TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

BÁO CÁO ĐỒ ÁN 1: Lập Trình

Ngành Kỹ thuật máy tính

**Chương trình xem ảnh và lưu lịch sử người dùng**

Sinh viên thực hiện: **Phạm Hoài Lâm**

Lớp CN CNTT-2

MSSV: 20166339

Giáo viên hướng dẫn: ThS. **Nguyễn Đức Tiến**

HÀ NỘI 05 -2018

# Yêu cầu

-Tạo một chương trình xem ảnh trong các thư mục và ghi lại lịch sử người dùng.

-Viết bằng ngôn ngữ C# trên nền Visual studio 2017 và sử dụng một project dạng Window Forms

Mục lục

[Yêu cầu 2](#_Toc514358534)

[I. Giới thiệu các kỹ thuật 3](#_Toc514358535)

[1. Microsoft Visual Studio 3](#_Toc514358536)

[2. Window Forms 3](#_Toc514358537)

[3. ListView 4](#_Toc514358538)

[4. ListBox 4](#_Toc514358539)

[5. PictureBox 4](#_Toc514358540)

[6. ImageList 4](#_Toc514358541)

[7. Button 4](#_Toc514358542)

[8. TextBox 4](#_Toc514358543)

[9. ToolTip 4](#_Toc514358544)

[10. FolderBrowserDialog 4](#_Toc514358545)

[11. Resource 4](#_Toc514358546)

[12. Settings 4](#_Toc514358547)

[Settings dùng để lưu trữ các thiết lập ứng dụng hoặc cài đặt người dùng. 4](#_Toc514358548)

[=>Ở đây sử dụng settings để lưu lại lịch sử giao diện của chương trình trước khi người dùng tắt chương trình đi. 4](#_Toc514358549)

[II. Phân tích thiết kế 5](#_Toc514358550)

[1. Giao diện 5](#_Toc514358551)

[2. Mở thư mục chứa ảnh 6](#_Toc514358552)

[3. Chọn ảnh để hiển thị 6](#_Toc514358553)

[4. Khôi phục lịch sử người dùng 7](#_Toc514358554)

[III. Các hàm và biến đáng chú ý 8](#_Toc514358555)

[1. Các biến toàn cục đáng chú ý 8](#_Toc514358556)

[2. btnLoadImage\_Click 8](#_Toc514358557)

[3. listView1\_Click 9](#_Toc514358558)

[4. ImageShow\_Load 10](#_Toc514358559)

[5. ImageShow\_FormClosing 11](#_Toc514358560)

[6. PictureBox1\_MouseHover 11](#_Toc514358561)

[7. Các hàm khác 11](#_Toc514358562)

[Kết Luận 11](#_Toc514358563)

# Giới thiệu các kỹ thuật

## 1. Microsoft Visual Studio

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C,[4] C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010[5]). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

=>Đồ án này sử dụng 1 project dạng Window Forms để thiết kế và xây dựng chương trình.

## 2. Window Forms

Windows Forms (WinForms) là một thư viện lớp đồ họa (GUI), là một phần của Microsoft .NET Framework, cung cấp một nền tảng để viết các ứng dụng cho máy tính để bàn, máy tính xách tay và máy tính bảng. Trong khi nó được xem như là sự thay thế cho thư viện lớp Microsoft Foundation dựa trên C ++, nó không cung cấp một mô hình so sánh và chỉ hoạt động như một nền tảng cho tầng giao diện người dùng.

Windows Forms cơ bản bao gồm:

-Một Form là khung dùng hiển thị thông tin đến người dùng

-Các Control được đặt trong form và được lập trình để đáp ứng sự kiện

=>Đồ án này sử dụng các control được cài đặt trong Form là: ListView, ListBox, PictureBox, ImageList, Button, TextBox, ToolTip và FolderBrowserDialog.

## 3. ListView

ListView là một control dùng để hiển thị một danh sách các item với các biểu tượng. Chúng ta có thể sử dụng một  ListView để tạo ra một giao diện giống như cửa sổ bên phải của Windows Explorer.

## 4. ListBox

ListBox cung cấp một giao diện để hiển thị một danh sách các mục. Người dùng có thể chọn một hoặc nhiều mục từ danh sách. Một ListBox có thể được sử dụng để hiển thị nhiều cột và các cột này có thể có hình ảnh và các điều khiển khác.

## 5. PictureBox

PictureBox dùng để hiển thị hình ảnh trong Window Forms

## 6. ImageList

Một ImageList là một điều khiển hỗ trợ thường được sử dụng bởi các điều khiển khác, ở đây chính là ListView nhưng được tiếp cận như một thành phần cho các nhà phát triển.

## 7. Button

Button cho phép tạo một hay nhiều nút bấm khi ta click vào sẽ chạy các chức năng đã được lập trình sẵn.

=>Ở đây button được dùng để tạo 1 nút mà khi bấm vào sẽ gọi tới control FolderBrowserDialog

## 8. TextBox

TextBox là một control vừa có khả năng xuất thông tin, vừa có khẳ năng nhập thông tin.

->Ở đây chỉ sử dụng chức năng xuất thông tin của TextBox.

## 9. ToolTip

ToolTip là một cửa sổ pop-up nhỏ hiển thị một số thông tin khi bạn ??? vào mộ control khác.

## 10. FolderBrowserDialog

FolderBrowserDialog được sử dụng để duyệt và chọn một thư mục trên máy tính.

## 11. Resource

Resource dùng để đưa dữ liệu không phải mã nguồn như dữ liệu nhị phân, văn bản, string, biểu tượng hoặc bất kỳ kiểu dữ liệu nào khác vào project C#, giúp chương trình khoa học hơn và bớt lộn xộn trong việc phát triển cũng như sửa đổi

## 12. Settings

## Settings dùng để lưu trữ các thiết lập ứng dụng hoặc cài đặt người dùng.

## =>Ở đây sử dụng settings để lưu lại lịch sử giao diện của chương trình trước khi người dùng tắt chương trình đi.

# Phân tích thiết kế

## 1. Giao diện



PictureBox: Hiển thị ảnh

ListBox: Hiện lịch sử

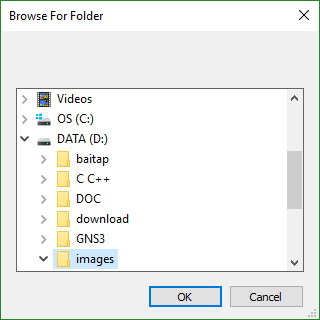
LoadImage: Tải ảnh trong thư mục

ListView: Hiển thị ảnh trong thư mục



## 2. Mở thư mục chứa ảnh

-Nhấn vào nút Load Image sẽ xuất hiện cửa sổ Browse For Folder như sau:



sau đó ấn vào nút OK để chọn thư mục.

-Khi đó trên control ListView sẽ xuất hiện tất cả ảnh trong thư mục vừa chọn

## 3. Chọn ảnh để hiển thị

-Click vào ảnh để chọn, lúc này trên PictureBox sẽ xuất hiện ảnh được chọn, đồng thời lịch sử bao gồm thời gian xem và đường dẫn ảnh sẽ hiển thị trong mục History:



-Nếu ta di chuột vào ảnh sẽ xuất hiện ToolTip hiển thị thông tin về ảnh gồm: tên ảnh, kích thước ảnh, dung lượng ảnh và ngày giờ khởi tạo ảnh.



## 4. Khôi phục lịch sử người dùng

Khi bật chương trình sau khi tắt, chương trình sẽ tự động khôi phục lại lịch sử xem ảnh của người dùng trước đó và thư mục được load ảnh gần nhất.

# Các hàm và biến đáng chú ý

## Các biến toàn cục đáng chú ý

string Dirpath;

StringCollection stringCol = new StringCollection();

int ii= Properties.Settings.Default.History.Count;

int i=0;

int iold, inew;

* **string Dirpath**: Dùng để lưu lại đường dẫn folder mở ảnh
* StringCollection stringCol: Dùng để lưu lại tất cả ảnh đã mở trong phần lịch sử. Biến này phục vụ cho Properties.Settings.History lưu lại lịch sử xem ảnh cho lần truy cập ứng dụng trước của người dùng.
* **int ii**: Biến lưu lại vị trí ảnh mới được thêm vào lịch sử trong listBox1(auto scroll), đồng thời phục vụ cho việc khôi phục lại lịch sử giao diện chương trình khi tắt đi. Vì vậy nó được gán giá trị là số phần tử của mảng StringCollection History như trên.
* **int i**: Biến index của image trong listView1. Vì số lượng ảnh sau mỗi lần ta nhấn vào nút LoadImage sẽ tăng lên theo công thức: i = i+số ảnh trong thư mục (Không thế gán i=0 khi ta thực hiện việc nhấn vào nút Load Image vì sẽ xảy ra hiện tượng sai lệch chỉ số trong imageList1 nằm trong listView1, dẫn đến việc không hiển thị được ảnh trong thư mục như mong muốn). Nên ta muốn lấy số lượng ảnh trong thư mục được chọn, chỉ cần lưu lại giá trị i cũ và i mới rồi lấy hiệu là được kết quả như mong muốn. Và đó chính là chức năng của hai biến **int iold** và **int inew**.

## btnLoadImage\_Click

private void btnLoadImage\_Click(object sender, EventArgs e)

{

DialogResult mess = folderBrowserDialog1.ShowDialog();

iold = i;

if (mess == DialogResult.OK)

{

listView1.Items.Clear();//Lam moi listview

Dirpath = folderBrowserDialog1.SelectedPath;

label2.Text = Dirpath;//Hien thi duong dan thu muc

pictureBox1.Image = null;

string[] Files = Directory.GetFiles(Dirpath);

foreach (String fn in Files)//lay cac duong dan anh

{

if (fn.ToLower().EndsWith(".jpg") || fn.ToLower().EndsWith(".gif") ||

fn.ToLower().EndsWith(".png") || fn.ToLower().EndsWith(".bmp") ||

fn.ToLower().EndsWith(".jpeg"))

{

ListViewItem item = new ListViewItem();

//them cac item anh vao listview

Image pic;

pic = Image.FromFile(fn);

imageList1.Images.Add(pic);

item.Name = fn;

item.ImageIndex = i;

listView1.Items.Add(item);

i++;

}

}

}

inew = i;

}

* Khi ấn vào nút Load Image, biến mess được tạo ra để chắc chắn rằng người dùng ấn vào nút OK trên cửa sổ Browse For Folder mới được tạo ra. Lúc này listView1 sẽ bị xóa hết các ảnh trong thư mục và biến Dirpath sẽ nhận giá trị là đường dẫn thư mục.
* Sau đó sử dụng vòng lặp để lấy tất cả các file ảnh có dạng phổ biến như jpg, gif, png, bmp và jpeg và thêm tất cả các ảnh đó vào listView1.
* Xuyên suốt quá trình này, lable2 sẽ hiển thị tên đường dẫn thư mục, các biến iold và inew cũng được cập nhật giá trị tương ứng

## listView1\_Click

private void listView1\_Click(object sender, EventArgs e)

{

label1.Text = Resource1.History;

string NameImage = listView1.SelectedItems[0].Name;

string NameAndTime;//lay ngay gio he thong

int day = DateTime.Now.Day;

string sDay;

if (day < 10)

sDay = "0" + day.ToString();

else sDay = day.ToString();

int month = DateTime.Now.Month;

string sMonth;

if (month < 10)

sMonth = "0" + month.ToString();

else sMonth = month.ToString();

int year = DateTime.Now.Year;

int hour = DateTime.Now.Hour;

string sHour;

if (hour < 10)

sHour = "0" + hour.ToString();

else sHour = hour.ToString();

int minute = DateTime.Now.Minute;

string sMinute;

if (minute < 10)

sMinute = "0" + minute.ToString();

else sMinute = minute.ToString();

int second = DateTime.Now.Second;

string sSecond;

if (second < 10)

sSecond = "0" + second.ToString();

else sSecond = second.ToString();

NameAndTime = sHour + ":" + sMinute + ":" + sSecond + "\_" + sDay +"/"+ sMonth +"/"+ year.ToString()+" "+NameImage;

//ten item gom thoi gian he thong va duong dan anh

listBox1.Items.Add(NameAndTime);//them item vao listbox

stringCol.Add(NameAndTime);//luu lich su

listBox1.SelectedIndex =ii;//tro den item moi dc them vao

ii++;

pictureBox1.Image = Image.FromFile(NameImage);//hien thi anh vao picturebox

textBox1.Text = NameImage;//dua duong dan ra textbox

textBox1.Enabled = true;

}

* Khi thực hiện việc chọn ảnh, chương trình sẽ gọi đến hàm **listView1\_Click**.
* Lúc này các câu lệnh lấy ngày giờ hệ thống được thực hiện và listBox1 cùng với stringCol sẽ được thêm item là một string gồm ngày giờ hệ thống cùng đường dẫn ảnh.
* Đông thời listBox1 sẽ chỉ đến item mới được thêm vào và pictureBox1 cũng hiển thị hình ảnh đó.

## ImageShow\_Load

private void ImageShow\_Load(object sender, EventArgs e)

{

Dirpath = Properties.Settings.Default.RecentFolder;

label2.Text = Dirpath;

string[] Files = Directory.GetFiles(Dirpath);

foreach (String fn in Files)//lay cac duong dan anh

{

if (fn.ToLower().EndsWith(".jpg") || fn.ToLower().EndsWith(".gif") ||

fn.ToLower().EndsWith(".png") || fn.ToLower().EndsWith(".bmp") ||

fn.ToLower().EndsWith(".jpeg"))

{

ListViewItem item = new ListViewItem();//them cac item anh vao listview

Image pic;

pic = Image.FromFile(fn);

imageList1.Images.Add(pic);

item.Name = fn;

item.ImageIndex = i;

listView1.Items.Add(item);

i++;

}

}

inew = i; iold = 0;

//tai lai lich su xem luc truoc

for (int j=0;j<Properties.Settings.Default.History.Count;j++)

{

listBox1.Items.Add(Properties.Settings.Default.History[j]);

}

listBox1.SelectedIndex = Properties.Settings.Default.History.Count - 1;

//chi den hinh anh dc xem gan nhat

if(listBox1.Items.Count>0)

{

label1.Text = Resource1.History;

string ListBoxItemName = listBox1.SelectedItem.ToString();

int strindex = 0;//loai bo ngay gio trong cac item cua listbox

char temp = '.';

while (temp != ' ')

{

temp = ListBoxItemName[strindex];

strindex++;

}

string lbin = ListBoxItemName.Remove(0, strindex - 1);

textBox1.Text = lbin;//dua duong dan ra textbox

textBox1.Enabled = true;

pictureBox1.ImageLocation = lbin;//hien thi anh

}

}

* Hàm **ImageShow\_Load** sẽ được gọi đến khi chương trình được khởi động. Lúc này Dirpath sẽ được nạp giá trị là đương dẫn thư mục gần nhất khi chương trình bị tắt. Sau đó sẽ tải tất cả các ảnh trong thư mục vào listView1 như hàm **btnLoadImage\_Click** đã làm.
* Tiếp theo chương trình sẽ tải lại tất cả các lịch sử xem lúc trước.
* listBox1 sẽ chỉ vào item mở gần nhất và pictureBox1 sẽ hiển thị hình ảnh đó

## ImageShow\_FormClosing

private void ImageShow\_FormClosing(object sender, FormClosingEventArgs e)

{

Properties.Settings.Default.RecentFolder = Dirpath;

//luu lai duong dan thu muc truoc do

Properties.Settings.Default.History = stringCol;//luu lai lich su xem

Properties.Settings.Default.Save();

}

* Hàm **ImageShow\_FormClosing** sẽ được gọi đến khi chương trình đóng. Trong hàm này biến RecentFolder sẽ nhận giá trị đường dẫn thư mục gần nhất và History sẽ lưu lại lịch sử xem ảnh trong listBox1

## PictureBox1\_MouseHover

private void pictureBox1\_MouseHover(object sender, EventArgs e)

{

string Info = "";//thong tin tooltip cua anh

if (pictureBox1.Image != null)

{

FileInfo FI = new FileInfo(textBox1.Text.ToString());

Info = Resource1.Name\_Image + ": " + FI.Name;

Info = Info + "\n" + Resource1.Dimensions\_Image+": " + pictureBox1.Image.Width.ToString()+ "x" +

pictureBox1.Image.Height.ToString();

float lenImage = (float)FI.Length / 1024;

Info = Info + "\n" + Resource1.Size\_Image + ": " + lenImage + "Kb";

Info = Info + "\n" + Resource1.CreationTime\_Image + ": " + FI.CreationTime;

toolTip2.SetToolTip(pictureBox1, Info);

}

else

{

toolTip2.SetToolTip(pictureBox1, Resource1.PicBoxToolTip);

}

}

* Khi người dùng di chuột đến pictureBox1, hàm **pictureBox1\_MouseHover** sẽ được thực hiện.
* **pictureBox1\_MouseHover** có chức năng chính là hiển thị thông tin về ảnh bao gồm tên ảnh, kích thước, dung lượng và ngày giờ tạo file.
* Sử dụng FileInfo để thực hiện điều trên.

## Các hàm khác

Ngoài ra chương trình còn sử dụng một số hàm phụ trợ để hiển thị tool tip cho nút, cho đường dẫn thư mục,… như **btnLoadImage\_MouseHover, textBox1\_MouseHover, listBox1\_MouseHover**

# Kết Luận

Phần mềm đã đạt được một số mục tiêu nhất định, tuy nhiên vẫn còn nhiều hạn chế và khuyết điểm

Ưu điểm:

* Hoàn thành tất cả các yêu cầu được giao
* Bổ sung thêm vài chức năng phụ trợ (ToolTip cho tất cả control, số lượng ảnh trong thư mục,…)

Hạn chế:

* Giao diện chương trình chưa tối ưu