

Khi đọc qua tài liệu này, nếu phát hiện sai sót hoặc nội dung kém chất lượng xin hãy thông báo để chúng tôi sửa chữa hoặc thay thế bằng một tài liệu cùng chủ đề của tác giả khác. Tài liệu này bao gồm nhiều tài liệu nhỏ có cùng chủ đề bên trong nó. Phần nội dung bạn cần có thể nằm ở giữa hoặc ở cuối tài liệu này, hãy sử dụng chức năng Search để tìm chúng.

Bạn có thể tham khảo nguồn tài liệu được dịch từ tiếng Anh tại đây:

http://mientayvn.com/Tai_lieu_da_dich.html

Thông tin liên hệ:

Yahoo mail: thanhlam1910_2006@yahoo.com

Gmail: frbwrthes@gmail.com

Theo yêu cầu của khách hàng, trong một năm qua, chúng tôi đã dịch qua 16 môn học, 34 cuốn sách, 43 bài báo, 5 sổ tay (chưa tính các tài liệu từ năm 2010 trở về trước) Xem ở đây	
DỊCH VỤ DỊCH TIẾNG ANH CHUYÊN NGÀNH NHANH NHẤT VÀ CHÍNH XÁC NHẤT	Chỉ sau một lần liên lạc, việc dịch được tiến hành
	Giá cả: có thể giảm đến 10 nghìn/1 trang
	Chất lượng: <u>Tạo dựng niềm tin cho khách hàng bằng công nghệ 1.</u> Bạn thấy được toàn bộ bản dịch; 2. Bạn đánh giá chất lượng. 3. Bạn quyết định thanh toán.

LẬP TRÌNH WINDOWS

Giới thiệu tổng quan về học phần lập trình Windows

Trần Ngọc Bảo
Email: tnbao.dhsp@gmail.com

Mục tiêu

- **Môn học cung cấp cho sinh viên kiến thức về phương pháp lập trình trên môi trường Windows**
 - Cơ chế quản lý chương trình và xử lý thông điệp trên MS Windows
 - Lập trình giao diện đồ họa - người dùng (GUI)
 - Cơ chế giao tiếp thiết bị đồ họa (GDI)
 - Cơ chế quản lý bộ nhớ, tập tin.
 - Môn học này trình bày phương pháp lập trình API trên Windows 32 và **kỹ thuật lập trình MFC với ngôn ngữ minh họa là Visual C++ 6.0**

Mục tiêu

- **Tạo cho sinh viên một nền tảng trong việc tiếp thu các ngôn ngữ lập trình cấp cao khác trên Windows như: Visual Basic, Delphi, C++ Builder, C#, Visual basic.Net, J Builder, Java...**
cuu duong than cong. com
- **Tạo cho sinh viên một kiến thức cơ bản để có thể tự nghiên cứu tiếp các vấn đề khác trong việc lập trình trên Windows như: DDE, OLE, giao tiếp với thiết bị ngoại vi, ...**
cuu duong than cong. com

Kiến thức - kỹ năng

- **Kiến thức**

- Có kiến thức cơ bản về lập trình trong môi trường Windows
- Lập trình giao diện người dùng
- Phương pháp trình giao tiếp hệ quản trị cơ sở dữ liệu
- Lập trình sử dụng DLL, Component,...

- **Kỹ năng**

- Sử dụng được công cụ Visual C++
 - Xây dựng được ứng dụng đồ họa
 - Xây dựng được ứng dụng quản lý tương tác với CSDL SQL Server 2000
 - Xây dựng được ứng dụng tích hợp một số Third-Party Component
- Sử dụng được một số công cụ tạo Report như: Crystal Report, C1 Report.

Một số ứng dụng minh họa

- Một số ứng dụng lập trình cơ bản như:
 - Máy tính bỏ túi
 - Điện thoại
 - Chương trình nghe nhạc
- Ứng dụng biểu diễn các đối tượng hình học
 - Paint
 - Xử lý ảnh

Một số ứng dụng minh họa

- **Ứng dụng mô phỏng thuật toán**
 - Cấu trúc dữ liệu: mảng, danh sách liên kết, cây nhị phân,...
 - Lý thuyết đồ thị
- **Một số ứng dụng cơ sở dữ liệu như:**
 - Quản lý học sinh, Quản lý sinh viên, Quản lý nhà thuốc, Bưu điện
 - Quản lý dự án (Project Management)
 - Quản lý luồng công việc (Workflow)
 - Quản lý nhân sự
 -

Đề cương môn học (5đvht= 75t)

- Chương 1 Khái niệm về lập trình Windows
- Chương 2 Kỹ thuật lập trình Visual C++ (MFC)
- Chương 3 Quản lý bộ nhớ
- Chương 4 Thư viện đồ họa GDI
- Chương 5 Thư viện liên kết động
- Chương 6 Lập trình CSDL ADO – SQL Server
- Chương 7 Lập trình đa tiểu trình

cuu duong than cong. com

Công cụ thực hành

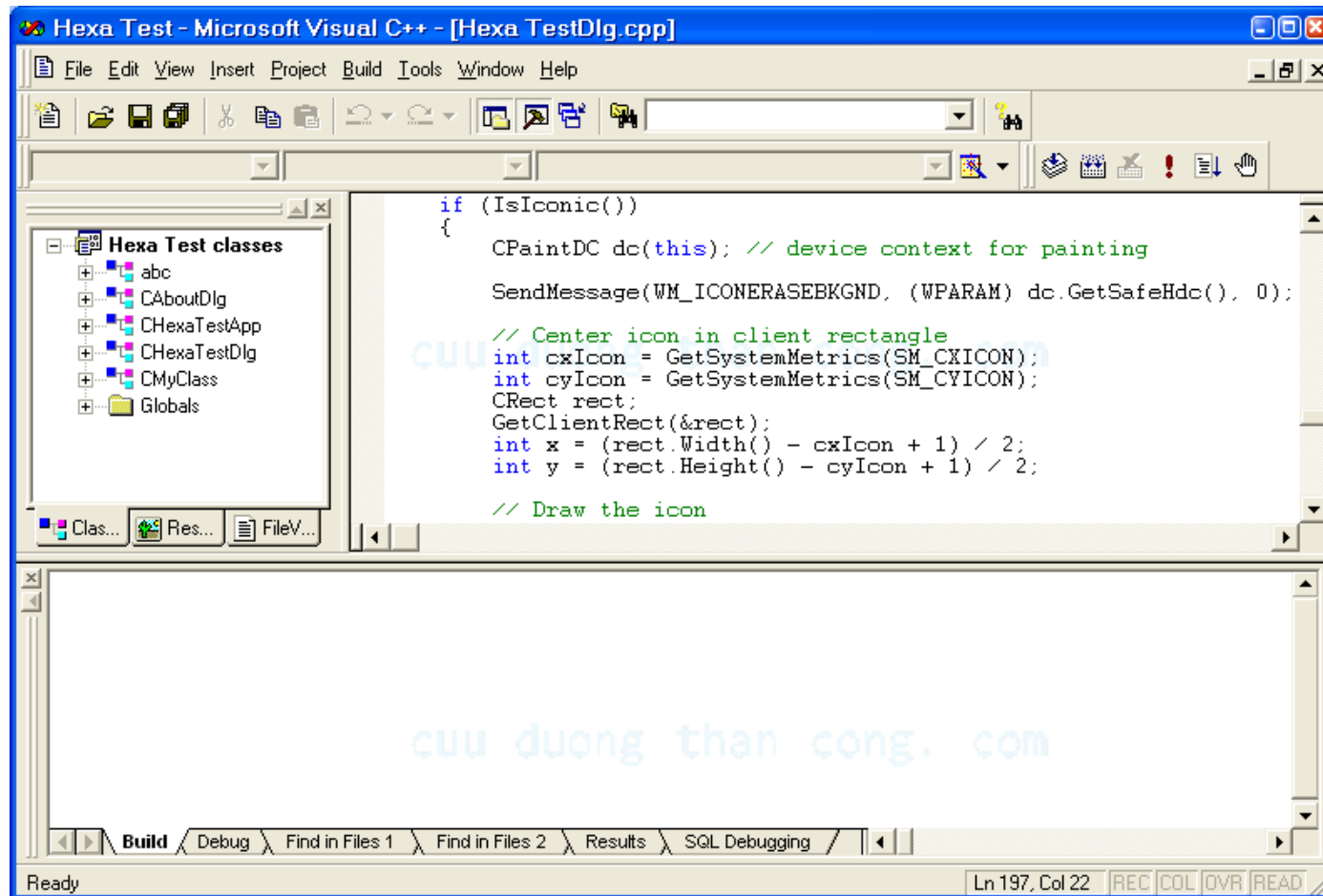
- Visual Studio 6.0
 - Visual C++ 6.0
- Visual Studio.Net 2003
 - Visual C++ Projects
- Visual Studio.Net 2005

cuu duong than cong. com

cuu duong than cong. com

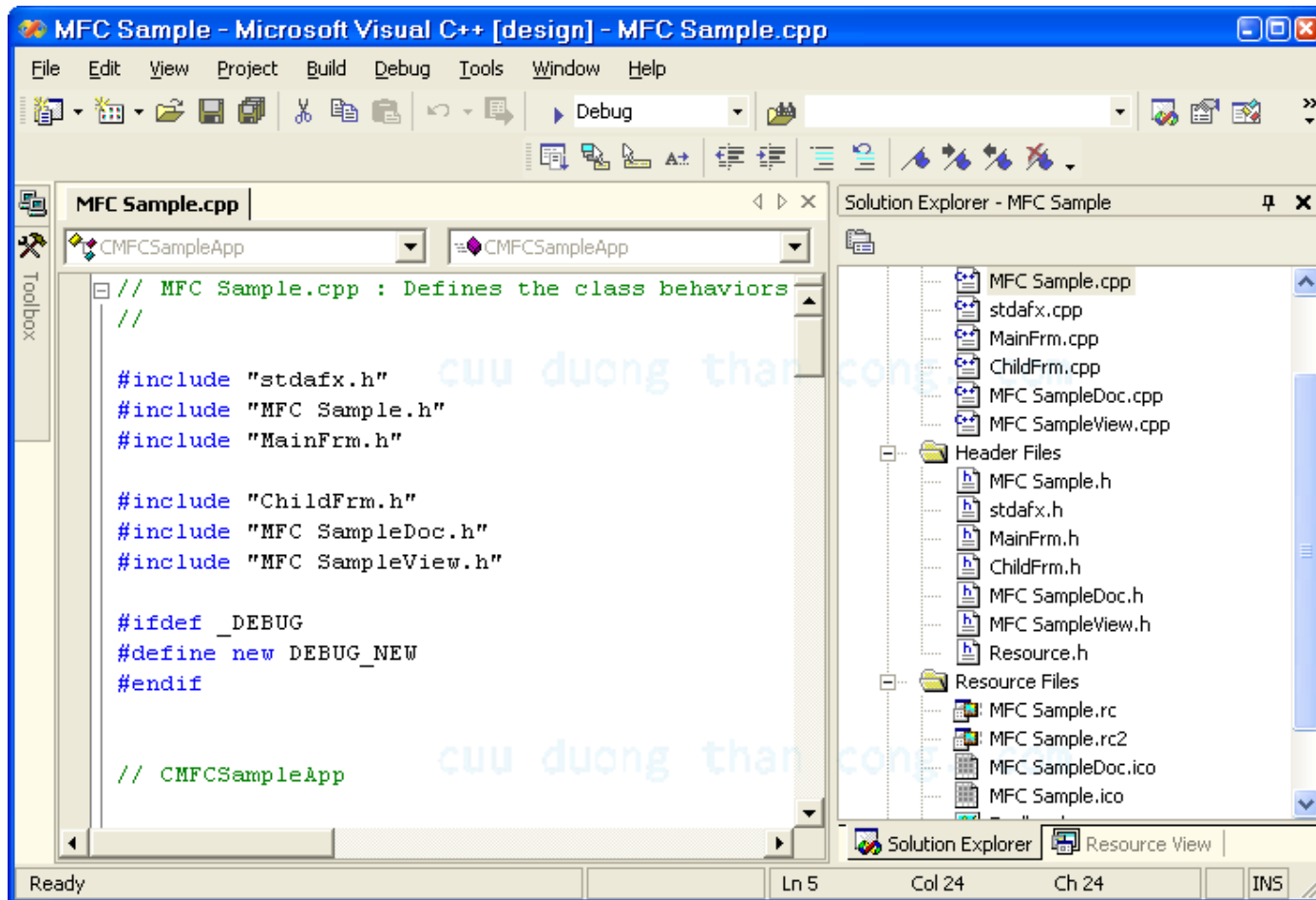
Công cụ thực hành

- Visual Studio 6.0 - Visual C++ 6.0



Công cụ thực hành

- Visual Studio.Net 2003 - Visual C++ Projects



Thực hành và thi kết thúc học phần

- Bài tập lớp (Project):
 - Hình thức: bắt buộc.
 - Số lượng: 2 bài, mỗi bài 1.5đ
- Thi kết thúc học phần
 - Hình thức: thi thực hành trực tiếp trên máy
 - Thời gian: 90 – 120 phút
 - Điểm thi: 7đ

cuu duong than cong. com

Bài tập project

- Bài tập project 1
 - Nội dung: Lập trình căn bản, xây dựng ứng dụng tương tự như: Paint, Calculator, Telephone,...
 - Kiến thức liên quan: Dialogs, Menu, Toolbar, GDI,...
- Bài tập project 2
 - Nội dung: Lập trình cơ sở dữ liệu
 - Cơ sở dữ liệu: Access 2003 hoặc SQL Server 2000
 - Report: Crystal Report, C1Report,...

Nộp bài tập project

- Bài tập project 1
 - Nộp vào giữa học kỳ
 - Sinh viên nộp bài cho lớp trưởng
- Bài tập project 2
 - Nộp vào cuối học kỳ, trước khi thi thực hành 1 tuần.
 - Sinh viên nộp bài cho lớp trưởng

cuu duong than cong. com

Thực hành và thi kết thúc học phần

- Bài tập miễn thi
 - Hình thức: không bắt buộc.
 - Là một dạng bài tập project nâng cao, liên quan đến một chủ đề do GV đề xuất, sinh viên tự thực hiện.
 - Mỗi sinh viên làm một đề tài.
 - Sinh viên làm bài tập miễn thi sẽ không phải làm 2 bài tập project bắt buộc trên.
 - Trong trường hợp sinh viên không được miễn thi, điểm bài tập miễn thi sẽ được tính là điểm bài tập project (tối đa 3đ).

Bài tập miễn thi

- Nội dung
 - Báo cáo viết mô tả ứng dụng theo mẫu qui định
 - CD chương trình được tổ chức theo mẫu qui định
- Nộp bài
 - Thời gian: sinh viên nộp bài vào cuối học kỳ, trước khi thi thực hành 2 tuần.
 - Nộp bài trực tiếp cho giáo viên lý thuyết

cuu duong than cong. com

Tài liệu tham khảo

- [1] Đặng Văn Đức, Lê Quốc Hưng (1999), Lập trình Windows bằng Visual C++, Nhà xuất bản giáo dục.
- [2] Advanced MFC programming
- [3] Jeff Prosise (1999), Programming Windows with MFC 2nd Edition, Microsoft Press.
- [4] Microsoft Press (1999), Desktop application with MS Visual C++ 6.0 MCSD Training Kit.
- [5] David J. Kruglinski (1998), Programming Microsoft Visual C++ Fifth Edition, Microsoft Press.
- [6] Charles Petzold (1998), Programming Windows Fifth Edition, Microsoft Press.
- [7] Michael Pickens and Scot Wingo (1998), Microsoft Visual C++ MFC FAQ v5.6, Microsoft Press.
- [8] MSDN
- [9] <http://msdn.microsoft.com>
- [10] <http://www.codeguru.com>
- [11] <http://codeproject.com>
- [12] <http://expert-exchange.com>

MỘT SỐ KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH WINDOWS

- Một số khái niệm cơ bản
- Lập trình sự kiện – thông điệp
- Mô hình lập trình Windows

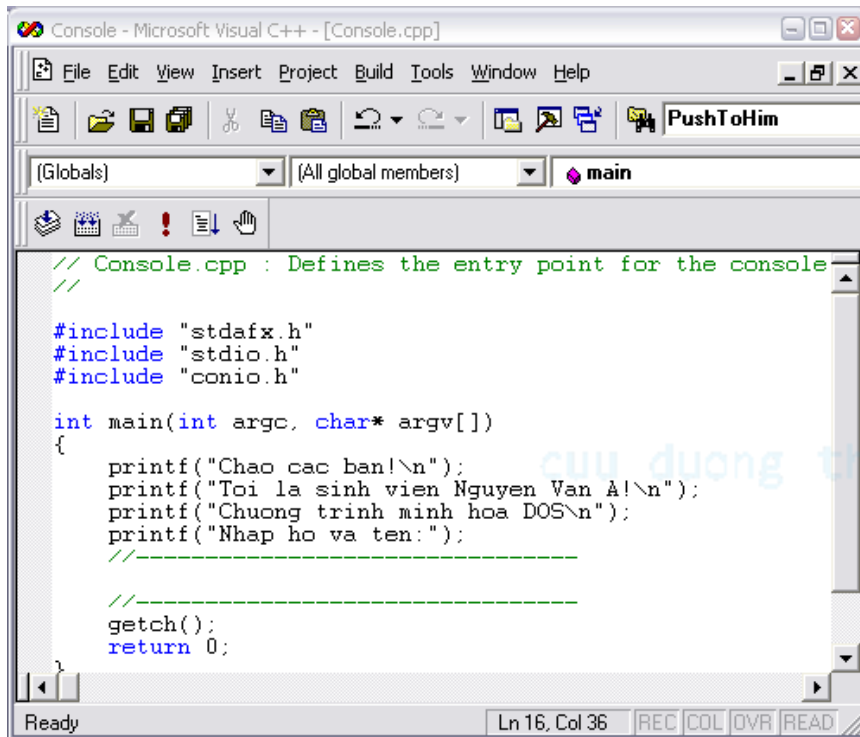
NỘI DUNG

- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Event driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

NỘI DUNG

- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Even driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

Một số điểm khác biệt giữa lập trình Windows & DOS

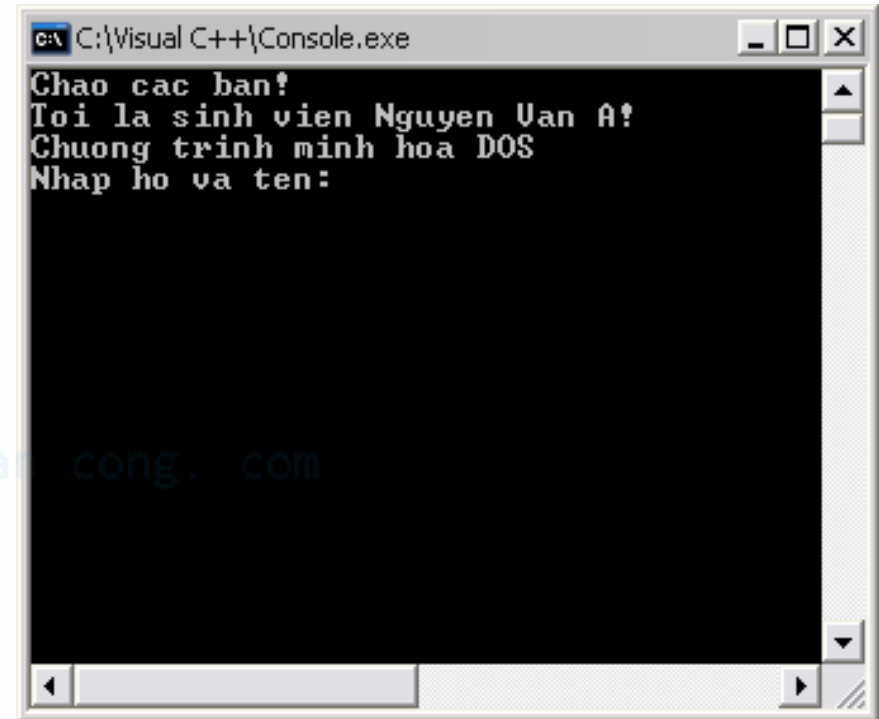


The screenshot shows the Microsoft Visual C++ IDE with the file 'Console.cpp' open. The code defines the entry point for the console application. It includes standard headers and uses printf to output a series of messages in Vietnamese. The status bar at the bottom indicates 'Ln 16, Col 36'.

```
// Console.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include "stdio.h"
#include "conio.h"

int main(int argc, char* argv[])
{
    printf("Chao cac ban!\n");
    printf("Toi la sinh vien Nguyen Van A!\n");
    printf("Chuong trinh minh hoa DOS\n");
    printf("Nhap ho va ten:");
    //-----

    //-----
    getch();
    return 0;
}
```

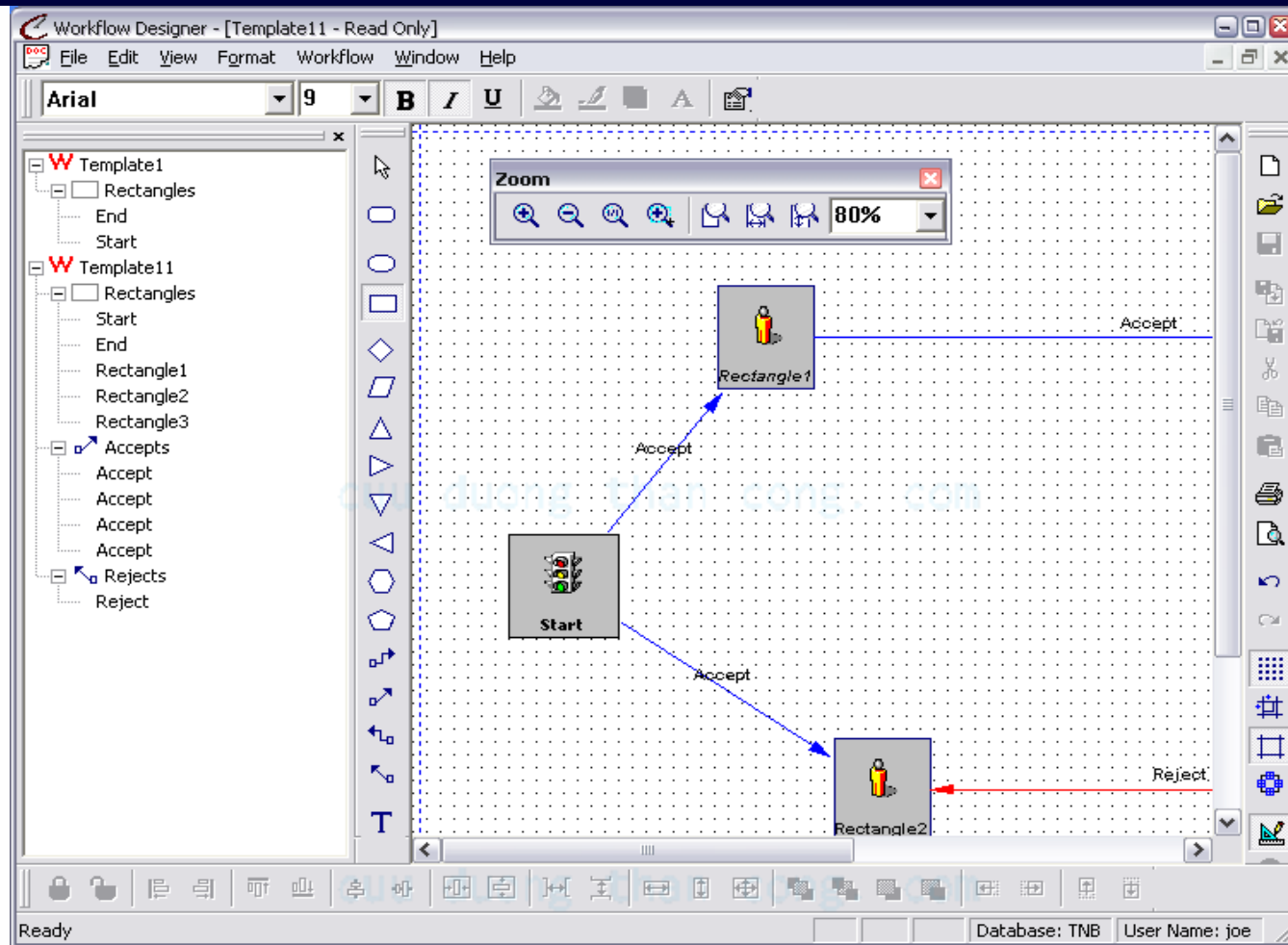


The screenshot shows a DOS command prompt window titled 'C:\Visual C++\Console.exe'. The output of the program is displayed in the window, matching the printf statements in the source code.

```
C:\Visual C++\Console.exe
Chao cac ban!
Toi la sinh vien Nguyen Van A!
Chuong trinh minh hoa DOS
Nhap ho va ten:
```

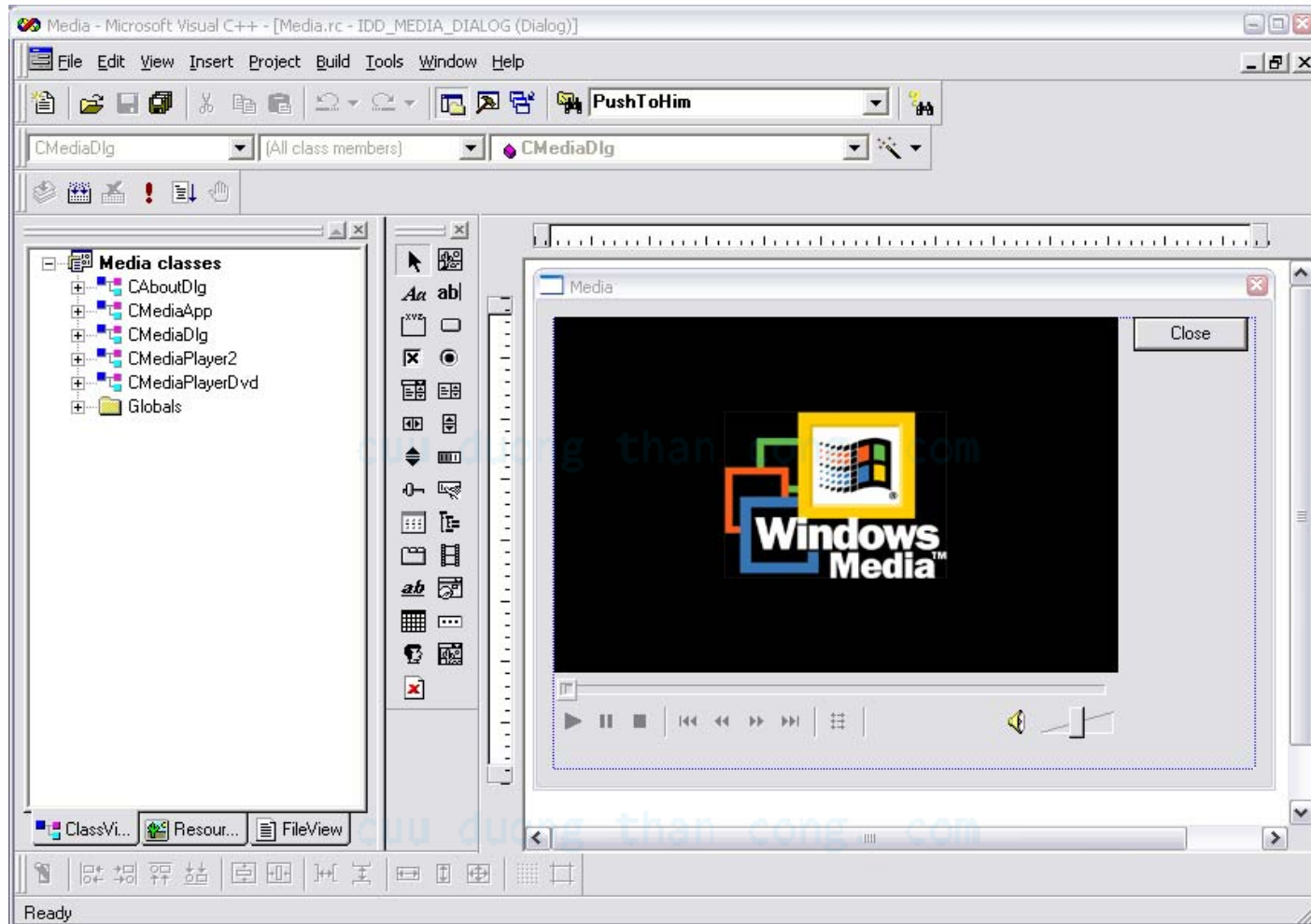
Chương trình minh họa cơ chế thực hiện tuần tự
trong môi trường DOS

Một số điểm khác biệt giữa lập trình Windows & DOS



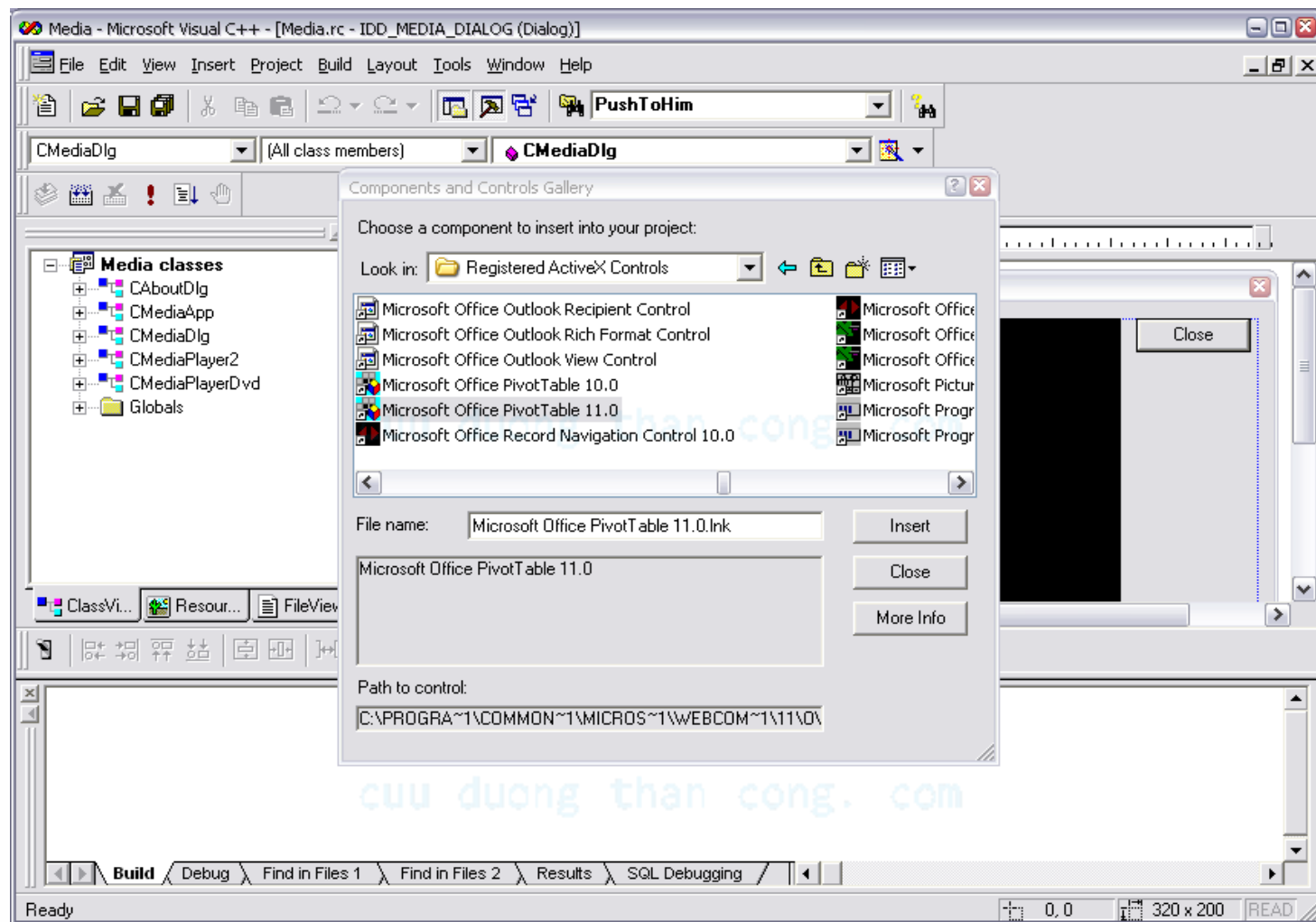
Chương trình minh họa trong môi trường Window

Một số điểm khác biệt giữa lập trình Windows & DOS



Tích hợp sẵn Multimedia

Một số điểm khác biệt giữa lập trình Windows & DOS



Hỗ trợ nhiều công nghệ trong lập trình Windows

Một số điểm khác biệt giữa lập trình Windows & DOS

DOS	WINDOWS
Thực hiện tuần tự	Lập trình sự kiện dựa vào thông điệp (message)
Single Task	Multi-tasking
Single CPU	Multi – CPU
Phải dùng các thư viện Multimedia riêng	Tích hợp sẵn Multimedia
Ứng dụng 16 bits	Hỗ trợ 32 bits hoặc hơn nữa
Không hỗ trợ nhiều công nghệ	Hỗ trợ nhiều công nghệ DLL, COM, DDE, OLE...

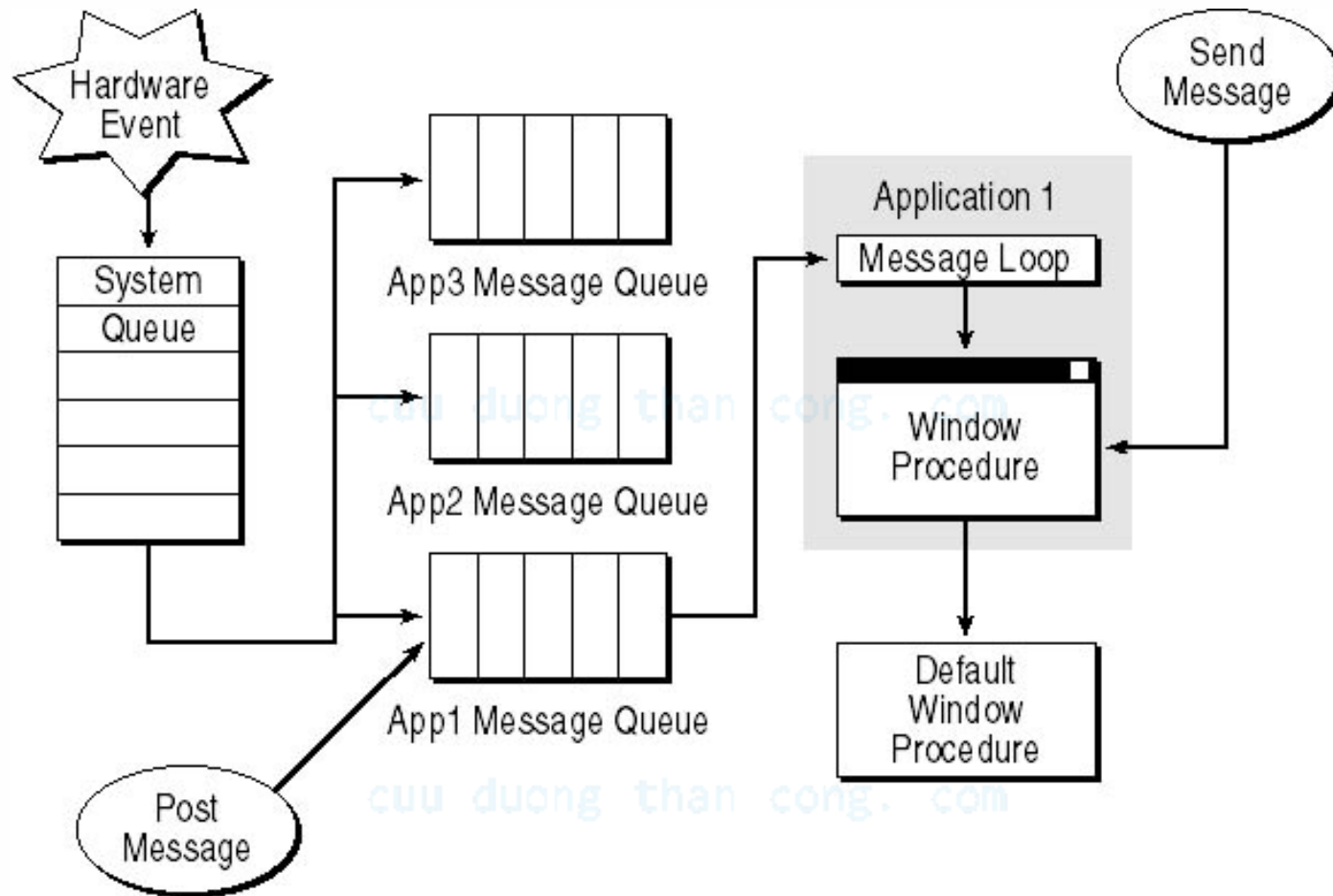
Một số khái niệm cơ bản

Khái niệm	Diễn giải
Handle	Một giá trị 32bits không dấu (unsigned) do HĐH tạo ra để làm định danh cho 1 đối tượng (file, cửa sổ, vùng nhớ, menu, toolbar...)
ID (Identifier)	Một giá trị nguyên do ứng dụng tạo ra để làm định danh cho 1 đối tượng (menu item, tool button, control,...) cuu duong than cong. com
Instance	Một giá trị nguyên do HĐH tạo ra để định danh cho 1 thể hiện đang thực thi của ứng dụng
Callback	Thuộc tính của một hàm/thủ tục sẽ được gọi bởi HĐH, không phải bởi ứng dụng
Library	Thư viện thủ tục/hàm, lớp đối tượng được xây dựng sẵn bởi window và các hãng khác

NỘI DUNG

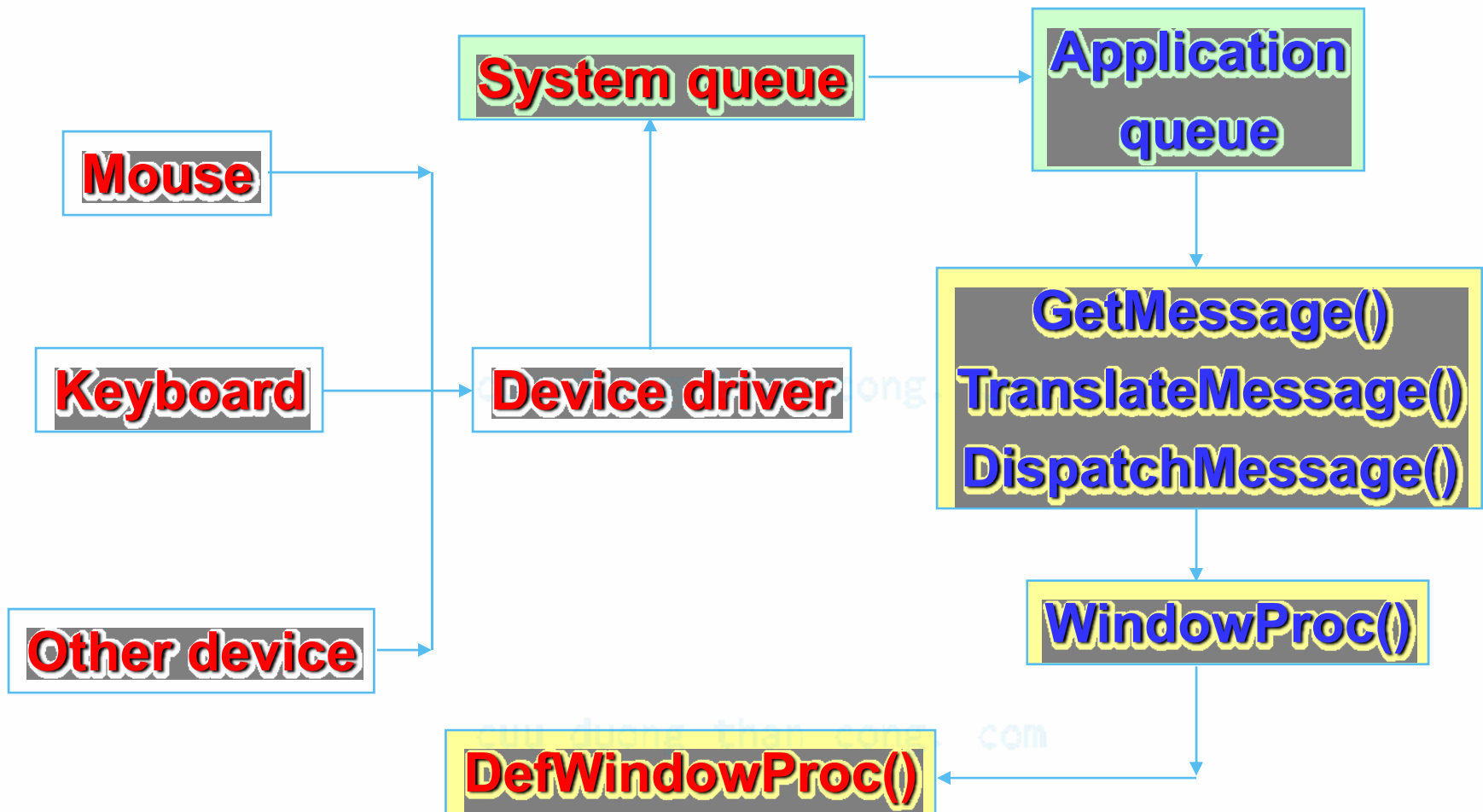
- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ **Lập trình sự kiện (Even driven programming)**
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

Lập trình Windows - lập trình sự kiện



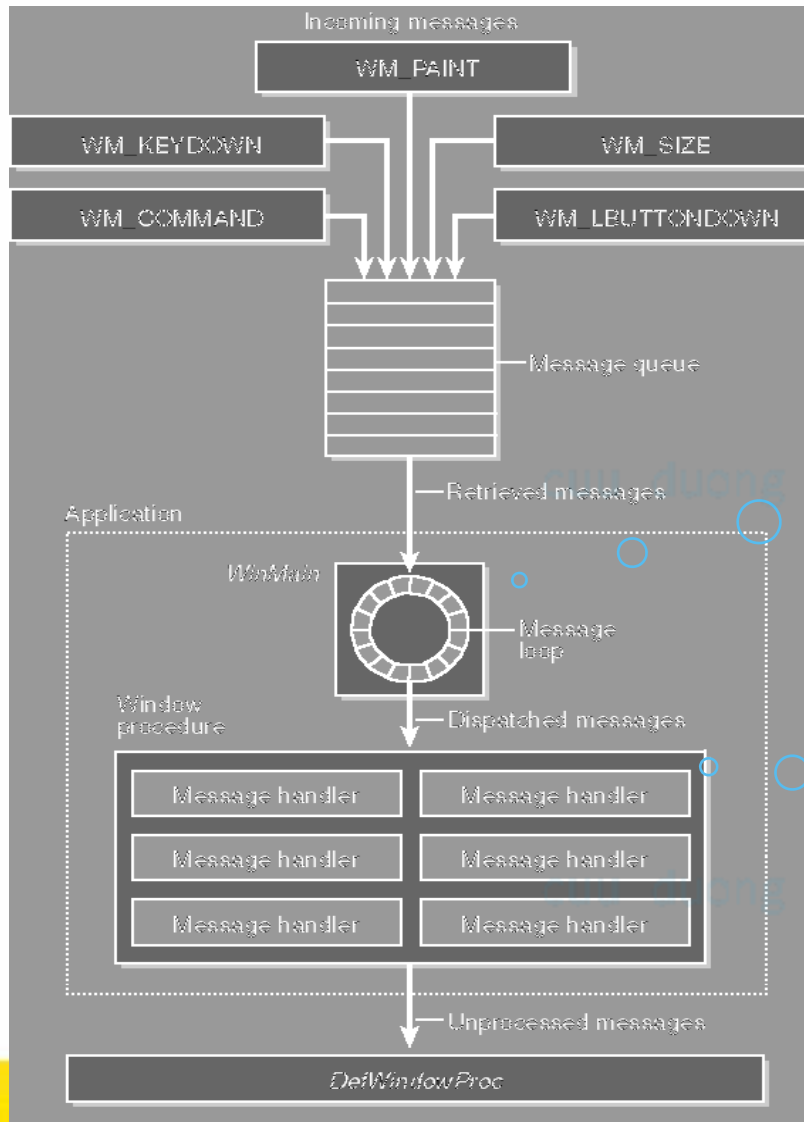
Sự kiện và thông điệp

Lập trình Windows - lập trình sự kiện



Qui trình xử lý thông điệp

Sự kiện và xử lý thông điệp



WinMain

**Window
Procedure**

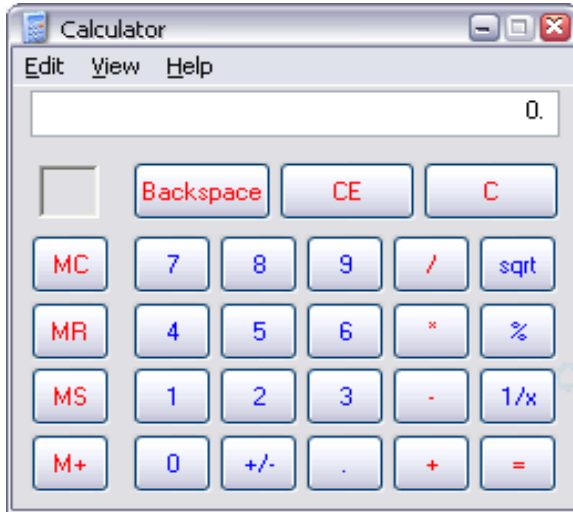
NỘI DUNG

- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Even driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

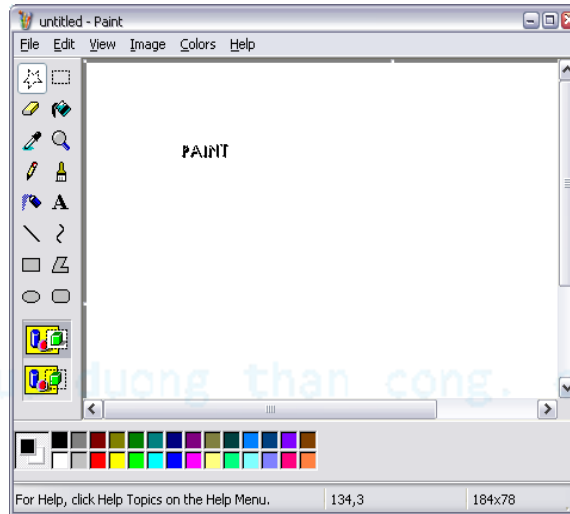
Các thành phần giao diện đồ họa - GUI

- ❖ GUI : Graphics User Interface
- ❖ Các dạng GUI cơ bản

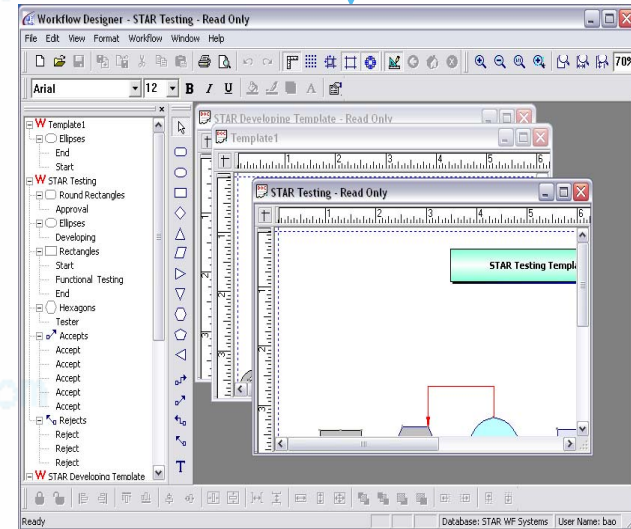
Dialog



SDI



MDI



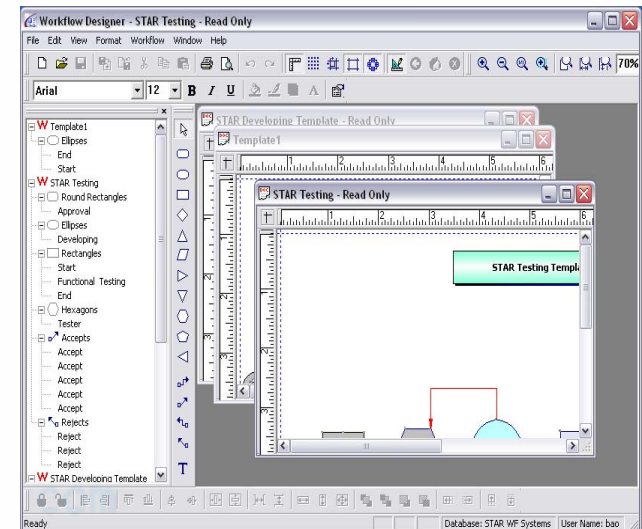
Các thành phần giao diện đồ họa - GUI

❖ Dialog

- Có một cửa sổ làm việc
- Thường có kích thước cố định
- Thường không có menu bar

❖ SDI – Single Document Interface

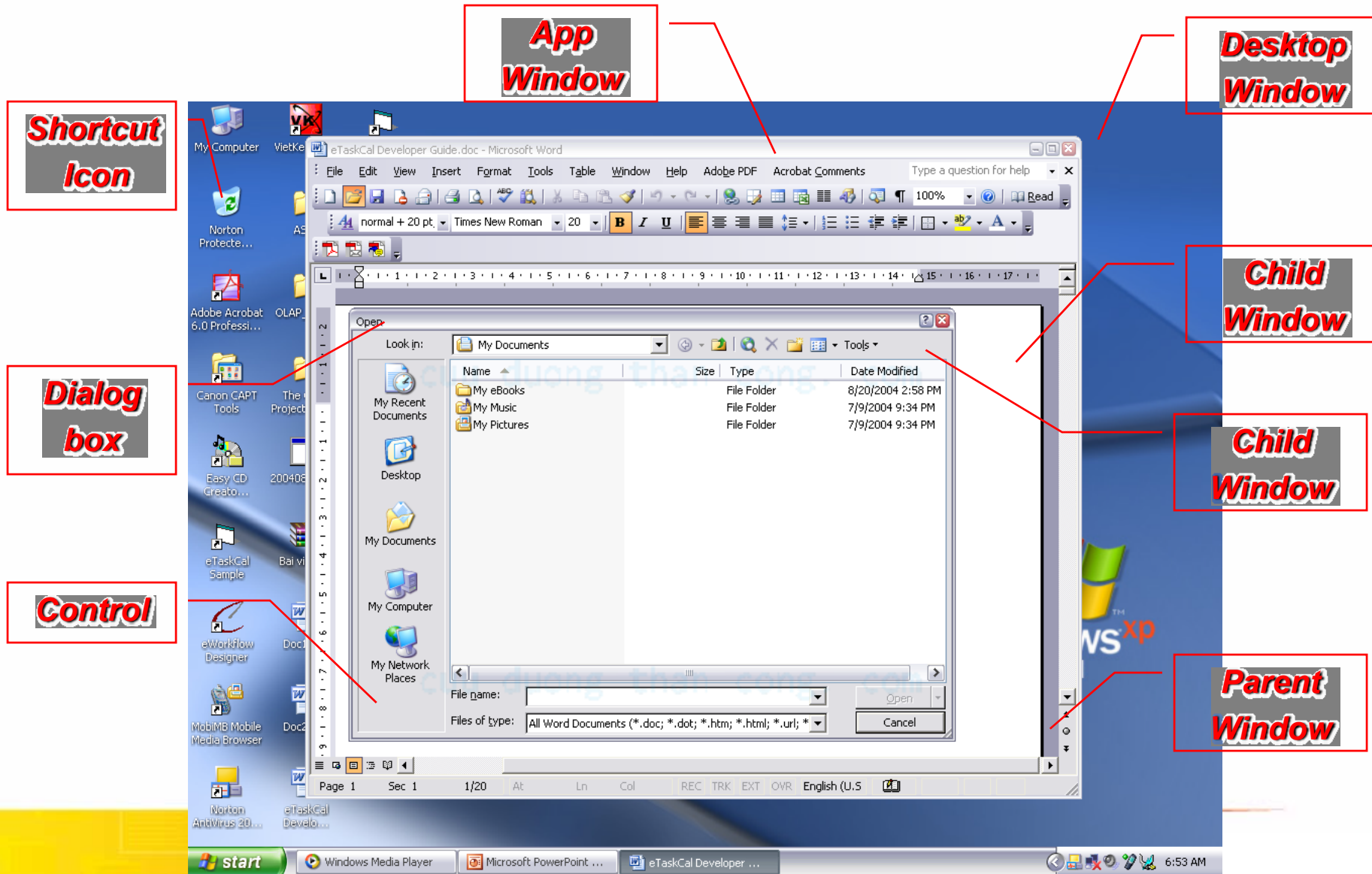
- Có một cửa sổ làm việc
- Cho phép thay đổi kích thước cửa sổ (Resizable)
- Không có cửa sổ con



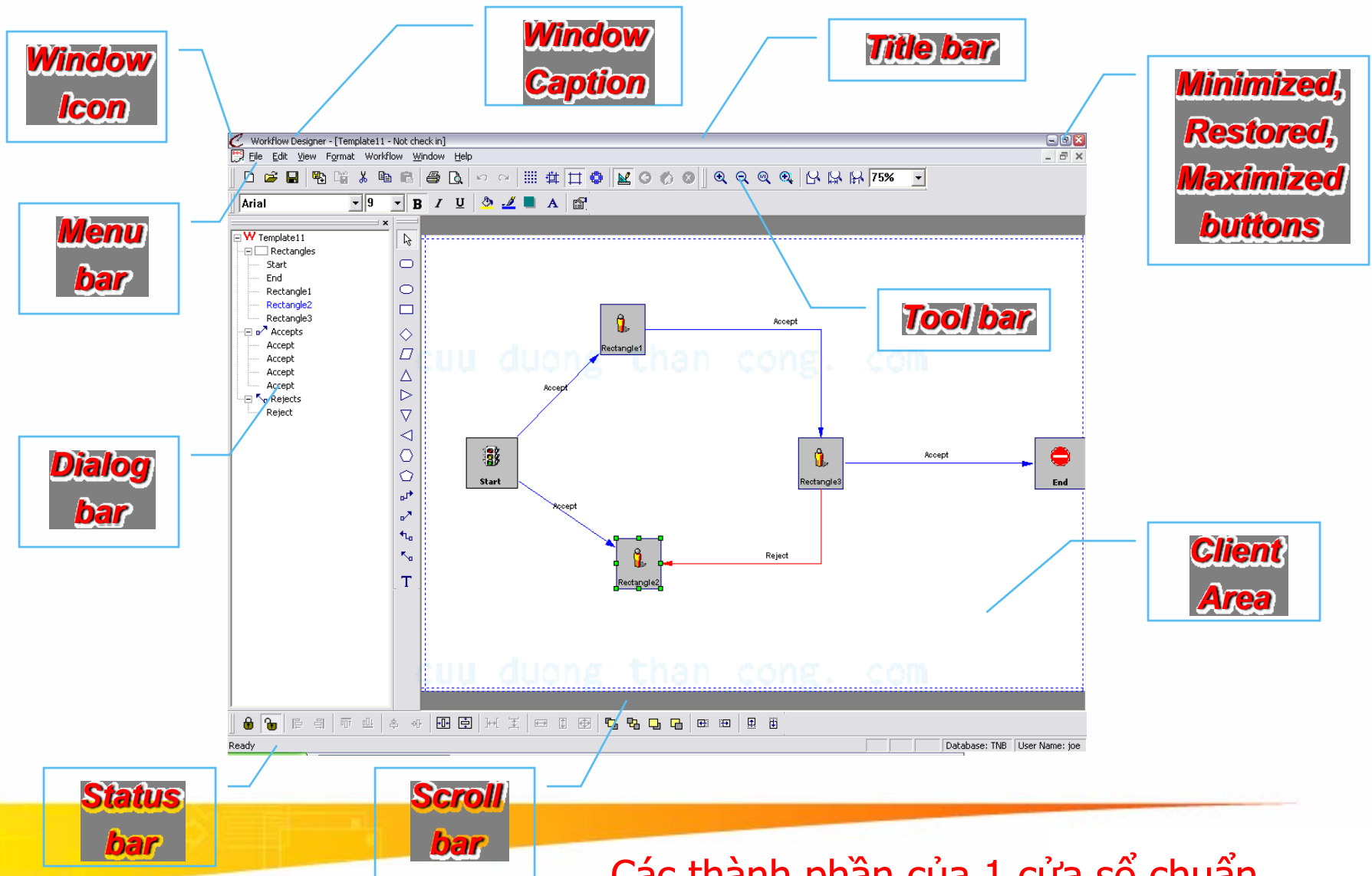
❖ MDI – Multiple Document Interface

- Có một cửa sổ làm việc chính (Main frame) và nhiều cửa sổ con (Child Frame)
- Cho phép thay đổi kích thước cửa sổ (Resizable)
- Cho phép Minimize/Maximize/Close/Arrange/Cascade cửa sổ con

Các thành phần giao diện đồ họa - GUI



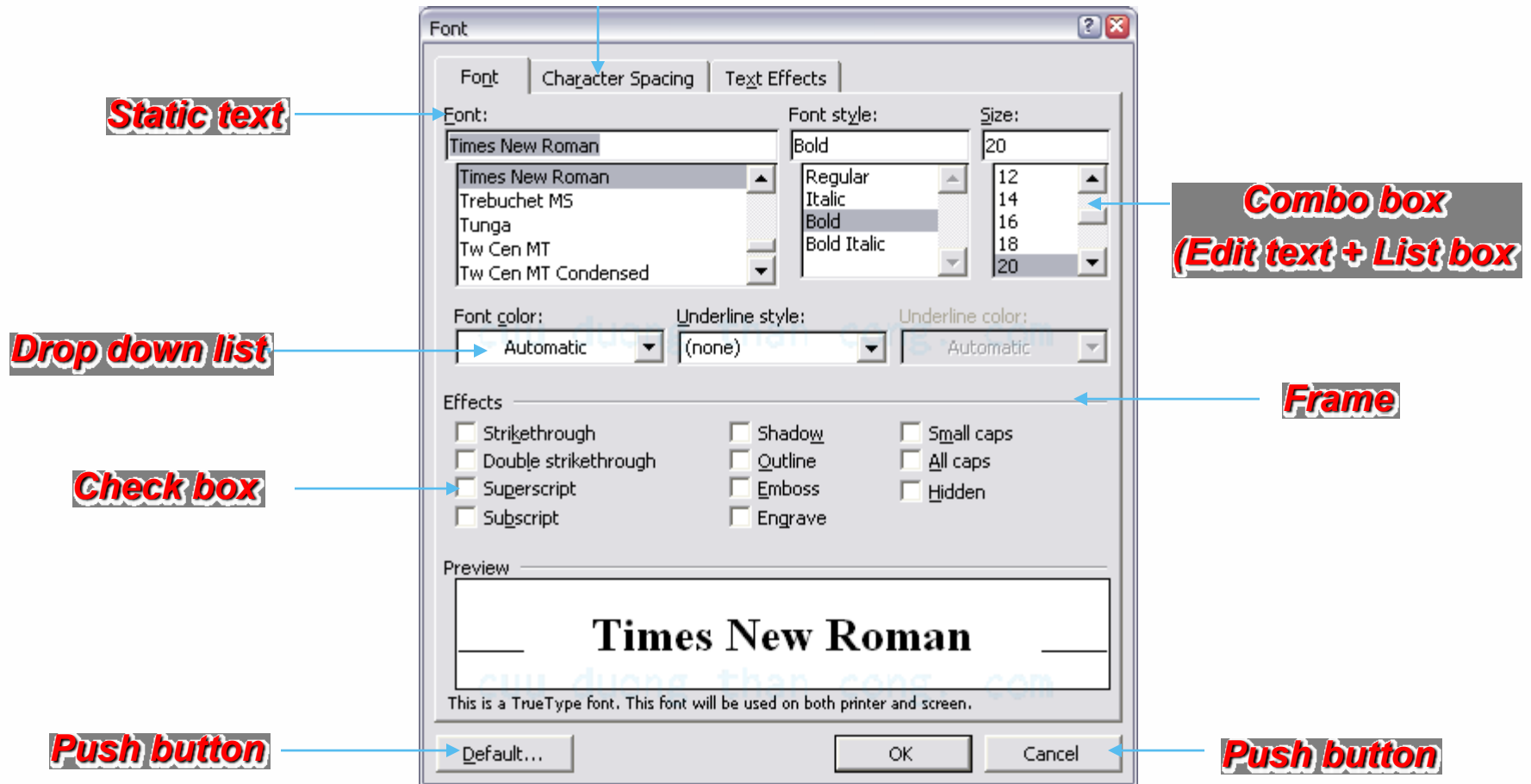
Các thành phần giao diện đồ họa - GUI



Các thành phần của 1 cửa sổ chuẩn

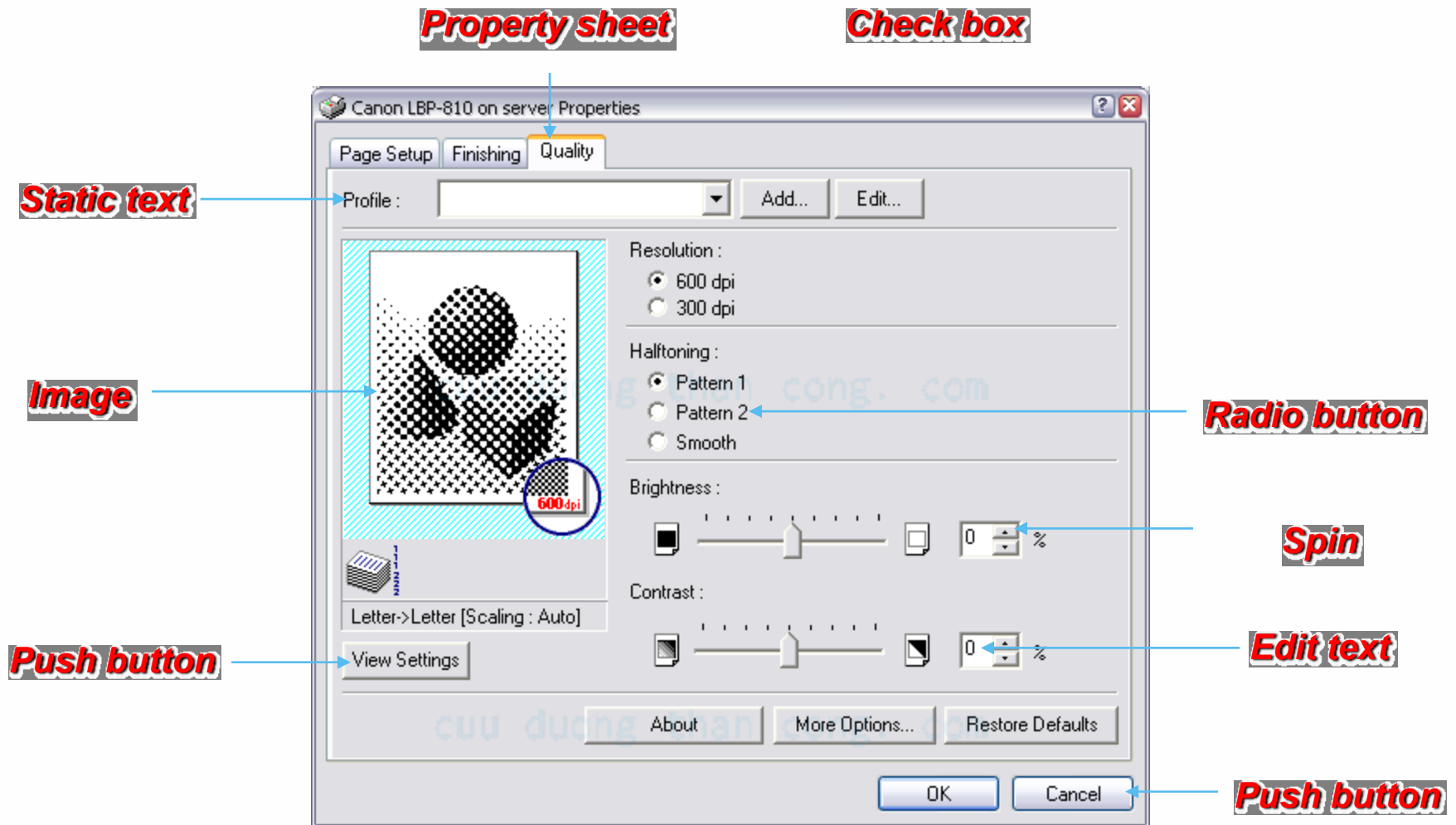
GUI – Controls

Property sheet



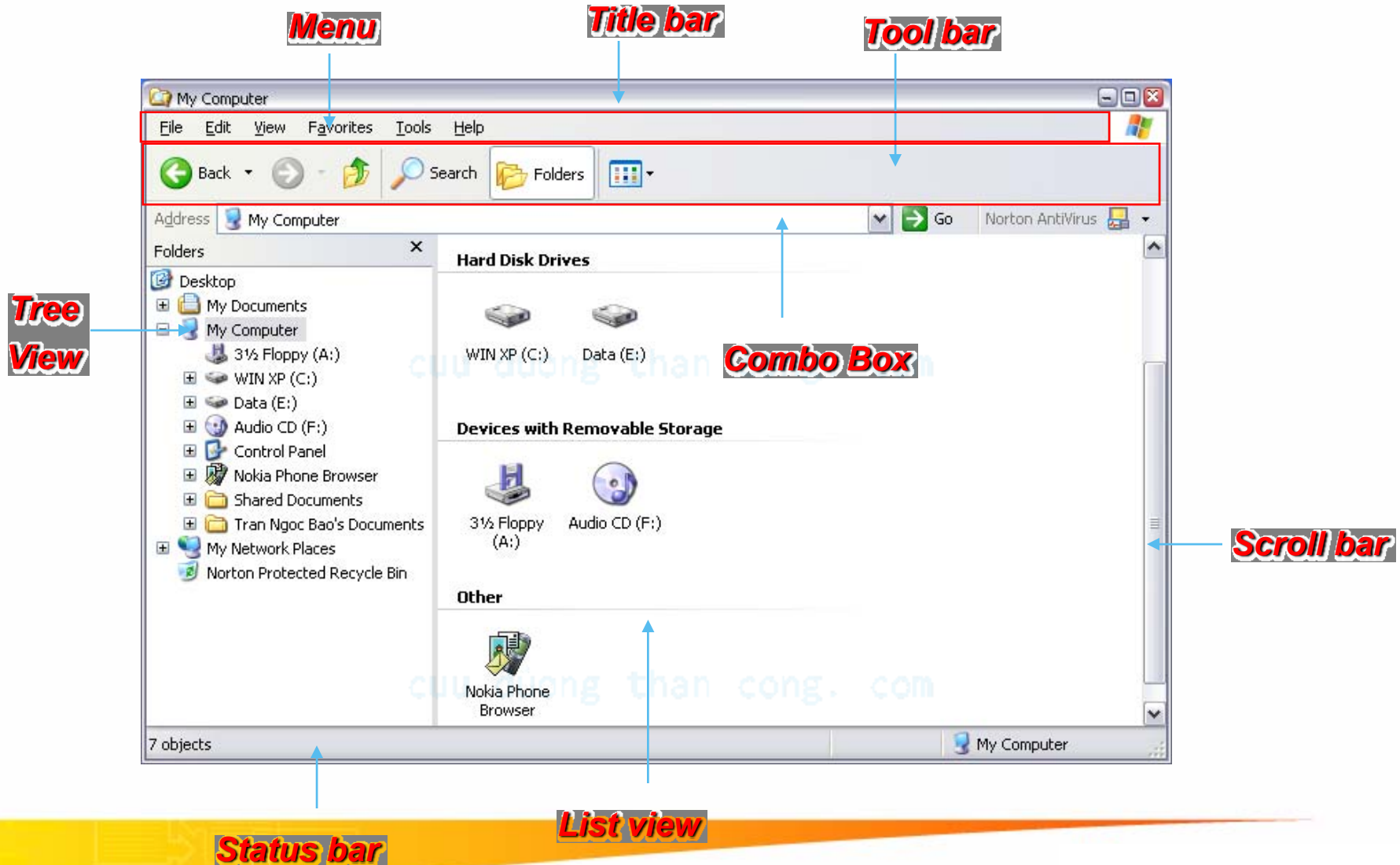
Các dạng control chuẩn của windows

GUI – Controls



Các dạng control chuẩn của windows

GUI – Controls

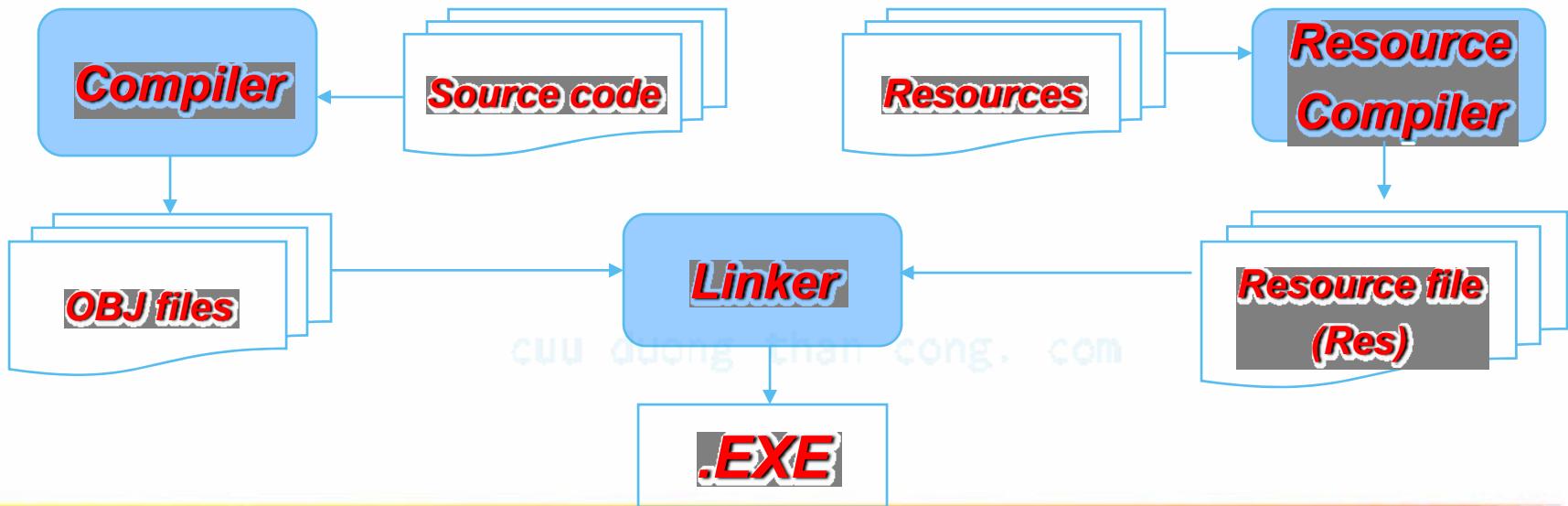


NỘI DUNG

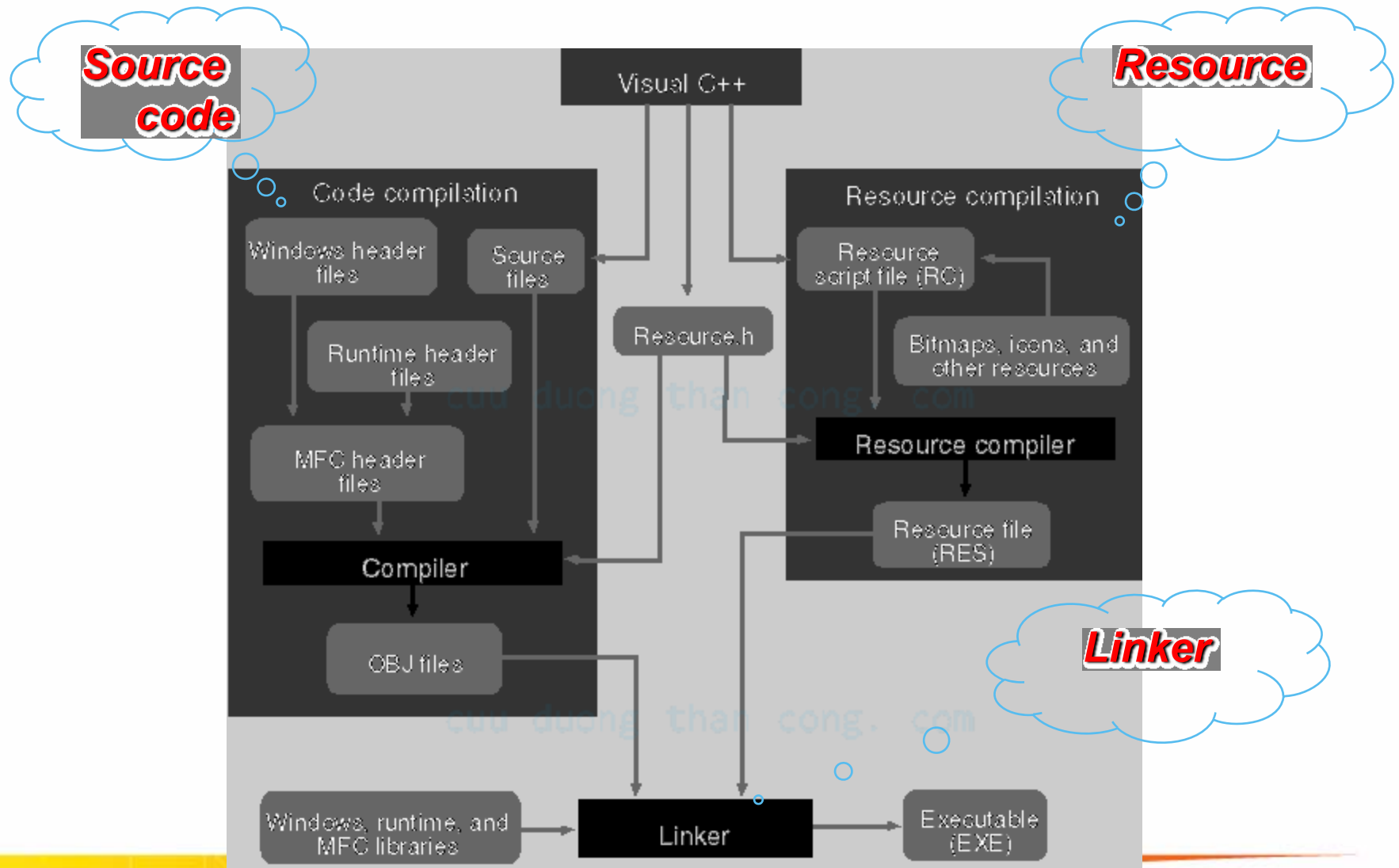
- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Even driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

Tài nguyên của ứng dụng (Resources)

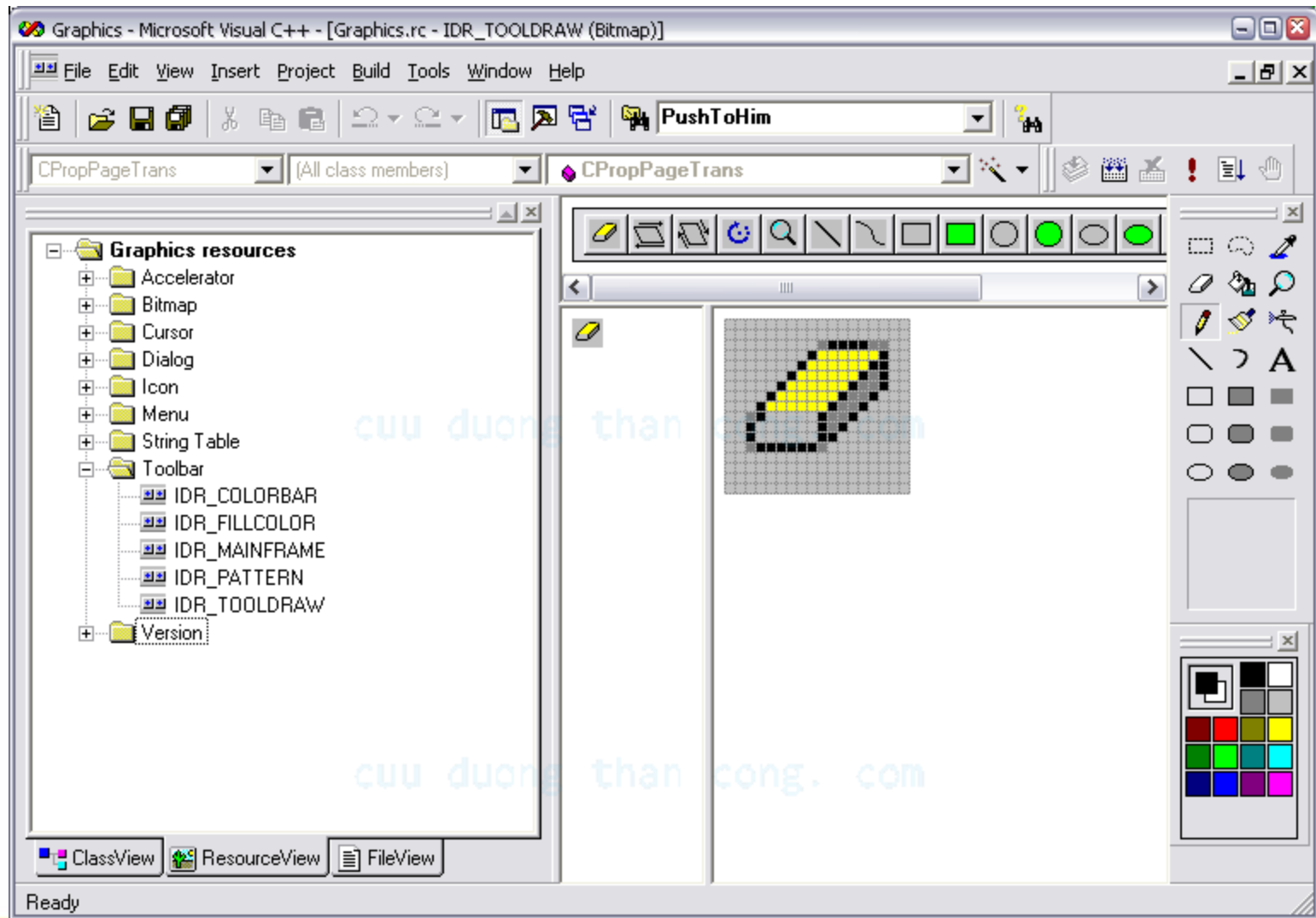
- ❖ Resource là một đối tượng (Object) được sử dụng trong ứng dụng, ví dụ: Menu bar, tool bar, dialog, icon, cursor, bitmap...
- ❖ Resource được định nghĩa bên ngoài (tách biệt với source code) và được thêm vào trong tập tin thi hành (EXE) của ứng dụng khi biên dịch (linking)



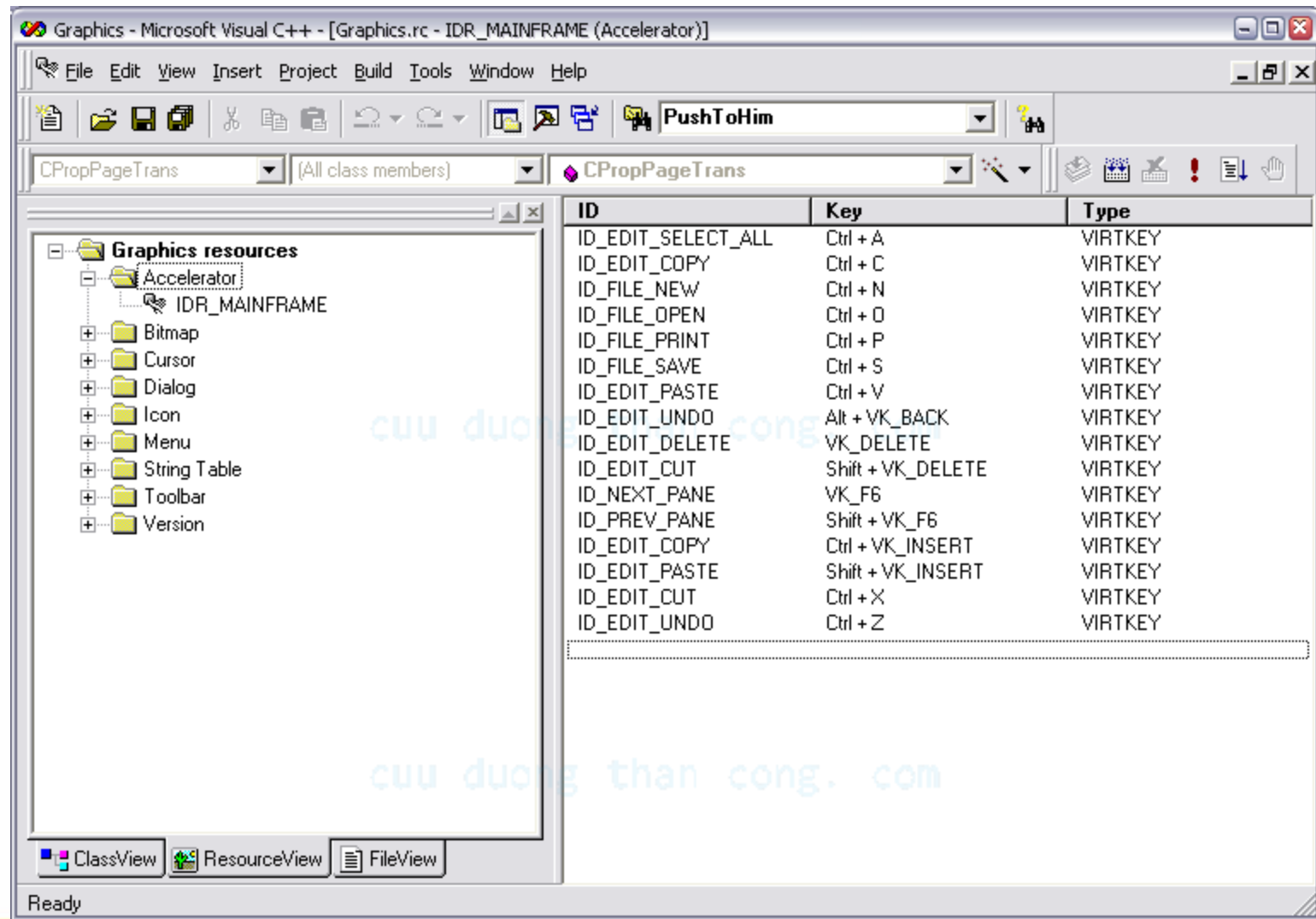
Quy trình biên dịch ứng dụng Visual C++



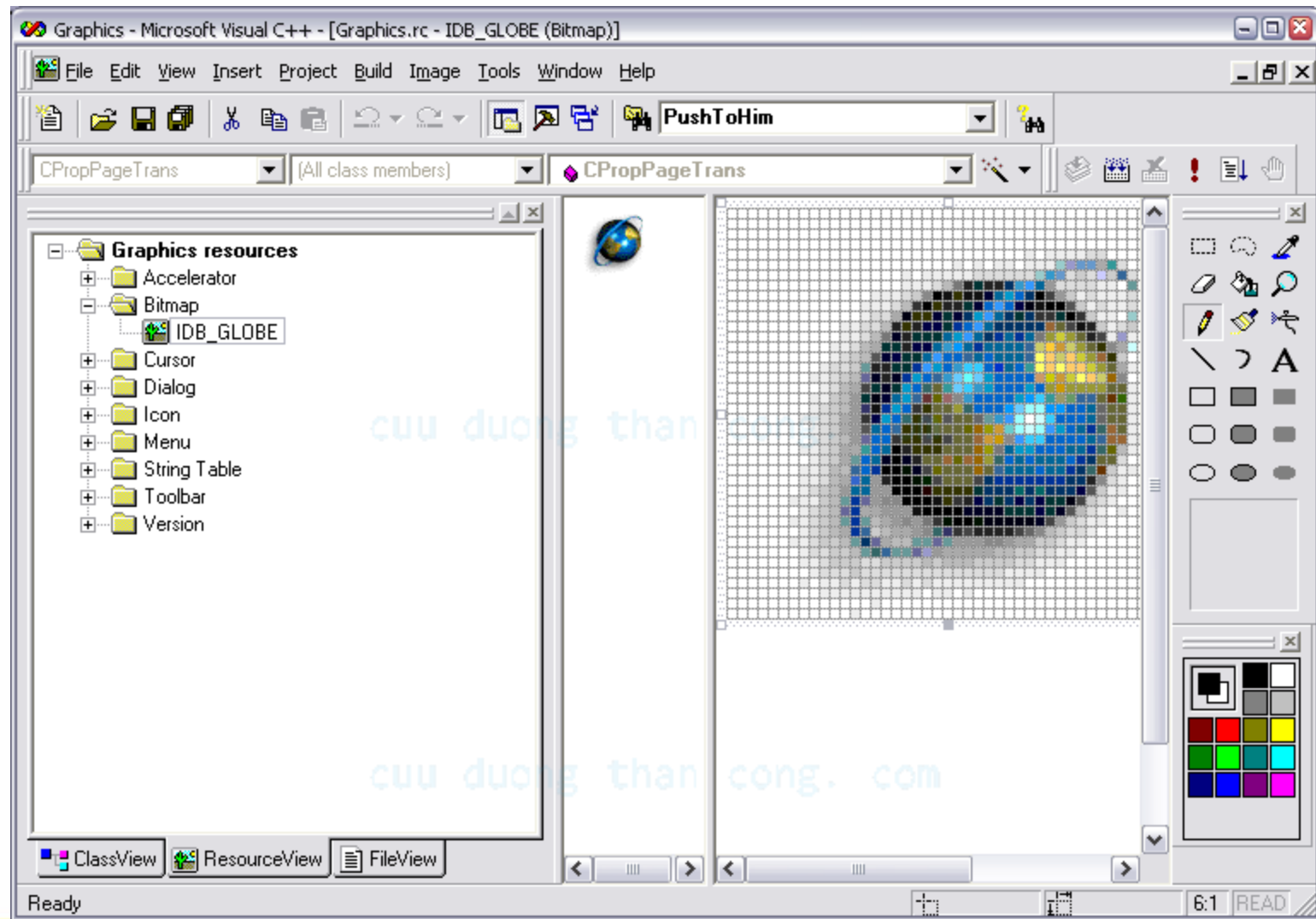
Tài nguyên của ứng dụng – Tool bar



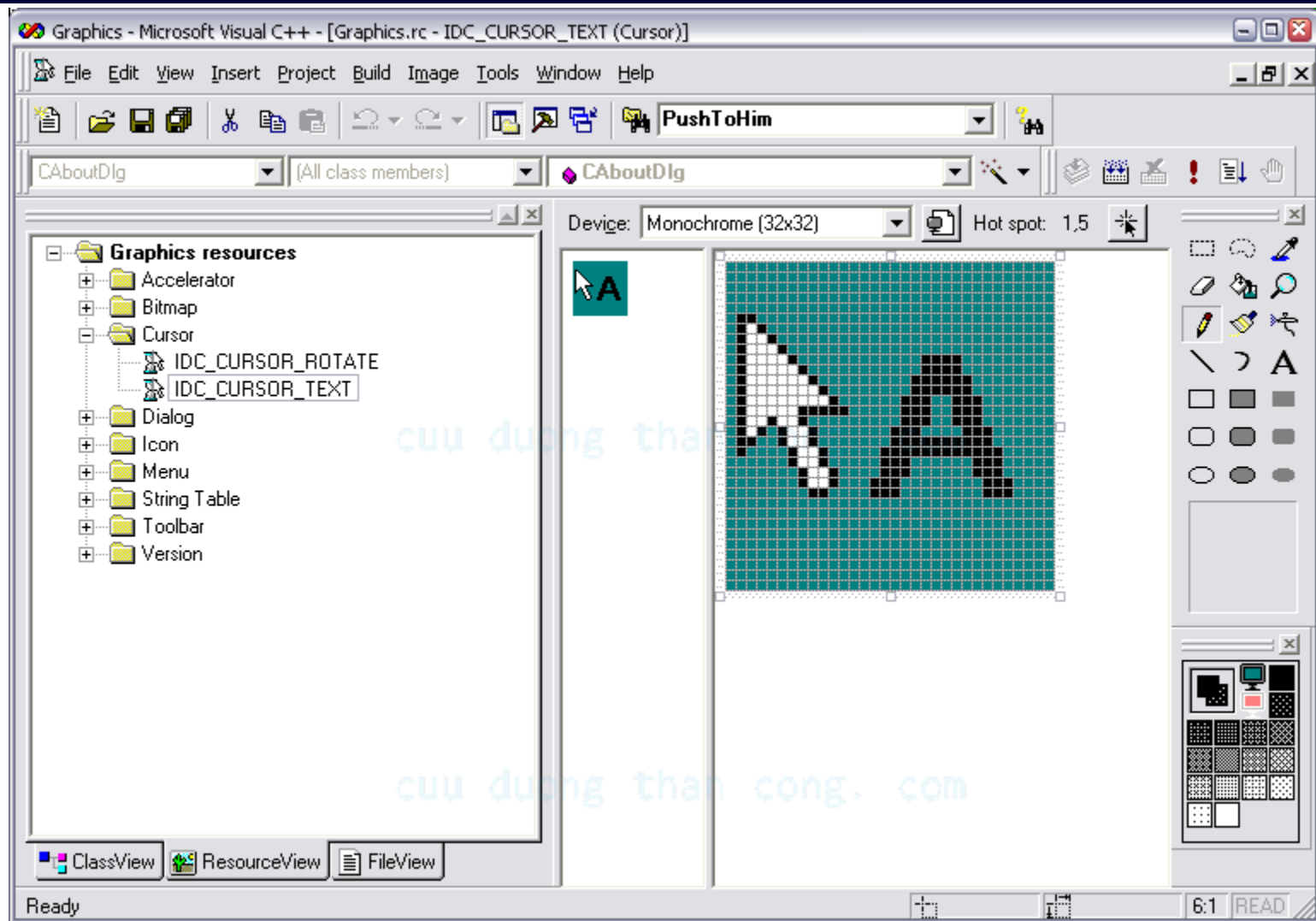
Tài nguyên của ứng dụng – Accelerator



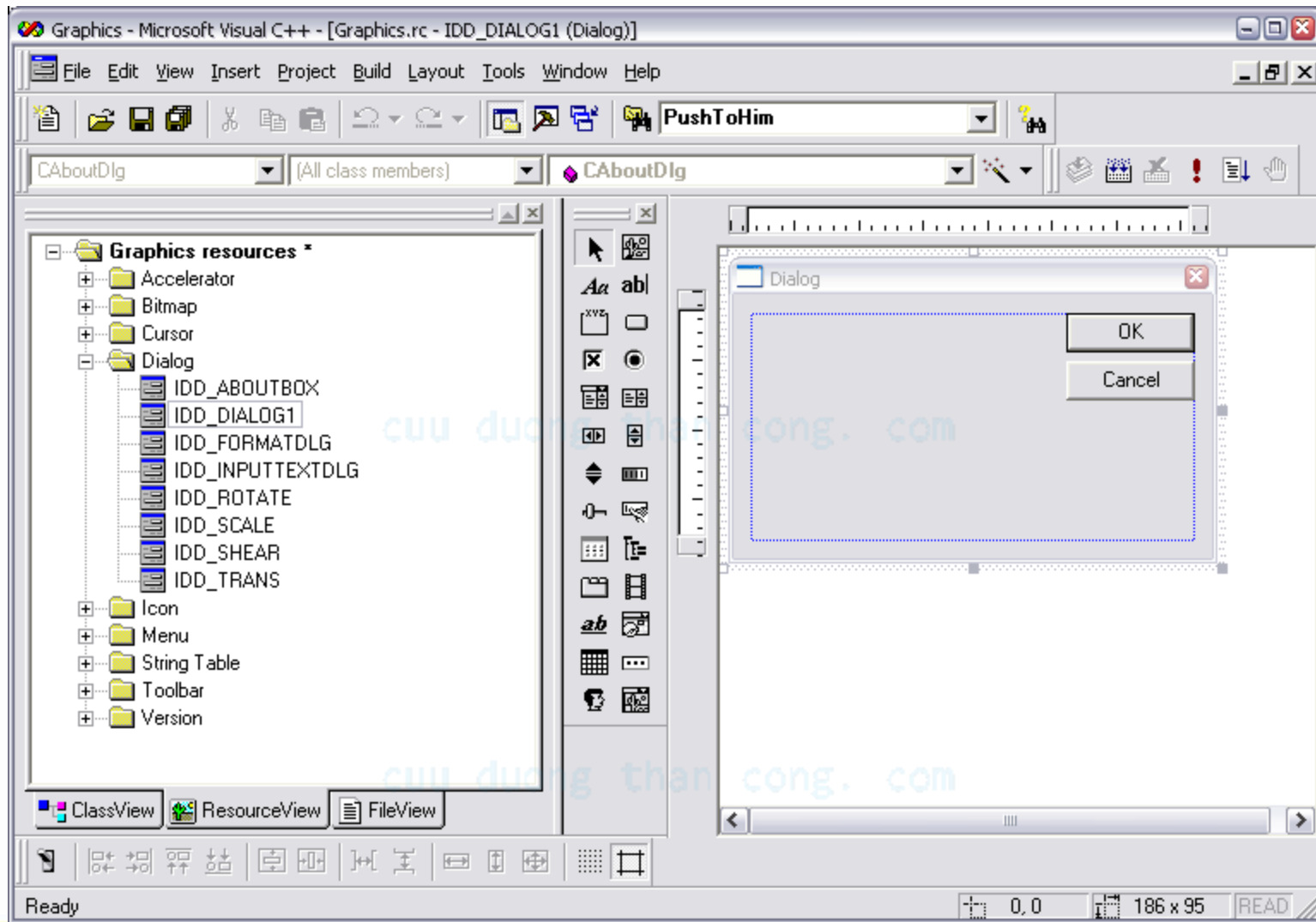
Tài nguyên của ứng dụng – Bitmap



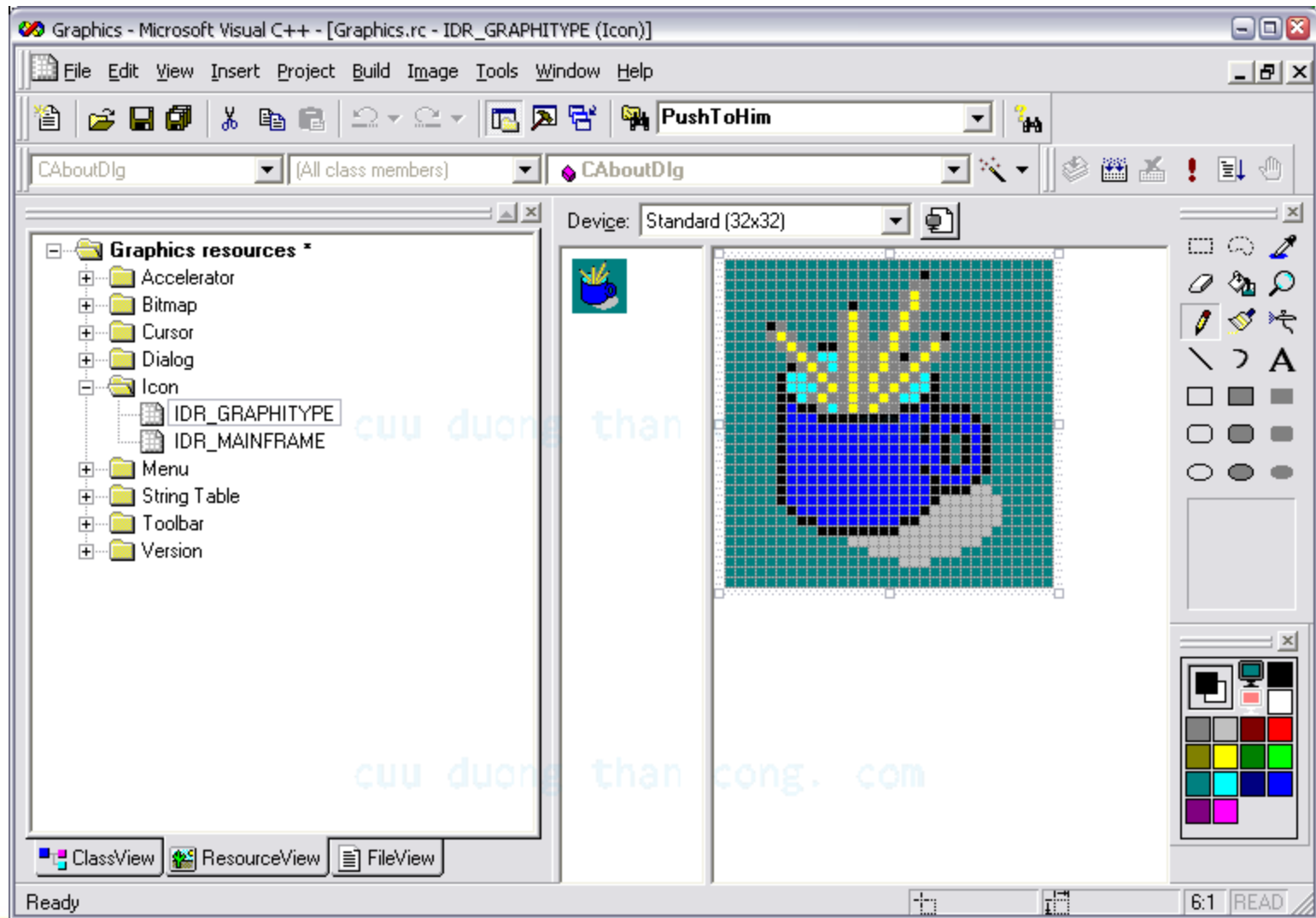
Tài nguyên của ứng dụng – Cursor



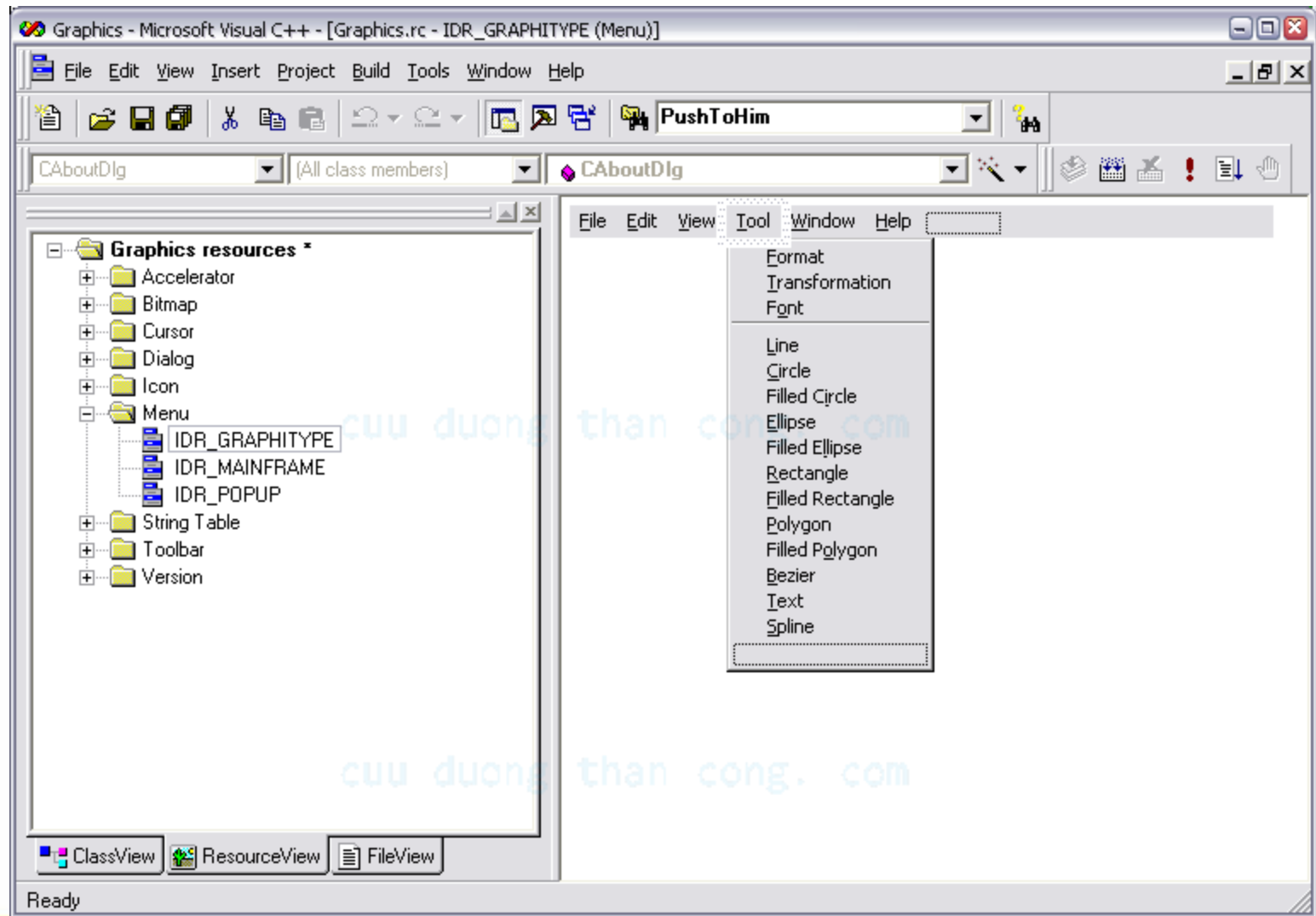
Tài nguyên của ứng dụng – Dialog



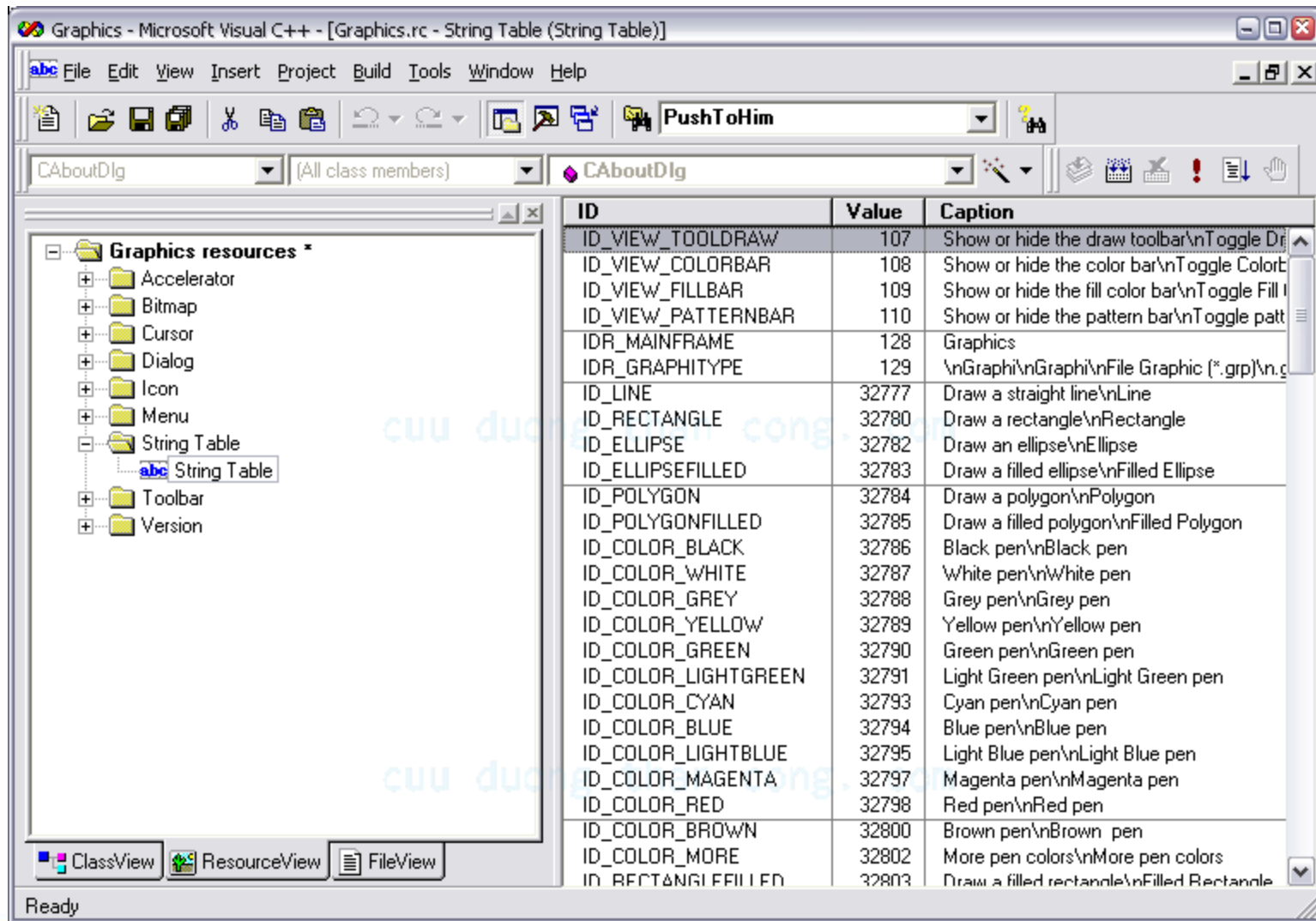
Tài nguyên của ứng dụng – Icon



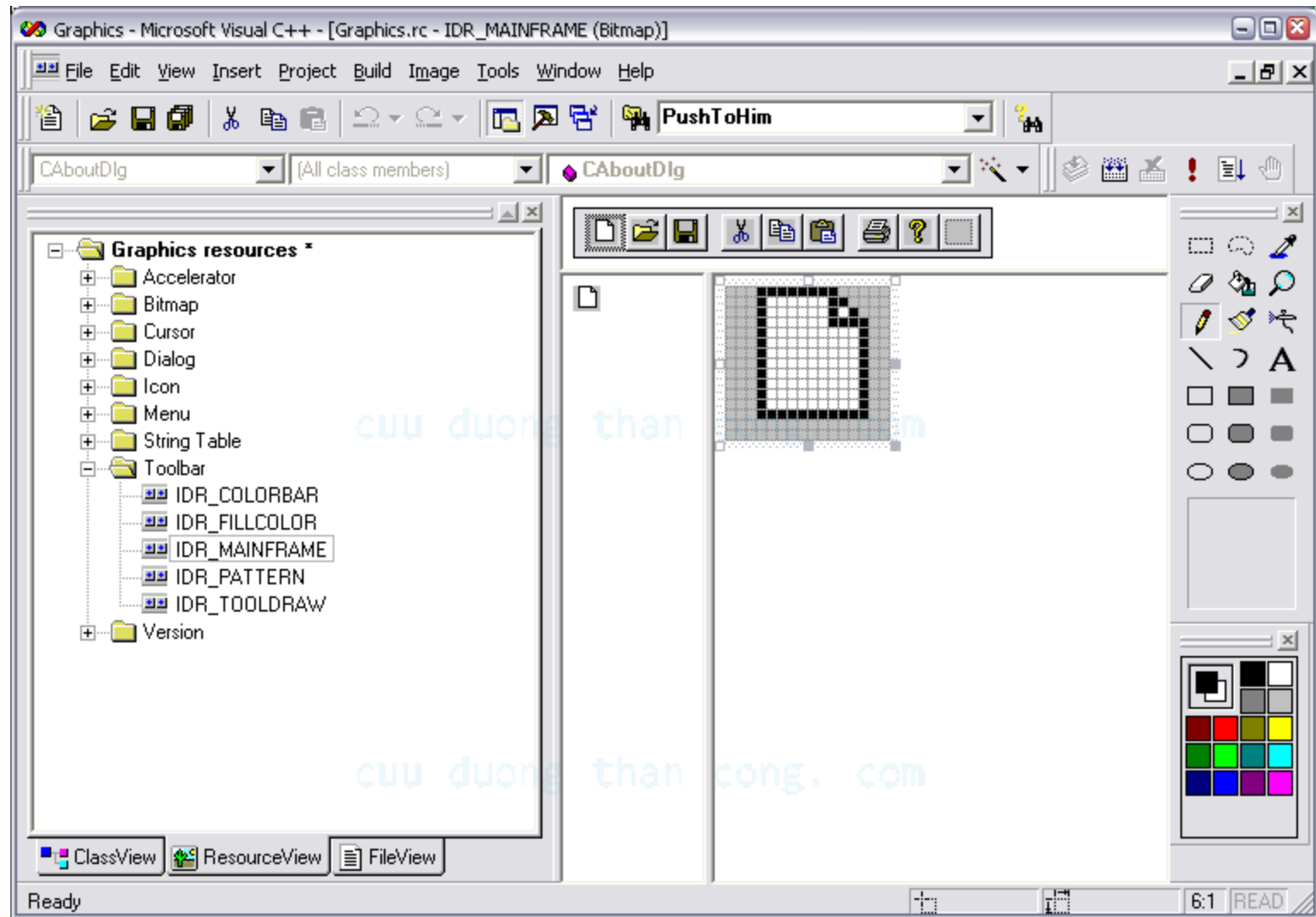
Tài nguyên của ứng dụng – Menu



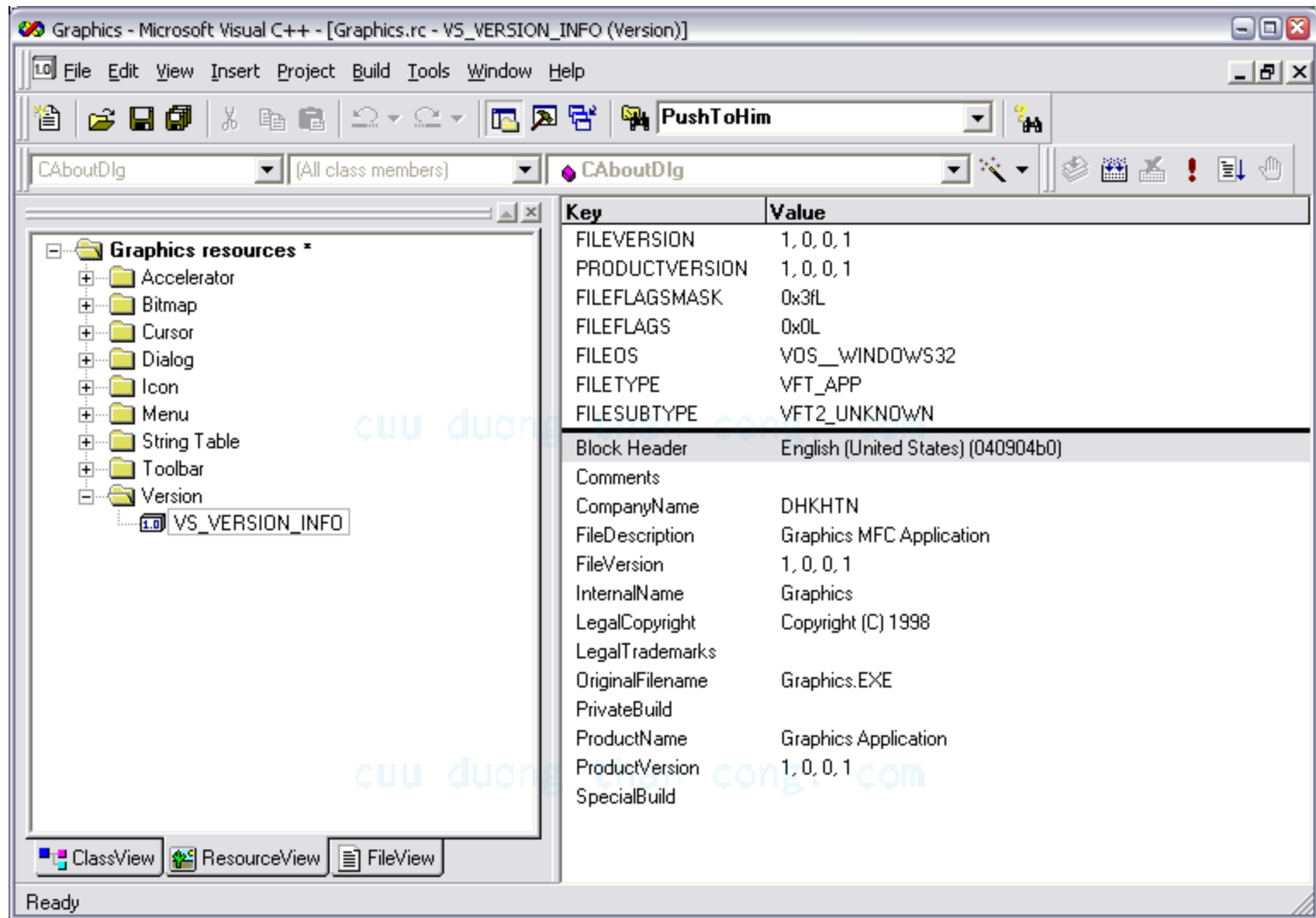
Tài nguyên của ứng dụng – String table



Tài nguyên của ứng dụng – Tool bar



Tài nguyên của ứng dụng – Version Info



Tài nguyên của ứng dụng (Resources)

Loại	Diễn giải
Accelerator	Bảng mô tả phím tắt (Hot- key)
Bitmap	Ảnh bitmap
Caret	Con trỏ văn bản
Cursor	Con trỏ chuột
Dialog box	Khung hộp thoại
Enhanced metafile	Tập hợp các cấu trúc để lưu ảnh (picture) theo định dạng "độc lập thiết bị" (Device Independent format)
Font	Font chữ
Icon	Biểu tượng
Menu	Menu
String table	Bảng mô tả các chuỗi ký tự
Version information	Bảng mô tả thông tin "phiên bản"

NỘI DUNG

- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Event driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ Mô hình lập trình Windows
- ❖ Một chương trình tổng quát xây dựng trên Win API

Bộ thư viện SDK – Software Developer Kit

- ❖ **SDK là bộ thư viện lập trình nền tảng của hệ điều hành Windows**
- ❖ **SDK cung cấp tất cả các công cụ cần thiết để xây dựng một ứng dụng trên Windows**
- ❖ **SDK được sử dụng như là thư viện cơ sở để tạo ra những thư viện cấp cao hơn trong những ngôn ngữ lập trình. Ví dụ: MFC của Visual C++, OWL của Borland,...**

Một số thành phần tiêu biểu của SDK

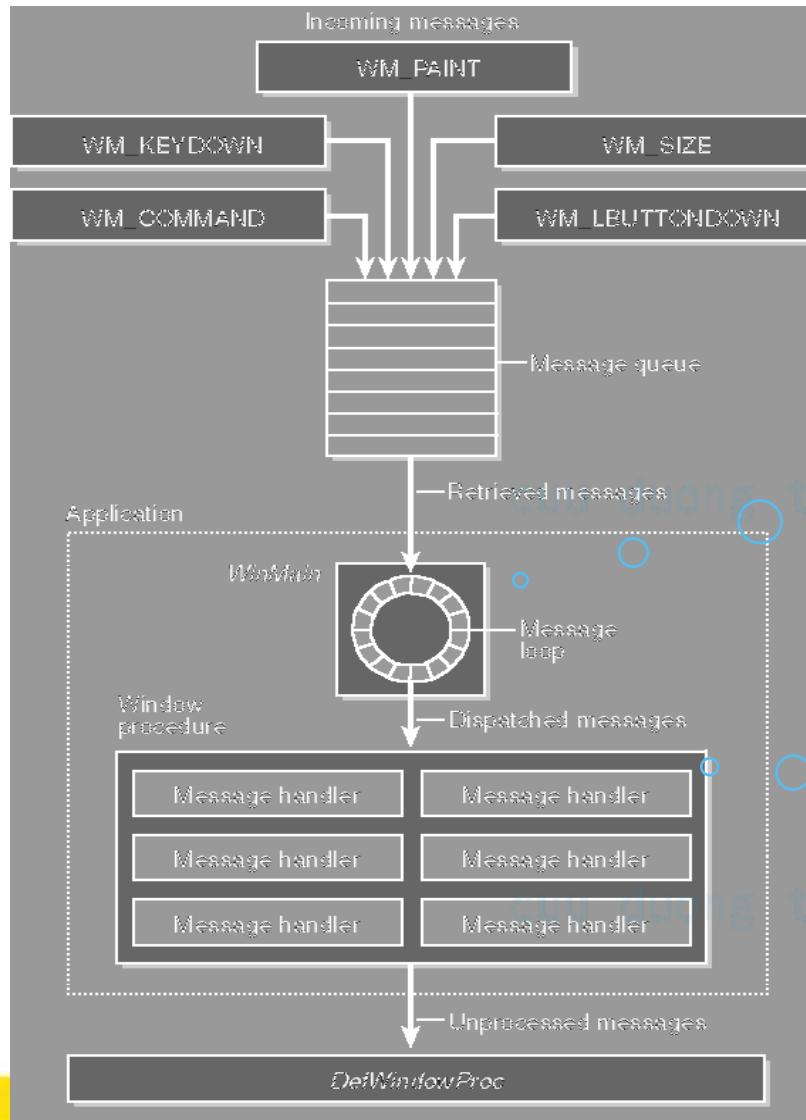
- ❖ Win32 API
- ❖ GDI/GDI+
- ❖ Windows Multimedia
- ❖ OpenGL
- ❖ DirectX
- ❖ COM/COM+
- ❖ ADO (ActiveX Data Object)
- ❖ OLE DB...

Tham khảo chi tiết trong
MSDN

NỘI DUNG

- ❖ Một số điểm khác biệt giữa lập trình Windows & DOS
- ❖ Một số khái niệm cơ bản
- ❖ Lập trình sự kiện (Event driven programming)
- ❖ Các thành phần giao diện đồ họa GUI
- ❖ Tài nguyên của ứng dụng (Resource)
- ❖ Thư viện lập trình Windows
- ❖ **Mô hình lập trình Windows**
- ❖ Một chương trình tổng quát xây dựng trên Win API

Mô hình lập trình Windows



WinMain

**Window
Procedure**

Một chương trình tổng quát xây dựng trên API

```
#include <windows.h>
LONG WINAPI WndProc (HWND, UINT, WPARAM, LPARAM);

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
    lpszCmdLine, int nCmdShow)
{
    //Khai báo lớp cửa sổ
    //Đăng ký lớp cửa sổ
    //Tạo lập cửa sổ
    //Hiển thị cửa sổ
    //Thực hiện vòng lặp xử lý thông điệp
}
```

Khai báo lớp cửa sổ

WNDCLASS wc;

HWND hwnd;

MSG msg;

wc.style = 0; // Class style

wc.lpfnWndProc = (WNDPROC) WndProc; // Window procedure

address wc.cbClsExtra = 0; // Class extra bytes

wc.cbWndExtra = 0; // Window extra bytes

wc.hInstance = hInstance; // Instance handle

wc.hIcon = LoadIcon (NULL, IDI_WINLOGO); // Icon handle

wc.hCursor = LoadCursor (NULL, IDC_ARROW); // Cursor handle

wc.hbrBackground = (HBRUSH) (COLOR_WINDOW + 1); //
Background color

wc.lpszMenuName = NULL; // Menu name

wc.lpszClassName = "MyWndClass"; // WNDCLASS name

Cấu trúc WNDCLASS

```
typedef struct WNDCLASS {  
    UINT          style;  
    WNDPROC       lpfnWndProc;  
    int           cbClsExtra;  
    int           cbWndExtra;  
    HINSTANCE     hInstance;  
    HICON         hIcon;  
    HCURSOR       hCursor;  
    HBRUSH        hbrBackground ;  
    LPCTSTR       lpstrMenuName;  
    LPCTSTR       lpstrClassName;  
} WNDCLASS, *PWNDCLASS;
```


Cấu trúc WNDCLASS

Thuộc tính	Ý nghĩa
Style	Kiểu dáng của lớp
lpfnWndProc	Hàm tiếp nhận và xử lý thông điệp
cbClsExtra	Số byte dành riêng cho class (≤ 40 bytes)
cbWndExtra	số byte dành riêng cho mỗi cửa sổ thuộc class (≤ 40 bytes)
hInstance	Instance của ứng dụng thực hiện việc đăng ký class
hIcon	Handle của icon, xác định biểu tượng của cửa sổ
hCursor	Handle của cursor, xác định dạng cursor trong vùng client
hbrBackground	Handle của brush, xác định cách tô vùng client; +1 nếu muốn sử dụng màu mặc định của hệ thống. Ví dụ: COLOR_BACKGROUND + 1
lpszMenuName	Tên menu bar, chuỗi kết thúc bằng 0
lpszClassName	Tên lớp, chuỗi kết thúc bằng 0

Đăng ký và tạo lập lớp cửa sổ

```
RegisterClass (&wc);  
hwnd = CreateWindow (  
    "MyWndClass", // WNDCLASS name  
    "SDK Application", // Window title  
    WS_OVERLAPPEDWINDOW, // Window style  
    CW_USEDEFAULT, // Horizontal position  
    CW_USEDEFAULT, // Vertical position  
    CW_USEDEFAULT, // Initial width  
    CW_USEDEFAULT, // Initial height  
    HWND_DESKTOP, // Handle of parent window  
    NULL, // Menu handle  
    hInstance, // Application's instance handle  
    NULL // Window-creation data );
```

Hiển thị cửa sổ - vòng lặp xử lý thông điệp

```
ShowWindow (hwnd, nCmdShow);  
UpdateWindow (hwnd);  
while (GetMessage (&msg, NULL, 0, 0))  
{  
    TranslateMessage (&msg);  
    DispatchMessage (&msg);  
}  
return msg.wParam;
```

cuu duong than cong. com

Hàm xử lý Message của cửa sổ

```
LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM  
    lParam)  
{  
    PAINTSTRUCT ps;  
    HDC hdc;  
    switch (message) {  
        case WM_PAINT:  
            hdc = BeginPaint (hwnd, &ps);  
            Ellipse (hdc, 0, 0, 200, 100);  
            EndPaint (hwnd, &ps);  
            return 0;  
        case WM_DESTROY:  
            PostQuitMessage (0);  
            return 0;  
    }  
    return DefWindowProc (hwnd, message, wParam, lParam);  
}
```

Tạo ứng dụng Win API bằng AppWizard

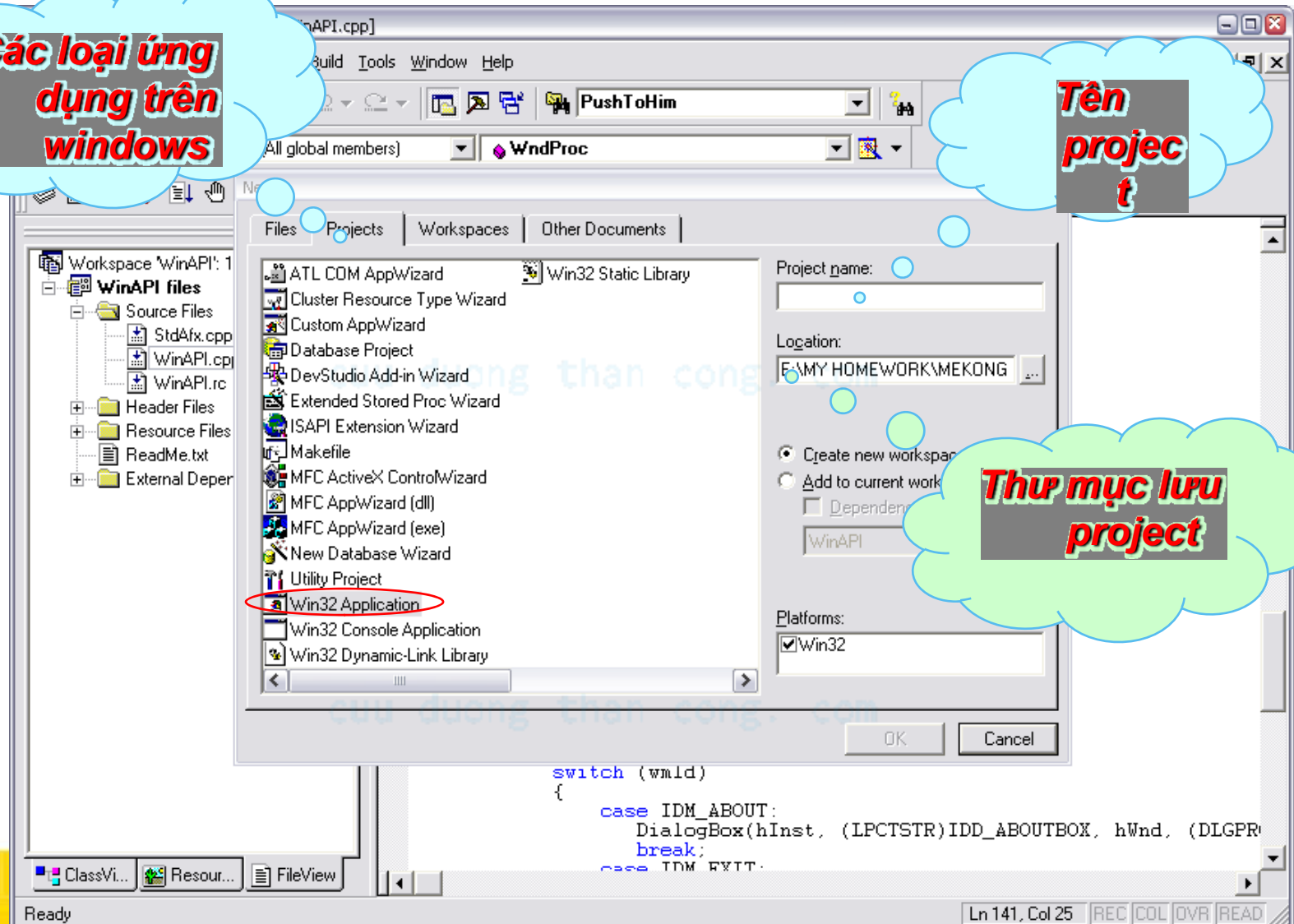
- ❖ Chọn menu **File → New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project **"Win32 Application"**
- ❖ Đặt tên project trong ô **"Project name"**
- ❖ Xác định đường dẫn thư mục trong **"Location"**
- ❖ Chọn option **"Create new workspace"**
- ❖ Nhấn **"OK"** qua bước sau

Tạo ứng dụng Win API bằng AppWizard

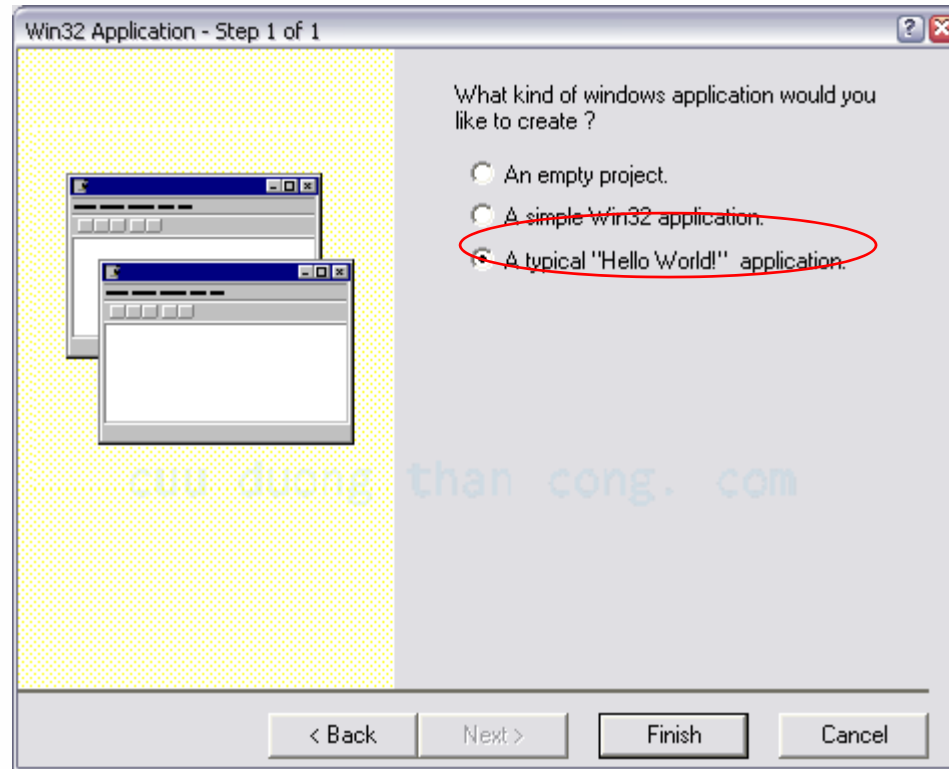
Các loại ứng dụng trên windows

Tên project

Thư mục lưu project

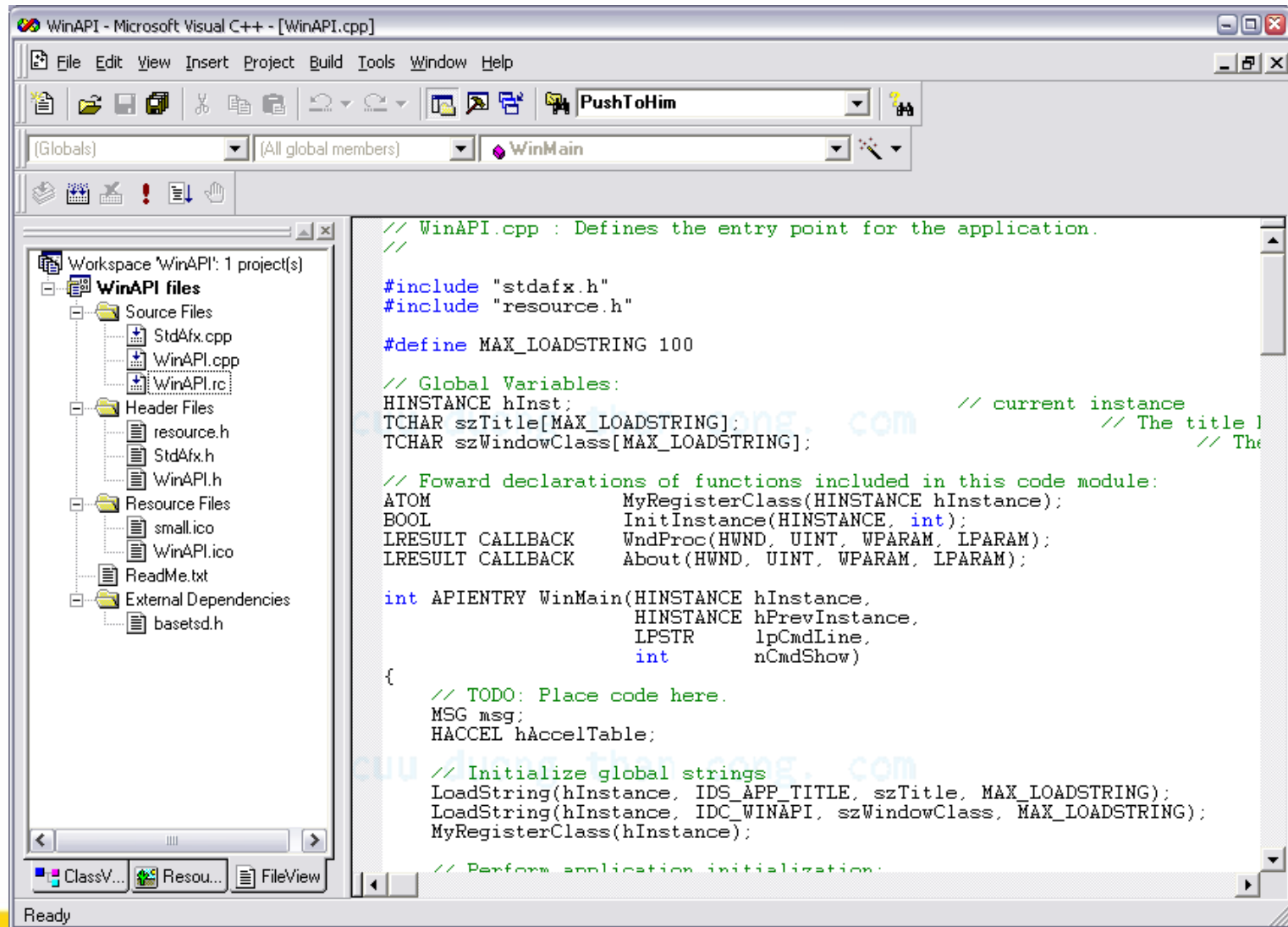


Tạo ứng dụng Win API bằng AppWizard

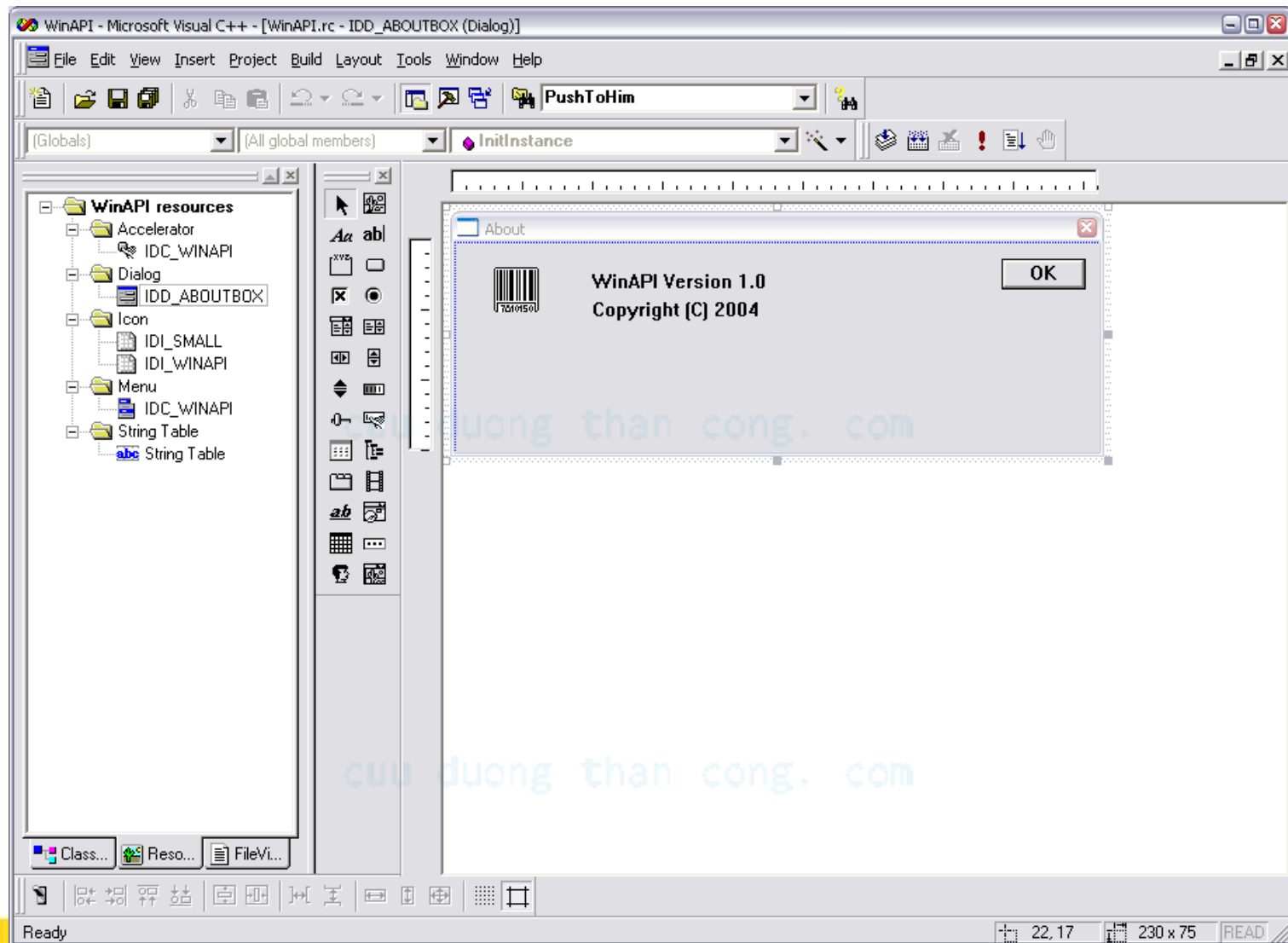


Chọn option **"A typical Hello World application"** để AppWizard tự động tạo ra ứng dụng mẫu

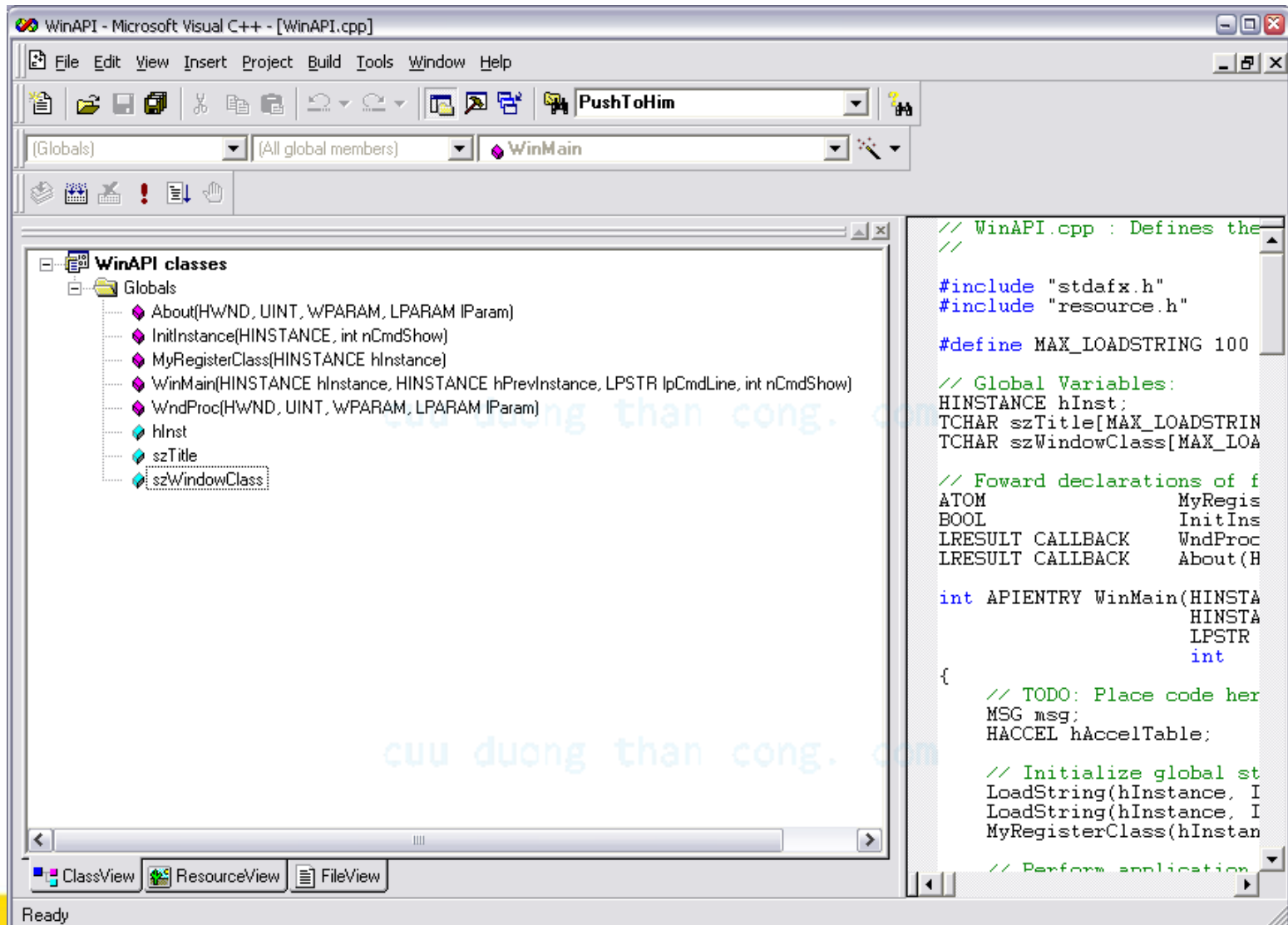
Ứng dụng Win API đơn giản – Source file



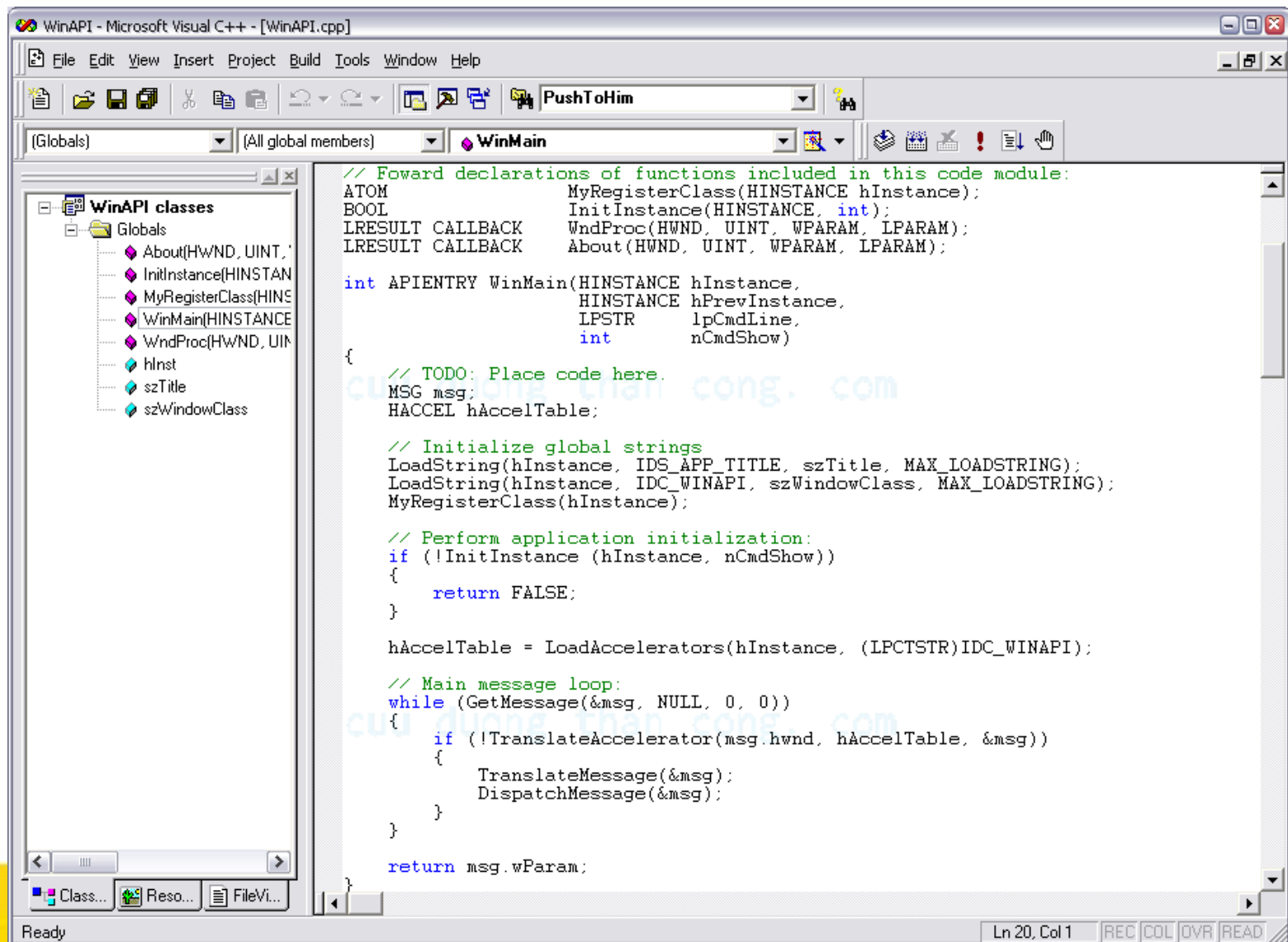
Ứng dụng Win API - Resource



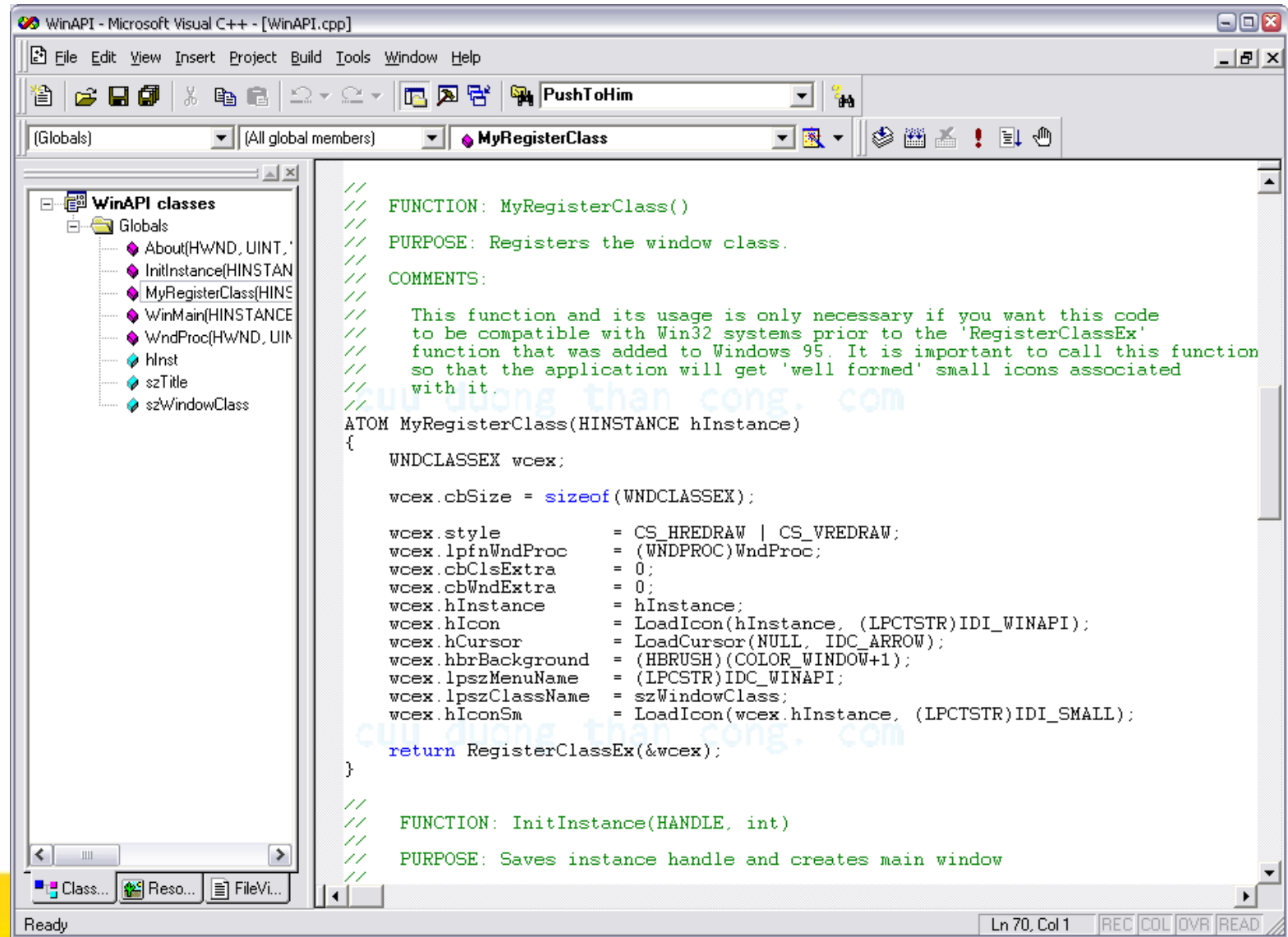
Ứng dụng Win API – Source code



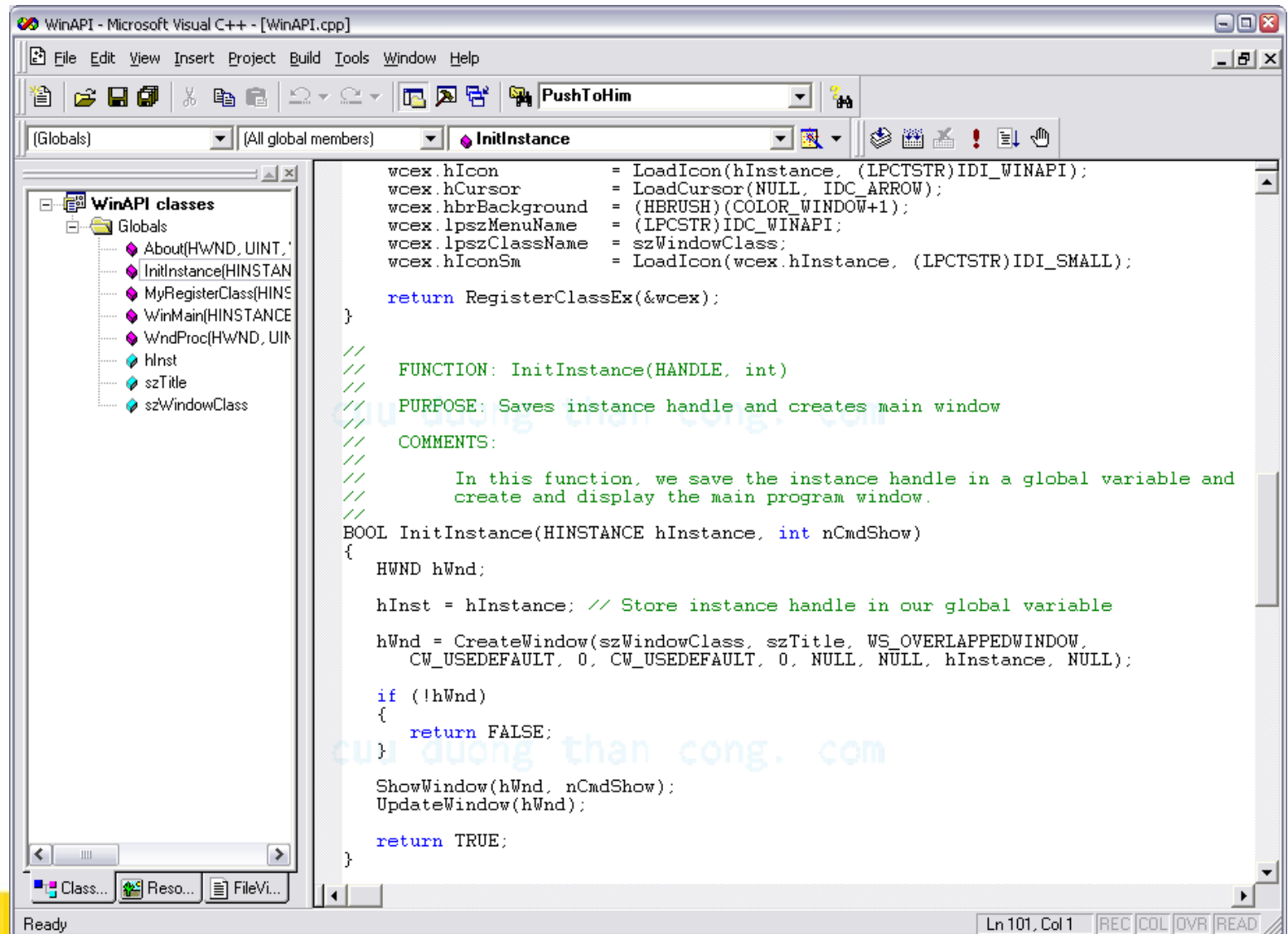
Ứng dụng Win API – hàm WinMain



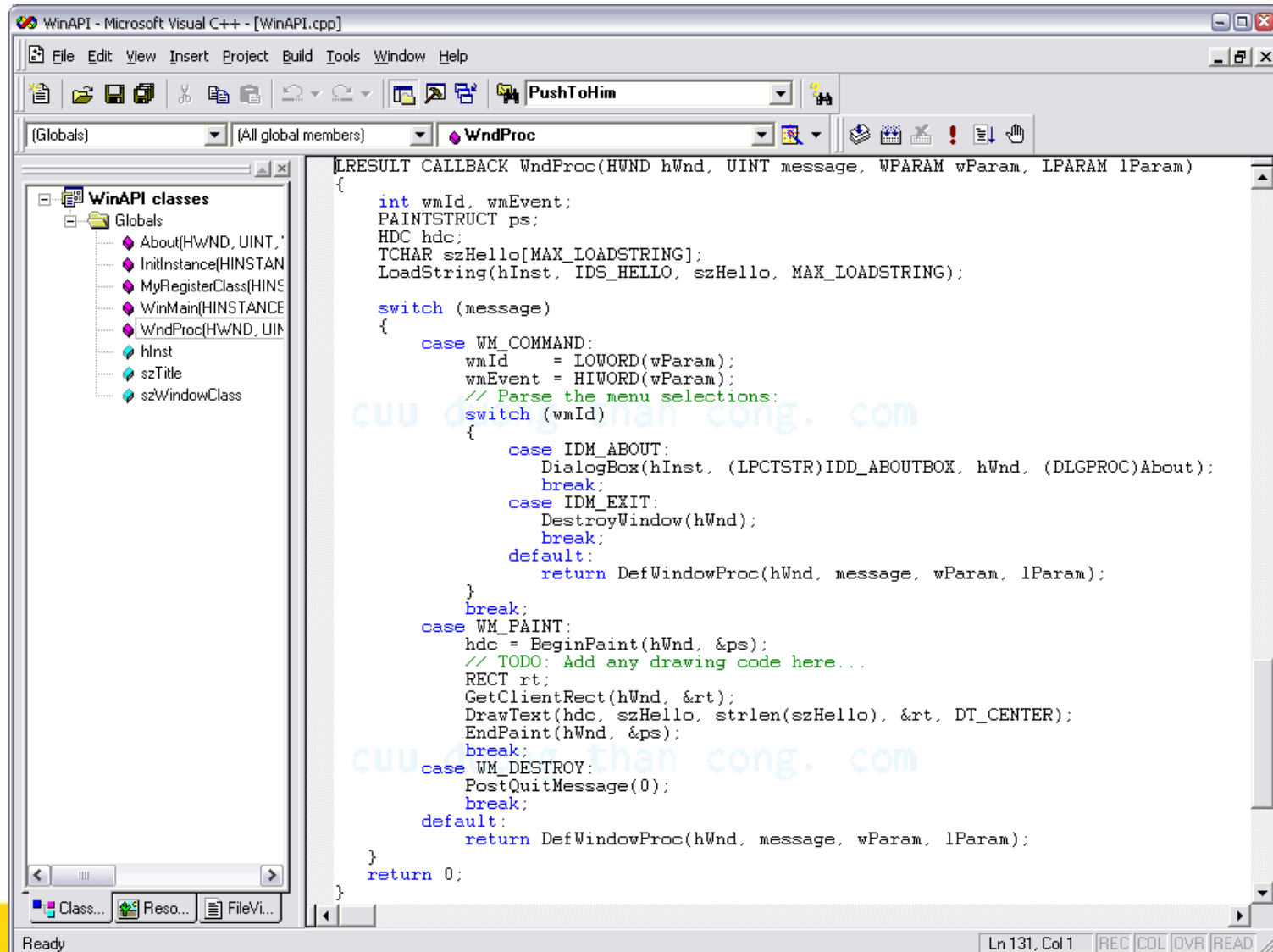
Ứng dụng Win API – hàm đăng ký lớp cửa sổ



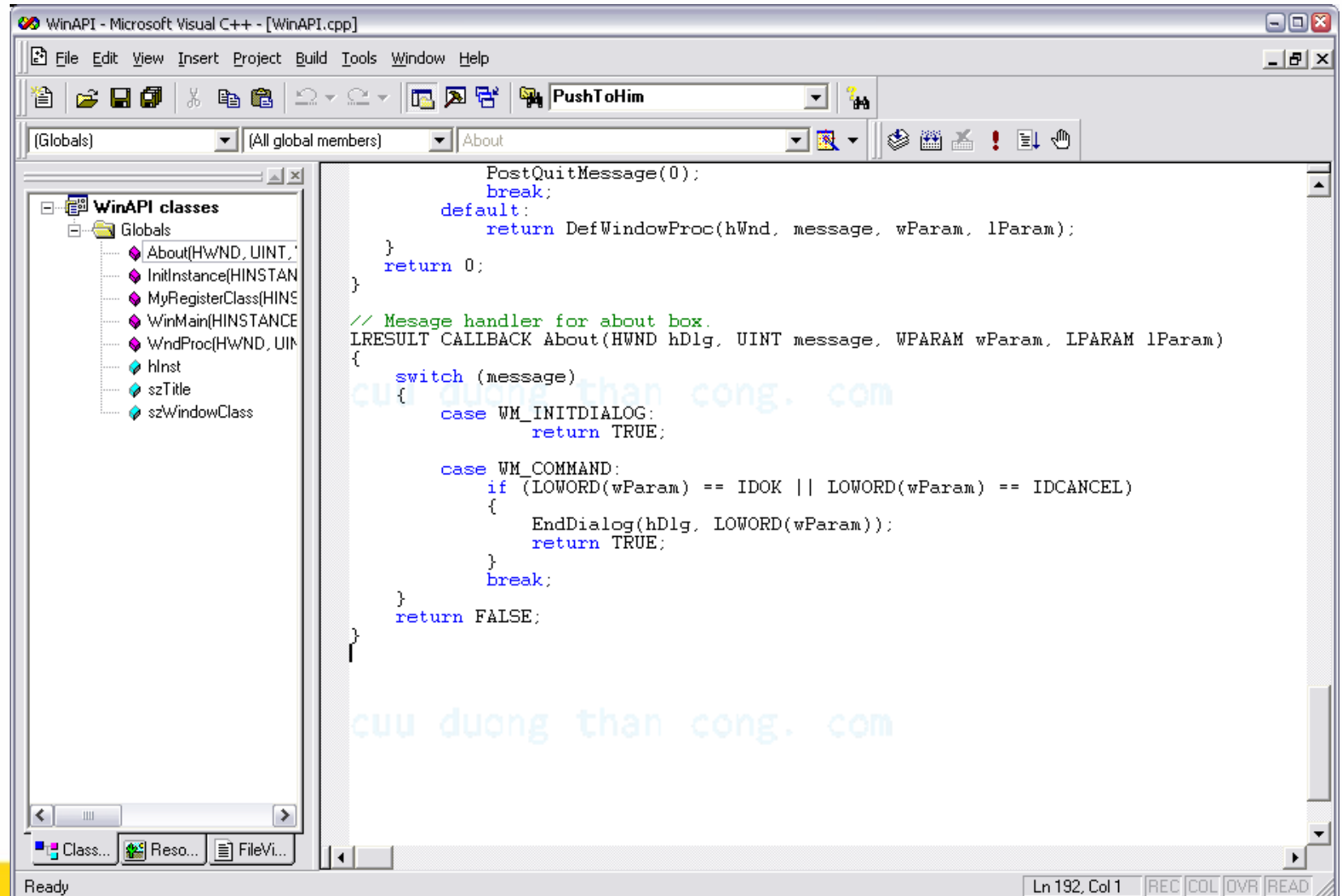
Ứng dụng Win API – hàm tạo cửa sổ



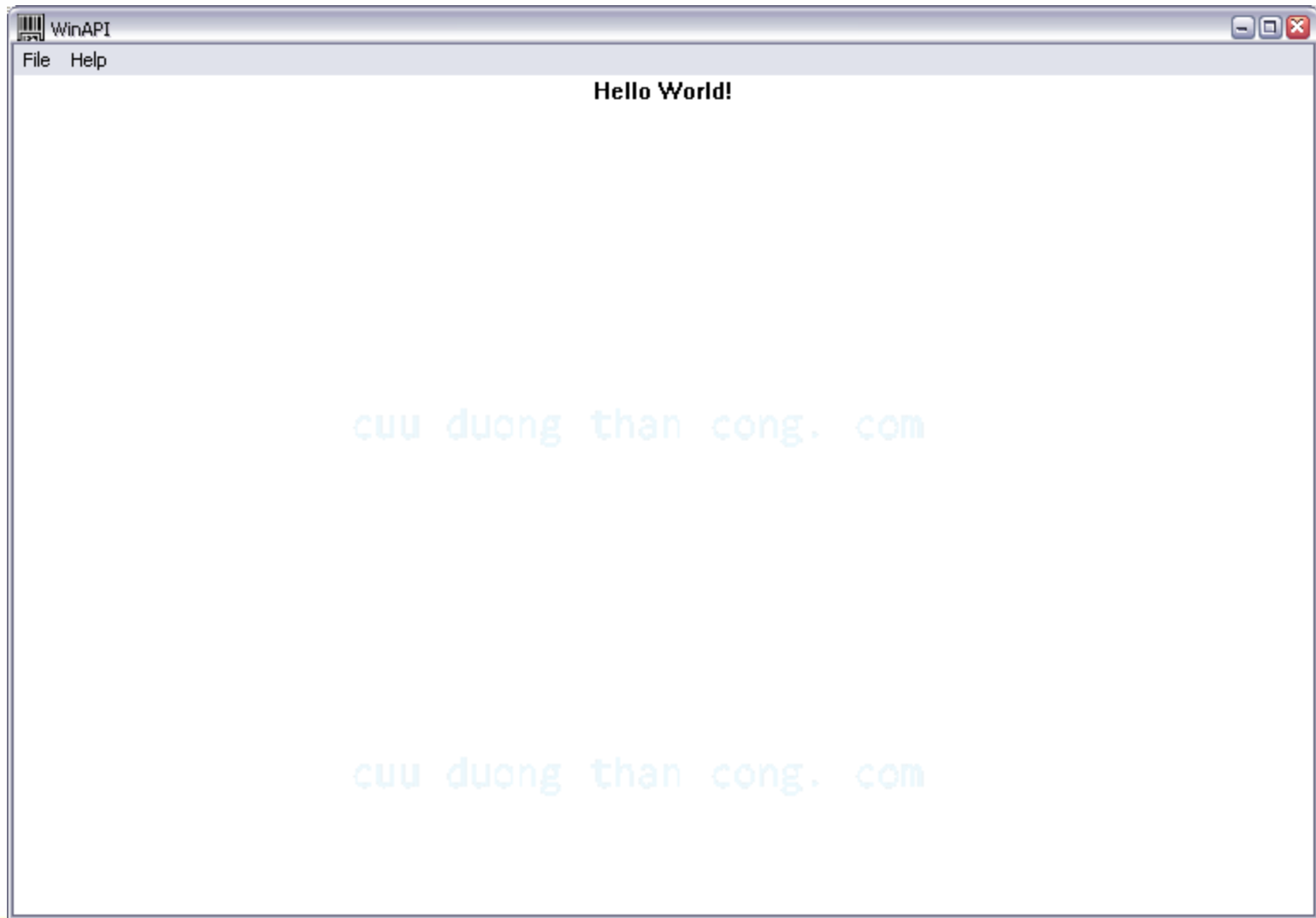
Ứng dụng Win API – hàm xử lý thông điệp cửa sổ



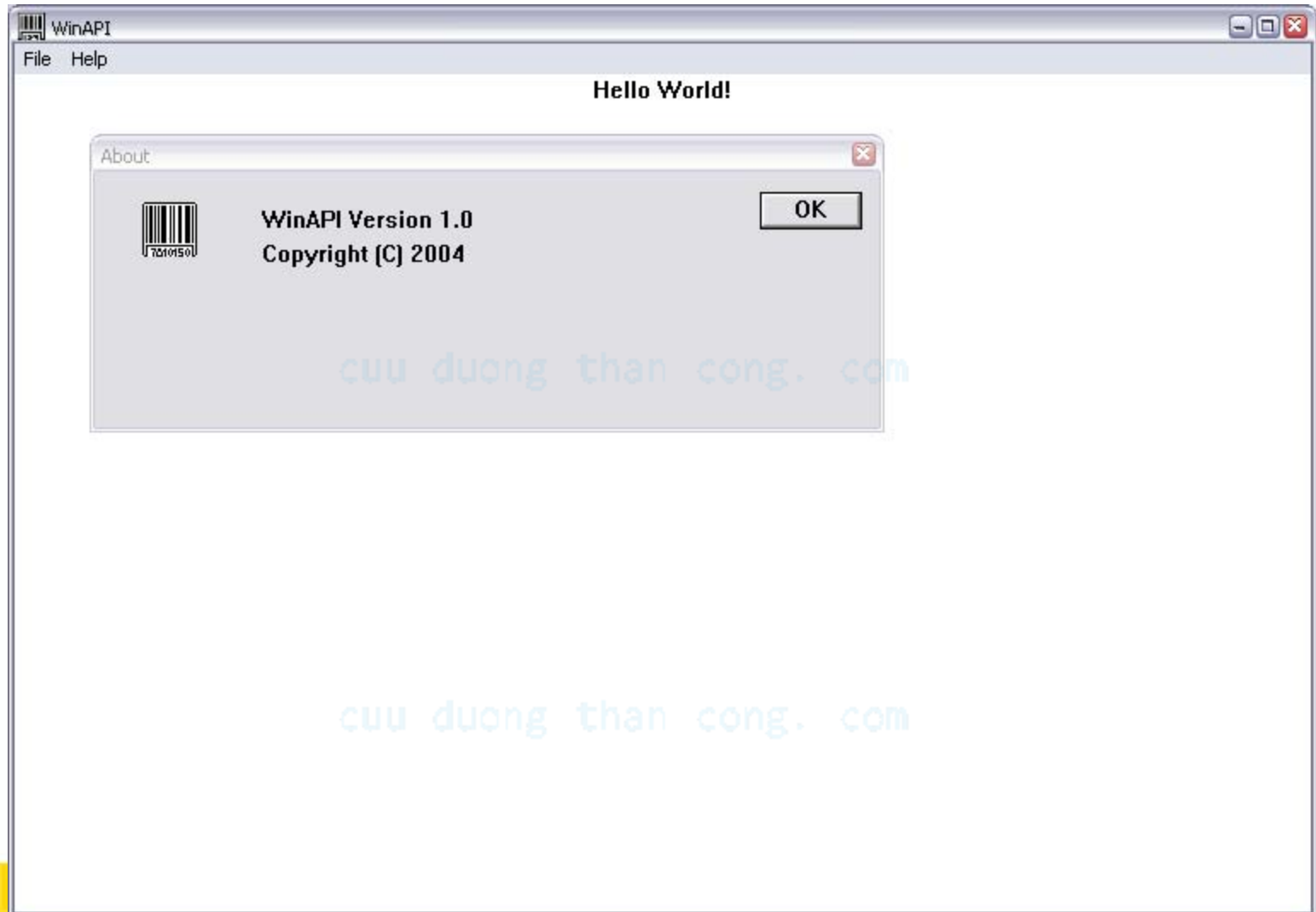
Ứng dụng Win API – hàm xử lý hộp thoại About



Ứng dụng Win API – kết quả chạy chương trình

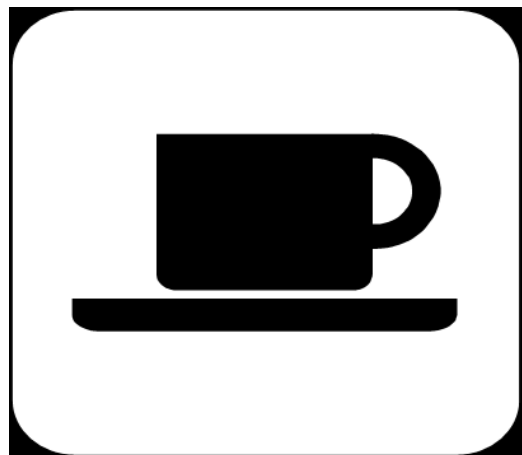


Ứng dụng Win API – kết quả chạy chương trình





cuu duong nhan cong. com



cuu duong nhan cong. com



KỸ THUẬT LẬP TRÌNH VISUAL C++ (MFC – Microsoft Foundation Classes)

- ❖ Giới thiệu về MFC, Visual C++ 6.0
- ❖ Các loại ứng dụng trong Visual C++ 6.0
- ❖ Xử lý chuột, bàn phím
- ❖ MessageBox

NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Giới thiệu về MFC (1/2)

- ❖ **MFC** – Microsoft Foundation Class
- ❖ **MFC** là một thư viện các lớp (class, OOP) trong ngôn ngữ Visual C++, được sử dụng để xây dựng các ứng dụng trên Windows
- ❖ **MFC** được xây dựng trên cơ sở các hàm thư viện API của Windows
- ❖ Trong một ứng dụng MFC, ta sẽ không gọi trực tiếp các hàm **Windows API** mà sẽ tạo các object từ những lớp của MFC, sau đó gọi các hàm của object này
- ❖ Đa số các hàm của lớp MFC có cùng tên với những hàm Windows API

Giới thiệu về MFC (2/2)

- ❖ **MFC** tạo ra một Application Framework giúp cho lập trình viên thiết kế kiến trúc của ứng dụng nhất quán và khoa học hơn
- ❖ Thông qua các class trong MFC, người lập trình có thể xây dựng ứng dụng nhanh và ít tốn công sức hơn so với việc sử dụng đơn thuần các hàm thư viện API của Windows
- ❖ Trong MFC ta vẫn có thể gọi các hàm Windows API
- ❖ MFC có nhiều version 1, 2, 3,..6. Version 6.0 có khoảng 200 class

Một số tính năng của MFC (1/4)

❖ Version 1: (Visual C++ 1.0)

- Các lớp List, Array, String, Time, Date, File access,...
- Các lớp giao diện cơ bản
- MDI, OLE 1.0

❖ Version 2: (Visual C++ 1.0)

- File open, save
- Print preview, printing
- Scrolling window, Splitter window
- Tool bar, status bar
- Access to Visual Basic controls
- Context-sensitive help
- An improved interface to OLE version 1.0
- DLL

Một số tính năng của MFC (2/4)

❖ Version 2.5: (Visual C++ 1.5)

- Hỗ trợ kết nối ODBC (Open Database Connectivity) cho phép truy xuất đến các CSDL Access, FoxPro, SQL Server
- OLE 2.01

❖ Version 3: (Visual C++ 1.51)

- Hỗ trợ Tab dialog (property sheet)
- Docking tool bar

❖ Version 3.1: (Visual C++ 2.1)

- Hỗ trợ các control chuẩn của Windows 95
- ODBC Level 2
- Các lớp Winsock phục vụ lập trình TCP/IP

Một số tính năng của MFC (3/4)

❖ Version 4.0: (Visual C++ 4.0)

- Hỗ trợ kết nối DAO (Data Access Object)
- Windows 95 docking control bar
- Bổ sung thêm Treeview và RichEdit
- Các lớp đồng bộ hóa tiểu trình
- OLE control container

❖ Version 4.2: (Visual C++ 4.2)

- Các lớp WinInet
- Các lớp ActiveX document server
- Các tính năng mở rộng của ActiveX control
- Cải tiến một số tính năng của ODBC

Một số tính năng của MFC (4/4)

❖ Version 4.21: (Visual C++ 5.0)

- Sửa một số lỗi trong version 4.2
- Redesigned IDE, Developer Studio 97
- Hỗ trợ lập trình COM (Component Object Model)
- ...

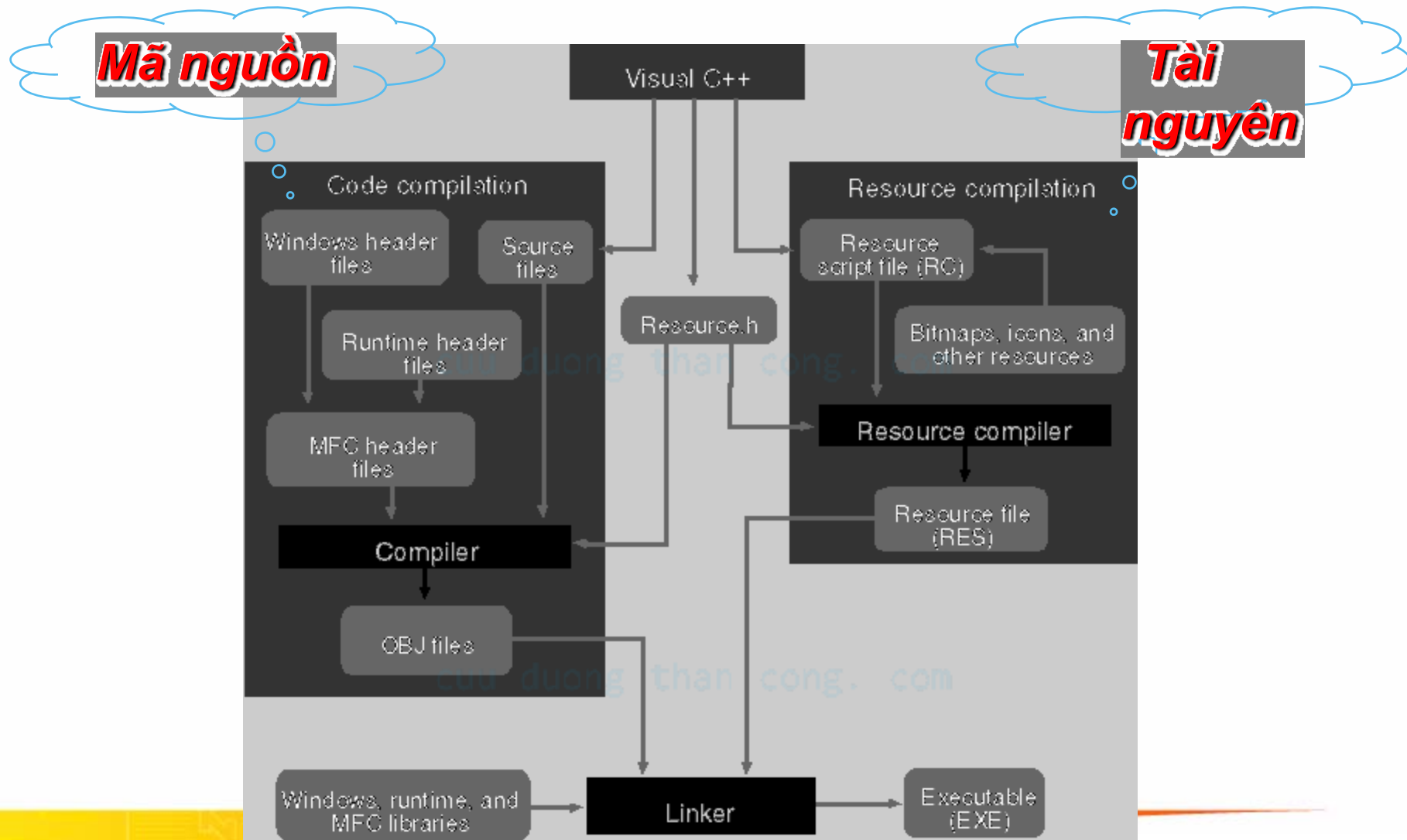
❖ Version 6.0: (Visual C++ 6.0)

- Hỗ trợ Các lớp cho những control chuẩn trong ID 4.0
- Hỗ trợ Dynamic HTML, cho phép tạo lập động các trang HTML
- Active Document Containment, cho phép ứng dụng MFC có thể chứa các Active document
- OLE DB và ADO

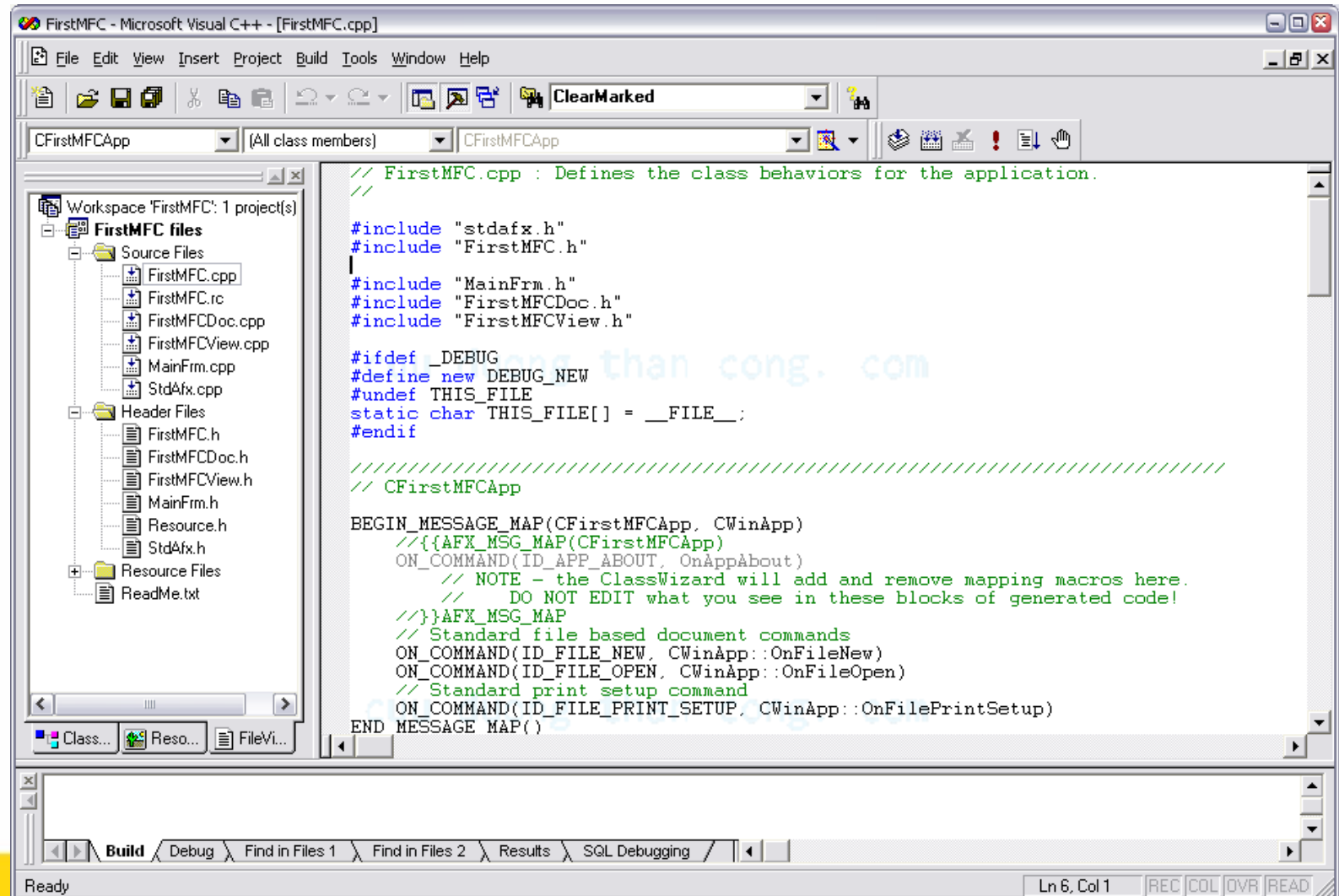
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Các thành phần trong ứng dụng Visual C++



Cửa sổ chính của MS Visual C++ 6.0



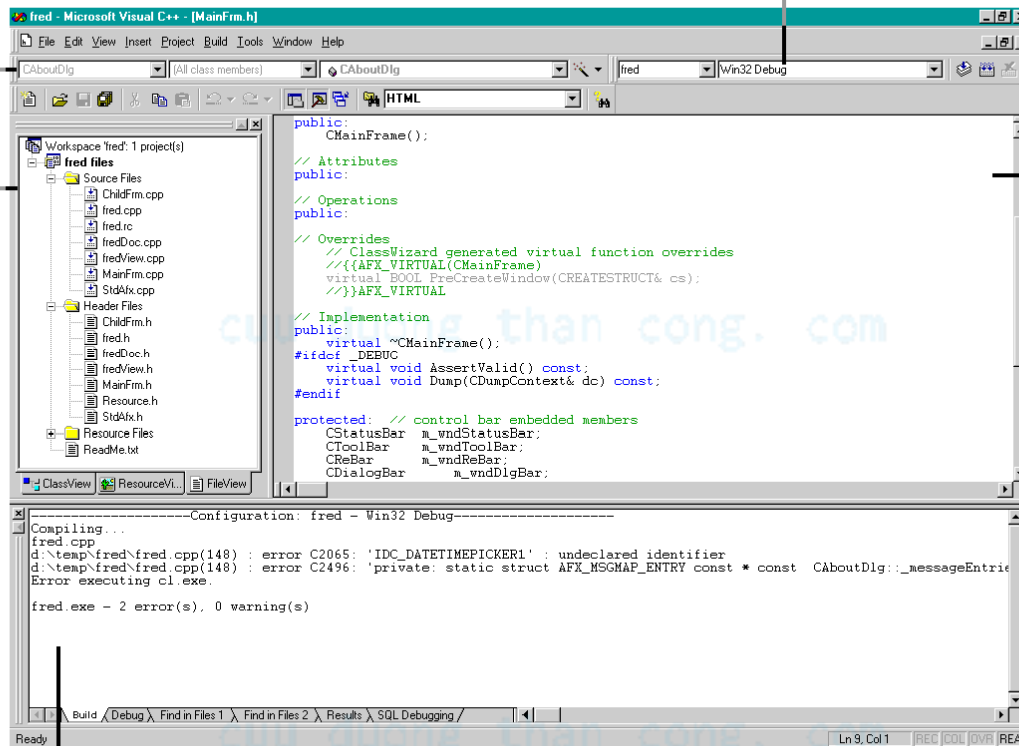
Cửa sổ chính của MS Visual C++ 6.0

Workspace window

WizardBar

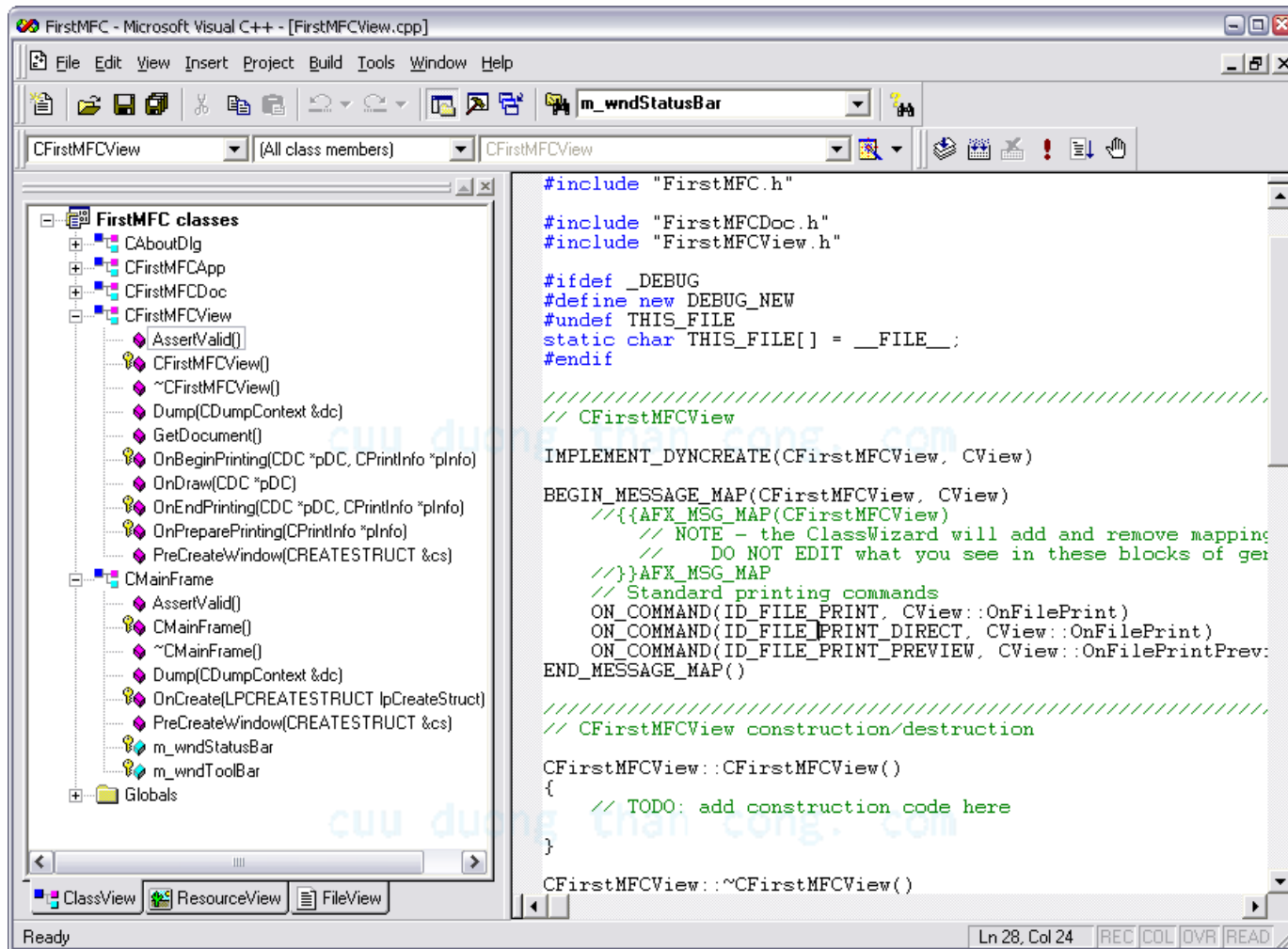
Source code

Build target

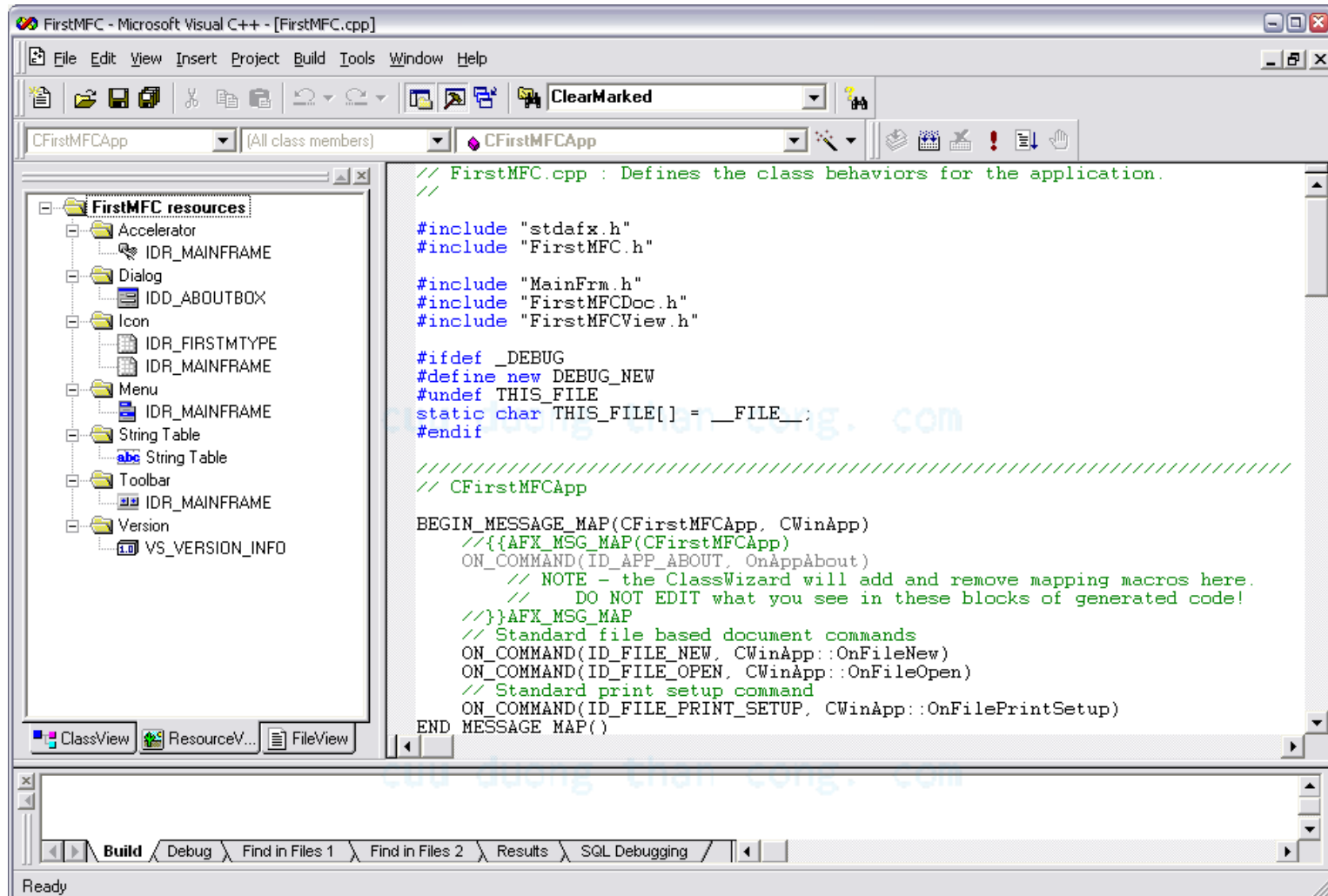


Compiler and linker output messages

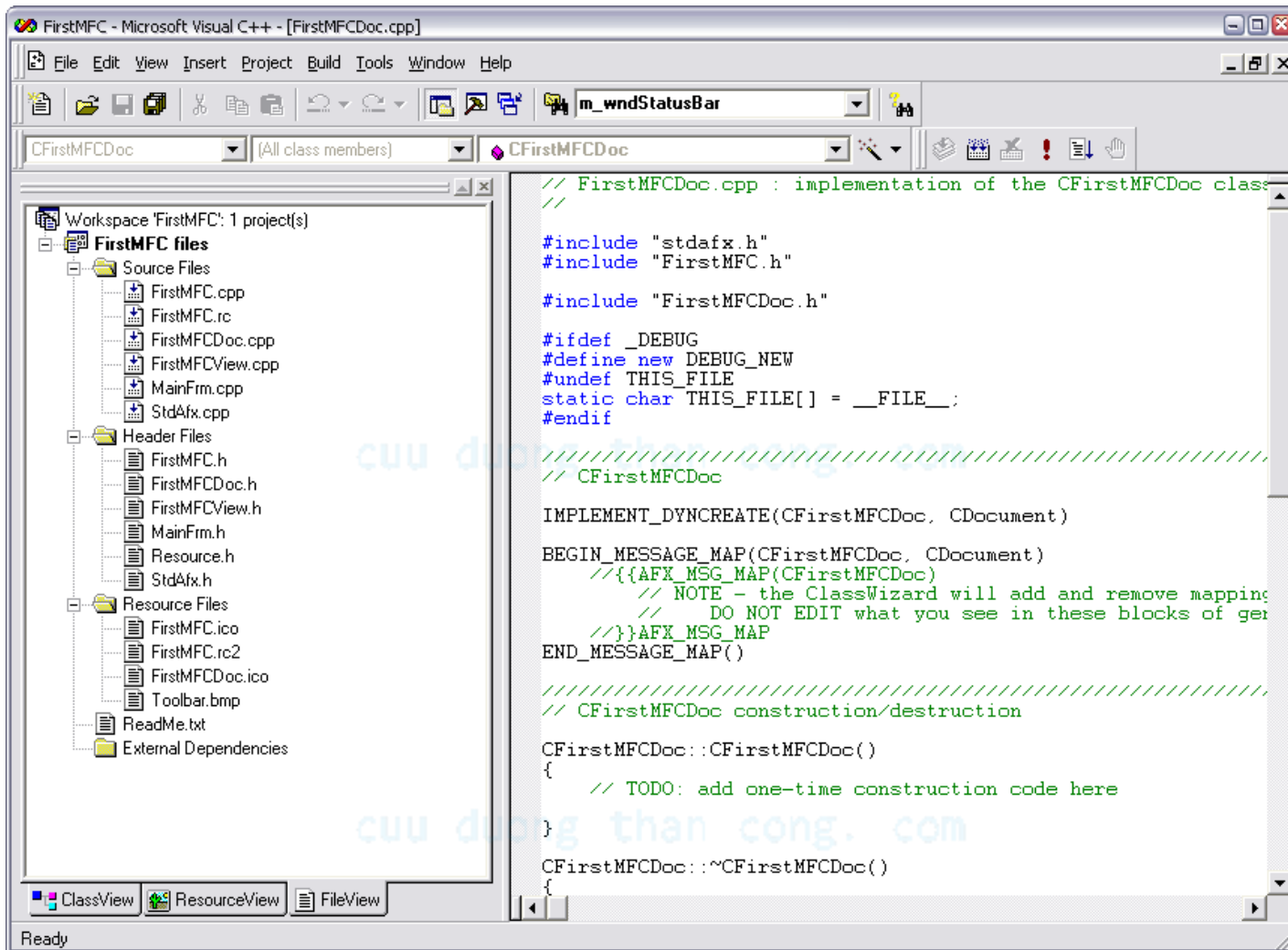
MS Visual C++ 6.0 – Class View



MS Visual C++ 6.0 – Resource View



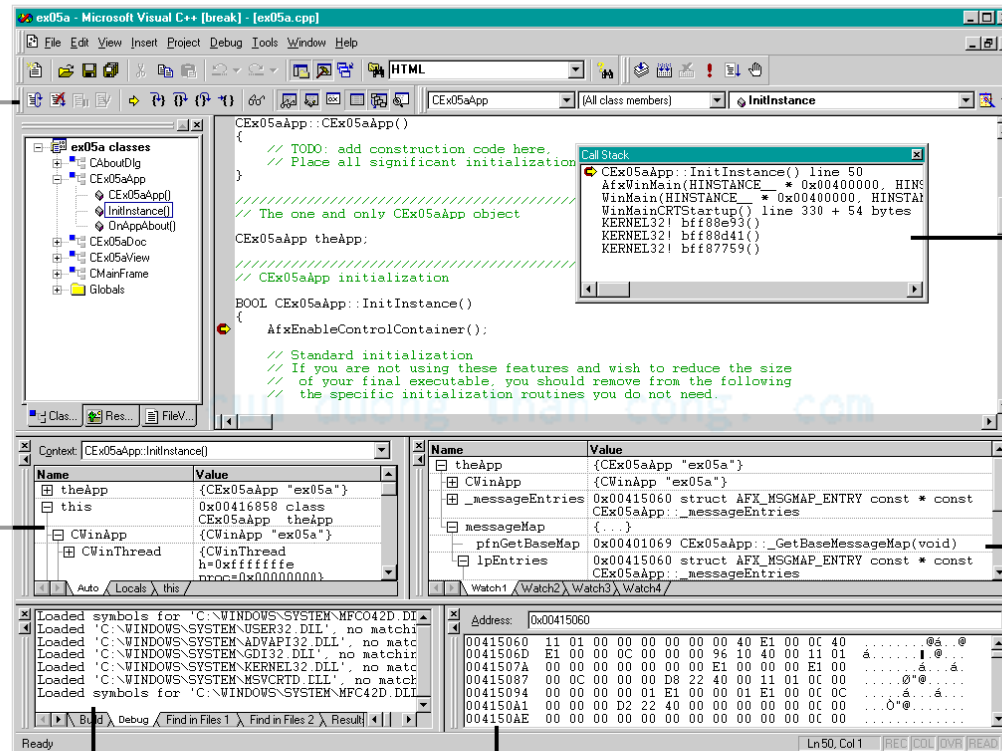
MS Visual C++ 6.0 – File View



Cửa sổ Debugger trong MS Visual C++ 6.0

Debug toolbar

Call stack



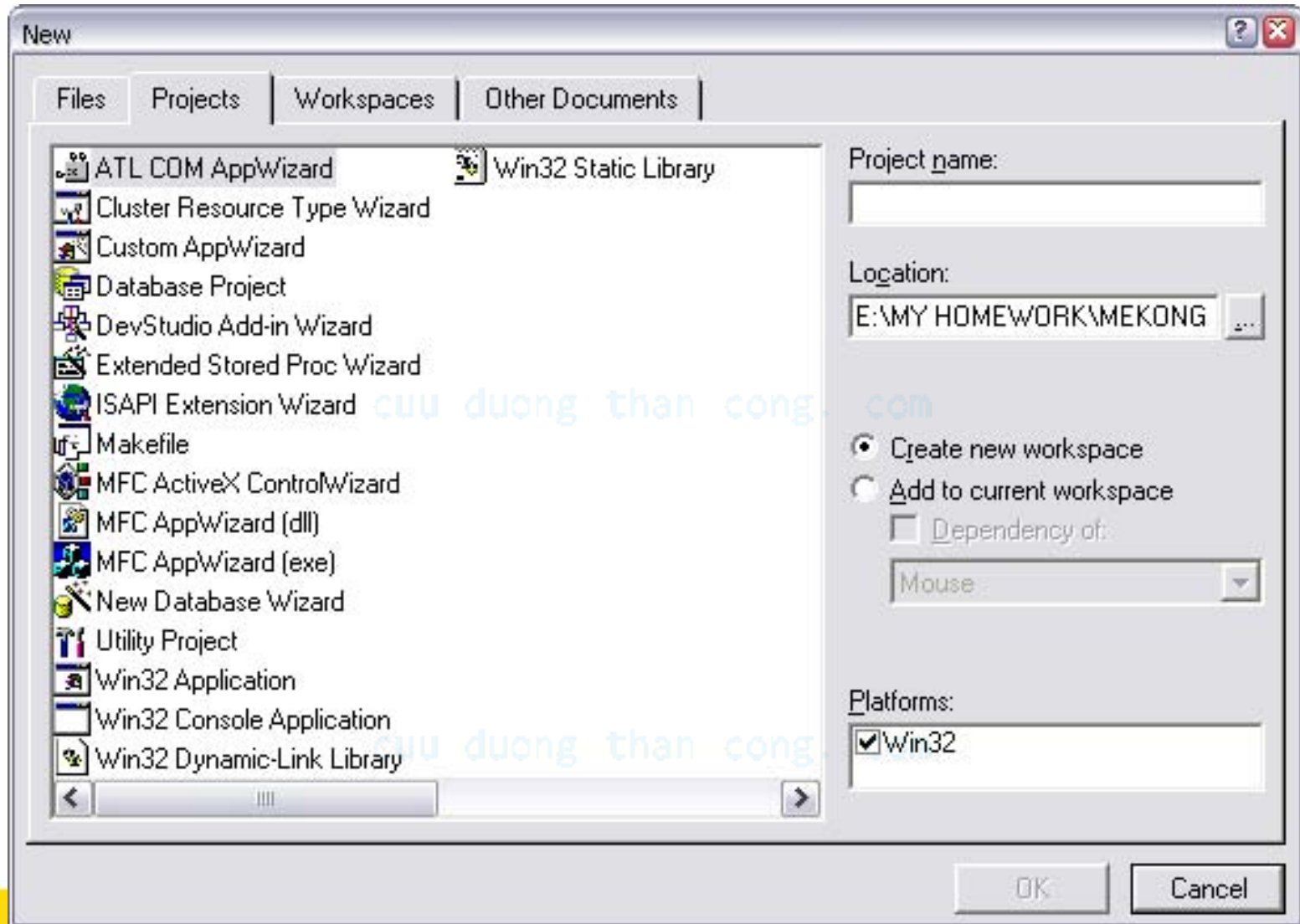
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Các loại ứng dụng (1/2)

- ❖ **Win32 Console:** ứng dụng 32 bits, với giao diện dạng DOS command line
- ❖ **Win32 (SDK):** ứng dụng 32 bits, chỉ sử dụng thư viện SDK
- ❖ **Win32 DLL:** ứng dụng 32 bits, dạng thư viện liên kết động (Dynamic – Linked Library) sử dụng SDK
- ❖ **Win32 LIB:** ứng dụng 32 bits, dạng thư viện liên kết tĩnh (Static – Linked Library) sử dụng SDK
- ❖ **MFC EXE:** ứng dụng 32 bits, sử dụng thư viện MFC
- ❖ **MFC DLL:** ứng dụng 32 bits, dạng thư viện liên kết động (Dynamic – Linked Library) sử dụng MFC

Các loại ứng dụng (2/2)



NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Ứng dụng MFC đầu tiên – Hello.h

```
class CMyApp : public CWinApp
{
    public:
        virtual BOOL InitInstance ();
};
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint ();
        DECLARE_MESSAGE_MAP ()
};
```


Ứng dụng MFC đầu tiên – Hello.cpp

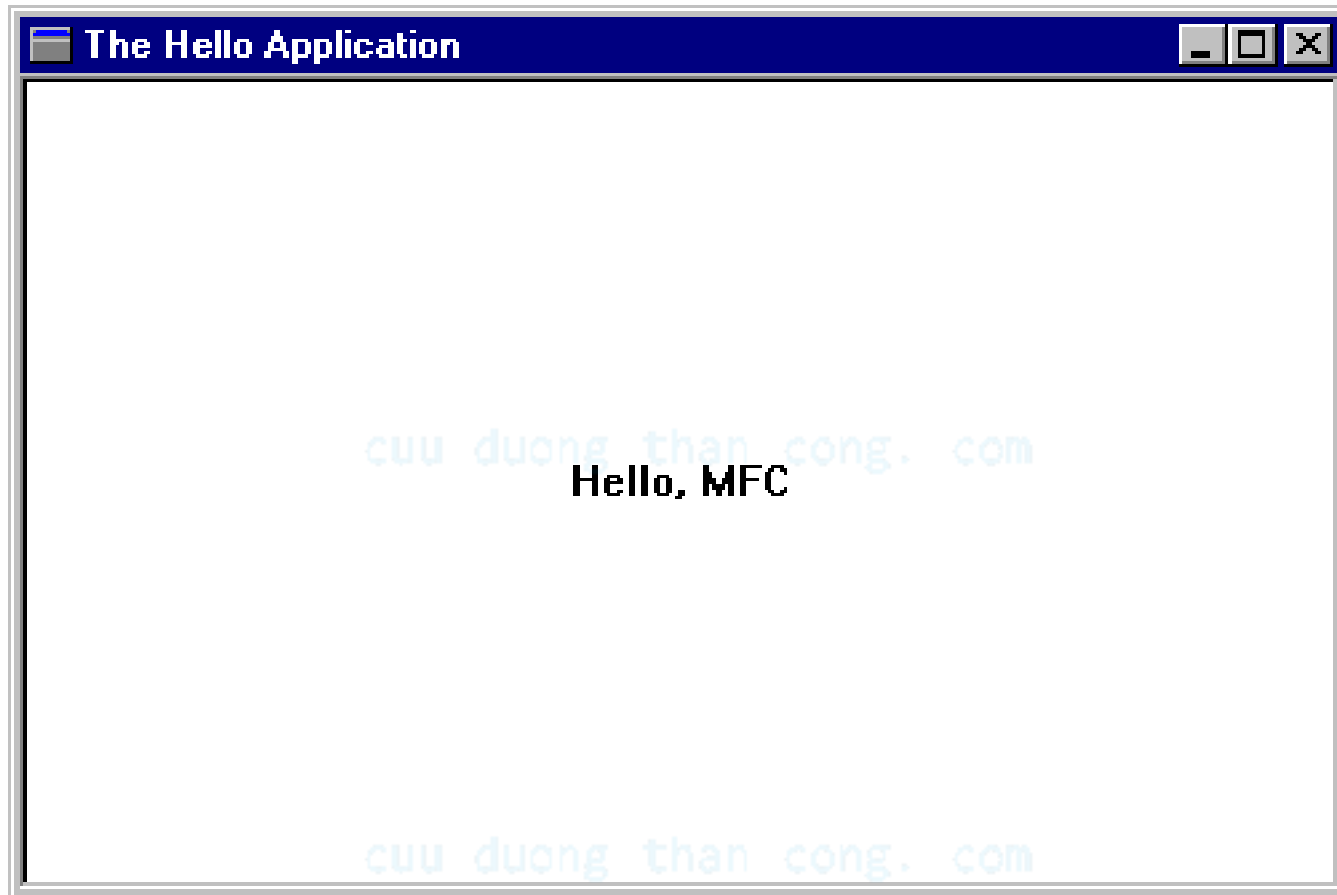
```
#include <afxwin.h>
#include "Hello.h"
CMyApp myApp;
////////////////////////////////////
// CMyApp member functions
BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow;
    m_pMainWnd->ShowWindow (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
} //////////////////////////////////////
```

Ứng dụng MFC đầu tiên – Hello.cpp (tt)

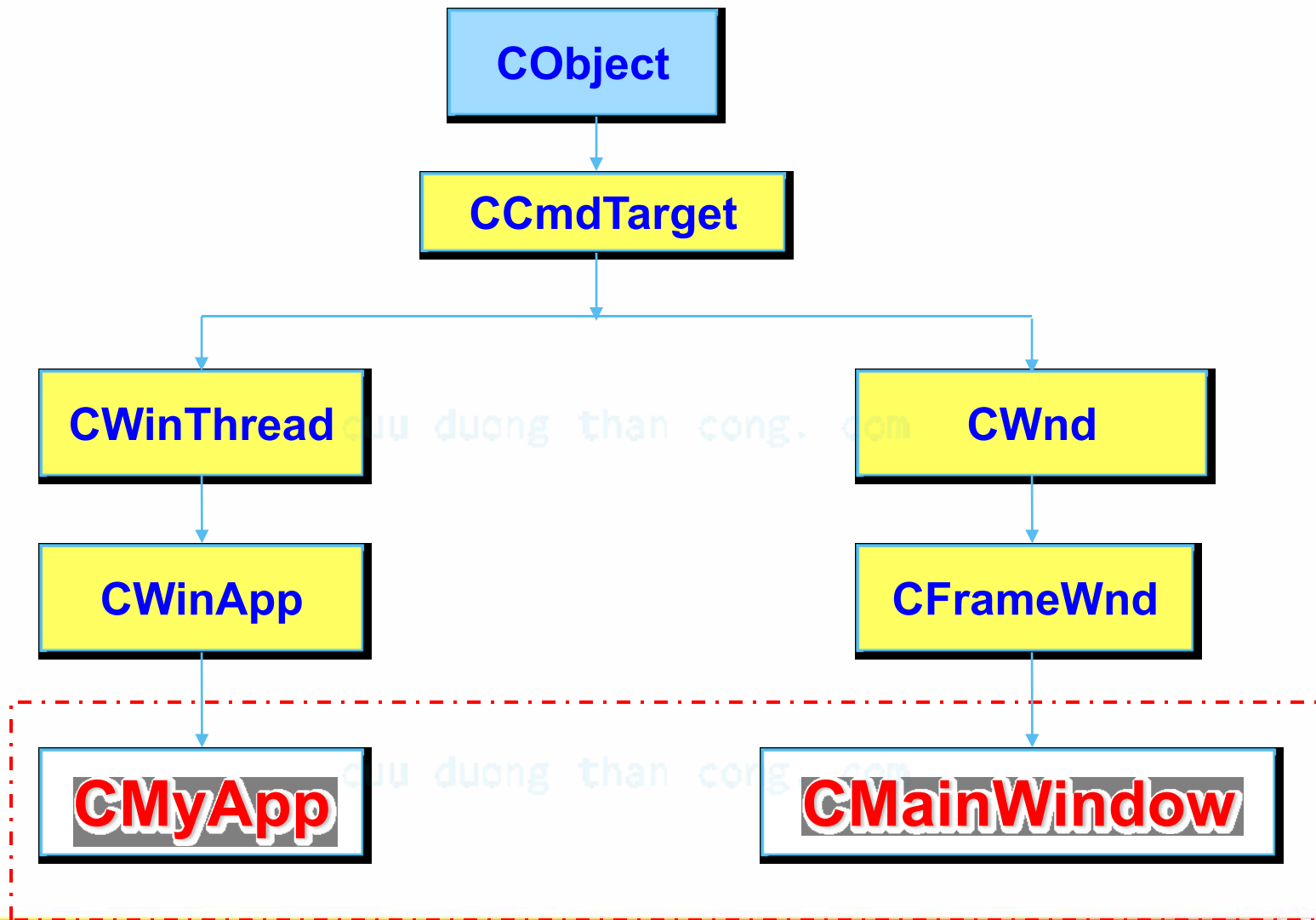
```
// CMainWindow message map and member functions
BEGIN_MESSAGE_MAP (CMainWindow, CFrameWnd)
ON_WM_PAINT ()
END_MESSAGE_MAP ()

CMainWindow::CMainWindow ()
{
    Create (NULL, _T ("The Hello Application"));
}
void CMainWindow::OnPaint ()
{
    CPaintDC dc (this);
    CRect rect;
    GetClientRect (&rect);
    dc.DrawText (_T ("Hello, MFC"), -1, &rect, DT_SINGLELINE |
DT_CENTER | DT_VCENTER);
}
```

Kết quả chương trình Hello



Các class của ứng dụng Hello



Các bước thực thi ứng dụng

```
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
    int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;

    MyRegisterClass(hInstance);
    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }
    // Main message loop:
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}
```

Đăng ký lớp cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

CMyApp

```
BOOL CMyApp::InitInstance ()
```

```
{
```

```
    m_pMainWnd = new CMainWindow;
```

```
    m_pMainWnd->ShowWindow  
    (m_nCmdShow);
```

```
    m_pMainWnd->UpdateWindow ();
```

```
    return TRUE;
```

```
}
```

```
ATOM MyRegisterClass(HINSTANCE hInstance)
```

```
{
```

```
    WNDCLASSEX wcex;
```

```
    wcex.cbSize = sizeof(WNDCLASSEX);
```

```
    wcex.style = CS_HREDRAW | CS_VREDRAW;
```

```
    wcex.lpfnWndProc = (WNDPROC)WndProc;
```

```
    wcex.cbClsExtra = 0;
```

```
    wcex.cbWndExtra = 0;
```

```
    wcex.hInstance = hInstance;
```

```
    wcex.hIcon = LoadIcon(hInstance, (LPCTSTR)IDI_BAI02);
```

```
    wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
```

```
    wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW+1);
```

```
    wcex.lpszMenuName = (LPCSTR)IDC_BAI02;
```

```
    wcex.lpszClassName = szWindowClass;
```

```
    wcex.hIconSm = LoadIcon(wcex.hInstance, (LPCTSTR)IDI_SMALL);
```

```
    return RegisterClassEx(&wcex);
```

```
}
```

Tạo lập cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

CMainWindow

```
CMainWindow::CMainWindow ()  
{  
    Create (NULL, _T ("The Hello Application"));  
}
```

```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  
{  
    HWND hWnd;  
  
    hInst = hInstance; // Store instance handle in our global variable  
  
    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,  
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);  
  
    if (!hWnd)  
    {  
        return FALSE;  
    }  
  
    ShowWindow(hWnd, nCmdShow);  
    UpdateWindow(hWnd);  
  
    return TRUE;  
}
```

Hàm tạo lập cửa sổ

CMainWindow

```
CMainWindow::CMainWindow ()  
{  
    Create (NULL, _T ("The Hello Application"));  
}
```

cuu duong than cong. com

```
BOOL Create (LPCTSTR lpszClassName,  
             LPCTSTR lpszWindowName,  
             DWORD dwStyle = WS_OVERLAPPEDWINDOW,  
             const RECT& rect = rectDefault,  
             CWnd* pParentWnd = NULL,  
             LPCTSTR lpszMenuName = NULL,  
             DWORD dwExStyle = 0,  
             CCreateContext* pContext = NULL)
```

cuu duong than cong. com

Các tham số trong hàm tạo lập cửa sổ

Thuộc tính	Ý nghĩa
lpszClassName	Tên lớp cửa sổ
lpszWindowName	Tiêu đề cửa sổ
<u>dwStyle</u>	Kiểu dáng cửa sổ
rect	Kích thước và vị trí ban đầu của cửa sổ
pParentWnd	Cửa sổ cha, NULL cửa sổ không có cửa sổ cha
lpszMenuName	Thực đơn liên quan đến cửa sổ, NULL nghĩa là cửa sổ không có Menu
dwExStyle	Kiểu mở rộng của cửa sổ
pContext	Sử dụng trong ứng dụng Documents/Views

Hiển thị cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;

    hInst = hInstance; // Store instance handle in our global variable

    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

    if (!hWnd)
    {
        return FALSE;
    }

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}
```

CMainWindow

```
BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow;
    m_pMainWnd->ShowWindow
        (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
}
```

Vòng lặp và xử lý thông điệp

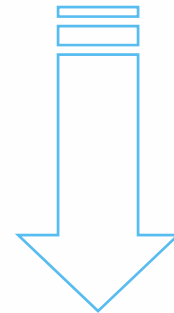
```
while (GetMessage (&msg, NULL, 0, 0))
{
    TranslateMessage (&msg);
    DispatchMessage (&msg);
}

LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;
    switch (message) {
        case WM_PAINT:
            hdc = BeginPaint (hwnd, &ps);
            Ellipse (hdc, 0, 0, 200, 100);
            EndPaint (hwnd, &ps);
            return 0;

        case WM_DESTROY:
            PostQuitMessage (0);
            return 0;
    }
    return DefWindowProc (hwnd, message, wParam, lParam);
}
```

CMainWindow

**Tiếp nhận và xử lý
thông điệp như
thế nào ?**



Message Map

Message Map – Ánh xạ thông điệp

- ❖ Ánh xạ thông điệp là bảng tương quan giữa thông điệp và các hàm thành phần quản lý các thông điệp đó
- ❖ Bất kỳ lớp nào suy diễn từ CCmdTarget đều có thể chứa ánh xạ thông điệp
- ❖ Ánh xạ thông điệp được khai báo như các thành phần lớp khác trong khai báo lớp

```
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint \(\);
        DECLARE_MESSAGE_MAP ()
};
```

Message Map (tt)

❖ Khai báo Message Map

```
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint ();
        afx_msg void OnLButtonDown(UINT, CPoint);
        DECLARE_MESSAGE_MAP ()
};
```

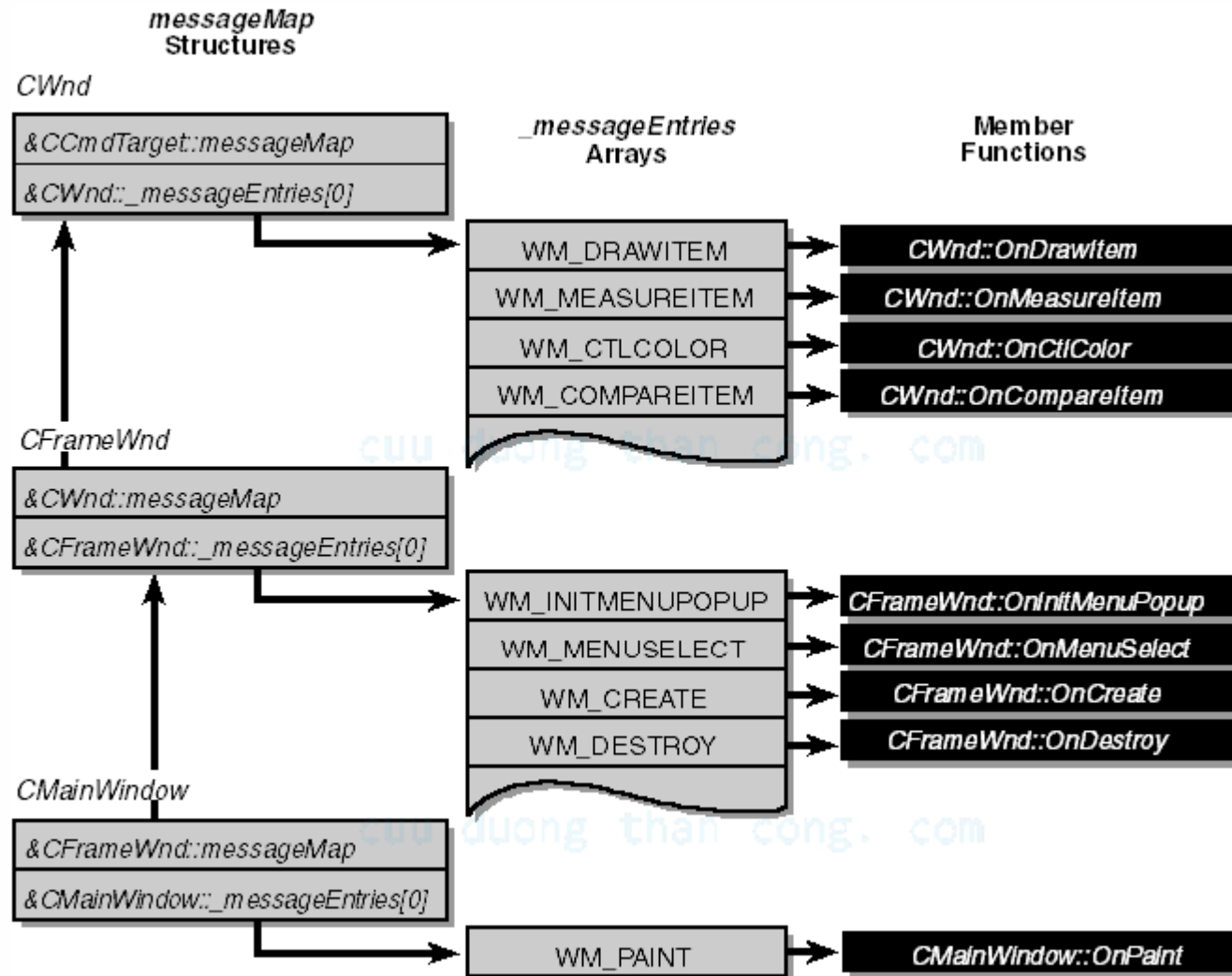
❖ Cài đặt xử lý Message

```
BEGIN_MESSAGE_MAP (CMainWindow, CFrameWnd)
    ON_WM_PAINT ()
    ON_WM_LBUTTONDOWN()
END_MESSAGE_MAP ()
Void CMainWindow::OnPaint()
{
}
Void CMainWindow::OnLButtonDown(UINT nFlags, CPoint point)
{
}
```

Thông điệp WM_XXX và hàm MFC tương ứng

Message	Afx function
WM_CHAR	afx_msg void OnChar(UINT,UINT,UINT)
WM_CREATE	afx_msg int OnCreate(LPCREATESTRUCT)
WM_DESTROY	afx_msg void OnDestroy()
WM_LBUTTONDOWN	afx_msg void OnLButtonDown(UINT, CPoint)
WM_LBUTTONUP	afx_msg void OnLButtonUp(UINT, CPoint)
WM_MOUSEMOVE	afx_msg void OnMouseMove(UINT, CPoint)
WM_RBUTTONDOWN	afx_msg void OnRButtonDown(UINT, CPoint)
WM_RBUTTONUP	afx_msg void OnRButtonUp(UINT, CPoint)
WM_SIZE	afx_msg void OnSize()
WM_PAINT	afx_msg void OnPaint()

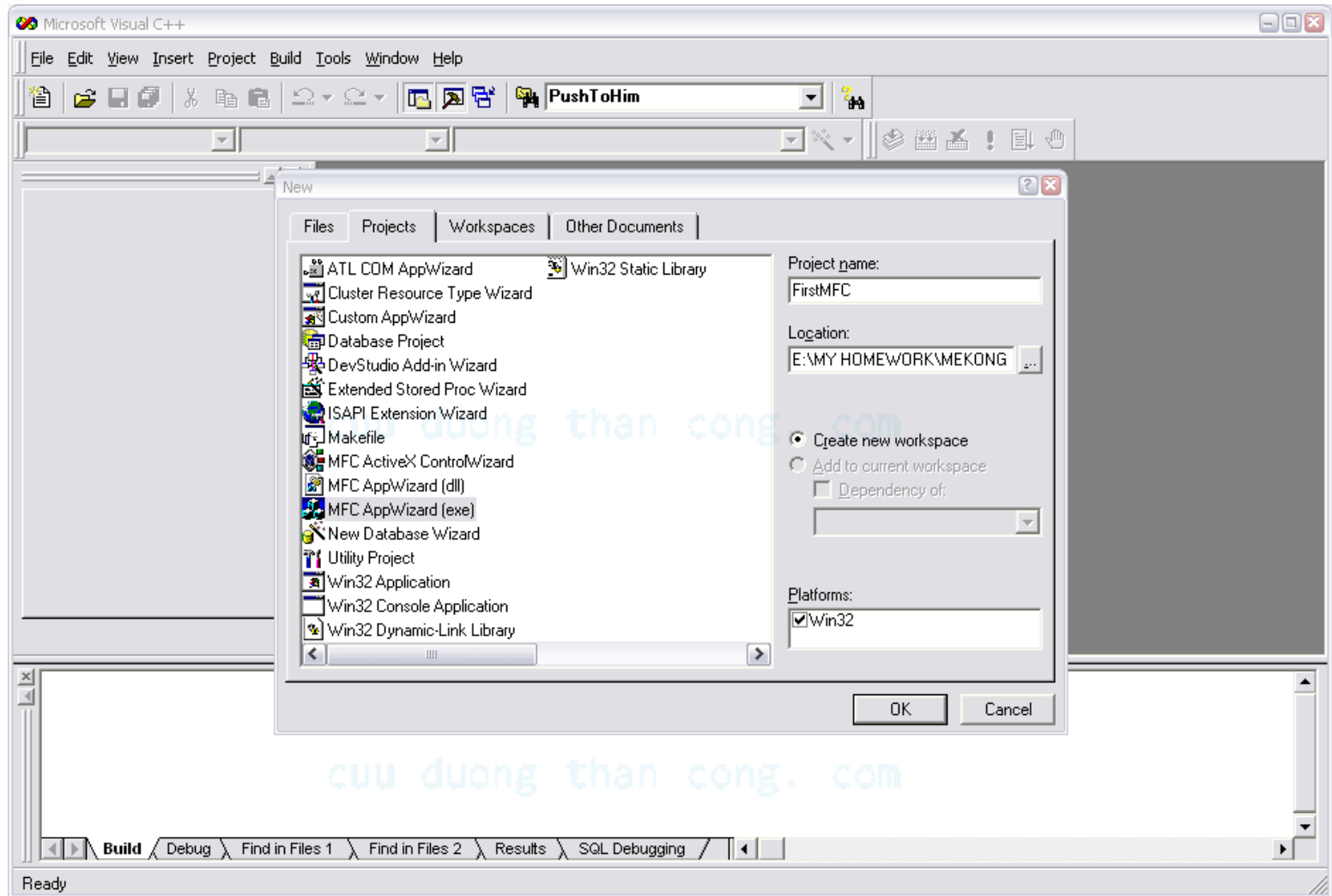
Xử lý thông điệp cửa sổ - Message Map



Tạo ứng dụng MFC bằng AppWizard

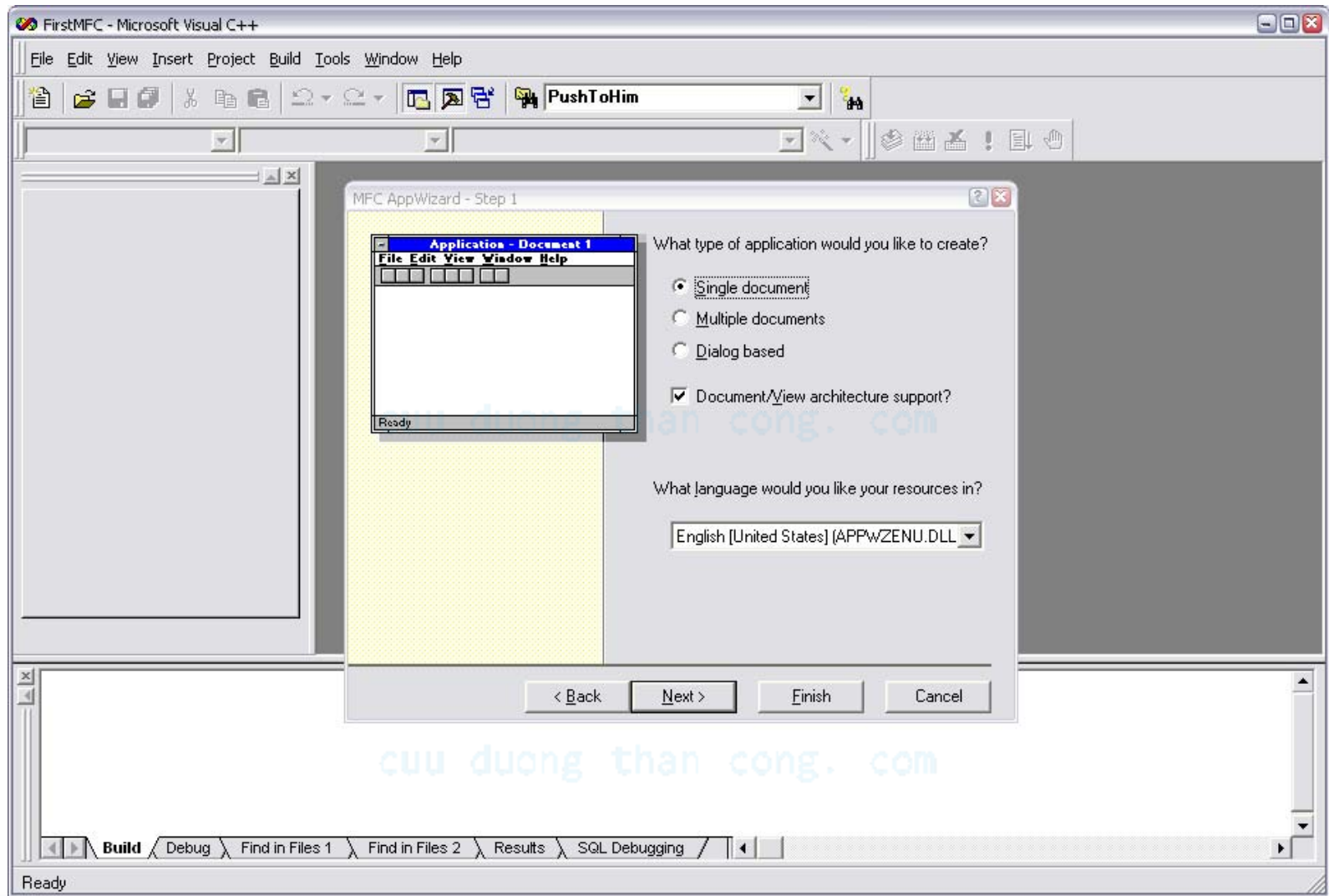
- ❖ Chọn menu **File → New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project **"MFC AppWizard (exe)"**
- ❖ Đặt tên project trong ô **"Project name"**
- ❖ Xác định đường dẫn thư mục trong **"Location"**
- ❖ Step 1: Chọn loại ứng dụng **"Single Document"**, bỏ option **"Document/View architecture support"**
- ❖ Nhấn **"Next"** đến **"Step 4 of 6"**

Tạo ứng dụng MFC bằng AppWizard



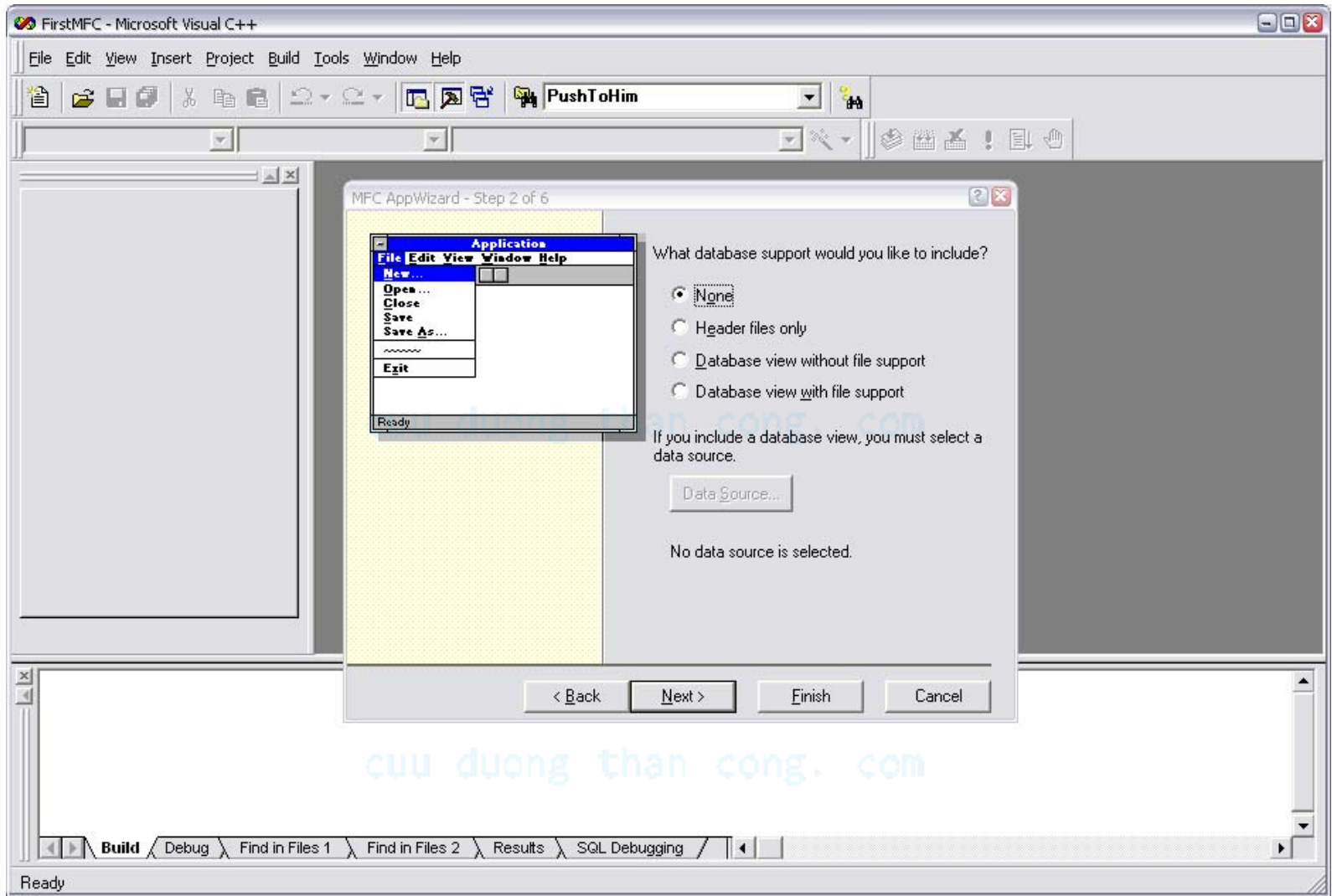
Chọn loại project “MFC AppWizard (EXE)”

Tạo ứng dụng MFC bằng AppWizard



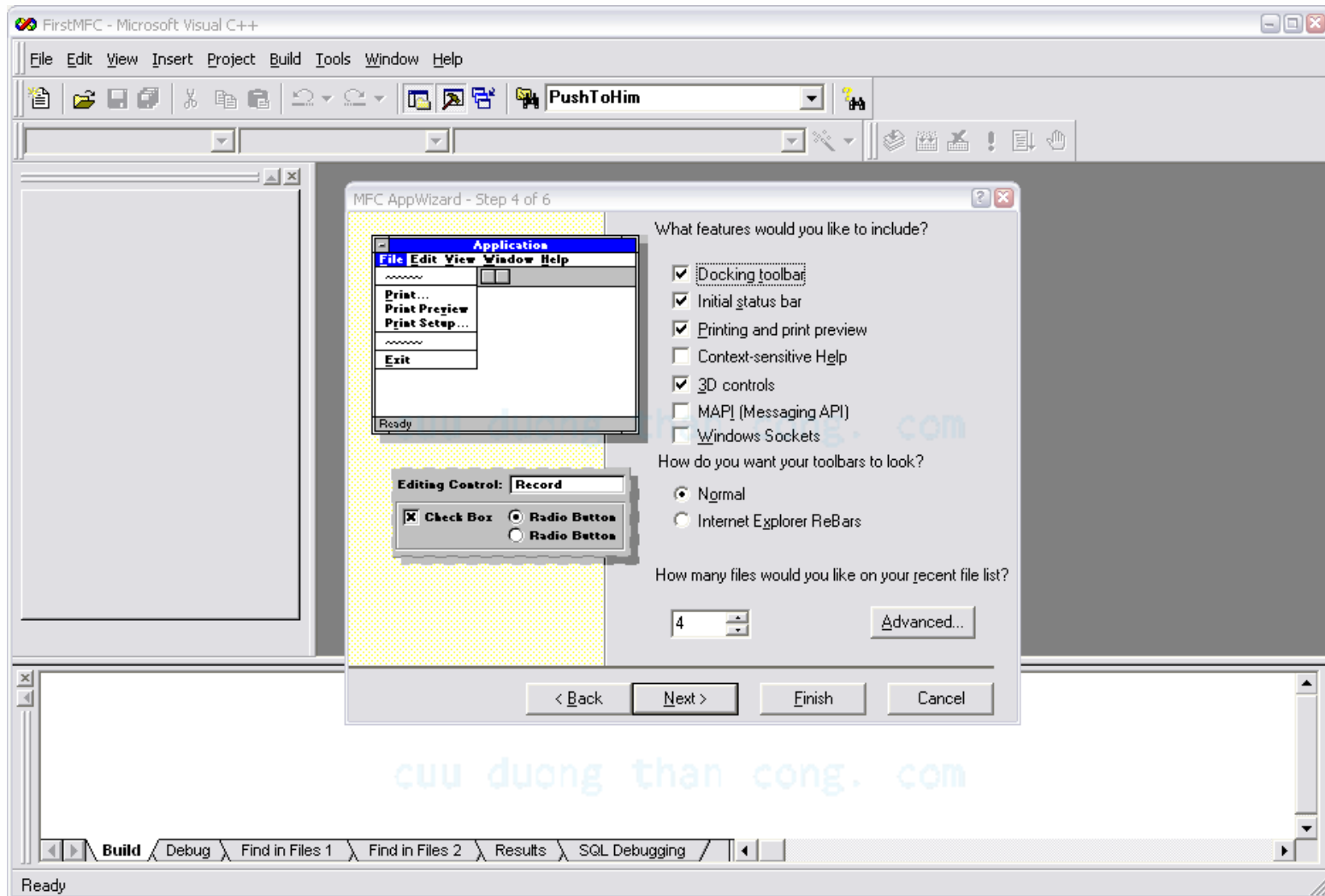
Chọn option “Single document”

Tạo ứng dụng MFC bằng AppWizard

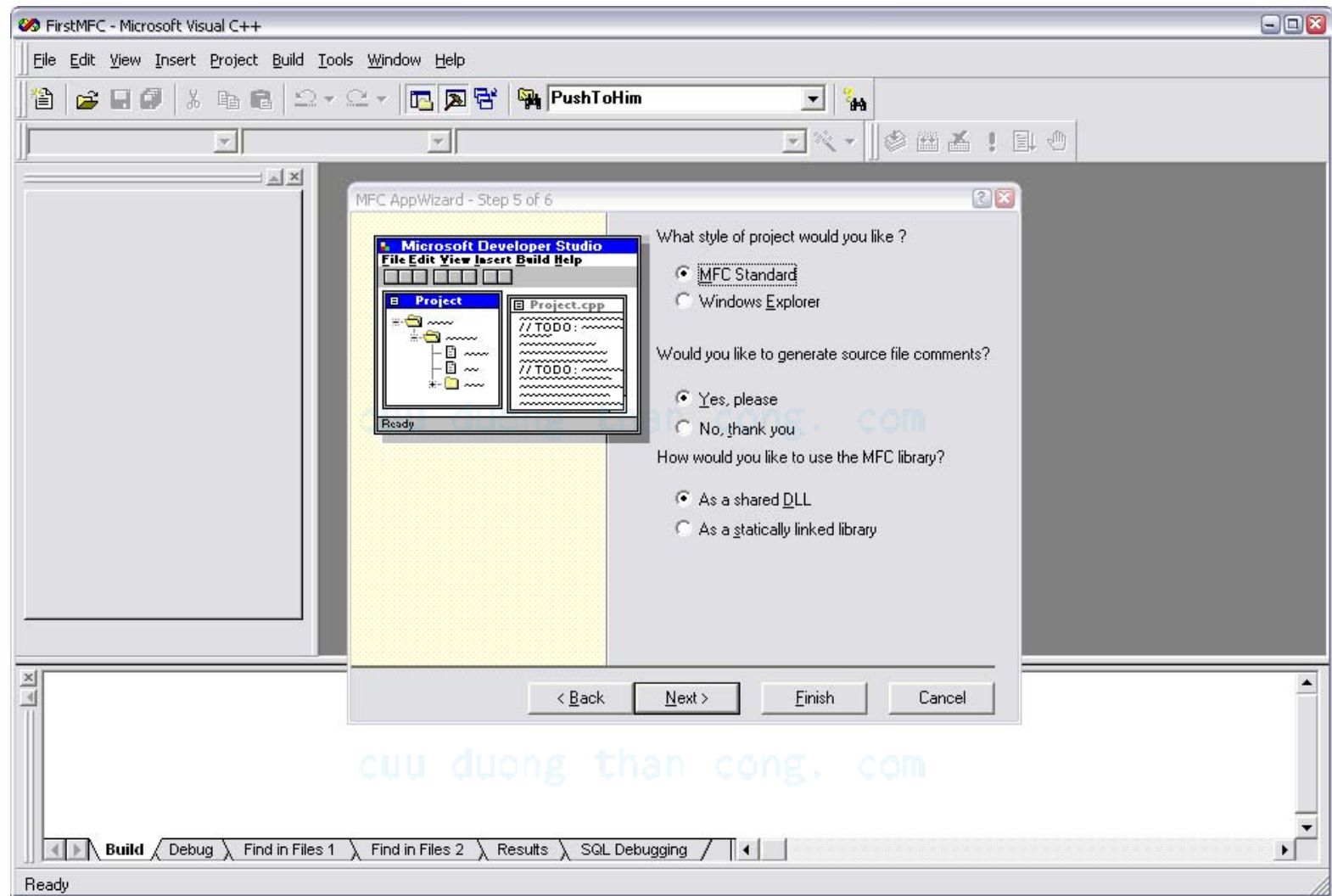


Chọn option "None"

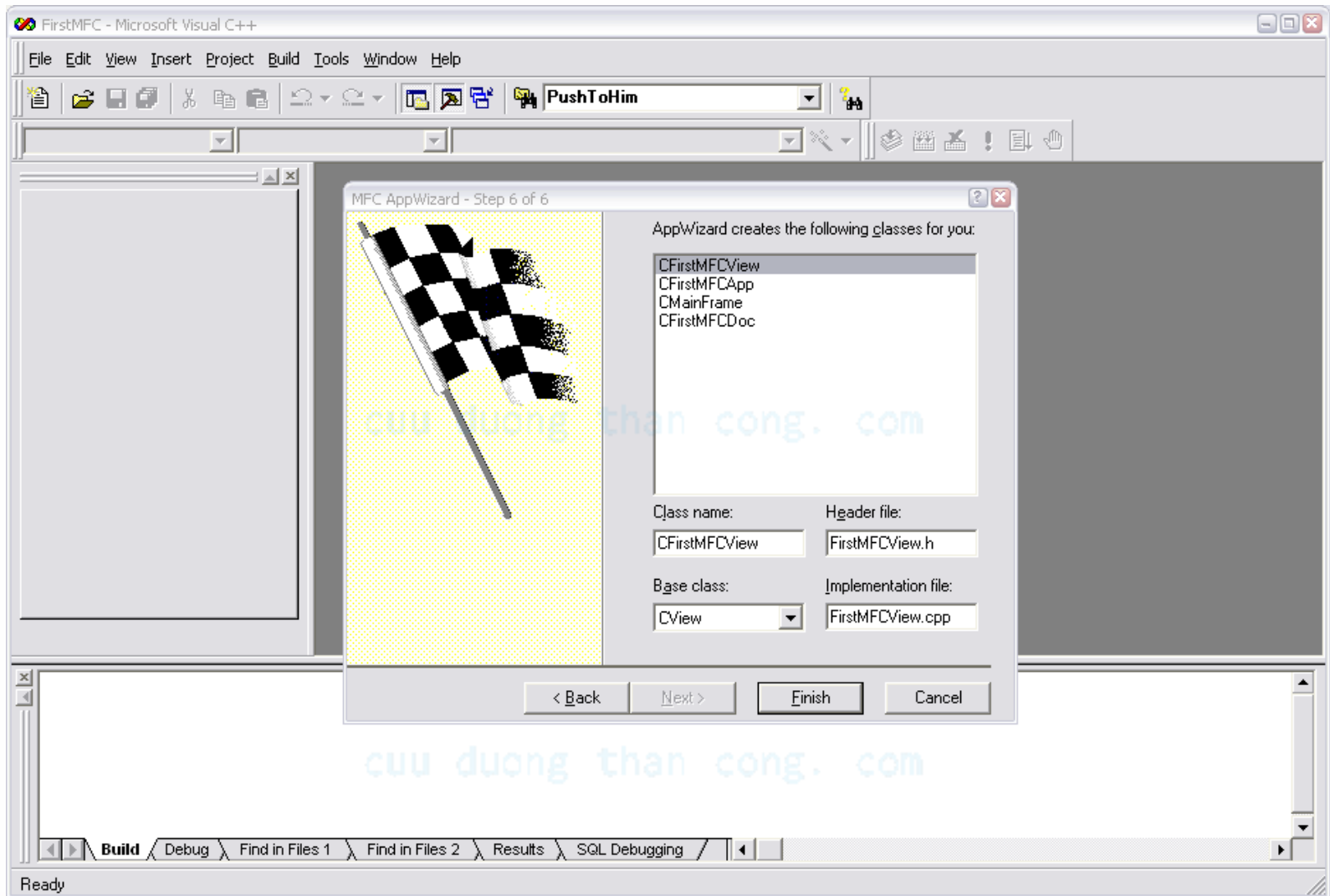
Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard

Application type of FirstMFC:

Single Document Interface Application targeting:
Win32

Classes to be created:

Application: CFirstMFCApp in FirstMFC.h and FirstMFC.cpp

Frame: CMainFrame in MainFrm.h and MainFrm.cpp

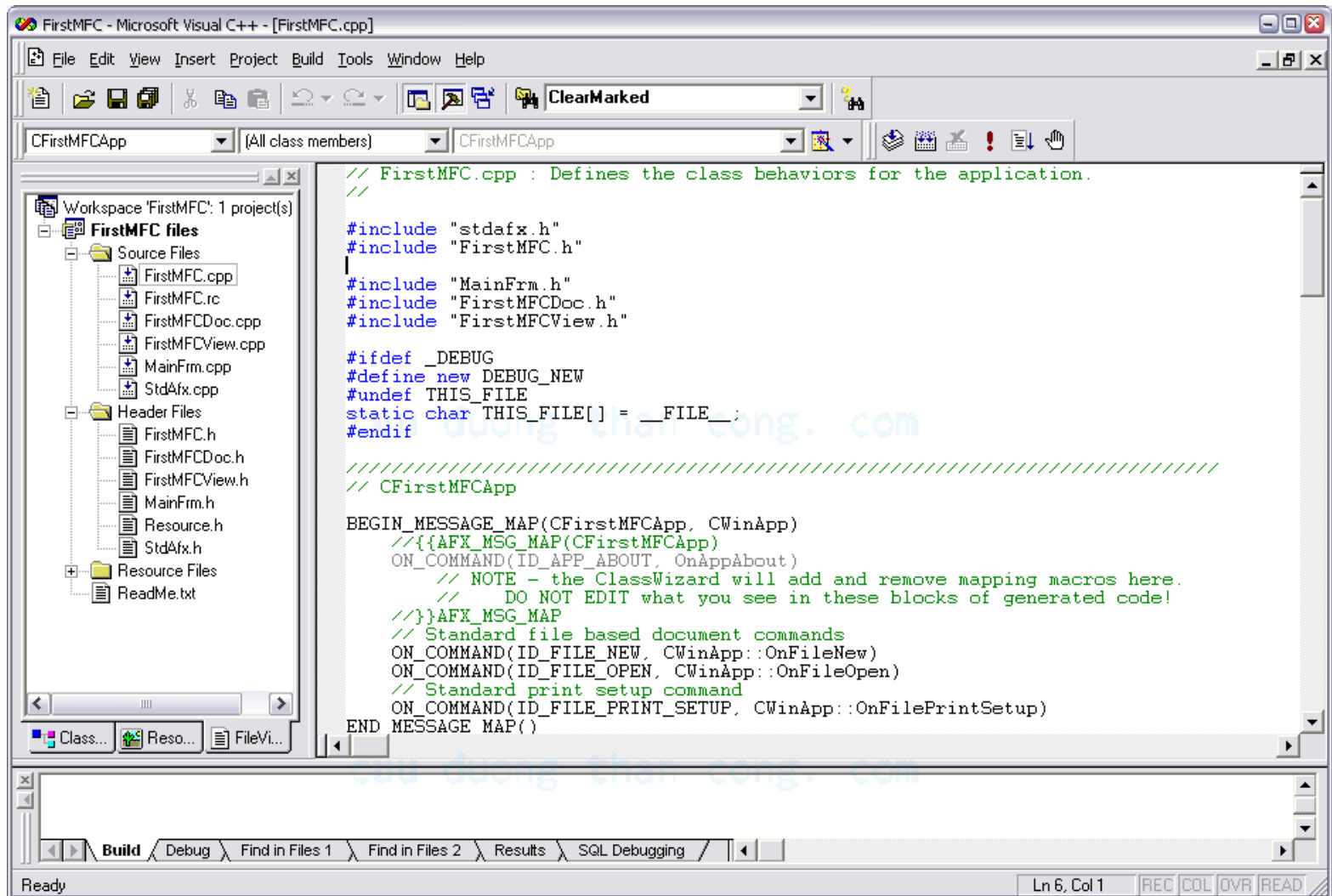
Document: CFirstMFCDoc in FirstMFCDoc.h and FirstMFCDoc.cpp

View: CFirstMFCView in FirstMFCView.h and FirstMFCView.cpp

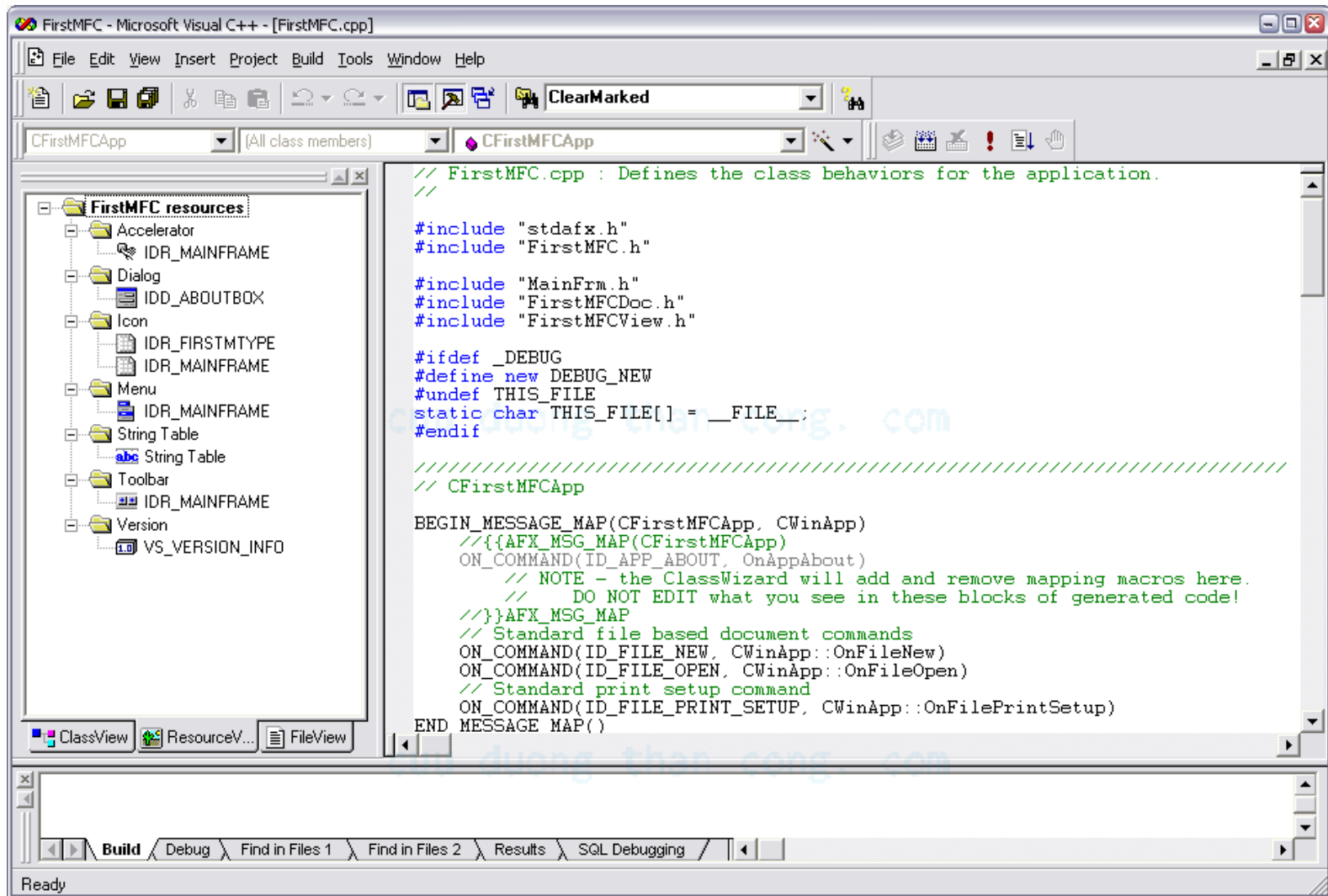
Features:

- + Initial toolbar in main frame
- + Initial status bar in main frame
- + Printing and Print Preview support in view
- + 3D Controls
- + Uses shared DLL implementation (MFC42.DLL)
- + ActiveX Controls support enabled
- + Localizable text in:
 - English [United States]

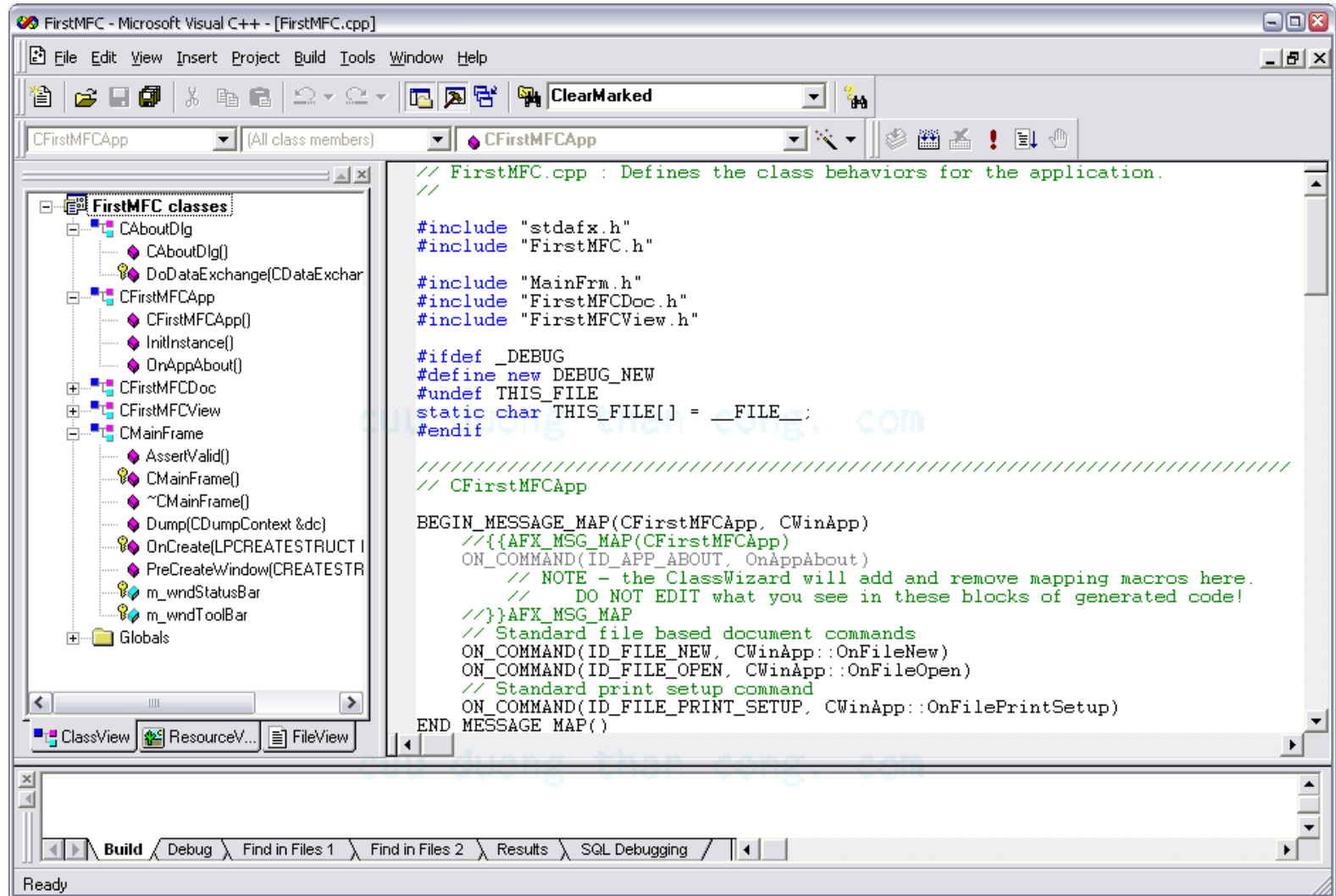
Ứng dụng MFC đầu tiên – File view



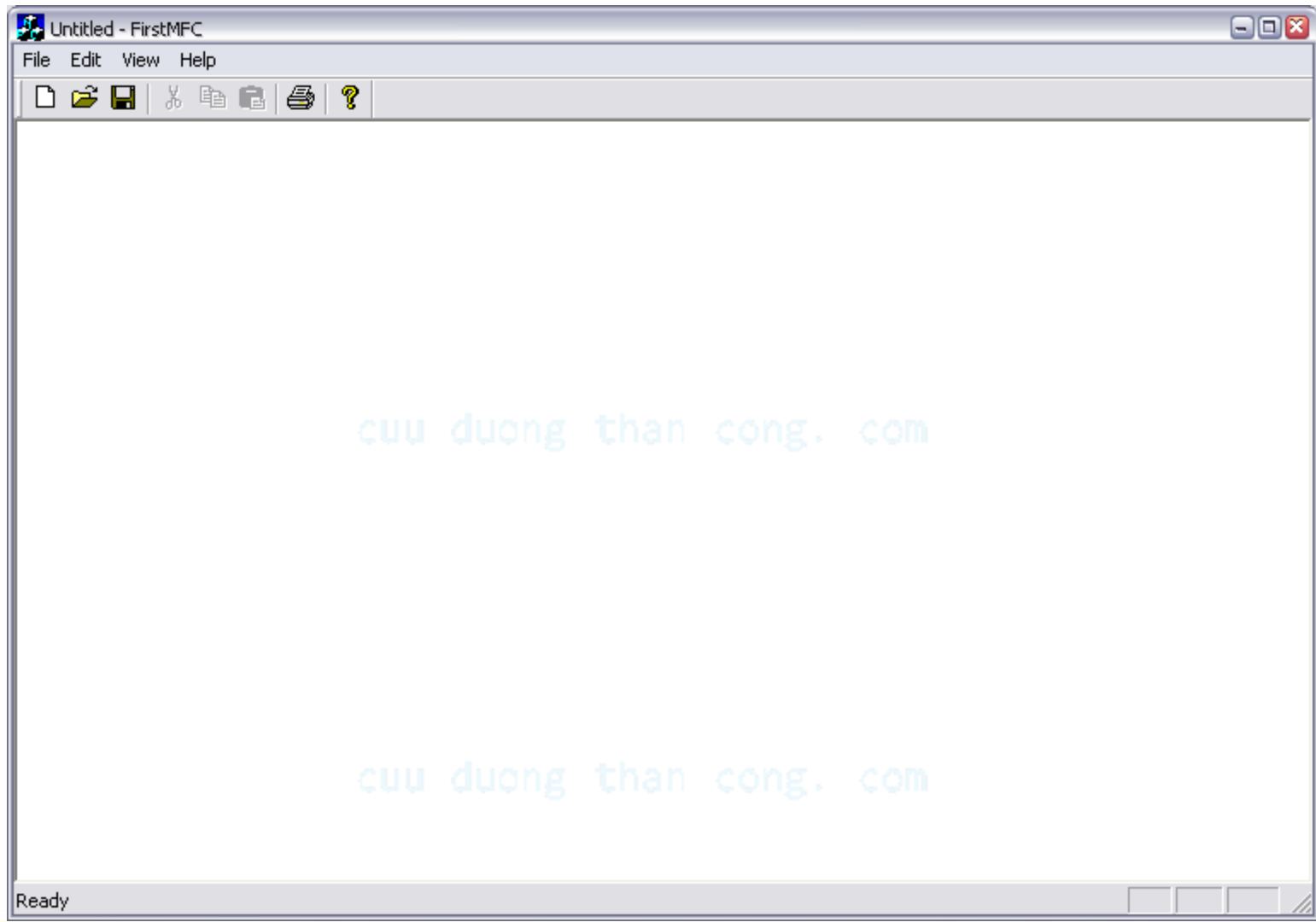
Ứng dụng MFC đầu tiên – Resource view



Ứng dụng MFC đầu tiên – Class view



Ứng dụng MFC đầu tiên – kết quả chương trình



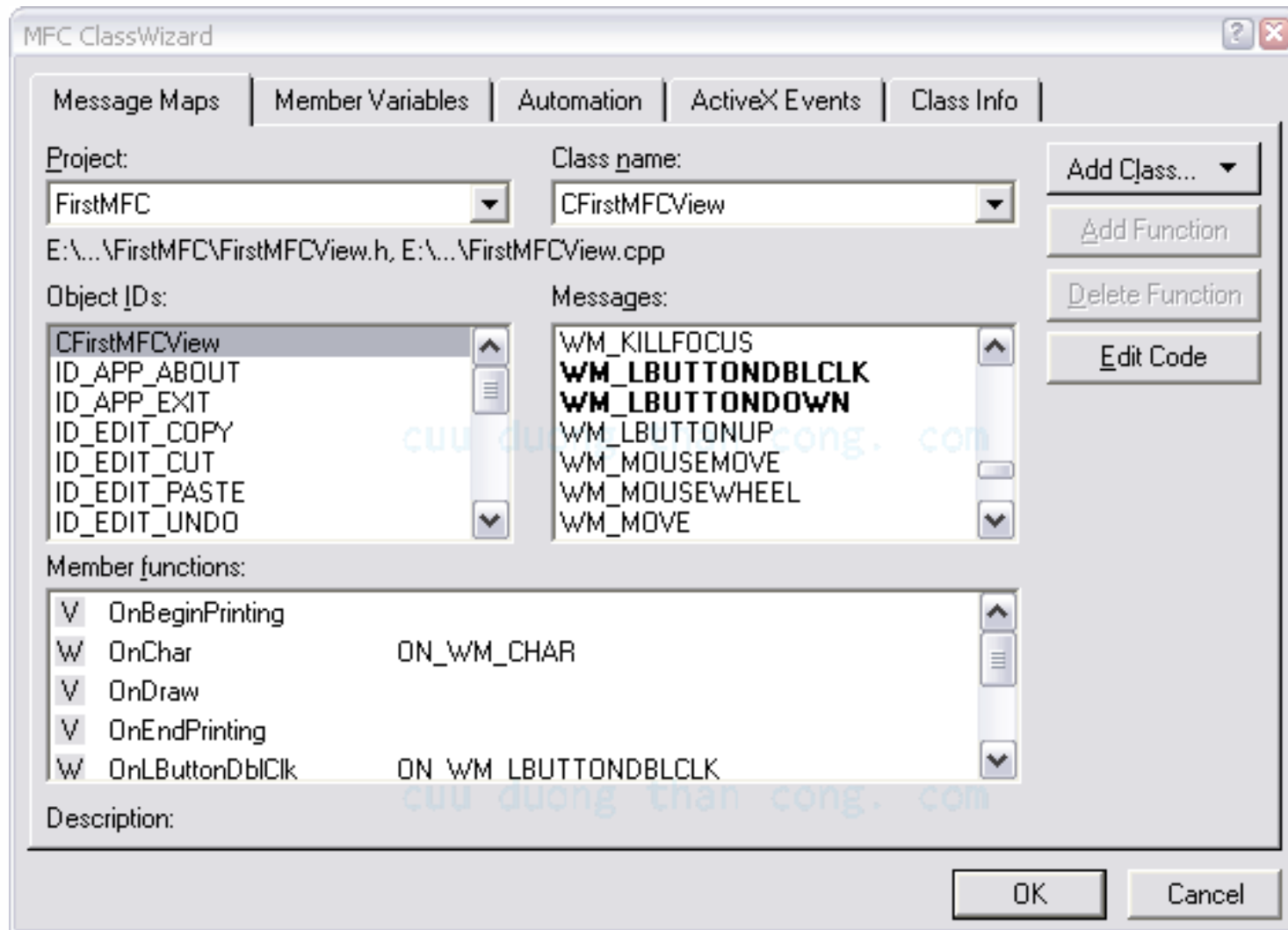
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

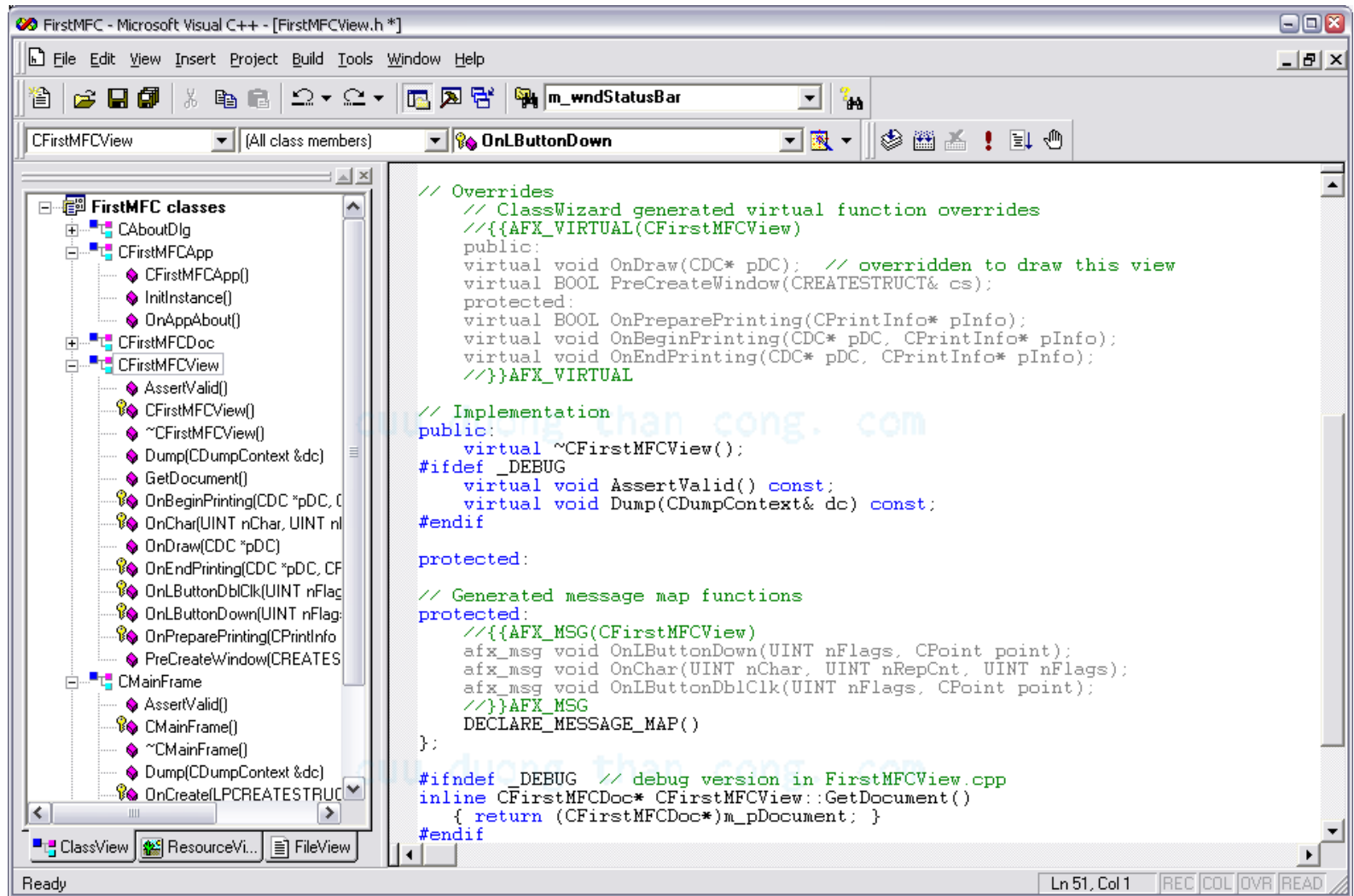
Xử lý thông điệp chuột

Message	Afx function
WM_LBUTTONDOWN	<code>afx_msg void OnLButtonDown(UINT, CPoint)</code>
WM_LBUTTONUP	<code>afx_msg void OnLButtonUp(UINT, CPoint)</code>
WM_LBUTTONBLCLK	<code>afx_msg void OnLButtonDblClk(UINT, CPoint)</code>
WM_RBUTTONDOWN	<code>afx_msg void OnRButtonDown(UINT, CPoint)</code>
WM_RBUTTONUP	<code>afx_msg void OnRButtonUp(UINT, CPoint)</code>
WM_RBUTTONBLCLK	<code>afx_msg void OnRButtonDblClk(UINT, CPoint)</code>
WM_MOUSEMOVE	<code>afx_msg void OnMouseMove(UINT, CPoint)</code>

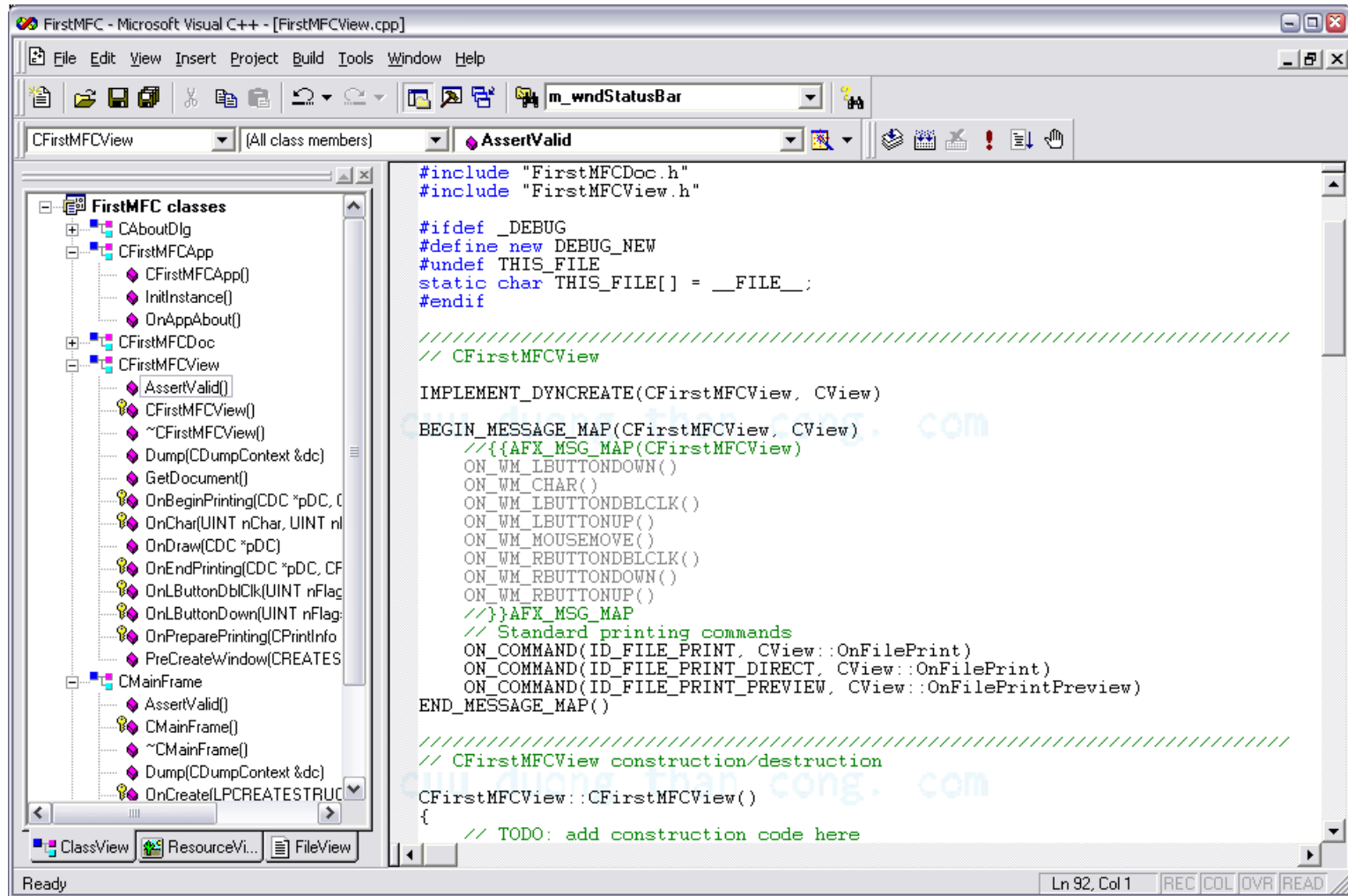
Message Map và ClassWizard



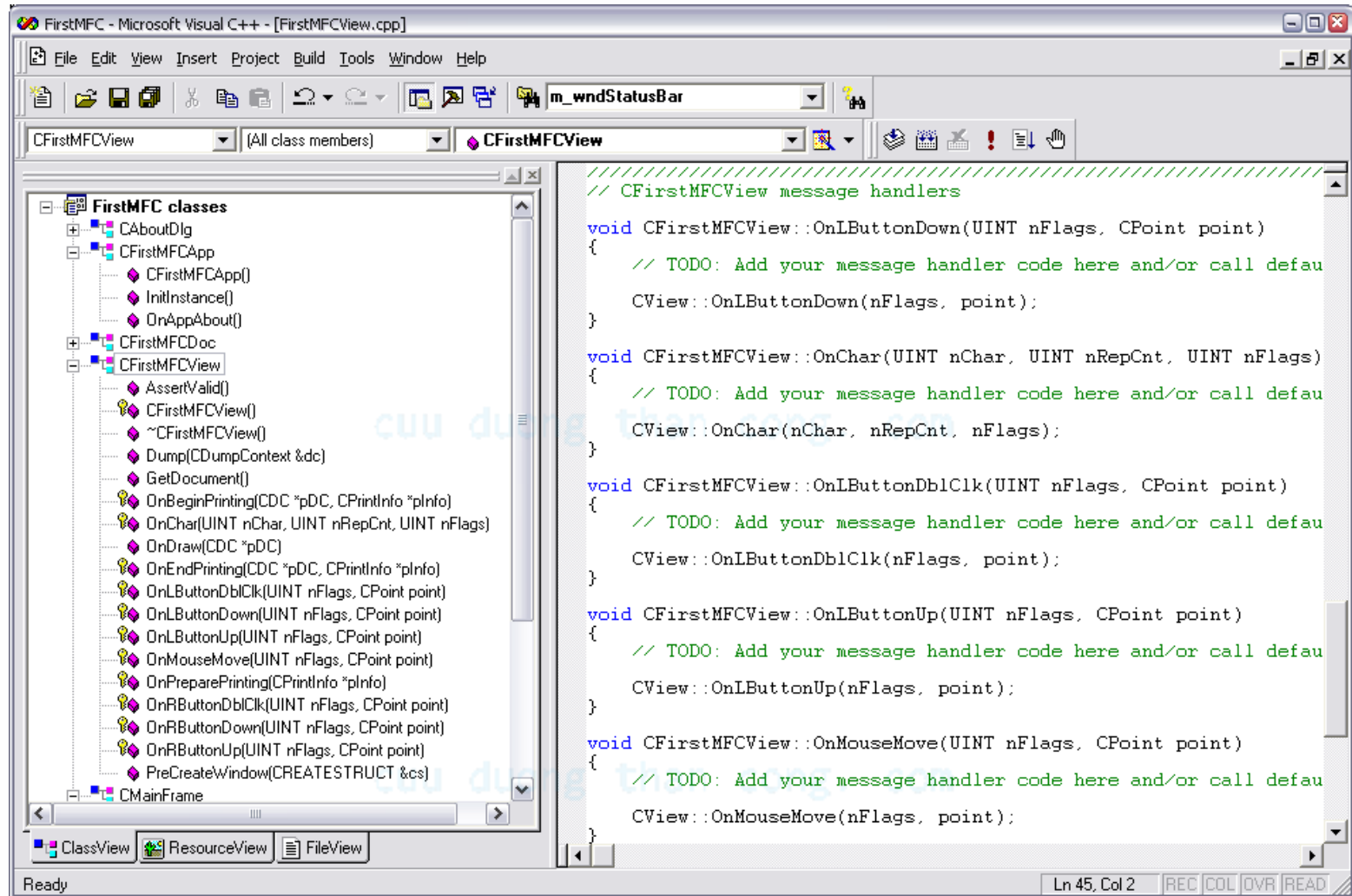
Hàm xử lý thông điệp chuột tương ứng



Hàm xử lý thông điệp chuột tương ứng



Hàm xử lý thông điệp chuột tương ứng



Hàm xử lý thông điệp chuột tương ứng

```
void CMapView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMapView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CView::OnLButtonUp(nFlags, point);
}
```

```
void CMapView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

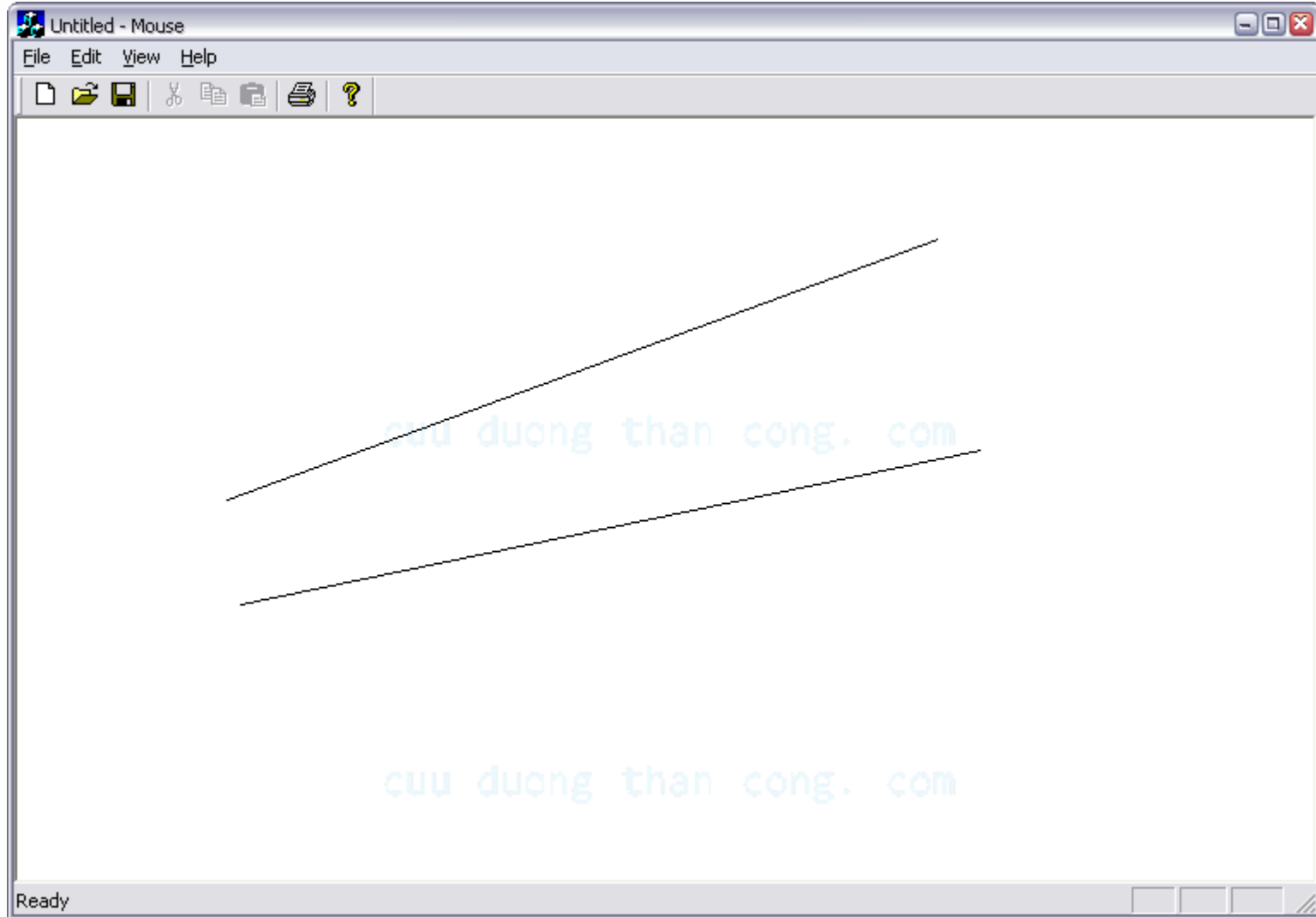
    CView::OnMouseMove(nFlags, point);
}
```

Hàm xử lý thông điệp chuột tương ứng

```
void CMouseView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nStartX = point.x;
    m_nStartY = point.y;
    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMouseView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    dc.MoveTo(m_nStartX, m_nStartY);
    dc.LineTo(point.x, point.y);
    CView::OnLButtonUp(nFlags, point);
}
```

Kết quả xử lý thông điệp chuột

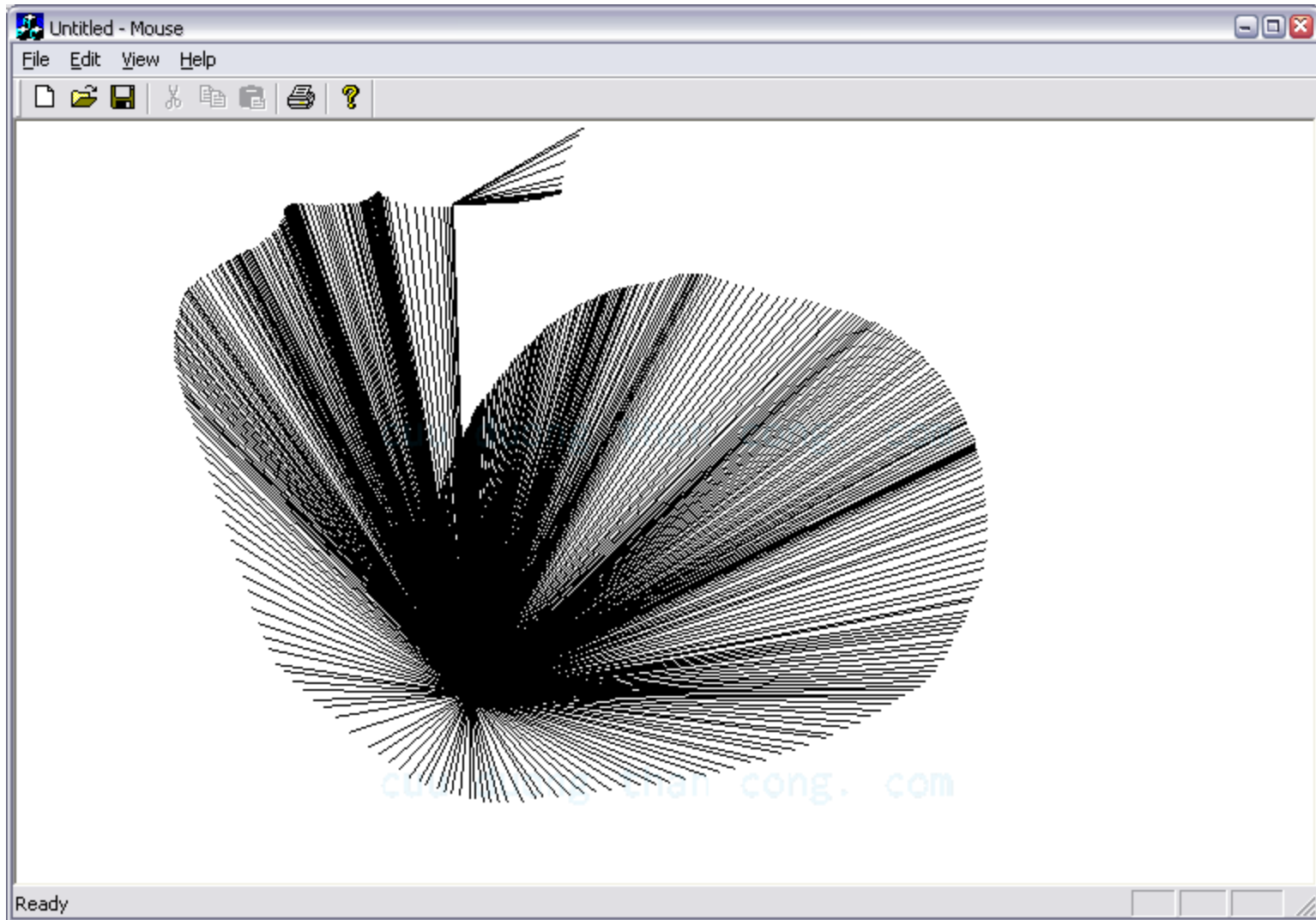


Hàm xử lý thông điệp chuột tương ứng

```
void CMouseView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nStartX = point.x;
    m_nStartY = point.y;
    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMouseView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    dc.MoveTo(m_nStartX, m_nStartY);
    dc.LineTo(point.x, point.y);
    CView::OnLButtonUp(nFlags, point);
}
```

Kết quả xử lý thông điệp chuột

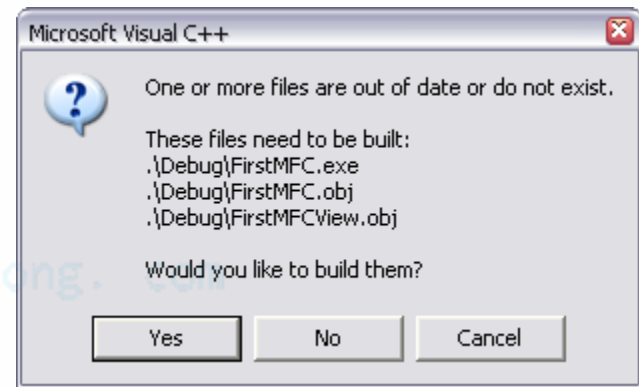
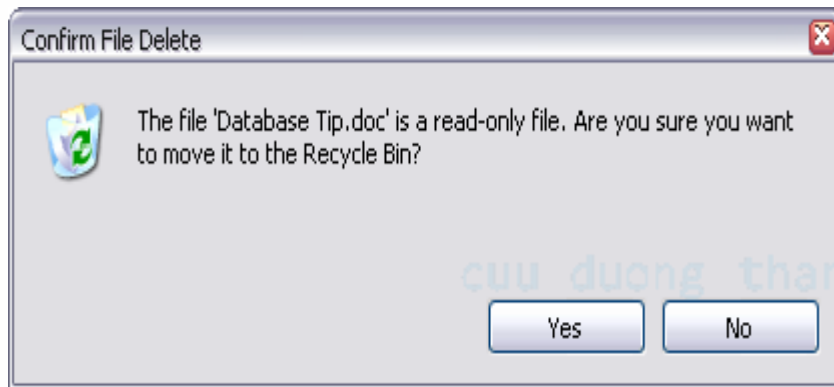


NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ **Message Box**

Message Box - Hộp thông báo

- ❖ **Hộp thông báo được sử dụng để thông báo, trao đổi giữa ứng dụng với người dùng**



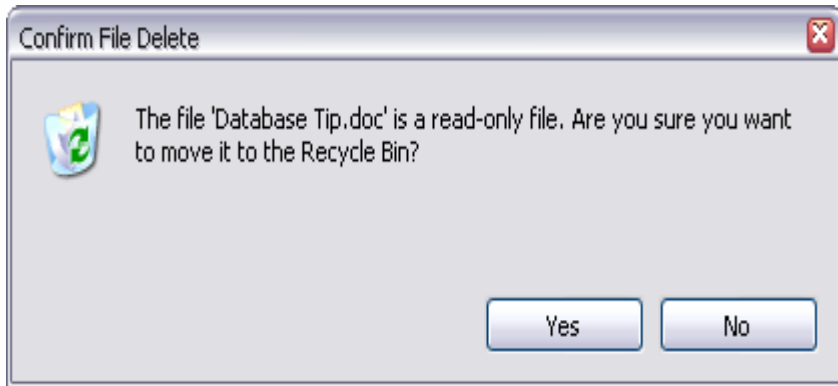
- ❖ **Để tạo lập hộp thông báo, ta sử dụng hàm MessageBox**

```
Cwnd::int MessageBox (LPCTSTR lpszText,  
                      LPCTSTR lpszCaption = NULL,  
                      UINT nType = MB_OK)
```


Các kiểu hộp thông báo và giá trị trả về tương ứng

Type	Buttons	Possible Return Codes
MB_ABORTRETRYIGNORE	Abort, Retry, Ignore	IDABORT, IDRETRY, IDIGNORE
MB_OK	OK	IDOK
MB_OKCANCEL	OK, Cancel	IDOK, IDCANCEL
MB_RETRYCANCEL	Retry, Cancel	IDRETRY, IDCANCEL
MB_YESNO	Yes, No	IDYES, IDNO
MB_YESNOCANCEL	Yes, No, Cancel	IDYES, IDNO, IDCANCEL

Message Box - Hộp thông báo

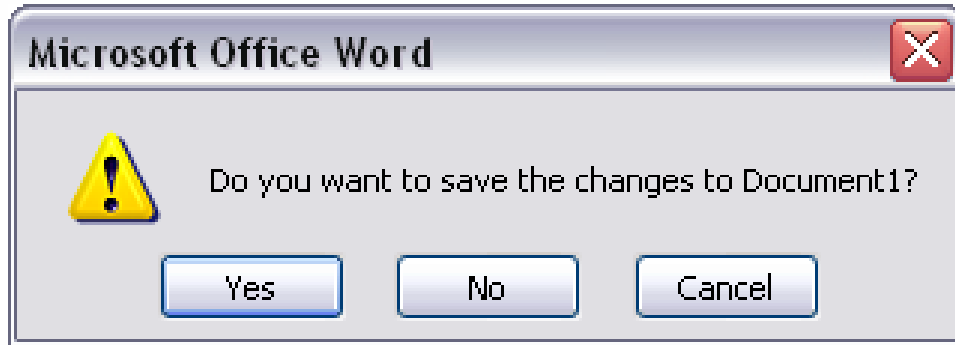


- ❖ Để tạo lập hộp thông báo trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("The file 'Database Tip.doc' is a read-only  
file. Are you sure you want to move it to the Recycle Bin ?",  
"Confirm File Delete", MB_YESNO)
```

- ❖ Chọn "Yes" → **nRes = IDYES**
- ❖ Chọn "No" → **nRes = IDNO**

Message Box - Hộp thông báo



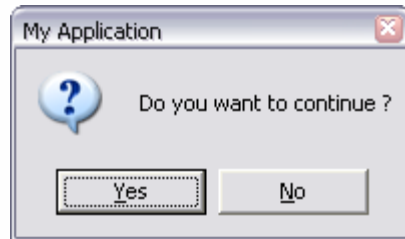
- ❖ Để tạo lập hộp thông báo trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("Do you want to save the changes to  
Document1 ?", "Microsoft Office Word", MB_YESNOCANCEL)
```

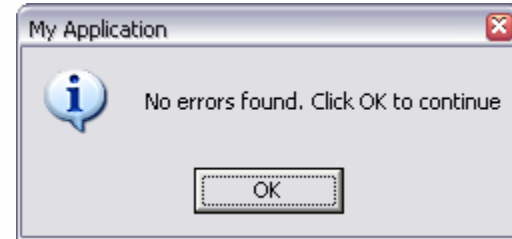
- ❖ Chọn "Yes" → **nRes = IDYES**
- ❖ Chọn "No" → **nRes = IDNO**
- ❖ Chọn "Cancel" → **nRes = IDCANCEL**

Các kiểu biểu tượng trong hộp thông báo

MB_ICONQUESTION

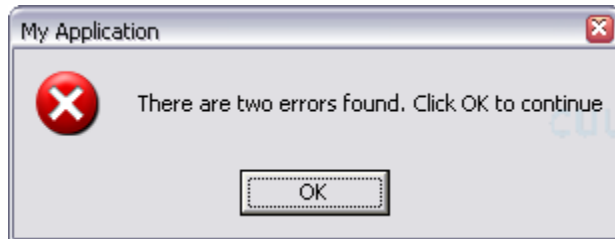


MB_ICONINFORMATION



cuu duong than cong. com

MB_ICONSTOP



MB_ICONEXCLAMATION



cuu duong than cong. com

Message Box - Hộp thông báo

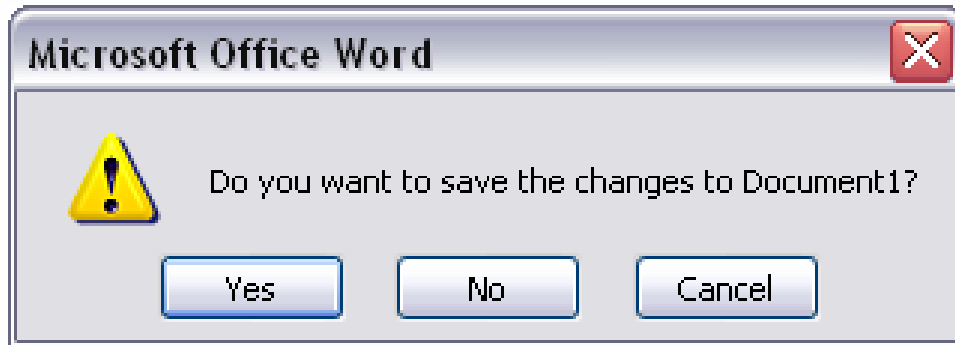


- ❖ Để tạo lập hộp thông báo với Icon như trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("Do you want to continue ?", "My  
Application", MB_YESNO | MB_ICONQUESTION )
```

cuu duong than cong. com

Message Box - Hộp thông báo



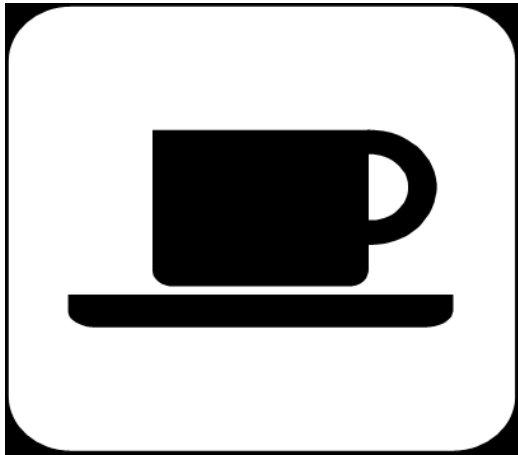
- ❖ Để tạo lập hộp thông báo với Icon như trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("Do you want to save the changes to  
Document1 ?", "Microsoft Office Word", MB_YESNOCANCEL |  
MB_ICONEXCLAMATION )
```

cuu duong than cong. com



cuu duong nhan cong. com



cuu duong nhan cong. com



Dialog Box Common Controls Property Sheet

- ❖ Tạo ứng dụng dạng Dialog Based
- ❖ Tạo và xử lý các control trong dialog
- ❖ Property Sheet
- ❖ Common Dialog

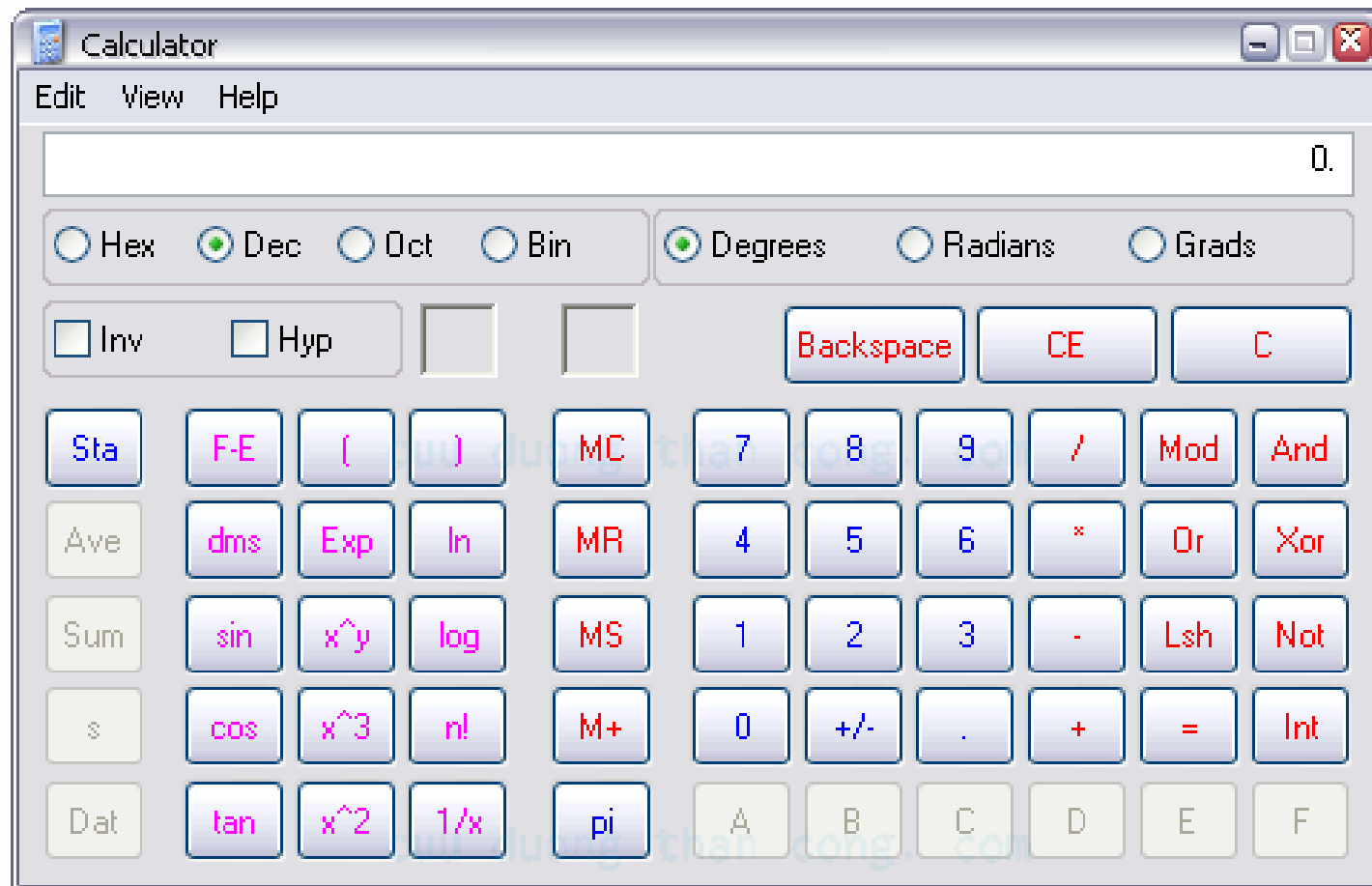
Trần Ngọc Bảo

Email: tnbao.dhsp@gmail.com

NỘI DUNG

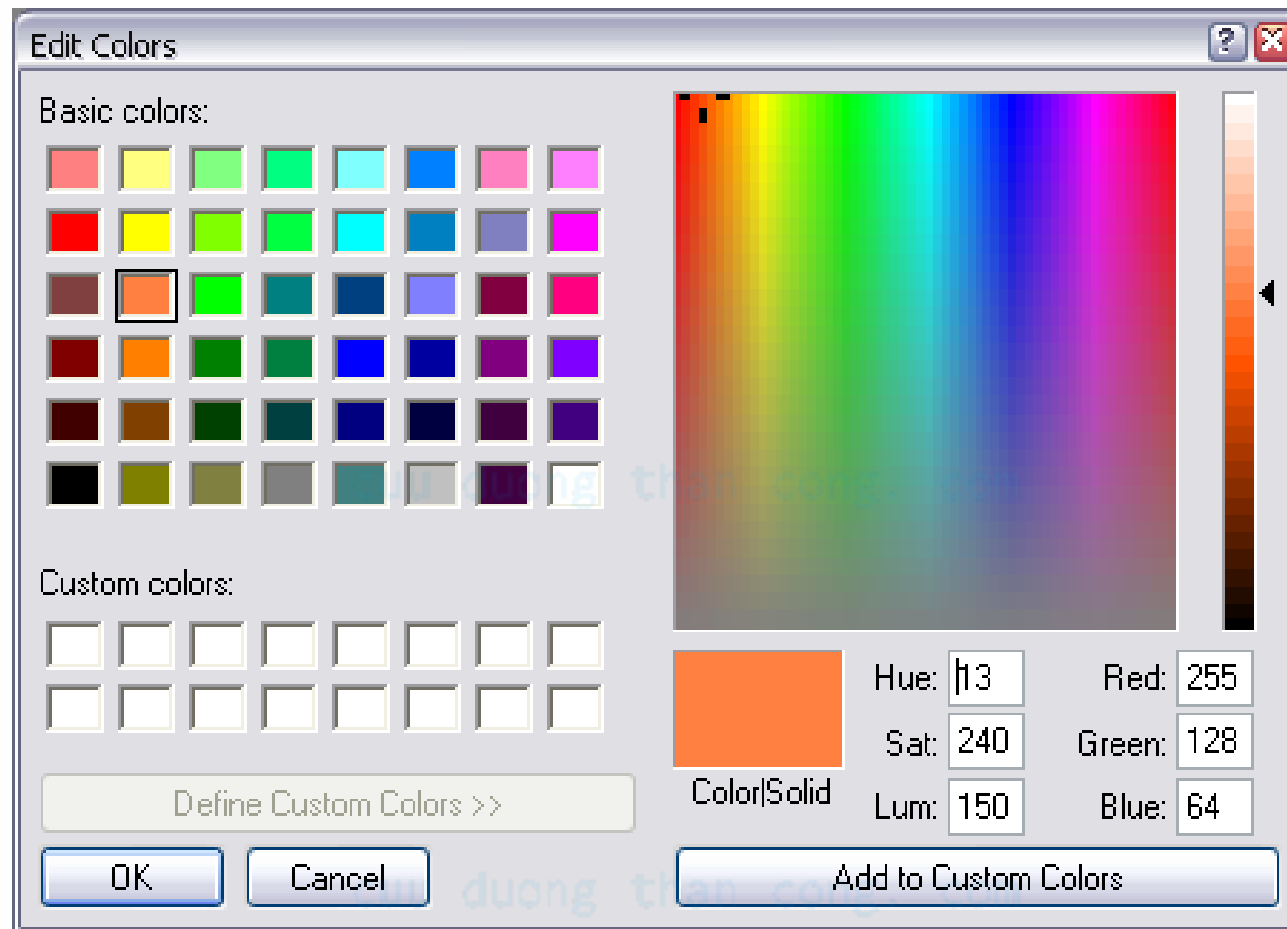
- ❖ Giới thiệu về Dialog box
- ❖ Chương trình minh họa Dialog Based
- ❖ Tạo và xử lý dialog
- ❖ Cài đặt và xử lý các control trong dialog
- ❖ Modeless dialog
- ❖ Property Sheet
- ❖ Common dialog

Giới thiệu về dialog



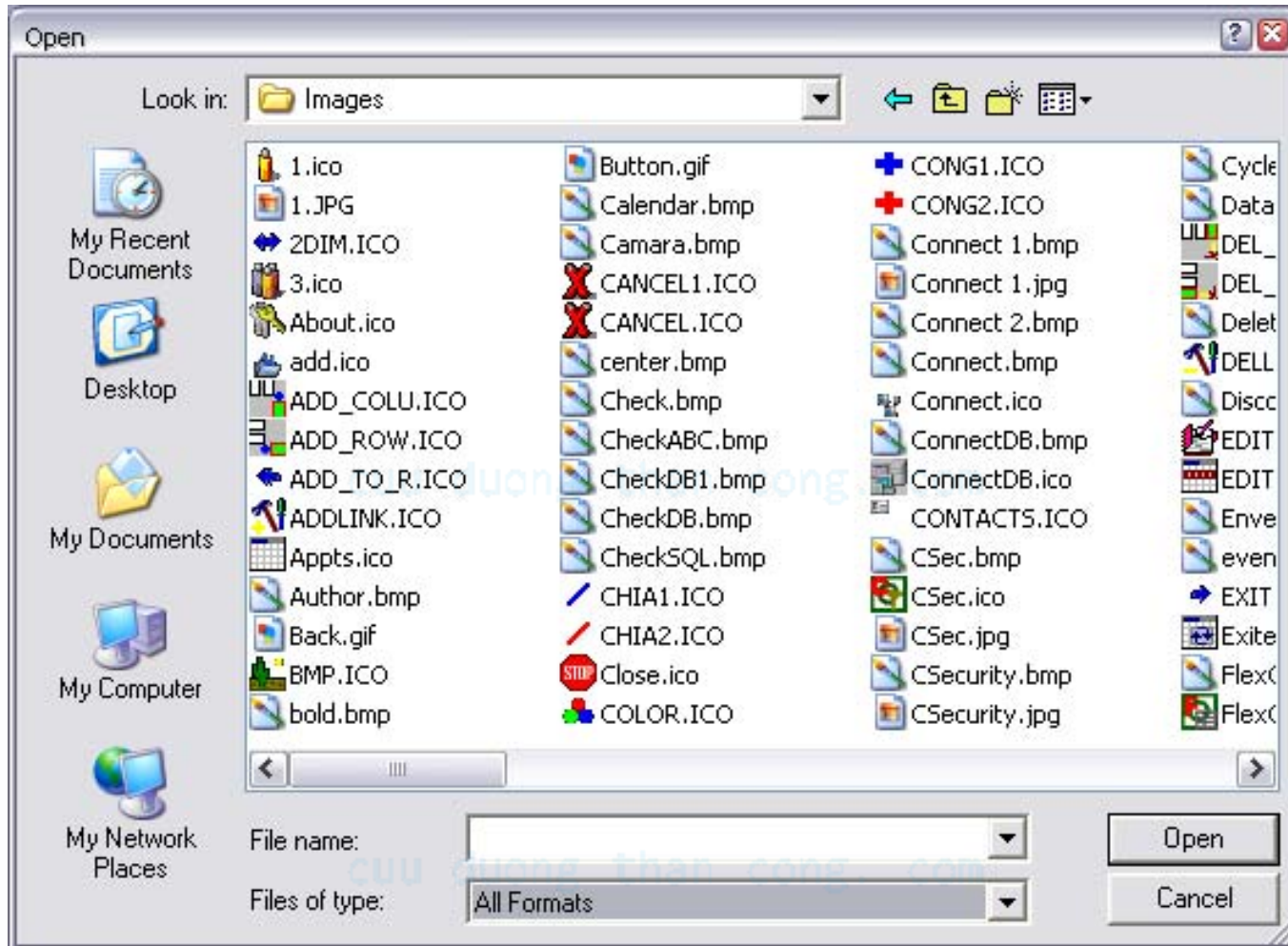
Dialog application

Giới thiệu về dialog – Common dialog



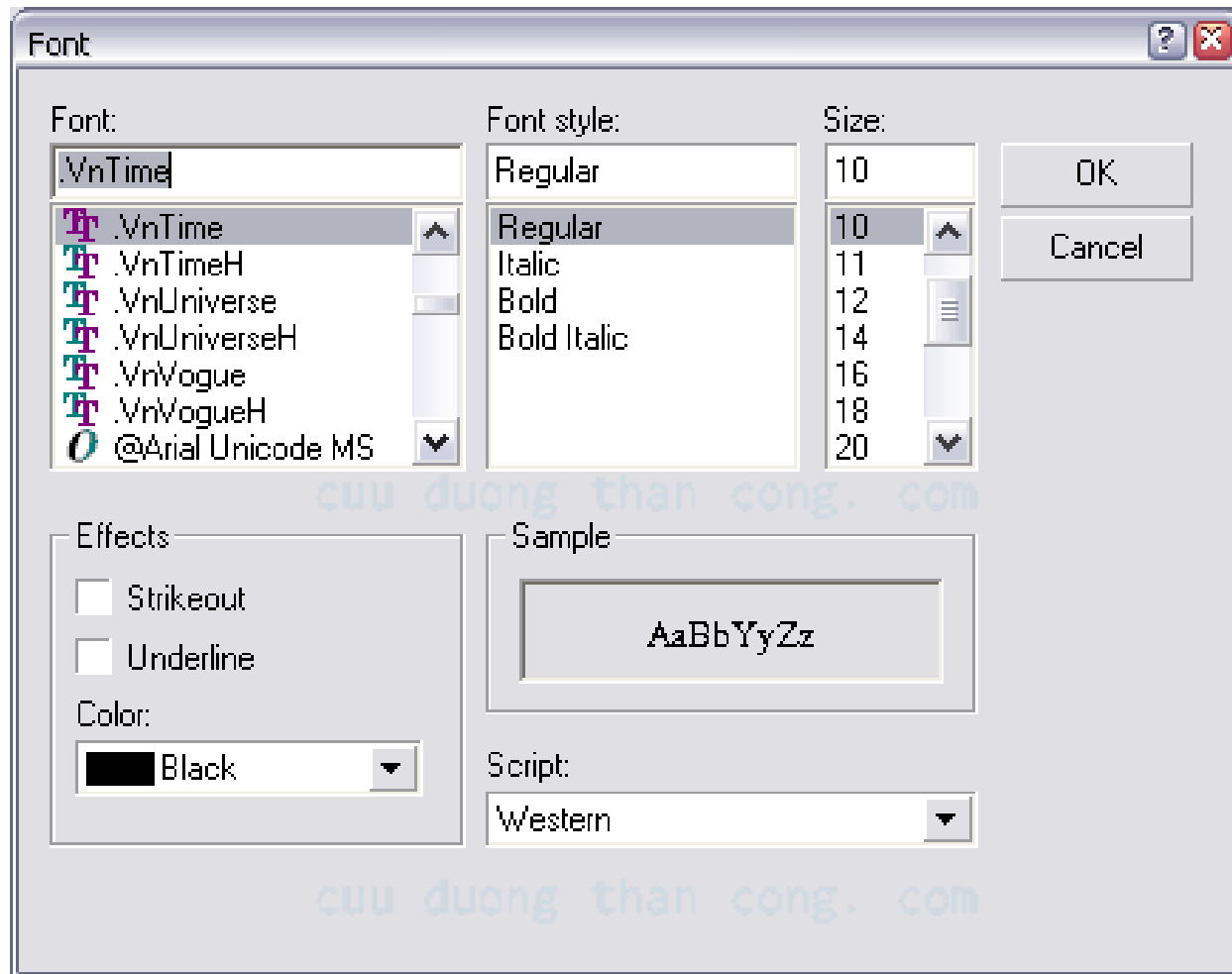
Color dialog

Giới thiệu về dialog – Common dialog



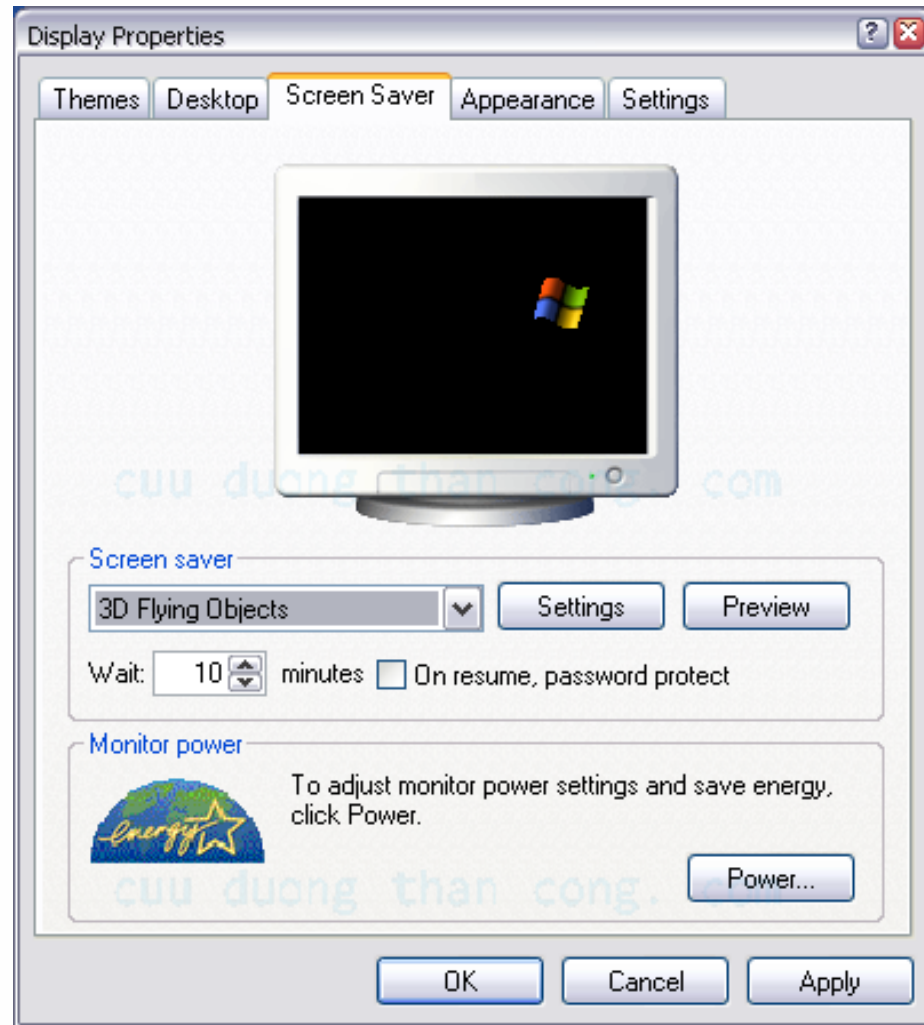
Open file dialog

Giới thiệu về dialog – Common dialog

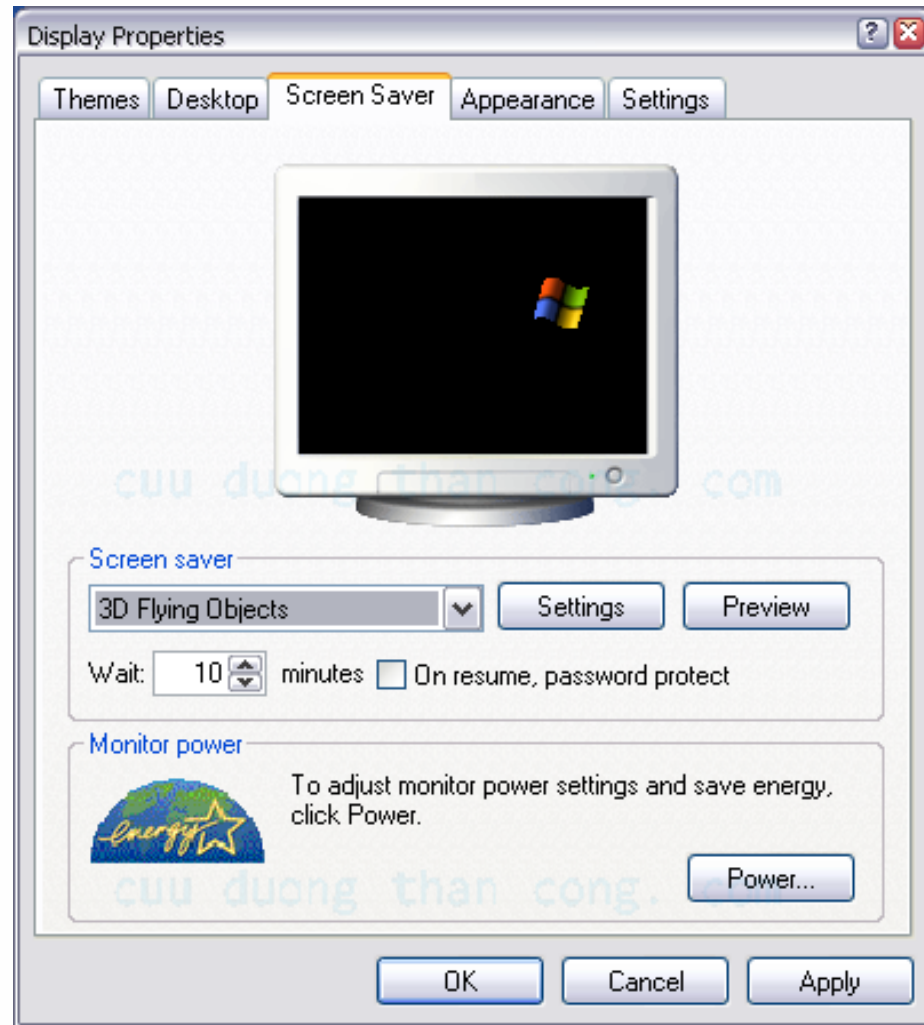


Fonts dialog

Giới thiệu về dialog – Property Sheet



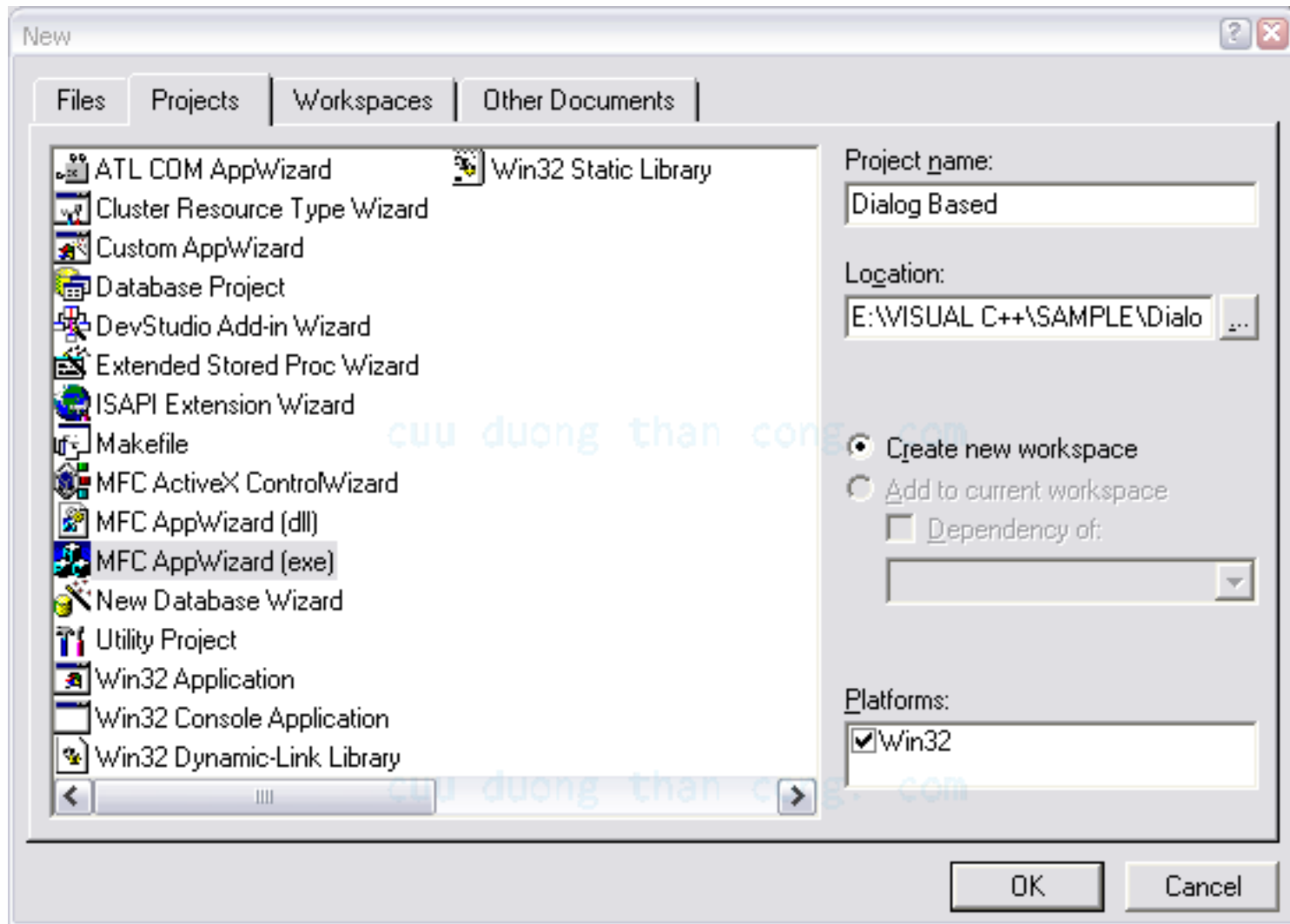
Giới thiệu về dialog – Property Sheet



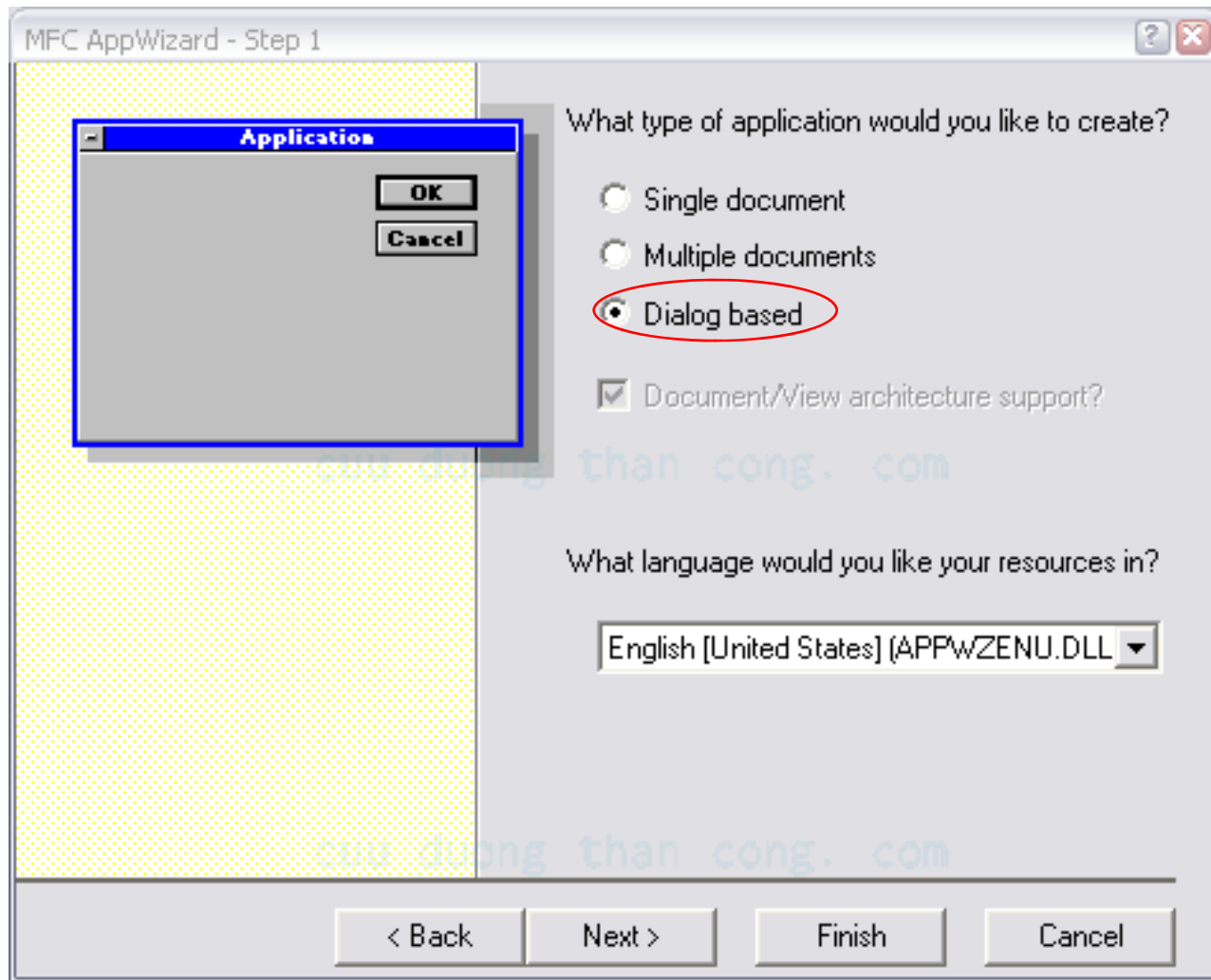
Tạo ứng dụng Dialog Based bằng AppWizard

- ❖ Chọn menu **File → New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project **"MFC AppWizard (exe)"**
- ❖ Đặt tên project trong ô **"Project name"**
- ❖ Xác định đường dẫn thư mục trong **"Location"**
- ❖ Step 1: Chọn loại ứng dụng **"Dialog based"**
- ❖ Step 2: Chỉ chọn Option **"3D controls"**
- ❖ Nhấn **"Finish"** để kết thúc

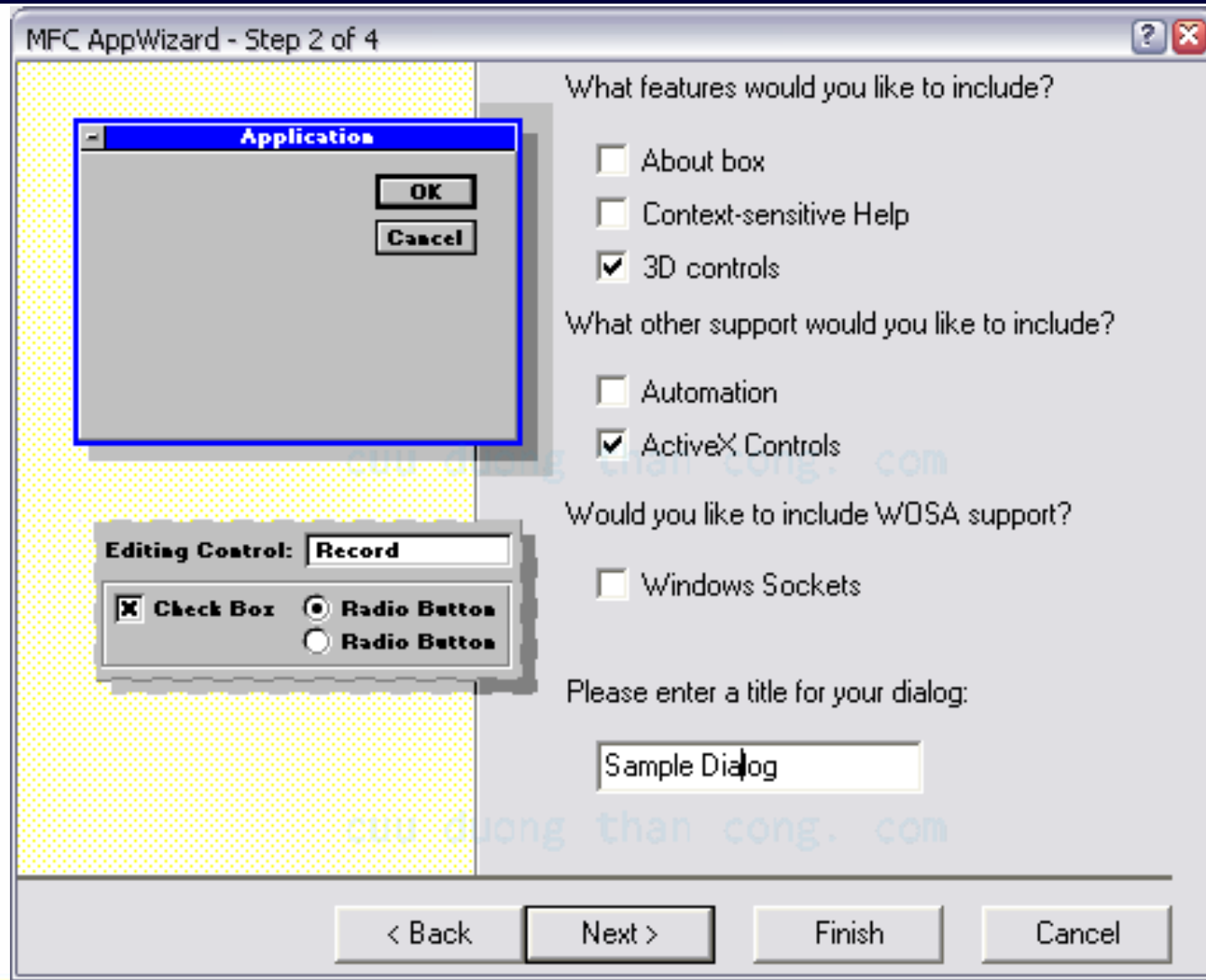
Dialog based - Tạo ứng dụng bằng AppWizard



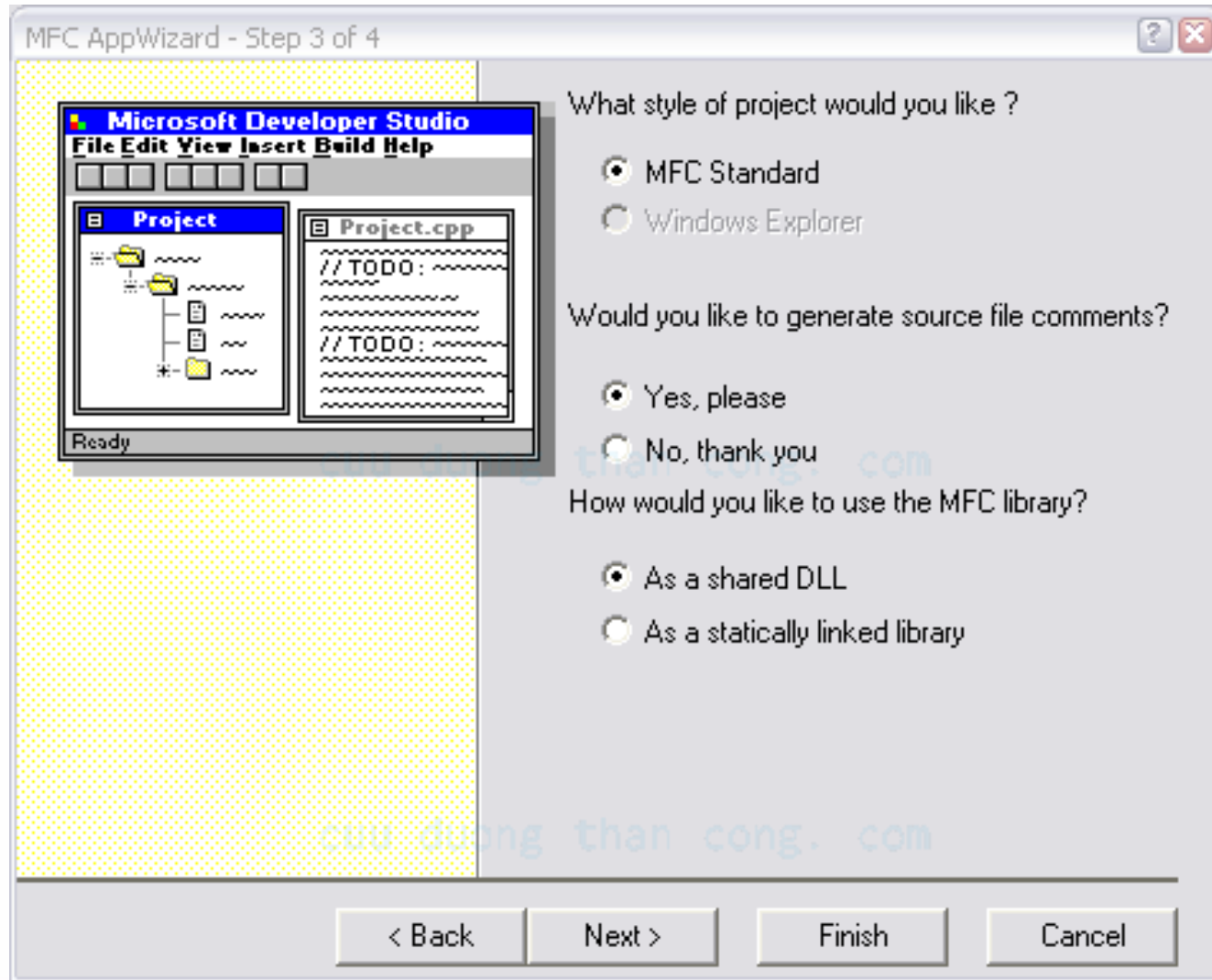
Dialog based - Tạo ứng dụng bằng AppWizard



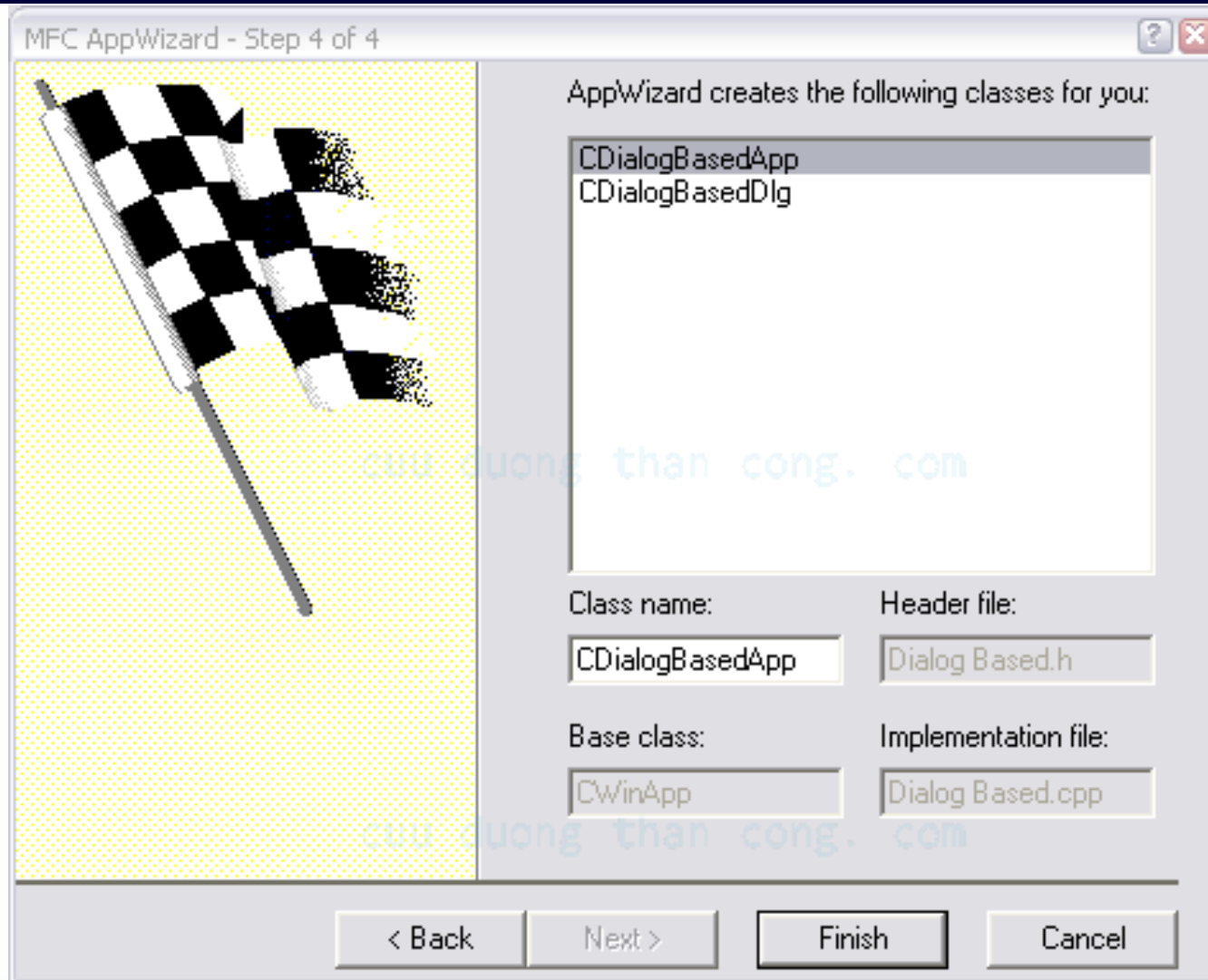
Dialog based - Tạo ứng dụng bằng AppWizard



Dialog based - Tạo ứng dụng bằng AppWizard



Dialog based - Tạo ứng dụng bằng AppWizard

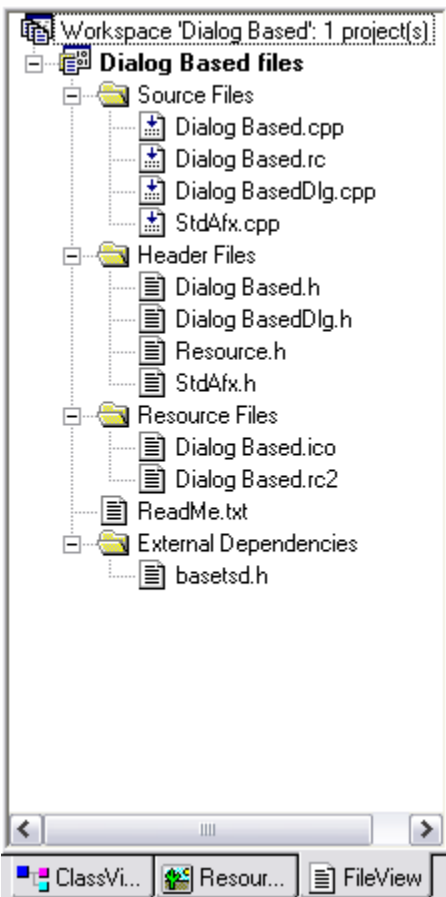


Dialog based - Tạo ứng dụng bằng AppWizard



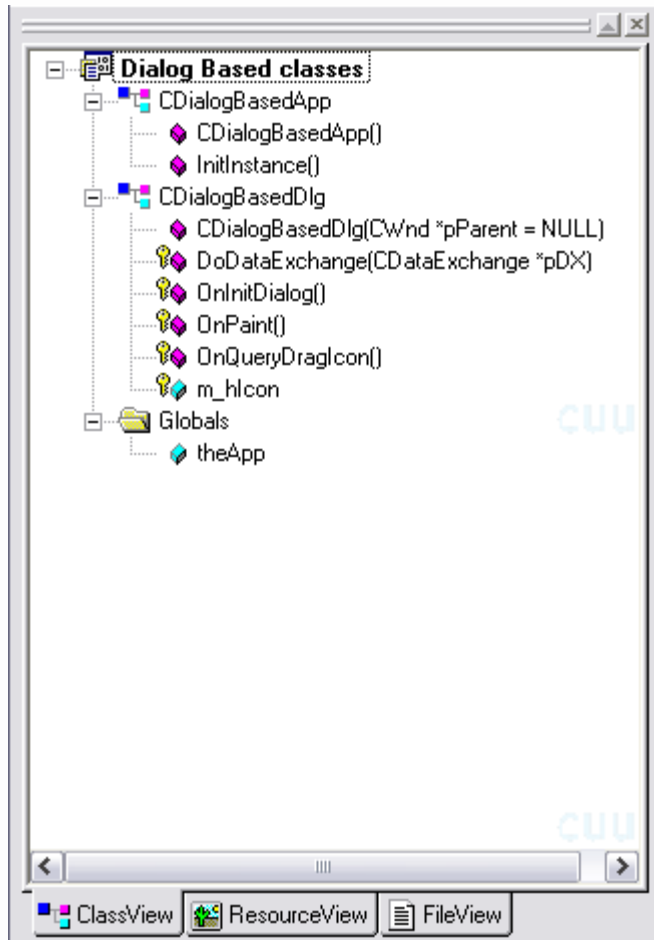
Dialog based - Các thành phần của ứng dụng (1/3)

❖ Các file chương trình: (abc là tên của ứng dụng)



- **abc.h:** header file của file abc.cpp, chứa khai báo CabcApp để quản lý toàn bộ ứng dụng. Lớp CabcApp kế thừa từ lớp CWinApp của MFC
- **abcDlg.h:** header file của file abcDlg.cpp, chứa khai báo CabcDlg để quản lý cửa sổ Dialog giao diện của ứng dụng. Lớp CabcDlg kế thừa từ lớp CDialog của MFC
- **Resource.h:** header file, chứa các hằng ID của các resource được định nghĩa trong file abc.rc
- **abc.cpp:** cài đặt các hàm thành phần của lớp CabcApp
- **abcDlg.cpp:** cài đặt các hàm thành phần của lớp CabcDlg
- **abc.rc:** mô tả các resource của ứng dụng

Dialog based - Các thành phần của ứng dụng (2/3)



```
class CDialogBasedApp : public
    CWinApp
{
public:
    CDialogBasedApp();

public:
    virtual BOOL InitInstance();
    DECLARE_MESSAGE_MAP()
};
```


Dialog based - Các thành phần của ứng dụng (3/3)

```
class CDialogBasedDlg : public CDialog
{
public:
    CDialogBasedDlg(CWnd* pParent = NULL);    // standard
    constructor

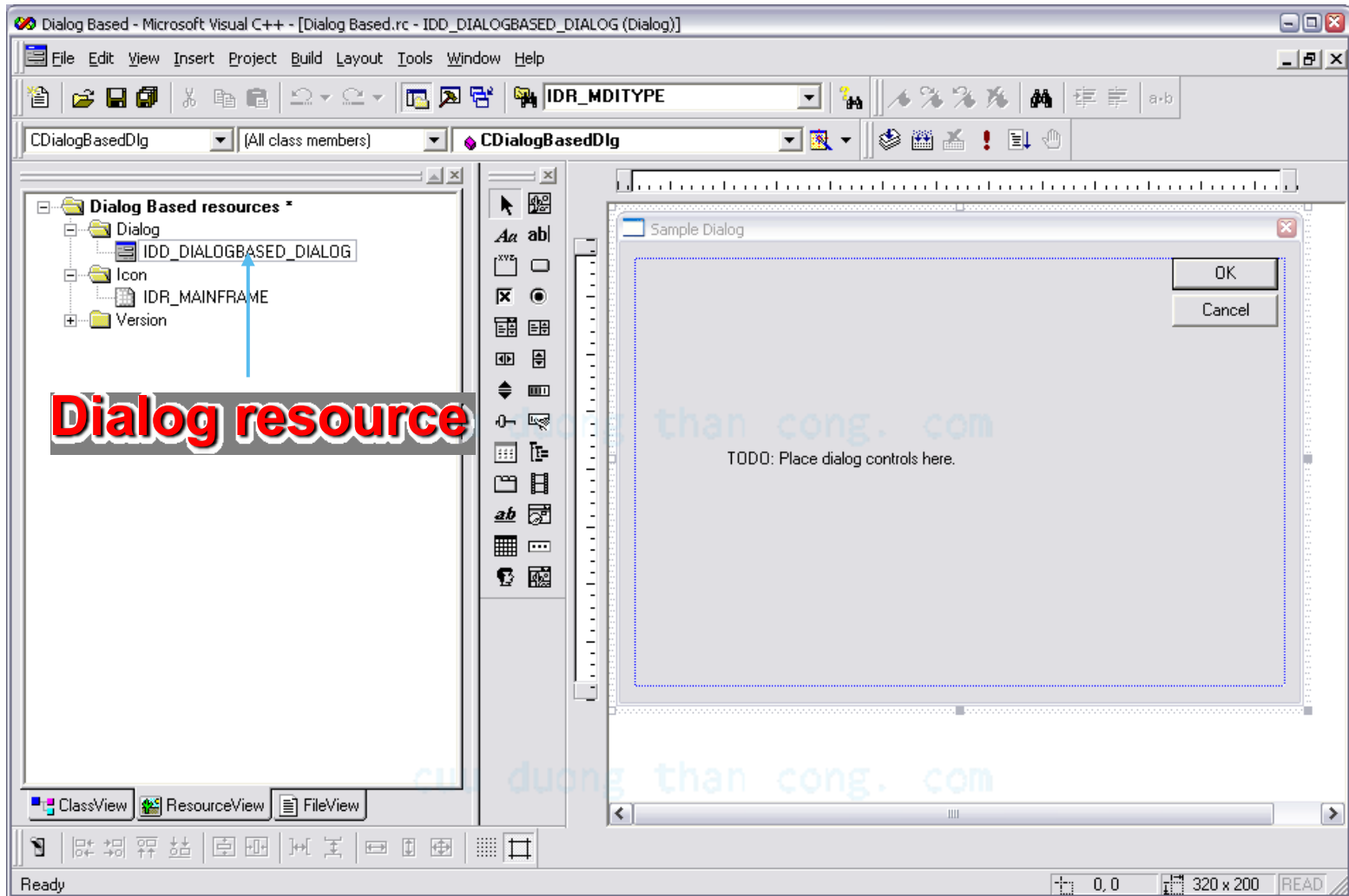
    enum { IDD = IDD_DIALOGBASED_DIALOG };
protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
    support
protected:
    HICON m_hIcon;

    virtual BOOL OnInitDialog();
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
};
```

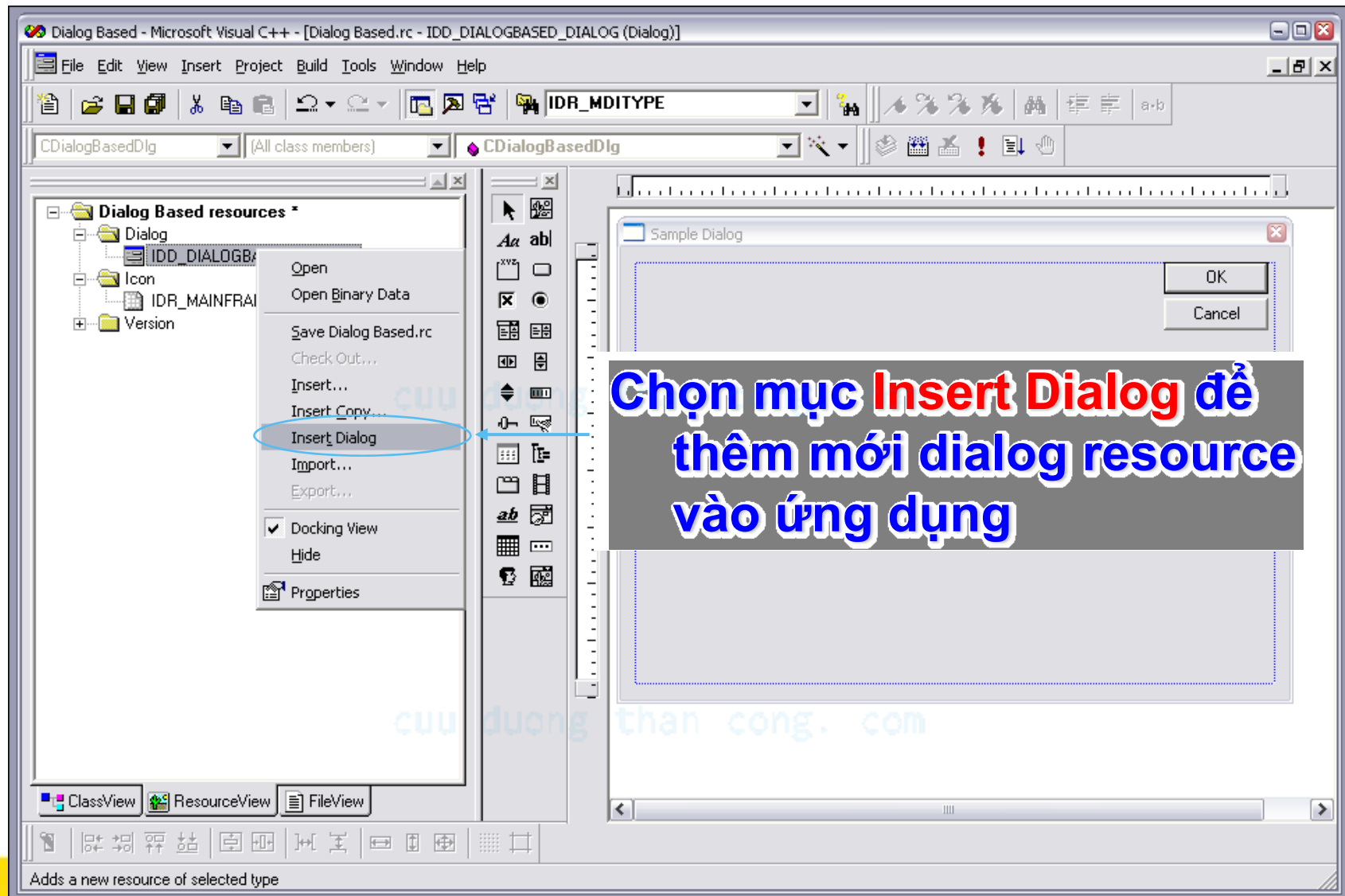
Tạo và xử lý dialog

- ❖ Thêm mới Dialog resource
- ❖ Thiết lập thuộc tính dialog
- ❖ Khai báo lớp CDialog
- ❖ Thêm các control vào dialog
- ❖ Hiển thị dialog
- ❖ Xử lý dialog

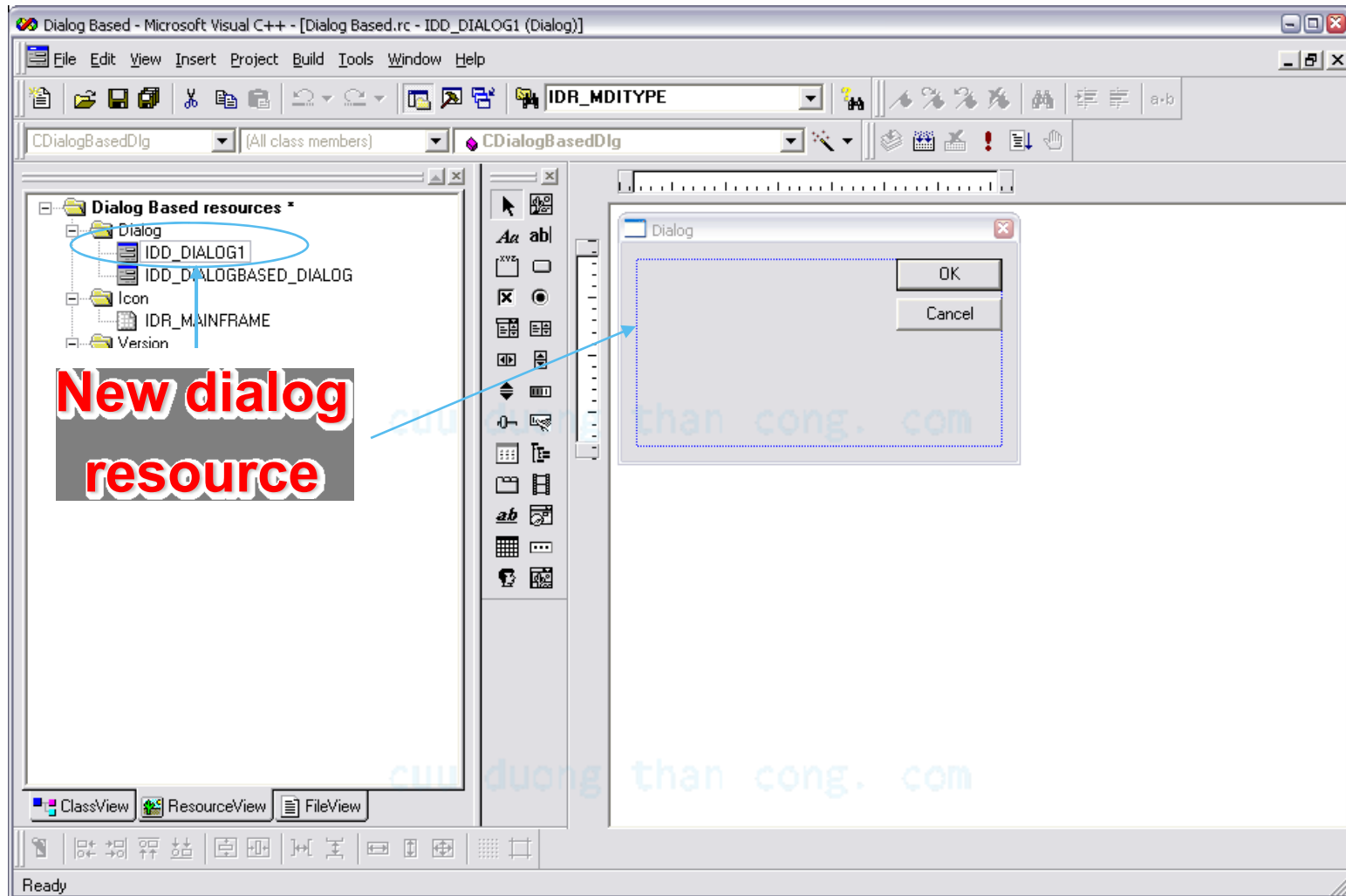
Thêm mới dialog resource



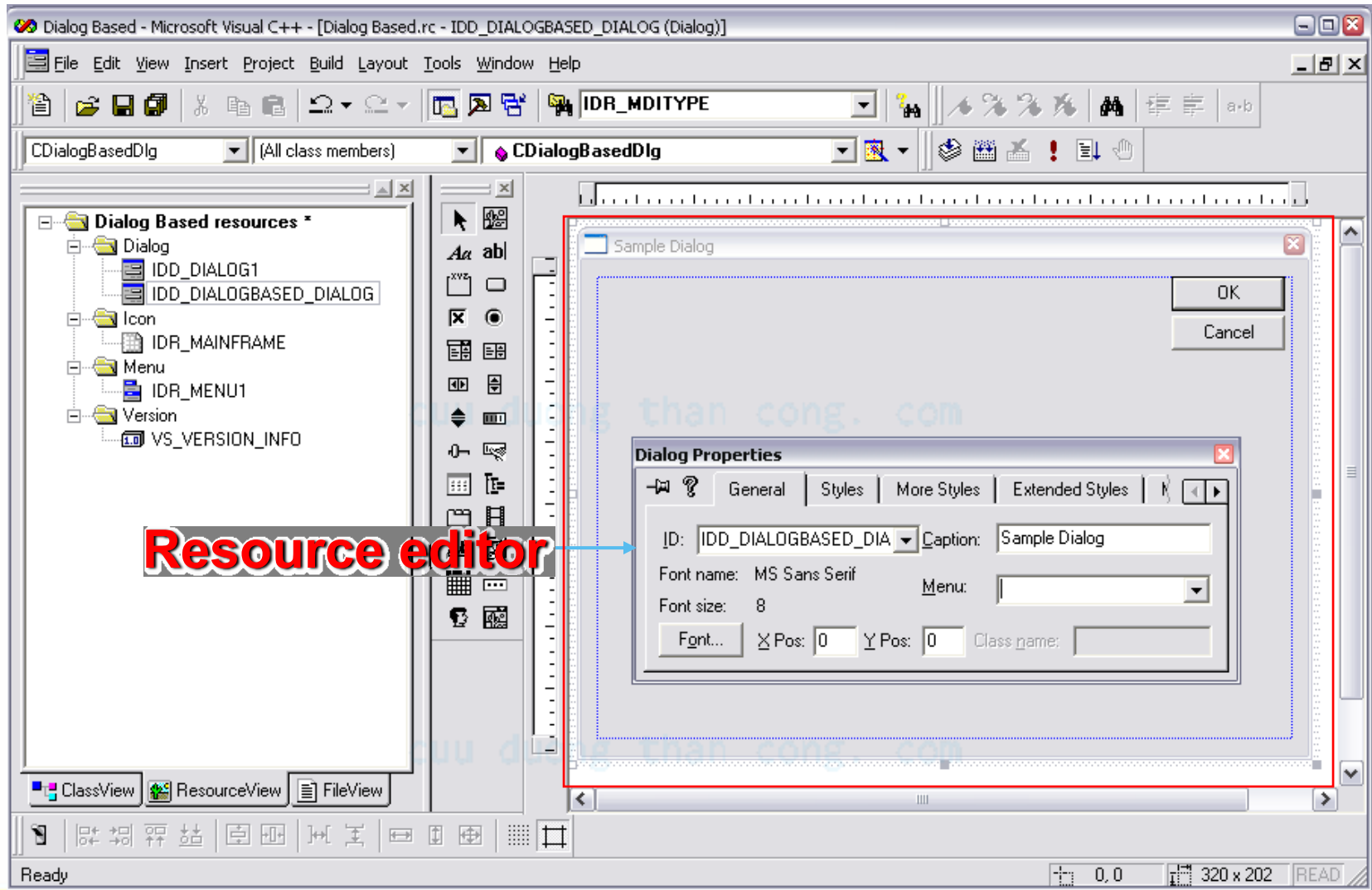
Thêm mới dialog resource (tt)



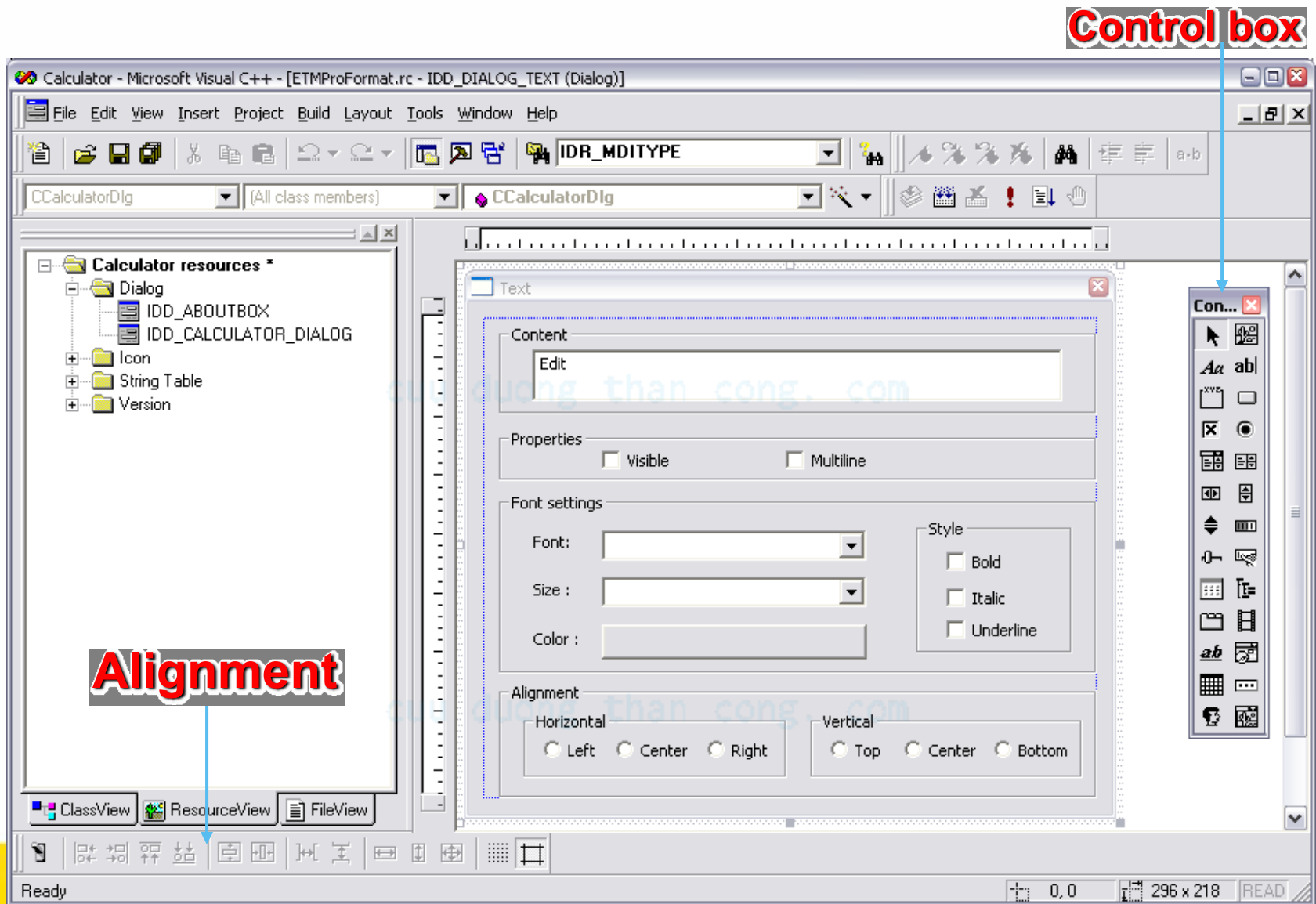
Thêm mới dialog resource (tt)



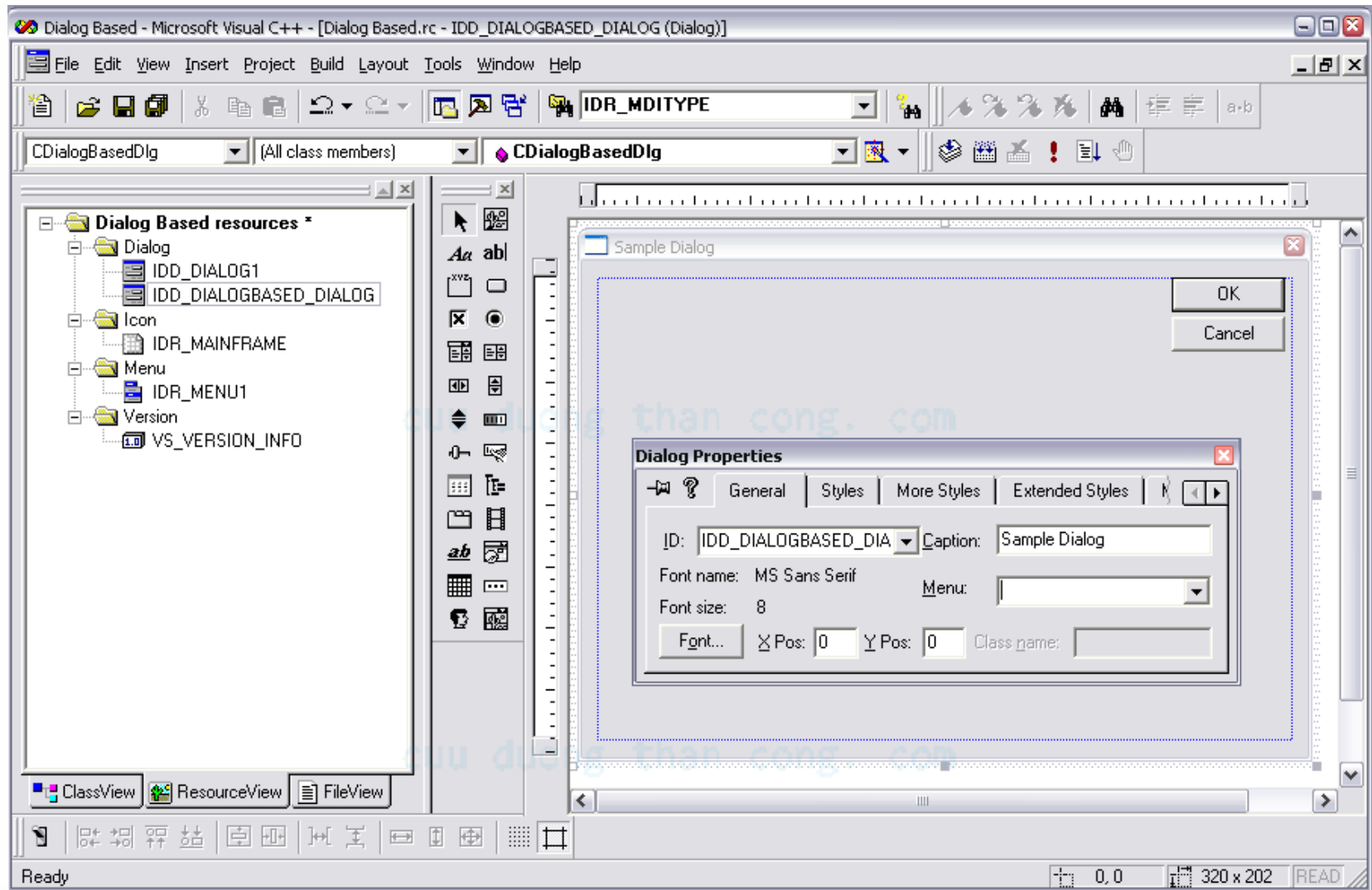
Thêm mới dialog resource (tt)



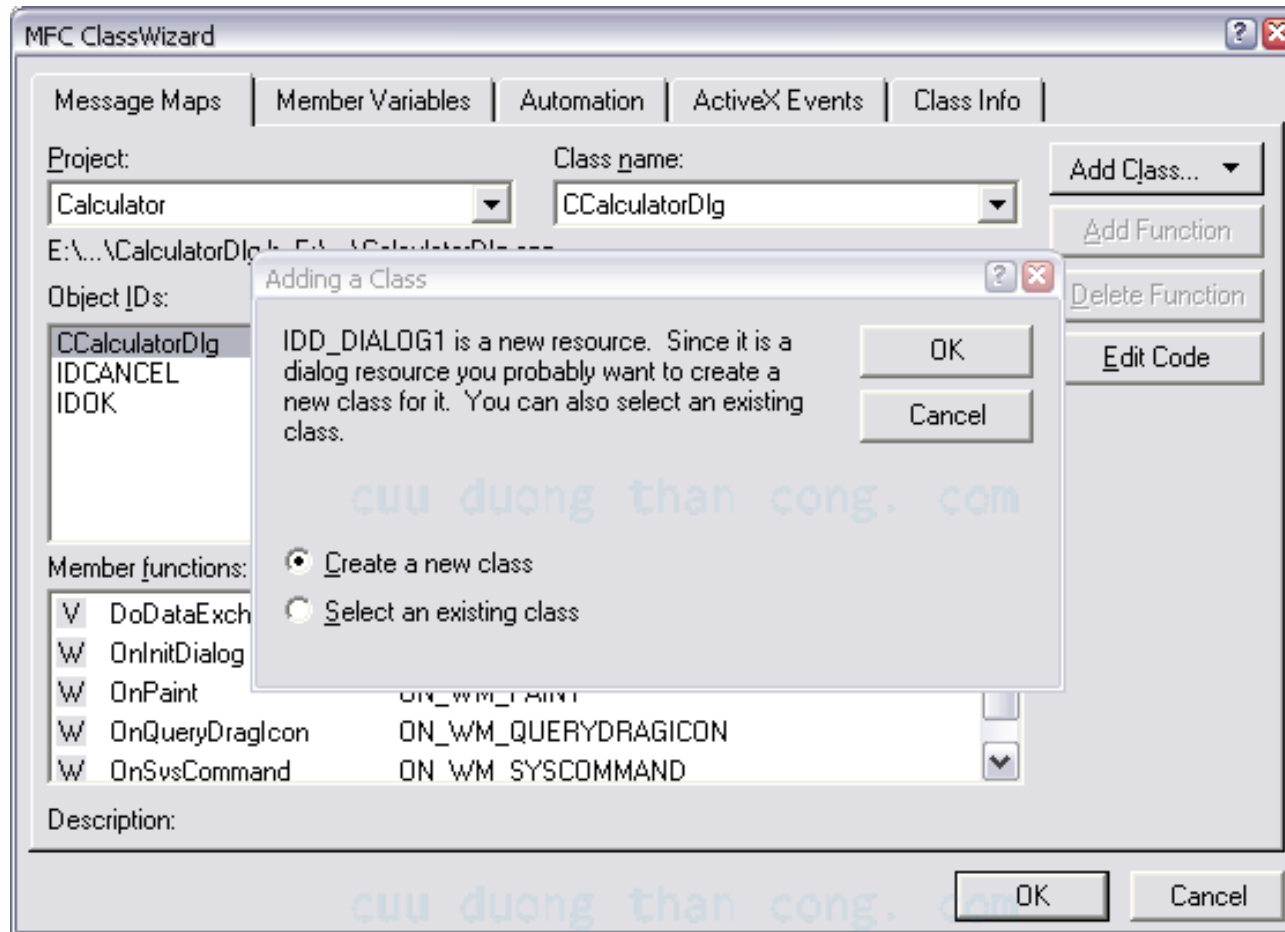
Thêm mới dialog resource (tt)



Xác lập thuộc tính của Dialog

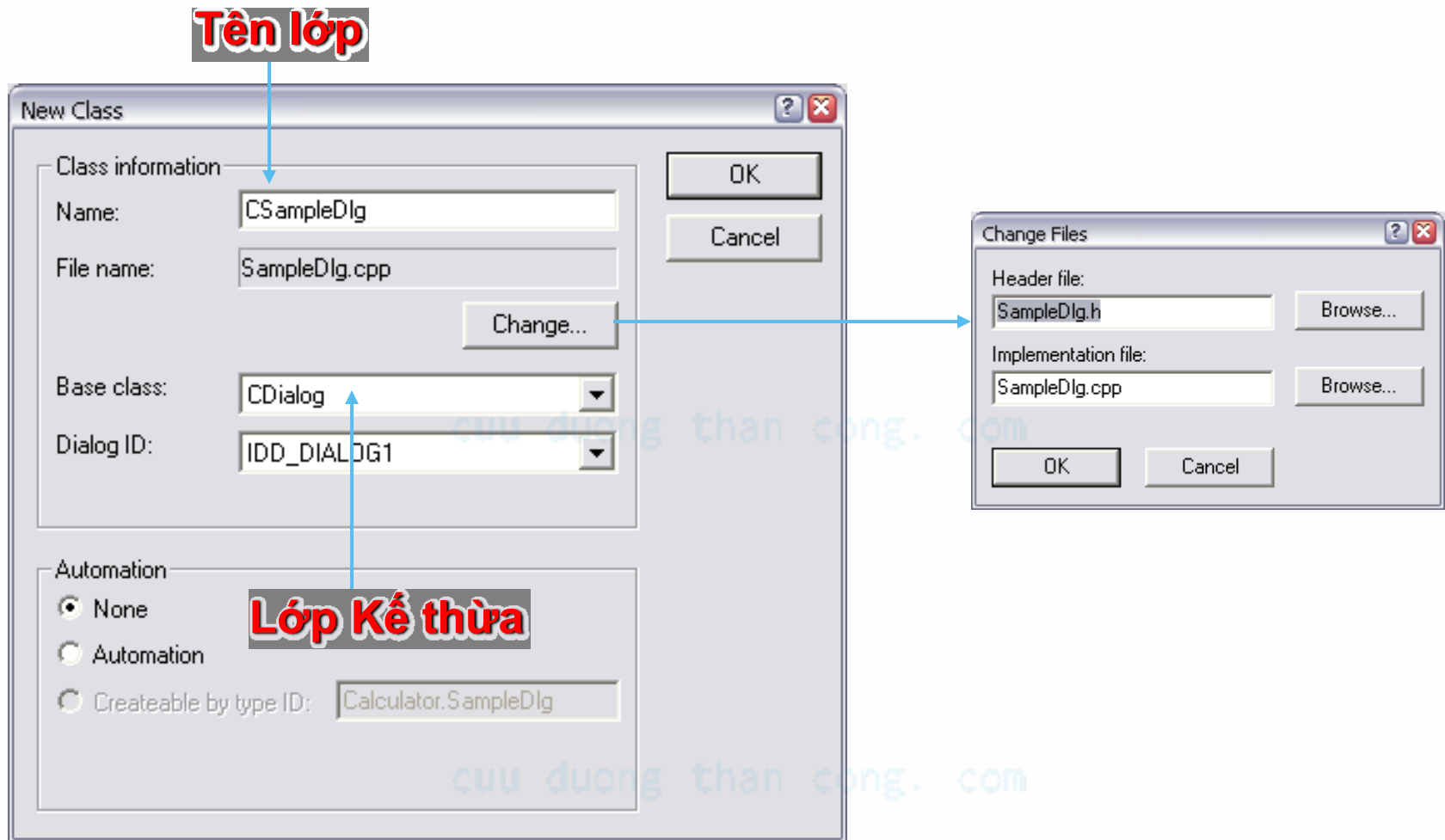


Khai báo lớp quản lý dialog



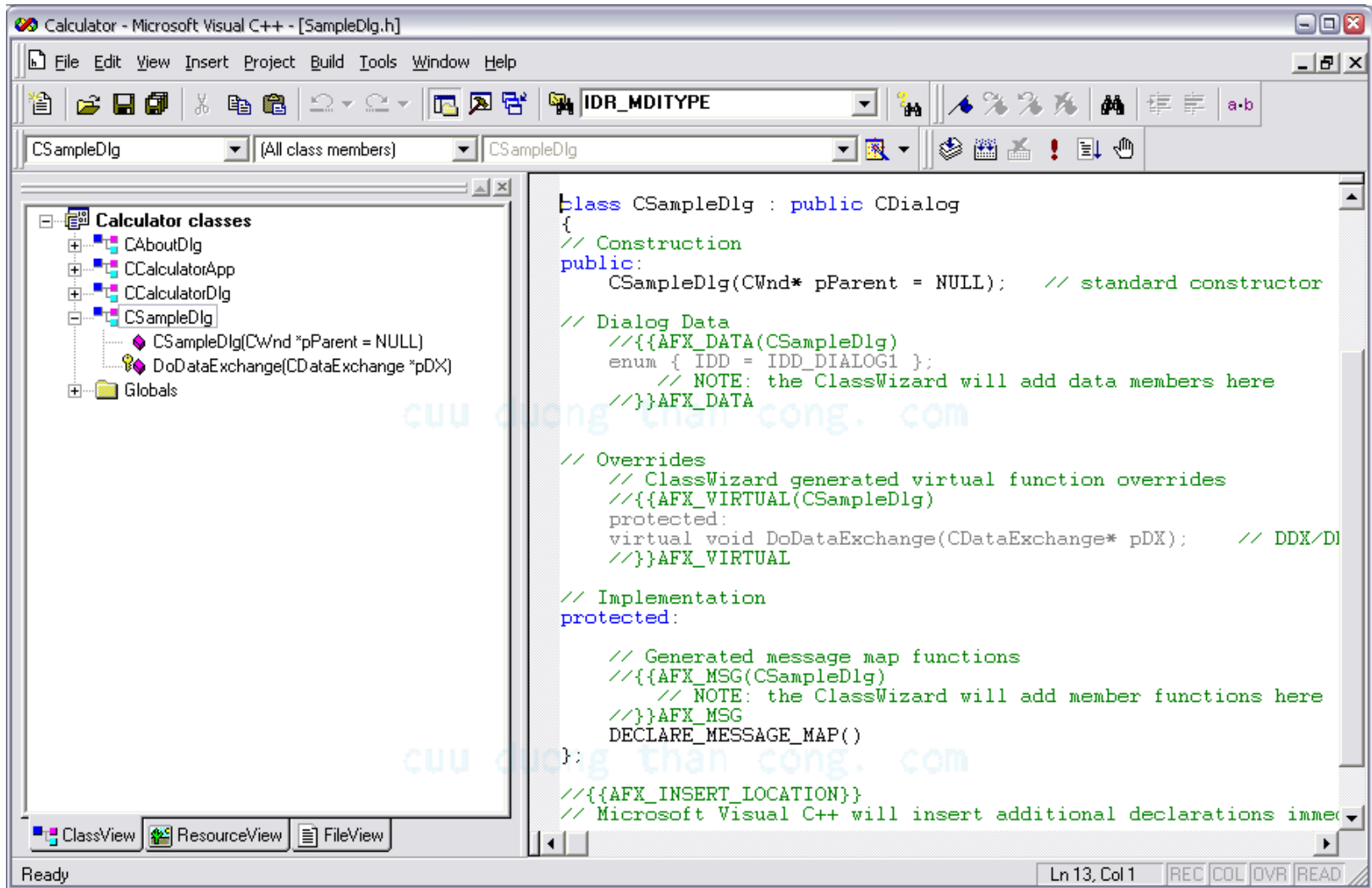
Ctrl + W để khai báo lớp xử lý tương ứng

Khai báo lớp xử lý dialog

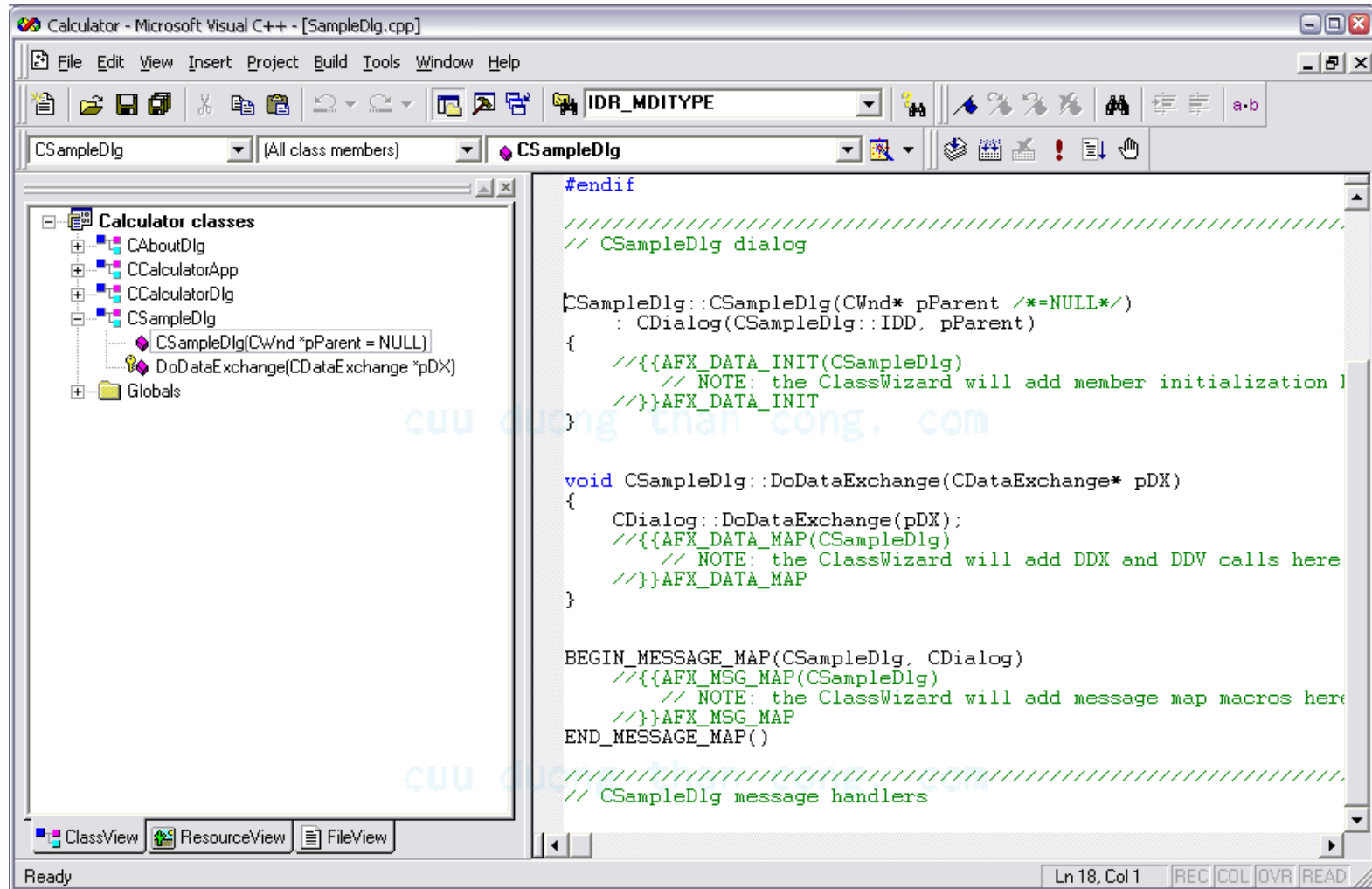


Ctrl + W để khai báo lớp xử lý tương ứng

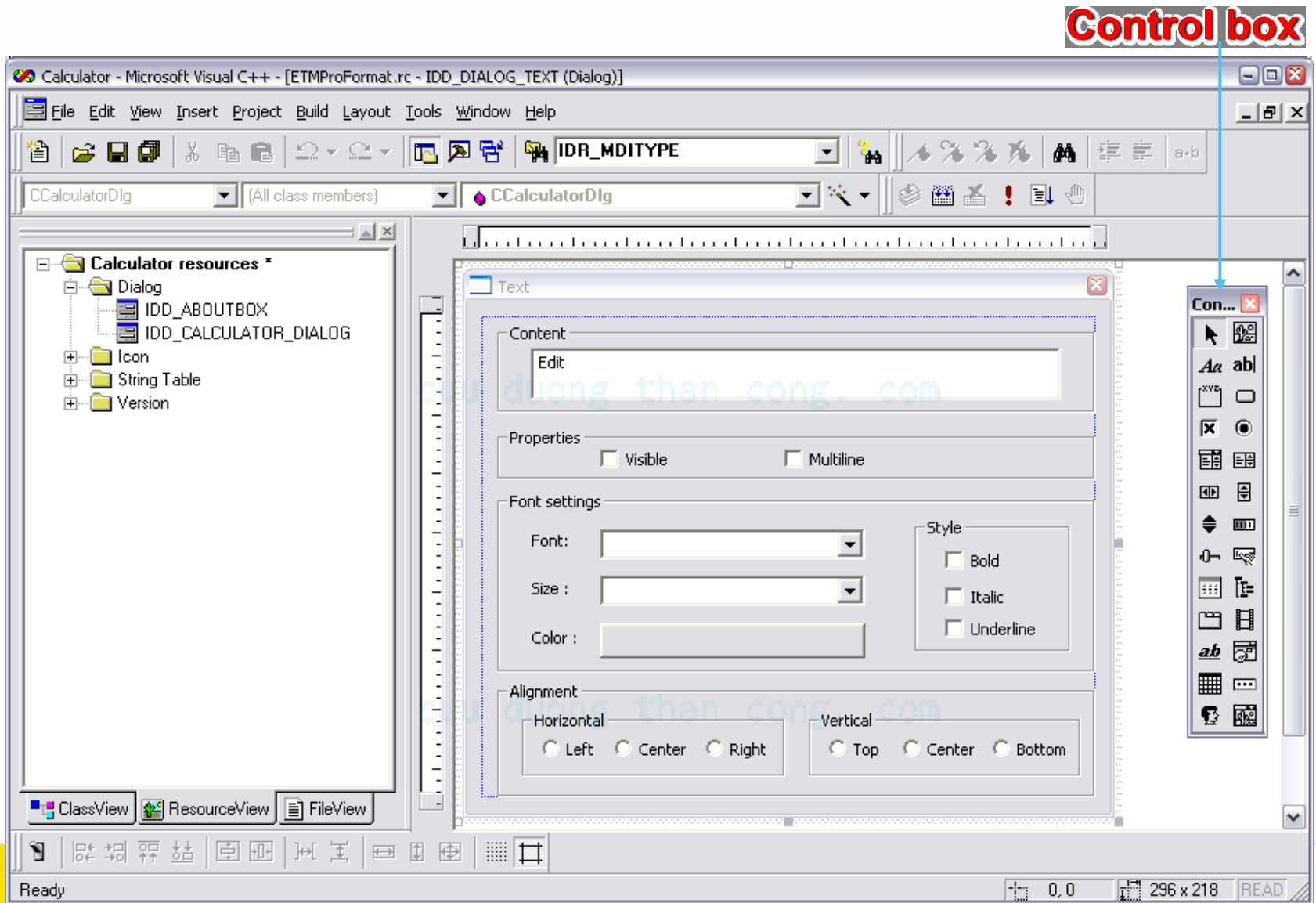
Header file - SampleDlg.h



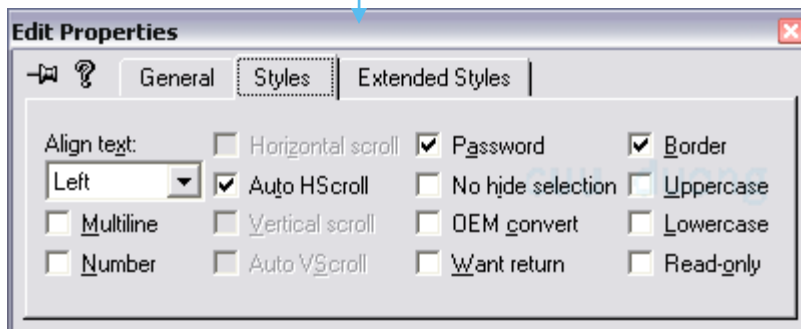
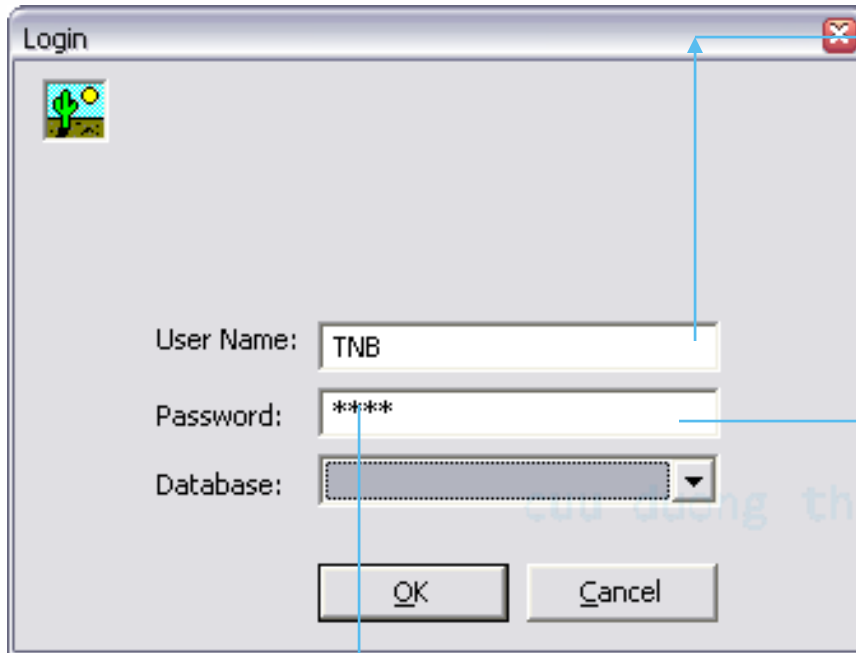
Source file - SampleDlg.cpp



Tạo và xử các control trong dialog



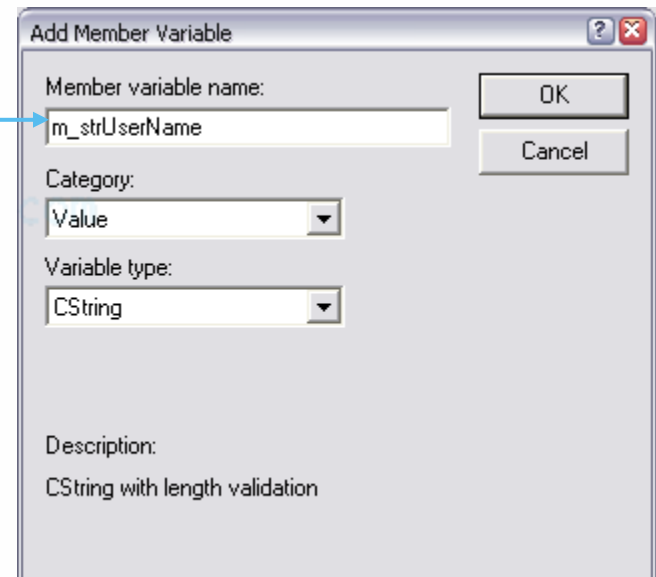
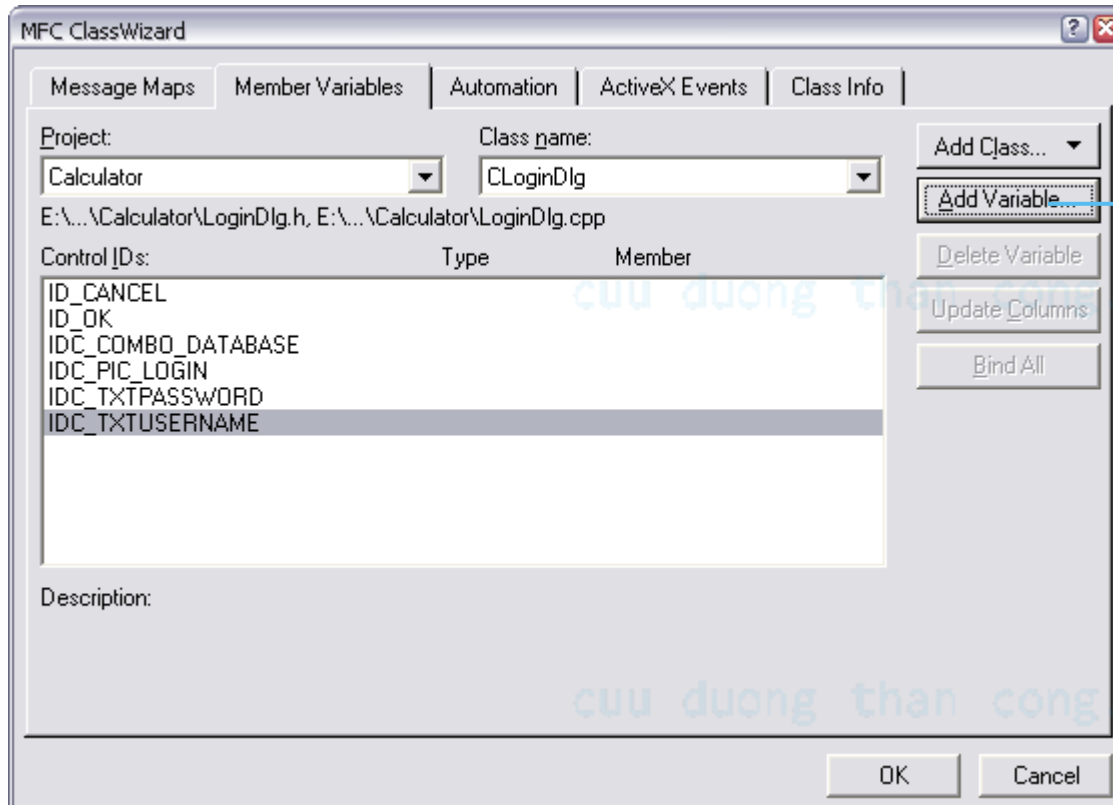
Xử lý control trong dialog box - EditText



Xử lý control trong dialog box - EditText

❖ Thông qua biến thành viên kiểu Value

Khai báo biến



Xử lý control trong dialog box - EditText

```
class CLoginDlg : public CDialog
{
// Construction
public:
    CString    m_strUserName;
    ....
}

void CLoginDlg::DoDataExchange (CDataExchange* pDX)
{
    CDialog::DoDataExchange (pDX) ;
    //{{AFX_DATA_MAP (CLoginDlg)
    DDX_Text (pDX, IDC_TXTUSERNAME, m_strUserName);
    //}}AFX_DATA_MAP
}
```


Xử lý sự kiện OnInitDialog

```
class CLoginDlg : public CDialog
{
...
protected:
    virtual BOOL OnInitDialog();
}

BOOL CLoginDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // TODO: Add extra initialization here
    return TRUE; // return TRUE unless you set the focus
                to a control
                // EXCEPTION: OCX Property Pages should
    return FALSE
}

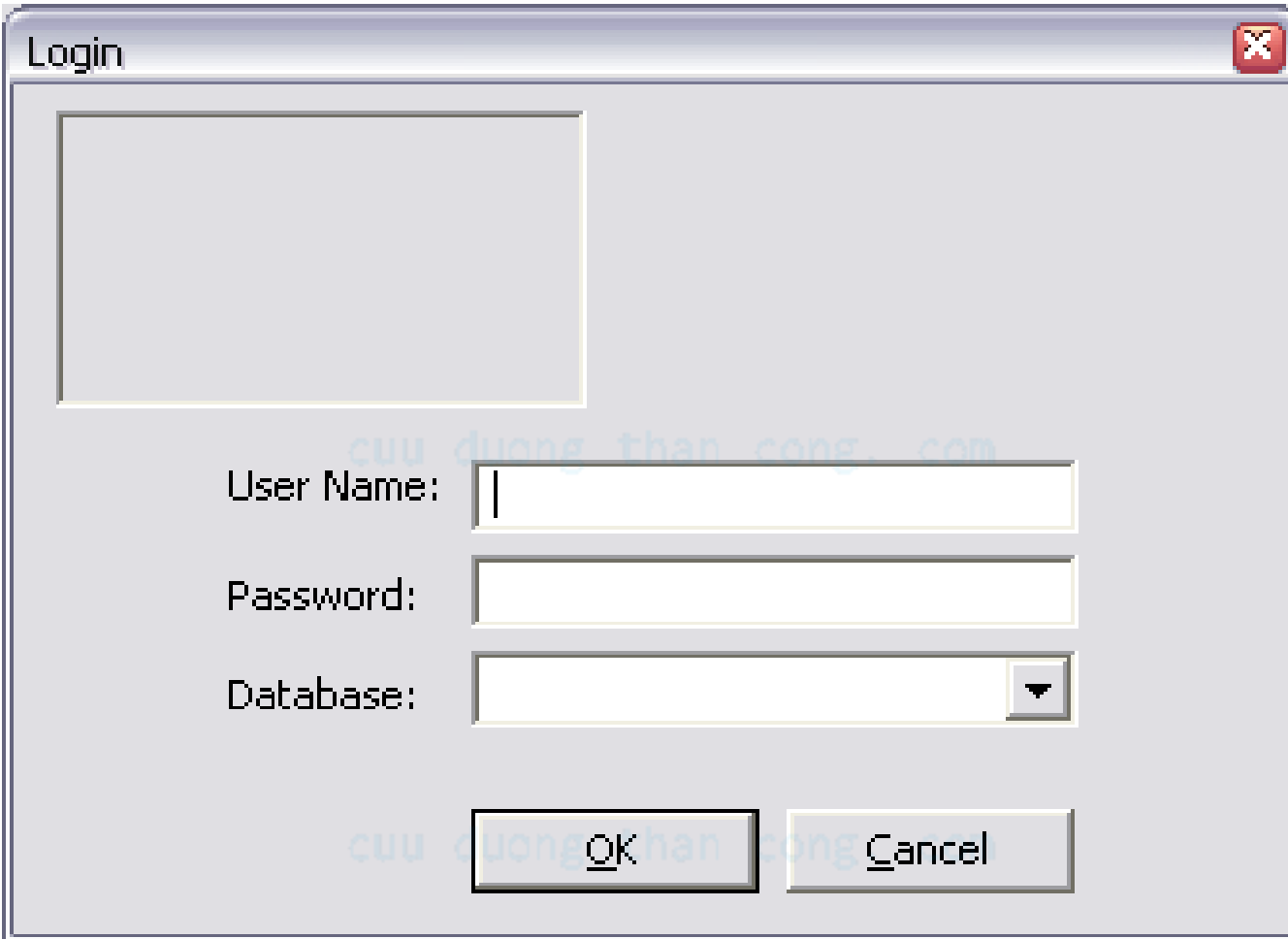
```

Gán giá trị cho EditText

```
BOOL CLoginDlg::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
    // TODO: Add extra initialization here  
    m_strUserName = "Nguyen Van A";  
    cuu duong than cong. com  
    return TRUE; // return TRUE unless you set the  
    focus to a control  
    // EXCEPTION: OCX Property Pages  
    should return FALSE  
}
```

cuu duong than cong. com

Gán giá trị cho EditText



A screenshot of a Java Swing dialog box titled "Login". The dialog box has a standard Windows-style title bar with a close button (red X) in the top right corner. Inside the dialog, there is a large, empty rectangular area in the top left. Below this area, there are three input fields: "User Name:" followed by a text field, "Password:" followed by a text field, and "Database:" followed by a text field with a dropdown arrow on the right. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Login

User Name:

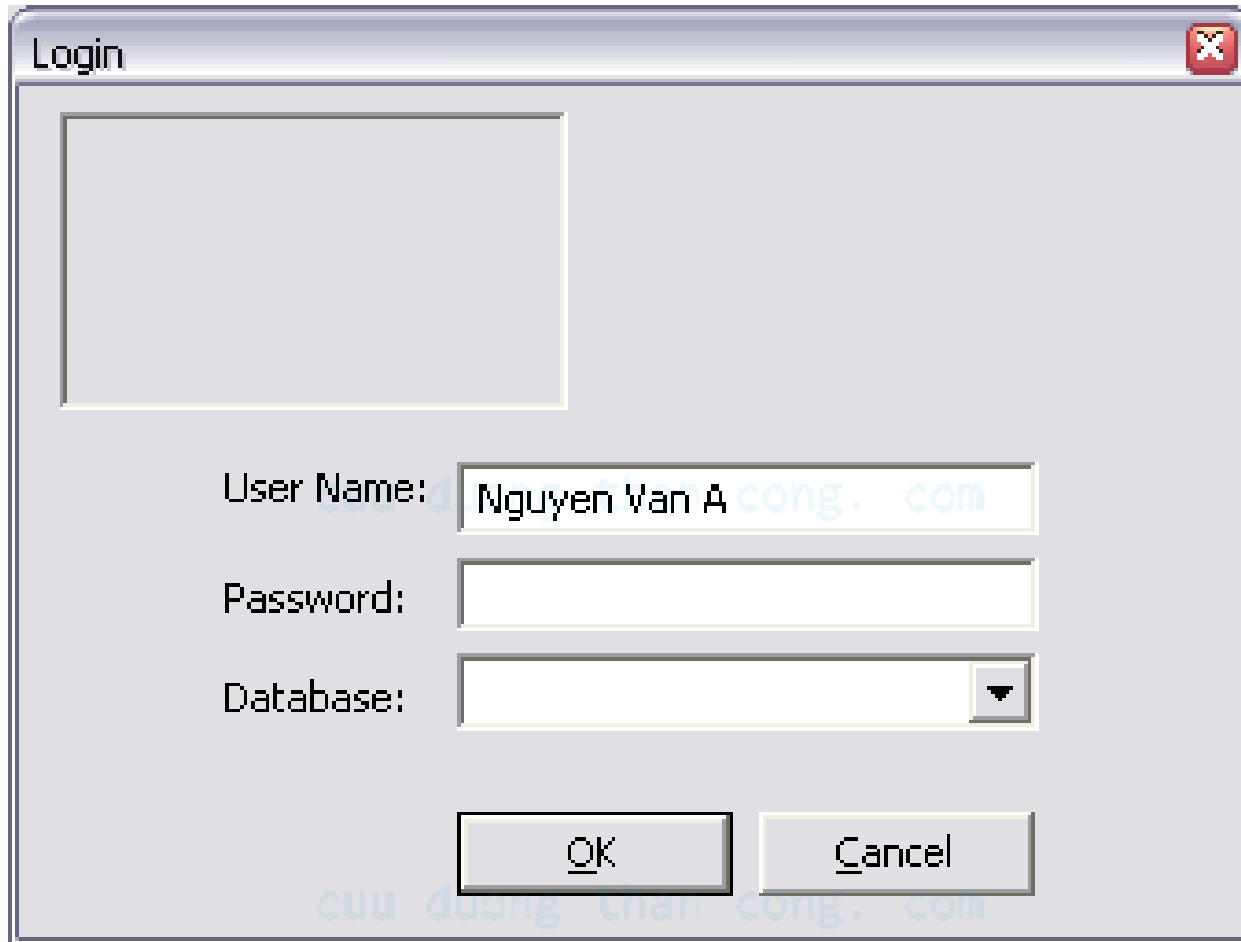
Password:

Database:

Gán giá trị cho EditText

```
BOOL CLoginDlg::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
    // TODO: Add extra initialization here  
    m_strUserName = "Nguyen Van A";  
    UpdateData(FALSE);  
    return TRUE; // return TRUE unless you set the  
                // EXCEPTION: OCX Property Pages  
                focus to a control  
    should return FALSE  
}
```

Gán giá trị cho EditText



Login

User Name: Nguyen Van A

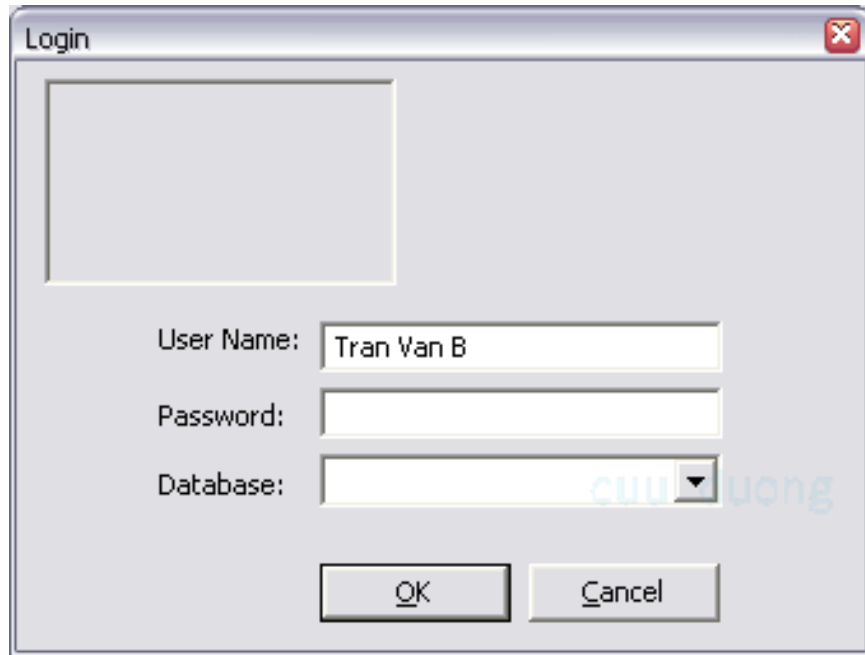
Password:

Database:

OK Cancel

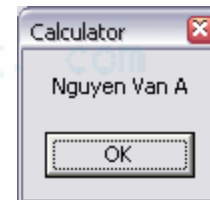
UpdateData(FALSE);

Lấy giá trị từ EditText

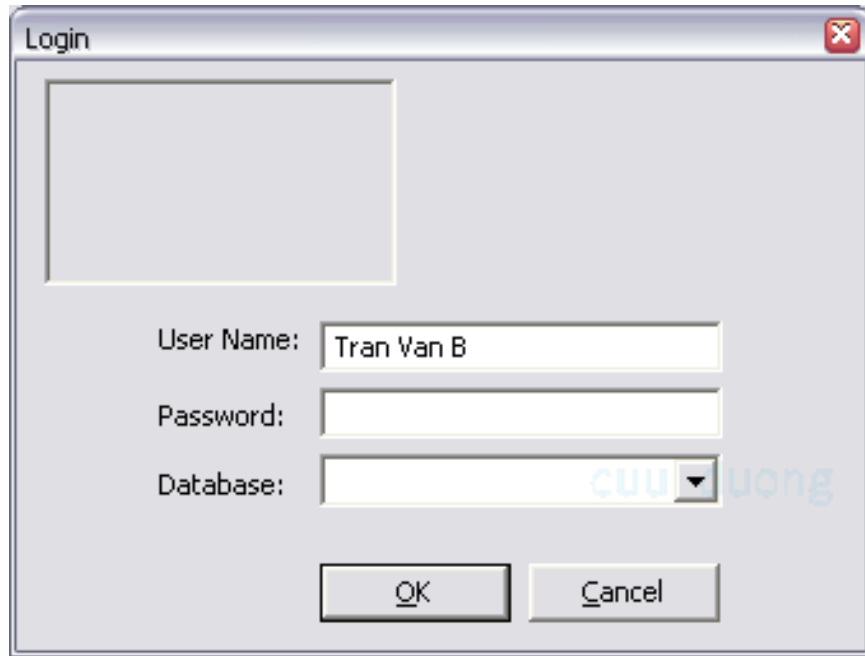


```
void CLoginDlg::OnOK()  
{  
    // TODO: Add extra validation  
    here  
    CString strUserName;  
    strUserName = m_strUserName;  
    MessageBox(strUserName);  
    CDialog::OnOK();  
}
```

Kết quả strUserName = “Nguyen Van A”

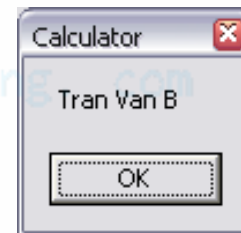


Lấy giá trị từ EditText

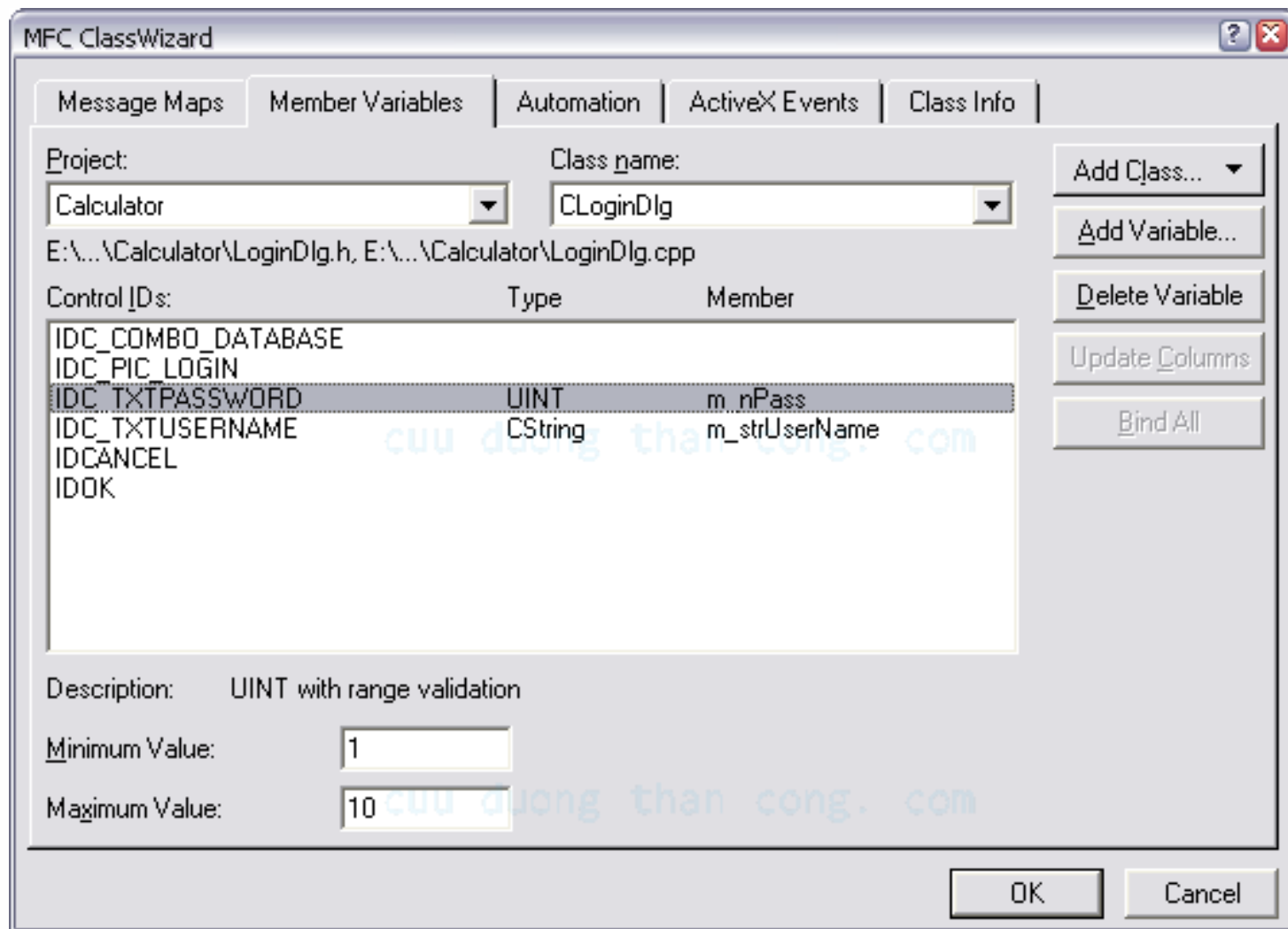


```
void CLoginDlg::OnOK()  
{  
    // TODO: Add extra validation  
    here  
    CString strUserName;  
    UpdateData(TRUE);  
    strUserName = m_strUserName;  
    MessageBox(strUserName);  
    CDialog::OnOK();  
}
```

Kết quả strUserName = “Tran Van B”



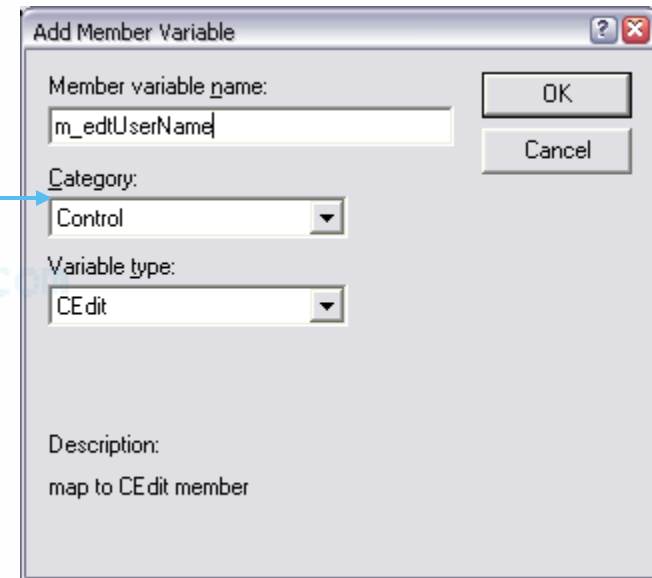
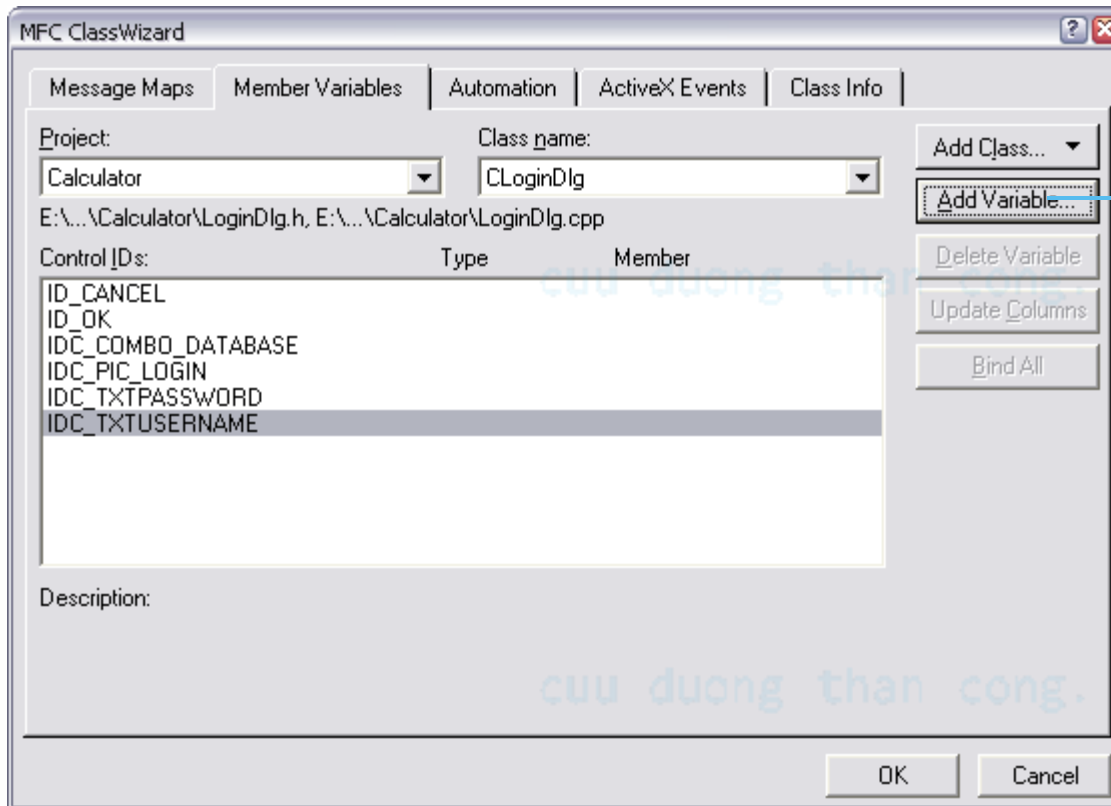
EditText - Dữ liệu loại VALUE



Xử lý control trong dialog box - EditText

❖ Thông qua biến thành viên kiểu CEdit

Khai báo biến



Xử lý control trong dialog box - EditText

```
class CLoginDlg : public CDialog
{
// Construction
public:
    CEdit      m_edtUserName;
    ...
}
void CLoginDlg::DoDataExchange (CDataExchange* pDX)
{
    CDialog::DoDataExchange (pDX) ;
   //{{AFX_DATA_MAP (CLoginDlg)
    DDX_Control (pDX, IDC_TXTUSERNAME, m_edtUserName);
    //}}AFX_DATA_MAP
}
```

Gán giá trị cho EditText

```
BOOL CLoginDlg::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
    // TODO: Add extra initialization here  
    m_edtUserName.SetWindowText("Nguyen Van A");  
    UpdateData(FALSE);  
    return TRUE; // return TRUE unless you set the  
                // EXCEPTION: OCX Property Pages  
                // should return FALSE  
}
```

cuu duong than cong. com

Lấy giá trị từ EditText

```
void CLoginDlg::OnOK()  
{  
    // TODO: Add extra validation here  
    CString strUserName;  
    UpdateData(TRUE);  
    m_edtUserName.GetWindowText(strUserName);  
    MessageBox(strUserName);  
  
    CDialog::OnOK();  
}
```

cuu duong than cong. com

Lấy và gán giá trị cho EditText

```
void CLoginDlg::OnOK()  
{  
    // Lấy giá trị từ EditText  
    CString strUserName;  
    GetDlgItem(IDC_TXTUSERNAME)->GetWindowText(strUserName);  
    MessageBox(strUserName);  
  
    // Gán giá trị vào EditText  
    CString strUserName="Nguyen Van A";  
    GetDlgItem(IDC_TXTUSERNAME)->SetWindowText(strUserName);  
  
    CDialog::OnOK();  
}
```

Lấy và gán giá trị cho EditText

```
void CLoginDlg::OnOK()  
{  
    // Lấy giá trị từ EditText  
    CString strUserName;  
    CEdit * pe;  
    pe = (CEdit*) GetDlgItem(IDC_TXTUSERNAME);  
    pe->GetWindowText(strUserName);  
    MessageBox(strUserName);  
  
    // Gán giá trị vào EditText  
    CString strUserName="Nguyen Van A";  
    pe->SetWindowText(strUserName);  
  
    CDialog::OnOK();  
}
```

Enable/Disable EditText

```
void CLoginDlg::Enable()
```

```
{  
    .....  
    m_edtUserName.EnableWindow(TRUE);  
    ....  
}
```

cuu duong than cong. com

```
void CLoginDlg::Disable()
```

```
{  
    .....  
    GetDlgItem(IDC_TXTUSERNAME)->EnableWindow(FALSE);  
    ....  
}
```

cuu duong than cong. com

Ví dụ minh họa

INU C4W Sample 04

STUDENT LIST

Code	Name	Address	Phone	Email
05.020.001	Ly Thanh	123 Truong Dinh	919123567	lythanh@yaf
05.020.002	Nguyen Duc Trung	153/2 Hoang Van Thu	958201234	trungnd@yal

Add New Update Delete Close

Student Info

STUDENT INFO

Code 05.020.002

Name Nguyen Duc Trung

Address 153/2 Hoang Van Thu

Phone 0958201234

Email trungnd@yahoo.com

OK Cancel

Gọi dialog trong các hàm xử lý

```
#include "MyClass.h"  
#include "LoginDlg.h"
```

```
void CMyClass::ShowMyDlg()  
{  
    CLoginDlg dlg;  
    dlg.DoModal();  
}
```

Truyền và nhận dữ liệu với dialog

```
#include "MyClass.h"
#include "LoginDlg.h"

void CMyClass::ShowMyDlg()
{
    CLoginDlg dlg;
    //Truyền giá trị cho các biến trong dialog
    int nRes = dlg.DoModal();
    if (nRes == IDOK)
    {
        //Nhận giá trị từ các biến trong dialog

    }
}
```

Truyền và nhận dữ liệu với dialog

```
void CMyClass::ShowMyDlg()  
{  
    CLoginDlg dlg;  
    //Truyền giá trị cho các biến trong dialog  
    CString s;  
  
    dlg.m_editUserName.SetWindowText(s);  
    int nRes = dlg.DoModal();  
    if (nRes == IDOK)  
    {  
        //Nhận giá trị từ các biến trong dialog  
        dlg.m_editUserName.GetWindowText(s);  
        MessageBox(s);  
    }  
}
```

Xử lý control trong dialog box - EditText

```
class CLoginDlg : public CDialog
{
public:
    CString    m_strUser;
}
BOOL CLoginDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    m_edtUserName.SetWindowText(m_strUser);
    return TRUE;
}
void CLoginDlg::OnOK()
{
    m_edtUserName.SetWindowText(m_strUser);
    CDialog::OnOK();
}
```

Truyền và nhận dữ liệu với dialog (tt)

```
void CMyClass::ShowMyDlg()  
{  
    CLoginDlg dlg;  
    //Truyền giá trị cho các biến trong dialog  
    CString s;  
    dlg.m_strUser = s;  
    int nRes = dlg.DoModal();  
    if (nRes == IDOK)  
    {  
        //Nhận giá trị từ các biến trong dialog  
        s = dlg.m_strUser;  
        MessageBox(s);  
    }  
}
```

Ví dụ minh họa

INU C4W Sample 04

STUDENT LIST

Code	Name	Address	Phone	Email
05.020.001	Ly Thanh	123 Truong Dinh	919123567	lythanh@yaf
05.020.002	Nguyen Duc Trung	153/2 Hoang Van Thu	958201234	trungnd@yal

Add New Update Delete Close

Student Info

STUDENT INFO

Code 05.020.002

Name Nguyen Duc Trung

Address 153/2 Hoang Van Thu

Phone 0958201234

Email trungnd@yahoo.com

OK Cancel

Dialog Box Common Controls Property Sheet

Lập trình với Common Controls

NỘI DUNG

- ❖ Lập trình với StaticText
- ❖ Lập trình với Combobox
- ❖ Lập trình với Listbox
- ❖ Lập trình với Check Box
- ❖ Lập trình với Radio button
- ❖ Lập trình với Listview & Treeview
- ❖ Lập trình với ActiveX control (3-party components)

Ví dụ về lập trình với common control

Viết chương trình nhập/xuất thông tin sinh viên, học sinh, giáo viên,...từ tập tin

Thông tin sinh viên

THÔNG TIN SINH VIÊN

Họ và chữ lót: Nguyễn Văn Tên: Thịnh

Ngày sinh: 05/11/2005 Tôn giáo: Không

Địa chỉ: 123 Trương Định - Q3

Điện thoại: 8234567 Email: nvthinh@hcmup.edu.vn

Dân tộc: Kinh

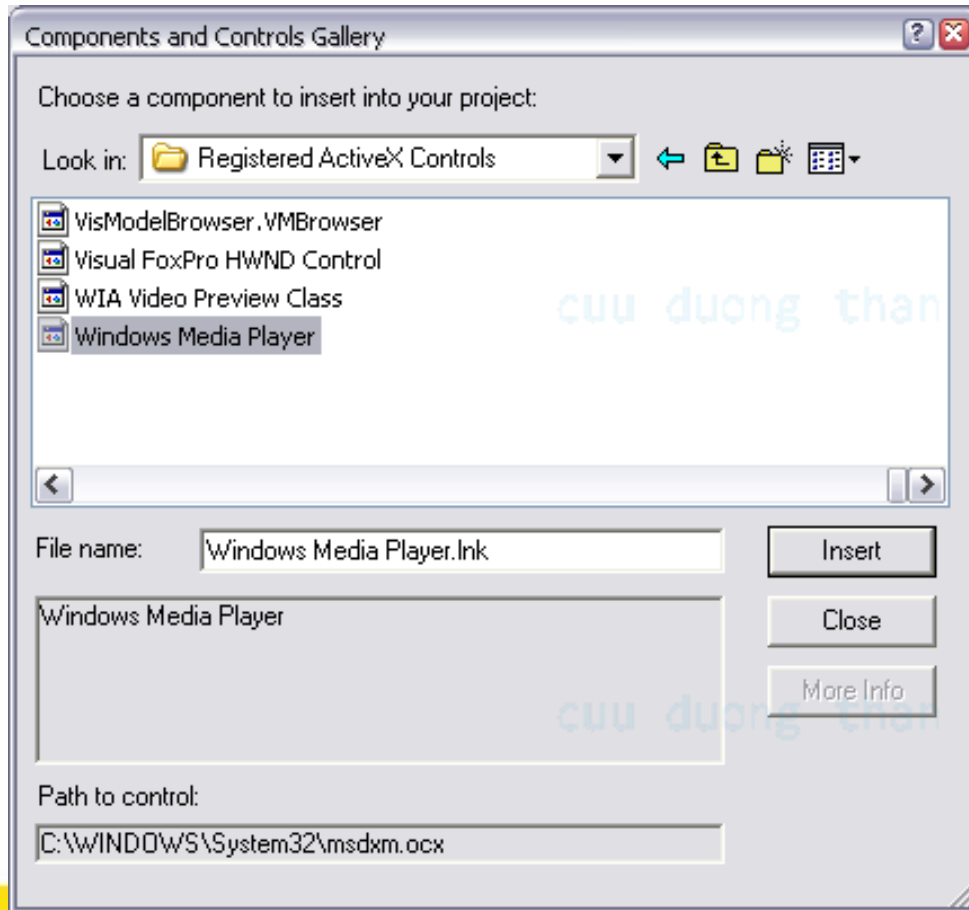
Giới tính: ☒ Nam ☐ Nữ

Ngoại ngữ: ☒ Anh văn ☒ Pháp văn ☐ Hán văn

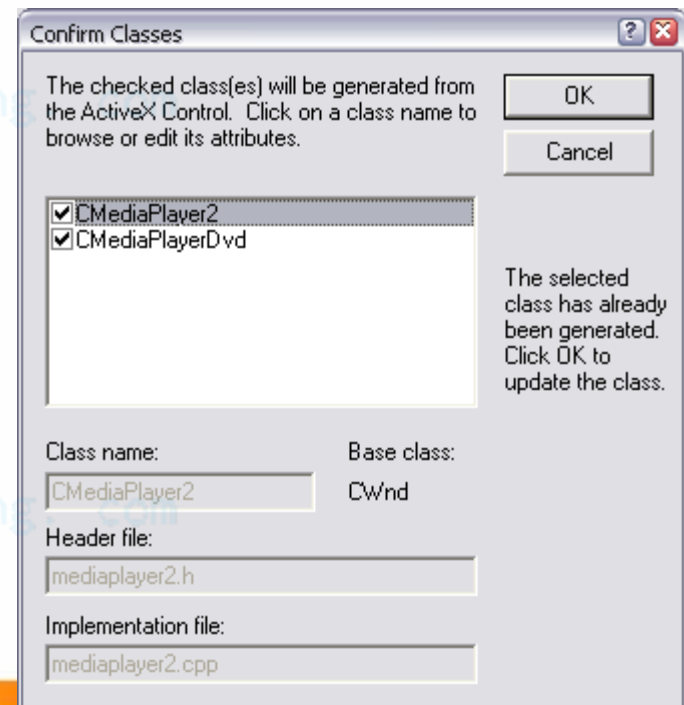
Đọc dữ liệu từ file Lưu dữ liệu xuống file Thoát

Thêm ActiveX control vào ứng dụng

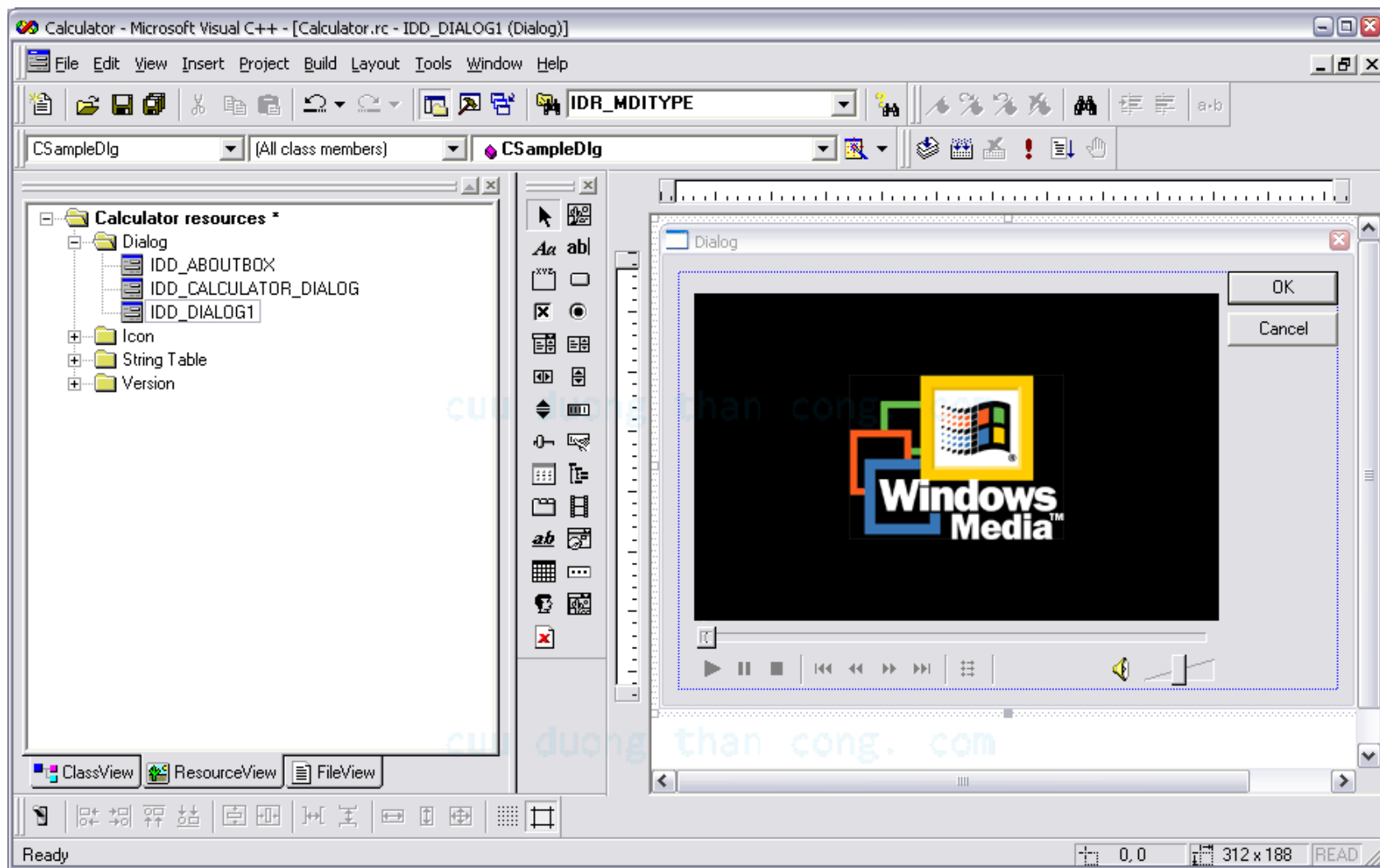
- ❖ Chọn menu **Project\Add to project\Components and Controls**



Thêm **Windows Media Player** control vào ứng dụng



Thêm ActiveX control vào ứng dụng

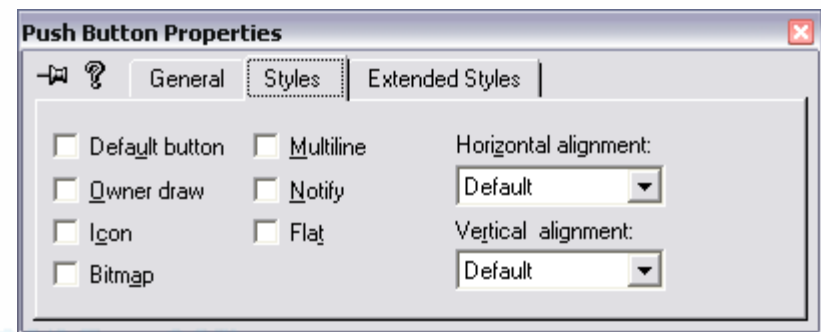
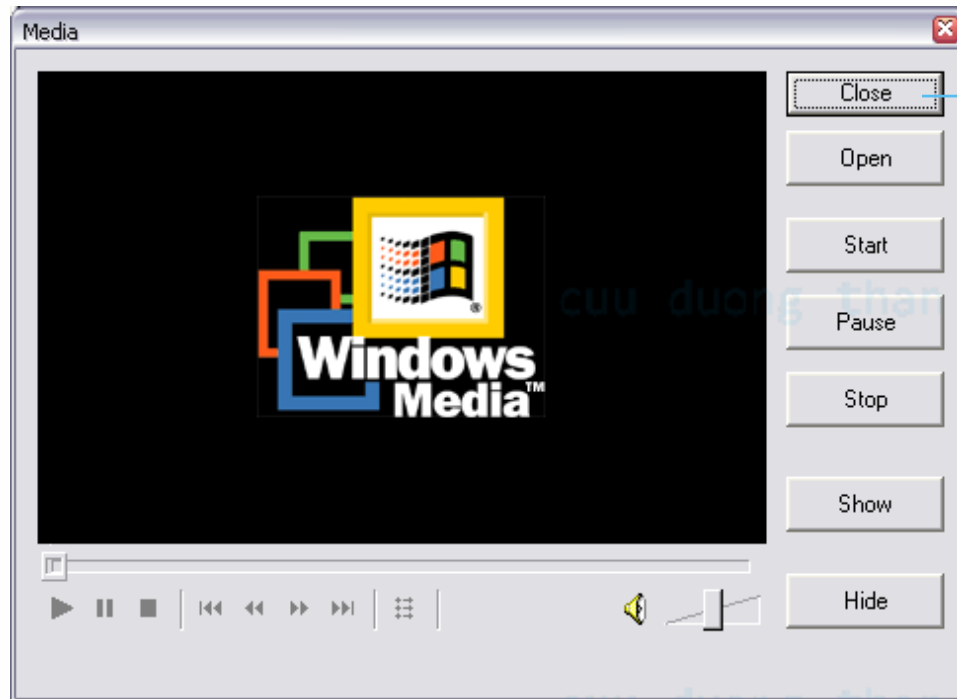


Thêm ActiveX control vào ứng dụng



Ứng dụng “nhúng” **Windows Media Player**

Xử lý control trong dialog box - Button



Viết lệnh xử lý Button Open

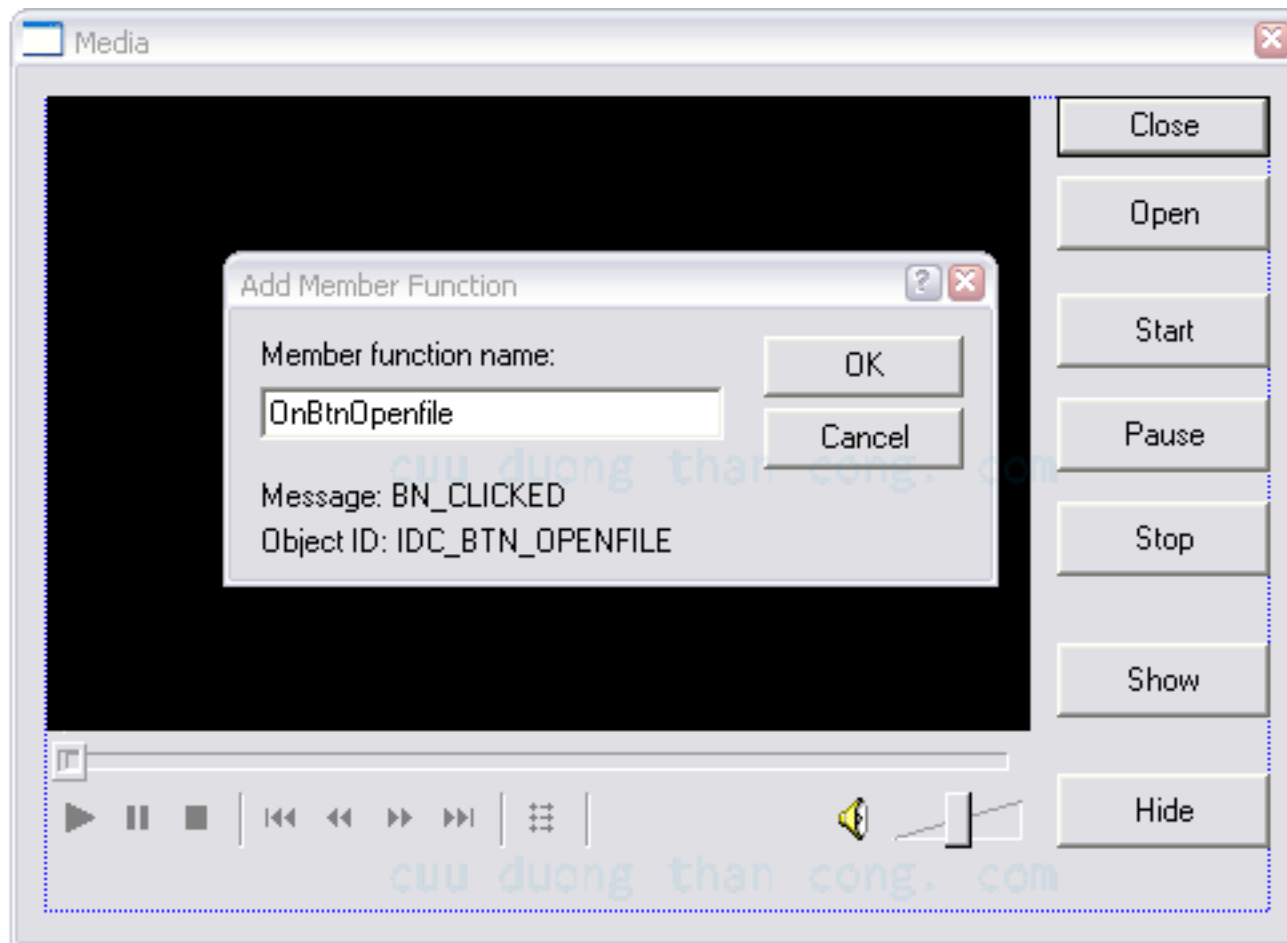
```
class CMediaDlg : public CDialog
{
   //{{AFX_MSG(CMediaDlg)
    afx_msg void OnBtnOpen();
    afx_msg void OnBtnShow();
    //{{AFX_MSG
}
```

```
BEGIN_MESSAGE_MAP(CMediaDlg, CDialog)
    //{{AFX_MSG_MAP(CMediaDlg)
    ON_BN_CLICKED(IDC_BTN_OPEN, OnBtnOpen)
    ON_BN_CLICKED(IDC_BTN_SHOW, OnBtnShow)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
void CMediaDlg::OnBtnOpen()
```

```
{
    // TODO: Add your control notification handler code here
    CString strFileName = _T("C:\\Music\\Track01.MP3");
    m_MediaPlayer.Open(strFileName);
}
```

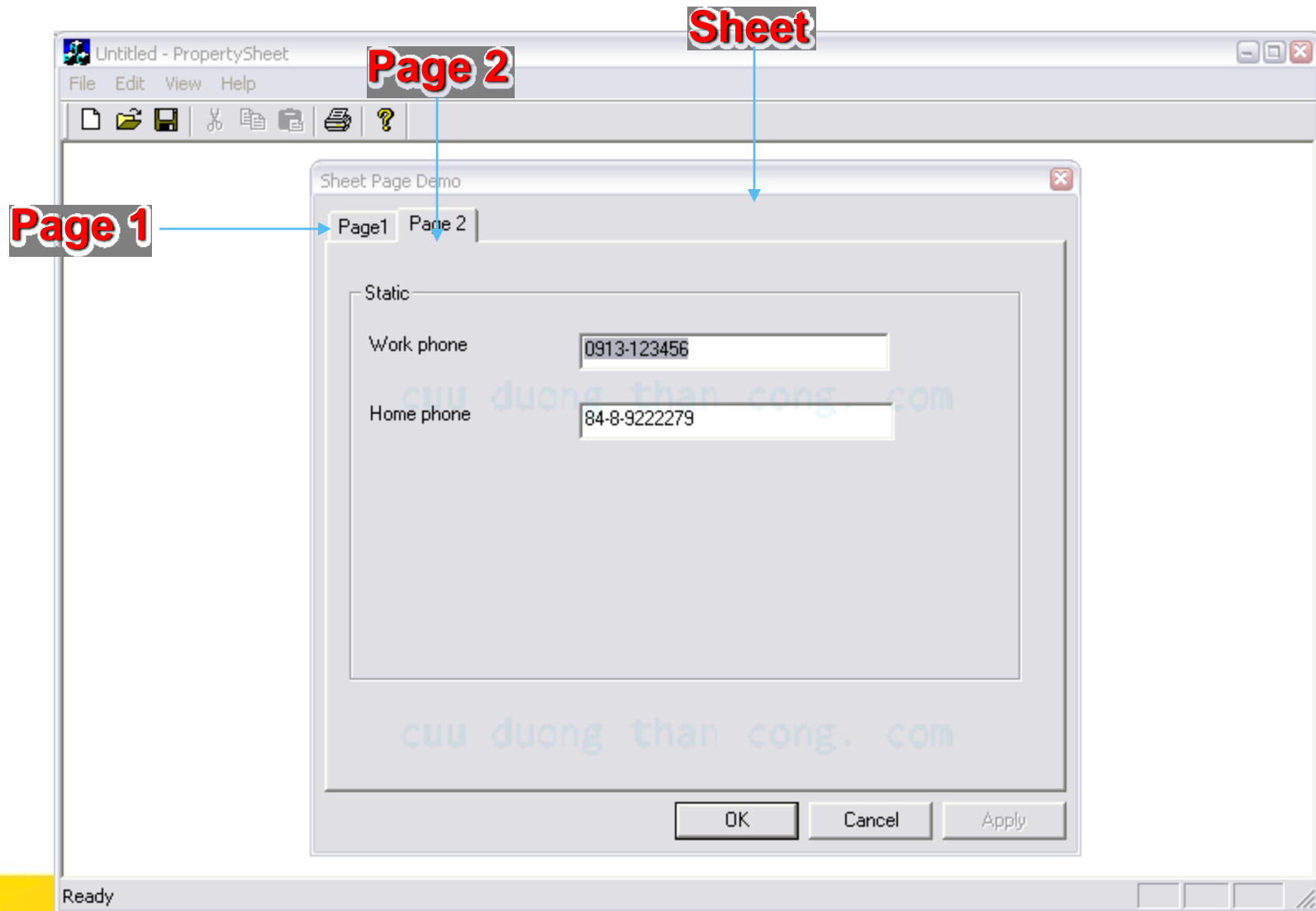
Tạo hàm xử lý Button Open bằng Wizard



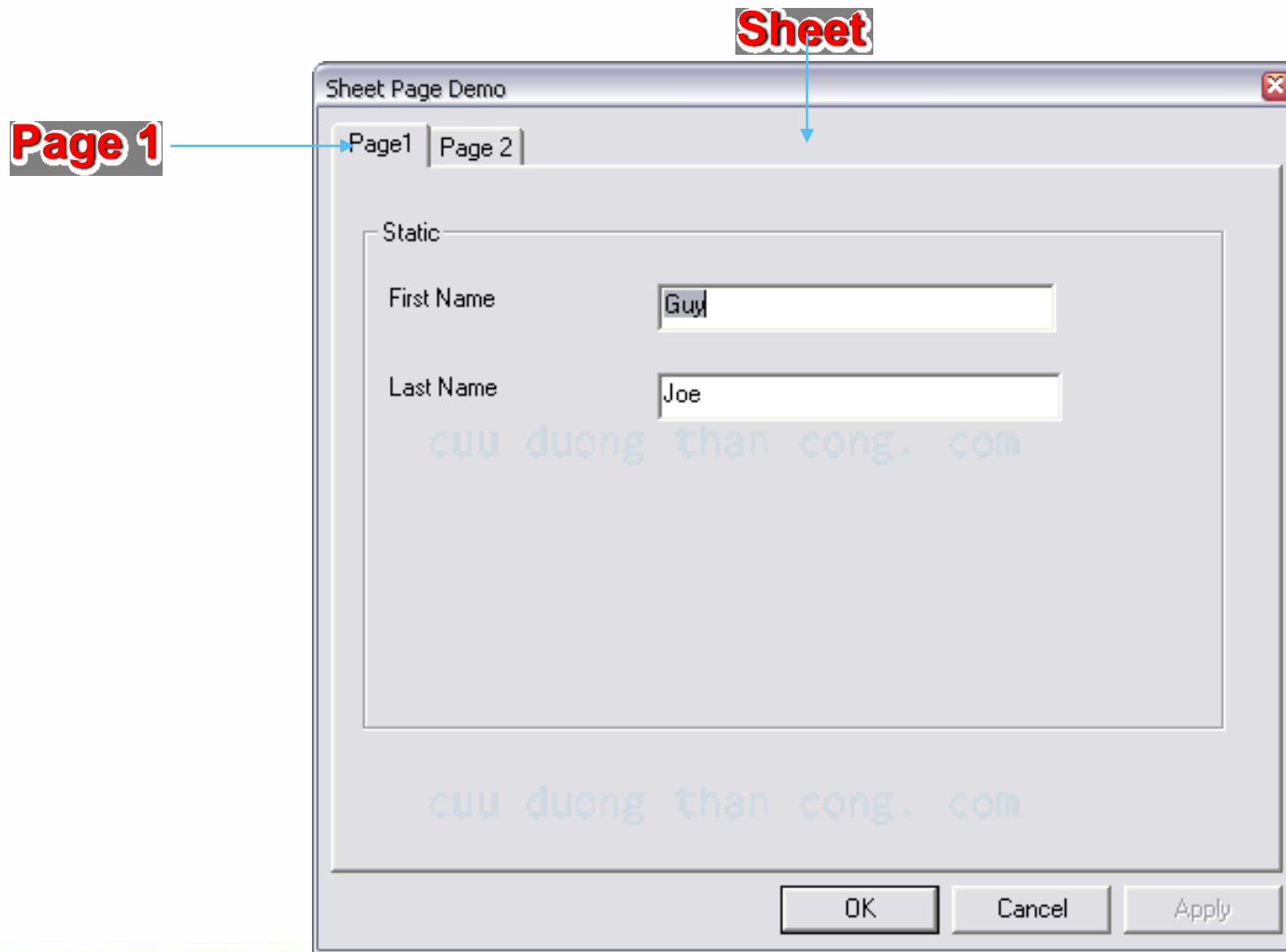
Dialog Box Common Controls Property Sheet

Lập trình với Property Sheet

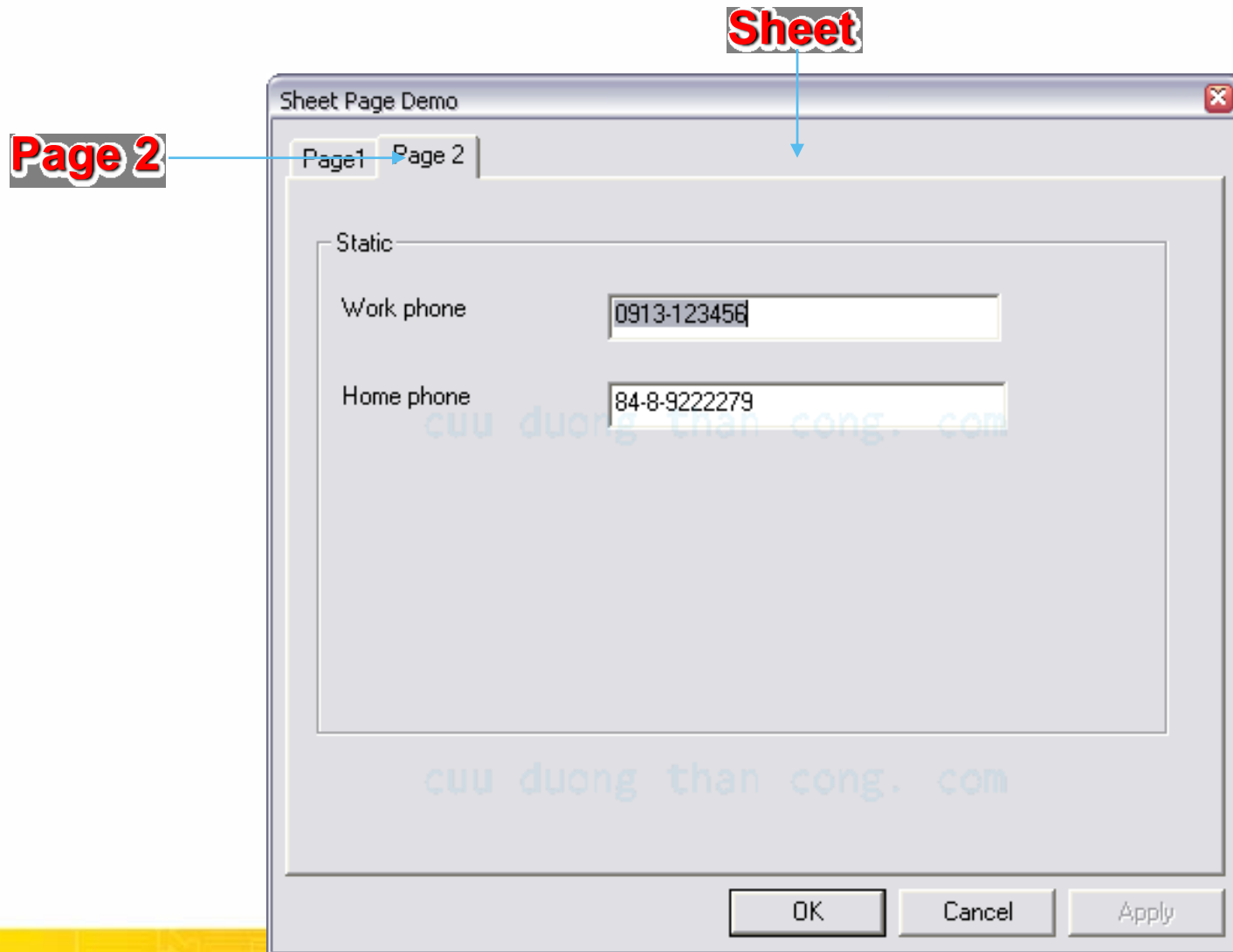
Các thành phần trong Property Sheet



Các thành phần trong Property Sheet



Các thành phần trong Property Sheet



Các thành phần trong Property Sheet

Page 1

Page1

Static

First Name

Last Name

Page 2

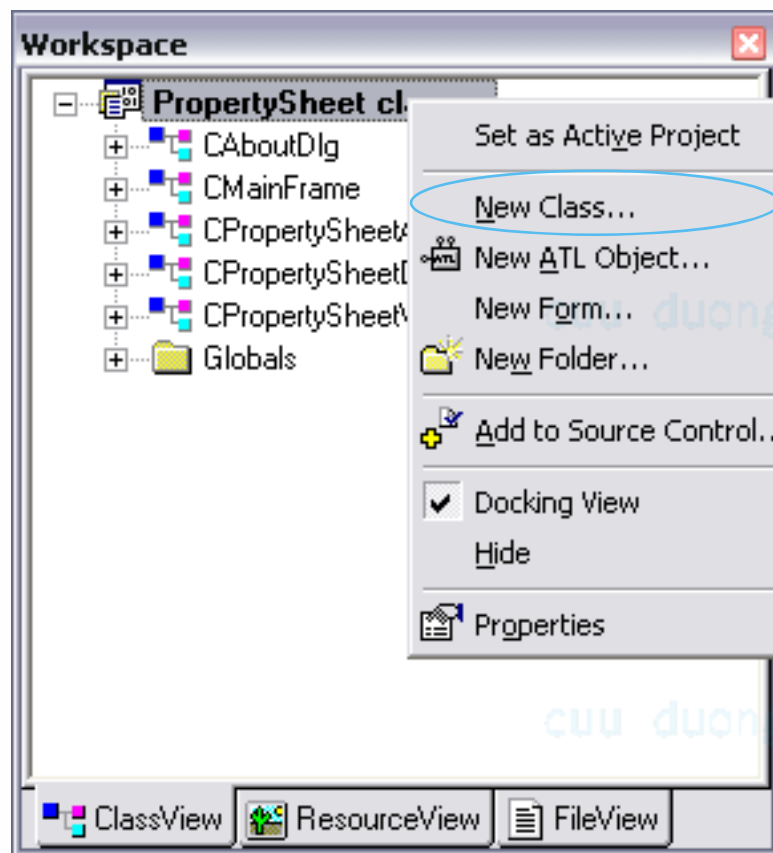
Page 2

Static

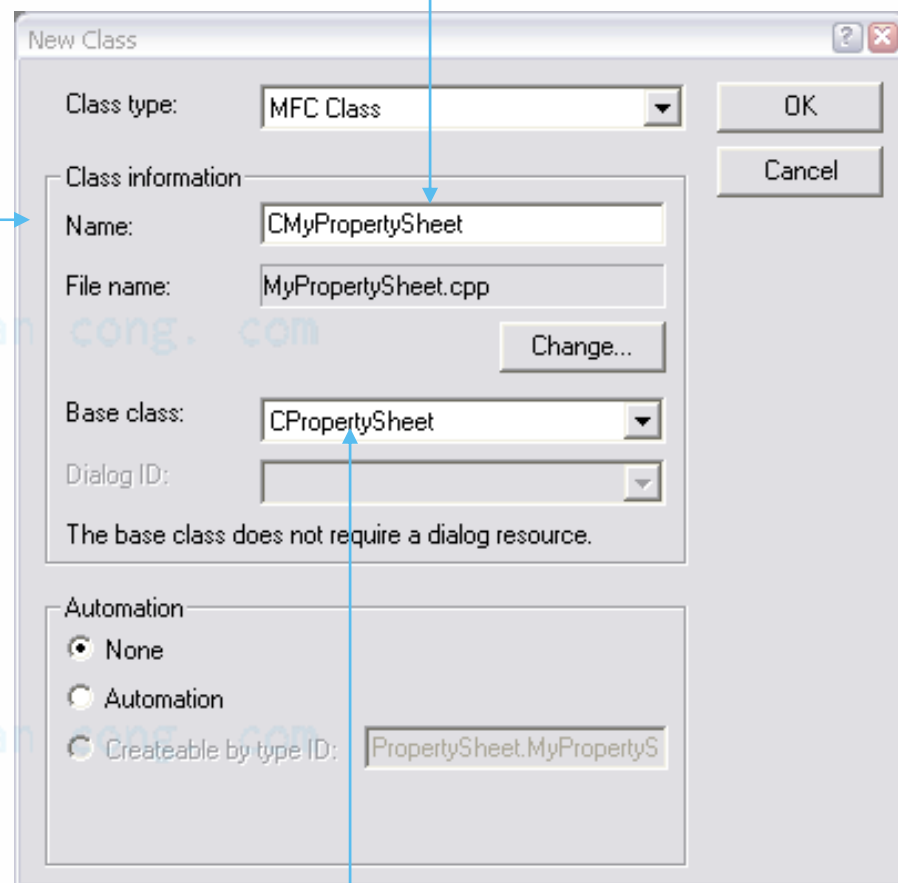
Work phone

Home phone

Thêm mới Property Sheet

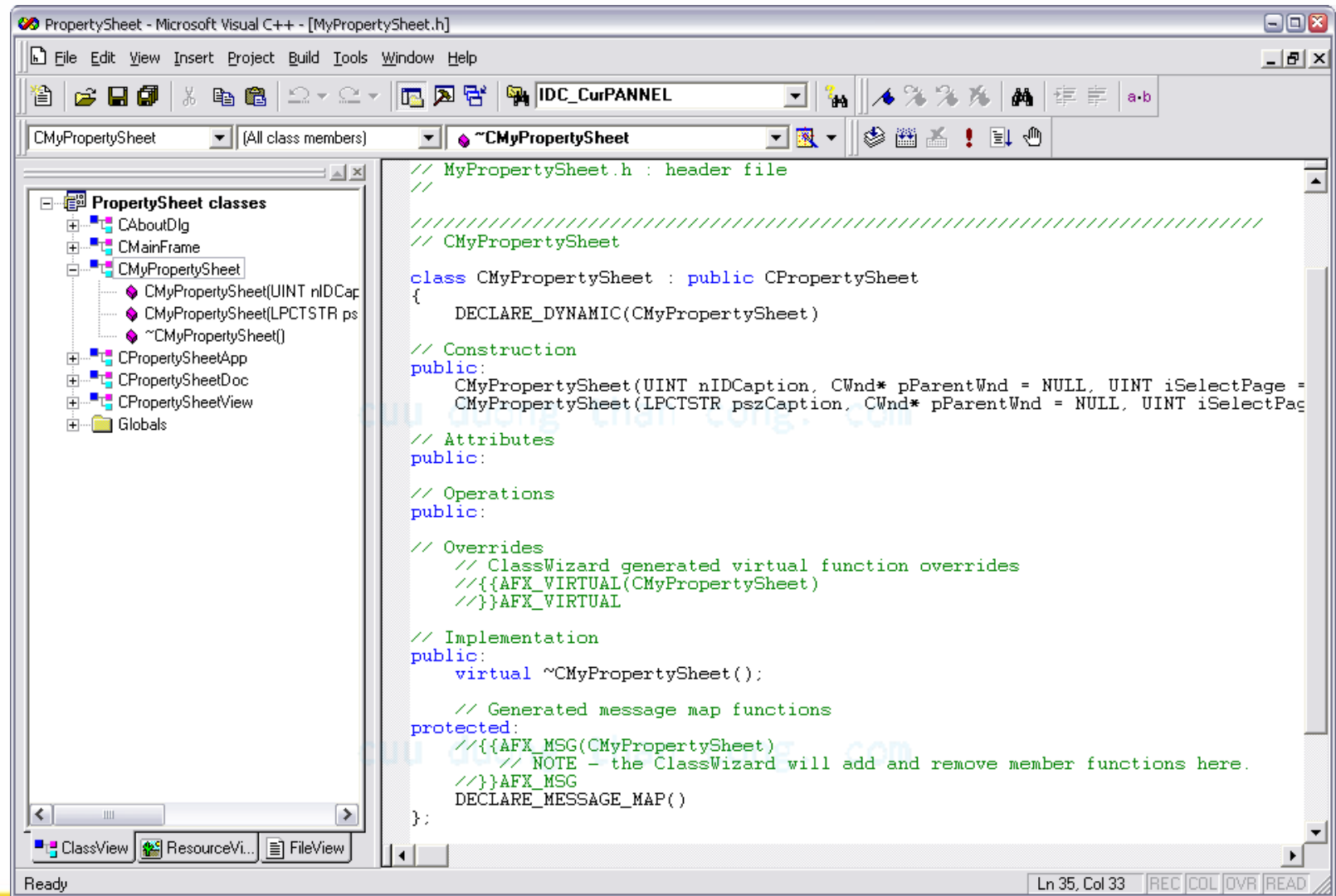


Nhập tên lớp

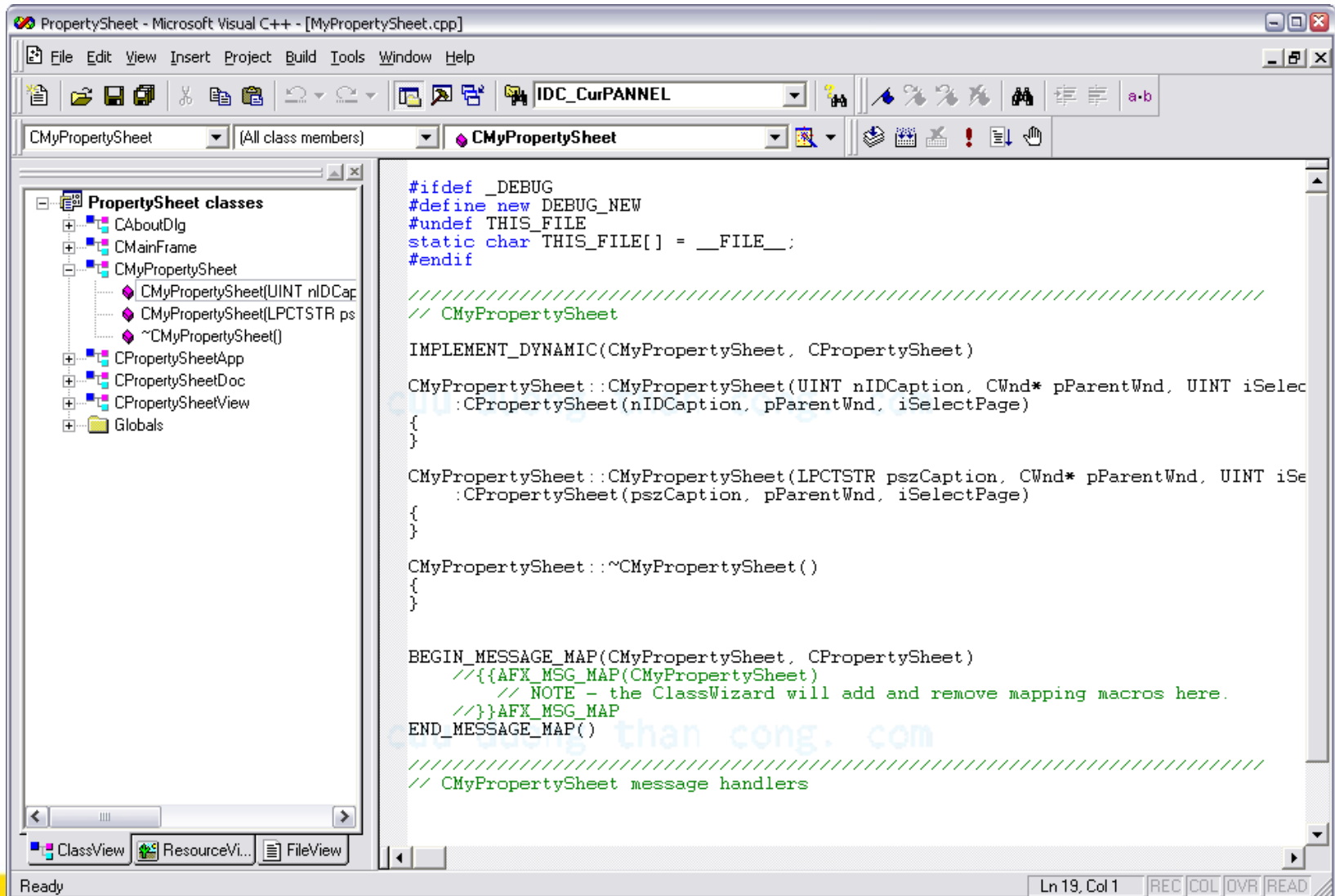


Chọn lớp kế thừa

Thêm mới Property Sheet (tt)



Thêm mới Property Sheet (tt)



Thêm mới Property Page

- ❖ Thêm mới Dialog tương ứng với mỗi Page trong ResourceView
- ❖ Khai báo lớp xử lý tương ứng với mỗi page, kế thừa từ lớp CPropertyPage
- ❖ Add các page vào Property Sheet

cuu duong than cong. com

Thêm mới Property Page

Nhập tên lớp

New Class

Class information

Name: CMyPage1

File name: MyPage1.cpp

Change...

Base class: CPropertyPage

Dialog ID: IDD_DIALOG1

Automation

☒ None

☐ Automation

☐ Createable by type ID: PropertySheet.MyPage1

OK

Cancel

**Chọn lớp
kế thừa**

**Khai báo lớp xử lý Page
1, kế thừa từ lớp
CPropertyPage**

Thêm mới Property Page

Nhập tên lớp

New Class

Class information

Name: CMyPage2

File name: MyPage2.cpp

Change...

Base class: CPropertyPage

Dialog ID: IDD_DIALOG2

Automation

☒ None

☐ Automation

☐ Createable by type ID: PropertySheet.MyPage2

OK

Cancel

**Chọn lớp
kế thừa**

**Khai báo lớp xử lý Page
2, kế thừa từ lớp
CPropertyPage**

Khai báo lớp CMyPropertySheet

```
#include "MyPage1.h"    // Added by ClassView
#include "MyPage2.h"    // Added by ClassView

class CMyPropertySheet : public CPropertySheet
{
    ...
public:
    virtual ~CMyPropertySheet();

    // Generated message map functions
protected:

    CMyPage1 m_myPage1;
    CMyPage2 m_myPage2;
    ...
}
```

Khởi tạo CMyPropertySheet

```
CMyPropertySheet::CMyPropertySheet(UINT nIDCaption, CWnd*
    pParentWnd, UINT iSelectPage)
    :CPropertySheet(nIDCaption, pParentWnd, iSelectPage)
{
    AddPage (&m_myPage1) ;
    AddPage (&m_myPage2) ;
}
```

cuu duong than cong. com

```
CMyPropertySheet::CMyPropertySheet(LPCTSTR pszCaption, CWnd*
    pParentWnd, UINT iSelectPage)
    :CPropertySheet(pszCaption, pParentWnd, iSelectPage)
{
    AddPage (&m_myPage1) ;
    AddPage (&m_myPage2) ;
}
```

cuu duong than cong. com

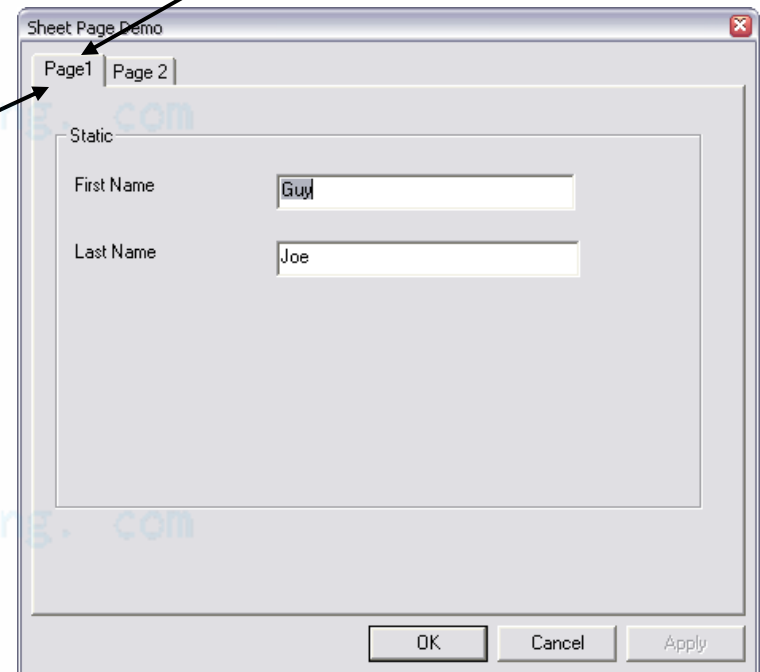
Khởi tạo CMyPropertySheet

```
CMyPropertySheet::CMyPropertySheet(LPCTSTR  
    pszCaption, CWnd* pParentWnd, UINT iSelectPage)  
    :CPropertySheet(pszCaption, pParentWnd,  
        iSelectPage)  
{  
    if (m_bPage1 == TRUE)  
        AddPage(&m_myPage1) ;  
    if (m_bPage2 == TRUE)  
        AddPage(&m_myPage2) ;  
    ....  
}
```

Hiển thị MyPropertySheet

```
void CPropertySheetView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CMyPropertySheet dlg("Sheet Page Demo",this,0);
    dlg.DoModal();
    CView::OnLButtonDown(nFlags, point);
}
```

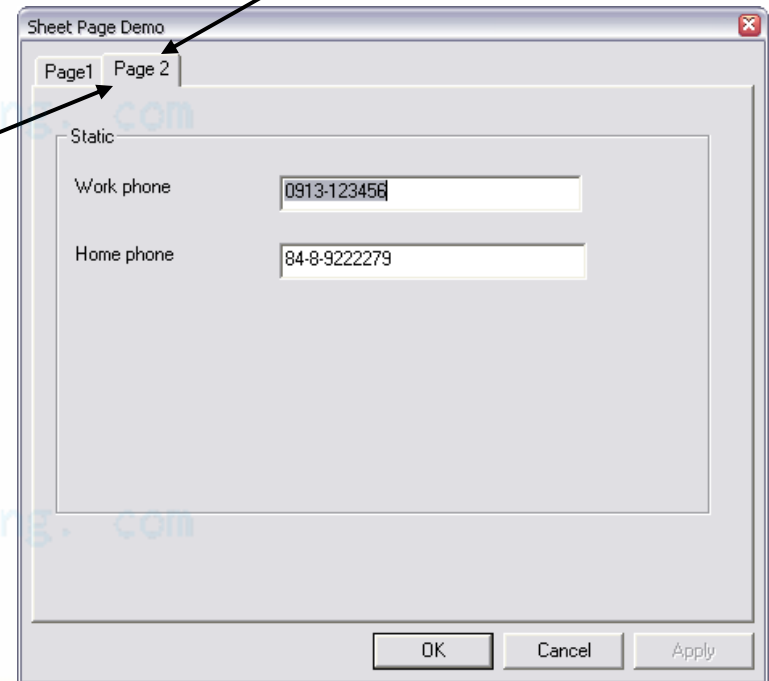
Hiển thị “**page 1**” của
PropertySheet



Hiển thị MyPropertySheet

```
void CPropertySheetView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CMyPropertySheet dlg("Sheet Page Demo",this,1);
    dlg.DoModal();
    CView::OnLButtonDown(nFlags, point);
}
```

Hiển thị “**page 2**” của
PropertySheet



Truyền và nhận dữ liệu với dialog

```
void CMyClass::ShowMyDlg()
```

```
{
```

```
    CMyPropertySheet mySheet("Sheet Page Demo", this, 0);
```

```
    //Truyền giá trị cho các biến trong property  
    sheet
```

```
    int nRes = mySheet.DoModal();
```

```
    if (nRes == IDOK)
```

```
{
```

```
    //Nhận giá trị từ các biến trong property  
    sheet
```

```
}
```

```
}
```


Xử lý sự kiện Apply

```
class CMyPropertySheet : public CPropertySheet
{
    afx_msg void OnApply ();
}
```

```
BEGIN_MESSAGE_MAP(CMyPropertySheet, CPropertySheet)
    //{{AFX_MSG_MAP(CMyPropertySheet)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        ON_BN_CLICKED (ID_APPLY_NOW, OnApply)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
void CMyPropertySheet::OnApply()
{
    GetActivePage ()->UpdateData (TRUE);
    m_myPage1.SetModified (FALSE);
    m_myPage2.SetModified (FALSE);
}
```

Xử lý sự kiện Apply (tt)

```
void CMyPropertySheet::OnApply()  
{  
    GetActivePage ()->UpdateData (TRUE);  
    m_myPage1.UpdateData(TRUE);  
  
    m_myPage2.UpdateData (TRUE);  
  
    m_myPage1.SetModified (FALSE);  
    m_myPage2.SetModified (FALSE);  
}
```

Xử lý sự kiện Apply (tt)

```
void CMyPropertySheet::OnApply()
{
    GetActivePage ()->UpdateData (TRUE);
    if (m_myPage1.m_bInit==TRUE)
    {
        m_myPage1.UpdateData();
    }
    if (m_myPage2.m_bInit==TRUE)
    {
        m_myPage2.UpdateData();
    }
    m_myPage1.SetModified (FALSE);
    m_myPage2.SetModified (FALSE);
}
```

Xử lý sự kiện OK

```
void CMyPage1::OnOk()
{
    GetDataControl();

    CPropertyPage::OnOK();
}
```

```
void CMyPage1::GetDataControl()
{
    UpdateData();
    strFirstName = m_strFirstName;
    .....
}
```

cuu duong than cong. com

```
void CMyPage2::OnOk()
{
    GetDataControl();
    CPropertyPage::OnOK();
}
```

```
void CMyPage2::GetDataControl()
{
    UpdateData();
    strHomePhone = m_strHomePhone;
    .....
}
```

cuu duong than cong. com

Xử lý dữ liệu trong dialog

```
class CMyName : public CObject
{
public:
    CMyName();
    CString m_strFirstName;
    CString m_strLastName;
    virtual ~CMyName();
};
```

```
#include "myName.h"
```

```
class CMyPage1 : public CPropertyPage
```

```
{
```

```
    CMyName m_myName;
```

```
    ....
```

```
}
```

Chuyển và nhận dữ liệu với dialog (tt)

```
void CMyClass::ShowMyDlg()
```

```
{
```

```
    CMyName myName;
```

```
    CMyPropertySheet mySheet("Sheet Page Demo",this,0);
```

```
    //Truyền giá trị cho các biến trong property sheet
```

```
    mySheet.m_myPage1.m_myName = myName;
```

```
    int nRes = mySheet.DoModal();
```

```
    if (nRes == IDOK)
```

```
    {
```

```
        //Nhận giá trị từ các biến trong property sheet
```

```
        myName = mySheet.m_myPage1.m_myName;
```

```
    }
```

```
}
```

Xử lý dữ liệu trong dialog (tt)

```
class CMyName : public CObject
{
public:
    CMyName();
    CString m_strFirstName;
    CString m_strLastName;
    virtual ~CMyName();
```

```
    void operator =(CMyName &tmp);
```

```
};
```

```
void CMyName::operator =(CMyName &tmp)
```

```
{
```

```
    m_strFirstName = tmp.m_strFirstName;
```

```
    m_strLastName = tmp.m_strLastName;
```

```
}
```

Xử lý dữ liệu trong dialog (tt)

```
BOOL CMyPage1::OnInitDialog()
{
    CPropertyPage::OnInitDialog();
    SetDataControl();
    return TRUE;
}

void CMyPage1::SetDataControl()
{
    m_strFirstName = m_myName.m_strFirstName;
    .....
    UpdateData(0);

}
```


Truyền và nhận dữ liệu với dialog (tt)

```
void CMyClass::ShowMyDlg()
```

```
{
```

```
    CMyName myName;
```

```
    CMyPropertySheet mySheet("Sheet Page Demo",this,0);
```

```
    //Truyền giá trị cho các biến trong property sheet
```

```
    mySheet.m_myPage1.SetData(myName);
```

```
    int nRes = mySheet.DoModal();
```

```
    if (nRes == IDOK)
```

```
    {
```

```
        //Nhận giá trị từ các biến trong property sheet
```

```
        mySheet.m_myPage1.GetData(myName);
```

```
    }
```

```
}
```

Xử lý dữ liệu trong dialog (tt)

```
void CMyPage1::SetData(CMyName myName)  
{  
    m_myName = myName;  
}
```

```
void CMyPage1::GetData(CMyName &myName)  
{  
    myName = m_myName;  
}
```

cuu duong than cong. com

Xử lý sự kiện OK

```
void CMyPage1::OnOk()
{
    GetDataControl();

    CPropertyPage::OnOK();
}
```

cuu duong than cong. com

```
void CMyPage1::GetDataControl()
{
    UpdateData();
    m_myName.m_strFirstName = m_strFirstName;
    .....
}
```

cuu duong than cong. com

Modeless Dialog Box

```
class CModelessDlg : public CDialog
{
// Construction
public:
    CModelessDlg(CWnd* pParent = NULL); // standard constructor
    CWnd *m_pParent;
    void DoModeless();
    enum { IDD = IDD_DIALOG1 };
protected:
    virtual void DoDataExchange(CDataExchange* pDX);
protected:
   //{{AFX_MSG(CModelessDlg)
    virtual void OnOK();
    virtual void OnCancel();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
```

Modeless Dialog Box (tt)

```
void CModelessDlg::DoModeless()
{
    if (GetSafeHwnd()==NULL)
    {
        Create(IDD_DIALOG1,m_pParent);
        ShowWindow(SW_SHOW);
        CenterWindow();
    }
    else
    {
        if (IsWindowVisible()==FALSE)
        {
            ShowWindow(SW_SHOW);
        }
    }
}
```

Modeless Dialog Box – SendMessage

```
void CModelessDlg::OnApply()
{
    // TODO: Add extra validation here
    UpdateData (TRUE);
    CMyClass my;

    my.s1 = m_S;
    my.n = 5;

    AfxGetMainWnd ()->SendMessage (WM_USER_APPLY, 0, (LPARAM) &my);
    //AfxGetMainWnd ()->SendMessage (WM_USER_APPLY, 0, 0L);
}
```

Modeless Dialog Box – SendMessage (tt)

```
class CMainFrame : public CFrameWnd
```

```
{  
    afx_msg LRESULT OnApply (WPARAM wParam, LPARAM lParam);  
};
```

```
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
```

```
//{{AFX_MSG_MAP(CMainFrame)
```

```
ON_MESSAGE (WM_USER_APPLY, OnApply)
```

```
//}}AFX_MSG_MAP
```

```
END_MESSAGE_MAP()
```

```
LRESULT CMainFrame::OnApply (WPARAM wParam, LPARAM lParam)
```

```
{  
    CMyClass* my =(CMyClass*)lParam ;
```

```
    .....
```

```
    Invalidate();
```

```
    return 0;
```

```
}
```

Modeless Dialog Box – SendMessage (tt)

```
if _MSC_VER > 1000
```

```
#pragma once
```

```
#endif // _MSC_VER > 1000
```

```
#define WM_USER_APPLY WM_USER+0x100
```

```
#define VC_EXTRALEAN
```

```
#include <afxwin.h>           // MFC core and standard components
```

```
#include <afxext.h>           // MFC extensions
```

```
#include <afxdisp.h>          // MFC Automation classes
```

```
#include <afxdtctl.h>          // MFC support for Internet Explorer 4
```

```
    Common Controls
```

```
#ifndef _AFX_NO_AFXCMN_SUPPORT
```

```
#include <afxcmn.h>           // MFC support for Windows
```

```
    Common Controls
```

```
#endif // _AFX_NO_AFXCMN_SUPPORT
```


Modeless Dialog Box – Show & Hide

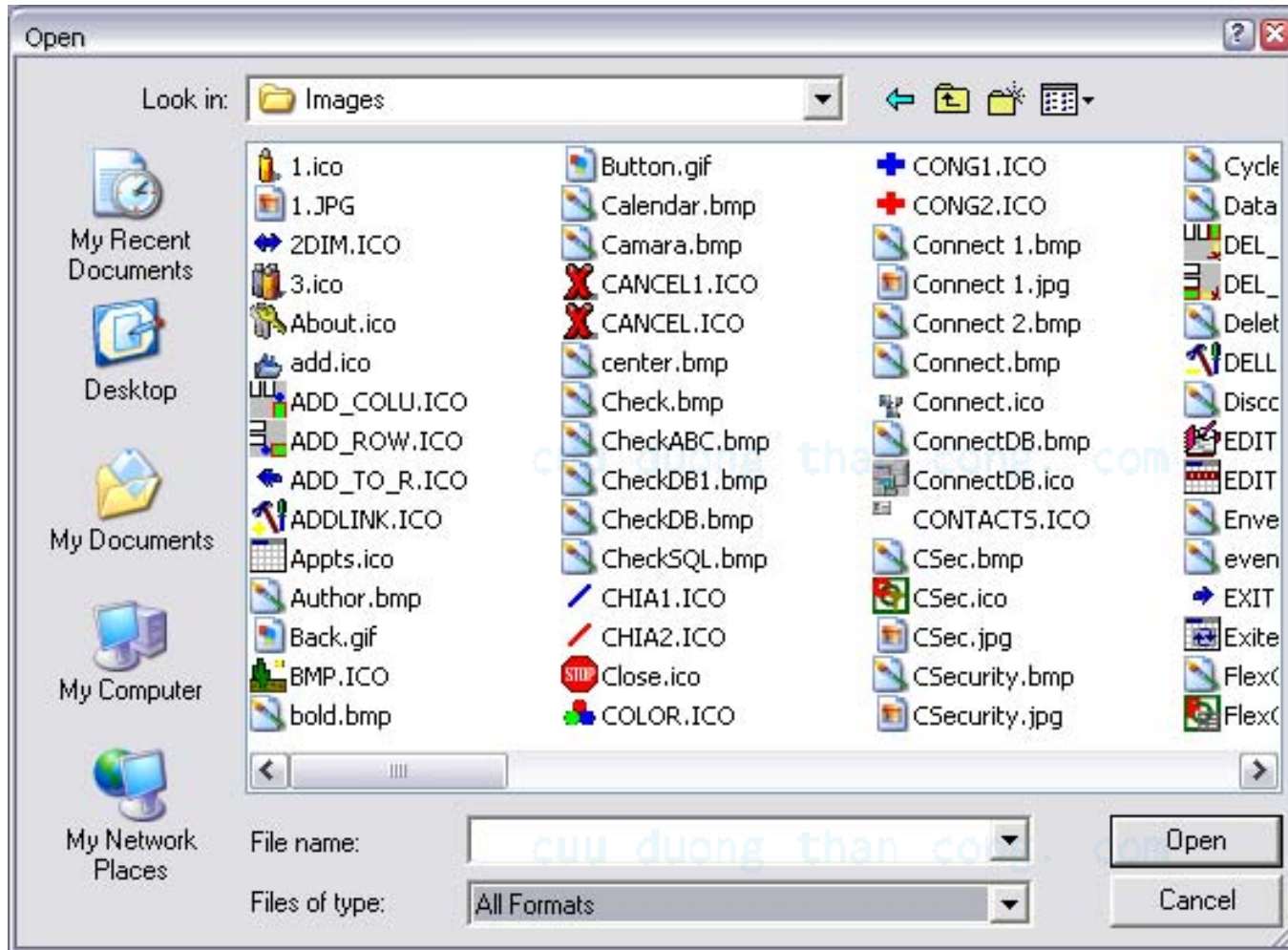
```
void CMyModelessView::OnShow()
{
    // TODO: Add your command handler code here
    m_dlgMyModeless.DoModeless();
}

void CMyModelessView::OnHide()
{
    // TODO: Add your command handler code here
    m_dlgMyModeless.ShowWindow(SW_HIDE);
}
```

Dialog Box Common Controls Property Sheet

Lập trình với Common Dialog

Giới thiệu về dialog – Common dialog



Open file
dialog

Common dialog – Open File

```
void CCommonDialogView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CFileDialog fDlg(1,NULL,NULL,OFN_HIDEREADONLY | OFN_FILEMUSTEXIST ,
        "C Header Files (*.h)|*.h|C++ Header Files (*.hpp)|*.hpp|");
    if ( fDlg.DoModal() == IDOK)
    {
        CString s= fDlg.GetPathName();
        if (s.Find(".h")==-1 && s.Find(".hpp")==-1)
            s+="*.hpp";
        //Xử lý với file s
        s.ReleaseBuffer(s.GetLength());
    }
    CView::OnLButtonDown(nFlags, point);
}
```

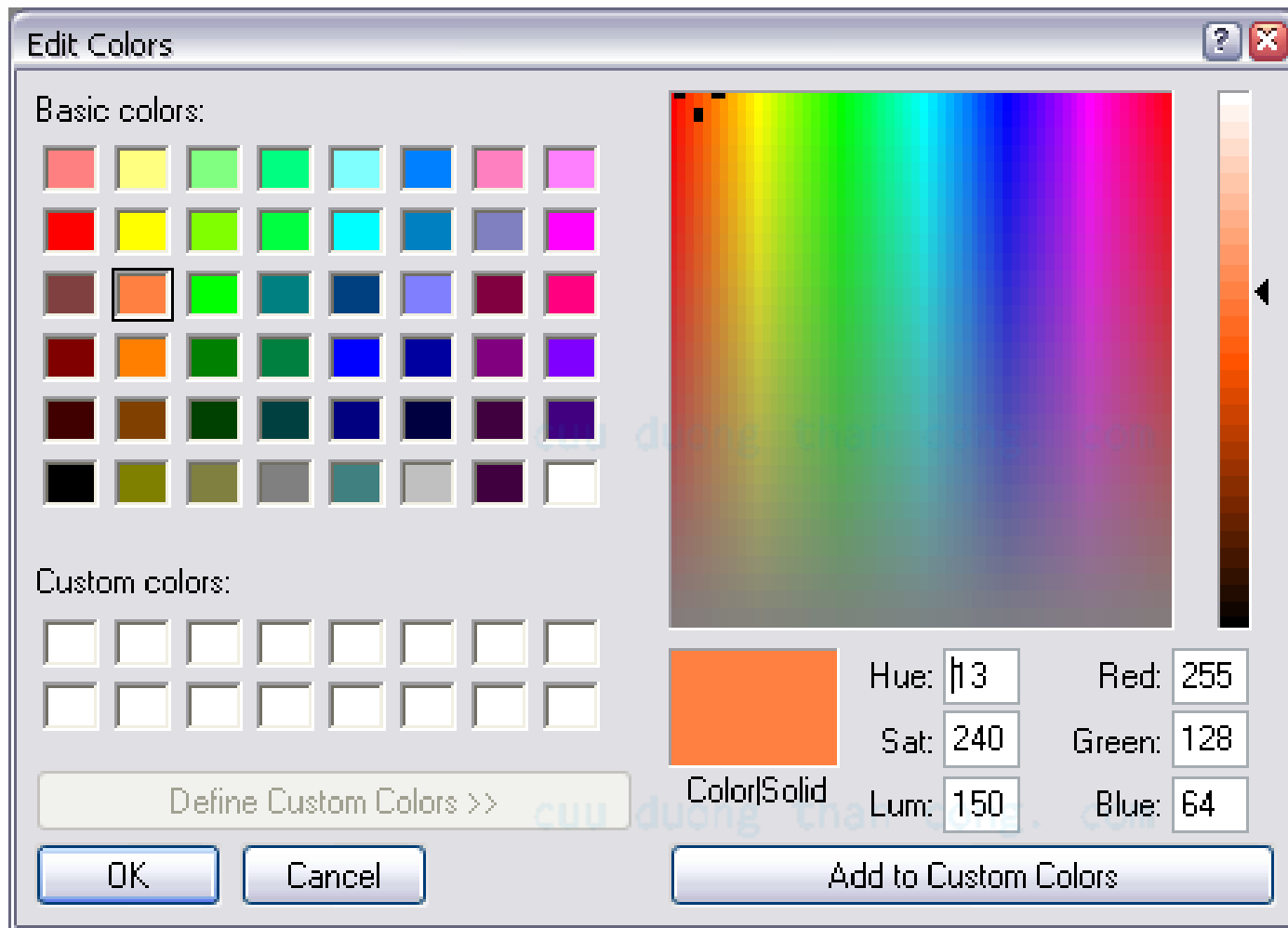
Common dialog – Save File

```
void CCommonDialogView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CFileDialog fDlg(0,NULL,NULL,OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT |
    OFN_PATHMUSTEXIST ,
        "C Header Files (*.h)|*.h|C++ Header Files (*.hpp)|*.hpp|");
    if ( fDlg.DoModal() == IDOK)
    {
        CString s= fDlg.GetPathName();
        if (s.Find(".h") == -1 && s.Find(".hpp") == -1)
            s += ".hpp";
        //Xử lý với tên file s

        s.ReleaseBuffer(s.GetLength());
    }
    CView::OnLButtonDown(nFlags, point);
}
```

Giới thiệu về dialog – Common dialog



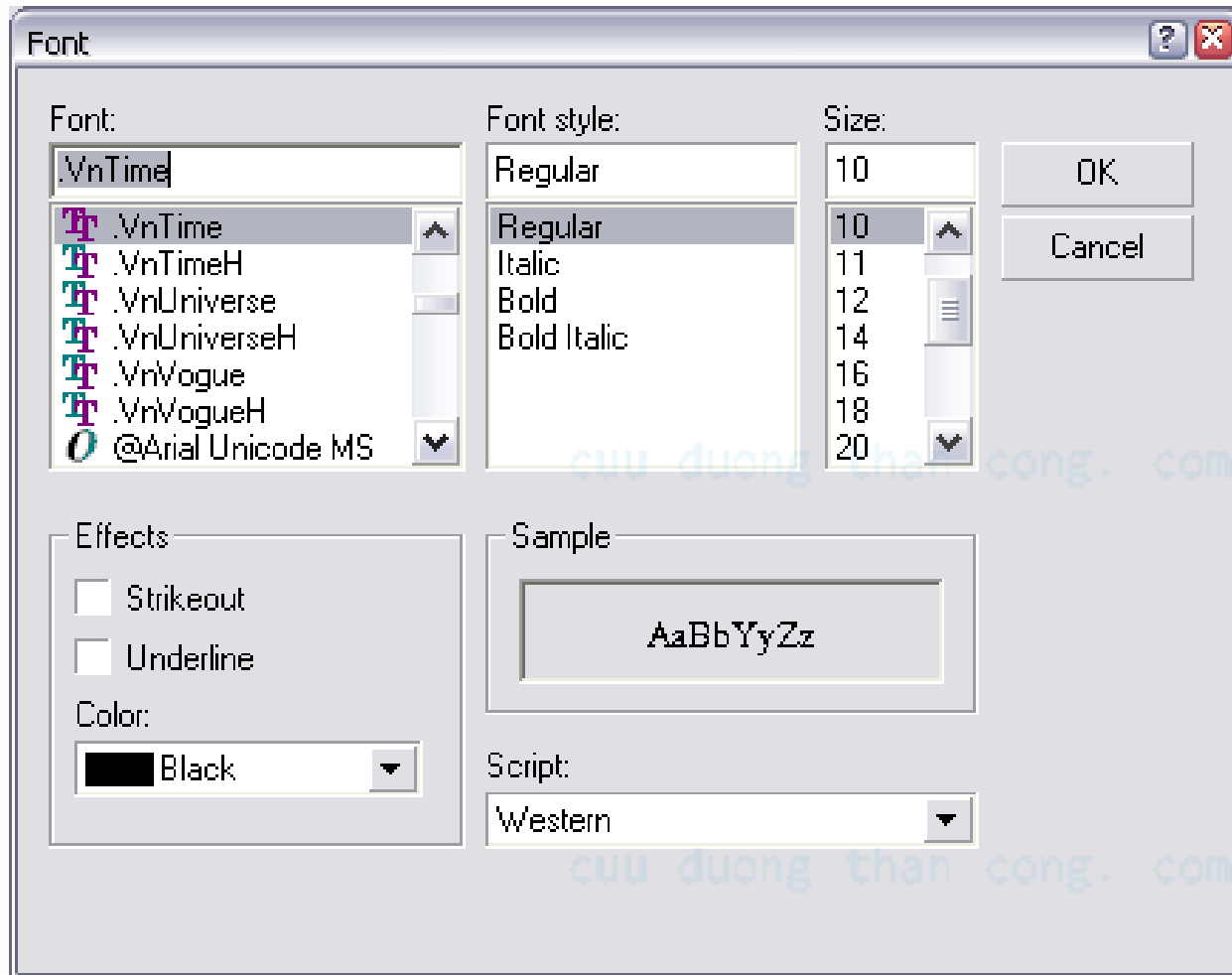
Color dialog

Common dialog – Color Dialog

```
void CColorDialogDlg::OnColor()
{
    // TODO: Add your control notification handler code here
    CColorDialog dlg;

    if (dlg.DoModal() == IDOK)
    {
        COLORREF color = dlg.GetColor();
        //Xử lý với color
        ...
    }
}
```

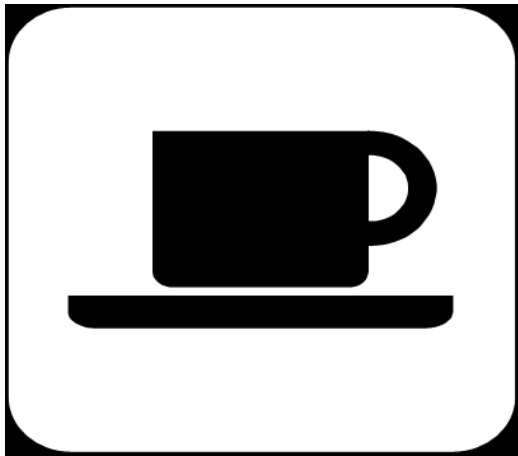
Giới thiệu về dialog – Common dialog



Fonts dialog



cuu duong nhan cong. com



cuu duong nhan cong. com



Toolbar - Menubar DialogBar

- ❖ **Tạo và xử lý Toolbar**
- ❖ **Tạo và xử lý Menubar**
- ❖ **Tạo và xử lý Dialogbar**

Trần Ngọc Bảo

Email: tnbao.dhsp@gmail.com

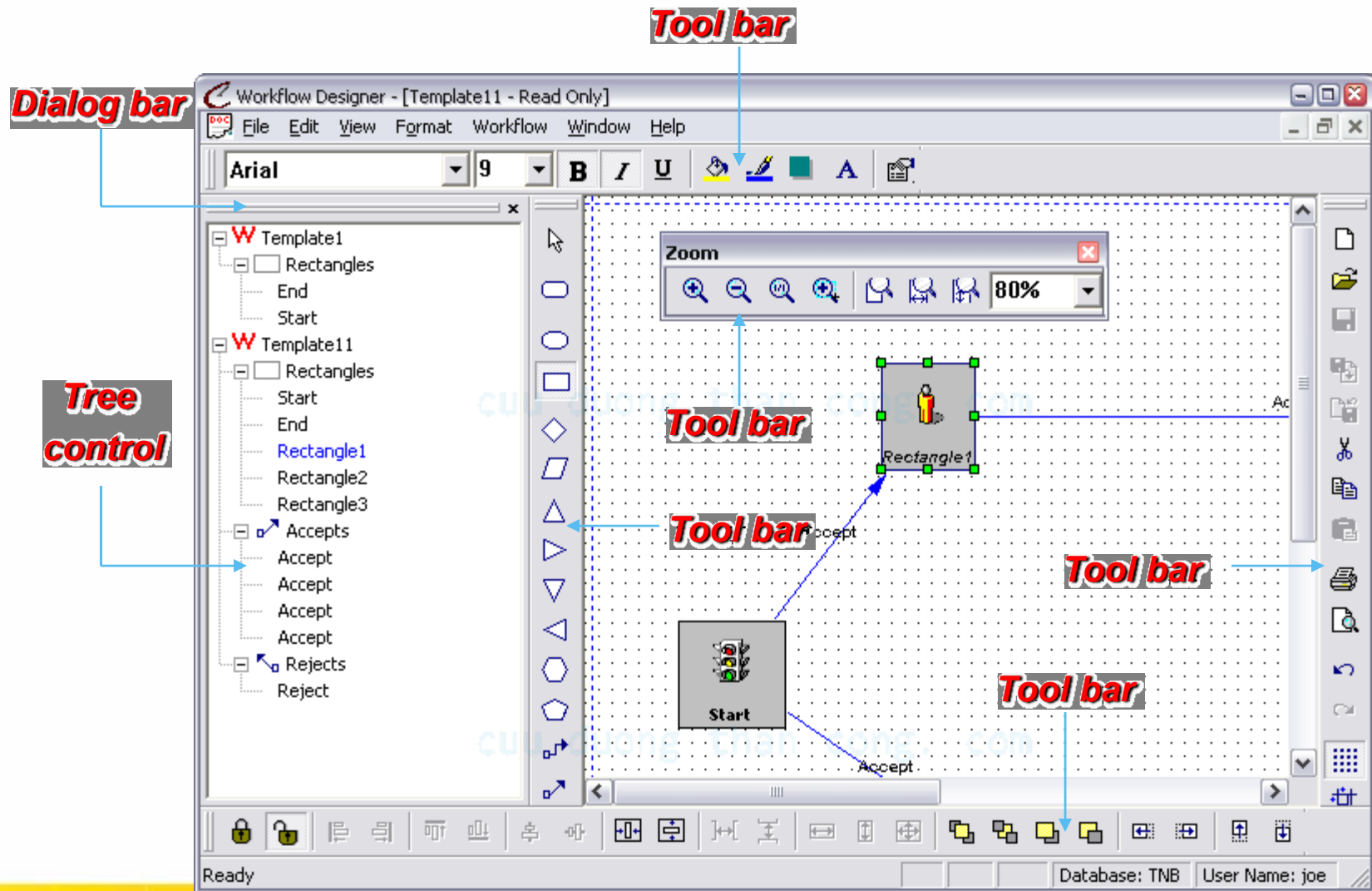
Toolbar - Menubar DialogBar

Lập trình với Toolbar

NỘI DUNG

- ❖ Giới thiệu về Tool bar và Dialog bar
- ❖ Tạo và xử lý tool bar
- ❖ Radio & Check Box buttons trên tool bar
- ❖ Thêm combo box vào Tool bar
- ❖ Thêm Flyby và ToolTip
- ❖ Dialog bar
- ❖ Resizable dialog bar

Tool bar – Dialog bar



Tool bar – Dialog bar

Tool bar (Docked to Top)

**Dialog bar
(docked to Left)**

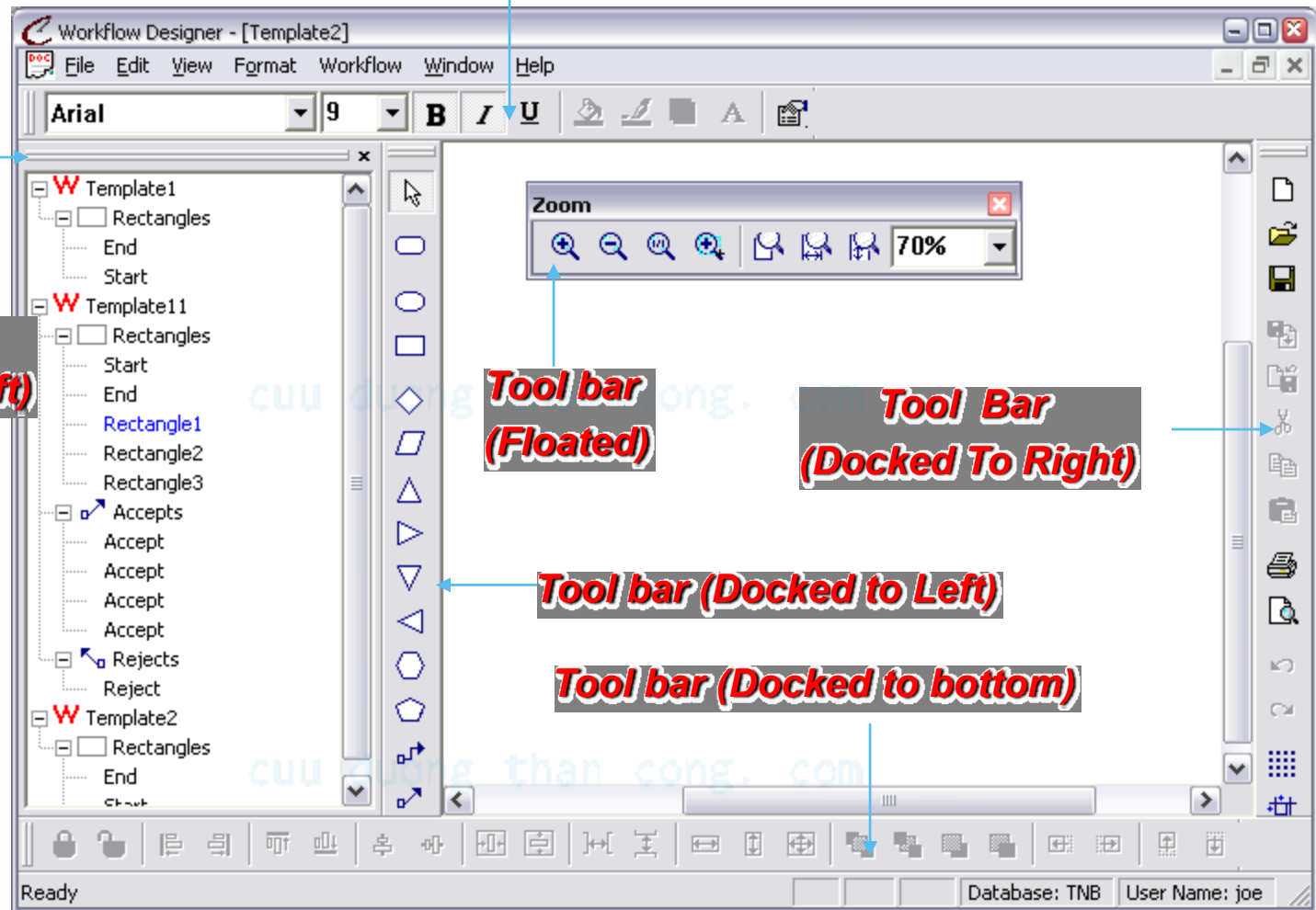
Tree control

**Tool bar
(Floated)**

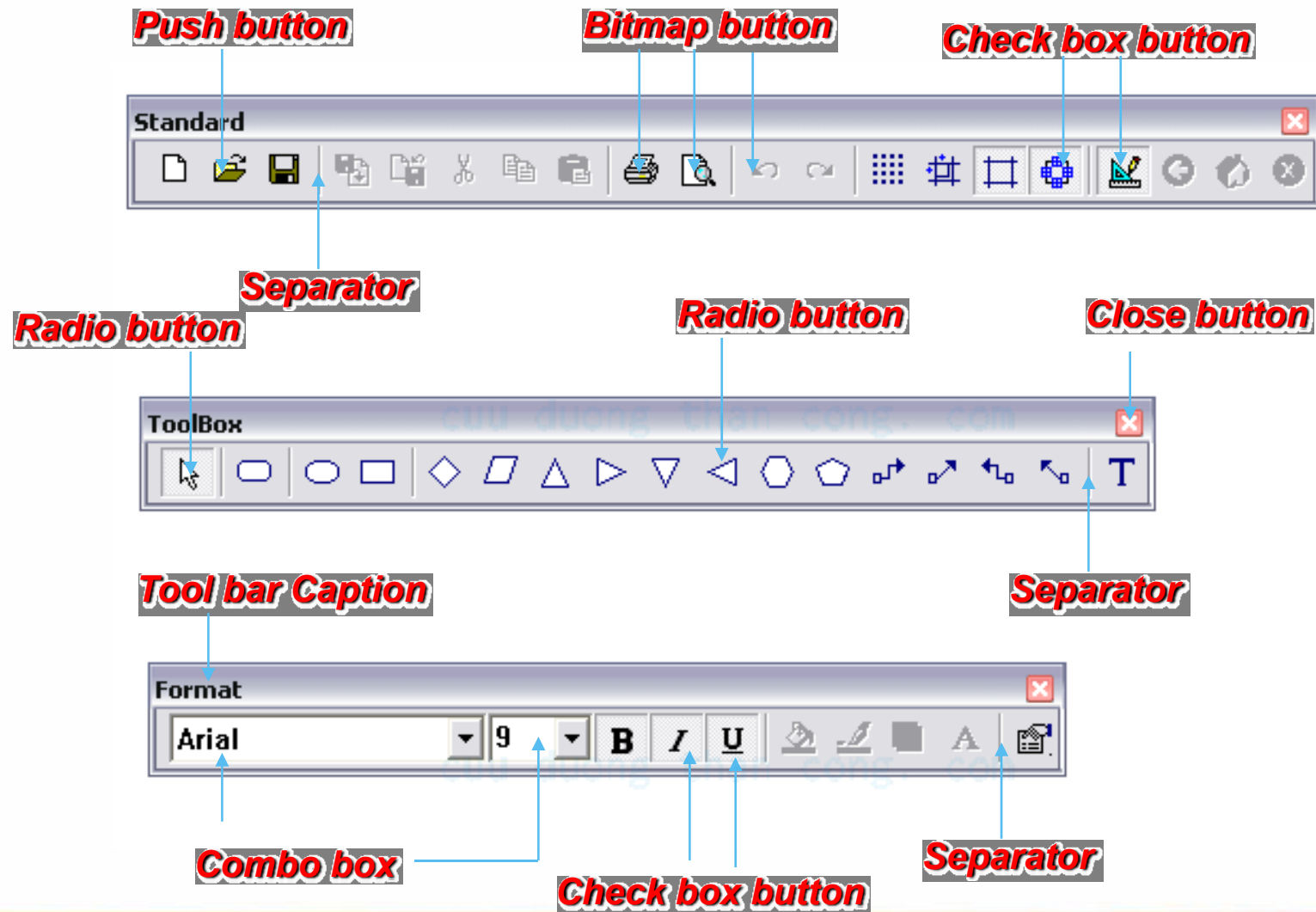
**Tool Bar
(Docked To Right)**

Tool bar (Docked to Left)

Tool bar (Docked to bottom)



Các thành phần trên Tool bar

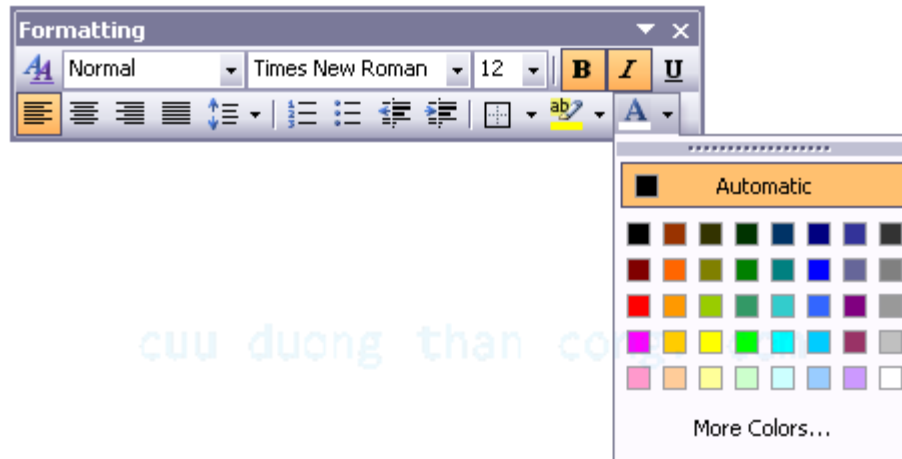


Tạo và xử lý Tool bar

- ❖ Tạo một ứng dụng có Tool bar mặc định bằng AppWizard



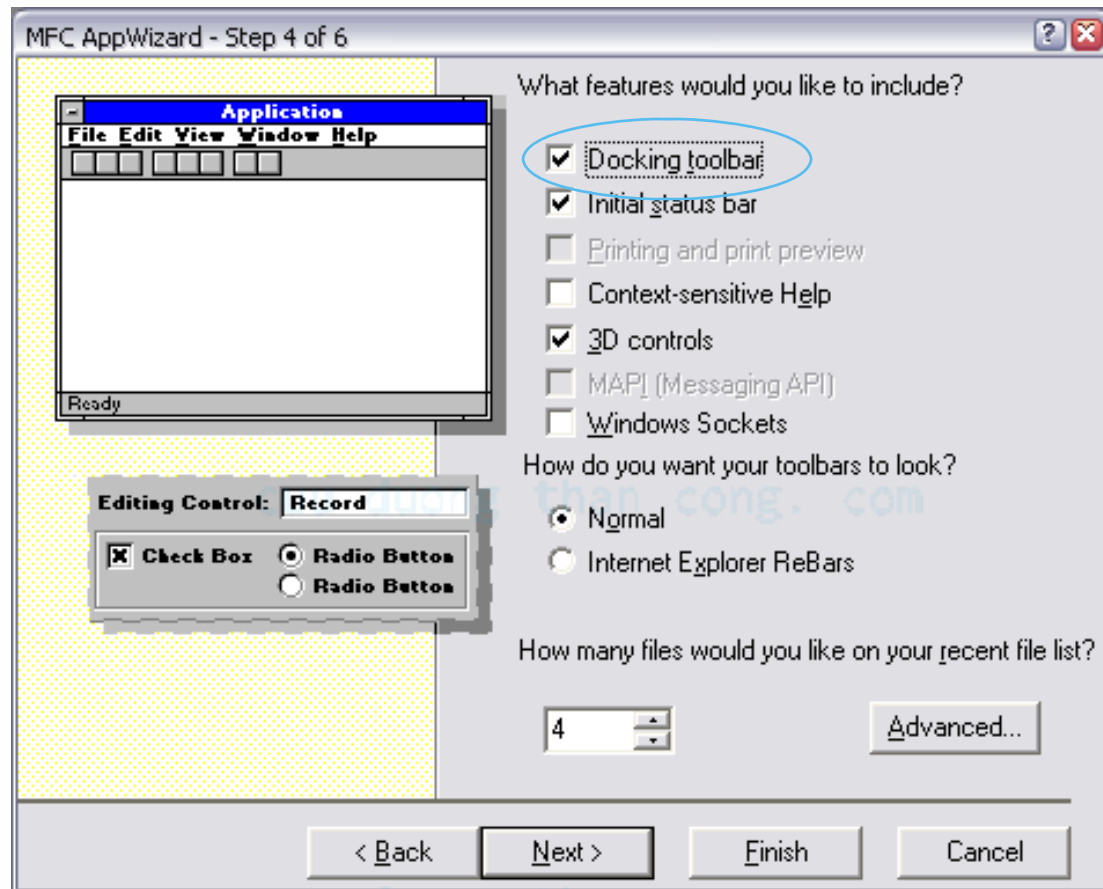
- ❖ Tạo và xử lý Tool bar bằng lớp CToolBar



Tool bar - Tạo ứng dụng bằng AppWizard

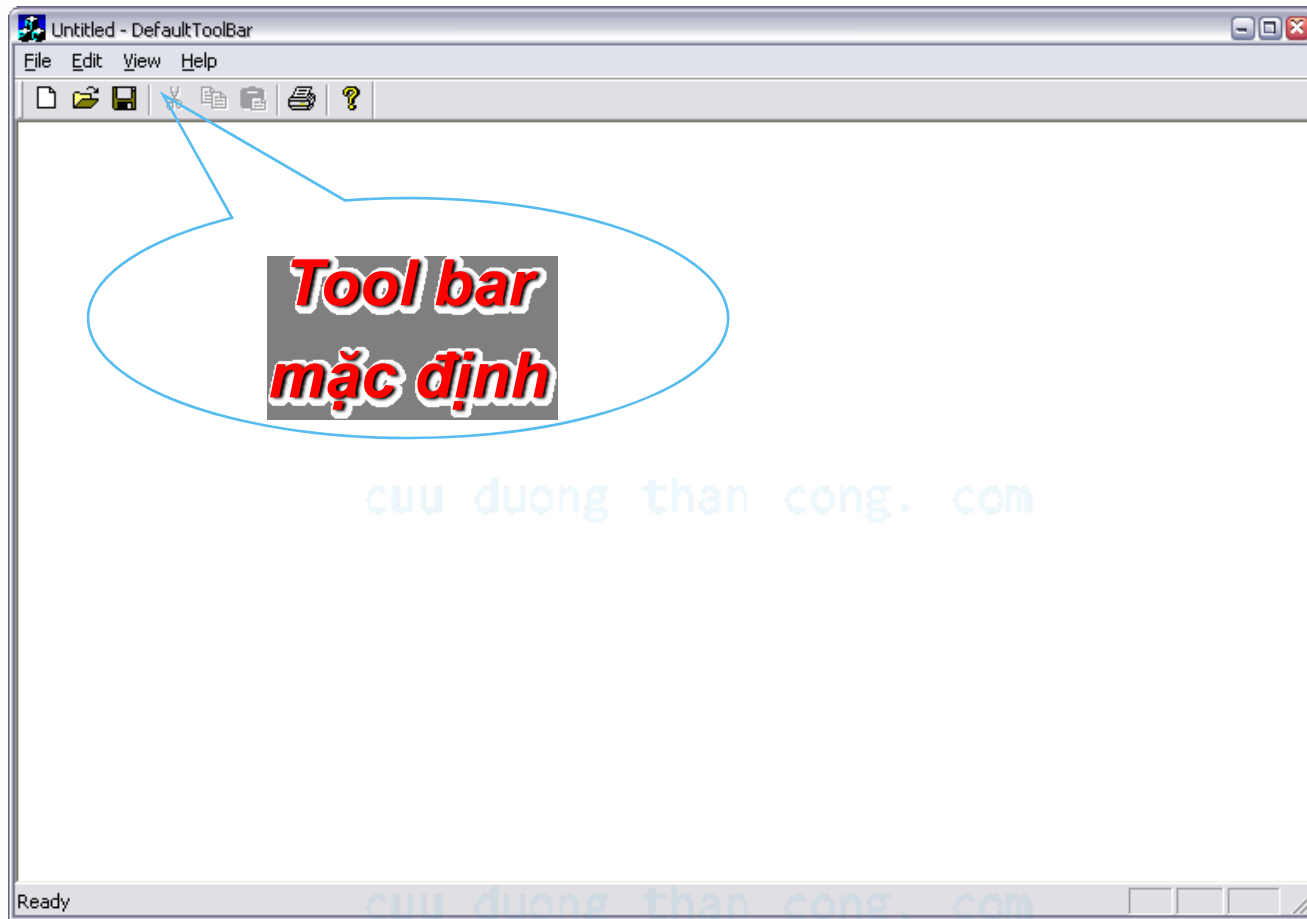
- ❖ Chọn menu **File → New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project **"MFC AppWizard (exe)"**
- ❖ Đặt tên project trong ô **"Project name"**
- ❖ Xác định đường dẫn thư mục trong **"Location"**
- ❖ Step 1: Chọn loại ứng dụng **"Single Document"**, bỏ option **"Document/View architecture support"**
- ❖ Nhấn **"Next"** đến **"Step 4 of 6"**

Tool bar - Tạo ứng dụng bằng AppWizard



Chọn option **"Docking toolbar"** để AppWizard tự động tạo ra tool bar mặc định

Tool bar - Tạo ứng dụng bằng AppWizard



Kết quả chạy chương trình tạo tool bar bằng AppWizard

Tool bar - Các xử lý trong lớp CMainFrame

❖ Khai báo biến CToolBar

```
class CMainFrame : public CFrameWnd
{
....
protected: //khai báo biến xử lý default tool bar
    CToolBar m_wndToolBar;
....
}
```

❖ Các xử lý trong hàm OnCreate

- **Bước 1: Gọi hàm tạo lập cửa sổ tool bar CToolBar::Create(...)**
- **Bước 2: Load resource tool bar CToolBar::LoadToolBar(...)**
- **Bước 3: Thiết lập thuộc tính tool bar CToolBar::SetBarStyle(...)**
- **Bước 4: Gọi hàm CToolBar::EnableDocking(...)**
- **Bước 5: Gọi hàm CToolBar::EnableDocking(...) để gắn tool bar**

Tool bar - Các xử lý trong hàm OnCreate

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    .....
    if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE |
        CBRS_TOP | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY |
        CBRS_SIZE_DYNAMIC) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);
    .....
}
```

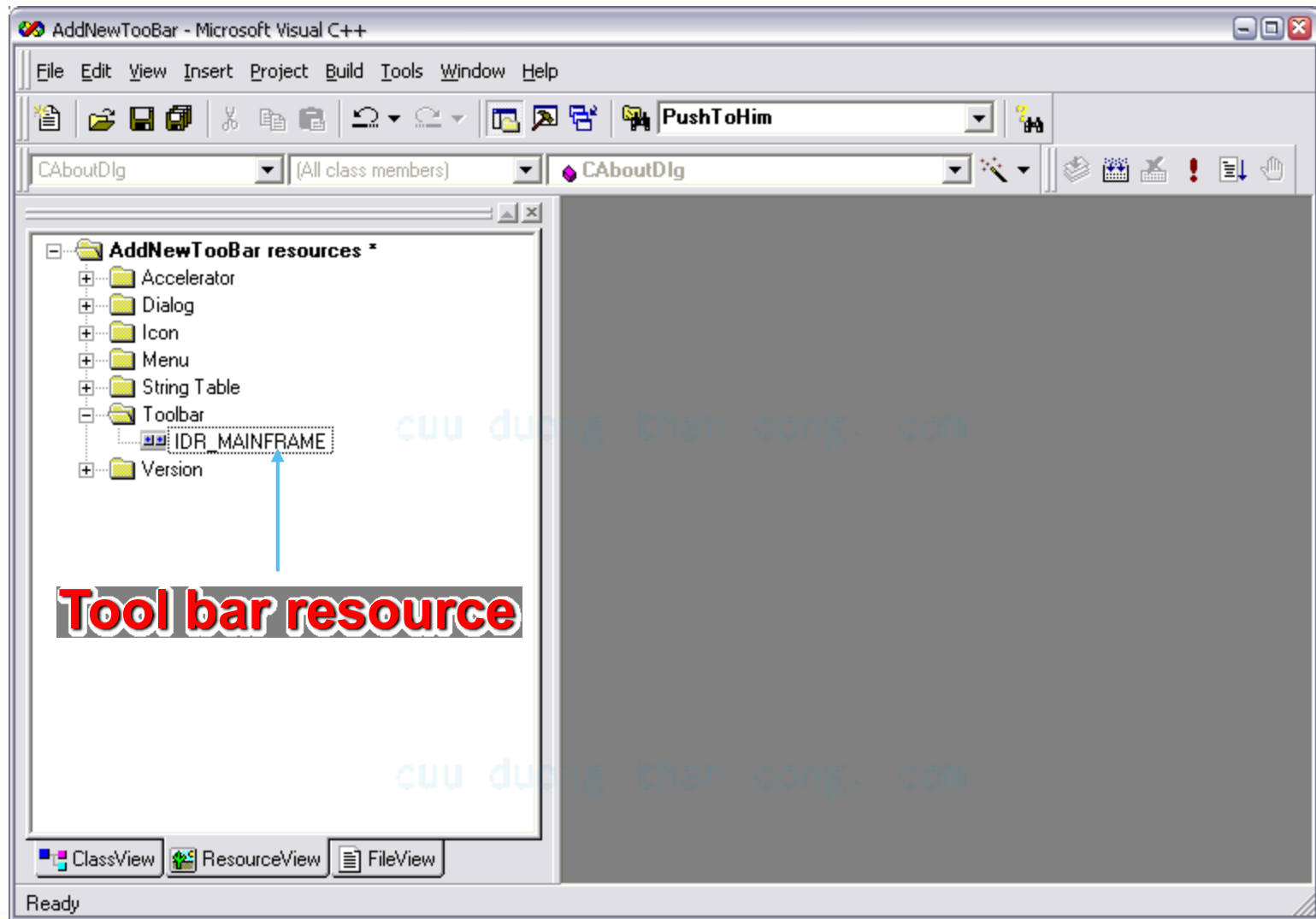
Thêm Tool bar vào ứng dụng

- ❖ Thêm mới Tool bar resource
- ❖ Khai báo biến CToolBar
- ❖ Tạo tool bar
- ❖ Thiết lập thuộc tính tool bar
- ❖ Gắn tool bar
- ❖ Xử lý message

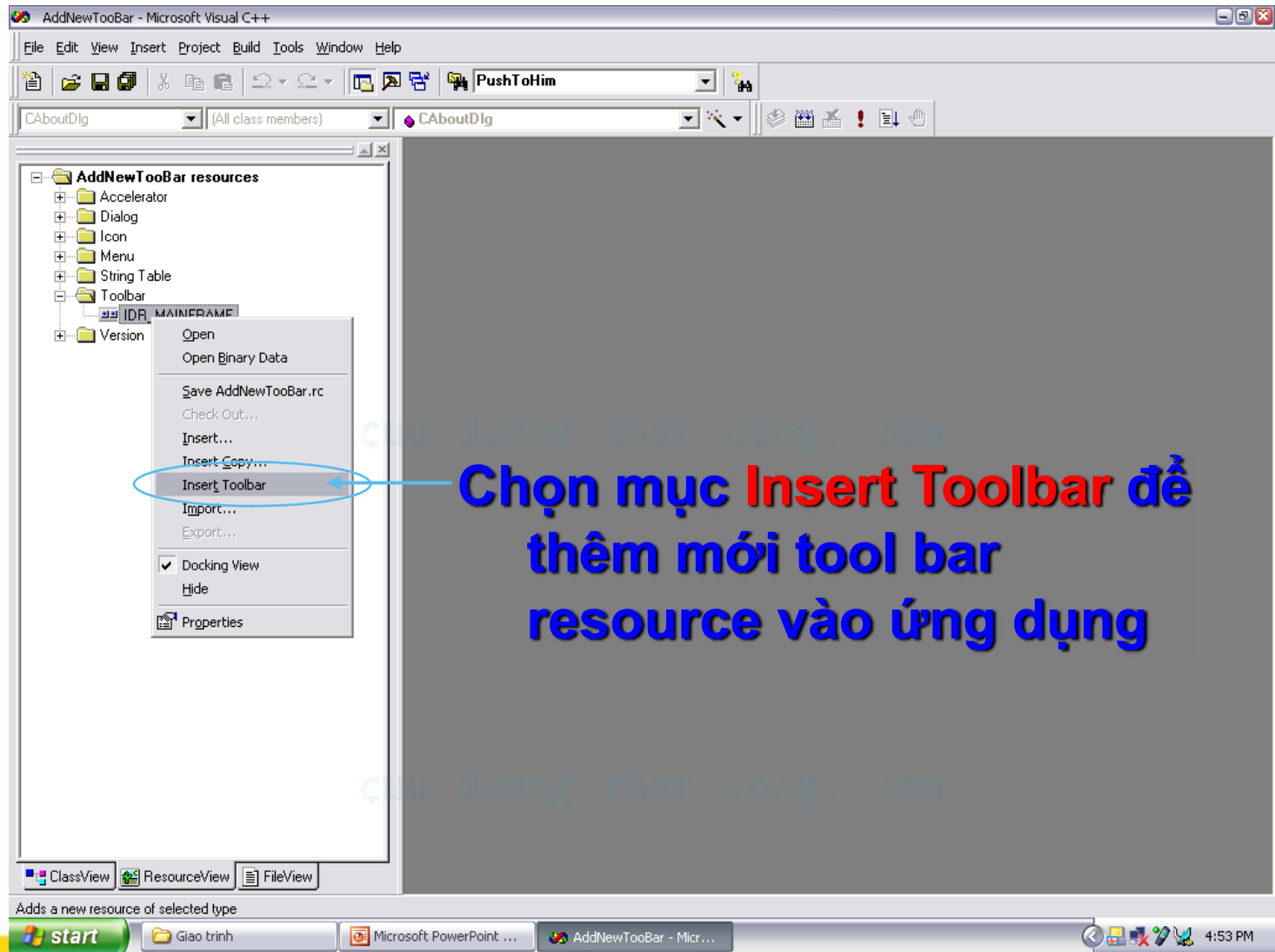
cuu duong than cong. com

cuu duong than cong. com

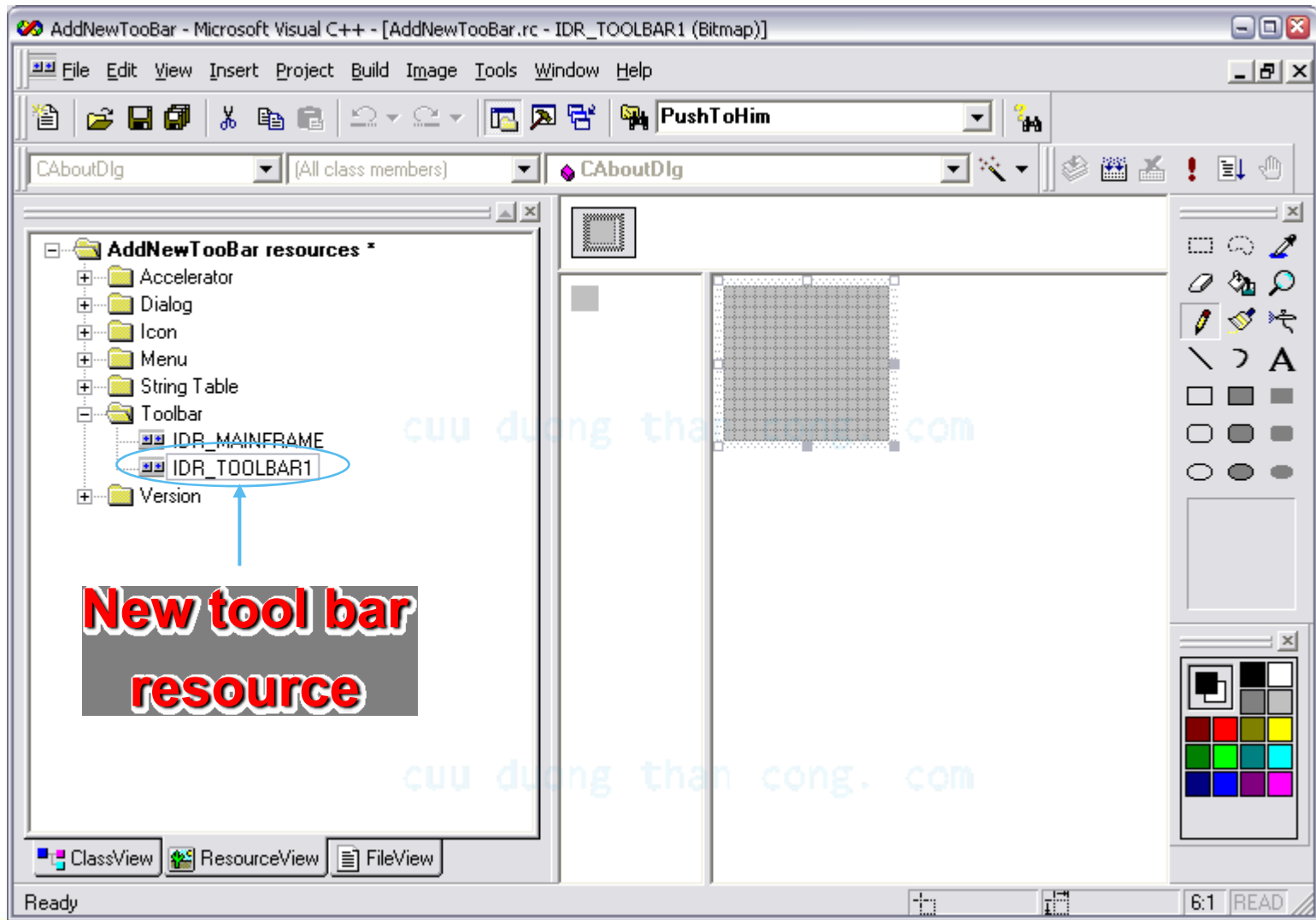
Thêm mới tool bar resource



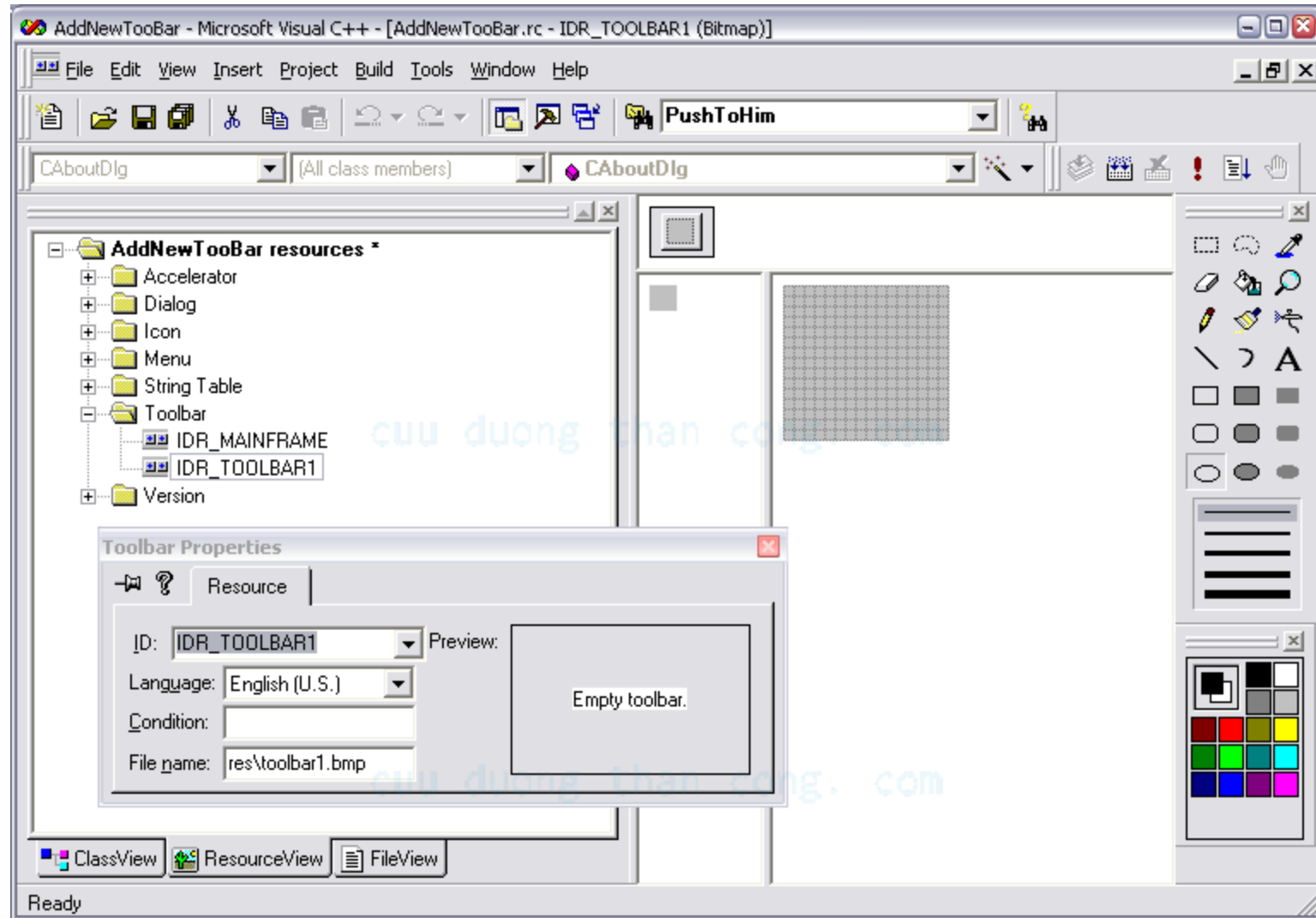
Thêm mới tool bar resource (tt)



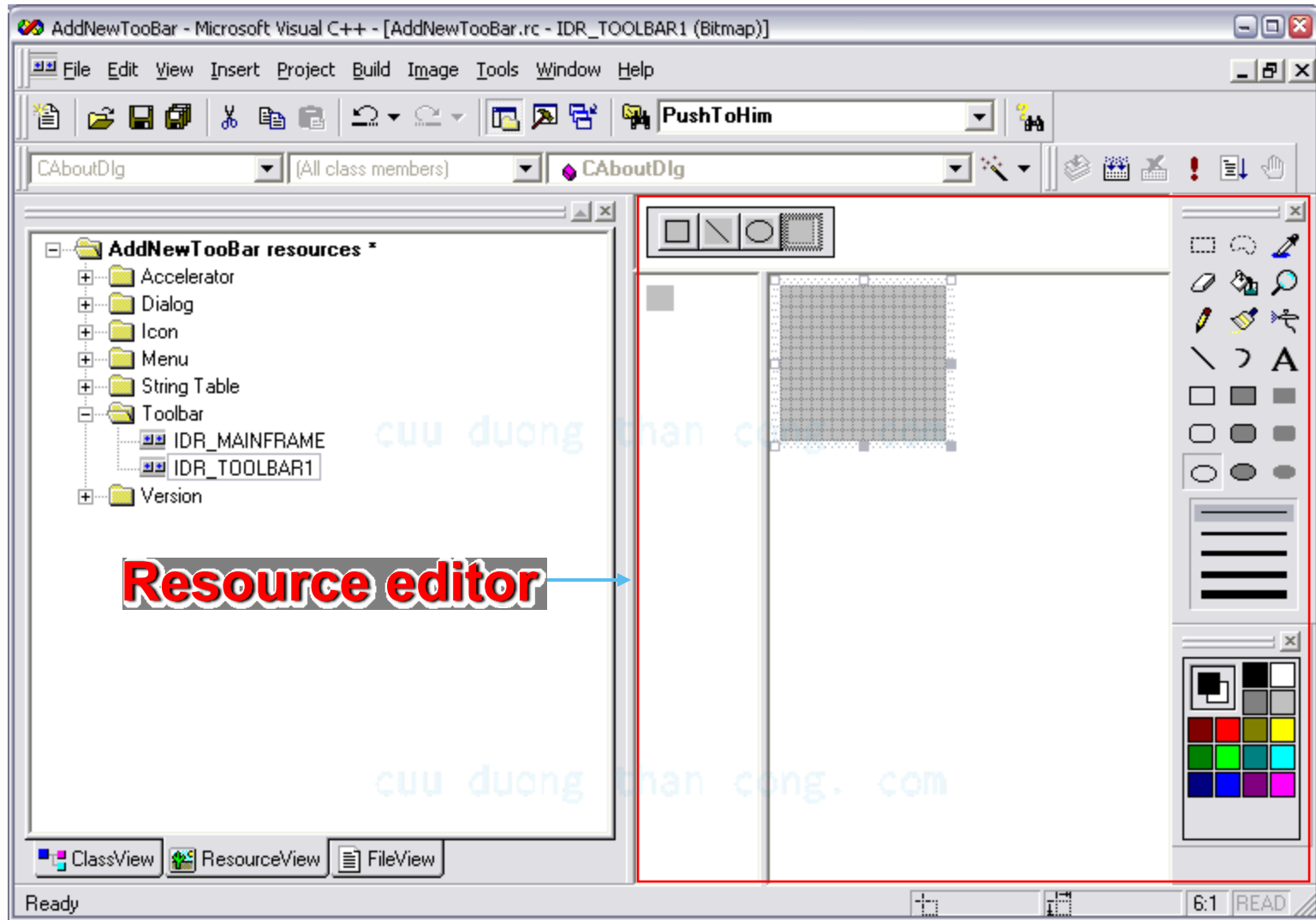
Thêm mới tool bar resource (tt)



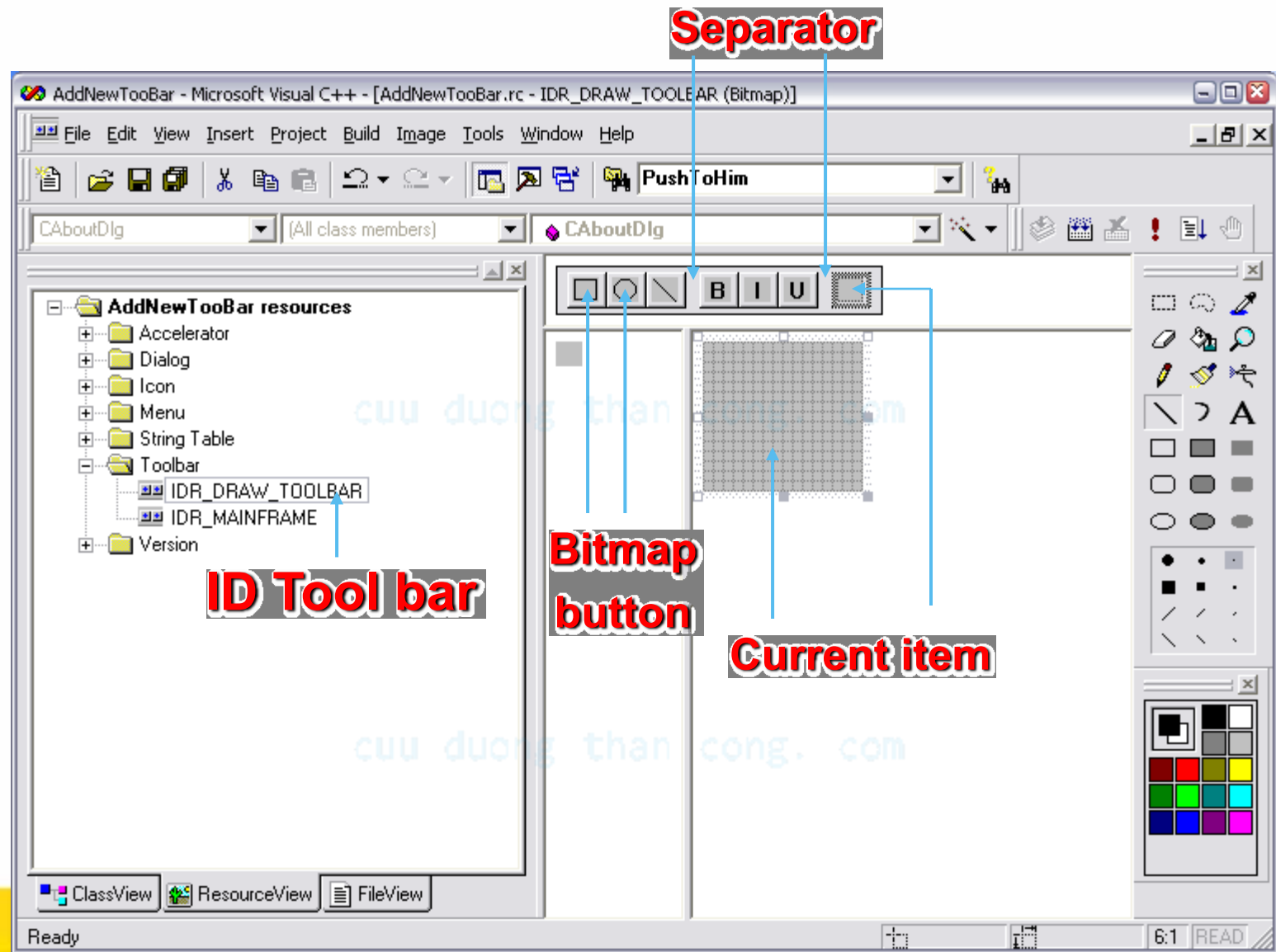
Thêm mới tool bar resource (tt)



Thêm mới tool bar resource (tt)



Thêm mới tool bar resource (tt)



Khai báo biến quản lý tool bar

```
class CMainFrame : public CFrameWnd
{
protected: // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar   m_wndToolBar;
    CToolBar   m_wndDrawBox;
};
```

cuu duong than cong. com

Tạo & xác lập thuộc tính cho Toolbar

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

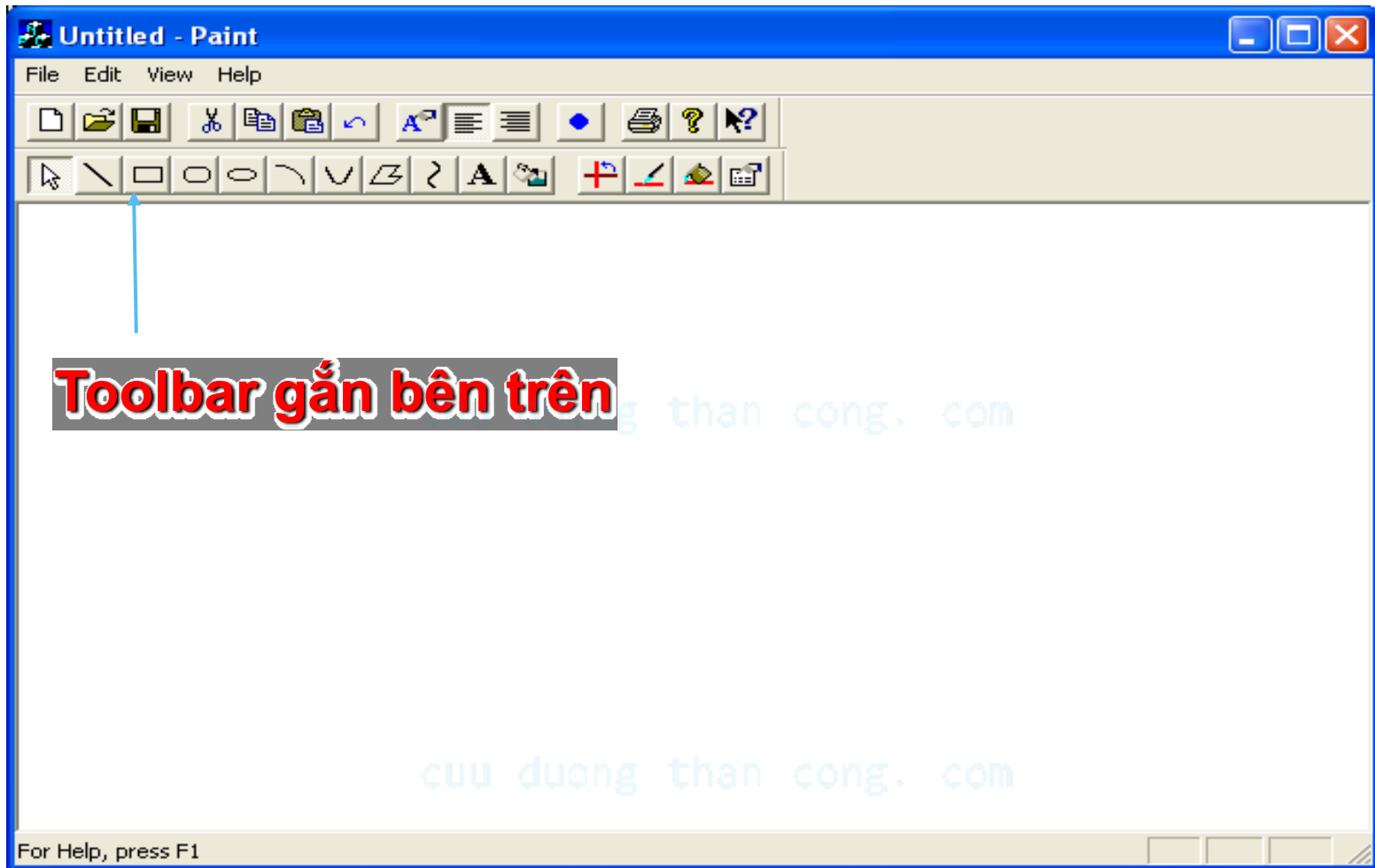
    // Tạo Toolbar
    if (!m_wndDrawBox.CreateEx(this) ||
        !m_wndDrawBox.LoadToolBar(IDR_DRAW_TOOLBAR))
    {
        return -1;
    }

    // Xác lập thuộc tính Toolbar
    m_wndDrawBox.SetBarStyle(m_wndDrawBox.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);
    ...
}
```

Gắn Toolbar vào cửa sổ chính

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ....
    // Tạo Toolbar
    // Xác lập thuộc tính Toolbar
    // Gắn Toolbar vào cửa sổ chính
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);
    ...
}
```

Gắn Toolbar vào bên trên cửa sổ chính



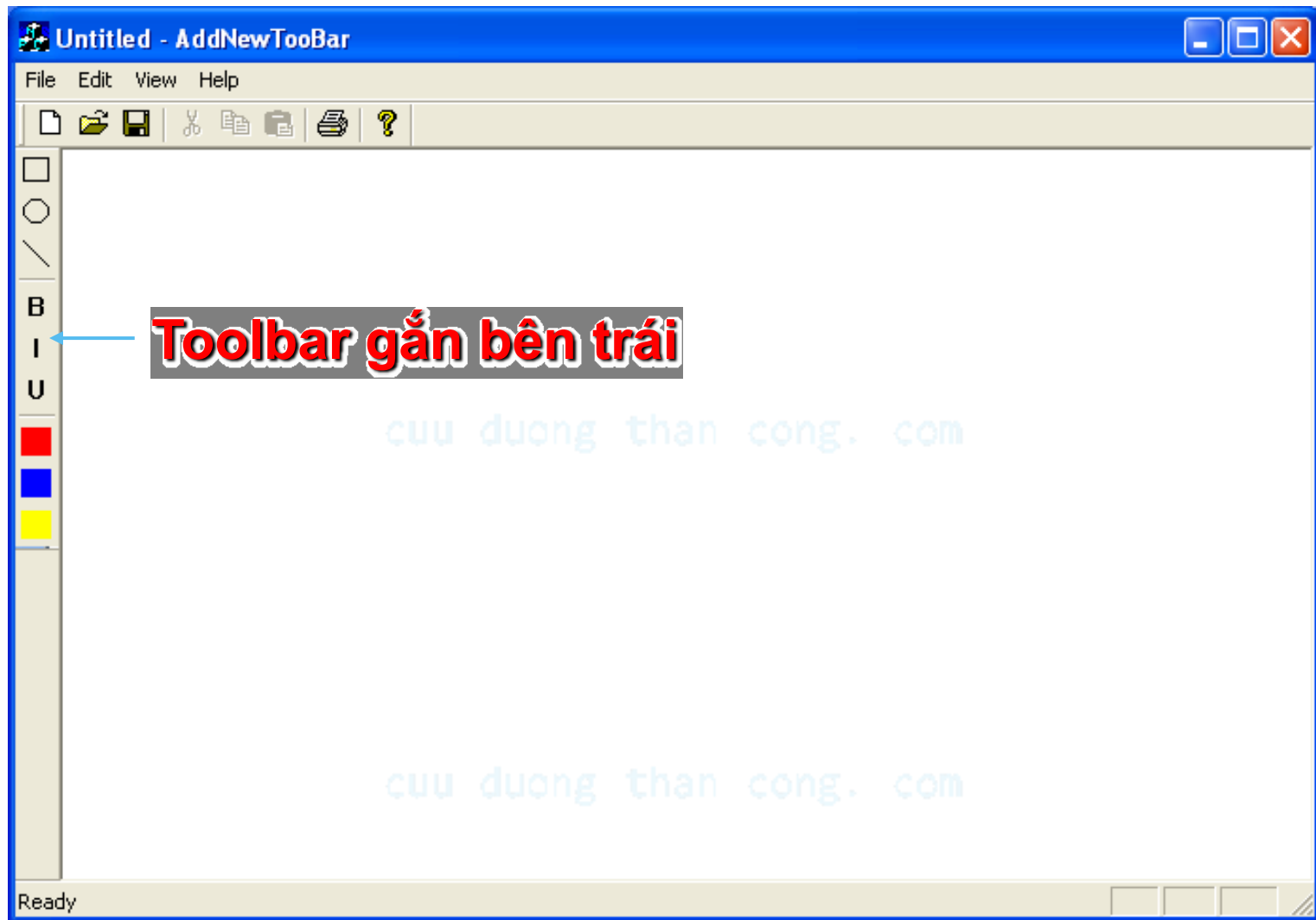
Toolbar gắn bên trên

cuu duong than cong. com

Gắn Toolbar vào bên trên cửa sổ chính

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ....
    // Tạo Toolbar
    // Xác lập thuộc tính Toolbar
    // Gắn Toolbar vào cửa sổ chính
    m_wndDrawBox.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_TOP);
    DockControlBar(&m_wndDrawBox,AFX_IDW_DOCKBAR_TOP);
    ...
}
```

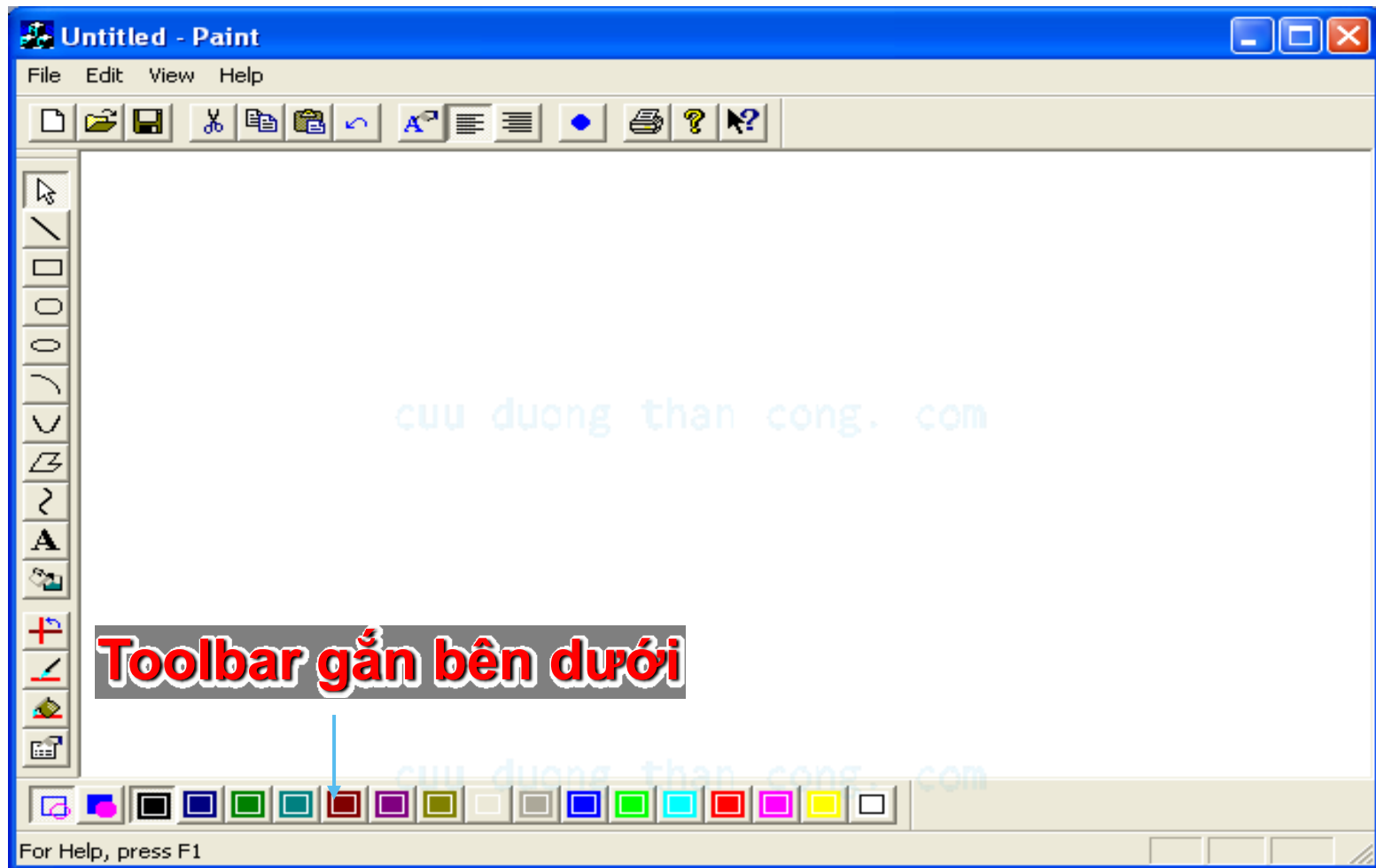
Gắn Toolbar vào bên trái cửa sổ chính



Gắn Toolbar vào bên trái cửa sổ chính

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ....
    // Tạo Toolbar
    // Xác lập thuộc tính Toolbar
    // Gắn Toolbar vào cửa sổ chính
    m_wndDrawBox.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_LEFT);
    DockControlBar(&m_wndDrawBox,AFX_IDW_DOCKBAR_LEFT);
    ...
}
```

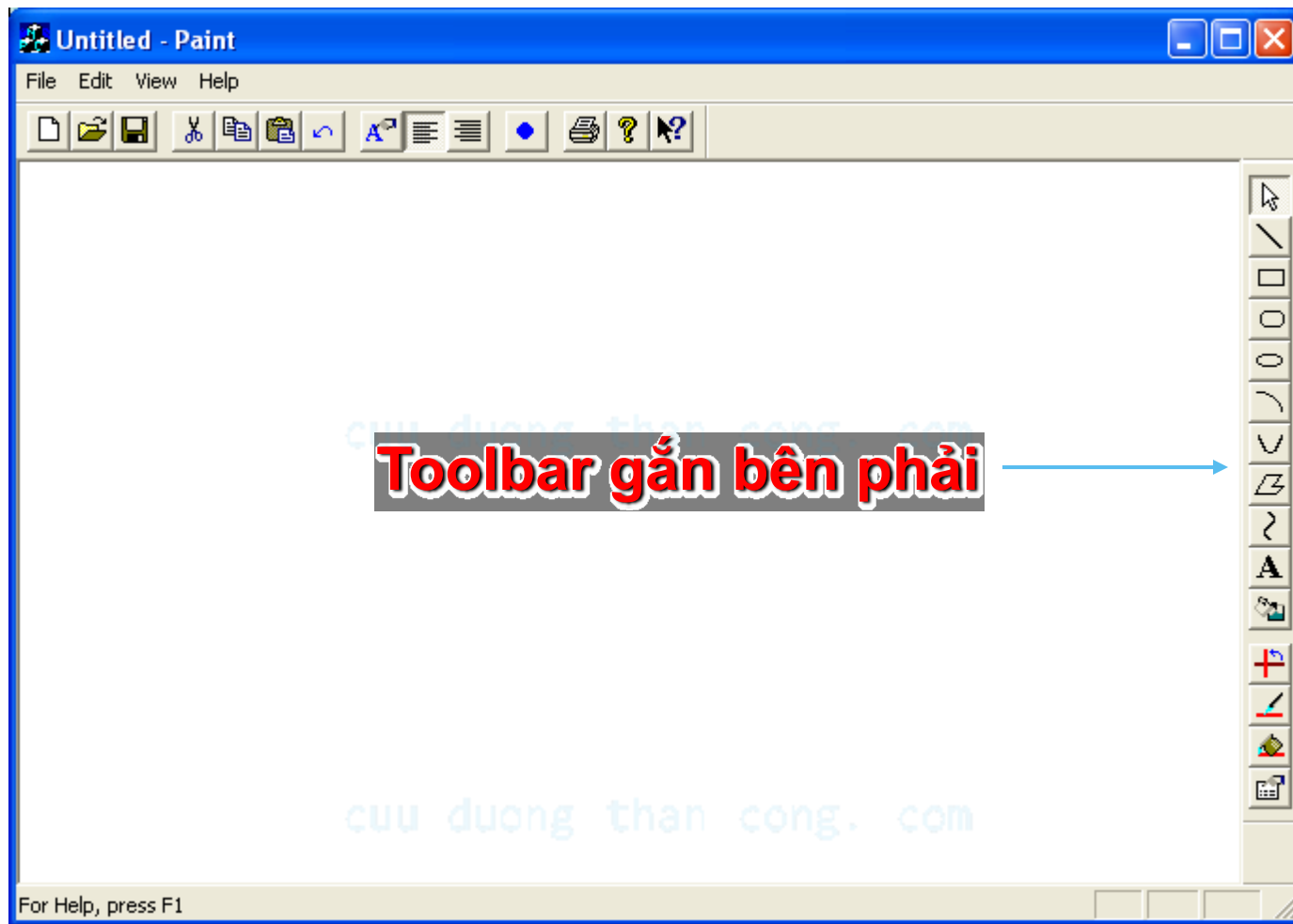
Gắn Toolbar vào bên dưới cửa sổ chính



Gắn Toolbar vào bên dưới cửa sổ chính

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ....
    // Tạo Toolbar
    // Xác lập thuộc tính Toolbar
    // Gắn Toolbar vào cửa sổ chính
    m_wndDrawBox.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_BOTTOM);
    DockControlBar(&m_wndDrawBox,AFX_IDW_DOCKBAR_BOTTOM);
    ...
}
```

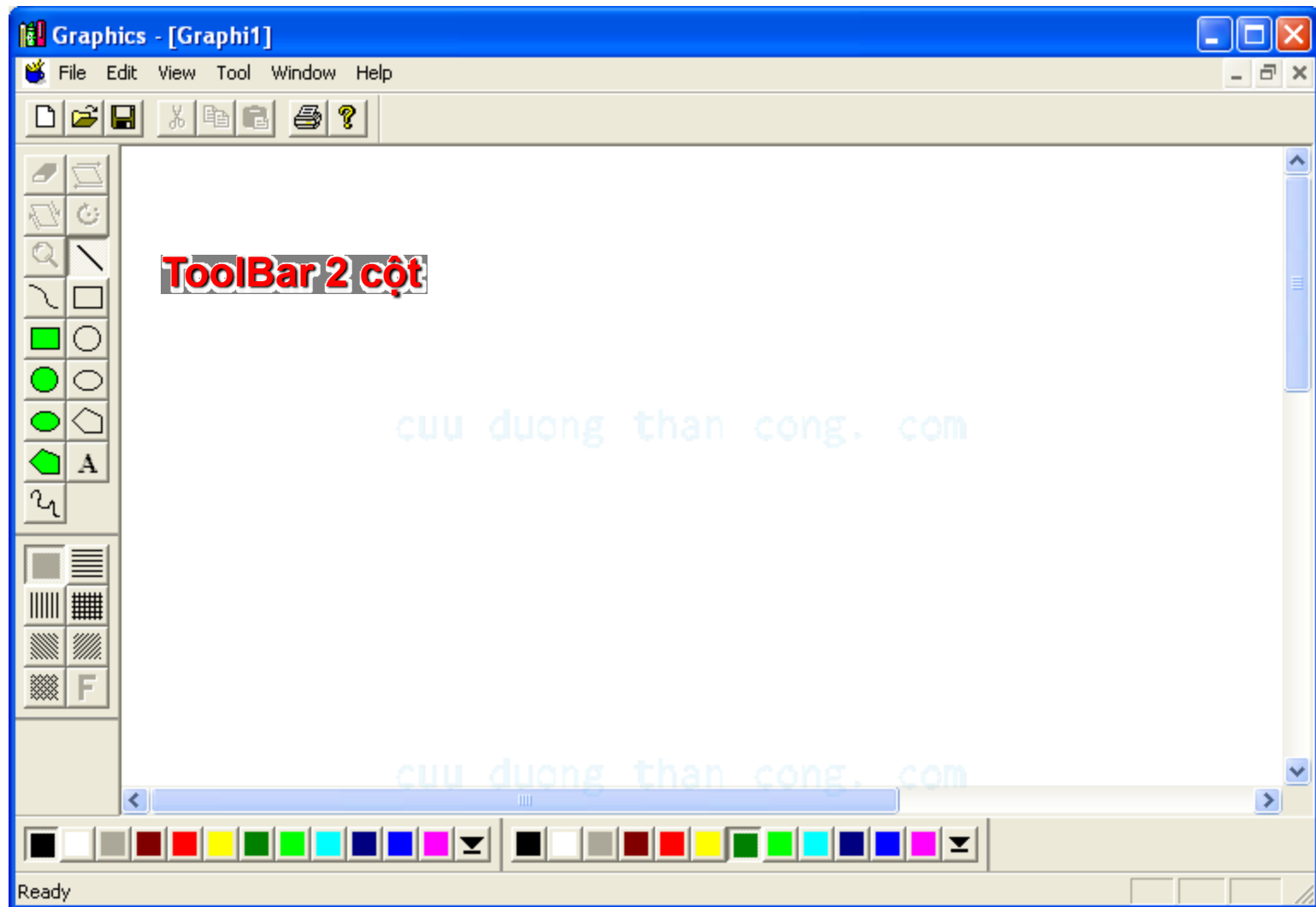
Gắn Toolbar vào bên phải cửa sổ chính



Gắn Toolbar vào bên dưới cửa sổ chính

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ....
    // Tạo Toolbar
    // Xác lập thuộc tính Toolbar
    // Gắn Toolbar vào cửa sổ chính
    m_wndDrawBox.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_RIGHT);
    DockControlBar(&m_wndDrawBox,AFX_IDW_DOCKBAR_RIGHT);
    ...
}
```

Hiển thị ToolBar N cột



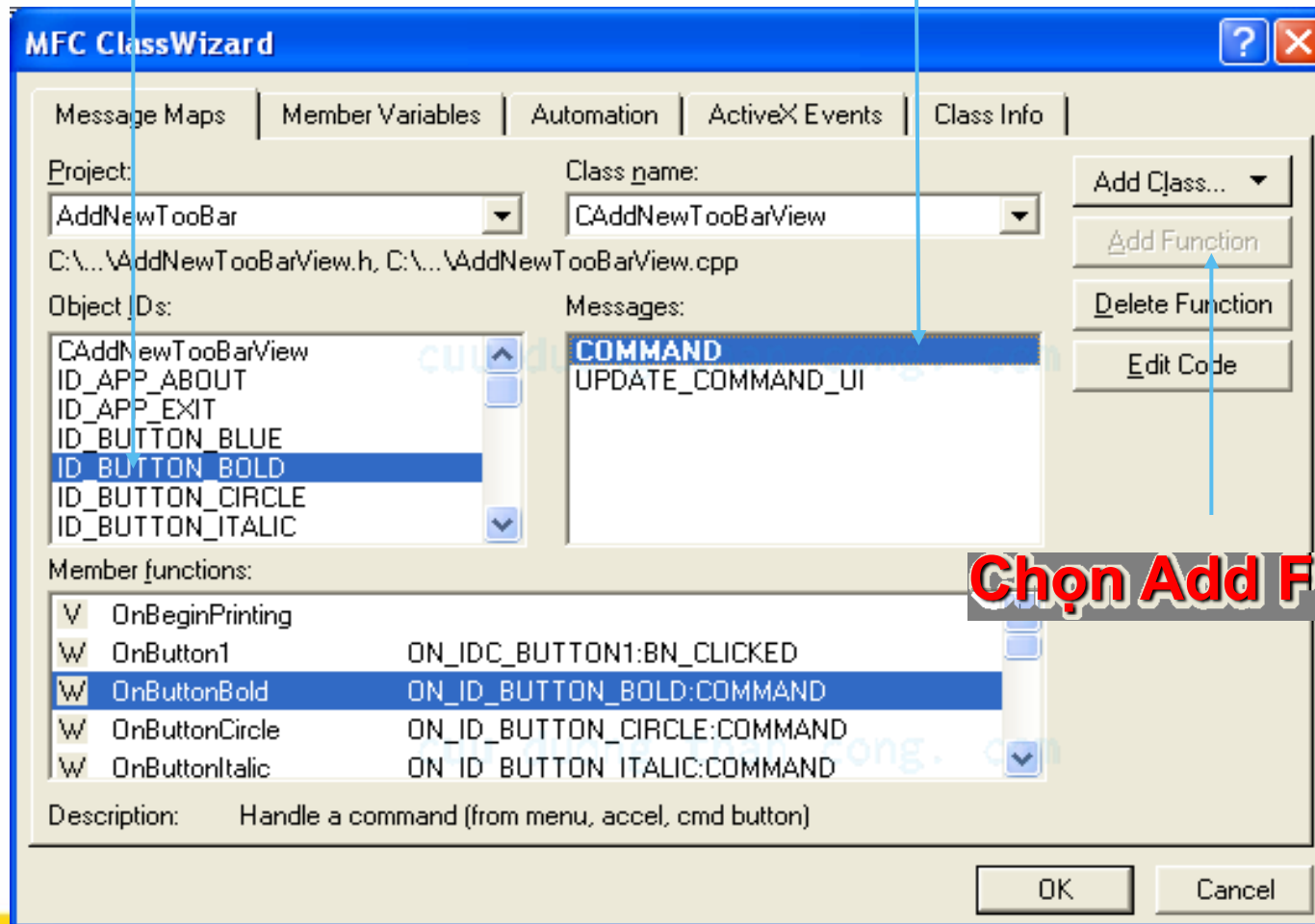
Hiển thị ToolBar N cột

```
void CMainFrame::SetColumns(int nCols,CToolBar tbToolBox)
{   int nColumns = nCols;
    int nCount = tbToolBox.GetCount();
    for(int i = 0; i < nCount; i++)
    {
        UINT nStyle = tbToolBox.GetButtonStyle(i);
        BOOL bWrap = (((i + 1) % nColumns) == 0);
        if (bWrap)
            nStyle |= TBBS_WRAPPED;
        else
            nStyle &= ~TBBS_WRAPPED;
        tbToolBox.SetButtonStyle(i, nStyle);
    }
    RecalcLayout();
    Invalidate();
}
```

Xử lý sự kiện click lên các button trong tool bar

Chọn button cần xử lý

Chọn COMMAND



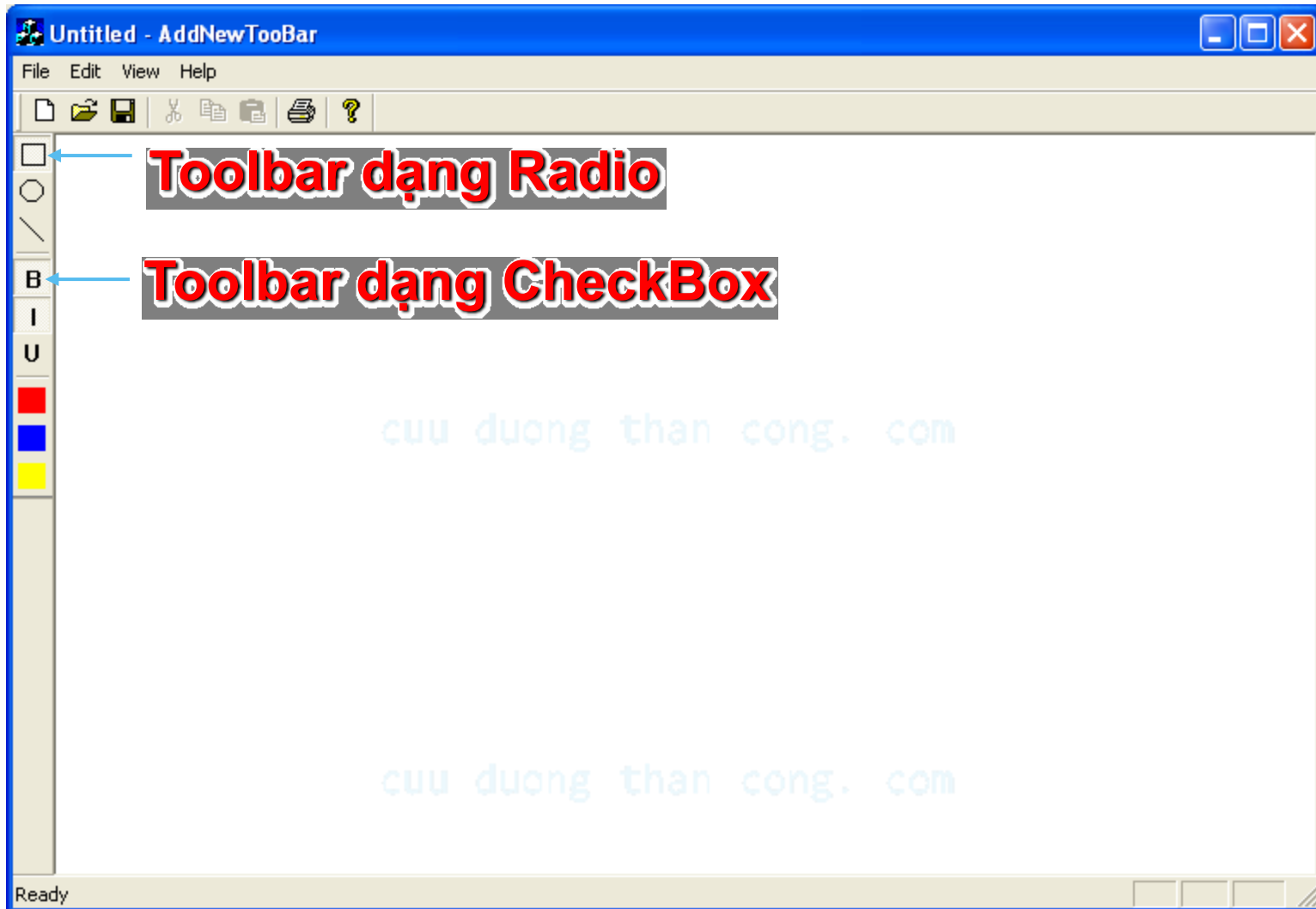
Xử lý sự kiện click lên các button trong tool bar

```
//tập tin *.h
class CAddNewToolBarView : public CView
{
    afx_msg void OnButtonBold();
};

//tập tin *.cpp
BEGIN_MESSAGE_MAP(CAddNewToolBarView, CView)
   //{{AFX_MSG_MAP(CAddNewToolBarView)
    ON_COMMAND(ID_BUTTON_BOLD, OnButtonBold)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CAddNewToolBarView::OnButtonBold()
{
    MessageBox("Nhan nut B (Bold)");
}
```

Xử lý button dạng CheckBox và Radio trên Toolbar



Xử lý CheckBox button (tt)

```
class CAddNewToolBarView : public CView
{
    //
    public:
        BOOL m_bBold; //Lưu trữ trạng thái button B
        BOOL m_bItalic; //Lưu trữ trạng thái button I
        BOOL m_bUnderline; //Lưu trữ trạng thái button U
};

CAddNewToolBarView ::CAddNewToolBarView
{
    m_bBold = TRUE; //Trạng thái button B ban đầu được "Check"
    m_bItalic=FALSE; //Trạng thái button I ban đầu không được "Check"
    m_bUnderline = FALSE; //Trạng thái button U ban đầu không được "Check"
};
```

Xử lý CheckBox button (tt)

```
void CAddNewToolBarView::OnButtonBold()
{
    m_bBold = !m_bBold;
}
void CAddNewToolBarView::OnUpdateButtonBold(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_bBold );
}
void CAddNewToolBarView::OnButtonItalic()
{
    m_bItalic = !m_bItalic;
}
void CAddNewToolBarView::OnUpdateButtonItalic(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_bItalic);
}
```

Xử lý Radio button (tt)

```
class CAddNewToolBarView : public CView
{
    //
    public:
        int m_nToolSelected; //Lưu trữ trạng thái button đang chọn
};

CAddNewToolBarView ::CAddNewToolBarView
{
    m_nToolSelected = -1; //Trạng thái ban đầu chưa có button nào được
    chọn
};
```

Xử lý Radio button (tt)

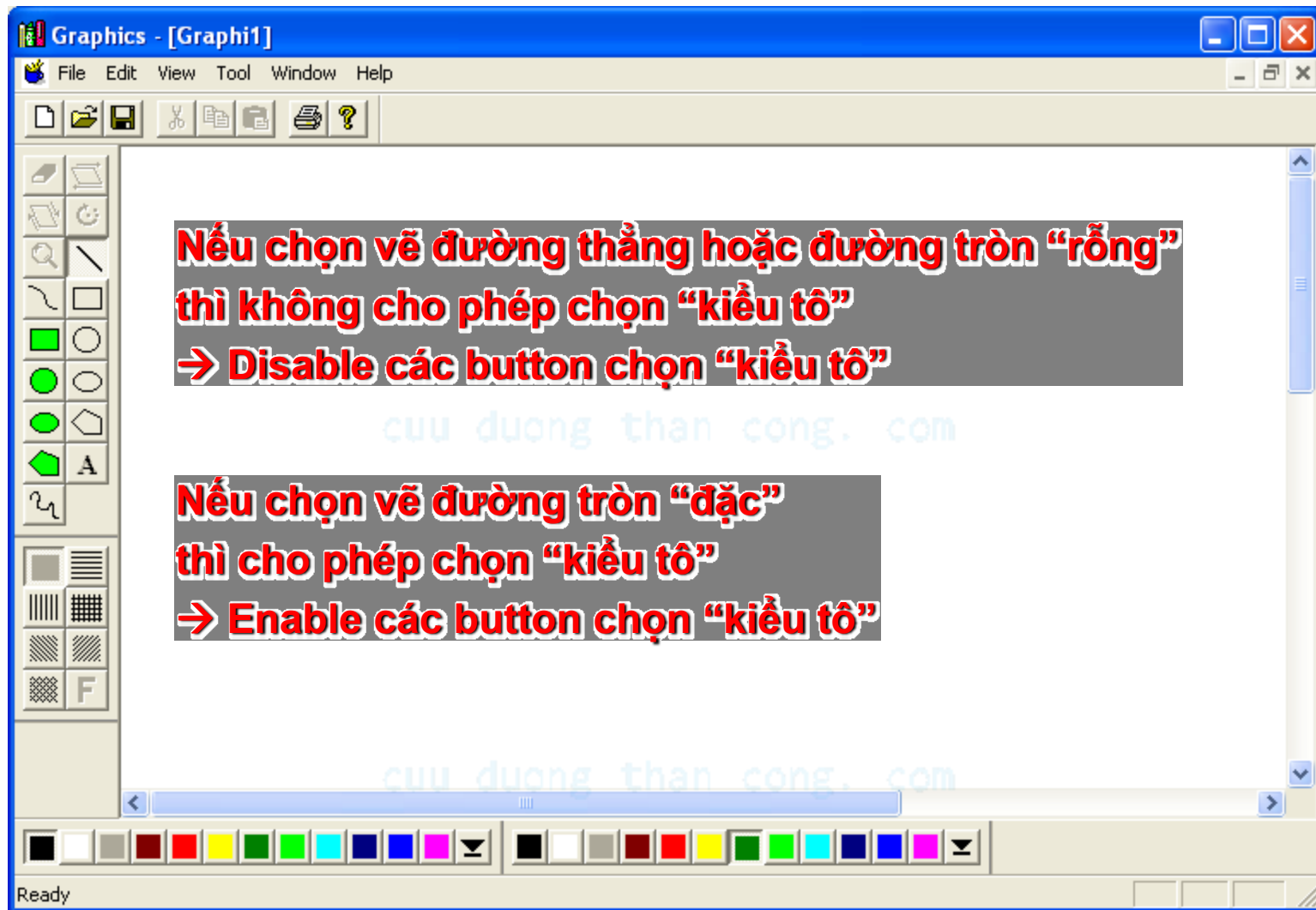
```
void CAddNewToolBarView::OnButtonLine()
{
    m_nToolSelected = 1;
}
void CAddNewToolBarView::OnUpdateButtonLine(CCmdUI* pCmdUI)
{
    cuu duong than cong. com
    if (m_nToolSelected == 1)
        pCmdUI->SetCheck(1);
    else
        pCmdUI->SetCheck(0);
}
cuu duong than cong. com
```


Xử lý Radio button (tt)

```
void CAddNewToolBarView::OnButtonCircle()
{
    m_nToolSelected = 2;
}
void CAddNewToolBarView::OnUpdateButtonItalic(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_nToolSelected == 2);
}
```

cuu duong than cong. com

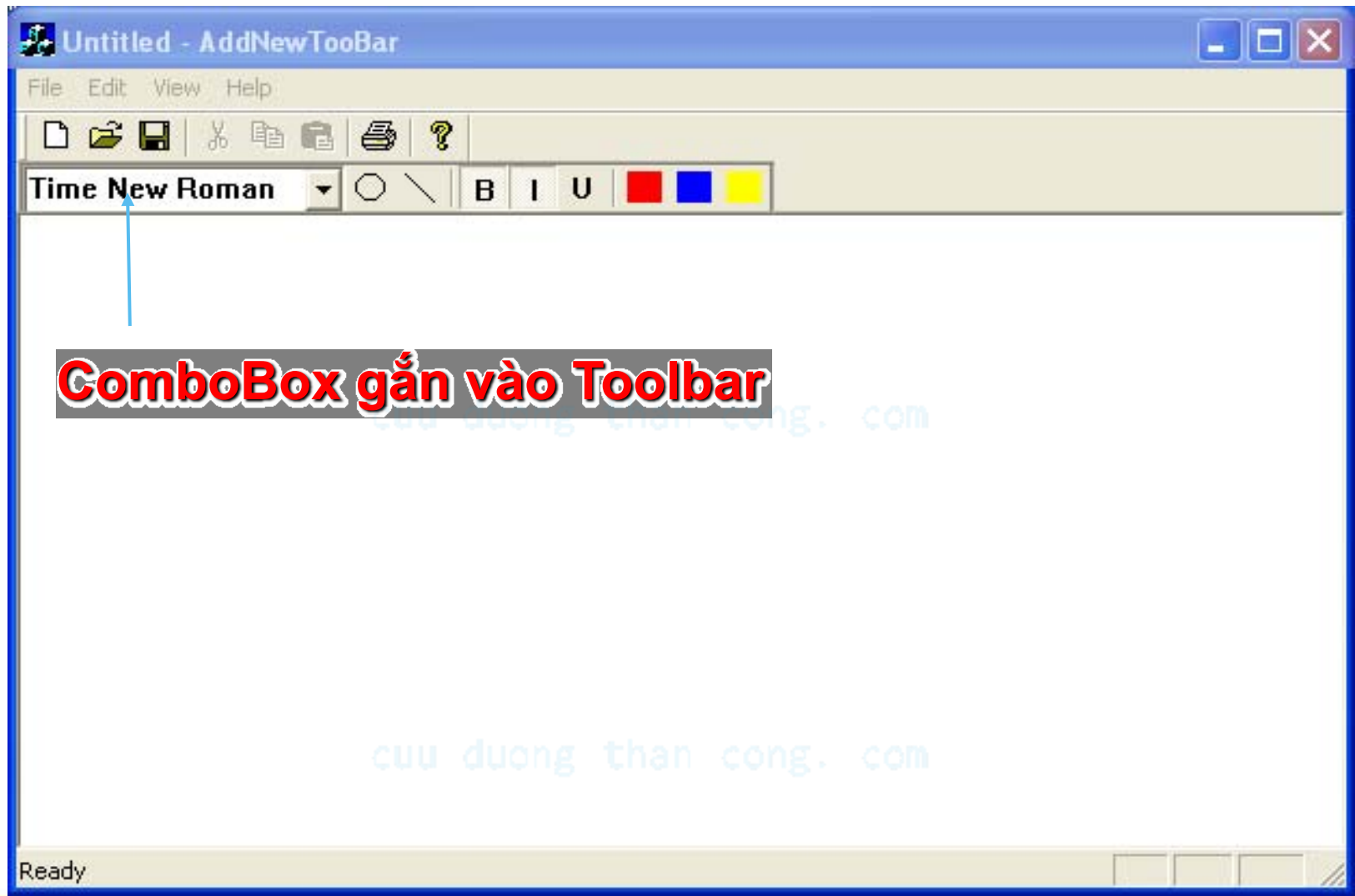
Enable/Disable button (tt)



Enable/Disable button (tt)

```
void CAddNewToolBarView::OnUpdateButtonFillStyle(CCmdUI* pCmdUI)
{
    switch (m_nToolSelected)
    {
        case 1: //Vẽ đường thẳng
            pCmdUI->Enable(0);
            break;
        case 2: //Vẽ đường tròn “đặc”
            pCmdUI->Enable(1);
            break;
    }
}
```

Gắn ComboBox vào Toolbar



Khai báo biến quản lý ComboBox

```
class CMainFrame : public CFrameWnd
{
protected:
    // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar   m_wndToolBar;
    CToolBar   m_wndDrawBox;
    CComboBox m_wndFontCombox;
};
```

Gắn ComboBox vào Toolbar (tt)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    m_wndDrawBox.SetButtonInfo(0,ID_BUTTON_RECT,TBBS_SEPARATOR,150);
    CRect rect;
    m_wndDrawBox.GetItemRect(0,rect);
    rect.bottom = rect.top+150;
    if (!m_wndFontCombox.Create(WS_CHILD
|CBS_DROPDOWN|CBS_AUTOHSCROLL|WS_VSCROLL|CBS_HASSTRINGS,
    rect,&m_wndDrawBox,ID_BUTTON_RECT))
    {
        return -1;
    }
    m_wndFontCombox.AddString("Time New Roman");
    m_wndFontCombox.AddString("Tahoma");
    m_wndFontCombox.SetCurSel(1);
    m_wndFontCombox.ShowWindow(SW_SHOW);
}
```

Xử lý dãy buttons trên Toolbar (tt)

```
class CAddNewToolBarView : public CView
{
    ....
    //Khai báo
    afx_msg void OnButtons(UINT uID);
    afx_msg void OnUpdateButtons(CCmdUI* pCmdUI);
};
//Message Map
ON_COMMAND_RANGE(ID_BUTTON_RED,ID_BUTTON_YELLOW,
OnButtons)
ON_UPDATE_COMMAND_UI_RANGE(ID_BUTTON_RED,ID_BUTTON_YELLOW,OnUpdateButtons)
```

Xử lý dãy buttons trên Toolbar (tt)

```
void CAddNewToolBarView::OnUpdateButtons(CCmdUI* pCmdUI)
{
    pCmdUI->SetRadio(pCmdUI->m_nID==m_uColor);
}
void CAddNewToolBarView::OnButtons(UINT uID)
{
    CString s;
    if (uID == ID_BUTTON_YELLOW)
        s = "Yellow";
    if (uID == ID_BUTTON_RED)
        s = "Red";
    if (uID == ID_BUTTON_BLUE)
        s = "Blue";
    MessageBox(s);
    m_uColor = uID;
}
```


Bật tắt (Ẩn/hiện) Toolbar

```
void CMainFrame::OnViewDrawBox()  
{  
    BOOL bShow;  
    m_wndDrawBox.IsWindowVisible()?FALSE: TRUE;  
    m_wndDrawBox.ShowWindow(bShow);  
}
```

cuu duong than cong. com

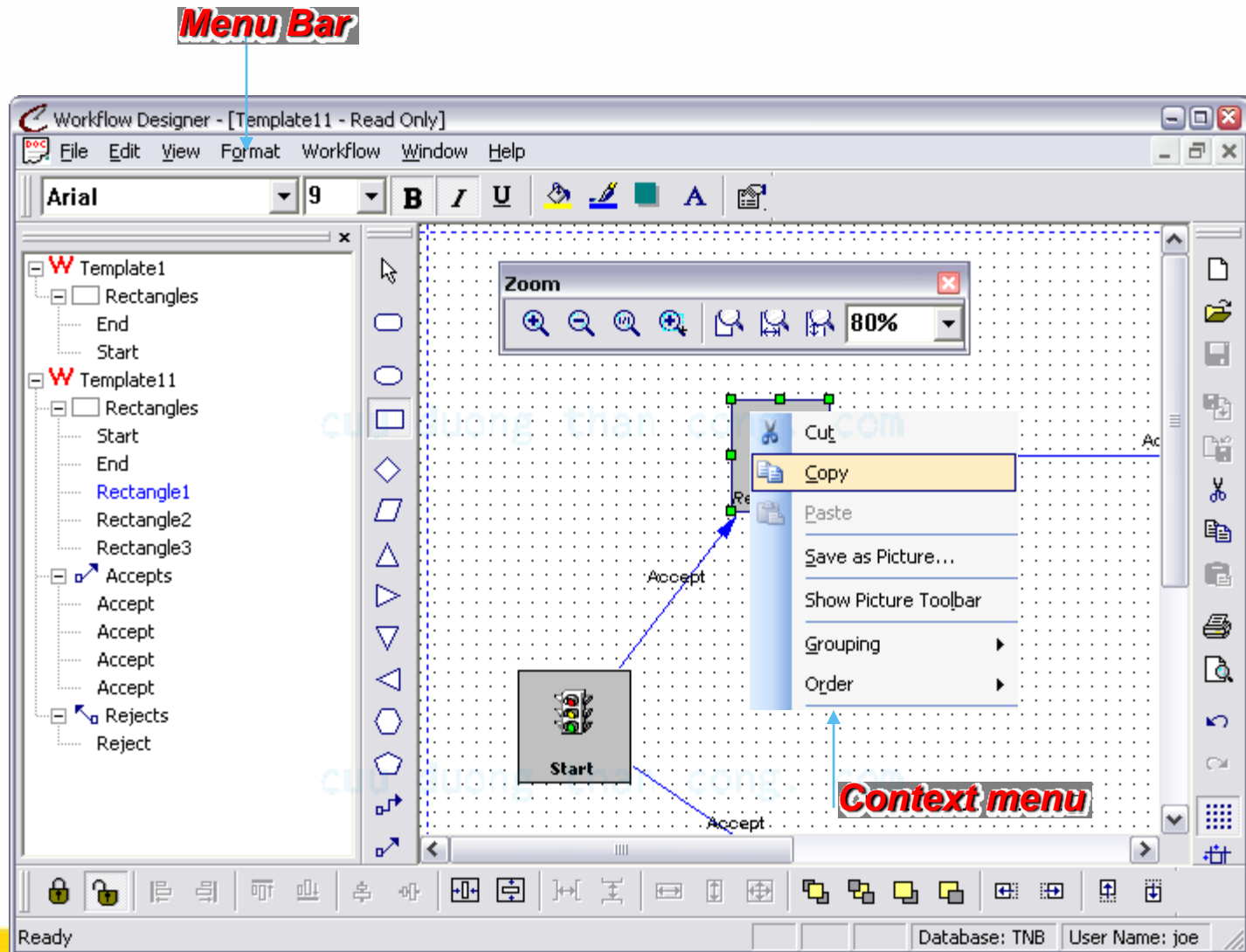
Toolbar - Menubar DialogBar

Lập trình với Menubar

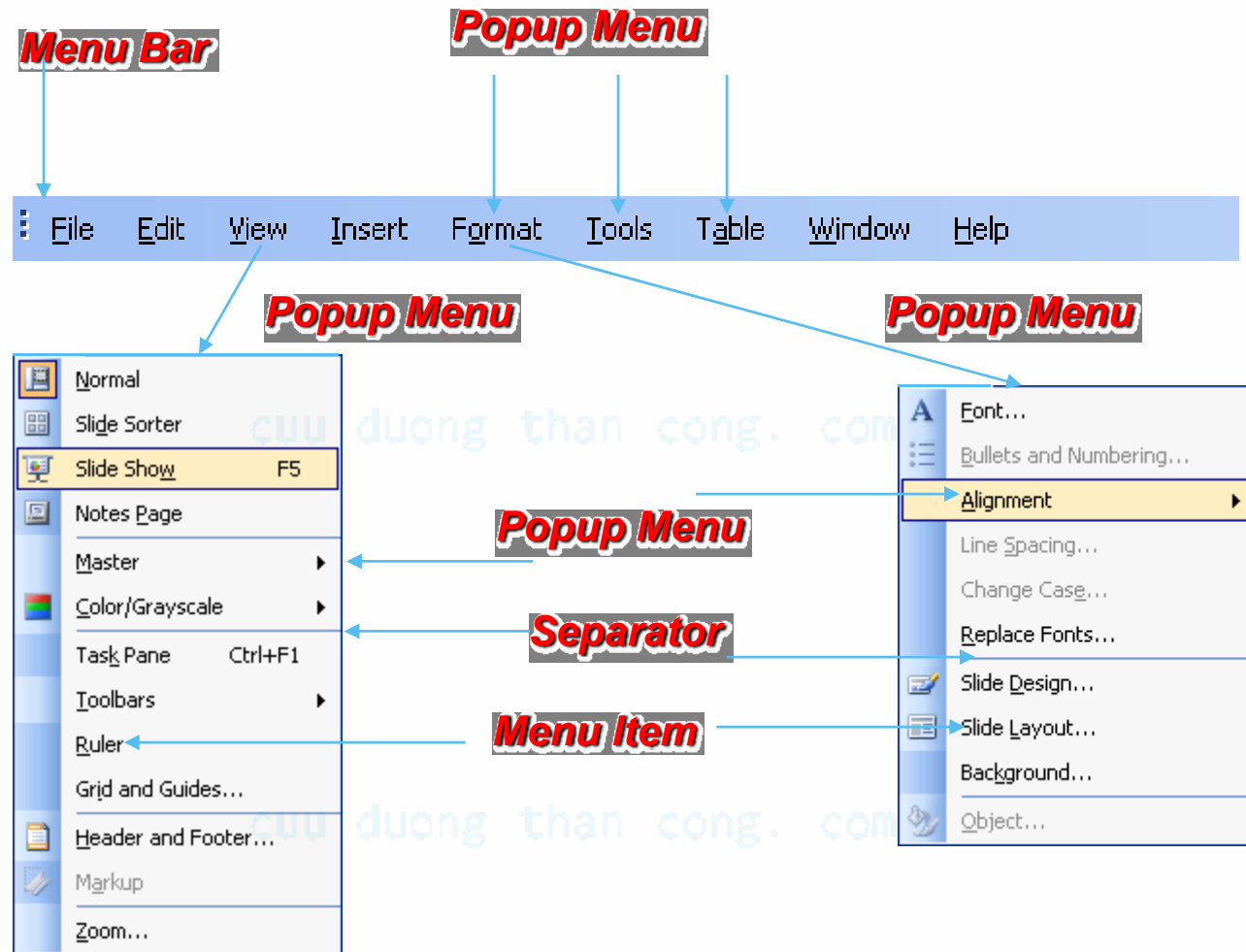
NỘI DUNG

- ❖ Giới thiệu về Menu
- ❖ Tạo và xử lý Menu
- ❖ Xử lý thay đổi Menu trong ứng dụng
- ❖ Gắn Menu vào dialog
- ❖ Right click popup menu (Context Menu)
- ❖ Thêm và hủy Menu
- ❖ Bitmap Item

Giới thiệu Menu



Các thành phần trên Menu bar



Tạo và xử lý Menu bar

- ❖ Tạo một ứng dụng có Menubar mặc định bằng AppWizard



File Edit View Help

cuu duong than cong. com

- ❖ Tạo và xử lý Menu bar bằng lớp CMenu

cuu duong than cong. com

Menubar - Tạo ứng dụng bằng AppWizard

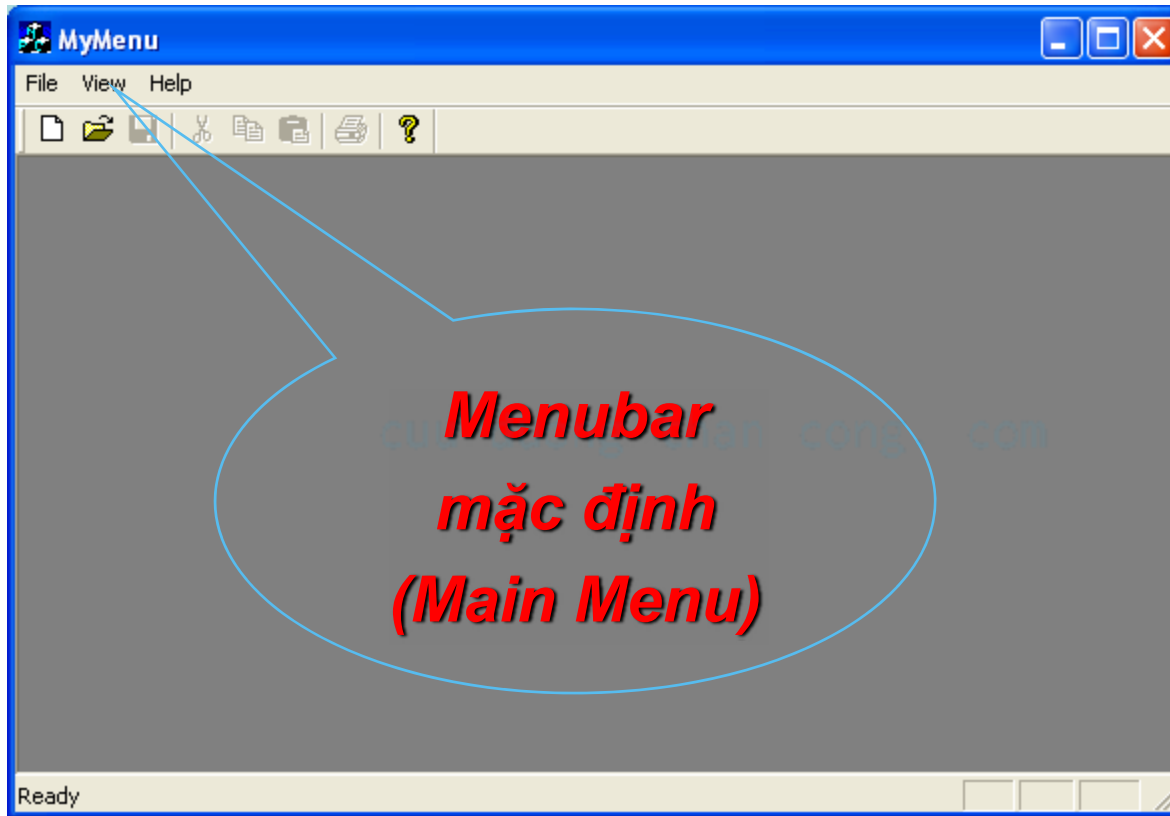
- ❖ Chọn menu **File** → **New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project "**MFC AppWizard (exe)**"
- ❖ Đặt tên project trong ô "**Project name**"
- ❖ Xác định đường dẫn thư mục trong "**Location**"
- ❖ Step 1: Chọn loại ứng dụng "**Multiple Document**", bỏ option "Document/View architecture support"
- ❖ Nhấn "**Next**" đến "**Step 4 of 6**"

Menubar - Tạo ứng dụng bằng AppWizard



Kết quả chạy chương trình tạo menu bar bằng AppWizard

Menubar - Tạo ứng dụng bằng AppWizard

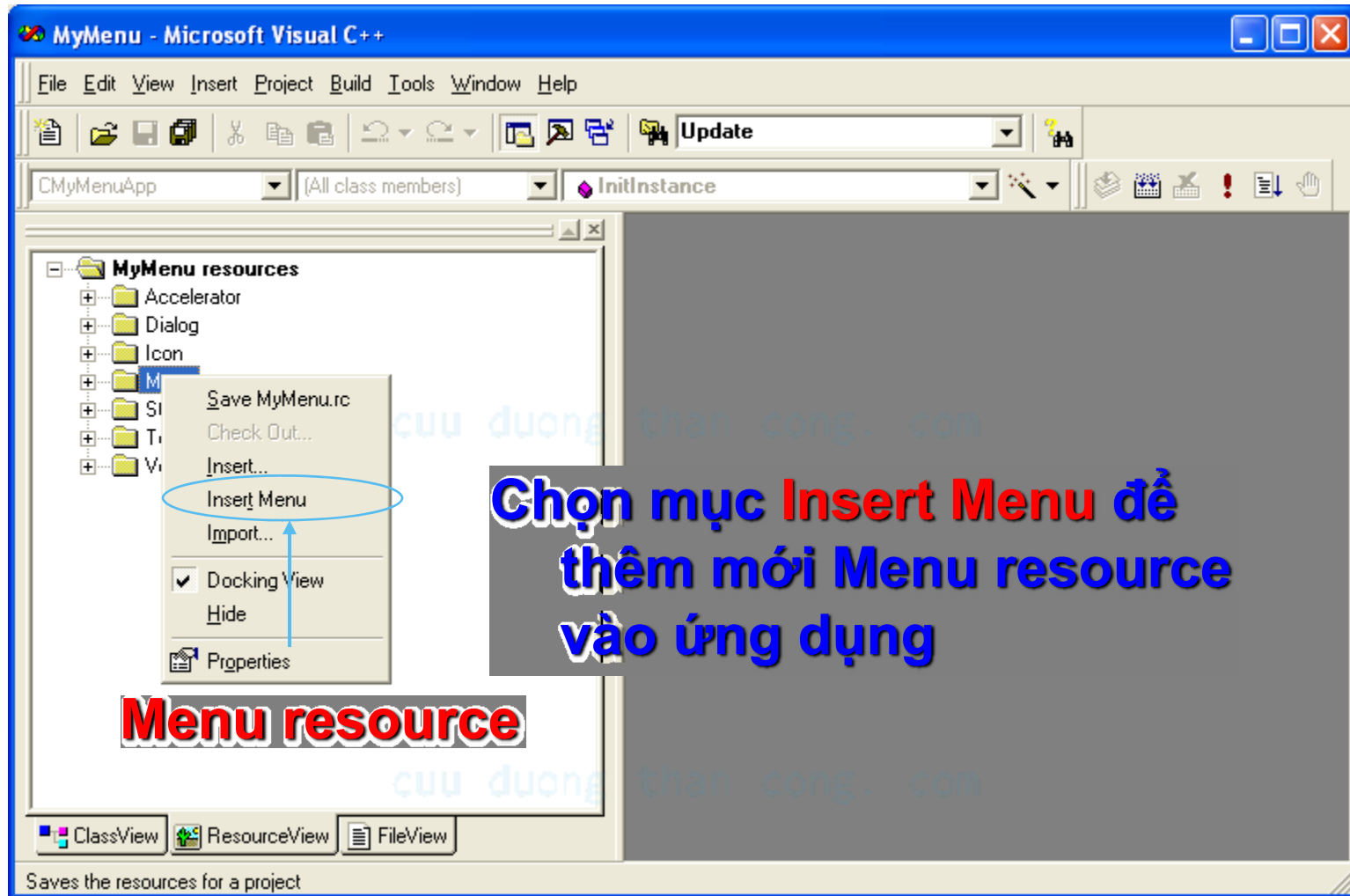


Kết quả chạy chương trình tạo menu bar bằng AppWizard

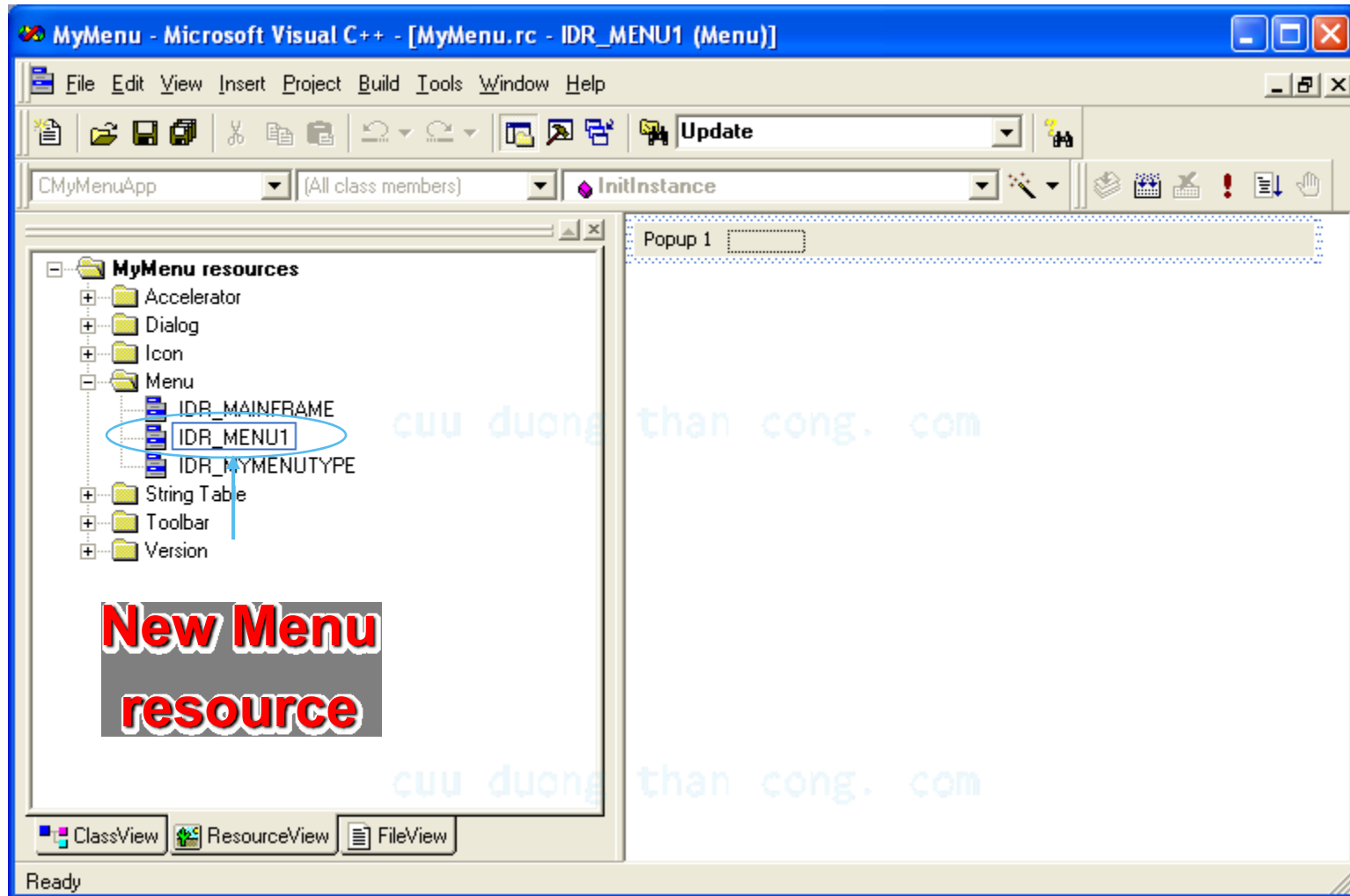
Các xử lý trong lớp CxxxApp

```
BOOL CMyFirstMenuApp::InitInstance()
{
    AfxEnableControlContainer();
    ....
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MENU_TOOL, //Menu o cho nay
        RUNTIME_CLASS(CMyFirstMenuDoc),
        RUNTIME_CLASS(CMainFrame), // main SDI frame window
        RUNTIME_CLASS(CMyFirstMenuView));
    AddDocTemplate(pDocTemplate);
    // Parse command line for standard shell commands, DDE, file open
    m_pMainWnd->ShowWindow(SW_SHOW);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}
```

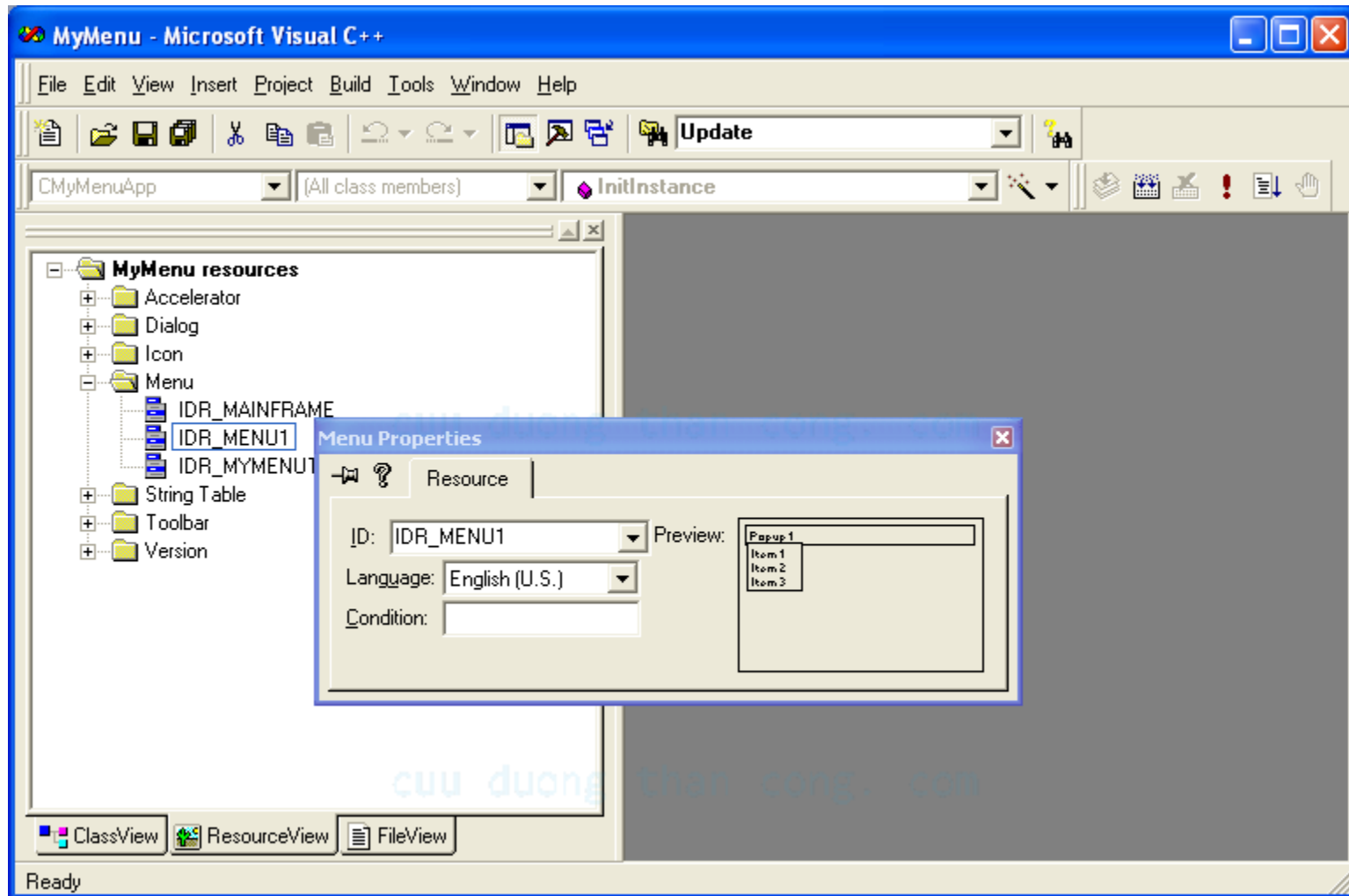
Thêm mới Menu resource



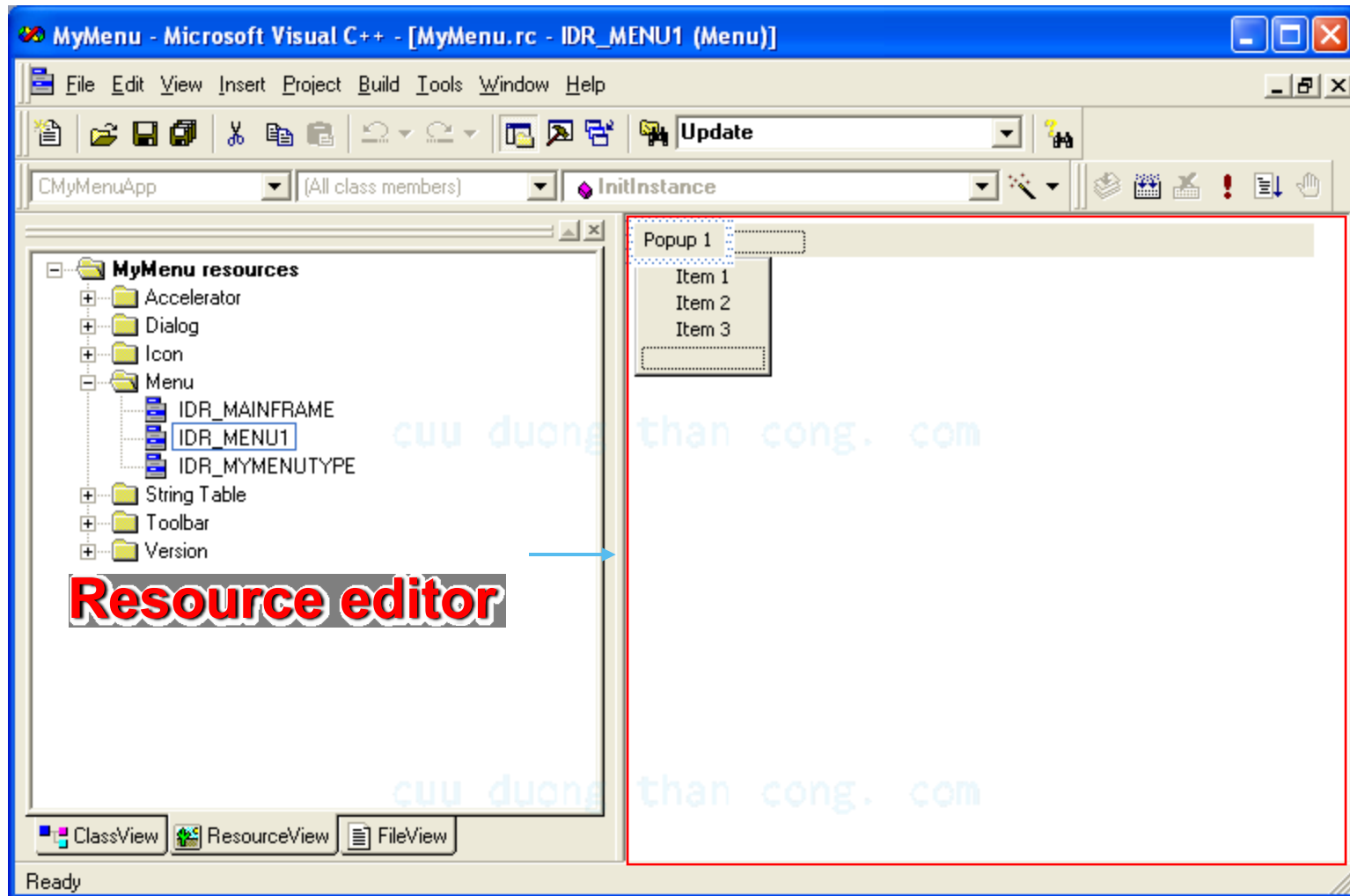
Thêm mới Menu resource (tt)



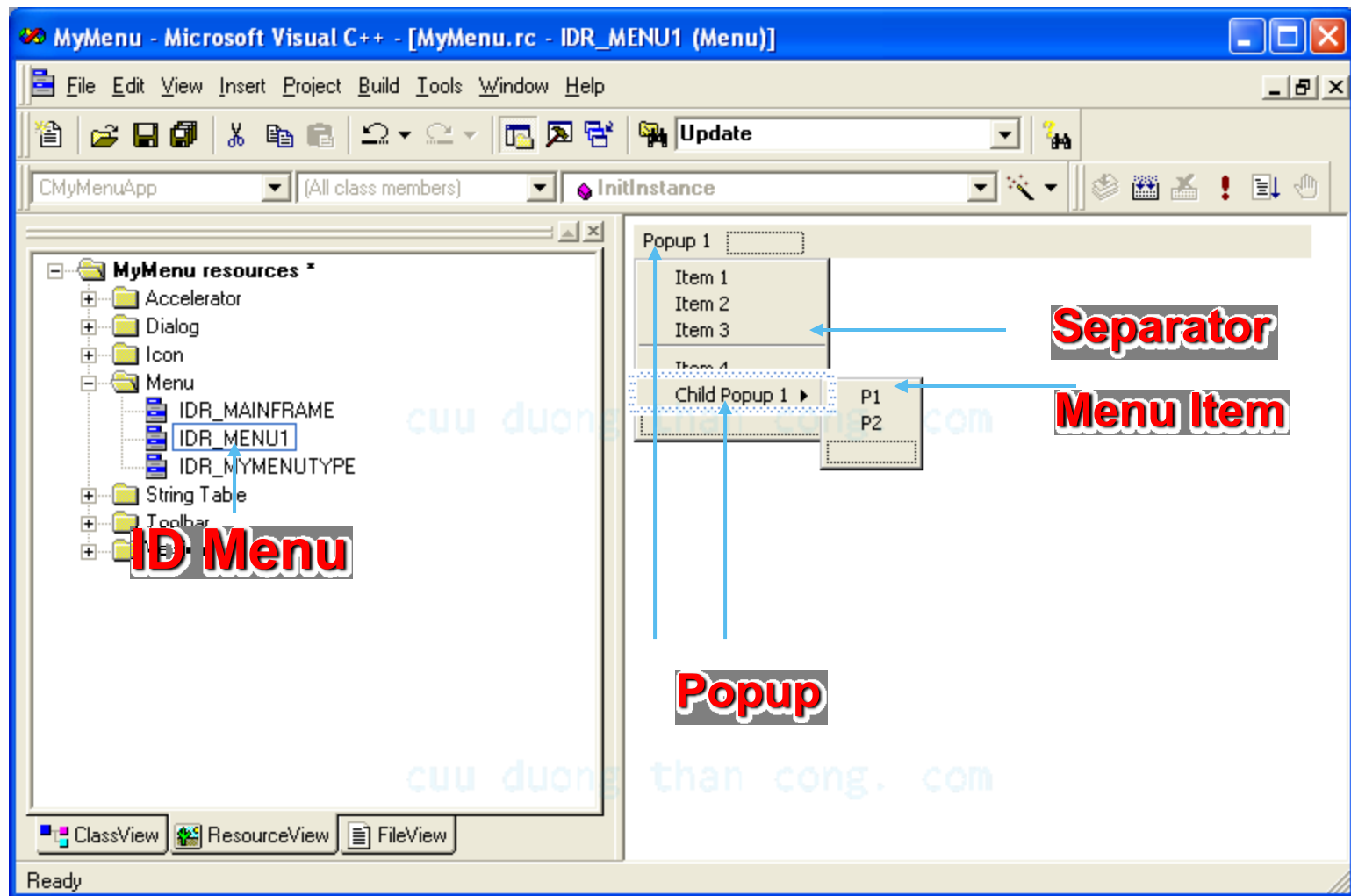
Thêm mới Menu resource (tt)



Thêm mới Menu resource (tt)



Thêm mới Menu resource (tt)



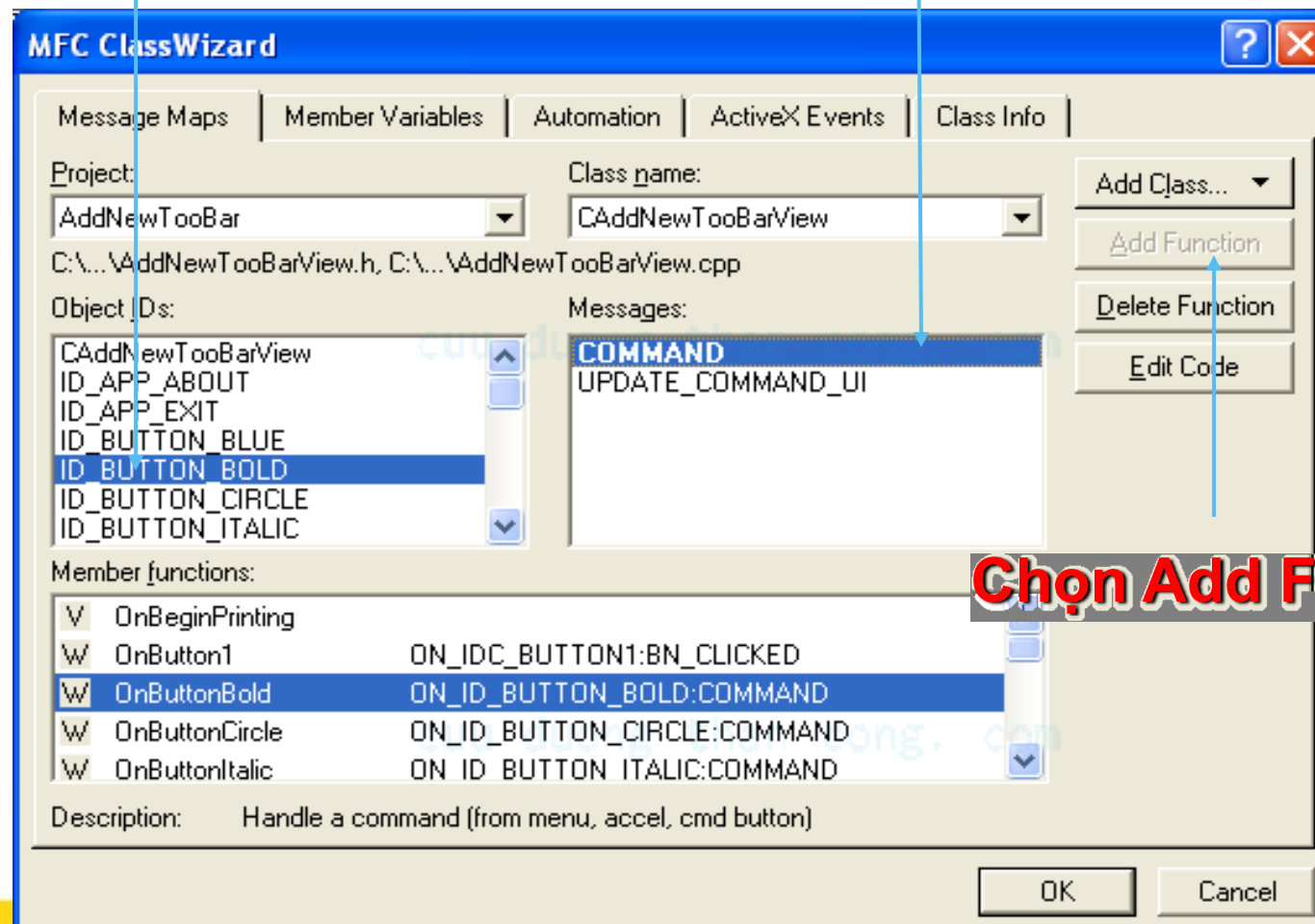
Xử lý sự kiện click lên các button trong Menu

- ❖ Thông điệp WM_COMMAND
- ❖ Thông điệp UPDATE_COMMAND_UI
- ❖ Bật/tắt MenuItem (Enable/Disable)
- ❖ Thay đổi Caption của MenuItem
- ❖ Checking MenuItem

Xử lý sự kiện click lên các button trong Menu

Chọn Item cần xử lý

Chọn COMMAND



Chọn Add Function

Xử lý sự kiện click lên các button trong tool bar

```
class CAddNewToolBarView : public CView
{
    afx_msg void OnButtonBold();
};
BEGIN_MESSAGE_MAP(CAddNewToolBarView, CView)
   //{{AFX_MSG_MAP(CAddNewToolBarView)
    ON_COMMAND(ID_BUTTON_BOLD, OnButtonBold)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()
void CAddNewToolBarView::OnButtonBold()
{
    MessageBox("Nhan nut B (Bold");
}
```

Xử lý Item dạng CheckBox và Radio trên Menu

- ❖ Xử lý tương tự như button dạng checkbox và Radio button trong Toolbar
- ❖ Gán ID của MenuItem có cùng ID với các Item trong Toolbar → Click lên MenuItem và Item trên Toolbar sẽ thực hiện cùng một chức năng trong chương trình

Xử lý CheckBox button (tt)

```
class CAddNewToolBarView : public CView
{
    //
    public:
        BOOL m_bBold; //Lưu trữ trạng thái button B
        BOOL m_bItalic; //Lưu trữ trạng thái button I
        BOOL m_bUnderline; //Lưu trữ trạng thái button U
};

CAddNewToolBarView ::CAddNewToolBarView
{
    m_bBold = TRUE; //Trạng thái button B ban đầu được "Check"
    m_bItalic=FALSE; //Trạng thái button I ban đầu không được "Check"
    m_bUnderline = FALSE; //Trạng thái button U ban đầu không được
    "Check"
};
```

Xử lý CheckBox button (tt)

```
void CAddNewToolBarView::OnButtonBold()
{
    m_bBold = !m_bBold;
}
void CAddNewToolBarView::OnUpdateButtonBold(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_bBold );
}
void CAddNewToolBarView::OnButtonItalic()
{
    m_bItalic = !m_bItalic;
}
void CAddNewToolBarView::OnUpdateButtonItalic(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_bItalic);
}
```

Xử lý Radio button (tt)

```
class CAddNewToolBarView : public CView
{
    //
    public:
        int m_nToolSelected; //Lưu trữ trạng thái button đang chọn
};
```

cuuduongthancong.com

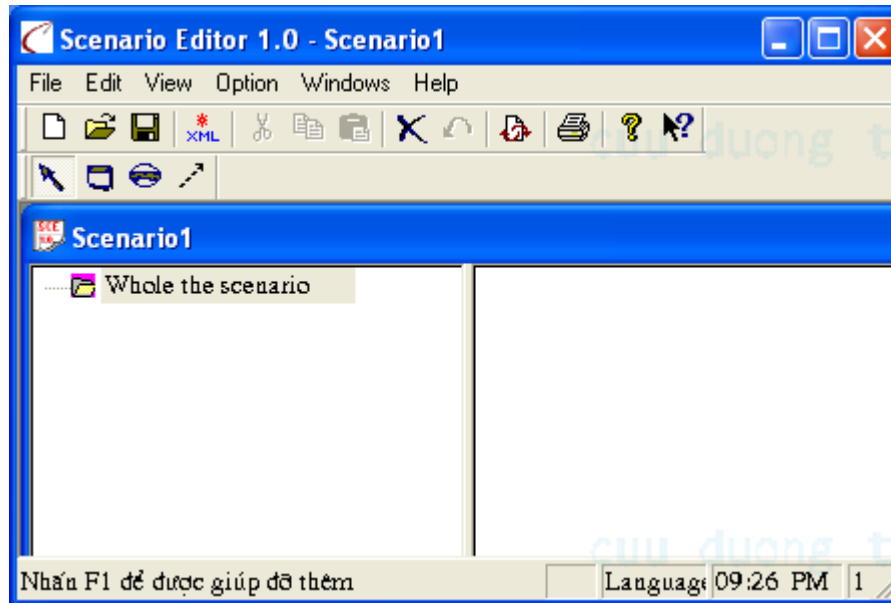
```
CAddNewToolBarView ::CAddNewToolBarView
{
    m_nToolSelected = -1; //Trạng thái ban đầu chưa có button nào được chọn
};
```

cuuduongthancong.com

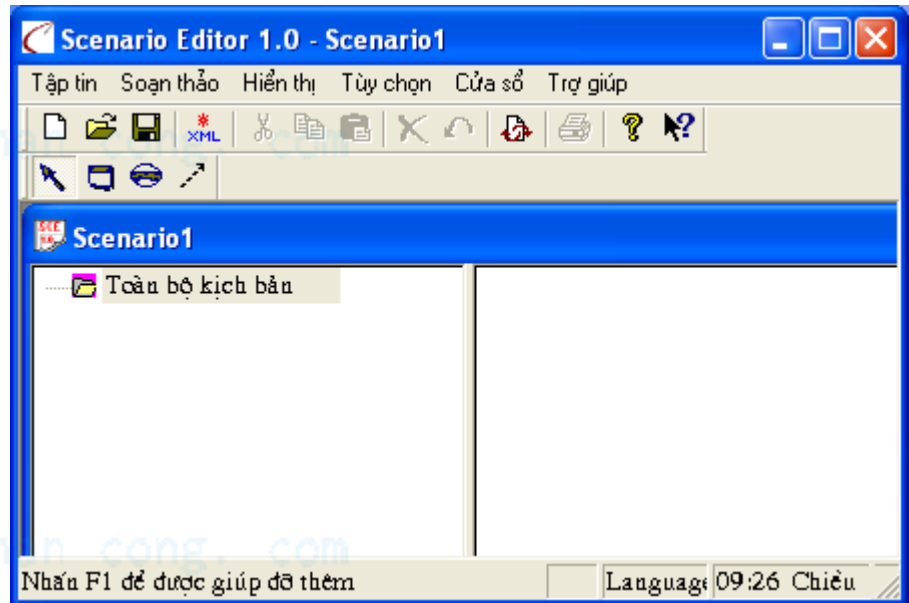
Thay đổi Menu trong ứng dụng

✓ Yêu cầu

Chương trình cho phép người dùng chọn hiển thị menu Tiếng Anh/Tiếng Việt như hình



Menu tiếng Anh



Menu tiếng Việt

Thay đổi Menu trong ứng dụng (tt)

✓ Cách 1:

Thay đổi thuộc tính Text trong mỗi MenuItem trong Menu

```
void CMyFirstMenuView::OnUpdateToolsCircle(CCmdUI*  
pCmdUI)  
{  
    CString s;    cuu duong than cong. com  
    if (m_LanguageSel== 'VN' )  
        s.LoadString(ID_TOOLS_CIRCLE_VN) ;  
    else  
        s.LoadString(ID_TOOLS_CIRCLE) ;  
    pCmdUI->SetText(s) ;    than cong. com  
}
```


Thay đổi Menu trong ứng dụng (tt)

✓ Cách 1:

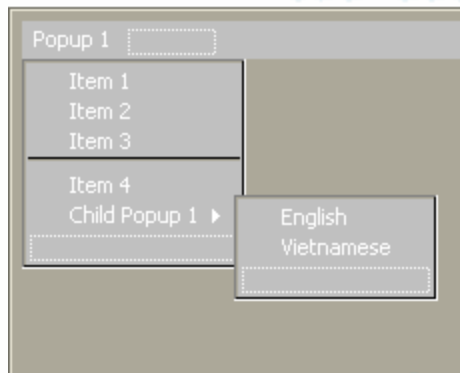
Thay đổi thuộc tính Text trong mỗi MenuItem trong Menu

```
void CMyFirstMenuView::OnUpdateToolsLine (CCmdUI*  
pCmdUI)  
{  
    CString s;  
    if (m_LanguageSel== 'VN' )  
        s.LoadString(ID_TOOLS_LINE_VN) ;  
    else  
        s.LoadString(ID_TOOLS_LINE) ;  
    pCmdUI->SetText (s) ;  
}
```

Thay đổi Menu trong ứng dụng (tt)

✓ Cách 2:

Thiết kế 2 Menu trong Menu Resource có cùng cấu trúc: cùng số Item và các Item trong 2 Menu resource đôi 1 có cùng ID



Menu tiếng Anh



Menu tiếng Việt

Thay đổi Menu trong ứng dụng (tt)

```
void CMainFrame::OnEnglish()  
{  
    // TODO: Add your command handler code here  
    m_myMenu.DestroyMenu();  
    m_NN =1;  
    NewMyMenu();  
}  
void CMainFrame::OnVietnamese()  
{  
    // TODO: Add your command handler code here  
    m_myMenu.DestroyMenu();  
    m_NN =2;  
    NewMyMenu();  
}
```

Thay đổi Menu trong ứng dụng (tt)

```
void CMainFrame::NewMyMenu()  
{  
    if (m_NN==1)  
    {  
        m_myMenu.LoadMenu(IDR_MENU_EN);  
    }  
    else  
        m_myMenu.LoadMenu(IDR_MENU_VN);  
  
    SetMenu(&m_myMenu);  
  
}
```

Gắn menu vào dialog

- i. Tạo menu resource
- ii. Tạo dialog resource
- iii. Gắn menu vào dialog
 - Chọn dialog resource trong ResourceView
 - Click phải chuột chọn Properties



- Trong Tab General, chọn Menu ID tương ứng trong combo Menu

Context Menu - Right Click Popup Menu

❖ Thêm Resource Menu

Thực hiện tương tự như thêm menu bình thường

❖ Cài đặt Right Click menu

Nhấn chuột phải lên vùng Client hiển thị popup menu

```
void CMyFirstMenuView::OnRButtonDown(UINT nFlags, CPoint
point)
{
    CMenu menu;
    menu.LoadMenu(IDR_MENU1);
    ClientToScreen(&point); // Hiển thị popup tại vị trí click chuột
    menu.GetSubMenu(0) ->TrackPopupMenu(
TPM_LEFTALIGN|TPM_RIGHTBUTTON,point.x,point.y,this,NULL);
}
```

Xác lập trạng thái của Item trong Popup Menu

```
void CMyFirstMenuView::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CMenu menu;
    CMenu *ptrMenu;
    //Load Menu
    menu.LoadMenu(IDR_MENU1);
    ptrMenu = menu.GetSubMenu(0);
    //Xac lap trang thai cua cac Item
    ptrMenu->EnableMenuItem(ID_POPUP_POPUPITEM01,MF_GRAYED);
    ptrMenu->EnableMenuItem(ID_POPUP_POPUPITEM02,MF_ENABLED);
    ptrMenu->EnableMenuItem(ID_POPUP_POPUPITEM03,MF_CHECKED);
    //Di chuyen vao dung vi tri cua con chuot
    ClientToScreen(&point);
    //Hien thi menu
    ptrMenu->TrackPopupMenu(
        TPM_LEFTALIGN|TPM_RIGHTBUTTON,point.x,point.y,this,NULL);
    CView::OnRButtonDown(nFlags, point);
}
```

Thêm và Huỷ MenuItem

❖Cú pháp

InsertMenu(nPosition, nFlags, NewID, Caption)

MF_BYPOSITION: dựa vào vị trí

MF_BYCOMMAND: dựa vào ID, SEPARATOR không có ID

❖Separator:

nFlags = MF_SEPARATOR

NewID = ANY

Caption = ANY

❖MenuItem

nFlags = None

NewID = NEWID

Caption

Thêm MenuItem

```
void CMyFirstMenuView::OnPopupInsert()  
{  
    // TODO: Add your command handler code here  
    CMenu *topMenu = AfxGetMainWnd()->GetMenu();  
    CMenu *ptrMenu = topMenu->GetSubMenu(0);  
    UINT nNewID = ID_USER_MENUITEM+m_nItem;  
    m_nItem = m_nItem+1;  
    ptrMenu->InsertMenu(1, MF_BYPOSITION, nNewID, "&Menu  
Caption");  
    AfxGetMainWnd()->DrawMenuBar();  
}
```

Thêm PopupMenu

```
void CMyFirstMenuView::OnPopupMenuInsert()  
{  
    // TODO: Add your command handler code here  
    CMenu *topMenu = AfxGetMainWnd()->GetMenu();  
    CMenu *ptrMenu = topMenu->GetSubMenu(0);  
    //Handle của Sub Menu(0) được lưu vào *ptrMenu  
    topMenu->  
>InsertMenu(1, MF_BYPOSITION, (UINT)ptrMenu->  
>GetSafeHmenu(), "&Menu Caption");  
    AfxGetMainWnd()->DrawMenuBar();  
}
```

Xóa/Hủy menuItem

❖ Remove 1 Popup trong MainMenu

```
void CMyFirstMenuView::OnPopupDelete()  
{  
    CMenu *topMenu = AfxGetMainWnd()->GetMenu();  
    topMenu->RemoveMenu(1, MF_BYPOSITION);  
    AfxGetMainWnd()->DrawMenuBar();  
}
```

cuu duong than cong. com

Xóa/Hủy menuItem

❖ Remove 1 Item trong Popup thứ nhất trong MainMenu

```
void CMyMenuView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CMenu *topMenu = AfxGetMainWnd()->GetMenu();
    CMenu *ptrMenu = topMenu->GetSubMenu(0);
    ptrMenu->RemoveMenu(1, MF_BYPOSITION);
    AfxGetMainWnd()->DrawMenuBar();
    CView::OnLButtonDown(nFlags, point);
}
```

cuu duong than cong. com

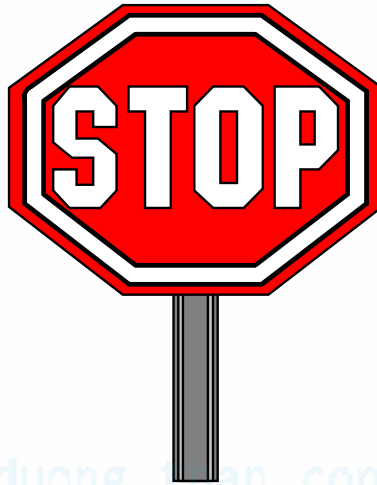
Toolbar - Menubar DialogBar

Lập trình với Dialogbar

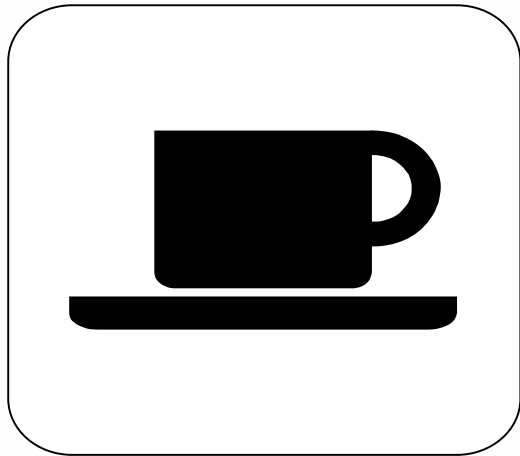
Tạo và xử lý Dialog Bar

```
class CMainFrame : public CFrameWnd
{
    ....
    CDialogBar    m_wndDialogBar;
};

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (!m_wndDialogBar.Create(this,IDD_DIALOG_BAR,CBRS_TOOLTIPS |
    CBRS_FLYBY,IDD_DIALOG_BAR))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }
    m_wndDialogBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_TOP);
    DockControlBar(&m_wndDialogBar,AFX_IDW_DOCKBAR_TOP);
}
```



cuu duong than cong. com



cuu duong than cong. com



Thư viện đồ họa GDI (Graphics Device Interface)

Lập trình đồ họa với thư viện GDI

Trần Ngọc Bảo

Email: tnbao.dhsp@gmail.com

TÌM HIỂU ỨNG DỤNG PAINT

- ❖ Vẽ các đối tượng trong GDI

 - ❖ Line

 - ❖ Rectangle

 - ❖ Circle

 - ❖ ...

- ❖ Chọn đối tượng

- ❖ Di chuyển đối tượng

- ❖ Lưu trữ các đối tượng

TÌM HIỂU ỨNG DỤNG PAINT

❖ Vẽ các đối tượng trong GDI

- ❖ Line

- ❖ Rectangle

- ❖ Circle

❖ Chọn đối tượng

❖ Di chuyển đối tượng

❖ Lưu trữ các đối tượng

VỀ ĐỐI TƯỢNG



TÌM HIỂU ỨNG DỤNG PAINT

❖ Vẽ các đối tượng trong GDI

❖ Line

❖ Rectangle

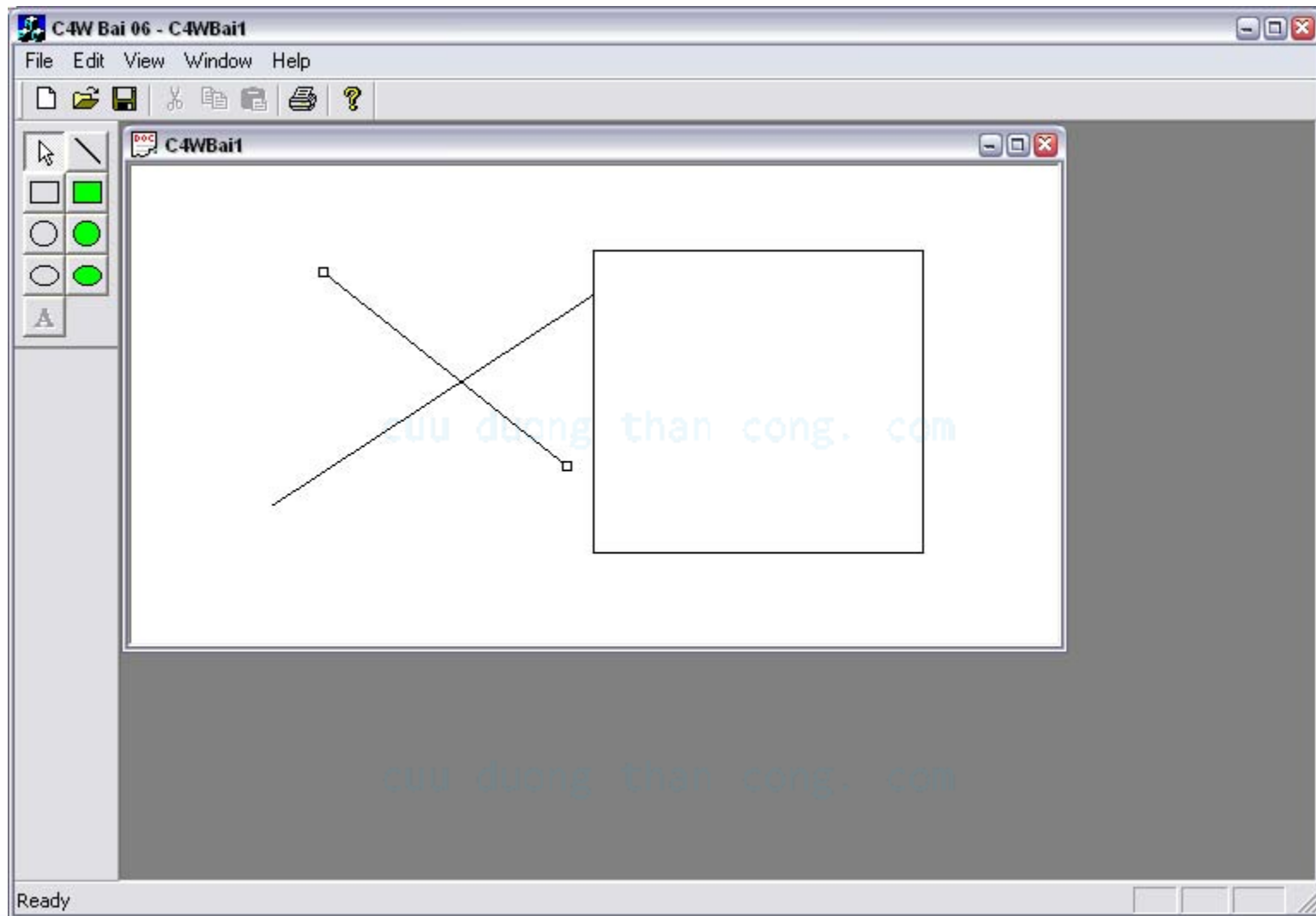
❖ Circle

❖ **Chọn đối tượng**

❖ Di chuyển đối tượng

❖ Lưu trữ các đối tượng

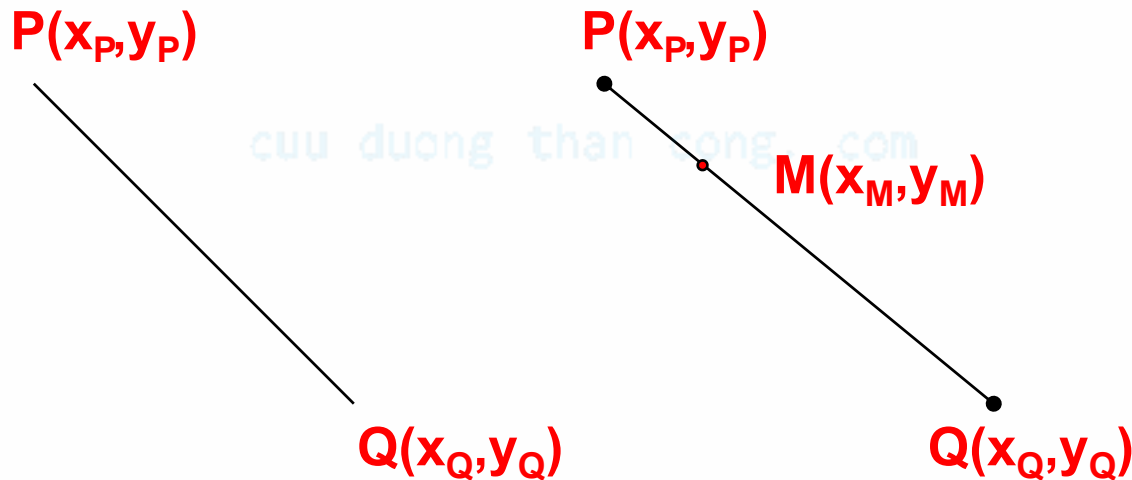
CHỌN ĐỐI TƯỢNG



CHỌN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

❖ Về mặt toán học

- Đoạn thẳng được tạo bởi 2 điểm $P(x_P, y_P), Q(x_Q, y_Q)$
- Gọi $M(x_M, y_M)$ là vị trí của chuột



Chọn đường thẳng tương đương với việc di chuyển chuột và click lên đường thẳng PQ
→ Tọa độ M của chuột nằm trên đường thẳng PQ hay $M \in PQ$

CHỌN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

❖ Về mặt toán học

- Phương trình chính tắc đường thẳng PQ

$$\frac{x - x_P}{x_P - x_Q} = \frac{y - y_P}{y_P - y_Q}$$

P

Q

$$Ax + By + C = 0$$

$$A = y_P - y_Q$$

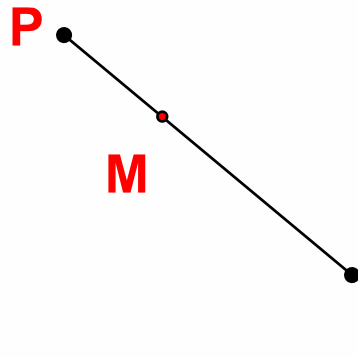
$$B = x_Q - x_P$$

$$C = x_P * y_Q - x_Q * y_P$$

CHỌN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

❖ Về mặt toán học

- Phương trình chính tắc đường thẳng PQ



$$Ax + By + C = 0 \text{ (d)}$$

$$A = y_P - y_Q$$

$$B = x_Q - x_P$$

$$C = x_P * y_Q - x_Q * y_P$$

Tọa độ M của chuột nằm trên đường thẳng PQ hay $M \in PQ$

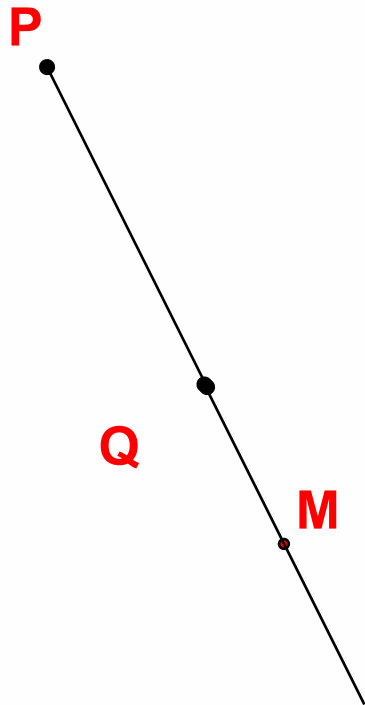
→ Khoảng cách từ M đến PQ bằng 0

$$\frac{Ax_M + By_M + C}{\sqrt{A^2 + B^2}} = 0$$

CHỌN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

❖ Về mặt toán học

- M thuộc đoạn PQ \rightarrow M thỏa 3 điều kiện sau



$$\left\{ \begin{array}{l} x_P \leq x_M \leq x_Q \\ y_P \leq y_M \leq y_Q \\ \frac{Ax_M + By_M + C}{\sqrt{A^2 + B^2}} = 0 \end{array} \right.$$

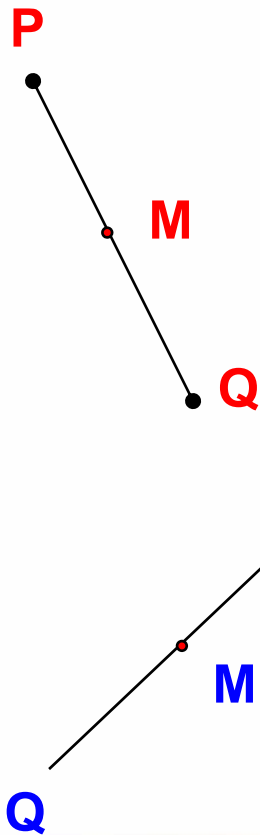
Với $Ax + By + C = 0$ (d)

là phương trình đường thẳng qua 2 điểm PQ

CHỌN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

❖ Cài đặt chương trình

- M thuộc đoạn PQ \rightarrow M thỏa 3 điều kiện sau



$$\left\{ \begin{array}{l} x_{\min} \leq x_M \leq x_{\max} \\ y_{\min} \leq y_M \leq y_{\max} \\ \frac{Ax_M + By_M + C}{\sqrt{A^2 + B^2}} \leq \varepsilon \end{array} \right.$$

Với

- $Ax + By + C = 0$ (d)

là phương trình đường thẳng qua 2 điểm PQ

- $x_{\min} = \min(x_P, x_Q)$, $x_{\max} = \max(x_P, x_Q)$

- $y_{\min} = \min(y_P, y_Q)$, $y_{\max} = \max(y_P, y_Q)$

CHỌN ĐƯỜNG THẲNG - DEMO

❖ Hệ số A, B, C của phương trình đường thẳng PQ

```
void CGDISampleView::LineEquation(CPoint point1, CPoint point2, long &A, long &B, long &C)
```

```
{  
    A = Q.y - P.y;  
    B = P.x - Q.x;  
    C = (long)P.y*Q.x - (long)P.x*Q.y;  
}
```

$$Ax + By + C = 0 \text{ (d)}$$

$$A = y_P - y_Q$$

$$B = x_Q - x_P$$

$$C = x_P * y_Q - x_Q * y_P$$

❖ Khoảng cách từ điểm M đến đường thẳng PQ

```
int CGDISampleView::LineDistance(CPoint M, CPoint P, CPoint Q)
```

```
{  
    long A,B,C;  
    LineEquation(P,Q,A,B,C);  
    double kc = abs(A*M.x+B*M.y+C)/sqrt(A*A+B*B);  
    return int(kc);  
}
```

$$\frac{Ax_M + By_M + C}{\sqrt{A^2 + B^2}}$$

CHỌN ĐƯỜNG THẲNG - DEMO

❖ Kiểm tra điều kiện $M \in PQ$

BOOL CGDISampleView::Between(CPoint M, CPoint P, CPoint Q)

```
{  
    const constDist = 5;  
    int x1,y1,x2,y2;  
    x2 = max(P.x,Q.x) + constDist;  
    x1 = min(P.x,Q.x);  
    y2 = max(P.y,Q.y) + constDist;  
    y1 = min(P.y,Q.y);  
    if ((M.x<=x2)&&(M.x>=x1)&&(M.y<=y2)&&(M.y>=y1))  
        return true;  
    return false;  
}
```

$$x_{\min} \leq x_M \leq x_{\max}$$

$$y_{\min} \leq y_M \leq y_{\max}$$

Với

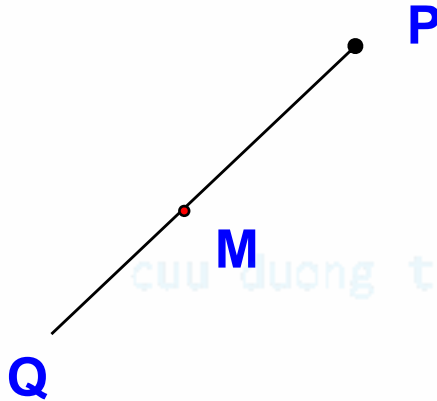
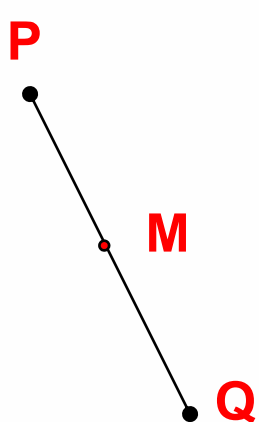
$$- x_{\min} = \min(x_P, x_Q), x_{\max} = \max(x_P, x_Q)$$

$$- y_{\min} = \min(y_P, y_Q), y_{\max} = \max(y_P, y_Q)$$

CHỌN ĐƯỜNG THẲNG - DEMO

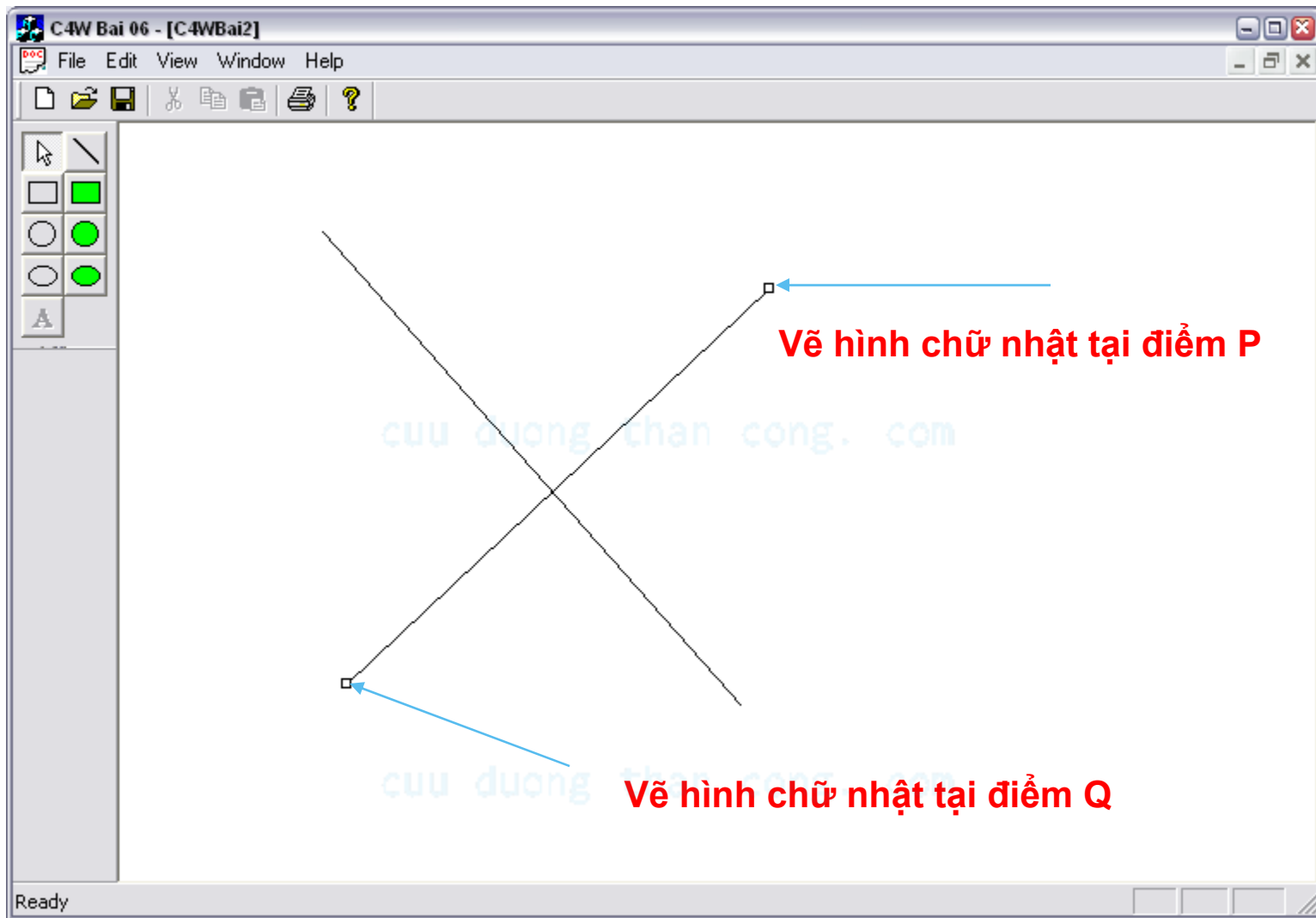
❖ Kiểm tra điểm $M \in PQ$

```
BOOL CGDISampleView::ContainsInBorder(CPoint M, CPoint P, CPoint Q)
{
    const constDist = 5;
    if ((Between(M,P,Q)) && (LineDistance(M,P,Q)<=constDist))
        return true;
    return false;
}
```



$$\left\{ \begin{array}{l} x_{\min} \leq x_M \leq x_{\max} \\ y_{\min} \leq y_M \leq y_{\max} \\ \frac{Ax_M + By_M + C}{\sqrt{A^2 + B^2}} \leq \varepsilon \end{array} \right.$$

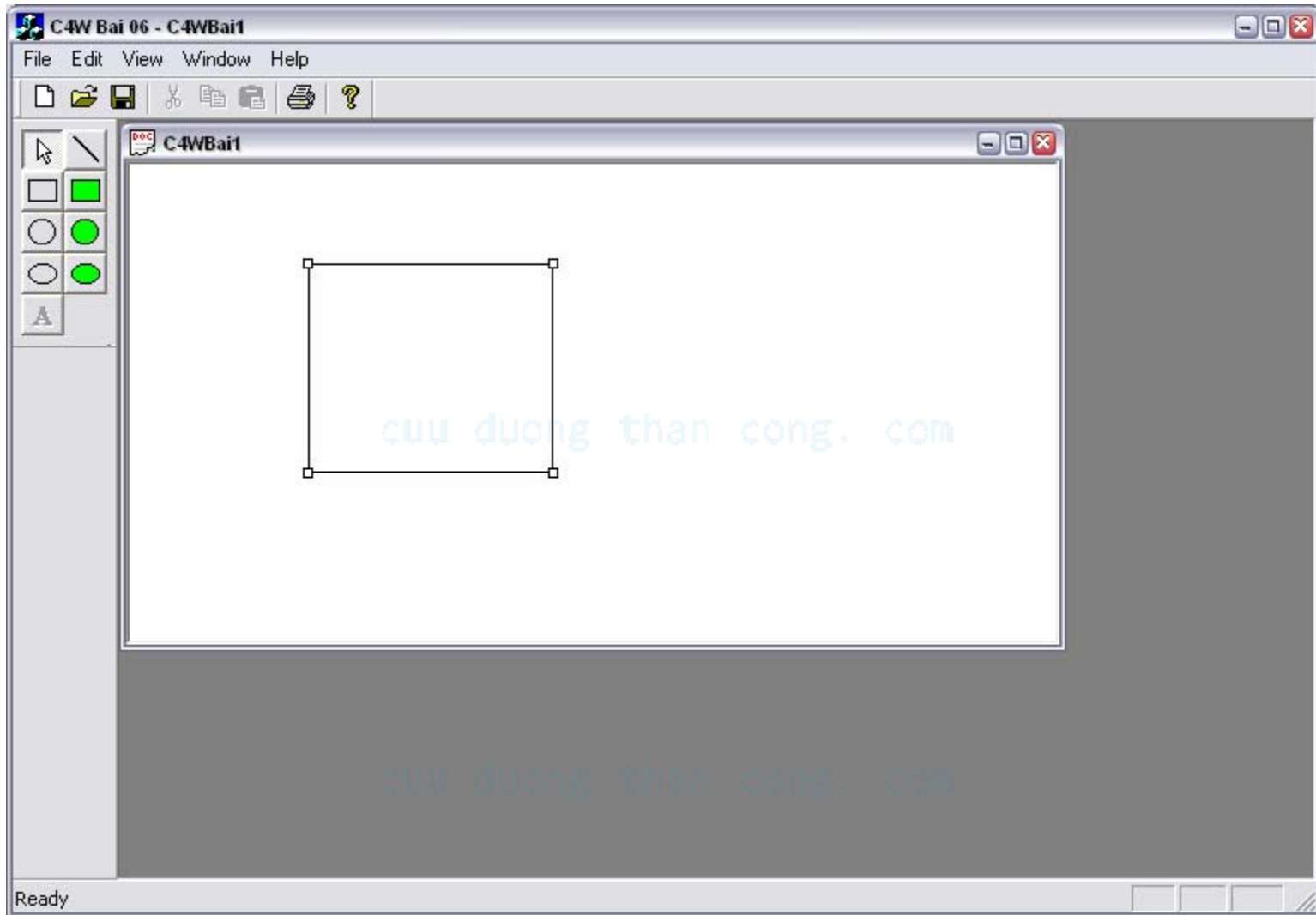
CHỌN ĐƯỜNG THẲNG - DEMO



CHỌN ĐƯỜNG THẲNG - DEMO

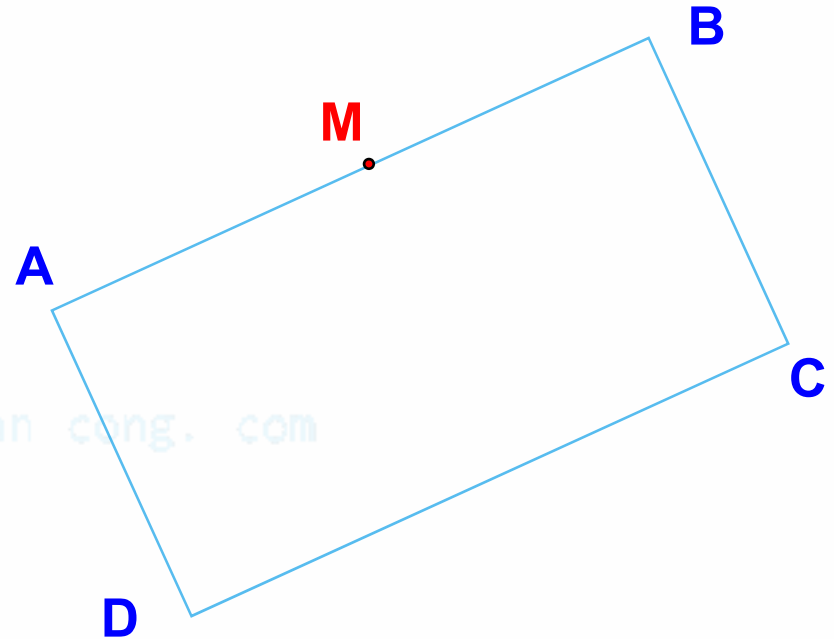
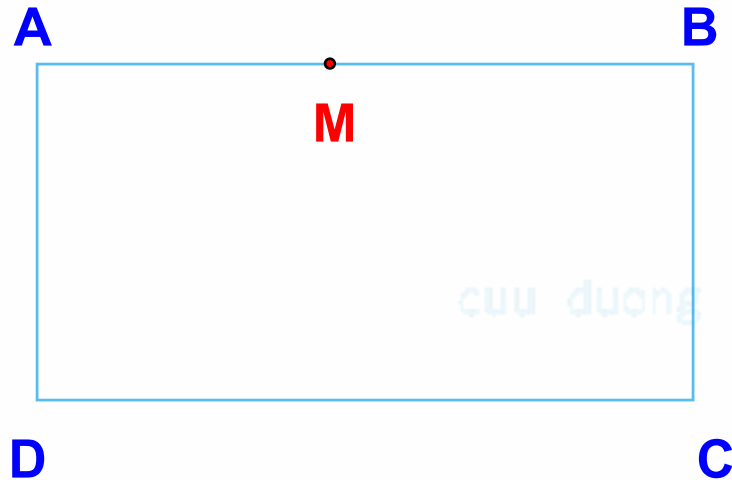
```
void CGDISampleView::ShowSelectedLine(CPoint point)
{
    CClientDC dc(this);
    CRect *ptrRect;
    const WIDTH = 4;
    int nOldMode = dc.SetROP2(R2_NOTXORPEN);
    CGdiObject *pOldBrush = dc.SelectStockObject(NULL_BRUSH);
    //Ve 2 dau cua diem chon
    CPoint p1,p2;
    p1 = ptrRect->TopLeft();
    p2 = ptrRect->BottomRight();
    dc.Rectangle(p1.x-WIDTH,p1.y-WIDTH,p1.x+WIDTH,p1.y+WIDTH);
    dc.Rectangle(p2.x-WIDTH,p2.y-WIDTH,p2.x+WIDTH,p2.y+WIDTH);
    dc.SelectObject(pOldBrush);
    dc.SetROP2(nOldMode);
    dc.MoveTo(ptrRect->TopLeft());
    dc.LineTo(ptrRect->BottomRight());
}
```

CHỌN HÌNH CHỮ NHẬT RỒI



CHỌN HÌNH CHỮ NHẬT RỒI

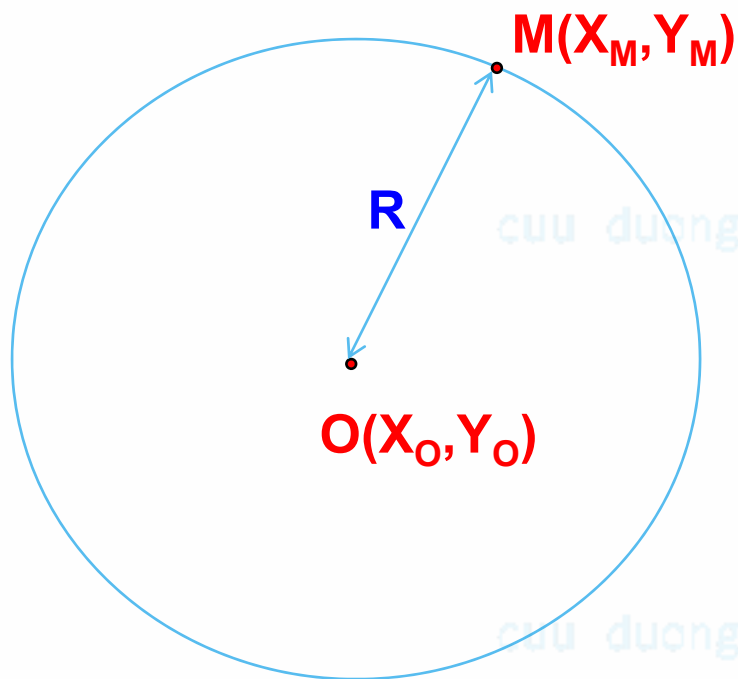
❖ Về mặt toán học



- Kiểm tra điểm $M \in AB$
- Kiểm tra điểm $M \in BC$
- Kiểm tra điểm $M \in CD$
- Kiểm tra điểm $M \in DA$

CHỌN HÌNH TRÒN

❖ Về mặt toán học



Phương trình đường tròn (C) tâm O bán kính R

$$(x - x_o)^2 + (y - y_o)^2 = R^2 \quad (c)$$

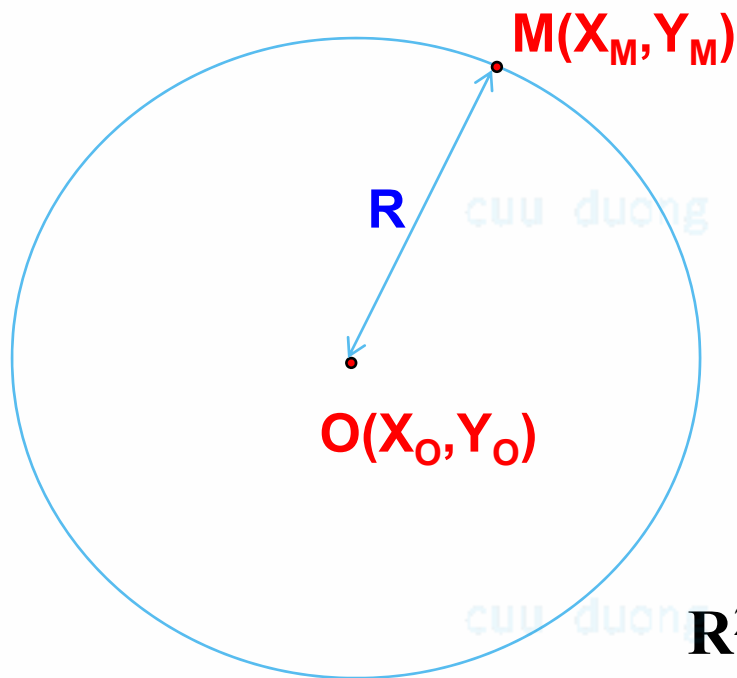
M thuộc đường tròn (C)

→ tọa độ điểm $M(x_M, y_M)$ thỏa phương trình (C)

$$(x_M - x_o)^2 + (y_M - y_o)^2 = R^2 \quad (c)$$

CHỌN HÌNH TRÒN

❖ Cài đặt chương trình



Phương trình đường tròn (C) tâm O bán kính R

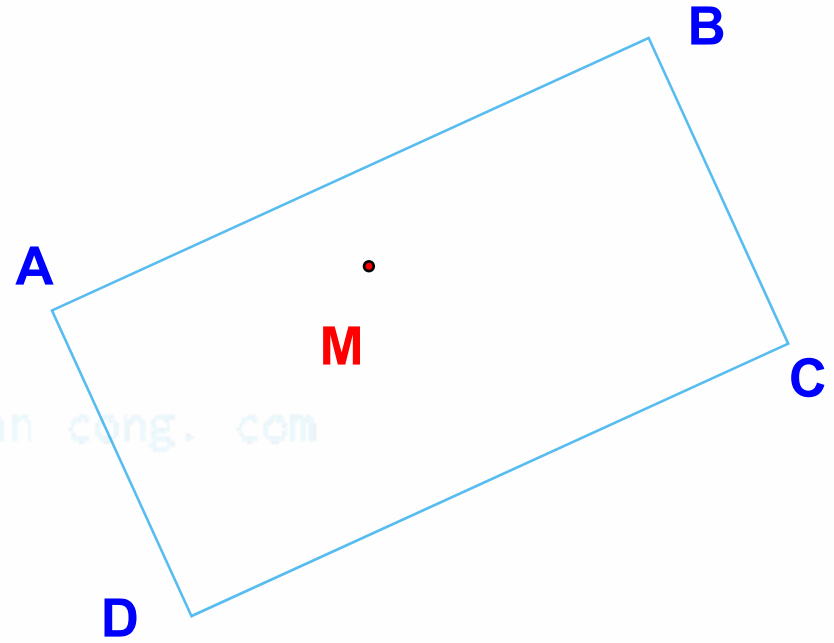
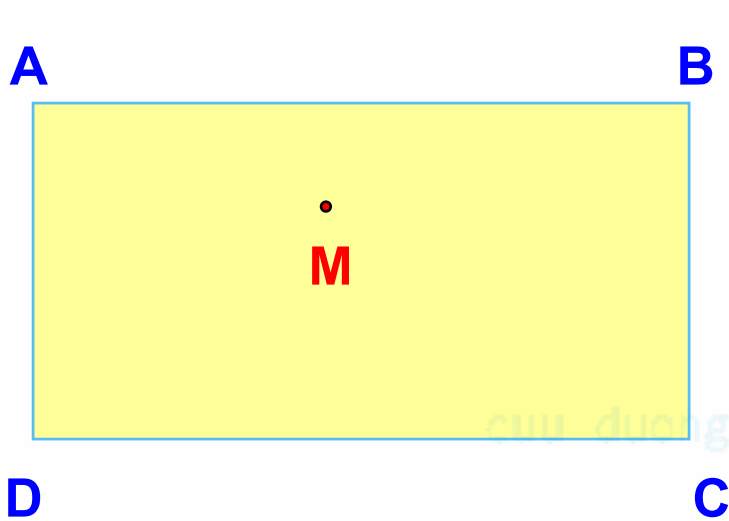
$$(x - x_o)^2 + (y - y_o)^2 = R^2 \quad (c)$$

M thuộc đường tròn (C)

→ tọa độ điểm $M(x_M, y_M)$ thỏa phương trình (C)

$$R^2 - \varepsilon \leq (x_M - x_o)^2 + (y_M - y_o)^2 \leq R^2 + \varepsilon$$

CHỌN HÌNH CHỮ NHẬT CÓ TÔ MÀU

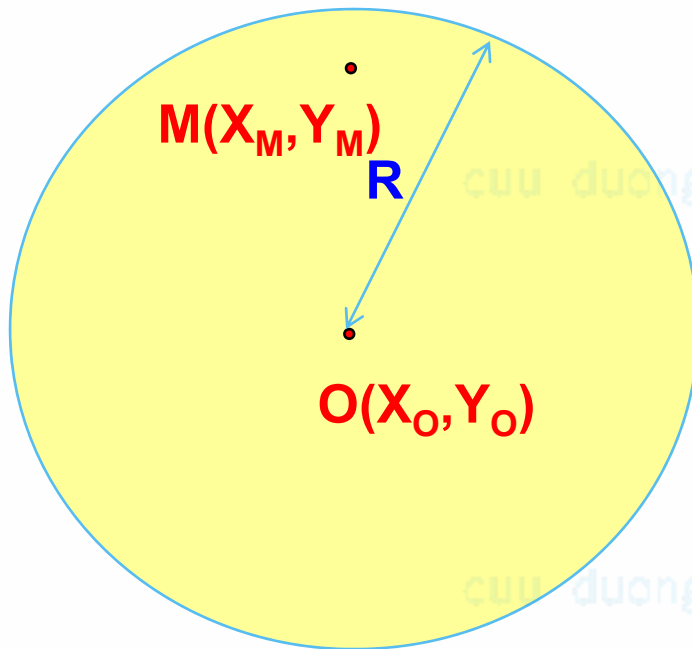


$$x_A \leq x_M \leq x_C$$

$$y_A \leq y_M \leq y_C$$

Làm thế nào để xác định được điểm M nằm trong HCN này hay không ?

CHỌN HÌNH TRÒN CÓ TÔ MÀU



Phương trình đường tròn (C) tâm O bán kính R

$$(x - x_o)^2 + (y - y_o)^2 = R^2 \quad (c)$$

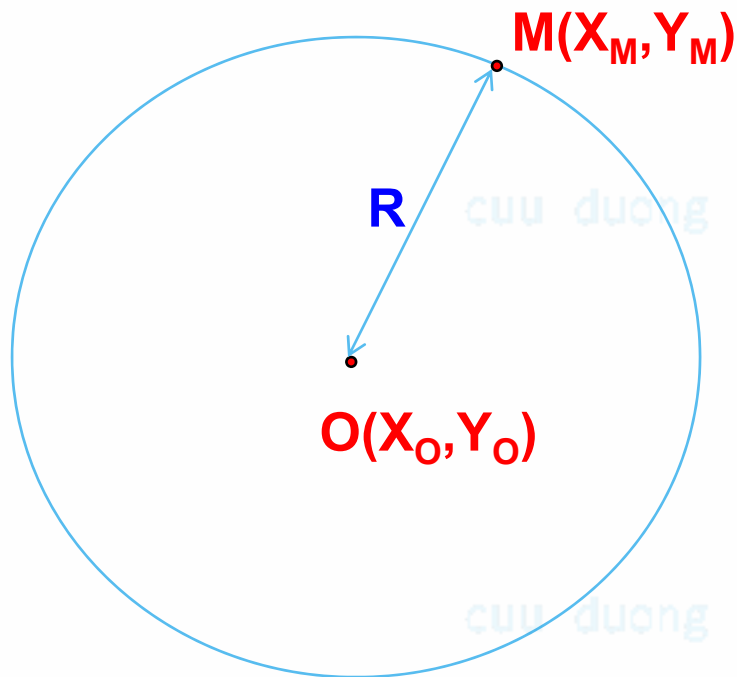
M thuộc đường tròn (C)

→ tọa độ điểm $M(x_M, y_M)$ thỏa phương trình (C)

$$(x_M - x_o)^2 + (y_M - y_o)^2 < R^2 \quad (c)$$

CHỌN HÌNH TRÒN CÓ TÔ MÀU

❖ Cài đặt chương trình



Phương trình đường tròn (C) tâm O bán kính R

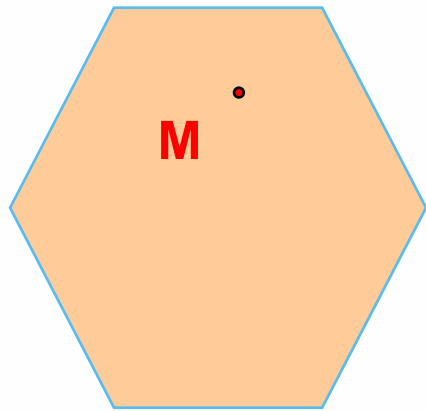
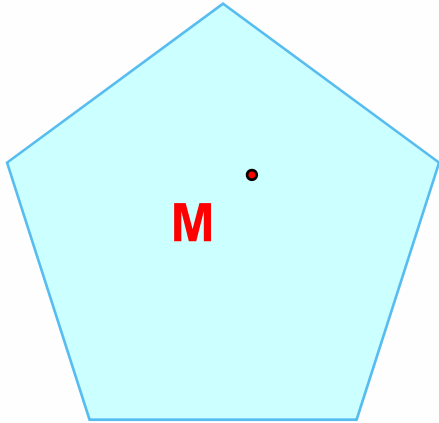
$$(x - x_o)^2 + (y - y_o)^2 = R^2 \quad (c)$$

M thuộc đường tròn (C)

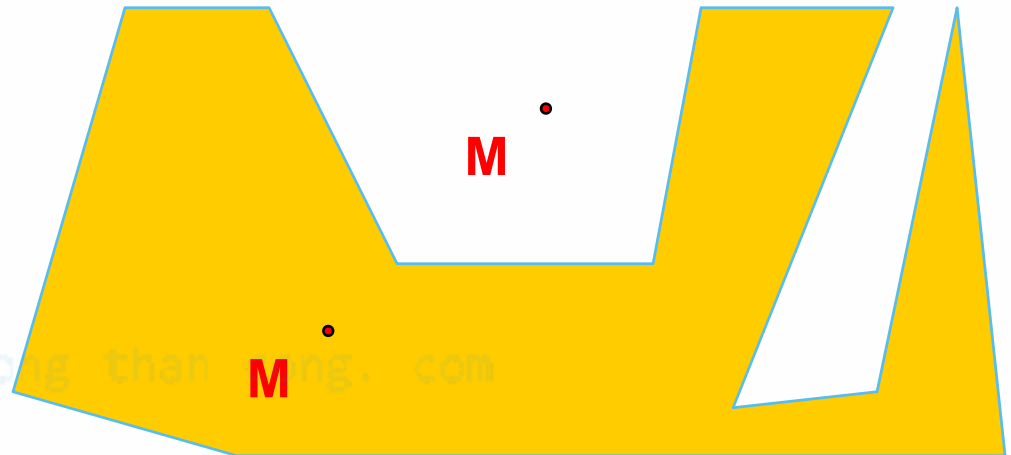
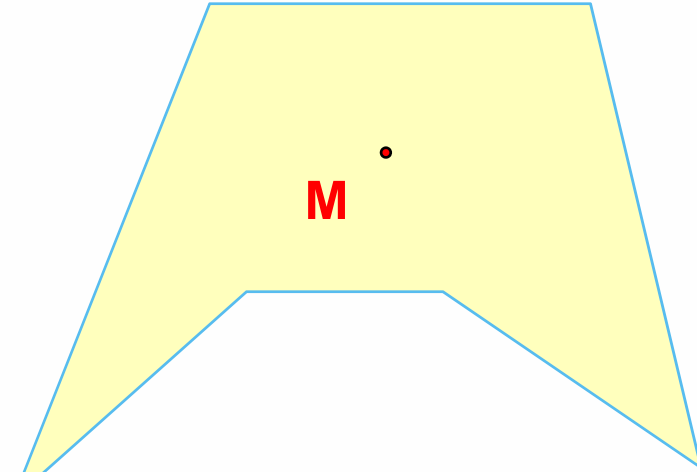
→ tọa độ điểm $M(x_M, y_M)$ thỏa phương trình (C)

$$(x_M - x_o)^2 + (y_M - y_o)^2 \leq R^2 + \varepsilon$$

CHỌN ĐA GIÁC BẤT KỲ CÓ TÔ MÀU

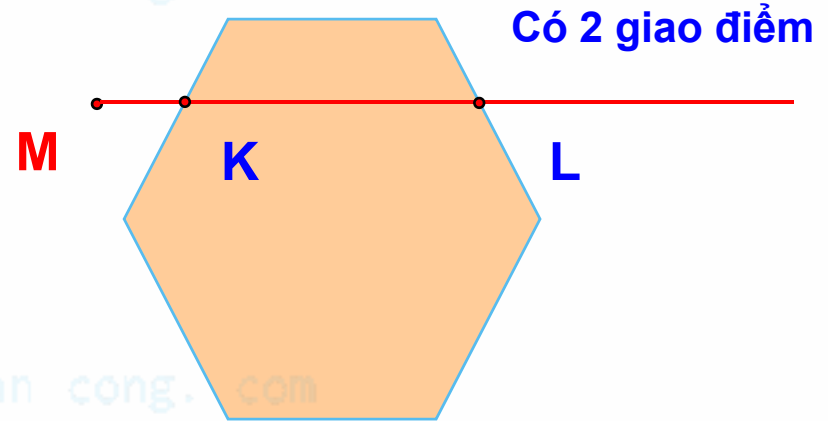
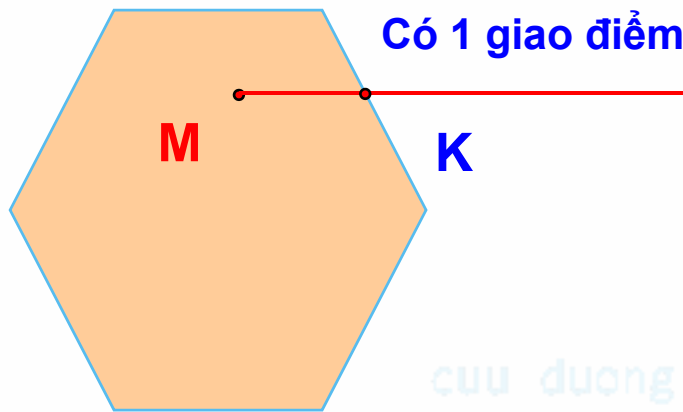
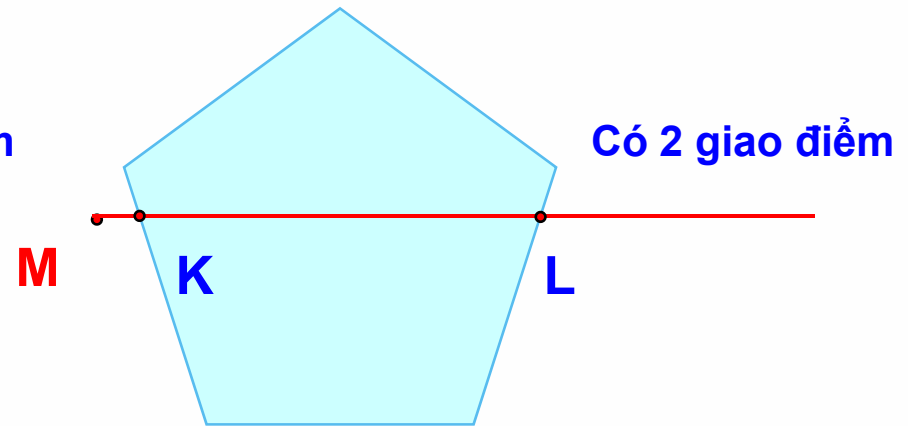
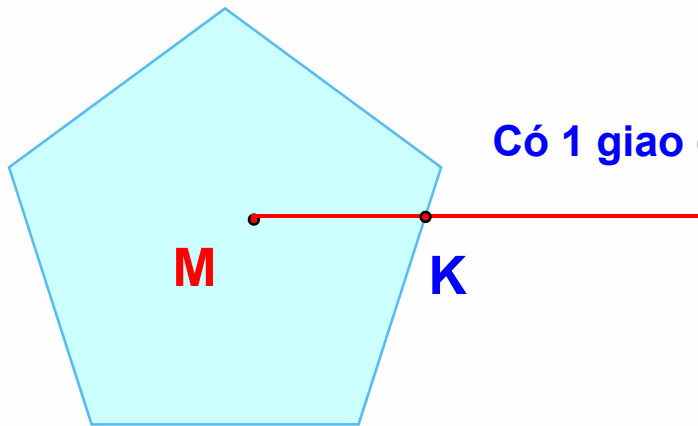


Đa giác lồi



Đa giác lõm

CHỌN ĐA GIÁC BẤT KỲ CÓ TÔ MÀU

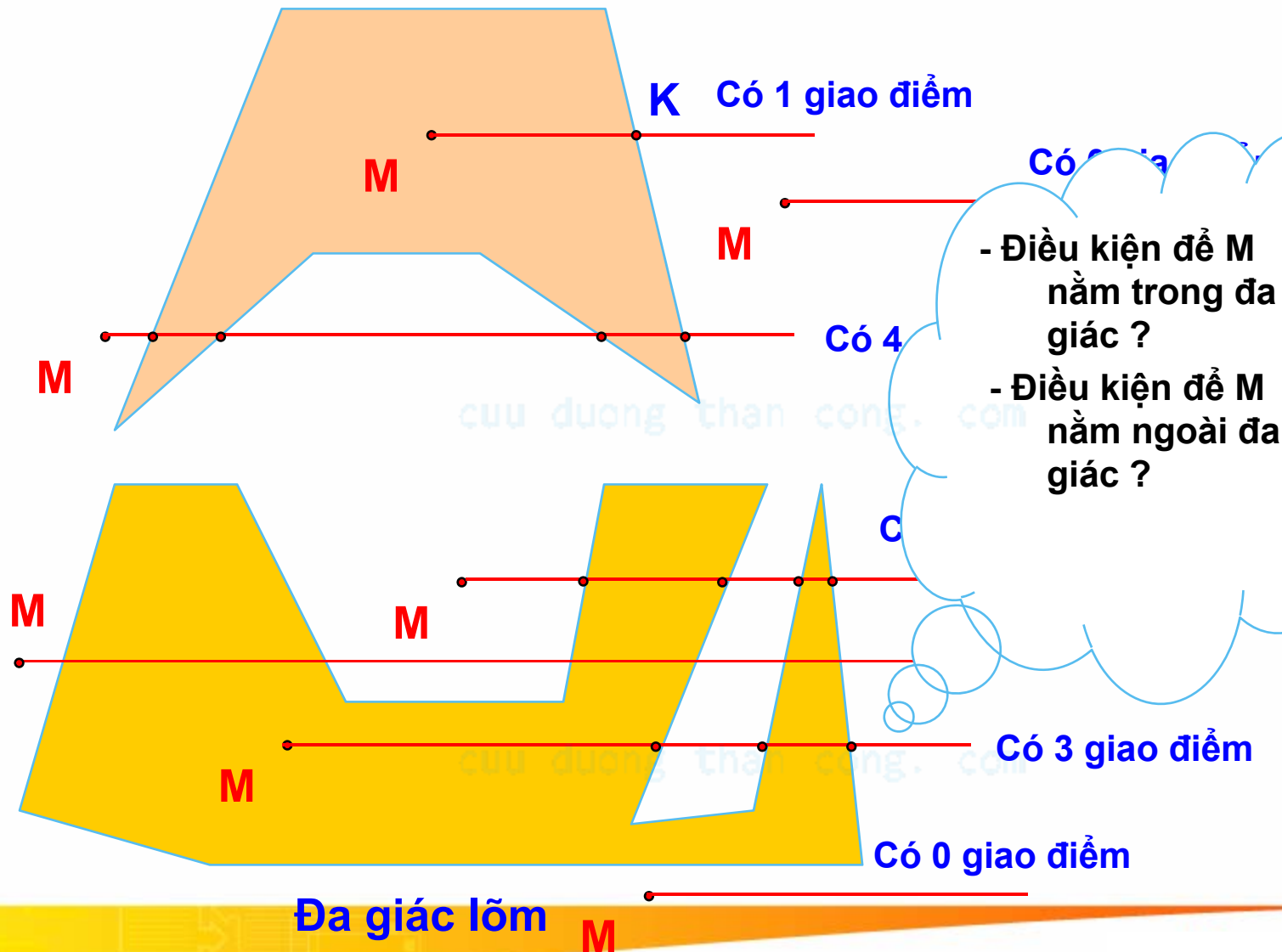


Đa giác lồi

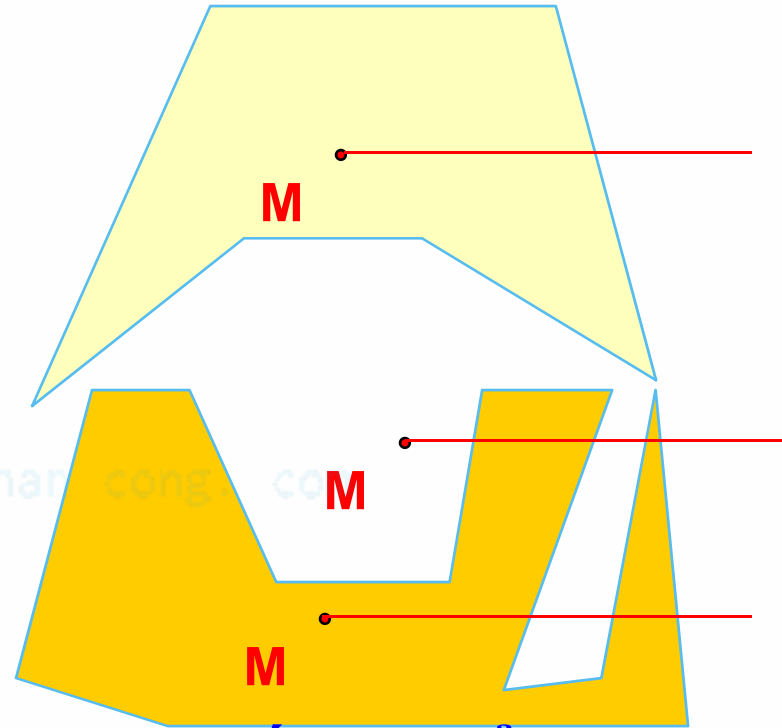
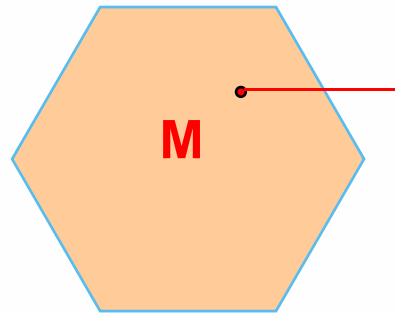
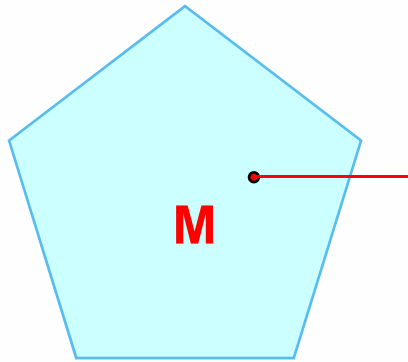
- Điều kiện để M nằm trong đa giác ?

- Điều kiện để M nằm ngoài đa giác ?

CHỌN ĐA GIÁC BẤT KỲ CÓ TÔ MÀU

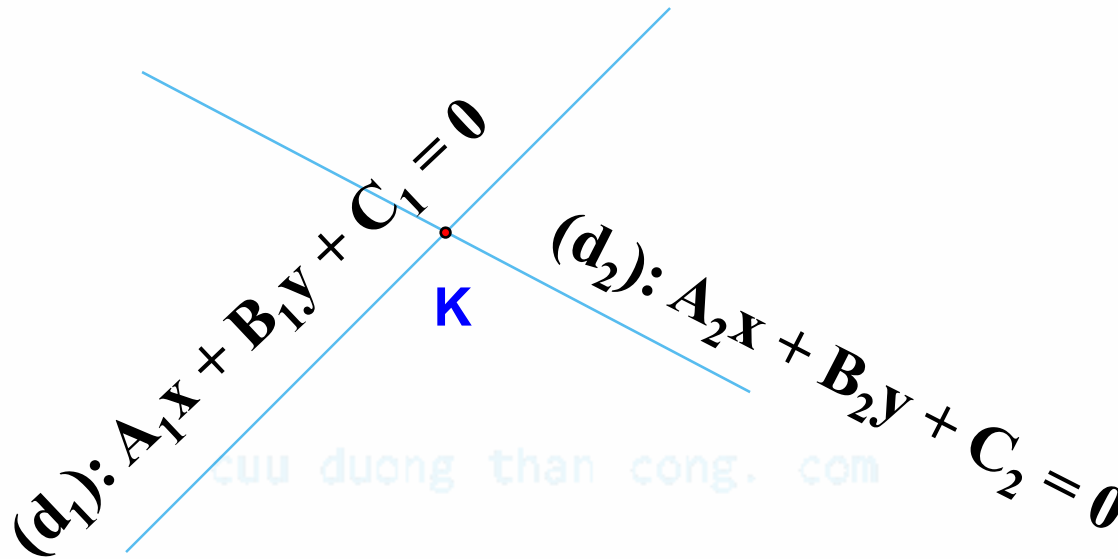


CHỌN ĐA GIÁC BẤT KỲ CÓ TÔ MÀU



- Điều kiện để M nằm trong đa giác \rightarrow số giao điểm của đường thẳng kẻ từ M đến các cạnh của đa giác là một số lẻ: 1, 3, 5,...
- Điều kiện để M nằm ngoài đa giác \rightarrow số giao điểm của đường thẳng kẻ từ M đến các cạnh của đa giác là một số chẵn: 0, 2, 4,...

XÁC ĐỊNH GIAO ĐIỂM GIỮA 2 ĐƯỜNG



Giao điểm của (d_1) và (d_2) là nghiệm của hệ phương trình

$$(d_1): A_1x + B_1y + C_1 = 0$$

$$(d_2): A_2x + B_2y + C_2 = 0$$

TÌM HIỂU ỨNG DỤNG PAINT

❖ Vẽ các đối tượng trong GDI

❖ Line

❖ Rectangle

❖ Circle

❖ Chọn đối tượng

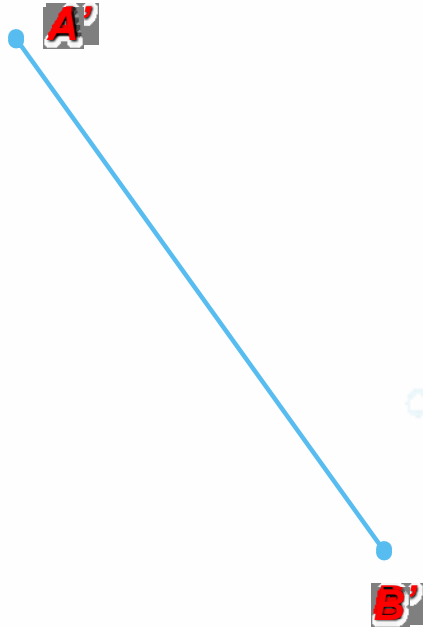
❖ Di chuyển đối tượng

❖ Lưu trữ các đối tượng

cuu duong than cong. com

cuu duong than cong. com

DI CHUYỂN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

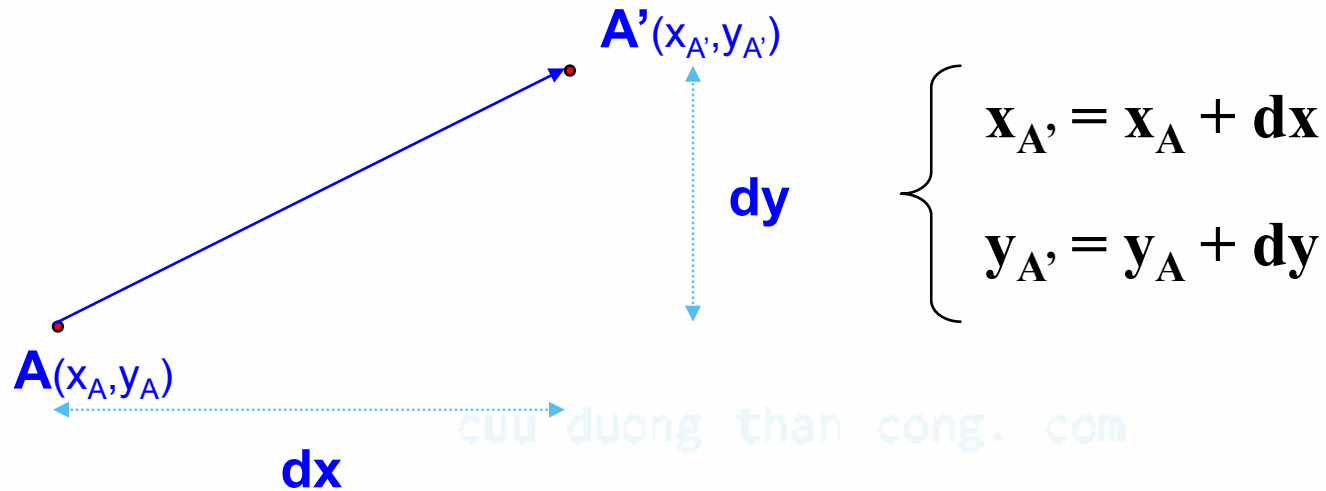


Di chuyển đường thẳng AB thực chất là phép tịnh tiến 2 điểm

AB: $A \rightarrow A'$

$B \rightarrow B'$

DI CHUYỂN ĐỐI TƯỢNG – PHÉP TÍNH TIẾN

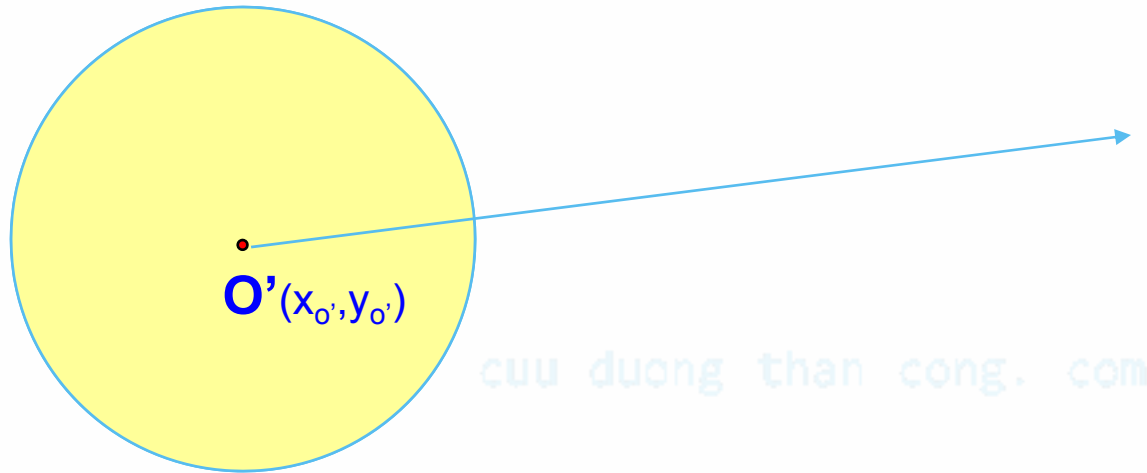


```
void TRANSLATION(CPoint &Destination,CPoint Source,int dx,int dy)
{
    Destination.x=Source.x+dx;
    Destination.y=Source.y+dy;
}
```

DI CHUYỂN ĐỐI TƯỢNG - ĐƯỜNG THẲNG

```
void CGDISampleView::MoveSelectedObject(CPoint point)
{
    ...
    CRect *ptrRect;
    ptrRect = pDoc->GetLine(m_nLineSel);
    //Xóa đường cũ
    dc.MoveTo(ptrRect->TopLeft());
    dc.LineTo(ptrRect->BottomRight());
    //Tịnh tiến đến vị trí mới
    int dx, dy;
    dx = -m_ptStartSel.x + point.x;
    dy = -m_ptStartSel.y + point.y;
    ptrRect->top = ptrRect->top + dy; ptrRect->left = ptrRect->left + dx;
    ptrRect->bottom = ptrRect->bottom + dy; ptrRect->right = ptrRect->right + dx;
    //Vẽ đường mới
    dc.MoveTo(ptrRect->TopLeft()); dc.LineTo(ptrRect->BottomRight());
    m_ptStartSel = point;
    ...
}
```

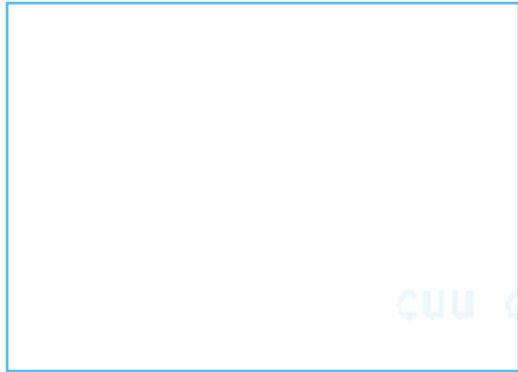
DI CHUYỂN ĐỐI TƯỢNG - ĐƯỜNG TRÒN



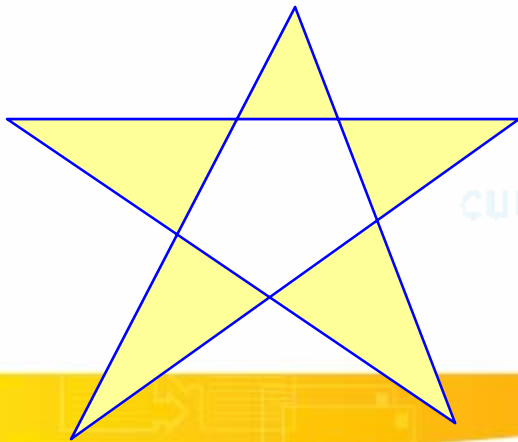
Di chuyển đường tròn tâm O bán kính R thực chất là tịnh tiến tâm O đến vị trí O'

DI CHUYỂN ĐỐI TƯỢNG

❖ Di chuyển hình chữ nhật



❖ Di chuyển đa giác



DI CHUYỂN ĐỐI TƯỢNG – CÀI ĐẶT CHƯƠNG TRÌNH

❖ Xử lý sự kiệnMouseDown

- Chọn đối tượng từ vị trí của chuột
- Lấy tọa độ điểm đầu (tọa độ của chuột)

❖ Xử lý sự kiệnMouseMove

- Tịnh tiến đối tượng đến vị trí hiện hành của chuột
- Xóa đối tượng cũ
- Vẽ đối tượng mới

❖ Xử lý sự kiệnMouseUp

- Cập nhật các thuộc tính (tọa độ) của đối tượng
- Gọi hàm Invalidate để phát sinh sự kiện OnDraw

TÌM HIỂU ỨNG DỤNG PAINT

❖ Vẽ các đối tượng trong GDI

❖ Line

❖ Rectangle

❖ Circle

❖ Chọn đối tượng

❖ Di chuyển đối tượng

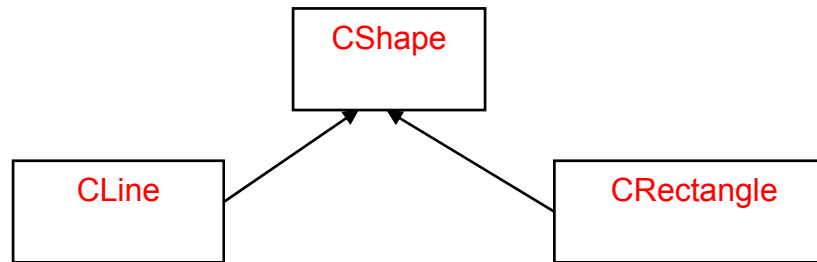
❖ Lưu trữ các đối tượng

cuu duong than cong. com

cuu duong than cong. com

QUẢN LÝ LƯU TRỮ ĐỐI TƯỢNG

Sơ đồ lớp đối tượng



cuu duong than cong. com

```
Class CShape : public CObject
{
};
```

```
Class CRectangle : public CShape
{
};

Class CLine : public CShape
{
};
```

QUẢN LÝ LƯU TRỮ ĐỐI TƯỢNG

```
Class CGDISampleDoc : public CDocument
{
    ...
    CTypedPtrList<CObList, CShape*> m_aSO;
    void Add(CShape &obj);
    ...
};
```

void CGDISampleDoc::Add(CShape &obj)

```
{
    CShape *temp;
    obj.DuplicateObject(temp);
    m_aSO.AddTail(temp);
}
```

Để sử dụng kiểu dữ liệu CTypedPtrList phải thêm vào tập tin Stdafx.h thư viện sau:

```
#include <afxtempl.h>
```

QUẢN LÝ LƯU TRỮ ĐỐI TƯỢNG

```
void CGDISampleView::OnDraw(CDC* pDC)
{
    CTNBPaintDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    //Vẽ các đối tượng: Line, Rectangle
    POSITION pos=pDoc->m_aSO.GetHeadPosition();
    while (pos != NULL)
    {
        CShape *myobj = pDoc->m_aSO.GetNext(pos);
        myobj->Show(pDC,R2_COPYPEN);
    }
}
```

**Làm thế nào để myobj gọi hàm Show tương ứng của từng đối tượng:
CShape, Cline, CRectangle,...?**

LỚP ĐỐI TƯỢNG CShape

❖ Thuộc tính đối tượng Shape

```
class CShape : public CObject
{
public:
    BOOL    m_bFilled;        // Đối tượng có được tô hay không
    int     m_nPenWidth;       // Bề rộng của bút vẽ
    int     m_nPenStyle;       // Kiểu bút vẽ (SOLID,DOT,DASH,...)
    COLORREF m_BrushColor;     // Màu tô
    COLORREF m_PenColor;       // Màu đường bao
    int     m_nPattern;        // Pattern của đối tượng
    CShape();
    virtual ~CShape();
};
```

LỚP ĐỐI TƯỢNG CShape

❖ Các hàm của lớp CShape

```
class CShape : public CObject
{
public:
    virtual void Draw(CDC * pDC);
    virtual void DuplicateObject(CShape * & pObj);
    virtual void Fill(CDC * pDC);

public:
    void Show(CDC * pDC, int nDrawMode);
    void operator    =(CShape &s);
    CShape();
    virtual ~CShape();
};
```


LỚP ĐỐI TƯỢNG CShape

❖ Các hàm của lớp CShape

```
void CShape::DuplicateObject(CShape *&pObj)
```

```
{
```

```
    pObj=new CShape();
```

```
    pObj->operator=(*this);
```

```
}
```

```
void CShape::operator =(CShape & s)
```

```
{
```

```
    m_BrushColor = s.m_BrushColor;
```

```
    m_bFilled = s.m_bFilled;
```

```
    m_PenColor = s.m_PenColor;
```

```
    m_nPenStyle = s.m_nPenStyle;
```

```
    m_nPenWidth = s.m_nPenWidth;
```

```
    m_nPattern=s.m_nPattern;
```

```
}
```

LỚP ĐỐI TƯỢNG CShape

❖ Các hàm của lớp CShape

```
CShape::CShape()  
{  
    m_PenColor = RGB(0,0,0);  
    m_BrushColor = RGB(0,128,0);  
    m_nPenStyle = PS_SOLID;  
    m_nPenWidth = 1;  
    m_bFilled = false;  
    m_nPattern=HS_SOLID;  
}
```

```
void CShape::Draw(CDC *pDC)  
{  
}  
void CShape::Fill(CDC *pDC)  
{  
}
```

cuu duong than cong. com

LỚP ĐỐI TƯỢNG CLine

❖ Các thành phần của lớp CLine

```
class CLine : public CShape
{
public:
    CPoint point2;
    CPoint point1;
public:
    void Draw(CDC * pDC);
    void operator =(CLine &l);
    void DuplicateObject(CShape*& pObj);
    void SetPoint(int nIndex, CPoint p);
    CLine();
    CLine(CPoint p1,CPoint p2);
    virtual ~CLine();
};
```

LỚP ĐỐI TƯỢNG CLine

❖ Các thành phần của lớp CLine

```
CLine::CLine()
```

```
{
```

```
    m_PenColor = RGB(255,0,0);
```

```
    m_BrushColor = RGB(0,128,0);
```

```
    m_nPenStyle = PS_SOLID;
```

```
    m_nPenWidth = 2;
```

```
    m_bFilled = false;
```

```
    m_nPattern=HS_SOLID;
```

```
}
```

```
CLine::CLine(CPoint p1,CPoint p2)
```

```
{
```

```
    point1 = p1;
```

```
    point2 = p2;
```

```
}
```

LỚP ĐỐI TƯỢNG CLine

❖ Các thành phần của lớp CLine

```
void CLine::DuplicateObject(CShape * & pObj)
```

```
{
```

```
    //Ham tao mot doi tuong moi
```

```
    pObj=new CLine();
```

```
    ((CLine*)pObj)->operator=(*this);
```

```
}
```

```
void CLine::operator =(CLine & l)
```

```
{
```

```
    CShape::operator =(l);
```

```
    point1=l.point1;
```

```
    point2=l.point2;
```

```
}
```

LỚP ĐỐI TƯỢNG CLine

❖ Các thành phần của lớp CLine

```
void CLine::Draw(CDC *pDC)
{
    //Tao mot pen moi
    CPen newPen(m_nPenStyle,m_nPenWidth,m_PenColor);
    CPen *pOldPen = pDC->SelectObject(&newPen);
    //Ve duong thang
    pDC->MoveTo(point1.x,point1.y);
    pDC->LineTo(point2.x,point2.y);

    pDC->SelectObject(pOldPen);
}
```

LỚP ĐỐI TƯỢNG CRectangle

❖ Các thành phần của lớp CRectangle

```
class CRectangle : public CShape
```

```
{
```

```
    public:
```

```
        CPoint point2;
```

```
        CPoint point1;
```

```
    public:
```

```
        void operator =(CRectangle & l);
```

```
        void DuplicateObject(CShape * & pObj);
```

```
        void Draw(CDC * pDC);
```

```
        void SetPoint(int nIndex, CPoint p);
```

```
        CRectangle();
```

```
        virtual ~CRectangle();
```

```
};
```

LỚP ĐỐI TƯỢNG CRectangle

❖ Các thành phần của lớp CRectangle

```
void CRectangle::operator =(CRectangle & l)
{
    CShape::operator =(l);
    point1=l.point1;
    point2=l.point2;
}

void CRectangle::DuplicateObject(CShape * & pObj)
{
    //Ham tao mot doi tuong moi
    pObj=new CRectangle();
    ((CRectangle*)pObj)->operator=(*this);
}
```


LỚP ĐỐI TƯỢNG CRectangle

❖ Các thành phần của lớp CRectangle

```
void CRectangle::Draw(CDC * pDC)
```

```
{
```

```
    CPen newPen(m_nPenStyle,m_nPenWidth,m_PenColor);
```

```
    CPen *pOldPen = pDC->SelectObject(&newPen);
```

```
    //Ve hình chu nhật
```

```
    pDC->Rectangle(point1.x,point1.y,point2.x,point2.y);
```

```
    pDC->SelectObject(pOldPen);
```

```
}
```

```
void CRectangle::Fill(CDC * pDC)
```

```
{
```

```
}
```

TẠO CÁC ĐỐI TƯỢNG

```
Class CGDISampleDoc : public CDocument
{
    CTypedPtrList<CObList, CShape*> m_aSO;
    void Add(CShape &obj);
};
```

❖ Tạo & vẽ đường thẳng CLine

```
void CGDISampleView::OnLine()
{
    CGDISampleDoc* pDoc = GetDocument();
    CPoint point1(100,100);
    CPoint point2(200,200);
    line.SetPoint(1,point1);
    line.SetPoint(2,point2);
    pDoc->Add(line);
    Invalidate();
}
```

TẠO CÁC ĐỐI TƯỢNG

```
Class CGDISampleDoc : public CDocument
{
    CTypedPtrList<CObList, CShape*> m_aSO;
    void Add(CShape &obj);
};
```

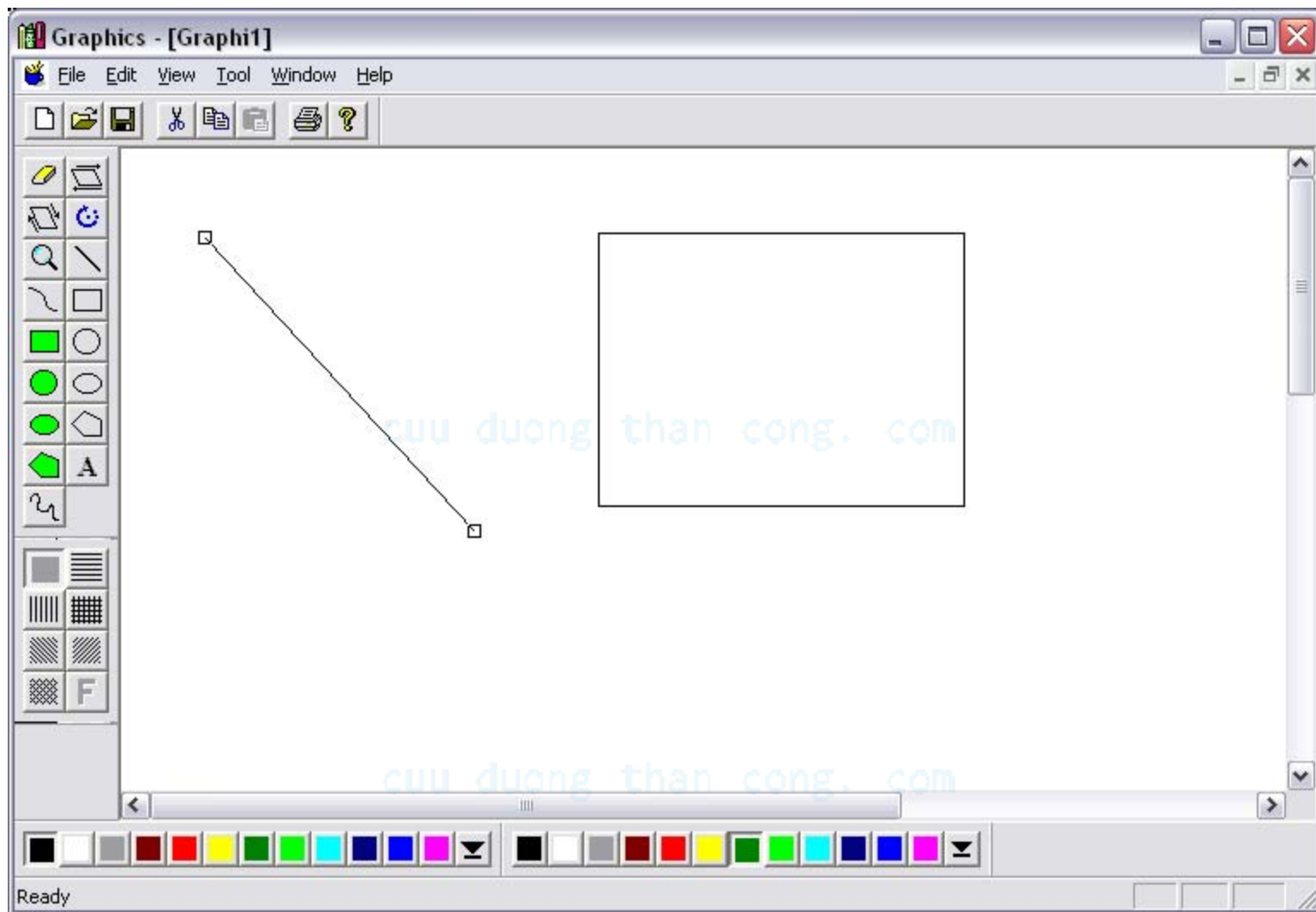
❖ Tạo & vẽ hình chữ nhật CRectangle

```
void CGDISampleView::OnRectangle()
{
    CGDISampleDoc * pDoc = GetDocument();
    CPoint point1(300,300);
    CPoint point2(400,400);
    rectangle.SetPoint(1,point1);
    rectangle.SetPoint(2,point2);
    pDoc->Add(rectangle);
    Invalidate();
}
```

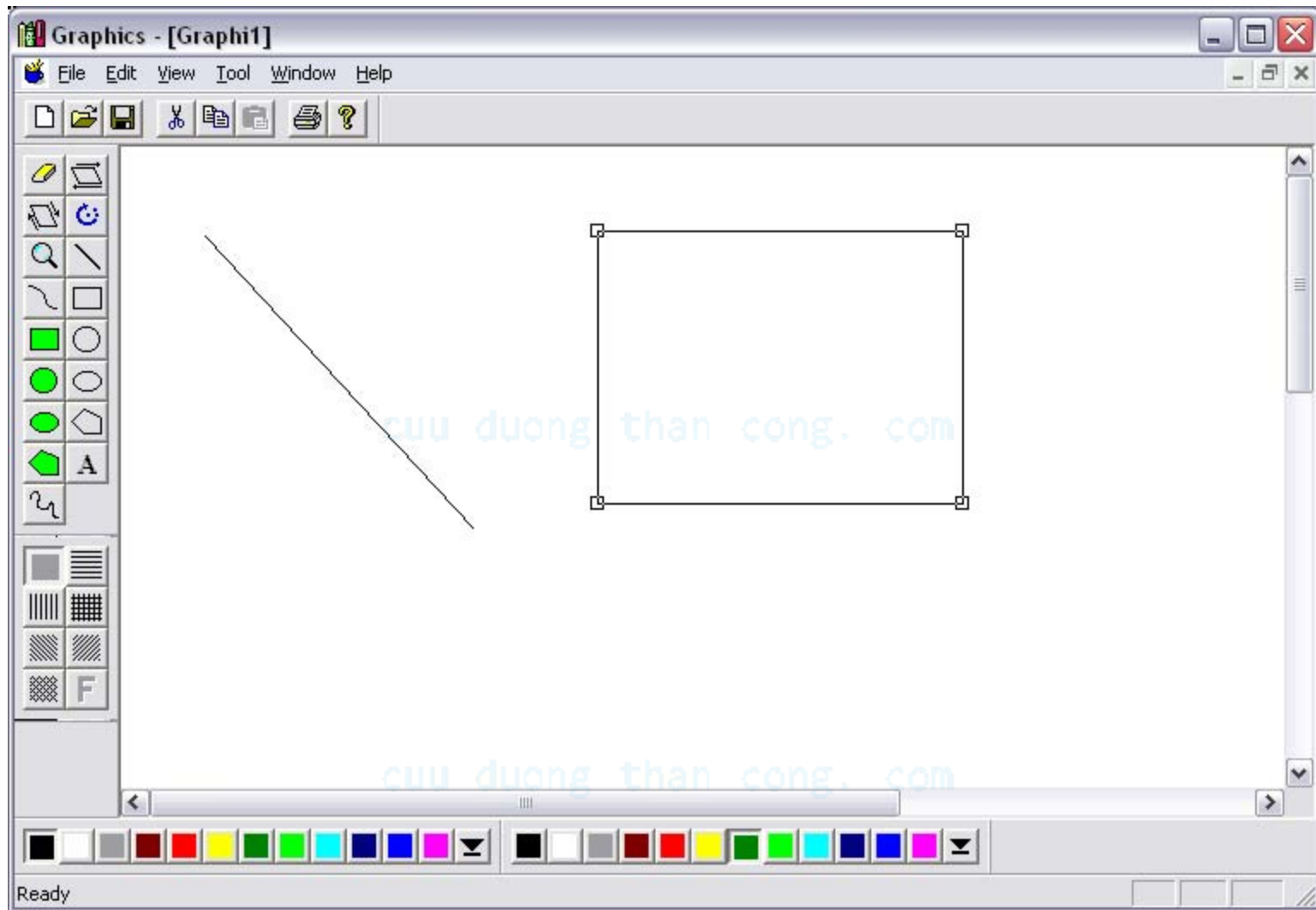
TÌM HIỂU ỨNG DỤNG PAINT

- ❖ Vẽ các đối tượng trong GDI
 - ❖ Line
 - ❖ Rectangle
 - ❖ Circle
- ❖ Chọn đối tượng
- ❖ Di chuyển đối tượng
- ❖ Lưu trữ các đối tượng
- ❖ Các xử lý khác
 - ❖ Resize
 - ❖ Cut, Copy, Paste

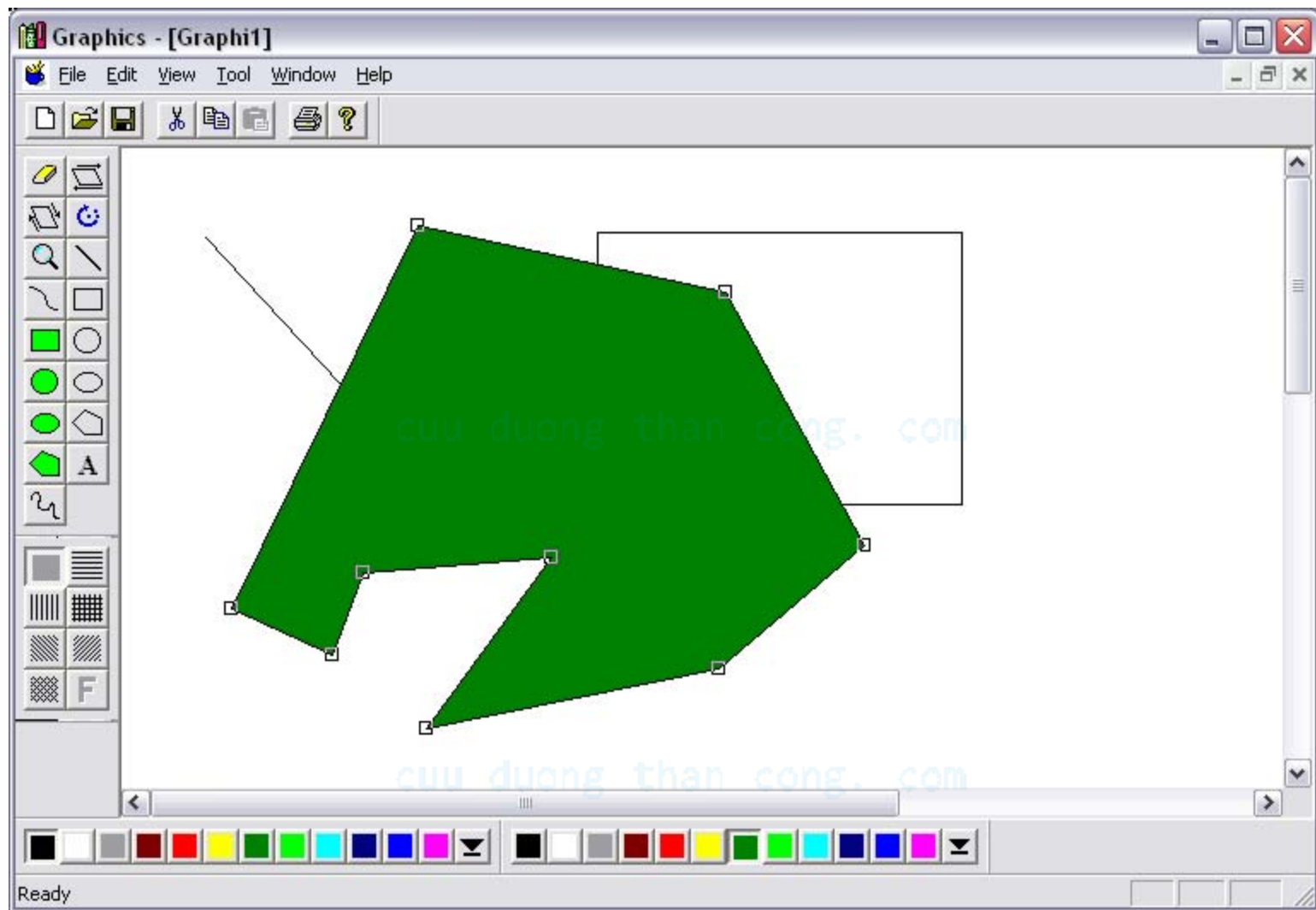
Các thao tác khác - Resize



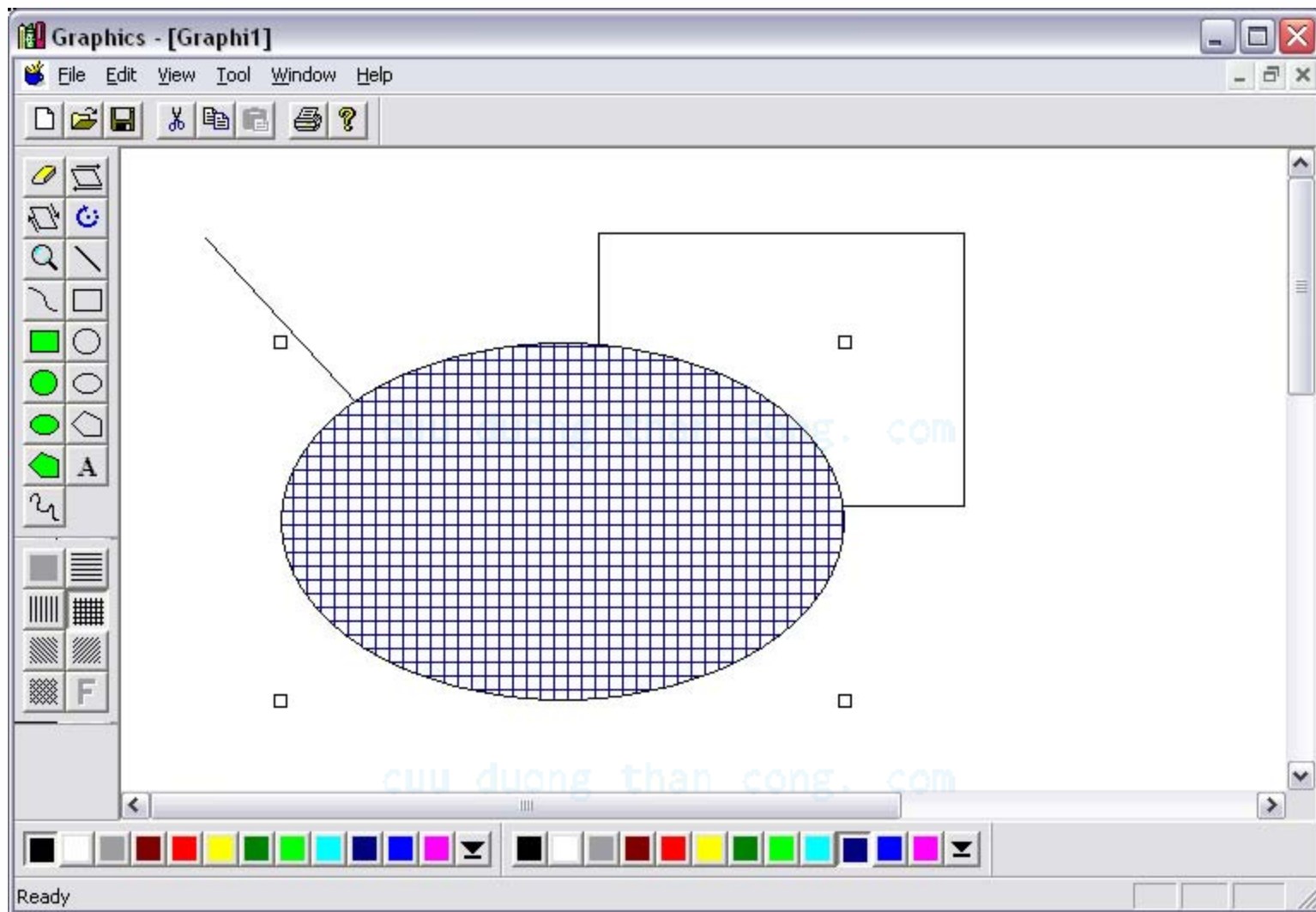
Các thao tác khác - Resize



Các thao tác khác - Resize

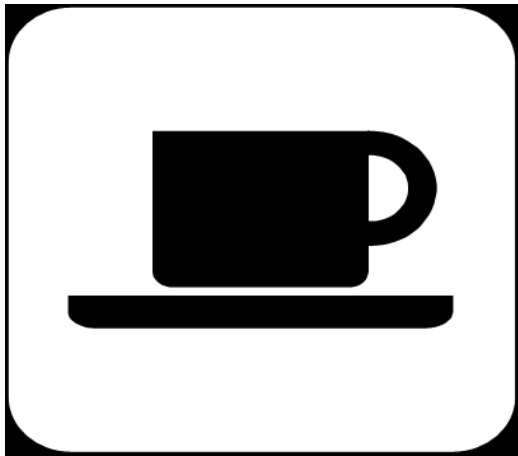


Các thao tác khác - Resize





cuu duong nhan cong. com



cuu duong nhan cong. com



Lập trình cơ sở dữ liệu với thư viện ADO

(ADO – ACTIVEX DATA OBJECT)

- Tổng quan về lập trình CSDL
- Thư viện ADO
- Lập trình CSDL với ADO

tnbao.dhsp@gmail.com

Lập trình cơ sở dữ liệu với thư viện ADO (ADO – ACTIVEX DATA OBJECT)

- **Tổng quan về lập trình CSDL**
- **Thư viện ADO**
- **Lập trình CSDL với ADO**

Ứng dụng quản lý

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH CSDL VỚI ADO

- Hệ thống quản lý học sinh: lưu trữ thông tin liên quan đến học sinh
- Hệ thống quản lý sinh viên: lưu trữ thông tin liên quan đến sinh viên
- Hệ thống quản lý nhân sự: lưu trữ thông tin liên quan đến nhân viên trong công ty
- Hệ thống quản lý tiền lương
- Hệ thống quản lý kế toán
- Hệ thống quản trị dự án
-



Ứng dụng Windows (Windows Application)

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS LẬP TRÌNH CSDL VỚI ADO

Hệ thống quản lý học sinh

Tổ chức Quản lý Báo cáo Hệ thống Trợ giúp Thoát

Niên khóa: 2000-2001 ...
Lớp học: 1 ...

Danh sách học sinh:
Tổng số HS: 1 – 1 Nữ

Số danh bộ	Họ HS	Tên HS
AK0001	Phan Thị Anh	Khanh

Cá Nhân Học Tập Khen thưởng Kỷ luật

Họ và tên: Phan Thị Anh Khanh Giới tính: N+
Ngày sinh: 02/09/197 Nơi sinh: Đà Lạt
Diện ưu tiên: Con thương binh Quê quán: TP Hồ Chí Minh
Dân tộc: Hoa Tôn giáo: Thiên chúa giáo

ĐC thường trú: 73/12 Phan Đình Phùng
Tỉnh thành: TP Hồ Chí Minh Quận/Huyện: Bình Thạnh
ĐC tạm trú: 182 Trần Bình Trọng
Tỉnh thành: TP Hồ Chí Minh Quận/Huyện: Bình Thạnh
Diện lưu trú: Tạm trú Email: ptakhanh@huongsen.co
Điện thoại: 8383962 Nhắn tin:

Họ tên cha: Phan Văn X Nghề nghiệp: Nghỉ hưu
Họ tên mẹ: Nguyễn Thị Y Nghề nghiệp: Nghỉ hưu
Họ tên NĐD: Nghề nghiệp:

Cập nhật Đồng ý Hủy Thoát

Thông tin học sinh được lưu trữ như thế nào ?

Ứng dụng Windows (Windows Application)

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS LẬP TRÌNH CSDL VỚI ADO

Danh mục nhân viên

Thông tin nhân viên

Mã NV: NV006 Họ tên: Trần Thị Mỹ Châu

Địa chỉ: 26 Nguyễn Bình Khiêm - Quận 1

Điện thoại: 0913670277 Email:

Ghi chú:

Danh sách nhân viên

Mã NV	Họ Tên	Địa chỉ	Điện thoại	Email
NV001	Lê Đức Long	297 Lê Quý Đôn - P.8 - Quận 3	8500347	ldlong@hcm.vnn.vn
NV002	Nguyễn Công Phú	397 Lê Quang Định - P.5 - Q. Bình Thạnh	8123567	ncphu@yahoo.com
NV003	Nguyễn Lương Anh Tuấn	153/2 Hoàng Văn Thụ - Q. Tân Bình	8723567	ltuan@yahoo.com
NV004	Cao Thị Tố Trinh	24 Nguyễn Thị Minh Khai - Q.3	9345882	ctttrinh@yahoo.com
NV005	Lý Thành	123 Trương Định - Quận 3	9321213	lthanh@yahoo.com
NV006	Trần Thị Mỹ Châu	26 Nguyễn Bình Khiêm - Quận 1	0913670277	
NV007	Trần Thị Minh Nguyệt	32 Trần Bình Trọng - Quận 5		

Thêm Xóa Sửa Ghi Không Thoát

Thông tin nhân viên được lưu trữ như thế nào ?



Mẫu biểu trong các ứng dụng

ỦY BAN NHÂN DÂN TP.HCM
PHÒNG THANH TRA
153/2 Hoàng Văn Thụ
ĐT: 9321213
-----oOo-----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự Do - Hạnh Phúc
-----oOo-----

TP.Hồ Chí Minh, ngày tháng năm

BÁO CÁO DANH SÁCH HỒ SƠ CHƯA GIẢI QUYẾT

Loại hồ sơ : Dân sự

Từ ngày : 11/12/2002 Đến ngày : 13/12/2002

Ngày	Số hồ sơ	Người gửi	Địa chủ	Điện thoại
Quận : Bình Thạnh		Số lượng hồ sơ : 3		
11/12/2002	HSDS100	Bùi Văn Lý	123 Phan Đăng Lưu	0908123456
12/12/2002	HSDS101	Nguyễn Công Phú	126 Lê Quang Định	
14/12/2002	HSDS130	Nguyễn Văn Hùng	345 Phan Văn Trị	
Quận : Phú Nhuận		Số lượng hồ sơ : 3		
11/12/2002	HSDS100	Bùi Xuân Giang	96 Huỳnh Văn Bánh	9321345-9632145
12/12/2002	HSDS101	Trịnh Minh Thành	126 Hai Bà Trưng	8245678
14/12/2002	HSDS130	Nguyễn Niềm Thoại Uyên	165 Nguyễn Kiệm	8456012 - 0903122345
Tổng cộng: 2 quận		Số lượng hồ sơ : 6		

Người lập

Trưởng phòng thanh tra

Thông tin thể hiện trên báo cáo được lấy từ đâu ?



Mẫu biểu trong các ứng dụng

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH CSDL VỚI ADO

Trường Cao Đẳng SP Bình Dương

Hồ Sơ nhân viên

	Mã nhân viên	2
	Tên nhân viên	Fuller Andrew
	Địa chỉ	908 W. Capital Way
	Điện thoại	(206) 555-9482

Thông tin thể hiện trên báo cáo được lấy từ đâu ?



Dữ liệu trong các ứng dụng quản lý

Thông tin học sinh được lưu trữ như thế nào ?



Tất cả thông tin liên quan đến học sinh như: Lý lịch học sinh, điểm thi, chuyên cần, lớp học, khối.... Được lưu trữ trong CƠ SỞ DỮ LIỆU

Thông tin nhân viên được lưu trữ như thế nào ?



Tất cả thông tin liên quan đến nhân viên như: Lý lịch nhân viên, quá trình công tác, quá trình lương, khen thưởng, kỷ luật....Được lưu trữ trong CƠ SỞ DỮ LIỆU

Thông tin thể hiện trên báo cáo được lấy từ đâu ?



Tất cả thông tin thể hiện trên báo cáo được lấy từ CƠ SỞ DỮ LIỆU



Tương tác dữ liệu: thêm, xóa,...

Danh mục giáo viên

Thông tin giáo viên

Mã GV: NV001 Họ tên: Lê Đức Long

Địa chỉ: 297 Lê Quý Đôn - P.8 - Quận 3

Điện thoại: 8500347 Email: ldlong@hcm.vnn.vn

Ghi chú:

Danh sách giáo viên

Mã GV	Họ Tên	Địa chỉ	Điện thoại	Email
NV001	Lê Đức Long	297 Lê Quý Đôn - P.8 - Quận 3	8500347	ldlong@hcm.vnn.vn
NV002	Nguyễn Công Phú	397 Lê Quang Định - P.5 - Q. Bình Thạnh	8123567	ncphu@yahoo.com
NV003	Nguyễn Lương Anh Tuấn	153/2 Hoàng Văn Thụ - Q. Tân Bình	8723567	ltuan@yahoo.com
NV004	Cao Thị Tố Trinh	24 Nguyễn Thị Minh Khai - Q.3	9345882	ctttrinh@yahoo.com
NV005	Lý Thành	123 Trương Định - Quận 3	9321213	lthanh@yahoo.com
NV006	Trần Thị Mỹ Châu	26 Nguyễn Bỉnh Khiêm - Quận 1	0913670277	
NV007	Trần Thị Minh Nguyệt	32 Trần Bình Trọng - Quận 5		

Thêm Xóa Sửa Ghi Không Thoát

Lấy dữ liệu

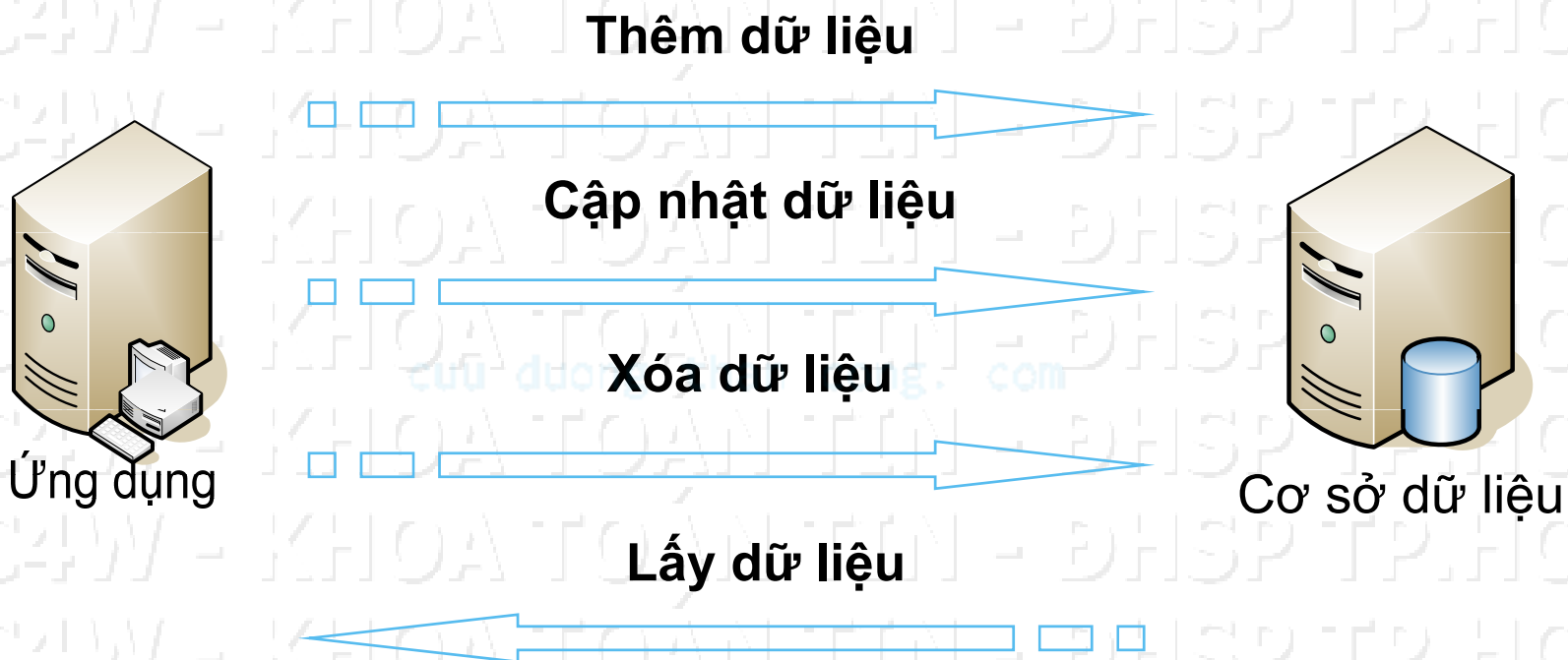
Thêm dữ liệu

Xóa dữ liệu

Sửa dữ liệu



Tương tác dữ liệu: thêm, xóa,...



Tất cả các thao tác THÊM, XÓA, SỬA, TRUY VẤN dữ liệu đều liên quan đến ngôn SQL: INSERT, DELETE, UPDATE, SELECT



Lệnh INSERT (tt)

- Dùng để thêm 1 hay nhiều dòng vào Table trong CSDL
- Cú pháp (thêm 1 dòng)

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)  
VALUES (<danh sách các giá trị>)
```

- Thêm mới một nhân viên có họ = 'Le', tên lót = 'Van', tên = 'Tuyen' với mã số = 'NV001' vào Table NhanVien

```
INSERT INTO NHANVIEN(HONV, TENLOT, TENNV, MANV)  
VALUES ('Le', 'Van', 'Tuyen', 'NV001')
```



Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

```
DELETE FROM <tên bảng>  
[WHERE <điều kiện>]
```

–Xóa nhân viên có mã số = '345345345' trong Table NhanVien

```
DELETE FROM NHANVIEN  
WHERE MANV='345345345'
```

–Xóa tất cả nhân viên có trong Table NhanVien

```
DELETE FROM NHANVIEN
```



Lệnh UPDATE

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

UPDATE <tên bảng>

SET <tên thuộc tính>=<giá trị mới>,

<tên thuộc tính>=<giá trị mới>,

...

[WHERE <điều kiện>]



Lệnh UPDATE

Ví dụ

– Sửa thuộc tính **ngày sinh** của nhân viên có **mã số** = '345345345' trong Table NhanVien là '08/12/1965'

```
UPDATE NHANVIEN
```

```
SET NGSINH='08/12/1965'
```

```
WHERE MANV='333445555'
```

– Sửa thuộc tính **lương** của tất cả nhân viên bằng **lương cũ * 1.1**

```
UPDATE NHANVIEN
```

```
SET LUONG=LUONG*1.1
```



CÚ PHÁP CÂU LỆNH SELECT

SELECT	<Danh sách các cột được chọn>
FROM	< Danh sách các bảng>
WHERE	<Điều kiện chọn>
GROUP BY	<Danh sách cột cần nhóm>
HAVING	< Điều kiện nhóm>
ORDER BY	< Danh sách cột cần sắp xếp>

Dùng để lấy dữ liệu từ CSDL hiển thị lên màn hình, report,...



Lập trình CSDL trong ứng dụng quản lý

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH CSDL VỚI ADO



❖ Thành phần giao diện



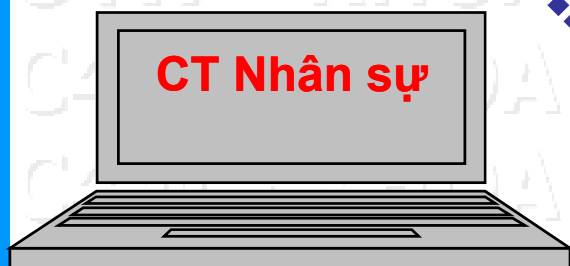
❖ Giao tiếp dữ liệu



❖ Thành phần dữ liệu



Lập trình CSDL trong ứng dụng quản lý



❖ Thành phần giao diện: Visual Basic, Visual C++, C#, VB.NET, Delphi...

❖ Giao tiếp dữ liệu: ODBC, DAO, ADODB, ADO.NET, ..

❖ Thành phần dữ liệu: MS Access, SQL Server, Oracle, ...



Một số hệ quản trị CSDL thông dụng

❖ Thành phần dữ liệu: MS Access, SQL Server, Oracle,...



❖ MS Access 2000/XP/2003

❖ MS SQL Server 7.0/2000/2005

❖ Oracle 8i/9i/10i/11i

❖ My SQL



Lập trình cơ sở dữ liệu với thư viện ADO

(ADO – ACTIVEX DATA OBJECT)

- Tổng quan về lập trình CSDL
- **Thư viện ADO**
- **Lập trình CSDL với ADO**

Sử dụng thư viện ADO

Làm thế nào để sử dụng được thư viện ADO để kết nối dữ liệu trong Visual C++ ?

Tạo CMyADO Class :

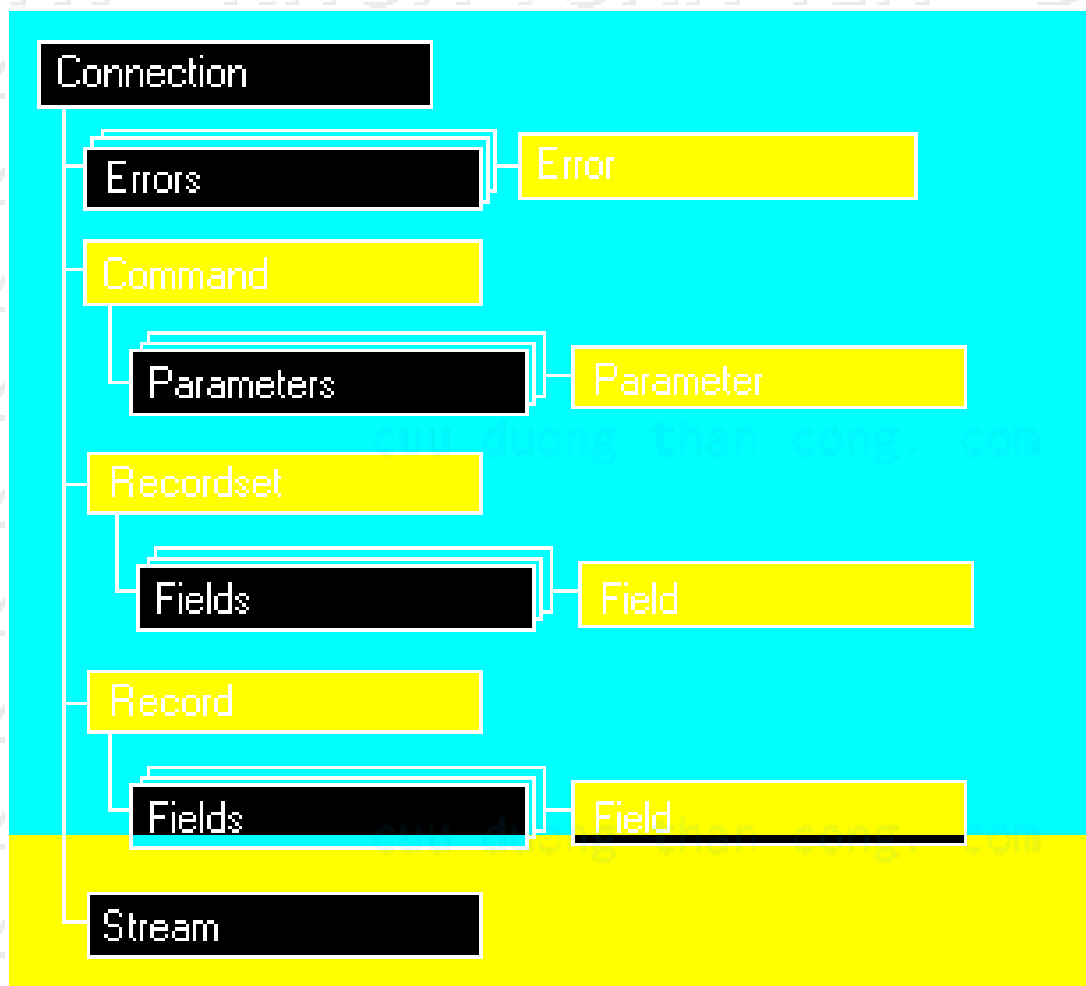
// Thêm dòng dưới đây vào đầu Class .

```
#import "C:\Program Files\Common Files\System\ADO\msado15.dll" \
    no_namespace rename("EOF", "EndOfFile")
```

❖ Tạo InitSQLServer() Function trong CMyADO Class :

```
BOOL CMyADO::InitSQLServer()
{
    if(FAILED(ColInitialize(NULL)))
        return FALSE;
    return TRUE;
}
```





Thư viện ADO

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH CSDL VỚI ADO

❖ **Connection**

❖ **Command**

❖ **RecordSet**

❖ **Field**

❖ **Parameter**



Kết nối Data Source

❖ MS Access



Ứng dụng

Connection



Cơ sở dữ liệu

File *.mdb nào ?

**VD: QLHS.mdb,
QLNS.mdb**

❖ MS SQL Server



Ứng dụng

Connection



Cơ sở dữ liệu

- **Server ?**
- **DatabaseName ?**
- **User Name ?**
- **Password ?**



Đối tượng Connection

□ Ý nghĩa

Đối tượng **Connection** dùng để mở kết nối với nguồn cơ sở dữ liệu, có thể hiểu đối tượng **Connection** đóng vai trò là chìa khóa mở kết nối giữa ứng dụng với cơ sở dữ liệu (Access, SQL Server, Oracle,...)

□ Ví dụ:

Để kết nối ứng dụng quản lý học sinh với cơ sở dữ liệu quản lý học sinh được lưu trữ trong **Access** chúng ta cần tạo một đối tượng **Connection** chỉ đến tập tin Access, ví dụ **QLHS.mdb** hay **C:\C4W\Data\QLHS.mdb**

Để kết nối ứng dụng quản lý học sinh với cơ sở dữ liệu quản lý học sinh được lưu trữ trong **SQL Server** chúng ta cần tạo một đối tượng **Connection** chỉ đến **DB** nào ? **Server** nào ? Với **User** và **password** nào ?



Đối tượng Connection

Connection object

Properties

Attributes
CommandTimeout
ConnectionString
ConnectionTimeout
CursorLocation
DefaultDatabase
Errors Collection
Isolation Level
Mode
Properties collection
Provider
State
Version

Method

Begin Trans
Cancel
Close
CommitTrans
Execute
Open
OpenShema
RollbackTrans

Event

BeginTransComplete
CommitTransComplete
ConnectComplete
Disconnect
ExecuteComplete
InfoMessage
RollbackTransComplete
WillConnect
WillExecute



Đối tượng Connection

☐ Thuộc tính: **ConnectionString**

Thuộc tính **ConnectionString** dùng để gán tên tập tin **MDB** nếu sử dụng **Access** hoặc xác lập **ServerName, DBName, Username, Password** nếu kết nối với **SQL Server**

☐ Phương thức: **Open**

Sau khi xác lập thuộc tính **ConnectionString**, gọi hàm **Open** để mở **Connection**.

Thông thường trong một ứng dụng **Connection** chỉ mở một lần khi load ứng dụng

→ **Biến Connection** thường được khai báo toàn cục để có thể sử dụng trong toàn bộ ứng dụng

☐ Phương thức: **Close**

Phương thức **Close** được gọi để đóng **Connection**. Thông thường trong một ứng dụng gọi **close** đối tượng **connection** khi kết thúc chương trình



❖ Kết nối CSDL SQL Server

//Khai báo biến quản lý connection

ConnectionPtr m_pConnection;

HRESULT hr;

hr = m_pConnection.CreateInstance(__uuidof(Connection));

if (SUCCEEDED(hr))

{

hr= m_pConnection->Open(_bstr_t(L"Provider=sqloledb;Data

Source=ServerName;InitialCatalog=DatabaseName;"),

_bstr_t(L"UserName"),

_bstr_t(L>Password"),

adModeUnknown);

if (SUCCEEDED(hr))

{

bRes = TRUE;

}

}



Tương tác dữ liệu: thêm, xóa,...

Lấy dữ
liệu

Thêm
dữ
liệu

Xóa dữ liệu

Sửa dữ liệu

Danh mục giao viên

Thông tin giáo viên

Mã GV: NV001 Họ tên: Lê Đức Long

Địa chỉ: 297 Lê Quý Đôn - P.8 - Quận 3

Điện thoại: 8500347 Email: ldlong@hcm.vnn.vn

Ghi chú:

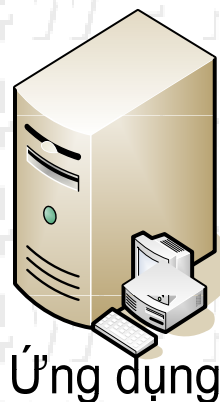
Danh sách giáo viên

Mã GV	Họ Tên	Địa chỉ	Điện thoại	Email
NV001	Lê Đức Long	297 Lê Quý Đôn - P.8 - Quận 3	8500347	ldlong@hcm.vnn.vn
NV002	Nguyễn Công Phú	397 Lê Quang Định - P.5 - Q. Bình Thạnh	8123567	ncphu@yahoo.com
NV003	Nguyễn Lương Anh Tuấn	153/2 Hoàng Văn Thụ - Q. Tân Bình	8723567	ltuan@yahoo.com
NV004	Cao Thị Tố Trinh	24 Nguyễn Thị Minh Khai - Q.3	9345882	ctttrinh@yahoo.com
NV005	Lý Thành	123 Trương Định - Quận 3	9321213	lthanh@yahoo.com
NV006	Trần Thị Mỹ Châu	26 Nguyễn Bình Khiêm - Quận 1	0913670277	
NV007	Trần Thị Minh Nguyệt	32 Trần Bình Trọng - Quận 5		

Thêm Xóa Sửa Ghi Không Thoát



Thực thi câu lệnh Insert/Delete...



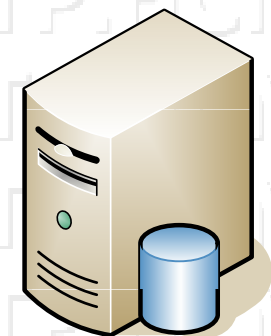
Ứng dụng

Thêm dữ liệu

Cập nhật dữ liệu

Xóa dữ liệu

Lấy dữ liệu

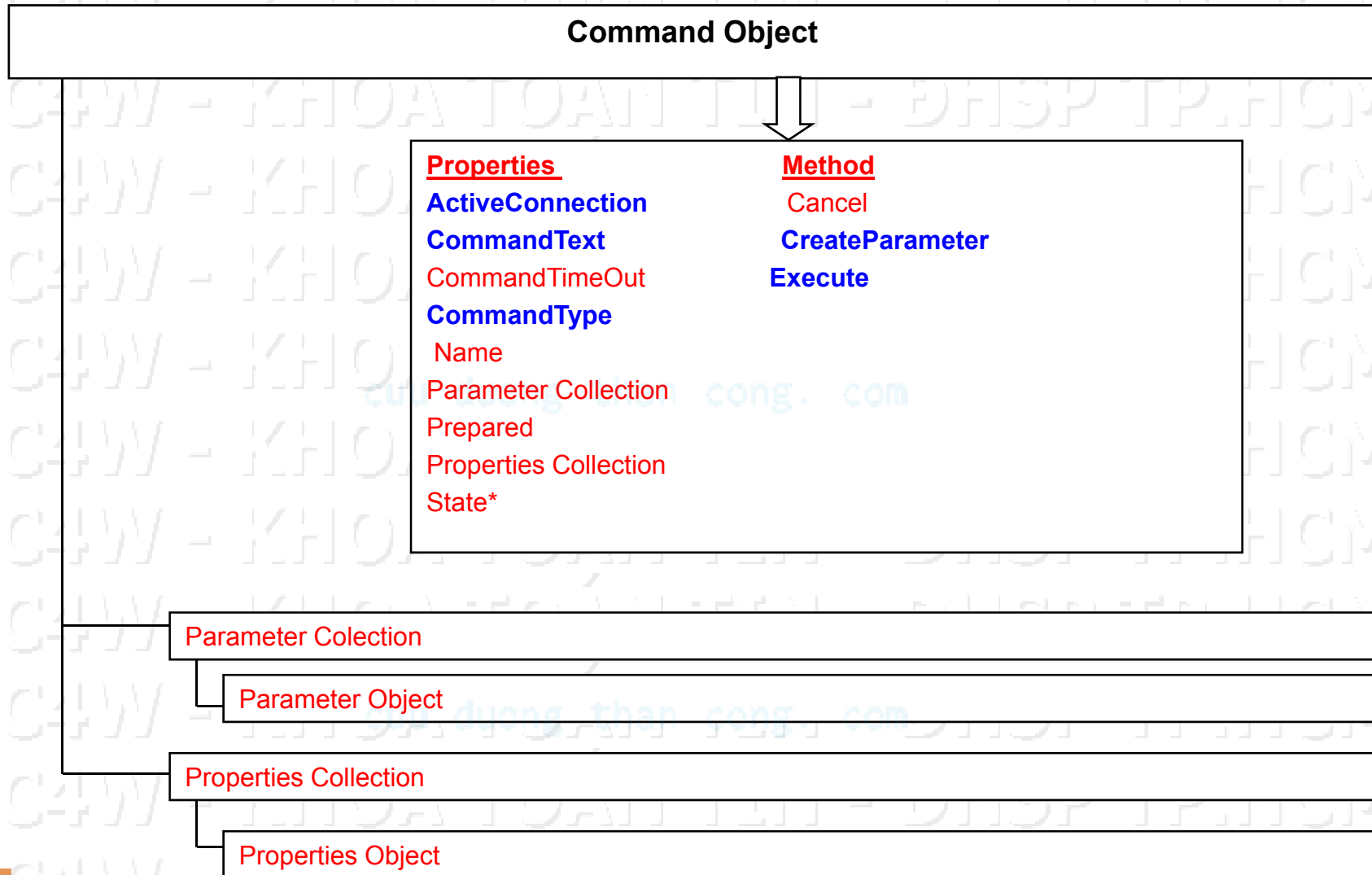


Cơ sở dữ liệu

Thực thi các câu lệnh Insert, Delete, update, Select thông qua
đối tượng **Command**



Đối tượng Command



Đối tượng Command

- ❑ Thuộc tính: **ActiveConnection**

Thuộc tính **ActiveConnection** dùng để gán một đối tượng Connection đang được mở

- ❑ Thuộc tính: **CommandText**

Thuộc tính **CommandText** dùng để xác định câu lệnh SQL “thao tác” cần thực hiện trong 4 loại thao tác : **SELECT, INSERT, DELETE, UPDATE**

- ❑ Thuộc tính: **CommandType**

Thuộc tính **CommandType** dùng để xác định nội dung trong **CommandText** thuộc loại nào ? CommandType là tên của Table/Views, Tên Stored procedure hay một câu lệnh SQL ?

- ❑ Phương thức: **Execute**

Phương thức **Execute** được gọi để thực thi một câu lệnh SQL được gán cho trong thuộc tính **CommandText**



Đối tượng Command

//Khai báo biến quản lý Command

_CommandPtr g_pCmd;

❖ **Kết nối CSDL thông qua Connection**

❑ **Cú pháp**

g_pCmd.CreateInstance(__uuidof(Command));

g_pCmd->ActiveConnection=m_pConnection;

//với m_pConnection là 1 connection đang mở



//Khai báo biến quản lý Command

_CommandPtr g_pCmd;

❖ **Thực thi câu lệnh SQL: Insert/Delete/Update**

❑ **Cú pháp**

```
CString sSQL;
```

```
g_pCmd->CommandText=_bstr_t(sSQL) ;
```

```
g_pCmd->CommandType=adCmdText;
```

```
_variant_t vRecsAffected(0L) ;
```

```
g_pCmd->Execute (&vRecsAffected, NULL, adOptionUnspecified) ;
```



❖ Thực thi câu lệnh SQL: Insert/Delete/Update

❑ Ví dụ: thực thi một câu lệnh SQL Xóa dữ liệu từ Table HocSinh

```
CString sSQL="Delete From HocSinh";  
g_pCmd->CommandText=_bstr_t(sSQL);  
g_pCmd->CommandType=adCmdText;  
_variant_t vRecsAffected(0L);  
try  
{  
    g_pCmd->Execute(&vRecsAffected,NULL,adOptionUnspecified);  
}  
catch( _com_error &e )  
{  
    AfxMessageBox("loi");  
}
```



❖ Thực thi câu lệnh SQL: Insert/Delete/Update

❑ Ví dụ: thực thi một câu lệnh SQL Insert dữ liệu vào Table HocSinh

```
CString sSQL="";
sSQL = "Insert into HocSinh (MaHS, TenHS) Values (5";
sSQL = sSQL + ", \"TNB\")";
g_pCmd->CommandText=_bstr_t(sSQL);
g_pCmd->CommandType=adCmdText;
_variant_t vRecsAffected(0L);
try
{
    g_pCmd->Execute(&vRecsAffected, NULL, adOptionUnspecified);
}
catch( _com_error &e )
{
    //Xu ly loi
}
```



❖ Thực thi câu lệnh SELECT

Kết quả câu truy vấn SELECT là một danh sách, ví dụ như danh sách sinh viên, danh sách học sinh, danh sách nhân viên,...

→ Cần có đối tượng lưu trữ kết quả thực thi lệnh SELECT

→ Đó là đối tượng Recordset

❑ Cú pháp

```
CString sSQL;  
  
g_pCmd->CommandText=_bstr_t(sSQL) ;  
  
g_pCmd->CommandType=adCmdText;  
  
_variant_t vRecsAffected(0L) ;  
  
g_pRS = g_pCmd->Execute(&vRecsAffected, NULL, adOptionUnspecified) ;
```



Đối tượng Recordset



Recordset Object

Properties

AbsolutePage
AssolutePosition
ActiveCommand
ActiveConnection
BOF
Bookmark
CacheSize
CursorLocation
CursorType
DataMember
DataSource
EditMode
EOF
Fields Collection
Filter
LockType
MarshalOptions
MaxRecords
PageCount
Pagesize
Properties Collection
RecordCount
Sort
State
Status
stayInSync

Method

AddNew
Cancel
CancelBatch
CancelUpdate
Clone
CompareBookmarks
Delete
Find
GetRows
GetString
Move
MoveFirst
MoveLast
MoveNext
MovePrevious
NextRecordset
Open
Requery
Resync
Save
Supports
Update
Updatebatch

Events

EndOfRecordset
FetchComplete
FetchProcess
FieldChangeComplete
MoveComplete
RecordChangeComplete
RecordsetChangeComlete
WillChangeField
WillChangeRecord
WillChangeRecordset
WillMove

Đối tượng Recordset

Recordset Object

Properties

AbsolutePage
AssolutePosition
ActiveCommand
ActiveConnection
BOF
.....

Method

AddNew
Cancel
CancelBatch
CancelUpdate
Clone

Events

EndOfRecordset
FetchComplete
FetchProcess
FieldChangeComplete
MoveComplete

Fields Collection

Fields object

Properties Collection

Property Object



Đối tượng Recordset

//Khai báo biến quản lý RecordSet

_RecordsetPtr g_pRS;

❖ Thực thi câu lệnh SELECT

❑ Cú pháp

```
g_pRS.CreateInstance(__uuidof(Recordset));  
g_pRS->CursorType = adOpenStatic;  
// Use client cursor to enable AbsolutePosition property.  
g_pRS->CursorLocation = adUseClient;  
CString sSQL;  
g_pCmd->CommandText=_bstr_t(sSQL);  
g_pCmd->CommandType=adCmdText;  
_variant_t vRecsAffected(0L);  
g_pRS = g_pCmd->Execute(&vRecsAffected, NULL, adOptionUnspecified);
```



Đối tượng Recordset

//Khai báo biến quản lý RecordSet

_RecordsetPtr g_pRS;

- ❑ Ví dụ: thực thi một câu lệnh SQL Select dữ liệu từ Table HocSinh

```
g_pRS.CreateInstance(__uuidof(Recordset));
```

```
g_pRS->CursorType = adOpenStatic;
```

```
// Use client cursor to enable AbsolutePosition property.
```

```
g_pRS->CursorLocation = adUseClient;
```

```
CString sSQL="Select * from HocSinh";
```

```
g_pCmd->CommandText=_bstr_t(sSQL);
```

```
g_pCmd->CommandType=adCmdText;
```

```
_variant_t vRecsAffected(0L);
```

```
g_pRS = g_pCmd->Execute(&vRecsAffected, NULL, adOptionUnspecified);
```



Đối tượng Recordset

❖ Duyệt toàn bộ các mẫu tin trong RecordSet

❑ Cú pháp

```
g_pRS->MoveFirst();  
while (! g_pRS->EOF)  
{  
    _variant_t vField1;  
    _variant_t vField2;  
    vField1 = g_pRS->GetCollect(L"FieldName1");  
    vField2 = g_pRS->GetCollect(L"FieldName2");  
  
    //////////////////////////////////////  
    //  
    g_pRS->MoveNext();  
}
```



Đối tượng Recordset

❖ Duyệt toàn bộ các mẫu tin trong RecordSet

❑ Ví dụ: thực thi một câu lệnh SQL Select dữ liệu từ Table HocSinh

```
CString sSQL="";
sSQL = "Select * from HocSinh";

...
_variant_t vRecsAffected(0L);
g_pRS=doc->myADO.g_pCmd->Execute(&vRecsAffected,NULL,adOptionUnspecified);
//Duyệt toàn bộ các mẫu tin
CString s;
g_pRS->MoveFirst();
while (!g_pRS->EndOfFile)
{
    _variant_t vMaHS;
    _variant_t vTenHS;
    vMaHS = doc->myADO.g_pRS->GetCollect(L"MaHS");
    vTenHS = doc->myADO.g_pRS->GetCollect(L"TenHS");
    s.Format("%d",vMaHS.intVal);
    MessageBox(s);
    g_pRS->MoveNext();
}
```



❖ Các phương thức duyệt mẫu tin

- ❑ Về đầu danh sách

`g_pRs->MoveFirst()`

- ❑ Về cuối danh sách

`g_pRs->MoveLast()`

- ❑ Di chuyển về sau một mẫu tin

`g_pRs->MoveNext()`

- ❑ Về đầu về trước một mẫu tin

`g_pRs->MovePrevious()`



❖ Các phương thức duyệt mẫu tin

Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT

`g_pRs->MoveFirst()`



Đối tượng Recordset

❖ Các phương thức duyệt mẫu tin

Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT

g_pRs->MoveLast()



Đối tượng Recordset

❖ Các phương thức duyệt mẫu tin

Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT

g_pRs->MoveNext()



Đối tượng Recordset

❖ Các phương thức duyệt mẫu tin

Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT

`g_pRs->EOF = True`

`g_pRs->MoveNext()`



❖ Các phương thức duyệt mẫu tin

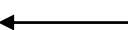
Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT

g_pRs->MovePrevious()



❖ Các phương thức duyệt mẫu tin

`g_pRs->BOF = True`



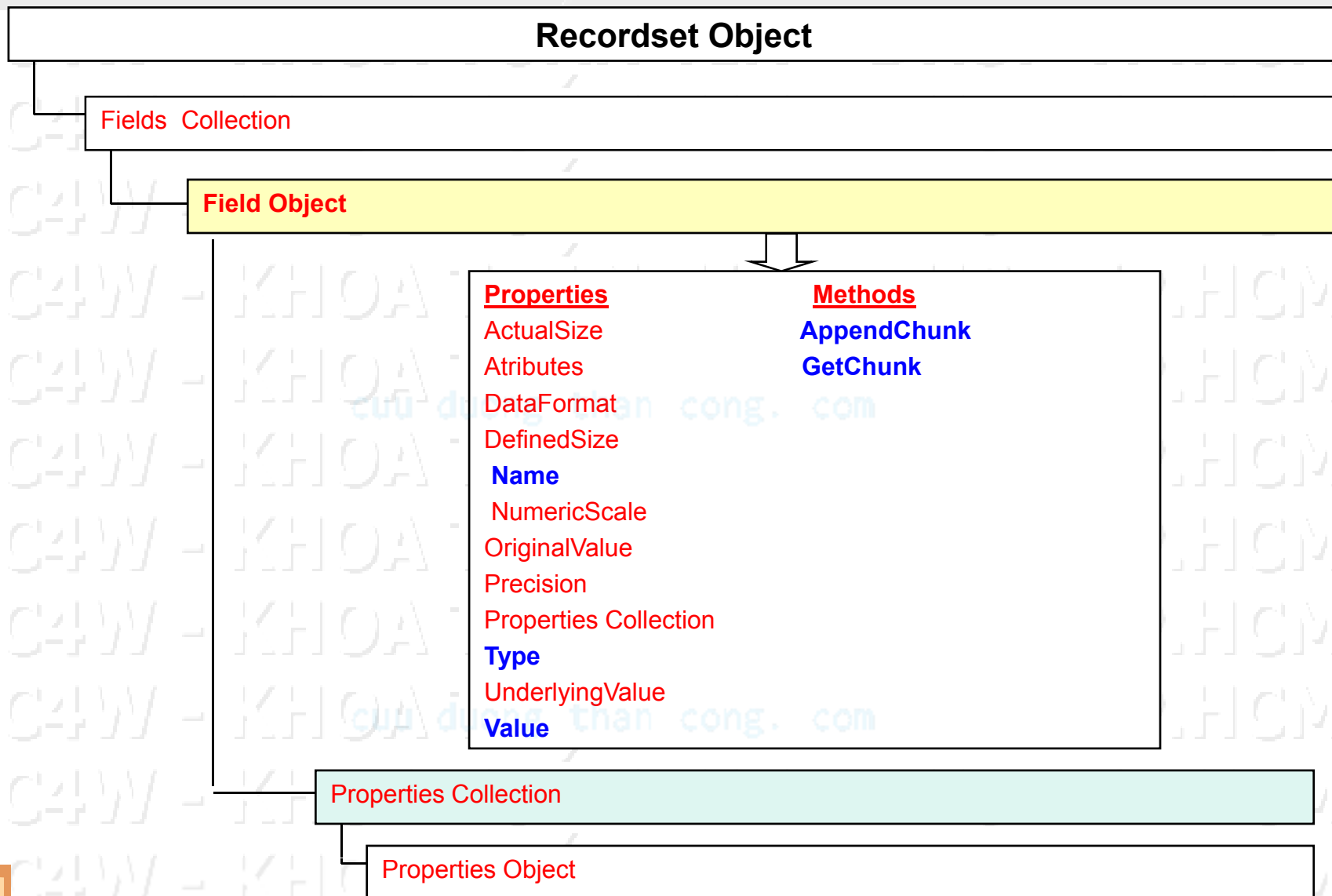
`g_pRs->MovePrevious()`



Mhoc	TenMH	MaMH	TinChi	Khoa
	Nhap mon tin hoc	COSC1310	4	CNTT
	Cau truc du lieu	COSC3320	4	CNTT
	Toan roi rac	MATH2410	3	TOAN
	Co so du lieu	COSC3380	3	CNTT



Đối tượng Field



Lập trình cơ sở dữ liệu với thư viện ADO

(ADO – ACTIVEX DATA OBJECT)

- Tổng quan về lập trình CSDL

- Thư viện ADO

- **Lập trình CSDL với ADO**

Lập trình CSDL với ADO

Phòng giáo vụ tại một trường Đại học muốn tin học hóa việc quản lý đăng ký học chuyên đề của sinh viên. Kết quả phân tích thiết kế được mô hình CSDL quan hệ như sau:

1. SINHVIEN (MASV, HOTEN, PHAI, NGAYSINH, DCHI, MANGANH)

Tên từ: Mỗi sinh viên có một mã số duy nhất (**MASV**), một họ tên (**HOTEN**), thuộc một phái (**PHAI**), có một ngày sinh (**NGAYSINH**), có một địa chỉ (**DCHI**), và học một ngành (**MANGANH**)

MaSV	TenSV	Nam	Khoa
K28.103.001	Trần Quốc Thanh	1	CNTT
K28.101.001	Nguyễn Công Phú	1	TOAN
K28.101.002	Phan Anh Khanh	1	TOAN
K27.101.001	Phạm Khánh Như	2	TOAN
K26.102.001	Trần Ngọc Dung	3	VLY
K25.201.001	Phạm Khánh Như	4	HOA



Lập trình CSDL với ADO

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

- ☐ Xem danh sách sinh viên
- ☐ Thêm mới sinh viên
- ☐ Xóa sinh viên khỏi danh sách
- ☐ Thay đổi thông tin sinh viên

The screenshot shows a Windows application window titled "Thông tin sinh viên". It features a form with the following fields: "Ma SV" (text box), "Ten SV" (text box), "Nam" (dropdown menu showing "0"), and "Khoa" (text box). Below the form is a table with four columns: "MaSV", "Ten SV", "Nam", and "Khoa". The table is currently empty. At the bottom of the window, there are six buttons: "Mo CNN", "Lay DS", "Them", "Xoa", "Sua", and "Thoat".



Xem danh sách sinh viên

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

Thông tin sinh viên

Ma SV:

Ten SV:

Nam: Khoa:

MaSV	Ten SV	Nam	Khoa
K25.201.001	Pham Khanh Nhu	4	HOA
K26.102.001	Tran Ngoc Dung	3	VLV
K27.101.001	Pham Khanh Nhu	2	TOAN
K28.101.001	Nguyen Cong Phu	1	TOAN
K28.101.002	Phan Anh Khanh	1	TOAN
K28.103.001	Tran Quoc Thanh	1	CNTT

Mo CNN Lay DS Them Xoa Sua Thoat

Lấy danh sách
sinh viên lưu trữ
trong **table**
SinhVien trong
CSDL hiển thị lên
Grid trong màn
hình Thông tin
sinh viên

Thực hiện truy vấn
dữ liệu với câu lệnh
SELECT



Xem danh sách sinh viên

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

Thông tin sinh viên

Ma SV:

Ten SV:

Nam: Khoa:

MaSV	Ten SV	Nam	Khoa
K25.201.001	Pham Khanh Nhu		4 HOA
K26.102.001	Tran Ngoc Dung		3 VLY
K27.101.001	Pham Khanh Nhu		2 TOAN
K28.101.001	Nguyen Cong Phu		1 TOAN
K28.101.002	Phan Anh Khanh		1 TOAN
K28.103.001	Tran Quoc Thanh		1 CNTT

Mo CNN Lay DS Them Xoa Sua Thoat

Lấy danh sách
sinh viên lưu trữ
trong **table**
SinhVien trong
CSDL hiển thị lên
Grid trong màn
hình Thông tin
sinh viên

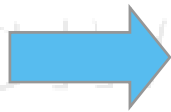
Thực hiện truy vấn
dữ liệu với câu lệnh
SELECT

SELECT MASV, TENSX, NAM, KHOA
FROM SINHVIEN



Xem danh sách sinh viên

Lấy danh sách sinh viên lưu trữ trong **table SinhVien** trong **CSDL** hiển thị lên **Grid** trong màn hình Thông tin sinh viên



Cần phải sử dụng các đối tượng sau trong thư viện ADO

☐ **Connection**

Thiết lập kết nối với CSDL có chứa Table SinhVien

☐ **Command**

Thực thi câu lệnh SQL

```
SELECT MASV, TENSX, NAM, KHOA  
FROM SINHVIEN
```

☐ **Recordset**

Chứa kết quả thực thi câu lệnh SQL **SELECT** trên



Thêm mới sinh viên

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

Thông tin sinh viên

Ma SV:

Ten SV:

Nam: Khoa:

MaSV	Ten SV	Nam	Khoa
K25.201.001	Pham Khanh Nhu	4	HOA
K26.102.001	Tran Ngoc Dung	3	VLV
K27.101.001	Pham Khanh Nhu	2	TOAN
K28.101.001	Nguyen Cong Phu	1	TOAN
K28.101.002	Phan Anh Khanh	1	TOAN
K28.103.001	Tran Quoc Thanh	1	CNTT

Mo CNN Lay DS Them Xoa Sua Thoat

Lấy thông tin sinh viên trong màn hình Thông tin sinh viên lưu vào Table HocSinh

Thực hiện câu lệnh
INSERT



Thêm mới sinh viên

Lấy thông tin sinh viên trong màn hình Thông tin sinh viên lưu vào table SinhVien



Cần phải sử dụng các đối tượng sau trong thư viện ADO

☐ Connection

Thiết lập kết nối với CSDL có chứa Table SinhVien

☐ Command

Thực thi câu lệnh SQL

```
INSERT INTO SINHVIEN (MASV, TENSX, NAM, KHOA)
VALUES ('????', '????', '??', '??')
```



Các giá trị MASV, TENSX,.. Được lấy từ các control (Edittext) trong màn hình thông tin sinh viên



Xóa sinh viên

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

Thông tin sinh viên

Ma SV:

Ten SV:

Nam: Khoa:

MaSV	Ten SV	Nam	Khoa
K25.201.001	Pham Khanh Nhu		4 HOA
K26.102.001	Tran Ngoc Dung		3 VLY
K27.101.001	Pham Khanh Nhu		2 TOAN
K28.101.001	Nguyen Cong Phu		1 TOAN
K28.101.002	Phan Anh Khanh		1 TOAN
K28.103.001	Tran Quoc Thanh		1 CNTT

Mo CNN Lay DS Them Xoa Sua Thoat

Xóa sinh viên
đang được chọn
trong danh sách
sinh viên

Thực hiện câu lệnh
DELETE



Xóa sinh viên

Xóa sinh viên đang được chọn trong danh sách sinh viên ra khỏi table SinhVien



Cần phải sử dụng các đối tượng sau trong thư viện ADO

☐ Connection

Thiết lập kết nối với CSDL có chứa Table SinhVien

☐ Command

Thực thi câu lệnh SQL

**DELETE FROM SINHVIEN
WHERE MASV = '???'**



Giá trị MASV được lấy từ các control (MaSV) trong màn hình thông tin sinh viên



Cập nhật thông tin sinh viên

Thiết kế màn hình quản lý sinh viên cho phép thực hiện các thao tác liên quan đến sinh viên như:

Thông tin sinh viên

Ma SV:

Ten SV:

Nam: Khoa:

MaSV	Ten SV	Nam	Khoa
K25.201.001	Pham Khanh Nhu		4 HOA
K26.102.001	Tran Ngoc Dung		3 VLY
K27.101.001	Pham Khanh Nhu		2 TOAN
K28.101.001	Nguyen Cong Phu		1 TOAN
K28.101.002	Phan Anh Khanh		1 TOAN
K28.103.001	Tran Quoc Thanh		1 CNTT

Mo CNN Lay DS Them Xoa Sua Thoat

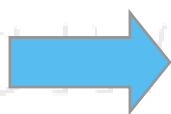
Thay đổi thông tin sinh viên đang được chọn bằng thông tin đã được sửa đổi trong màn hình thông tin sinh viên

Thực hiện câu lệnh
UPDATE



Cập nhật thông tin sinh viên

Để cập nhật thông tin sinh viên trên màn hình vào table SinhVien trong CSDL



Cần phải sử dụng các đối tượng sau trong thư viện ADO

☐ Connection

Thiết lập kết nối với CSDL có chứa Table SinhVien

☐ Command

Thực thi câu lệnh SQL

UPDATE SINHVIEN

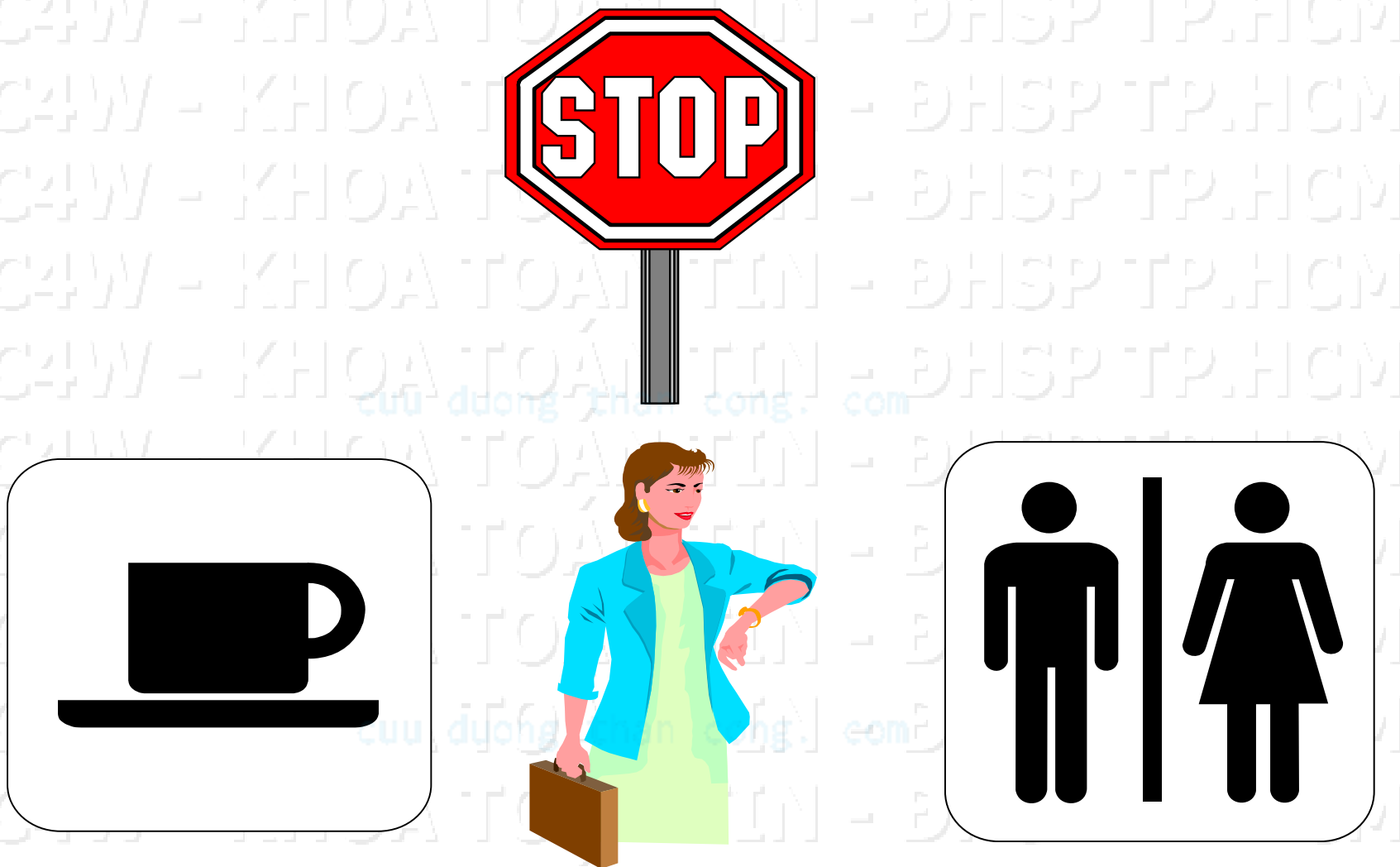
SET HOTEN = '???' , NAM = ??? , KHOA = '???'

WHERE MASV = '???'



Các giá trị MASV, TENSX, .. Được lấy từ các control (Edittext) trong màn hình thông tin sinh viên





Thư viện liên kết động

(DLL – DYNAMIC LINK LIBRARY)

- Giới thiệu thư viện liên kết
- Các loại thư viện liên kết
- Xây dựng thư viện liên kết động

tnbao.dhsp@gmail.com

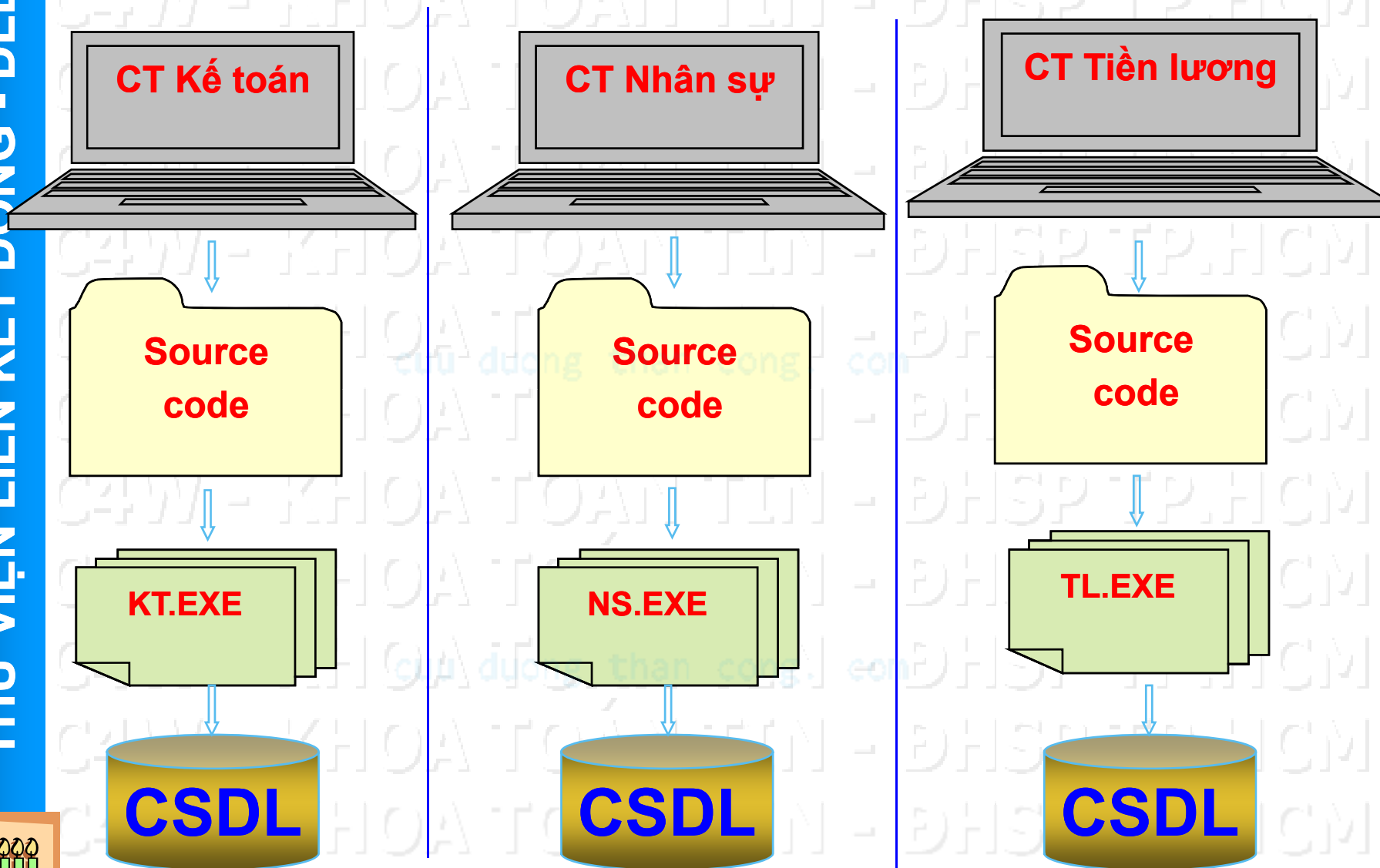
Thư viện liên kết động

(DLL – DYNAMIC LINK LIBRARY)

- **Giới thiệu thư viện liên kết**
- **Các loại thư viện liên kết**
- **Xây dựng thư viện liên kết động**

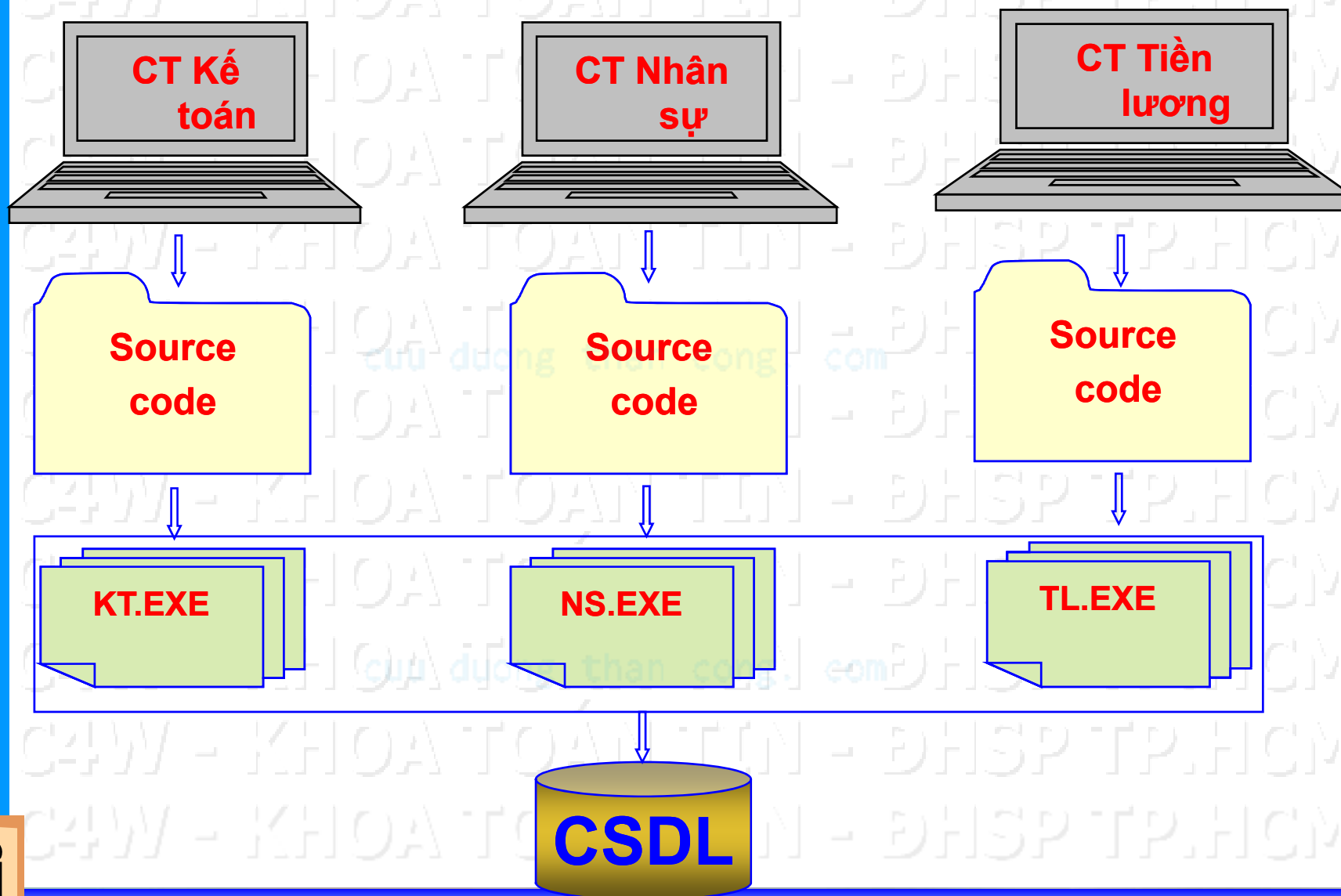
GIỚI THIỆU – LIÊN KẾT LÀ GÌ ?

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



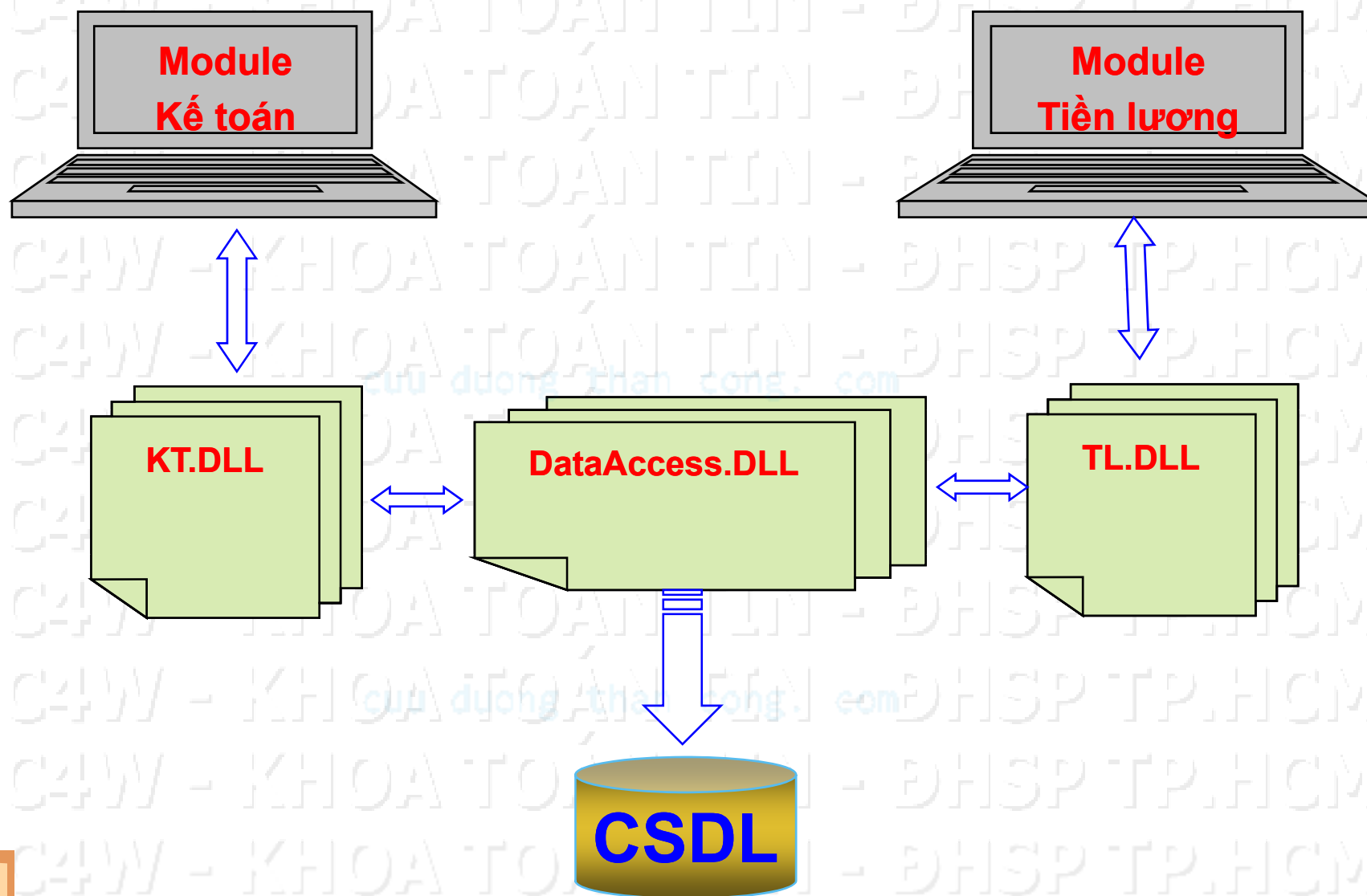
GIỚI THIỆU – LIÊN KẾT LÀ GÌ ?

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



GIỚI THIỆU – LIÊN KẾT LÀ GÌ ?

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



GIỚI THIỆU – LIÊN KẾT LÀ GÌ ?

- ❖ Thư viện là một tập tin chứa những hàm hoặc tài nguyên khác có sẵn mà các ứng dụng có thể lấy để sử dụng. Các tài nguyên này thường cho phép nhiều ứng dụng cùng chia sẻ sử dụng dễ dàng một đoạn mã hoặc tài nguyên
- ❖ Liên kết là cách thức mà trình biên dịch nhúng/kết hợp các đoạn mã thực thi của những module thư viện (Lib) vào chương trình
- ❖ Có 2 cách liên kết:
 - ❖ Liên kết tĩnh
 - ❖ Liên kết động



Thư viện liên kết động

(DLL – DYNAMIC LINK LIBRARY)

- Giới thiệu thư viện liên kết
- **Các loại thư viện liên kết**
- **Xây dựng thư viện liên kết động**

❖ Thư viện liên kết tĩnh (Static Linking Library)

- Dạng file: .LIB

- Chứa mã lệnh nhị phân của các hàm thư viện

- Dùng để nhúng vào file chương trình khi thực hiện giai đoạn liên kết (linking) của quá trình biên dịch

- Trình biên dịch sẽ copy đoạn mã lệnh của các hàm thư viện vào trong những module gọi chúng

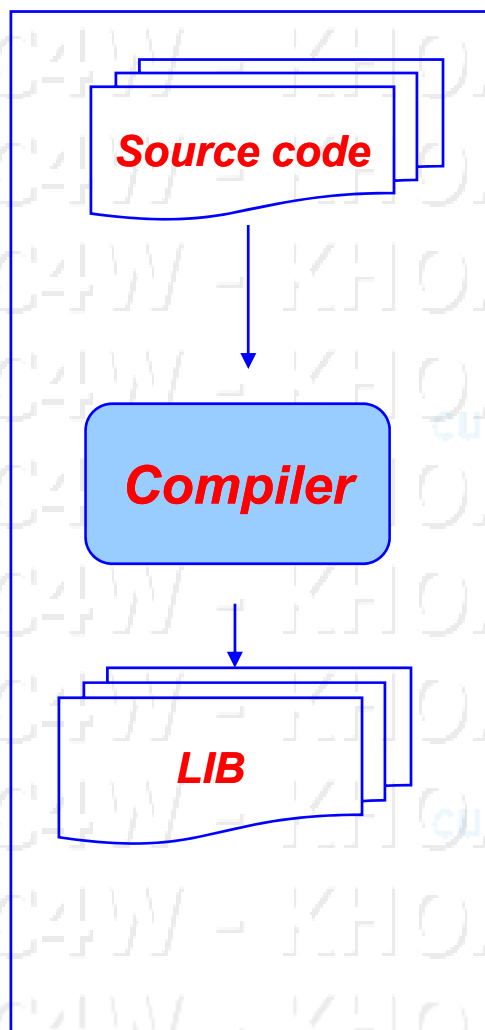


THƯ VIỆN LIÊN KẾT TĨNH

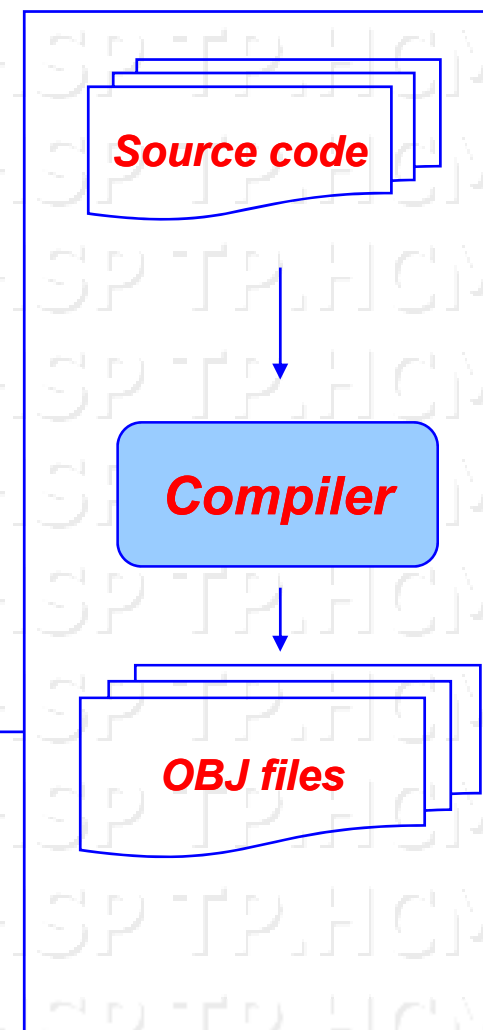
BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



Thư viện liên kết



Chương trình chính



❖ Ưu điểm

- Dễ thực hiện
- Chương trình có thể chạy độc lập, không cần các file kèm theo
- Chương trình chạy nhanh hơn vì không cần thực hiện load thư viện

❖ Khuyết điểm

- Đoạn mã lệnh được nhúng vào chương trình → kích thước chương trình lớn, tốn bộ nhớ.
- Đoạn mã lệnh được nhúng vào nhiều file chương trình khác nhau → không tối ưu
- Khi thay đổi hàm thư viện → phải biên dịch lại toàn bộ các file chương trình



❖ Thư viện liên kết động (Dynamic Linking Library)

- Dạng file: .LIB và DLL

- File .LIB

- Thư viện nhập (Import Library)
- Không chứa mã lệnh của các hàm
- Chỉ chứa các thông tin cần thiết để hệ điều hành nạp thư viện DLL và xác định các hàm export trong DLL
- Sử dụng khi dùng cách thức load – time dynamic linking

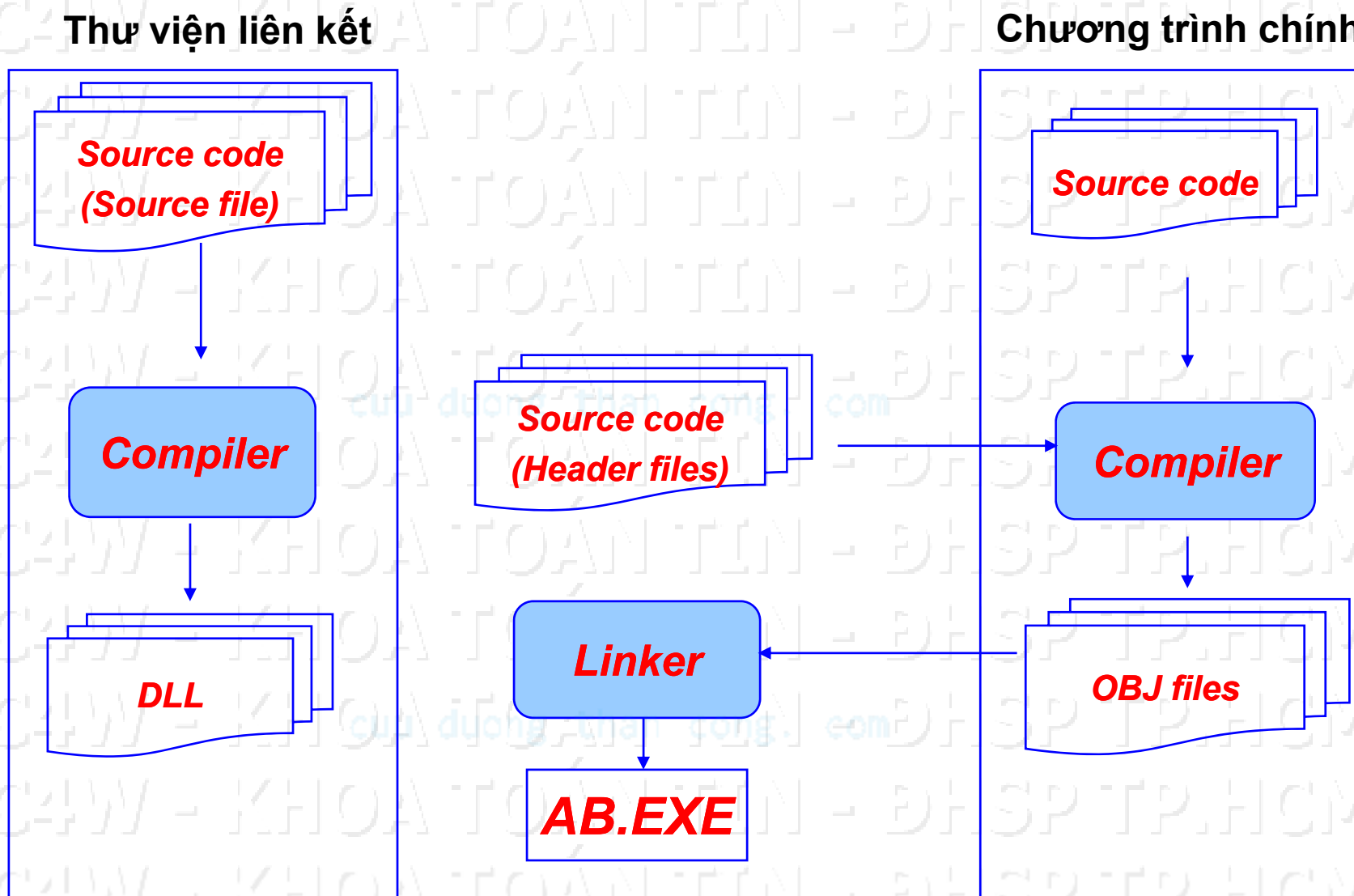
- File .DLL

- Chứa mã lệnh nhị phân của các hàm thư viện
- Được tải vào bộ nhớ khi ứng dụng gọi hàm thư viện
- Cần phải chép tập tin đi kèm khi thực thi ứng dụng



THƯ VIỆN LIÊN KẾT ĐỘNG

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



❖ Ưu điểm

- Mã lệnh của các hàm sẽ không được nhúng vào trong file chương trình của ứng dụng → Kích thước của ứng dụng nhỏ
- Ứng dụng chỉ cần lưu thông tin của hàm thư viện, khi cần hệ điều hành sẽ load các hàm thư viện vào bộ nhớ
- Nhiều ứng dụng có thể dùng chung 1 DLL. Do đó, tiết kiệm bộ nhớ (thông thường mỗi ứng dụng có data riêng, nhưng có thể chia sẻ mã lệnh)
- Khi không còn sử dụng, có thể giải phóng DLL khỏi bộ nhớ
- Khi cần nâng cấp, chỉ cần thay thế file DLL, các file chương trình khác không bị ảnh hưởng

❖ Khuyết điểm

- Khó sử dụng hơn

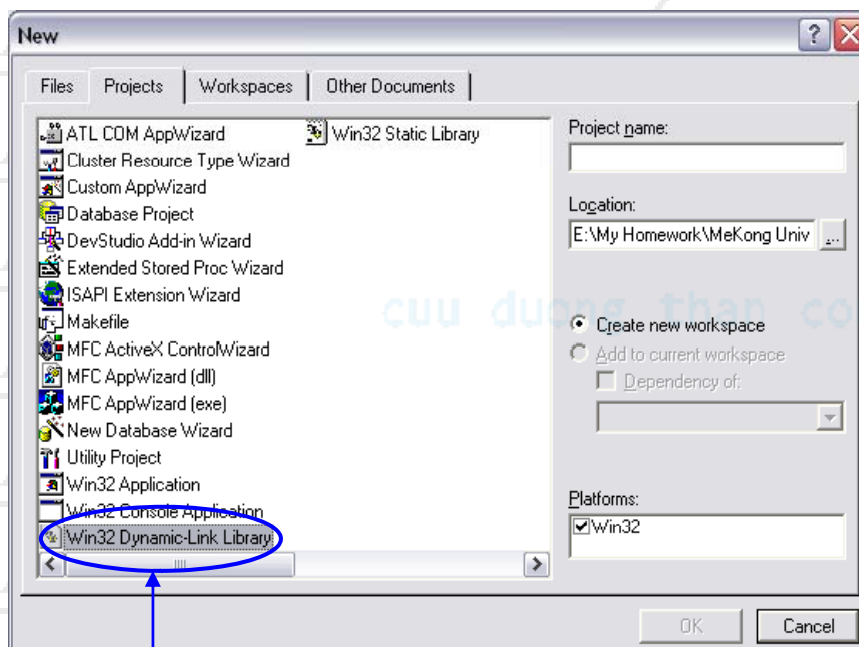


Thư viện liên kết động

(DLL – DYNAMIC LINK LIBRARY)

- Giới thiệu thư viện liên kết
- Các loại thư viện liên kết
- **Xây dựng thư viện liên kết động**

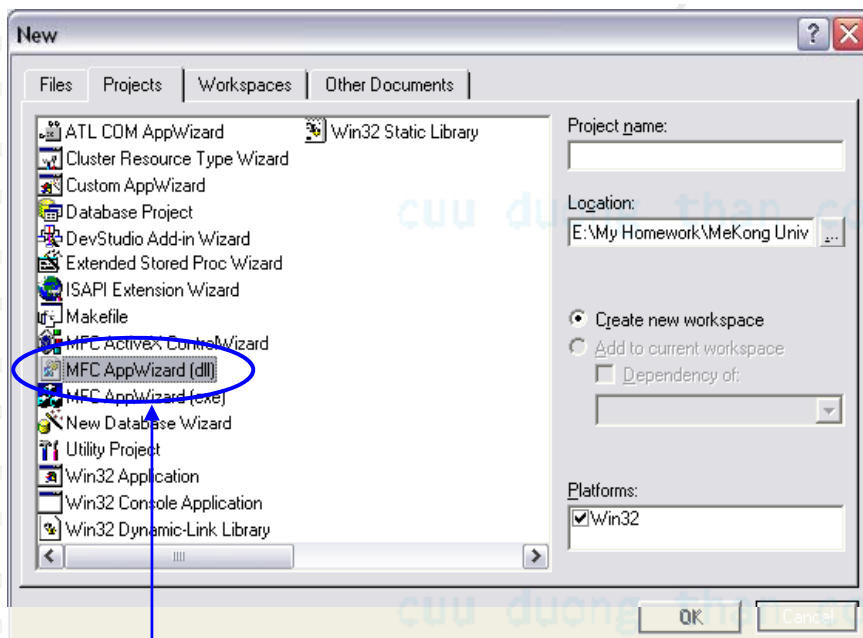
❖ Xây dựng thư viện DLL bằng thư viện Win32API



Chọn loại Project:
“Win32 Dynamic Link Library”



❖ Xây dựng thư viện DLL bằng thư viện MFC



Chọn loại Project:
“MFC AppWizard(Dll)”



- ❖ Tạo lập DLL
- ❖ Phân tích các khai báo
- ❖ Phân tích hàm DllMain
- ❖ Demo liên kết DLL với ứng dụng



XÂY DỰNG DLL DÙNG THƯ VIỆN WIN32 API

- ❖ Chọn menu **File** → **New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project "**Win32 Dynamic Link Library**"
- ❖ Đặt tên project trong ô "**Project name**"
- ❖ Xác định đường dẫn thư mục trong "**Location**"
- ❖ Step 1: Chọn loại ứng dụng "**A DLL that exports some symbols**"
- ❖ Nhấn "**Finish**" để kết thúc



XÂY DỰNG DLL DÙNG THƯ VIỆN WIN32 API

❖ **dllexport:** cung cấp hàm, dữ liệu, tài nguyên,...cho các chương trình/DLL khác sử dụng

```
#define DLLEXPORT __declspec(dllexport)
```

```
//Khai báo 1 biến "xuất khẩu"
```

```
DLLEXPORT int nMyValue;
```

```
//Khai báo 1 hàm "xuất khẩu"
```

```
DLLEXPORT int MyFunction(void)
```

❖ **dllimport:** load 1 hàm, dữ liệu, tài nguyên,...từ 1 DLL khác để sử dụng

```
#define DLLIMPORT __declspec(dllimport)
```

```
//Khai báo 1 biến "nhập khẩu"
```

```
DLLIMPORT int nMyValue;
```

```
//Khai báo 1 hàm "nhập khẩu"
```

```
DLLIMPORT int MyFunction(void)
```



❖ Phân tích hàm DllMain

```
BOOL WINAPI DllMain( HANDLE hModule,  
                    DWORD ul_reason_for_call,  
                    LPVOID lpReserved  
                    )  
{  
    switch (ul_reason_for_call)  
    {  
        case DLL_PROCESS_ATTACH:  
        case DLL_THREAD_ATTACH:  
        case DLL_THREAD_DETACH:  
        case DLL_PROCESS_DETACH:  
            break;  
    }  
    return TRUE;  
}
```



❖ Phân tích hàm DllMain (tt)

- Hàm DllMain là hàm chính của DLL
- Hàm DllMain được gọi khi DLL được load vào bộ nhớ hoặc khi Windows yêu cầu DLL kết thúc (unload khỏi bộ nhớ)
- Hàm DllMain có nhiệm vụ khởi tạo hoặc giải phóng các tài nguyên sử dụng cho DLL đó (nếu có)
- Các tham số:
 - **hModule**: handle của DLL, do Windows tạo ra
 - **ul_reason_for_call**: lý do hàm DllMain được gọi



❖ Phân tích hàm **DllMain** (tt)

▪ **DLL_PROCESS_ATTACH:**

- Hàm **DllMain** với tham số **DLL_PROCESS_ATTACH** được gọi khi process tiến hành load DLL
- Thư viện DLL đang được Windows ánh xạ vào vùng nhớ của tiến trình (thực hiện lời gọi DLL)
- Đây là thời điểm để DLL khởi tạo các biến, cấp phát vùng nhớ hay những thao tác cần thiết khác trước khi cho phép tiến trình gọi đến các hàm của thư viện

▪ **DLL_PROCESS_DETACH:**

- Thư viện DLL được giải phóng khỏi vùng nhớ của tiến trình do 1 trong 3 nguyên nhân: nạp DLL không thành công, tiến trình kết thúc, hay tiến trình gọi hàm **FreeLibrary**
- Đây là thời điểm để giải phóng các biến hay tài nguyên mà DLL đã cấp phát



❖ Phân tích hàm DllMain (tt)

▪ DLL_THREAD_ATTACH:

- Khi tiến trình tạo mới một tiểu trình (Thread), Windows gọi hàm DllMain của tất cả các thư viện DLL đang được sử dụng với tiến trình đó.
- Đây là thời điểm để khởi tạo các biến dùng cho tiểu trình
- Lưu ý rằng tình huống này chỉ xảy ra khi tiểu trình được tạo sau khi thư viện DLL đã load vào tiến trình, nghĩa là nếu DLL được load bằng hàm **LoadLibrary** thì tất cả các tiểu trình hiện có (trong tiến trình) sẽ không gọi hàm DllMain với tham số này.

▪ DLL_THREAD_DETACH:

- Khi 1 tiểu trình kết thúc, Windows gọi hàm DllMain của tất cả các thư viện DLL đang được sử dụng với tiểu trình này.
- Đây là thời điểm để giải phóng các biến dùng cho tiểu trình



LIÊN KẾT DLL VỚI ỨNG DỤNG WIN32 API

- ❖ Tạo Project loại **"Win32 Dynamic Link Library"**, với tên **"MyWin32DLL"**
- ❖ Tạo Project loại **"Win32 Application"** với tên là **"MyWin32Sample"**
- ❖ Biên dịch và chạy chương trình
- ❖ Copy tập tin **MyWin32DLL.DLL** vào thư mục Debug của **"MyWin32Sample"**
- ❖ Copy tập tin **MyWin32DLL.LIB** và tập tin **MyWin32DLL.H** vào thư mục **MyWin32Sample**
- ❖ Add tập tin **MyWin32DLL.LIB** vào project **MyWin32Sample**



XÂY DỰNG DLL DÙNG THƯ VIỆN MFC

- ❖ Tạo lập DLL
- ❖ Phân tích các khai báo
- ❖ Định nghĩa hàm export
- ❖ Các dạng thư viện DLL của MFC
- ❖ Liên kết DLL với ứng dụng



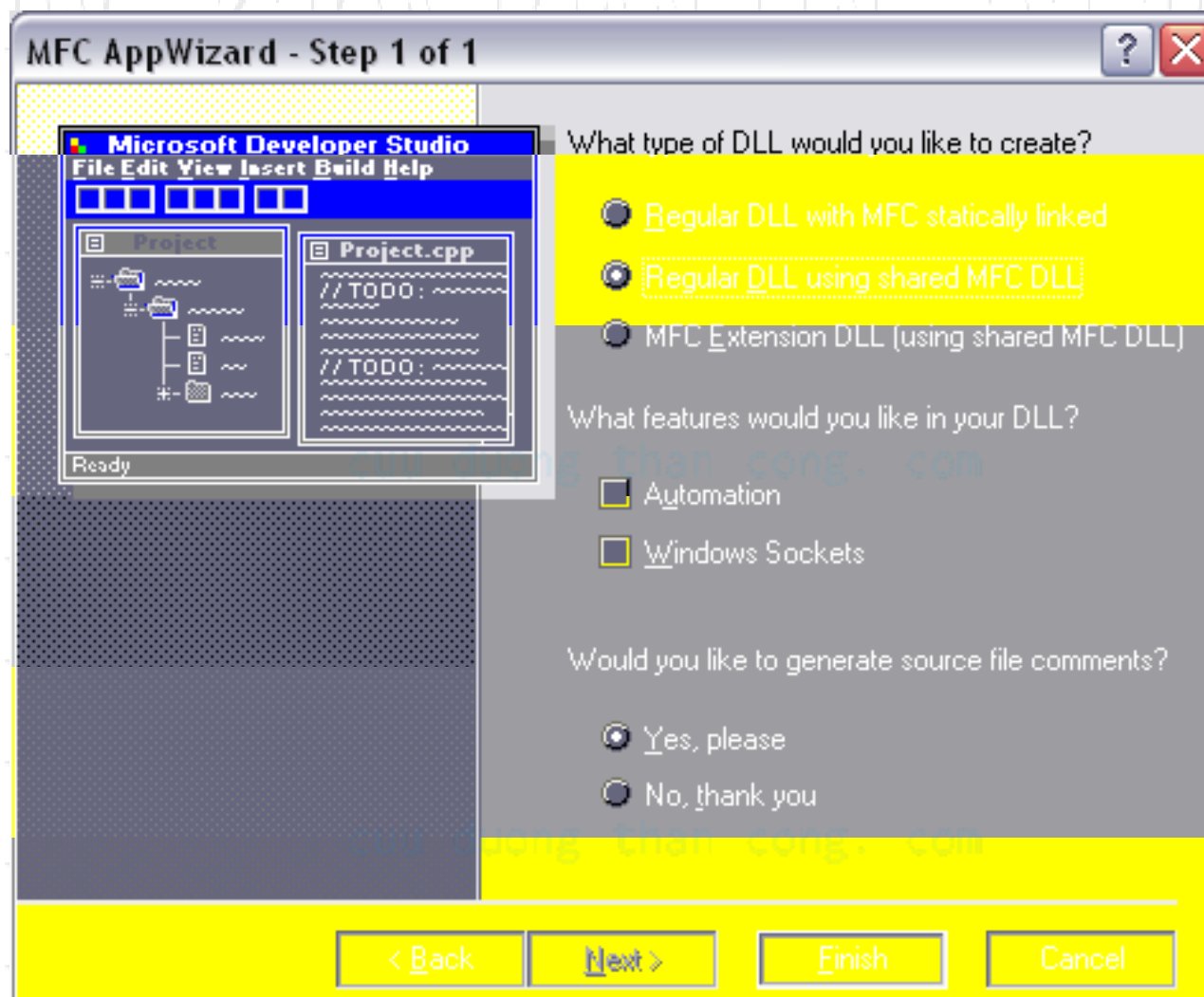
XÂY DỰNG DLL DÙNG THƯ VIỆN MFC

- ❖ Chọn menu **File** → **New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project "**MFC AppWizard (dll)**"
- ❖ Đặt tên project trong ô "**Project name**"
- ❖ Xác định đường dẫn thư mục trong "**Location**"
- ❖ Step 1: Chọn loại ứng dụng "**Regular DLL using shared MFC DLL**" hoặc "**MFC Extension DLL**"
- ❖ Nhấn "**Finish**" để kết thúc



CÁC DẠNG THƯ VIỆN DLL CỦA MFC

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL



CÁC DẠNG THƯ VIỆN DLL CỦA MFC

■ Regular DLL:

- Chỉ export các hàm theo dạng C-style, không thể export class, hàm thành phần của class, các hàm định nghĩa chồng (overloaded function)
- Các ứng dụng Win32 và MFC đều có thể sử dụng loại DLL
- “with MFC statically linked”: DLL sẽ được liên kết với các DLL chuẩn của MFC theo cách liên kết tĩnh
- “using shared MFC DLL”: DLL sẽ được liên kết với các DLL chuẩn của MFC theo cách liên kết động

■ MFC Extention DLL:

- Cho phép export các class. Ứng dụng khác có thể tạo các project từ các class, hoặc xây dựng các lớp kế thừa từ class này.
- DLL sẽ được liên kết với các DLL chuẩn của MFC theo cách liên kết động
- Chỉ có các ứng dụng MFC mới có thể dùng với loại DLL này.



CÁCH THỨC GỌI DLL TRONG ỨNG DỤNG

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
THƯ VIỆN LIÊN KẾT ĐỘNG - DLL

- ❖ Cách thức tìm kiếm file DLL
- ❖ Load-time Dynamic Linking
- ❖ Run-time Dynamic Linking



ỨNG DỤNG TÌM KIẾM FILE DLL

❖ **Hệ điều hành Windows sẽ tìm file DLL có sử dụng trong ứng dụng theo thứ tự các thư mục sau**

- Thư mục chứa tập tin EXE gọi DLL này
- Thư mục hiện hành
- Thư mục system, system32
- Thư mục Windows
- Các thư mục được liệt kê trong biến môi trường PATH



CÁCH THỨC GỌI DLL TRONG ỨNG DỤNG – Load-time

- ❖ Ứng dụng sẽ gọi hàm export của DLL một cách “tường minh” (giống như cách gọi hàm của thư viện liên kết tĩnh)
- ❖ Ứng dụng cần phải liên kết với file thư viện nhập (Import Lib) của DLL
- ❖ Ứng dụng sẽ nạp DLL ngay thời điểm đầu tiên chương trình chạy
- ❖ Ưu điểm:
 - Đơn giản, dễ sử dụng vì giống như cách dùng thư viện tĩnh
- ❖ Khuyết điểm:
 - Nếu không tìm ra DLL lúc nạp, ứng dụng sẽ kết thúc ngay



❖ Cách thực hiện

- Add file thư viện nhập (DLLName.Lib) vào project
- Khai báo các hàm, biến IMPORT từ DLL
- Gọi hàm của thư viện DLL như gọi hàm bình thường



CÁCH THỨC GỌI DLL TRONG ỨNG DỤNG – Run-time

- ❖ Ứng dụng sẽ gọi hàm **LoadLibrary** hay **LoadLibraryEx** để nạp DLL tại thời điểm cần thiết
- ❖ Sau khi DLL được nạp, ứng dụng sẽ dùng hàm **GetProcAddress** để lấy địa chỉ của hàm export trong DLL
- ❖ Ứng dụng sẽ gọi hàm export trong DLL bằng cách dùng **con trỏ hàm trả về từ hàm GetProcAddress**
- ❖ Ưu điểm:
 - Cách này không cần dùng đến file thư viện nhập (DllName.LIB)
 - Có thể xử lý lỗi không tìm thấy DLL, tránh kết thúc ứng dụng tức thời, thể hiện thông báo lỗi do ứng dụng qui định
- ❖ Khuyết điểm:
 - Phức tạp, khó sử dụng trong cài đặt



CÁCH THỨC GỌI DLL TRONG ỨNG DỤNG – Run-time

❖ Ví dụ: gọi hàm MyFunction(LPTSTR) từ thư viện MYDLL.DLL

```
typedef VOID (*MYPROC) (LPTSTR);
```

```
HINSTANCE hInstLib ;
```

```
MYPROC ProcAddr;
```

```
hInstLib = LoadLibrary("MYDLL");
```

```
if (hInstLib !=NULL)
```

```
{
```

```
    ProcAddr = (MYPROC) GetProcAddress(hInstLib,"MyFunction");
```

```
    if (ProcAddr !=NULL)
```

```
    {
```

```
        (ProcAddr)("A parameter string") ;
```

```
        FreeLibrary(hInstLib);
```

```
    }
```

```
}
```



❖ Xây dựng thư viện MFC Extension DLL **MyFirstDLL**

```
class AFX_EXT_CLASS CMyClass : public CObject
{
public:
    void SetFirstName(CString s);
    void ShowMessageBox();
    CMyClass();
    virtual ~CMyClass();

protected:
    CString m_FirstName;
};
```



- ❖ Xây dựng thư viện MFC Extension DLL **MyFirstDLL**

```
CMyClass::CMyClass()
{
    m_FirstName="MyFirstDLL Sample";
}

void CMyClass::ShowMessageBox()
{
    AfxMessageBox(m_FirstName);
}

void CMyClass::SetFirstName(CString s)
{
    m_FirstName = s;
}
```



- ❖ Tạo ứng dụng **MyFirstDLLSample** sử dụng thư viện **MyFirstDLL**

```
void CMFCSampleDllView::OnLButtonDown(UINT nFlags, CPoint point)
{
```

```
    // TODO: Add your message handler code here and/or call default
    CMyClass my;
    my.ShowMessageBox();
```

```
    my.SetFirstName("abc");
    my.ShowMessageBox();
```

```
    CView::OnLButtonDown(nFlags, point);
```

```
}
```



XÂY DỰNG THƯ VIỆN DLL BẰNG MFC

- ❖ Thay đổi hàm **ShowMessageBox** trong thư viện **MyFirstDLL**

```
void CMyClass::ShowMessageBox()  
{  
    AfxMessageBox(m_FirstName);  
}
```

```
void CMyClass::ShowMessageBox()  
{  
    AfxMessageBox(m_FirstName);  
    AfxMessageBox("DllSample: "+ m_FirstName);  
}
```

- ❖ Rebuild thư viện **MyFirstDLL**
- ❖ Copy tập tin **MyFirstDLL.DLL** vào thư mục **MyFirstDLLSample\Debug**



XÂY DỰNG THƯ VIỆN DLL BẰNG MFC

❖ Run chương trình **MyFirstDLLSample**

Điều gì xảy ra khi click
chuột trái lên màn
hình chương trình
MyFirstDLLSample
?

```
void CMFCSampleDllView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CMyClass my;
    my.ShowMessageBox();

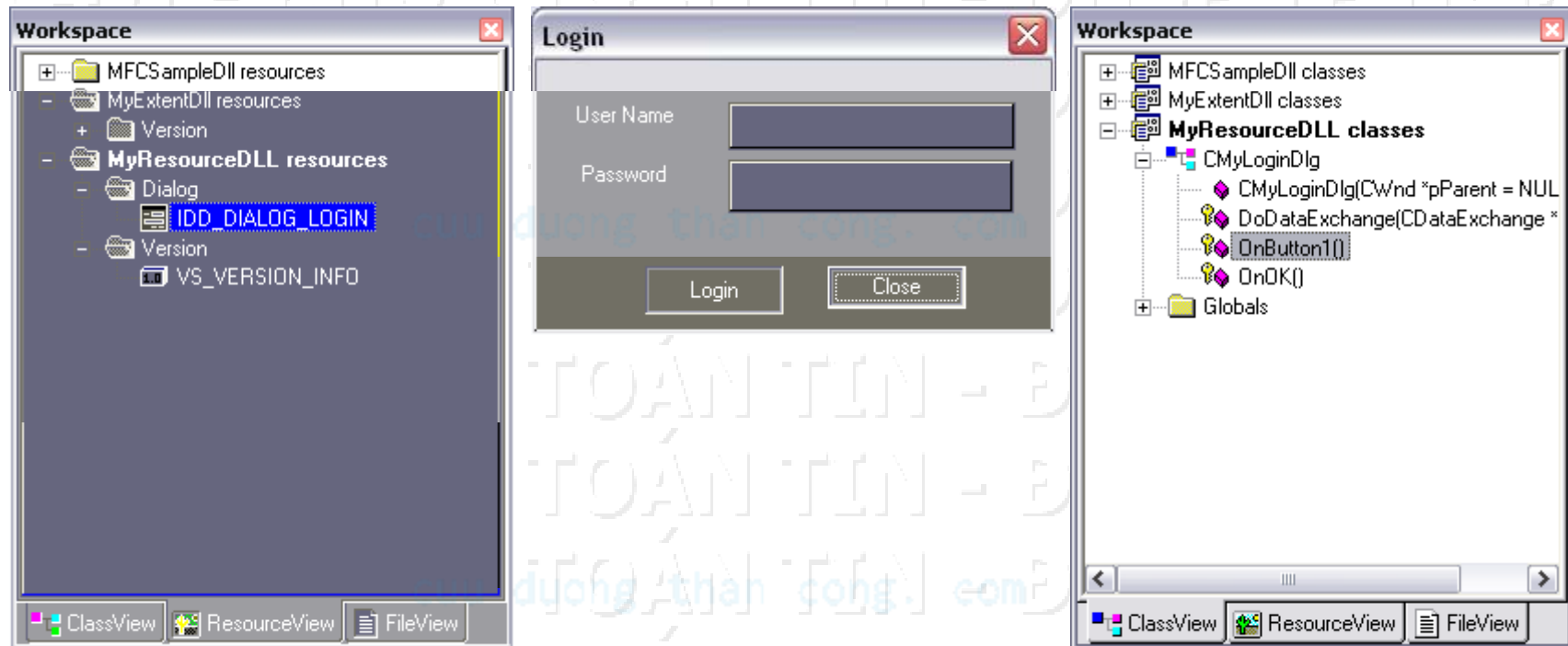
    my.SetFirstName("abc");
    my.ShowMessageBox();

    CView::OnLButtonDown(nFlags, point);
}
```



XÂY DỰNG THƯ VIỆN DLL BẰNG MFC

- ❖ Xây dựng thư viện **MySecondDLL** MFC Extension DLL có sử dụng resource



❖ Liên kết **MySecondDLL** với ứng dụng **MyFirstDLLSample**

```
#include "MyLoginDlg.h"
```

```
void CMFCSampleDllView::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CMyLoginDlg dlg;
    dlg.DoModal();

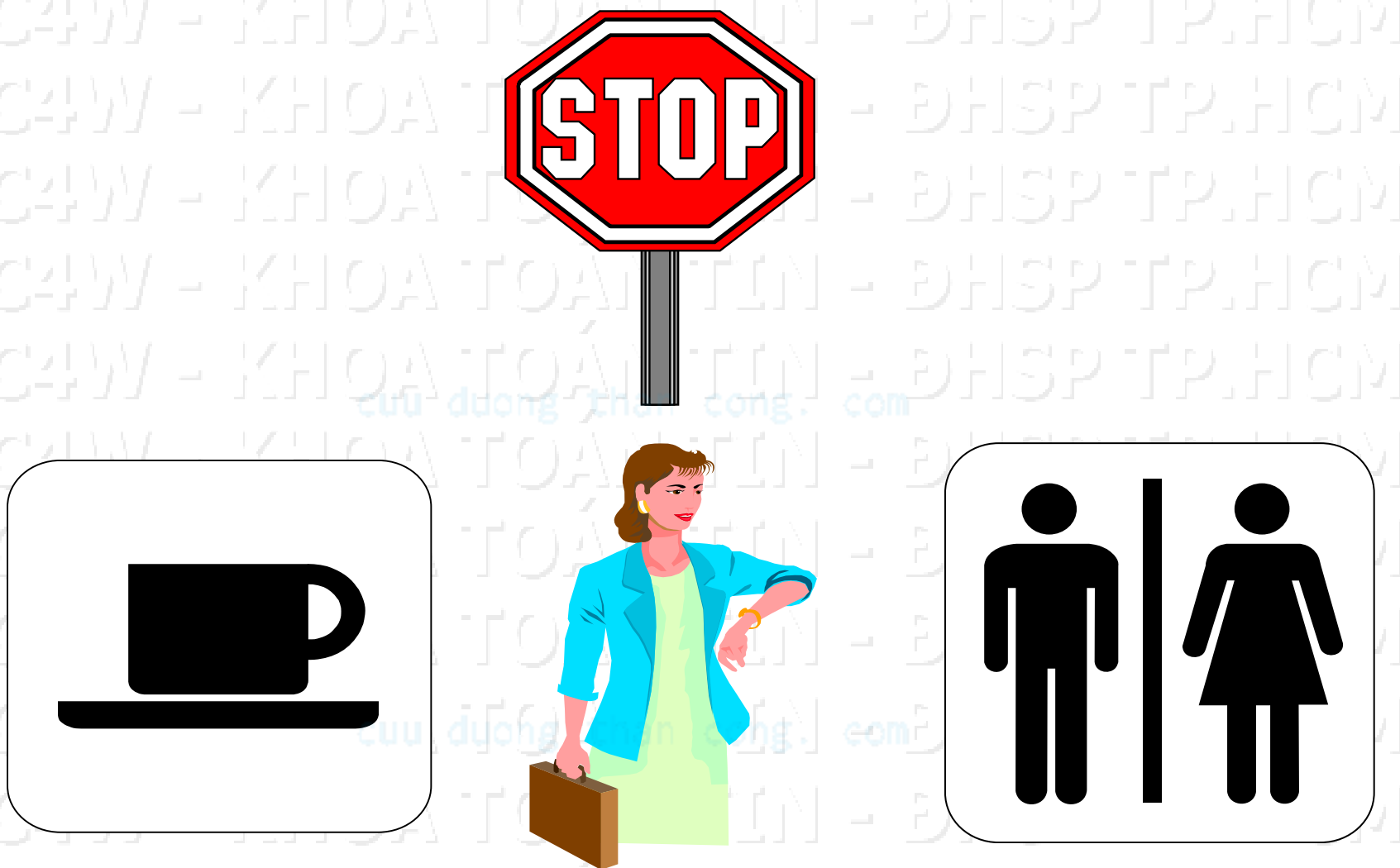
    CView::OnRButtonDown(nFlags, point);
}
```



TÀI LIỆU THAM KHẢO

- ❖ Charles Petzold (1998), *Programming Windows Fifth Edition*, Microsoft Press – Chapter 21
- ❖ David J. Kruglinski (1998), *Programming Microsoft Visual C++ Fifth Edition*, Microsoft Press – Chapter 22





Tạo báo biểu với Crystal Report

- Giới thiệu các công cụ tạo report
- Thiết kế Report với Crystal Report
- Liên kết Report với ứng dụng

tnbao.dhsp@gmail.com

Tạo báo biểu với Crystal Report

- **Giới thiệu các công cụ tạo report**
- **Thiết kế Report với Crystal Report**
- **Liên kết Report với ứng dụng**

tnbao.dhsp@gmail.com

Giới thiệu mẫu báo biểu trong ứng dụng

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT

Preview Employee Profiles

Help

100% 1 of 2+

Preview

- B
- C
- D
- F
- H
- K
- L
- M
- P
- S

Employee Profiles

Print Date/Time: 11/12/2005 - 12:23:29PM



Brid, Justin



Name: Justin Brid

Position: Marketing Director

Extention: 377

Birthday:

Home Address: 2 Impasse du Soleil
Haguenau, Bas-Rhin
France
67500

Home Phone: (80-1662

Notes: Justin Brid graduated from HEC Paris Hautes Etudes Commerciales de Paris in 1986 with high honors. Mr. Brid attended the American Graduate School of International Management Thunderbird in Glendale, Arizona. Prior to his employment with Northwind Traders, he held positions with several Import/Export companies Culinary Specialties in Germany, Switzerland, and Austria. Mr. Brid is equally fluent in French and German; he also speaks English and reads Spanish.

Exit



Giới thiệu mẫu báo biểu trong ứng dụng

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT

Trường Cao Đẳng SP Bình Dương

Hồ Sơ nhân viên



Mã nhân viên	2
Tên nhân viên	Fuller Andrew
Địa chỉ	908 W. Capital Way
Điện thoại	(206) 555-9482



Giới thiệu mẫu báo biểu trong ứng dụng

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS LẬP TRÌNH VỚI CRYSTAL REPORT

1 of 1 100% Total:0

ỦY BAN NHÂN DÂN TP.HCM
PHÒNG THANH TRA
153/2 Hoàng Văn Thụ
ĐT: 9321213
-----oO-----

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự Do - Hạnh Phúc
-----oO-----

TP.Hồ Chí Minh, ngày tháng năm

BÁO CÁO DANH SÁCH HỒ SƠ CHƯA GIẢI QUYẾT

Loại hồ sơ : Dân sự

Từ ngày : 11/12/2002 Đến ngày: 15/12/2002

Ngày	Số hồ sơ	Người gửi	Địa chỉ	Điện thoại
Quận : Bình Thạnh		Số lượng hồ sơ : 3		
11/12/2002	HSDS100	Bùi Văn Lý	123 Phan Đăng Lưu	0908123456
12/12/2002	HSDS101	Nguyễn Công Phú	126 Lê Quang Định	
14/12/2002	HSDS130	Nguyễn Văn Hùng	345 Phan Văn Trị	
Quận : Phú Nhuận		Số lượng hồ sơ : 3		
11/12/2002	HSDS100	Bùi Xuân Giang	96 Huỳnh Văn Bánh	9321345-9632145
12/12/2002	HSDS101	Trịnh Minh Thành	126 Hai Bà Trưng	8245678
14/12/2002	HSDS130	Nguyễn Niềm Thoại Uyên	165 Nguyễn Kiệm	8456012 - 0903122345
Tổng cộng: 2 quận		Số lượng hồ sơ : 6		



Một số công cụ tạo Report

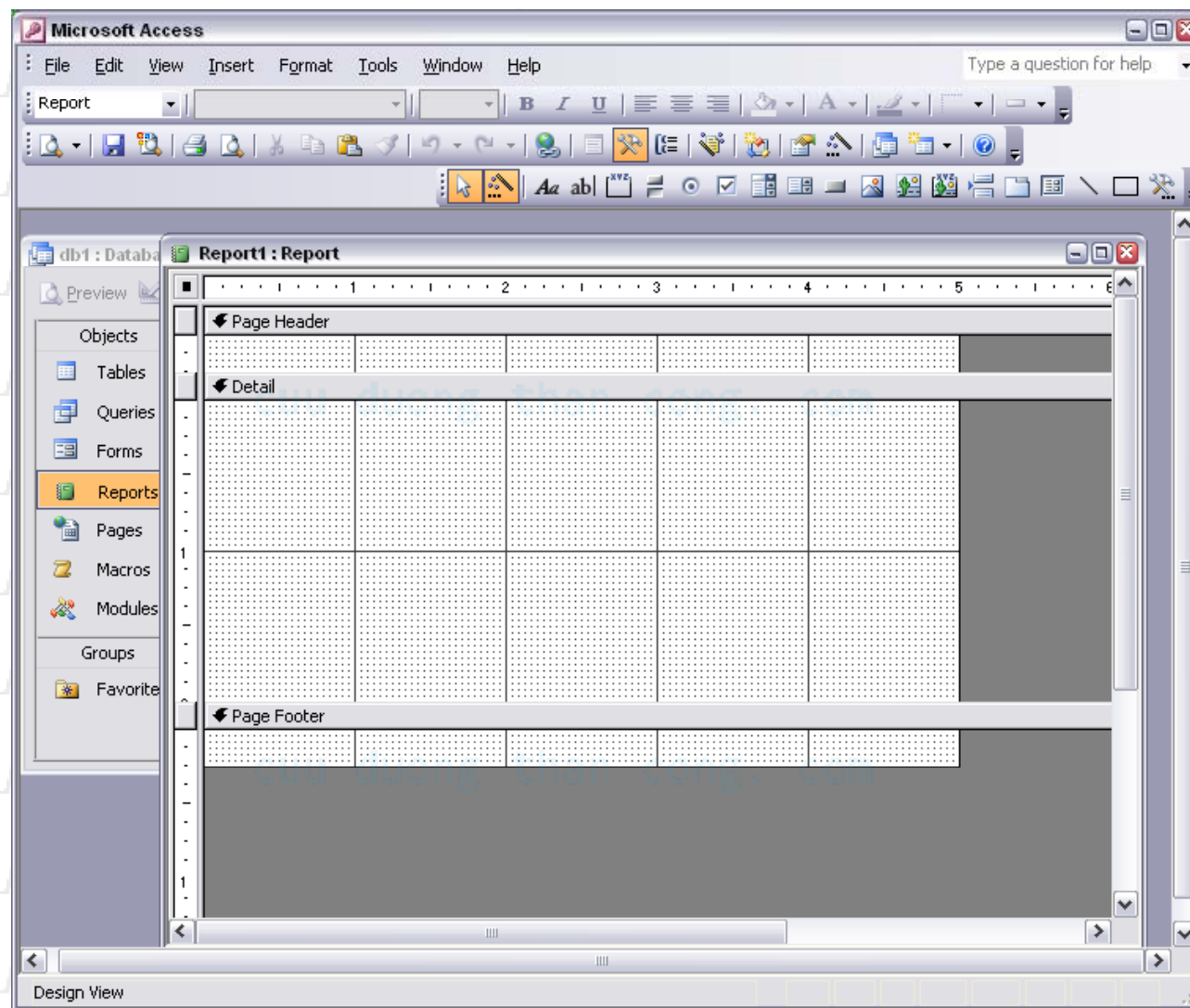
- ❖ **MS Access 2000/XP/2003**
- ❖ **Crystal Report 8.5/9/10/11**
- ❖ **ActiveReport - DataDynamics**
- ❖ **C1Report - ComponentOne**



Một số công cụ tạo Report

❖ MS Access 2000/XP/2003

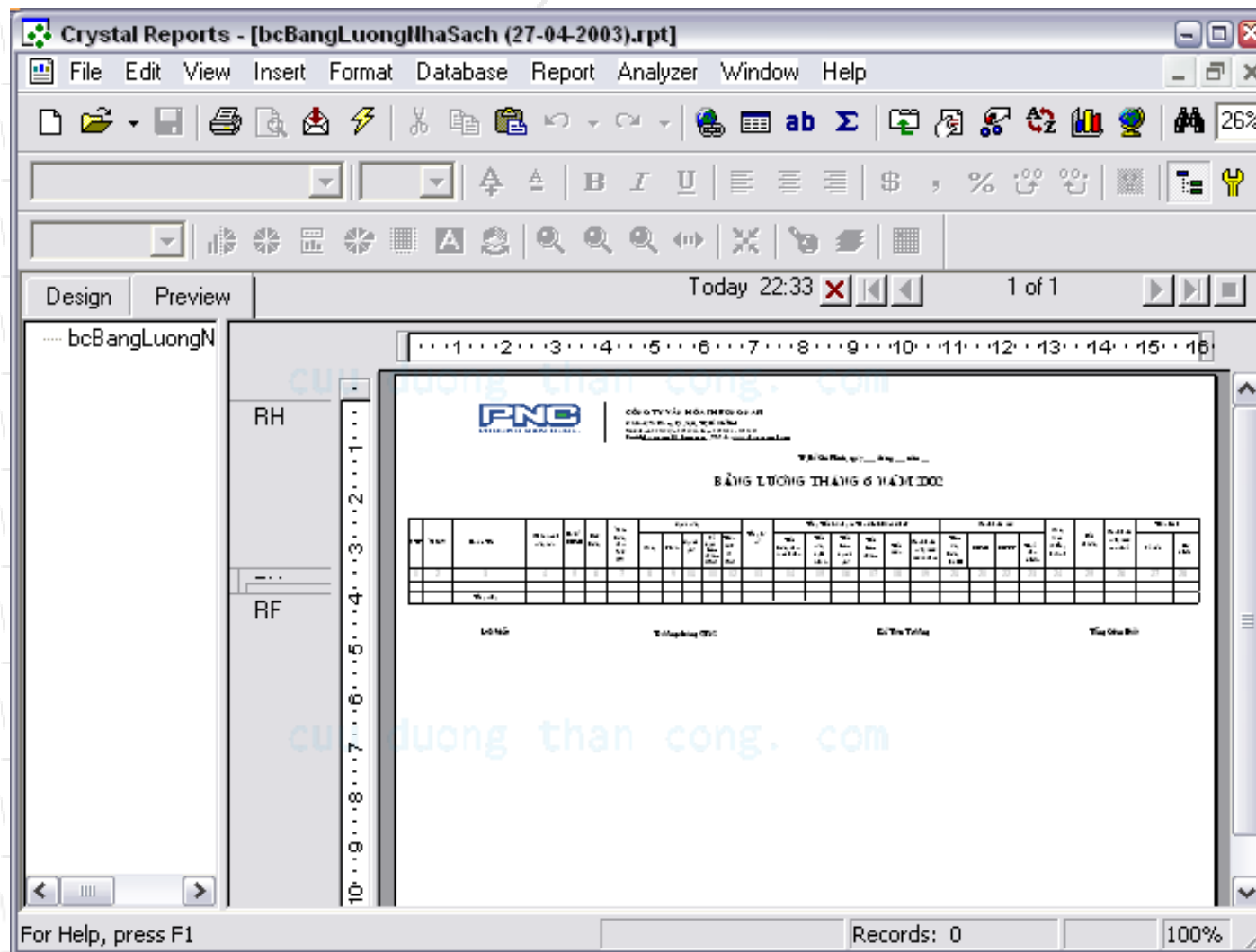
BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT



Một số công cụ tạo Report

❖ Crystal Report 8.5/9/10/11

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT



Một số công cụ tạo Report

❖ ActiveReport - DataDynamics

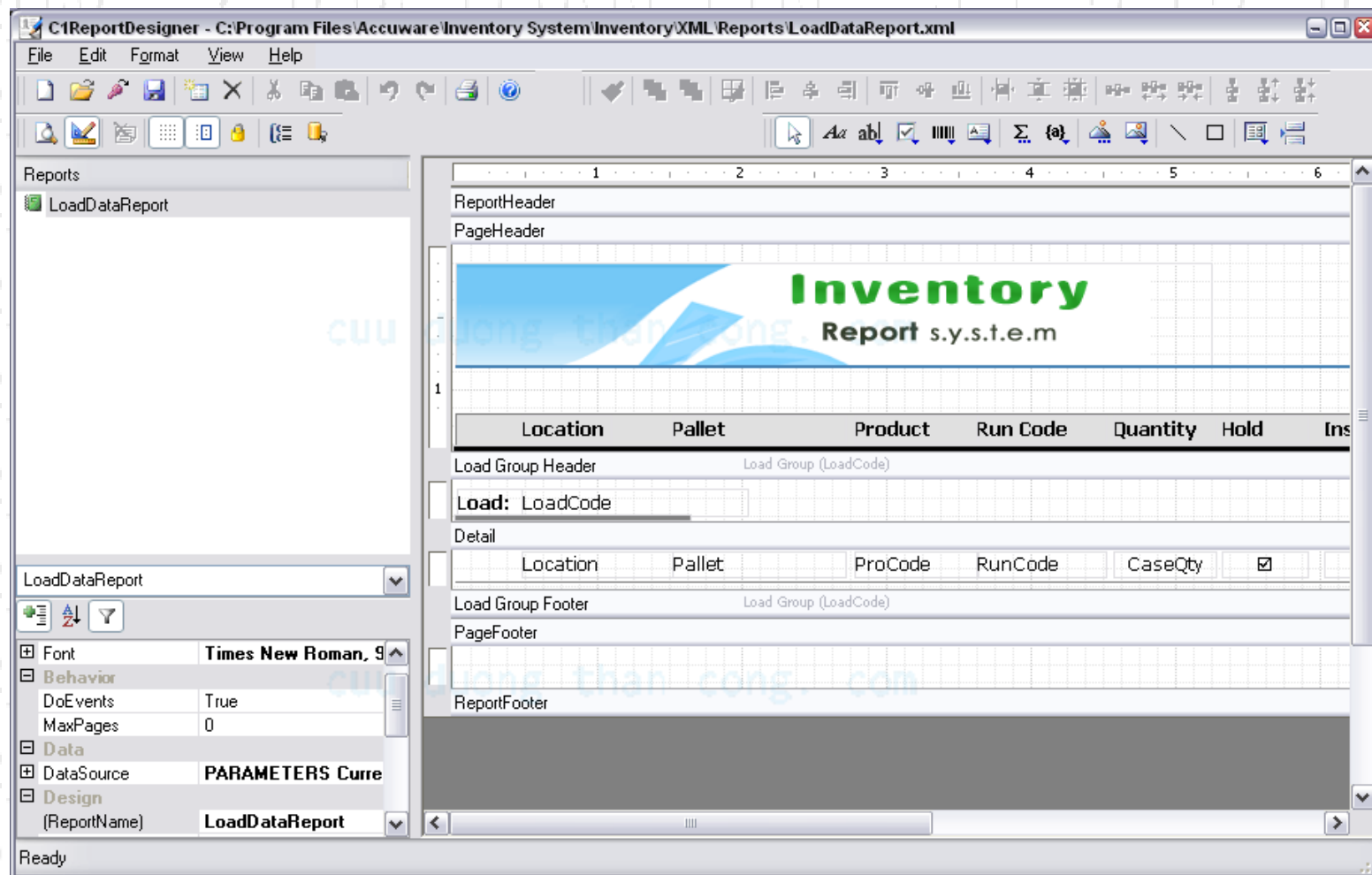


BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT



Một số công cụ tạo Report

❖ C1Report - ComponentOne



Crystal Report và ứng dụng

- **Bước 1: Thiết kế report**

sử dụng công cụ **Crystal Report** tạo và các report thành các tập tin ***.rpt**

- **Bước 2: Liên kết report với ứng dụng**

sử dụng **Crystal Report Control** để liên kết các tập tin report ***.rpt** vào ứng dụng



Tạo báo biểu với Crystal Report

- Giới thiệu các công cụ tạo report
- **Thiết kế Report với Crystal Report**
- Liên kết Report với ứng dụng

tnbao.dhsp@gmail.com

Thiết kế Report với Crystal Report

- ❖ **Run công cụ Crystal Report để tạo các tập tin report**
- ❖ **Liên kết với Data Source (MS Access/SQL Server,...)**
- ❖ **Sử dụng Formula**
- ❖ **Lọc dữ liệu**
- ❖ **Sắp xếp/gom nhóm dữ liệu**

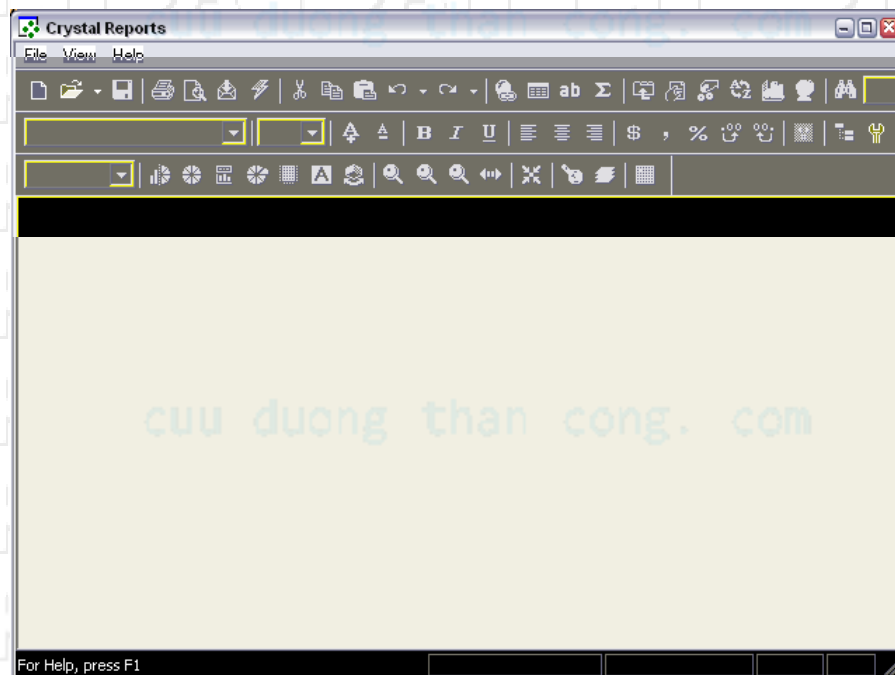


Thiết kế Report với Crystal Report

BÀI GIẢNG HỌC PHẦN LẬP TRÌNH WINDOWS
LẬP TRÌNH VỚI CRYSTAL REPORT

❖ Run công cụ Crystal Report để tạo các tập tin report (*.rpt)

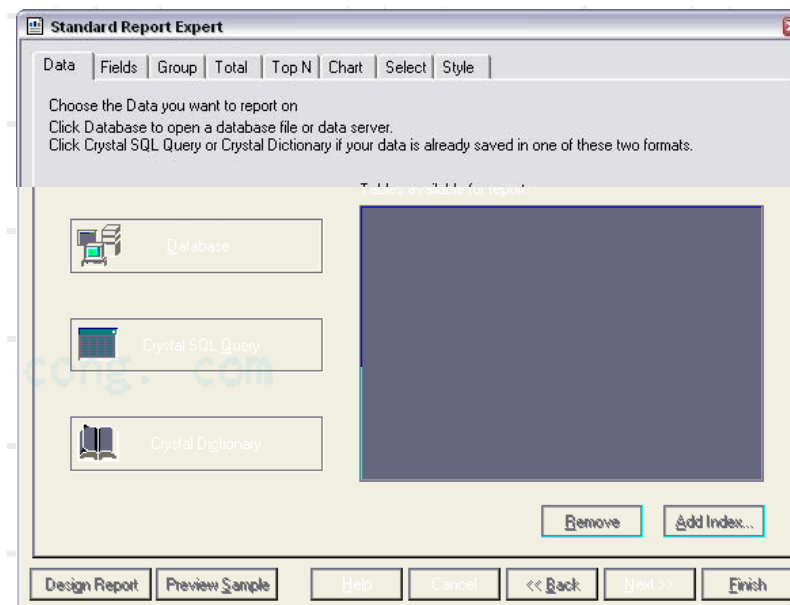
Khởi động chương trình Crystal Report bằng cách chọn lần lượt **Start/Programs/Seagate Crystal Report Tools/Crystal Reports**



Thiết kế Report với Crystal Report

❖ Liên kết với Data Source (MS Access/SQL Server,...)

- ❑ Chọn menu **File** → **New**
- ❑ Chọn **Standard**
- ❑ Click button **"Database"** để chọn Data source cho report



Tạo báo biểu với Crystal Report

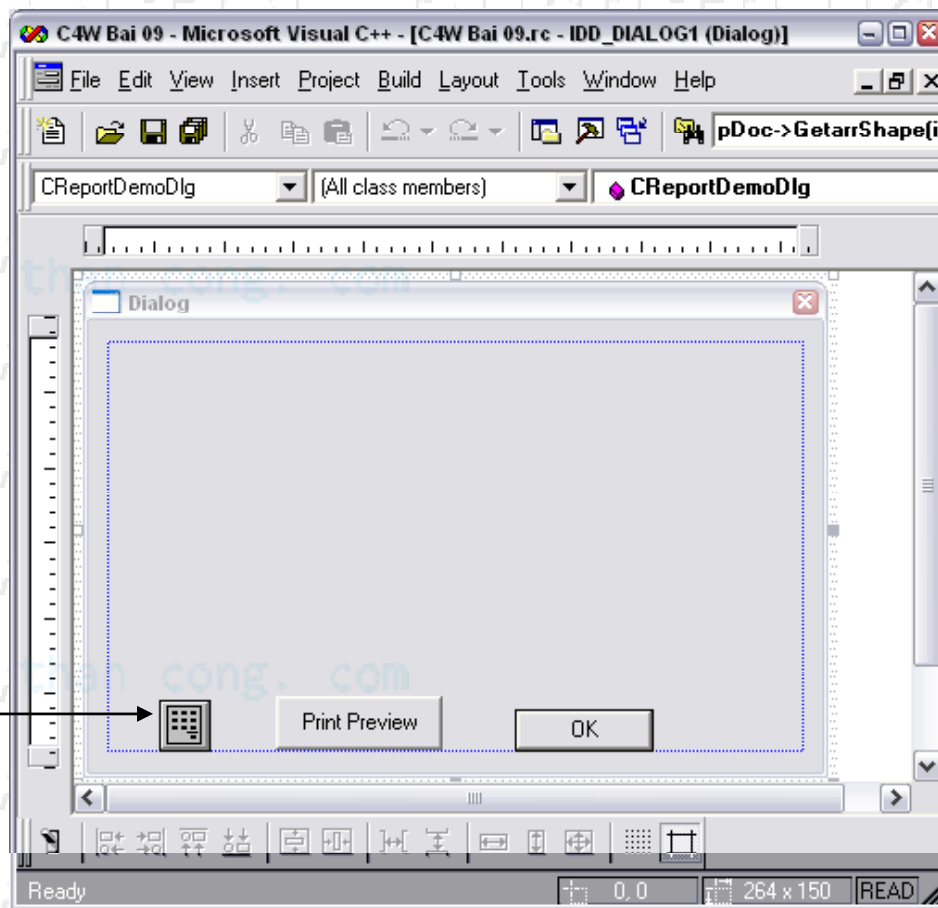
- Giới thiệu các công cụ tạo report
- Thiết kế Report với Crystal Report
- Liên kết Report với ứng dụng

tnbao.dhsp@gmail.com

Liên kết report vào ứng dụng

- Sử dụng **Crystal Report Control** trong danh sách ActiveX controls để liên kết các tập tin report (*.rpt) được thiết kế bằng công cụ **Crystal Report**

Crystal Report Control

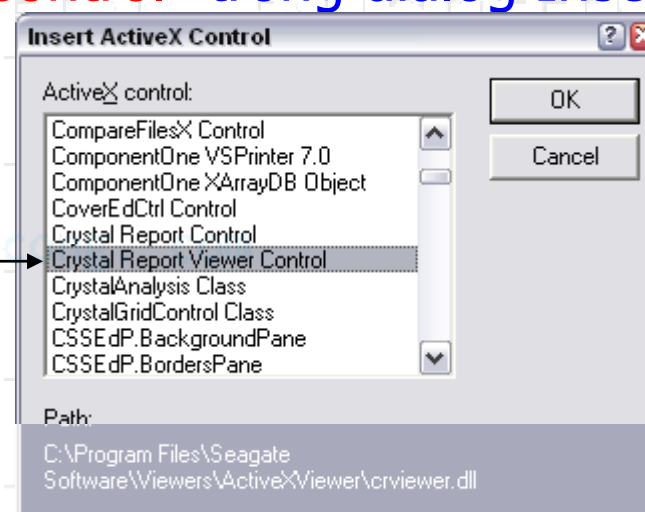


Liên kết report vào ứng dụng

- Sử dụng **Crystal Report Control** trong danh sách ActiveX controls để liên kết các tập tin report (*.rpt) được thiết kế bằng công cụ **Crystal Report**

- ❑ Click phải chuột lên Dialog
- ❑ Chọn mục "Insert ActiveX Control" trong popup menu
- ❑ Chọn mục "Crystal Report Control" trong dialog Insert ActiveX Control

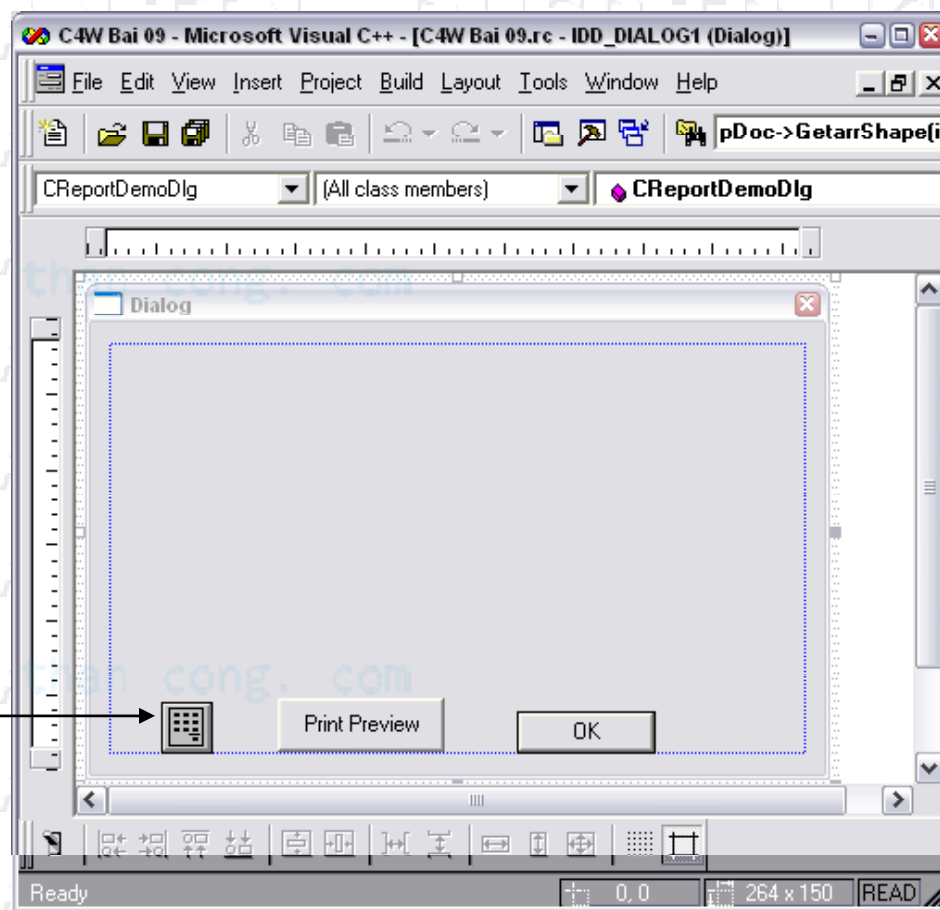
Crystal Report Control



Liên kết report vào ứng dụng

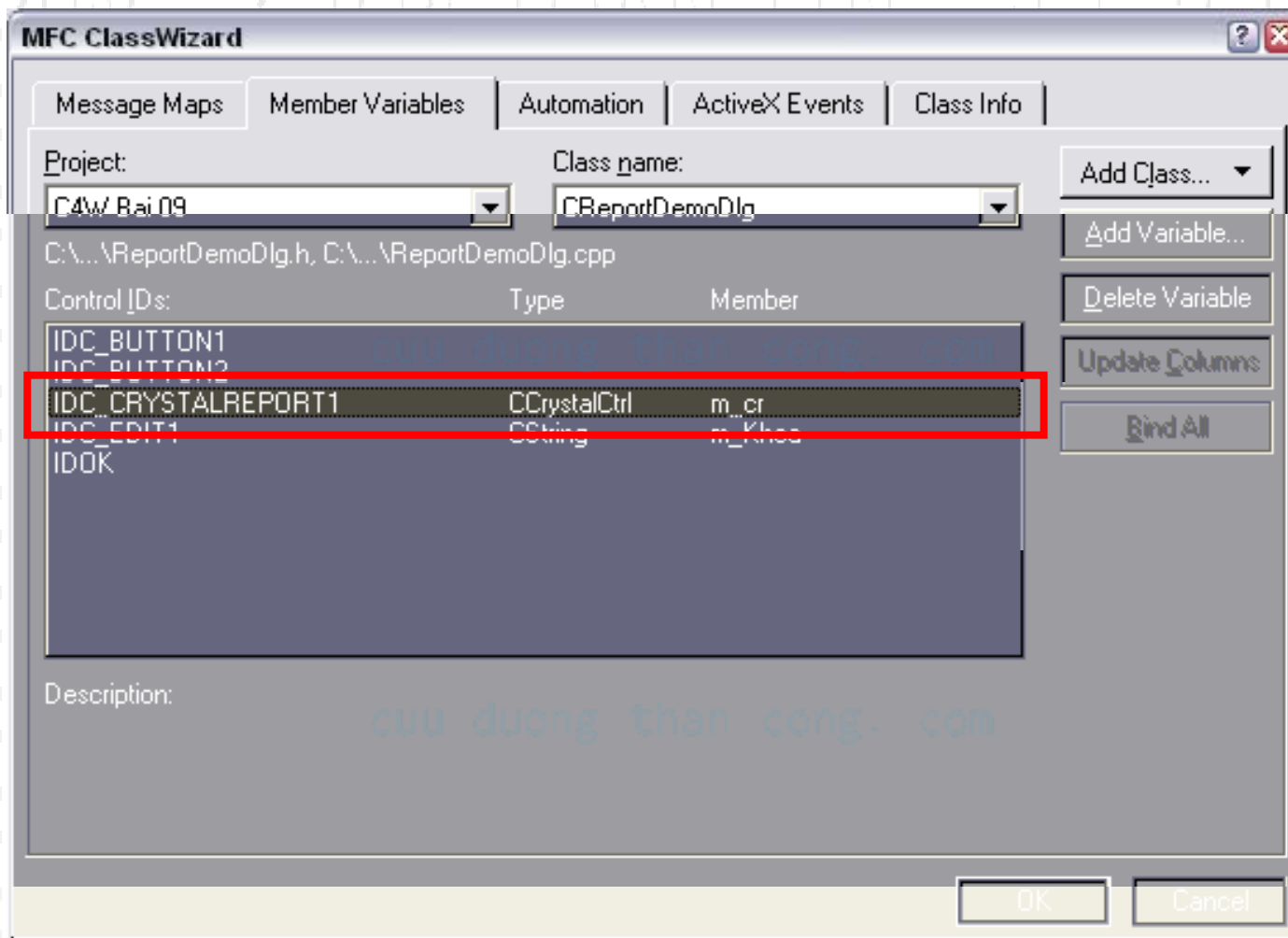
- Sử dụng **Crystal Report Control** trong danh sách ActiveX controls để liên kết các tập tin report (*.rpt) được thiết kế bằng công cụ **Crystal Report**

Crystal Report Control



Sử dụng Crystal Report Control

❖ Khai báo biến xử lý Report



Sử dụng Crystal Report Control

❖ Xác định tập tin *.Rpt

❑ Cú pháp

```
CString sTenFileReport;  
m_cr.SetReportFileName(sTenFileReport);
```

❑ Ví dụ

```
void CReportDemoDlg::OnPrintPreview()  
{  
    CString sTenFileReport = "..\\Reportsv.rpt";  
    m_cr.SetReportFileName(sTenFileReport);  
    .....  
}
```



Sử dụng Crystal Report Control

❖ Hiển thị report (Preview)

❑ Cú pháp

```
m_cr.SetAction (1);
```

❑ Ví dụ

```
void CReportDemoDlg::OnPrintPreview()  
{  
    CString sTenFileReport = "..\\Reportsv.rpt";  
    m_cr.SetReportFileName(sTenFileReport);  
    m_cr.SetAction(1);  
}
```



Sử dụng Crystal Report Control

❖ Report có tham số

❑ Cú pháp

```
m_cr.SetFormulas(Index, "TenFormula = GiaTri")
```

❑ Ví dụ

```
void CReportDemoDlg::OnPrintPreview()  
{  
    CString sTenFileReport = "..\\Reports\\rpt";  
    m_cr.SetReportFileName(sTenFileReport);  
    CString s = "Khoa= ";  
    UpdateData();  
    s = s + "\"" + m_Khoa + "\"";  
    m_cr.SetFormulas(0,s);  
    m_cr.SetAction(1);  
}
```



Sử dụng Crystal Report Control

❖ Thay đổi connectionstring

❑ Cú pháp

```
CString sConnect = "DSN=" + sServer + ";UID=" + sUser & ";PWD=" +  
sPass + ";DSQ=" + sDBName;  
m_cr.SetConnect(sConnect);
```

❑ Ví dụ

```
void CReportDemoDlg::OnPrintPreview()  
{  
    CString sTenFileReport = "..\\Reportsv.rpt";  
    m_cr.SetReportFileName(sTenFileReport);  
    ///////////////////////////////////  
    CString sConnect = "DSN=" + "tnb-laptop" + ";UID=" + "dhsp" +  
";PWD=" + "dhsp" + ";DSQ=" + "ADO";  
    m_cr.SetConnect(sConnect);  
    ///////////////////////////////////  
    m_cr.SetAction(1);  
}
```



Thiết kế Report với Crystal Report

- ❖ **Sử dụng Formula**
- ❖ **Lọc dữ liệu**
- ❖ **Sắp xếp/gom nhóm dữ liệu**



