**Đồ họa máy tính – lab-06**

# Xử lý ảnh với OpenCV-Xoay ảnh-Điều chỉnh độ sáng

**🗹 Bài tập cá nhân** 🞎 Bài tập nhóm

🞎 Bài tập tự luyện **🗹 Bài tập cần nộp**

Mục lục:

[1. Qui định về việc nộp bài: 2](file:///D:\2019\DHMT\Lab-04.docx#_Toc39830833)

[1.1 Thời hạn nộp bài: 2](file:///D:\2019\DHMT\Lab-04.docx#_Toc39830834)

[1.2 Cách nộp: nộp qua hệ thống courses.uit.edu.vn 2](file:///D:\2019\DHMT\Lab-04.docx#_Toc39830835)

[1.3 Bài làm được nén lại thành một tập tin ( .zip hoặc .rar ) MSSV.zip ( hoặc .rar ) 2](file:///D:\2019\DHMT\Lab-04.docx#_Toc39830836)

[2. Nội dung: 2](file:///D:\2019\DHMT\Lab-04.docx#_Toc39830837)

# Qui định về việc nộp bài:

## Thời hạn nộp bài:

* Sau buổi thực hành – 01 tuần

## Cách nộp: nộp qua hệ thống courses.uit.edu.vn

## Bài làm được nén lại thành một tập tin ( .zip hoặc .rar ) MSSV.zip ( hoặc .rar )

Bài nộp bao gồm 2 phần :

* Source : source code và thư viện sử dụng
* Release : chứa file thực thi exe.

# Nội dung:

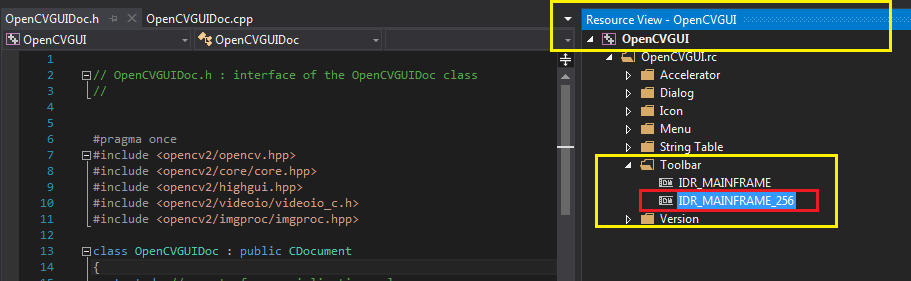
## Xoay ảnh:

### ****Tạo icon chức năng****

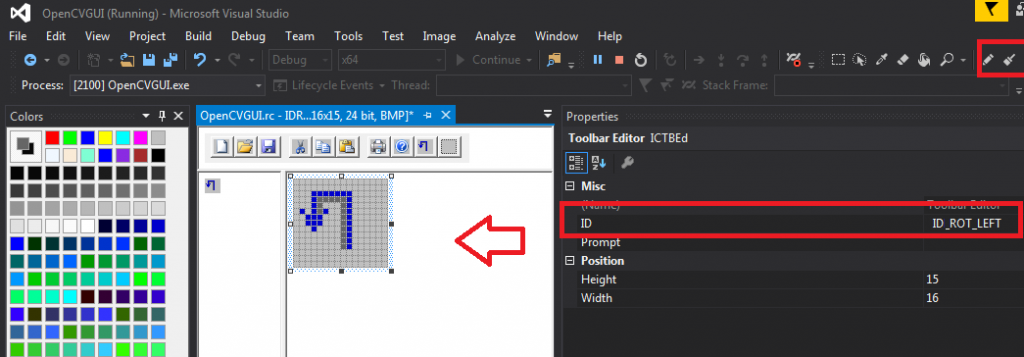
Trên thanh toolbar tạo icon để thực hiện chức năng, ấn vào icon thì quay phải 1 góc 90 độ.

Các bạn mở Resource view lên rồi mởi project ra,

sau đó mở folder toolbar và chọn **IDR\_MAINFRAME\_256**

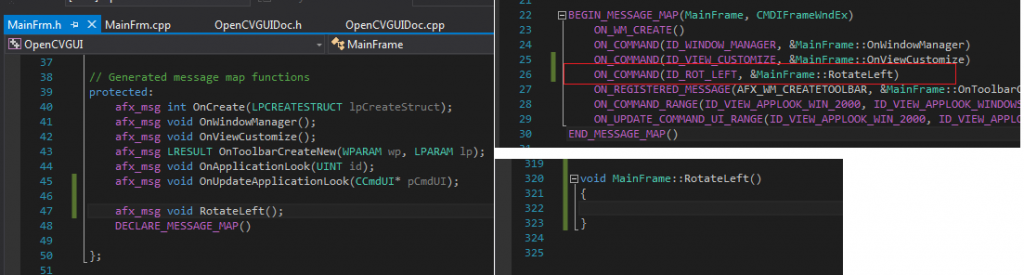


Sau đó các bạn tạo 1 icon đơn giản như sau, đặt tên ID = ID\_ROT\_LEFT



### ****Tạo hàm xử lý sự kiện****

Tạo hàm xử lý khi click vào button, và ánh xạ thông điệp.



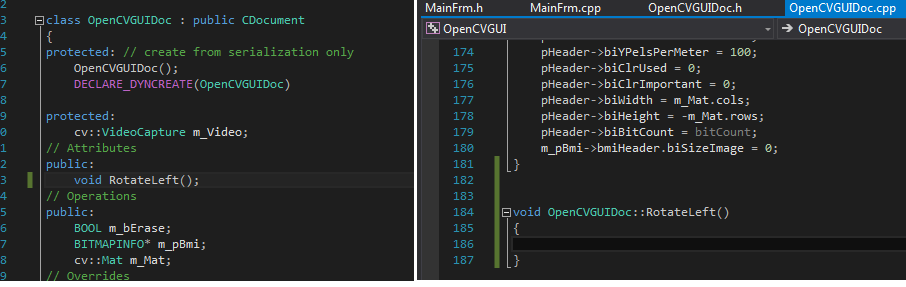
Đến đây các bạn có thể build và chạy thử, nếu thành công icon rot left sẽ được sáng lên

### ****Viết hàm xử lý chức năng xoay góc****

Chúng ta viết tiếp hàm RotateLeft tại lớp Doc để thực hiện xử lý chính.

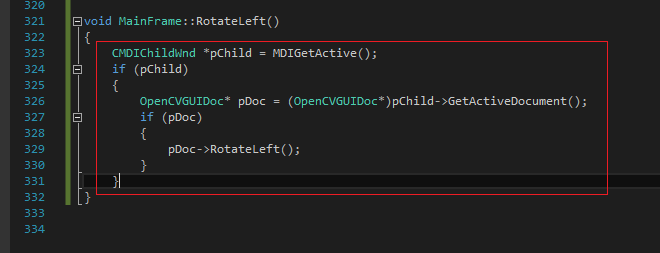
Lớp mainframe chỉ thực hiện nhận xử lý sự kiện từ toolbar

Các bạn viết tương tự như lớp main frame như minh họa sau



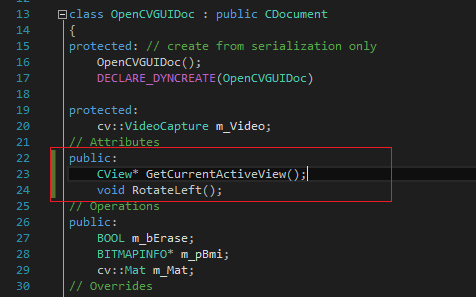
Sau đó tại lớp Mainframe sẽ gọi tới hàm của lớp Doc như sau

Nhớ gọi include lớp Doc bên mainframe.h



**Thực hiện xử lý coding quay góc cho ảnh bên hàm RotateLeft bên lớp document.**

Thực hiện tạo hàm lấy view để update view khi xử lý ảnh



Thực hiện định nghĩa hàm GetCurrentActiveView()

Đoạn code xử lý quay góc 90 sang theo chiều bên trái (ngược chiều kim đồng hồ) cho ảnh

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | void OpenCVGUIDoc::RotateLeft()  {    cv::Mat src = m\_Mat;    cv::Mat dst;    cv::Point2f pt(src.cols / 2., src.rows / 2.);    cv::Mat r = getRotationMatrix2D(pt, 90.0, 1.0);    warpAffine(src, dst, r, cv::Size(src.cols, src.rows));    m\_Mat.release();    m\_Mat = dst.clone();    dst.release();    if (GetCurrentActiveView())    {      GetCurrentActiveView()->InvalidateRect(NULL);    }  } |

Nguyên lý quay góc 90 độ đó là:

Xác định điểm gốc để quay, điểm quay là được tính là điểm trung tâm của tấm ảnh

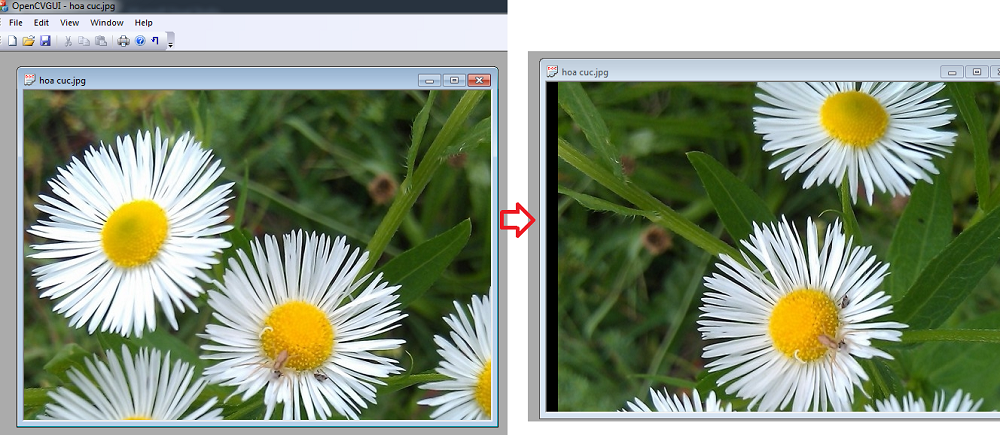
==> quay xung quanh tấm ảnh

Sau đó sử dụng hàm getRotationMatrix2D

Đối số truyền vào là :  điểm quay, giá trị góc quay, và tỉ lệ (để mặc định là 1.0)

Sau đó lấy kết quả được trả ra và gán lại vào biến m\_Mat  rồi chuyển sang update view để hiện thị kết quả.

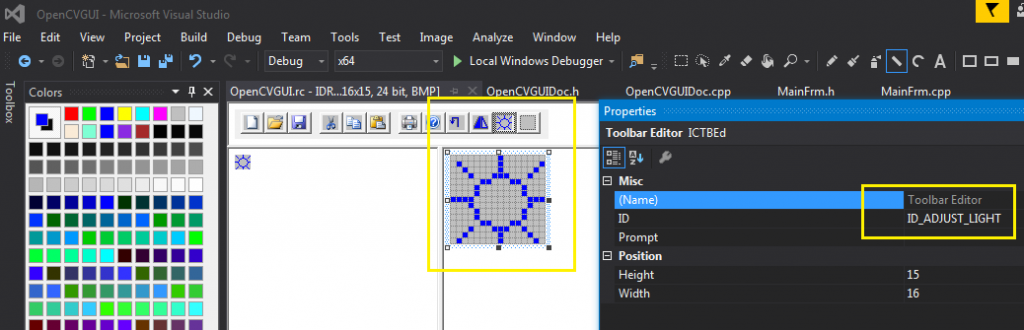
Ok, đến đây các bạn có thể build và test thử chương trình, kêt quả sẽ được như sau



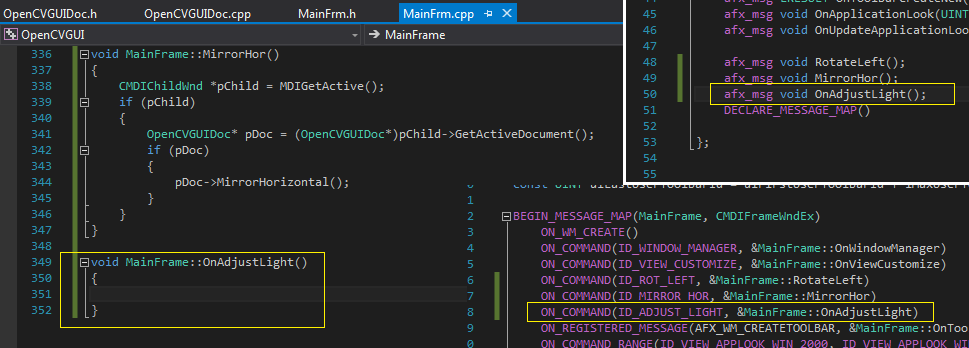
## Điều chỉnh độ sáng

### ****Tạo một icon trên toolbar cho chức năng****

Cũng tương tự như những bài trước, các bạn hãy tạo 1 icon cho chức năng điều chỉnh độ sáng



Sau đó cũng viết hàm xử lý chức năng tại lớp mainframe tương tự như các bài trước



### ****Xây dựng dialog điều chỉnh giá trị sáng tối.****

Không giống như những bài trước, điều chỉnh sáng tối cần có giá trị thông số được nhập vào

điều chỉnh sáng là một giá trị dương

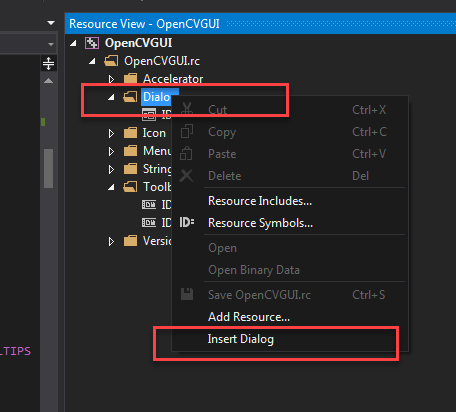
còn điều chỉnh tối là một giá trị âm

Giá trị gốc ban đầu sẽ là 0.0

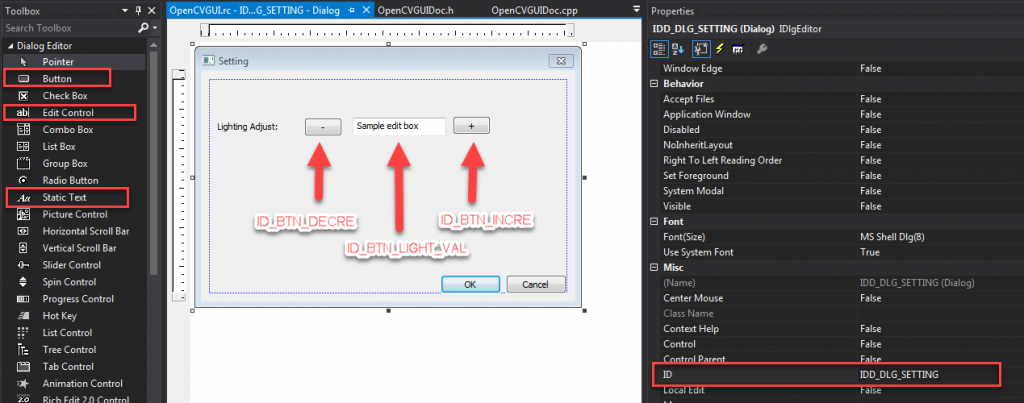
Do đó chúng ta sẽ thiết kế một dialog như sau

**a. Tạo một dialog trong folder resource của project**

Các bạn click chuột phải và chọn insert dialog như sau



Chúng ta sẽ thiết kế một dialog dạng như sau



Về cách thiết kế một dialog trong lập trình MFC,

có thể nhiều bạn chưa biết vì chưa học về mảng lập trình này

+ Các bạn đặt tên ID của dialog là : IDD\_DLG\_SETTING (để sau này còn dùng thêm các tham số khác cho xử lý ảnh)

+ Tạo 2 button tăng giảm giá trị với ID minh họa trên ảnh

+ Tạo một label và một edit box có tên ID như ví dụ trên ảnh

Như vậy chức năng của dialog đó là.

Khi ấn vào icon trên toolbar, dialog sẽ được gọi, với giá trị mặc định ban đầu là 0.0

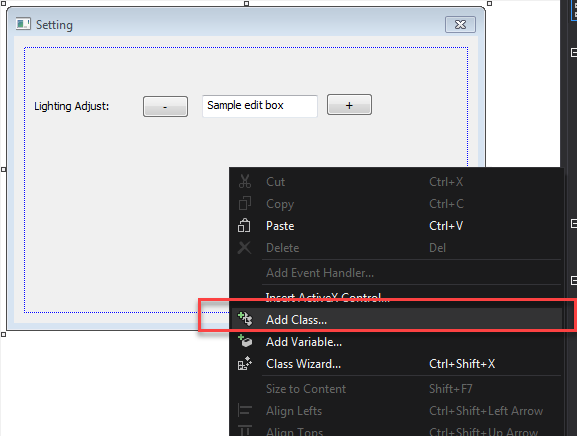
Nếu các bạn ấn vào button +, giá trị sẽ tăng têm 10, sau đó ấn ok thi các bạn sẽ thấy ảnh được sáng hơn

Ngược lại các bạn ấn vào button -, giá trị giảm thêm 10, và sau đó ấn ok thì ảnh sẽ bị tối dần đi.

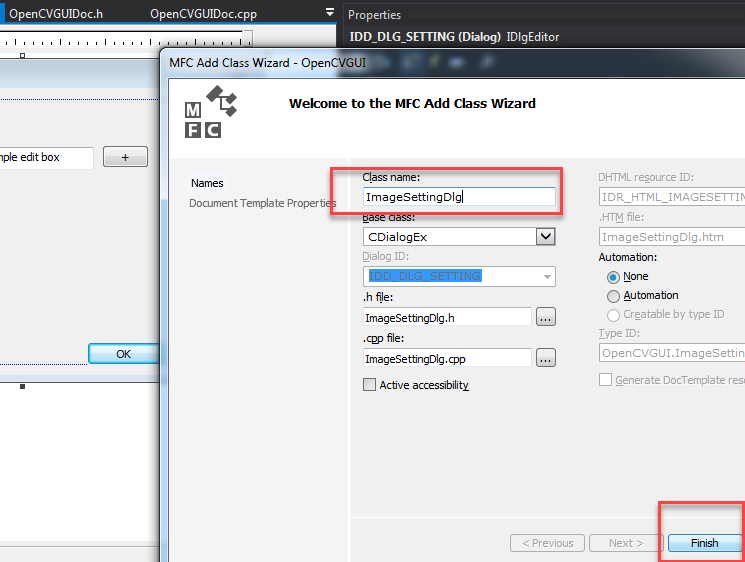
Ngoài việc tăng giảm giá trị theo button, các bạn cũng có thể nhập trực tiếp.

**b. Xây dựng lớp quản lý cho dialog, và các biến quản lý cho các control**

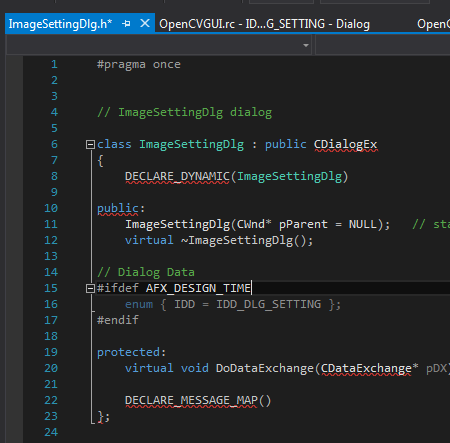
Chuột phải vào dialog, các bạn chọn item add class



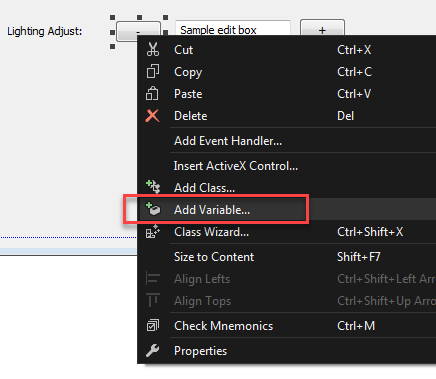
Sau đó đặt tên lớp ví dụ như sau, rồi bấm finish.



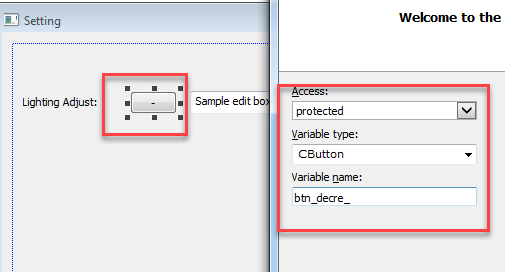
Các bạn sẽ lớp quản lý cho dialog được xuất hiện



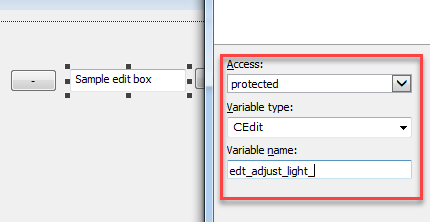
Trở lại phần dialog, tiếp tục tạo biến quản lý cho các control



Các bạn add btn\_decre cho button –



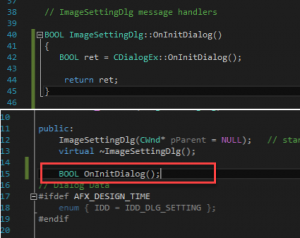
Tương tự add biến edt\_adjust\_light\_ cho control edit box



Và sau đó add tương tự biến btn\_incre\_ cho button +

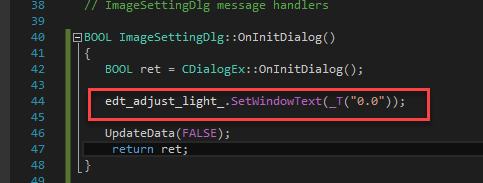
***c. Tạo các hàm xử lý cho lớp dialog***

Viết một hàm BOOL OnIntDialog () để thiết lập các giá trị ban đầu

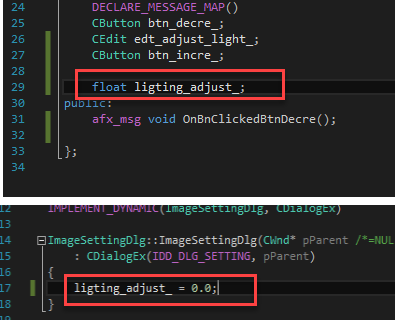


Hàm này là hàm được tự động gọi khi dialog được show nên không cần khai báo ánh xạ thông điệp

Trong hàm này chúng ta thiết lập giá trị ban đầu cho edit box là 0.0 như sau.



Chúng ta khai báo một biến quản lý giá trị trong edit box, gán giá trị khởi tạo cũng là 0.0

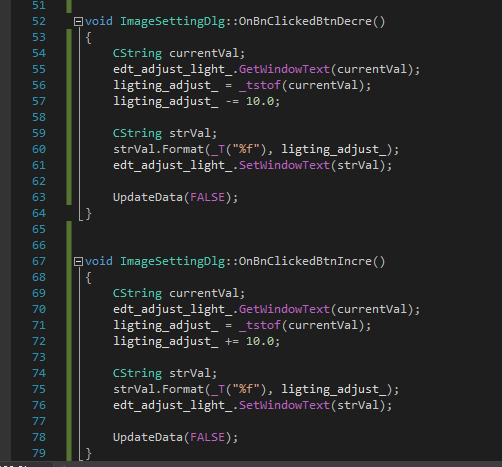


Thực hiện xử lý cho button –

Click double vào button – , chương trình sẽ tự động tạo ra một hàm xử lý cho riêng nó,

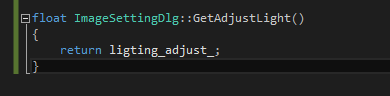
các bạn tăng giá trị của biến lên

Và sét lại giá trị vào edit box



Tương tự như vậy với button +, các bạn tăng giá trị lên.

Viết tiếp một hàm trả ra giá trị trên edit box()

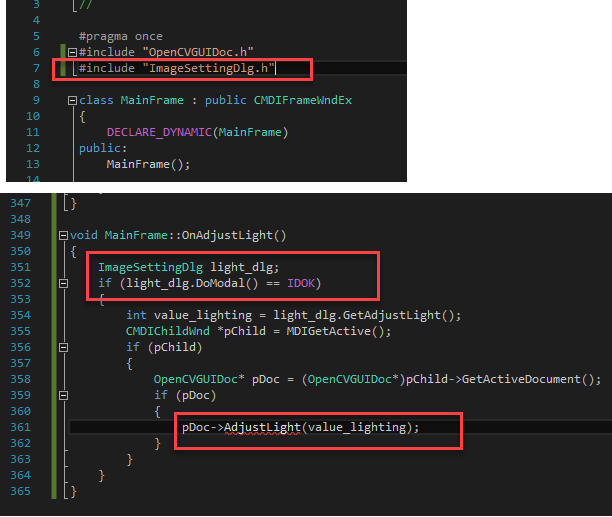


Ok, đến đây là xong phần xử lý cho dialog, các bạn build chương trình và đảm bảo không có lỗi, nếu có lỗi hay xem lỗi gì và sửa

### ****Thực hiện xử lý sáng tối cho ảnh.****

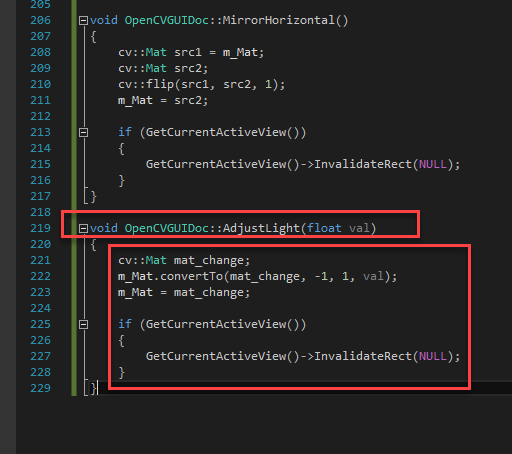
Các bạn quay trở lại lớp mainframe và tại hàm AdjustLight

Chúng ta tiến hành gọi dialog vừa thiết kế ở trên



Tất nhiên hàm AdjustLight của lớp Document là chưa có, và giờ chúng ta sẽ định nghĩa hàm đó

Các bạn sẽ định nghĩa một hàm AdjustLight bên lớp Document và xử lý độ sáng cho ảnh như sau.



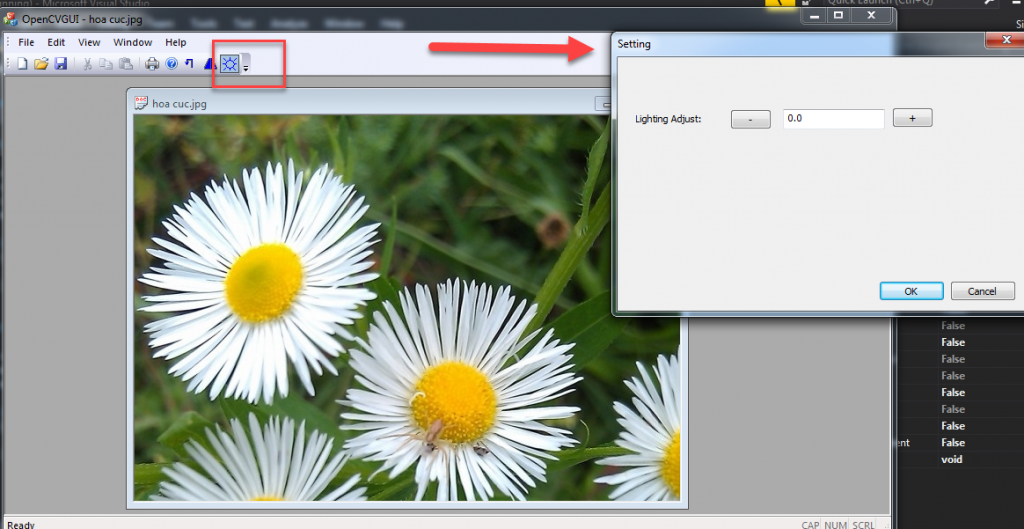
Các bạn nhớ khởi tạo cả bên file .h

Về cơ bản việc xử lý độ sáng của ảnh sẽ được thông qua hàm convertTo của lớp cv::Mat

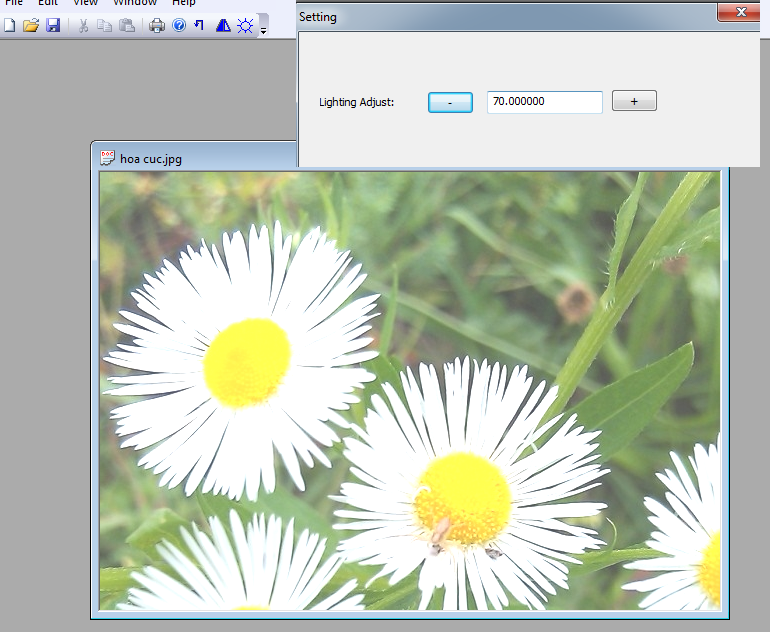
Đối số cuối cùng của hàm này chính là đối số dành cho độ sáng tối, và chúng ta truyền giá trị setting trên dialog vào đối số này.

Các bạn build và chạy thử chương trình.

Khi bấm vào icon mặt trời, sẽ thấy dialog được show lên như sau



Kết quả test với giá trị +70



Và kết quả test với -50

