

MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài thực hành số 8.1 : Xây dựng ứng dụng hiển thị nội dung file ở dạng Binary (Hexadecimal)

I. Mục tiêu :

- Giúp SV làm quen với qui trình thiết kế trực quan 1 ứng dụng Dialog Based.
- Giúp SV làm quen với việc dùng lại linh kiện phần mềm.
- Giúp SV thấy cụ thể cấu trúc ứng dụng cấu thành từ các đối tượng.
- Giúp SV thấy sự tương tác giữa các đối tượng.
- Giúp SV làm quen với cách thức viết code để đọc dữ liệu ở dạng nhị phân từ file để xử lý/tính toán ngay bên trong máy mà không cần tốn chi phí mã hóa chúng.

II. Nội dung :

- Xây dựng ứng dụng Dialog Based hiển thị nội dung file ở dạng Binary (Hexadecimal).
- Dịch và chạy chương trình.
- Debug chương trình để tìm lỗi nếu có.


III. Chuẩn đầu ra :

- Thành thạo việc xây dựng 1 ứng dụng theo qui trình thiết kế trực quan.
- Thành thạo việc dùng lại linh kiện phần mềm có sẵn, thấy rõ cấu trúc phần mềm và sự tương tác giữa các đối tượng trong phần mềm.
- Thành thạo việc viết code thay đổi kích thước và vị trí các đối tượng giao diện khi cửa sổ chứa chúng bị thay đổi.
- Sinh viên nắm vững và lập trình thành thạo các đoạn code để đọc dữ liệu nhị phân từ file nhị phân vào các biến bên trong chương trình.

IV. Phân tích :

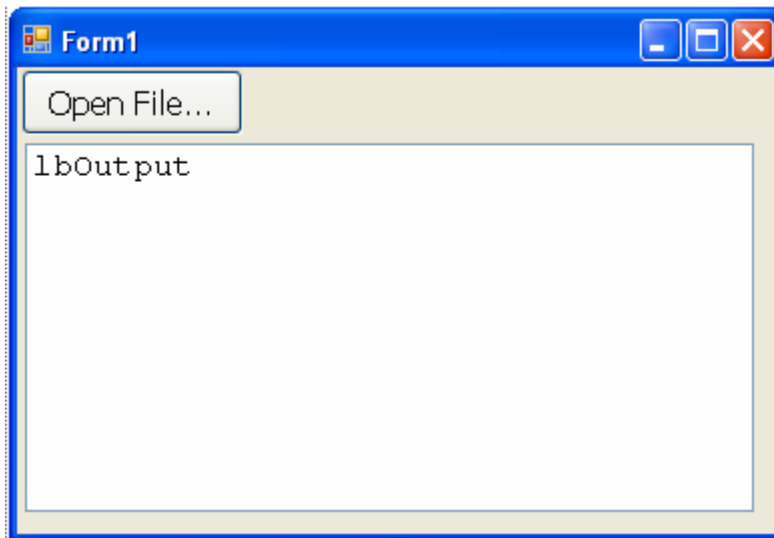
- Sau khi phân tích chức năng của chương trình, ta thấy chương trình là 1 form giao diện trực quan, nó chứa 1 button "Open File..." để người dùng kích hoạt chức năng hiển thị file, nó cũng cần 1 đối tượng cho phép người dùng duyệt trực quan cây thư mục để chọn file (giả sử ta biết đó là đối tượng OpenFileDialog). Cuối cùng chương trình cần 1 đối tượng có khả năng hiển thị nội dung file dạng Binary (đối tượng ListBox).
- Theo kết quả phân tích trên, cấu trúc chương trình cần viết khá đơn giản, nó chỉ chứa các đối tượng đã có sẵn, ta không cần phải bận tâm đặc tả chi tiết các đối tượng cần dùng mà chỉ cần dùng lại chúng, lắp ghép chúng lại để tạo thành chương trình. Đây là trường hợp may mắn nhất, nhưng trong thực tế lập trình hướng đối tượng, ta sẽ có được may mắn này thường xuyên.


V. Qui trình :

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Window, chọn icon "Windows Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (thí dụ BinaryView), click button OK để tạo Project theo các thông số đã khai báo.
3. Form đầu tiên của ứng dụng đã hiển thị trong cửa sổ thiết kế, việc thiết kế form là quá trình lập 4 thao tác tạo mới/xóa/hiệu chỉnh thuộc tính/tạo hàm xử lý sự kiện cho từng đối tượng cần dùng trong form.
4. Nếu cửa sổ ToolBox chưa hiển thị chi tiết, chọn menu View.Toolbox để hiển thị nó (thường nằm ở bên trái màn hình). Click chuột vào button  (Auto Hide) nằm ở góc trên phải của cửa sổ ToolBox để chuyển nó về chế độ hiển thị thường trực.

5. Duyệt tìm phần tử Button (trong nhóm Common Controls), chọn nó, dời chuột về góc trên trái của form và vẽ nó với kích thước mong muốn. Xem cửa sổ thuộc tính của Button vừa vẽ (thường ở góc dưới phải màn hình), duyệt tìm và hiệu chỉnh thuộc tính Text = "Open File...", duyệt tìm và thay đổi thuộc tính (Name) = btnOpen.
6. Duyệt tìm phần tử ListBox (trong nhóm Common Controls), chọn nó, dời chuột về ngay dưới Button và vẽ nó với kích thước mong muốn. Xem cửa sổ thuộc tính của ListBox vừa vẽ (thường ở góc dưới phải màn hình), duyệt tìm và hiệu chỉnh thuộc tính (Name) = lbOutput.

Sau khi thiết kế xong, Form có dạng sau :




7. Dời chuột về button btnOpen, ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị để ta bắt đầu viết code cho hàm. Cách tổng quát để tạo hàm xử lý sự kiện là chọn đối tượng btnOpen, cửa sổ thuộc tính của nó sẽ hiển thị, click icon  để hiển thị danh sách các sự kiện của đối tượng, duyệt tìm sự kiện quan tâm (Click), ấn kép chuột vào comboBox bên phải sự kiện Click để máy tạo tự động hàm xử lý cho sự kiện này. Cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau :

```
private void btnOpen_Click(object sender, EventArgs e) {
    //định nghĩa các biến cần dùng
    int i, j;
    int byt;
    String slLine, srLine;
    String sline;
    OpenFileDialog dlg = new OpenFileDialog();
    //hiển thị form duyệt chọn file cần chơi
    DialogResult ret = dlg.ShowDialog();
    //kiểm tra quyết định của người dùng, nếu người dùng chọn OK thì chơi
    if (ret != DialogResult.OK) return;
    //1. tạo đối tượng quản lý file
    FileStream stream = new FileStream(dlg.FileName, FileMode.Open);
    //2. xác định kích thước file
    long flen = stream.Length;
    //3. xuất từng nhóm 16 byte của file ra thành 1 dòng
    j = 0;
    slLine = srLine = "";
    lbOutput.Items.Clear();
    for (i = 0; i < flen; i++) {
```

```

byt = stream.ReadByte();
sline = String.Format("{0:X2} ", byt);
slLine = slLine + sline;
if (byt < 32) sline = ".";
else sline = Char.ToString((char)byt);
srLine = srLine + sline;
if (++j == 16)
{ //du dòng
    lbOutput.Items.Add(slLine + " " + srLine);
    j = 0;
    slLine = srLine = "";
}
}
if (j == 0) return;
//4. xử lý dòng cuối cùng
while (j++ < 16) slLine = slLine + " ";
lbOutput.Items.Add(slLine + " " + srLine);
//5. đóng file lại
stream.Close();
}

```

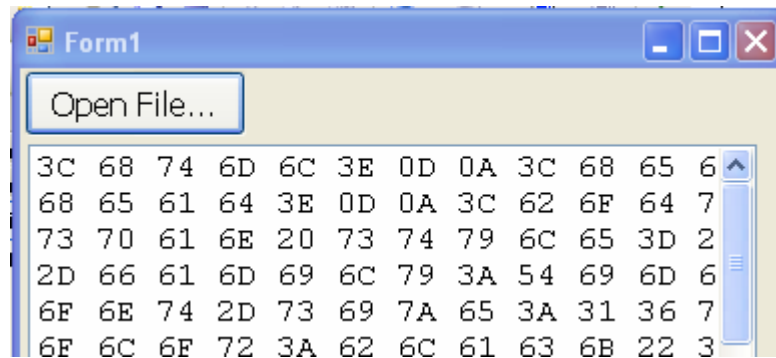
8. Dời chuột về cửa sổ "Solution Explorer", duyệt tìm mục Form1.cs, ấn kép chuột vào mục này để hiển thị lại cửa sổ thiết kế form. Chọn Form để hiển thị cửa sổ thuộc tính của Form, click icon  để hiển thị danh sách các sự kiện của Form, duyệt tìm sự kiện SizeChanged, ấn kép chuột vào comboBox bên phải sự kiện SizeChanged để máy tạo tự động hàm xử lý cho sự kiện này. Cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau :

```

//thay đổi kích thước ListBox theo kích thước mới của Form
private void Form1_SizeChanged(object sender, EventArgs e) {
    //xác định kích thước hiện hành của Form
    int cx = this.Size.Width;
    int cy = this.Size.Height;
    //tính lại kích thước của ListBox
    cx = cx - 8 - lbOutput.Location.X * 2;
    cy = cy - 8 - 25 - lbOutput.Location.Y;
    //thay đổi kích thước của ListBox theo kích thước Form
    lbOutput.Size = new Size(cx, cy);
}

```

9. Chọn menu Debug.Start Debugging để dịch và chạy thử ứng dụng. Khi Form chương trình hiển thị, hãy click chuột vào button "Open File...", cửa sổ duyệt chọn file sẽ hiển thị, hãy duyệt và chọn 1 file nào đó cần xem, click button Open để hiển thị nội dung file này.
10. Ta thấy nội dung trên từng hàng bị che mất bên phải vì độ rộng Form nhỏ quá. Quan sát góc trên phải của Form chương trình, ta thấy 3 button chức năng thông thường là 1. thu nhỏ form về dạng icon, 2. phóng to/thu nhỏ kích thước form, 3. đóng form :



11. Hãy click chuột vào button phóng to/thu nhỏ, form sẽ được phóng to để chiếm hết màn hình và ListBox cũng được phóng to theo, lúc này ta thấy đầy đủ thông tin trên từng hàng.
12. Dừng chương trình lại, hiển thị cửa sổ mã nguồn, cố gắng đọc hiểu đoạn code trong thân của 2 hàm xử lý sự kiện `btnOpen_Click()` và `Form1_SizeChanged()`.