiến thức đã được hệ thống sẽ biến một sinh viên trở thành những người lập trình viên vững cả về kiến thức lẫn chuyên môn thực tế.

- Bộ bài tập này làm phong phú thêm nguồn tài liệu của trường, sinh viên hoàn toàn có thể học một cách chủ động và độc lập với một phần nhỏ sự hướng dẫn của giảng viên. Hướng sinh viên dần quen với việc tự học tập.

2. Ý nghĩa giáo dục:

- Giảng viên sẽ tiếp kiệm công sức rất nhiều, không cần tốn công soạn lại bộ tài liệu thực hành. Một cách gián tiếp, thầy cô đứng lớp có thêm thời gian để hướng dẫn, giáo dục sinh viên tốt hơn. Qua đó, sinh viên sẽ có kết quả học tập cũng như rèn luyện về chuyên môn tốt hơn trước kia.

- Việc có một bộ bài tập làm chuẩn sẽ giúp cho việc trao đổi phương pháp, nội dung giảng dạy giữa giảng viên với nhau dễ dàng và nhanh chóng hơn so, kết quả đem lại là giảng viên ngày càng vững vàng về mặt sư phạm lẫn kiến thức chuyên môn cũng như nội dung truyền đạt cho sinh viên.

- Đề tài này giúp cho sinh viên hoàn thiện cả về hiểu biết lẫn khả năng thực hành, do đó sẽ tăng cường chất lượng sinh viên ra trường. Một cách gián tiếp, chất lượng giáo dục sẽ được nâng cao và đem lại vị thế xứng đáng cho nhà trường.

3. Mục đích nghiên cứu:

- Giải quyết những vấn đề thực tiễn đang còn tồn đọng trong chuyên môn lẫn cải thiện chất lượng sinh viên đầu ra đối với chuyên ngành Công nghệ phần mềm.

- Về mặt chuyên môn giảng viên, đề tài này hướng tới việc thống nhất nội dung giảng dạy thực hành. Việc làm này đem lại ích lợi trong toàn bộ quy trình giảng dạy, đánh giá cũng như nâng cao trình độ chuyên môn của toàn bộ giảng viên.

- Về mặt sinh viên, họ sẽ được trang bị kiến thức đã hệ thống hóa, giúp họ ghi nhớ dễ dàng nhanh chóng. Sinh viên sẽ được trang bị cả kiến thức đã hệ thống hóa và những kỹ năng để hành nghề.

- Về mặt nhà trường, kho tài liệu sẽ ngày càng tăng cả về số lượng lẫn chất lượng. Đặc biệt dành cho ngành Công nghệ phần mềm.

- Về phía doanh nghiệp, họ sẽ nhận được những sinh viên vững kiến thức chuyên môn, tác phong làm việc chuyên nghiệp, hiệu quả cao.

- Về mặt khoa học, giảng viên từng bước tiến đến việc nghiên cứu khoa học chuyên sâu và khoa học ứng dụng. Qua đó, làm giàu thêm sự hiểu biết và đào tạo ra lực lượng lao động có chất lượng cho xã hội.