

ACLLib

ACLLib

- 是一个基于Win32API的函数库，提供了相对较为简单的方式来做Windows程序
- 实际提供了一个.c和两个.h，可以在MSVC和Dev C++（MinGW）中使用
- 以GPL方式开源放在github上
- 纯教学用途，但是编程模型和思想可以借鉴

版本历史

- 作为浙大竺可桢学院“程序设计综合实践”的实验平台
- 2008年高远
- 2009年崔立崑
- 2010年蓝慧东
- 2012年李睿

Windows桌面程序

Windows API

- 从第一个32位的Windows开始就出现了，就叫做Win32API.
- 它是一个纯C的函数库，就和C标准库一样，使你可以写Windows应用程序
- 过去很多Windows程序是用这种方式做出来的

main()?

- main()成为C语言的入口函数其实和C语言本身无关，你的代码是被一小段叫做启动代码的程序所调用的，它需要一个叫做main的地方
- 操作系统把你的可执行程序装载到内存里，启动运行，然后调用你的main函数

WinMain()

```
#include <windows.h>
int WINAPI WinMain(
HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nCmdShow)
{
    MessageBox(NULL, "Goodbye, cruel world!", "Note",
MB_OK);
    return 0;
}
```

?

- 如何产生一个窗口?
- 如何在窗口中画东西?
- 如何获得用户的鼠标和键盘动作?
- 如何画出标准的界面： 菜单、按钮、输入框
 - acllib目前不能做

答案

```
WNDCLASSEX wc;
wc.cbSize = sizeof(WNDCLASSEX);
wc.style = 0;
wc.lpfnWndProc = WndProc;
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.hInstance = hInstance;
wc.hIcon = LoadIcon(NULL,
    IDI_APPLICATION);
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
wc.hbrBackground = (HBRUSH) (COLOR_WINDOW
    +1);
wc.lpszMenuName = NULL;
wc.lpszClassName = g_szClassName;
wc.hIconSm = LoadIcon(NULL,
    IDI_APPLICATION);
```

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

窗口结构

- 如何 **对初学者来说太难了**

- 如何得到用户的鼠标和键盘动作?

```
case WM_LBUTTONDOWN: {
    char szFileName[MAX_PATH];
    HINSTANCE hInstance = GetModuleHandle(NULL);
    GetModuleFileName(hInstance, szFileName,
        MAX_PATH);
    MessageBox(hwnd, szFileName, "This program is:",
        MB_OK | MB_ICONINFORMATION);
}
break;
```

Win32API 结构

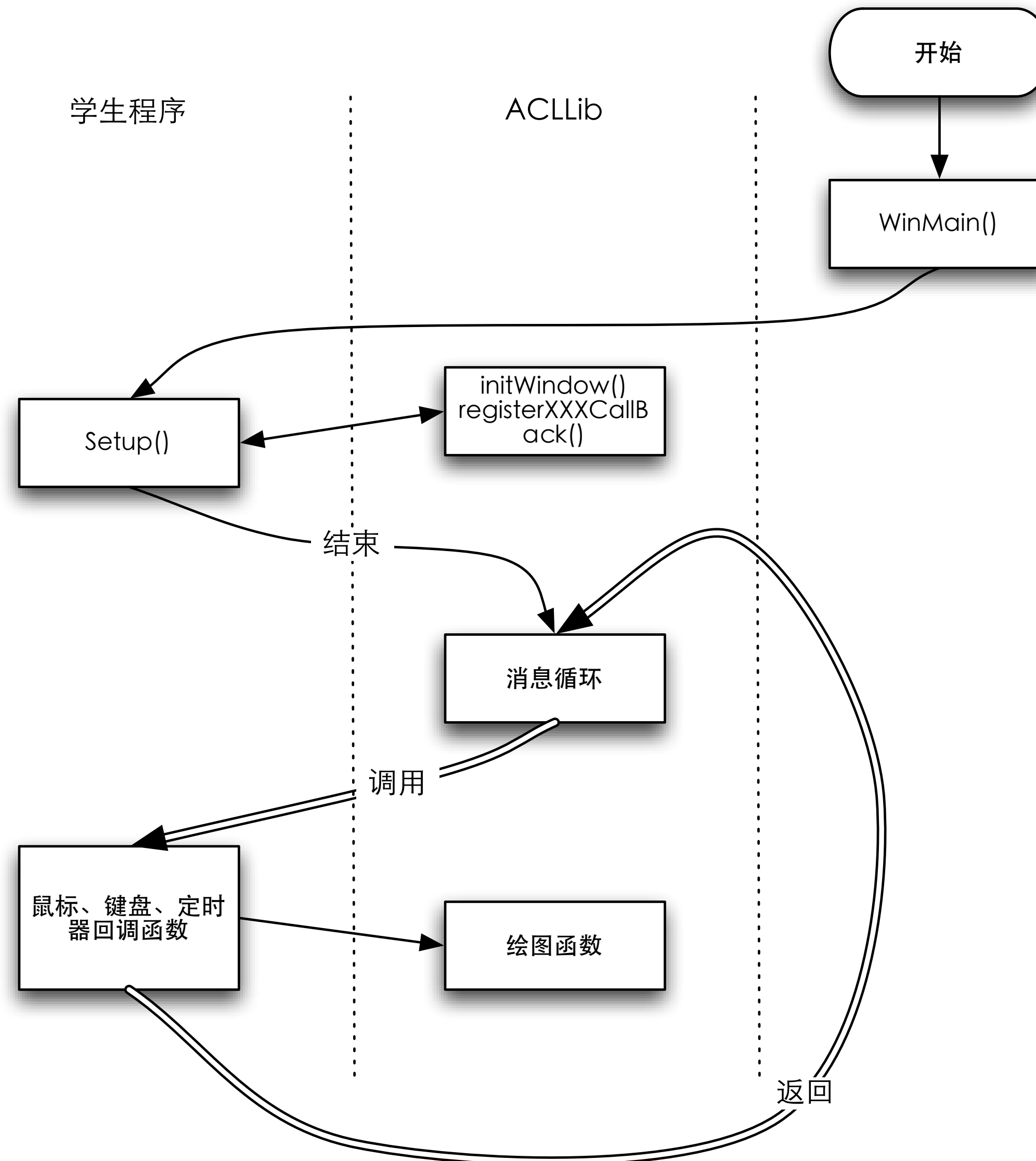
- WinMain()
 - 初始化一个窗口结构然后注册给 Windows OS
 - 运行一个无限循环来读入和处理 Windows 消息
 - 不同的函数来处理不同的消息

ACLLib 结构

- WinMain()
 - 初始化窗口结构并注册给Windows OS
 - 调用你的Setup()
 - 跑一个无限循环来读入并处理Windows消息
- 当有用户动作发生的时候，调用你的回调函数来处理

基本流程

Setup()的结束是程序的开始



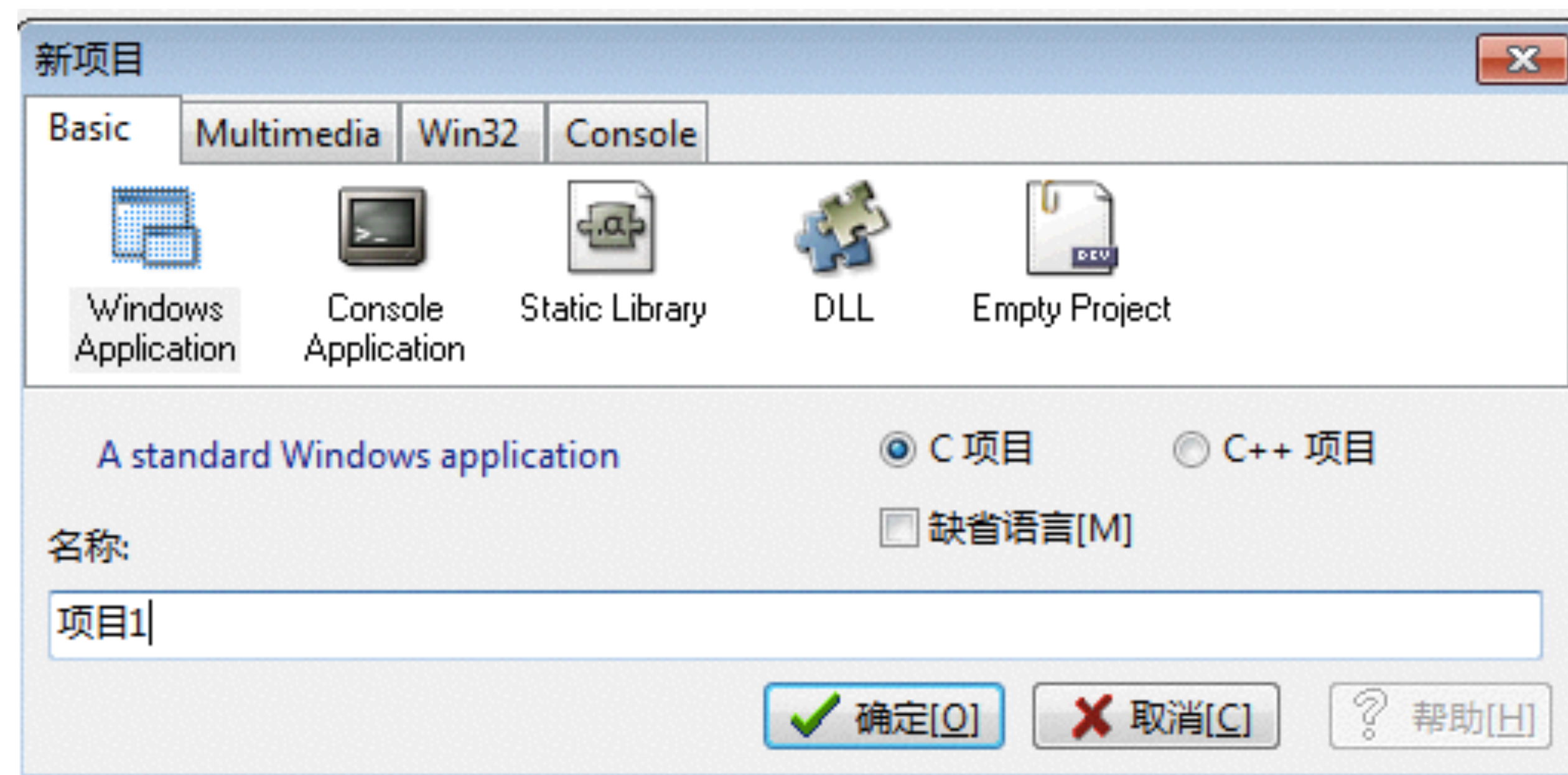
Setup()

- 是用来初始化窗口，注册接收消息的回调函数的
- Setup()的结束不是程序的结束而是开始！

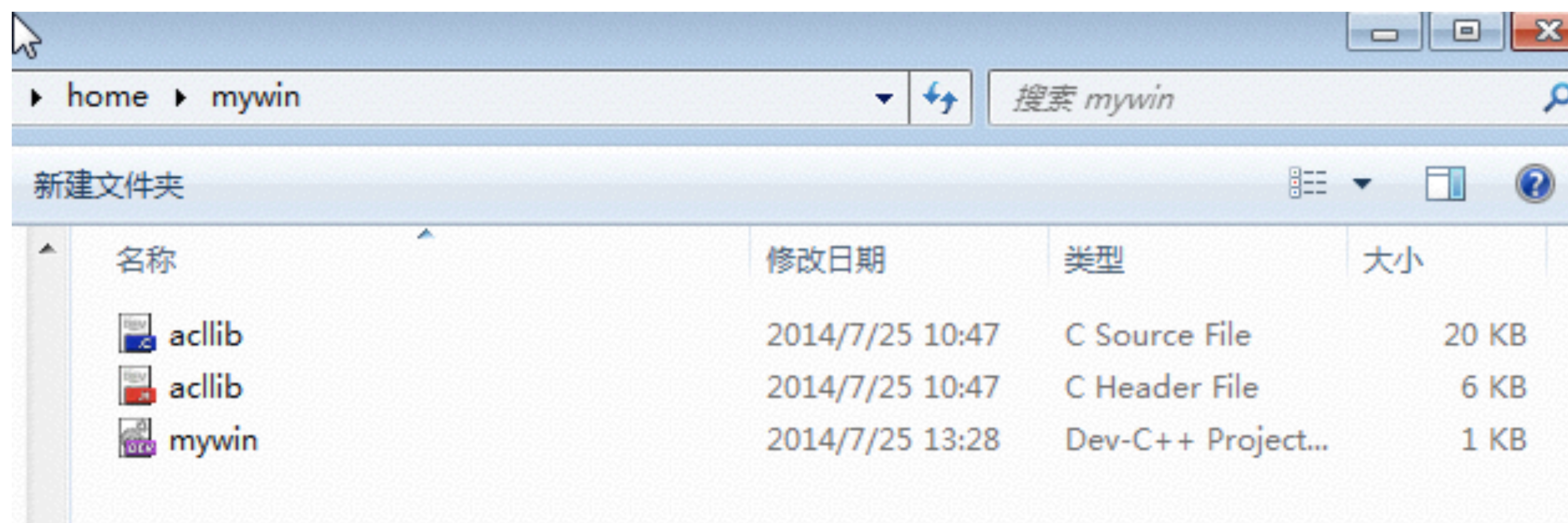
在Dev C++上使用 ACLLib

Dev C++

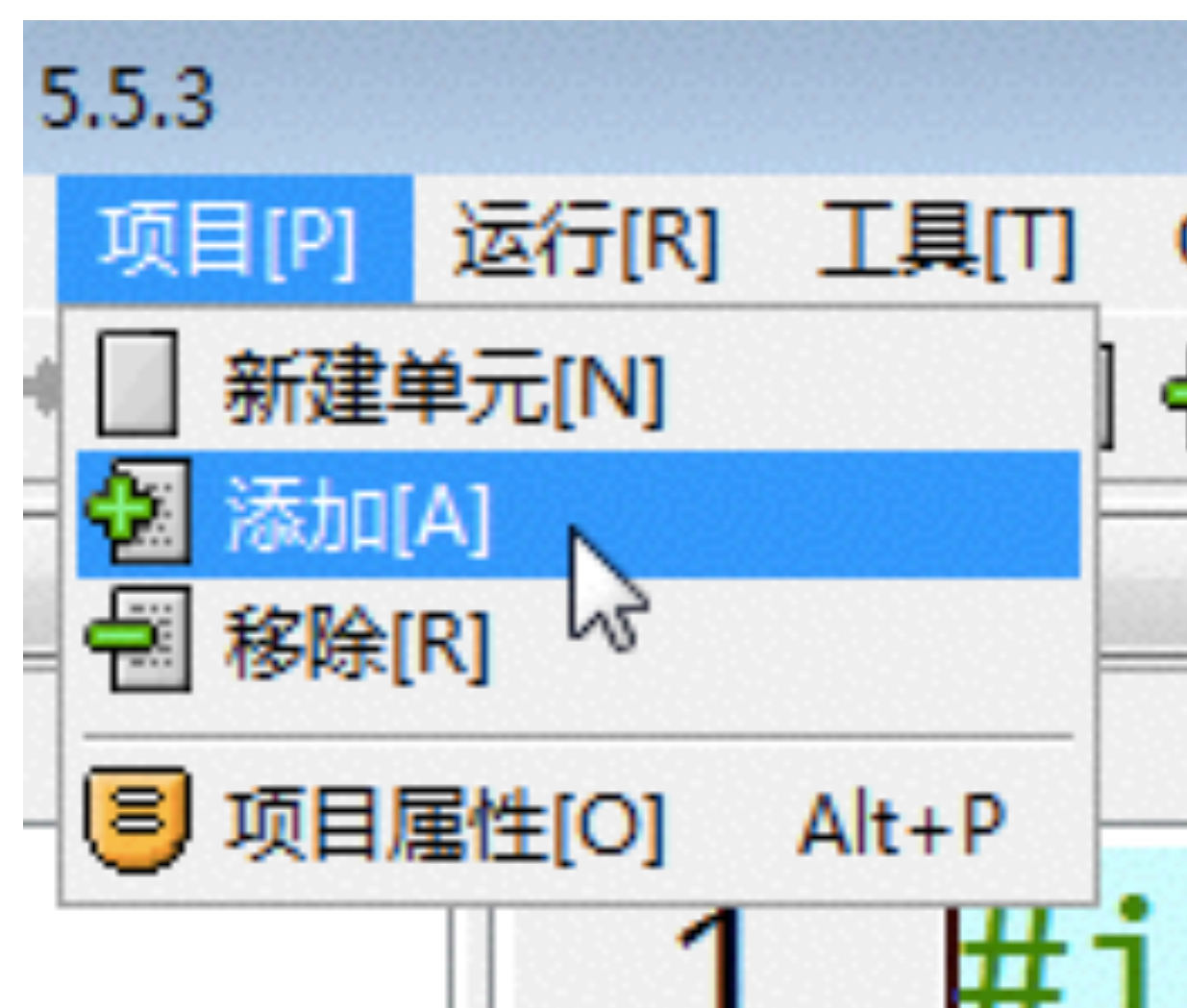
- 要新建一个项目，选择Windows Application类型



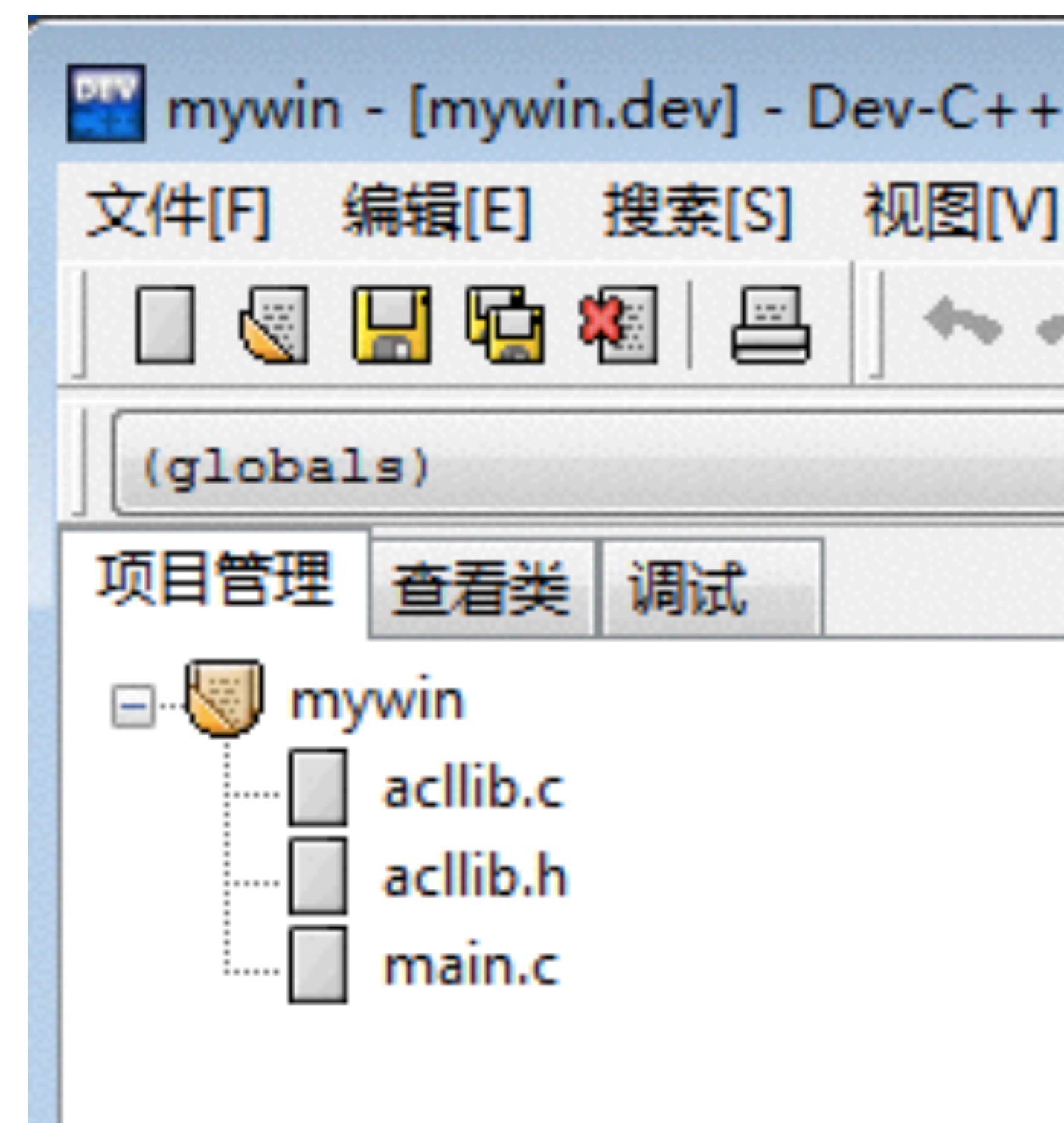
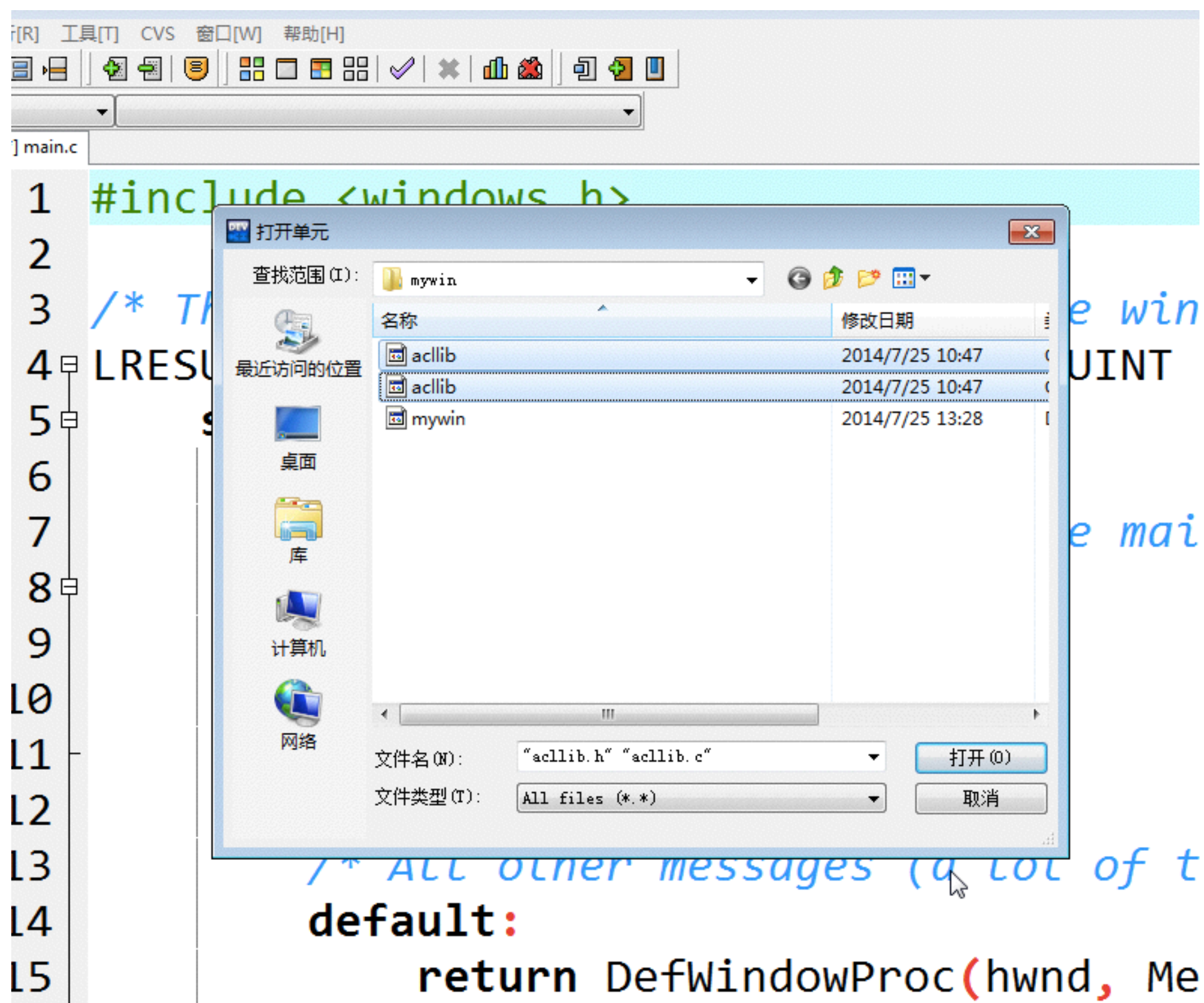
- 把acclib.h和acclib.c两个文件拷贝到项目文件夹中



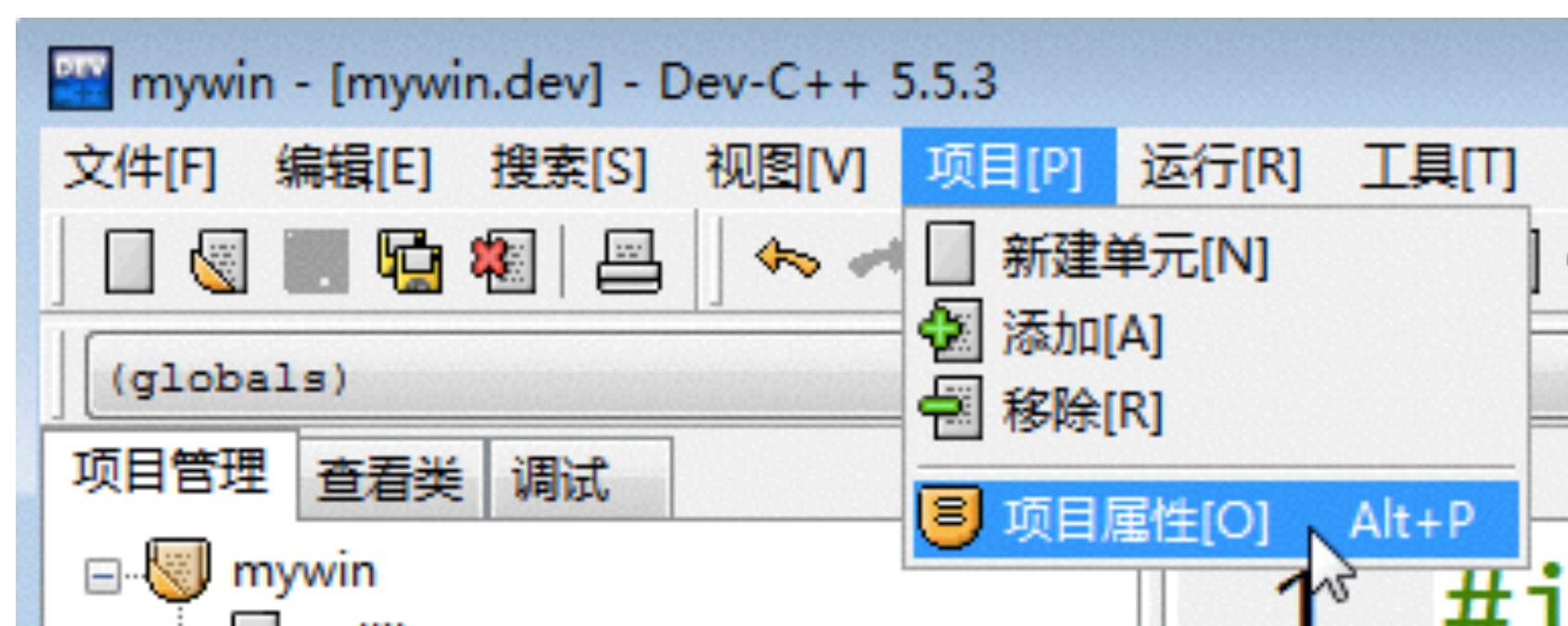
- 在“项目”菜单中选择“添加”



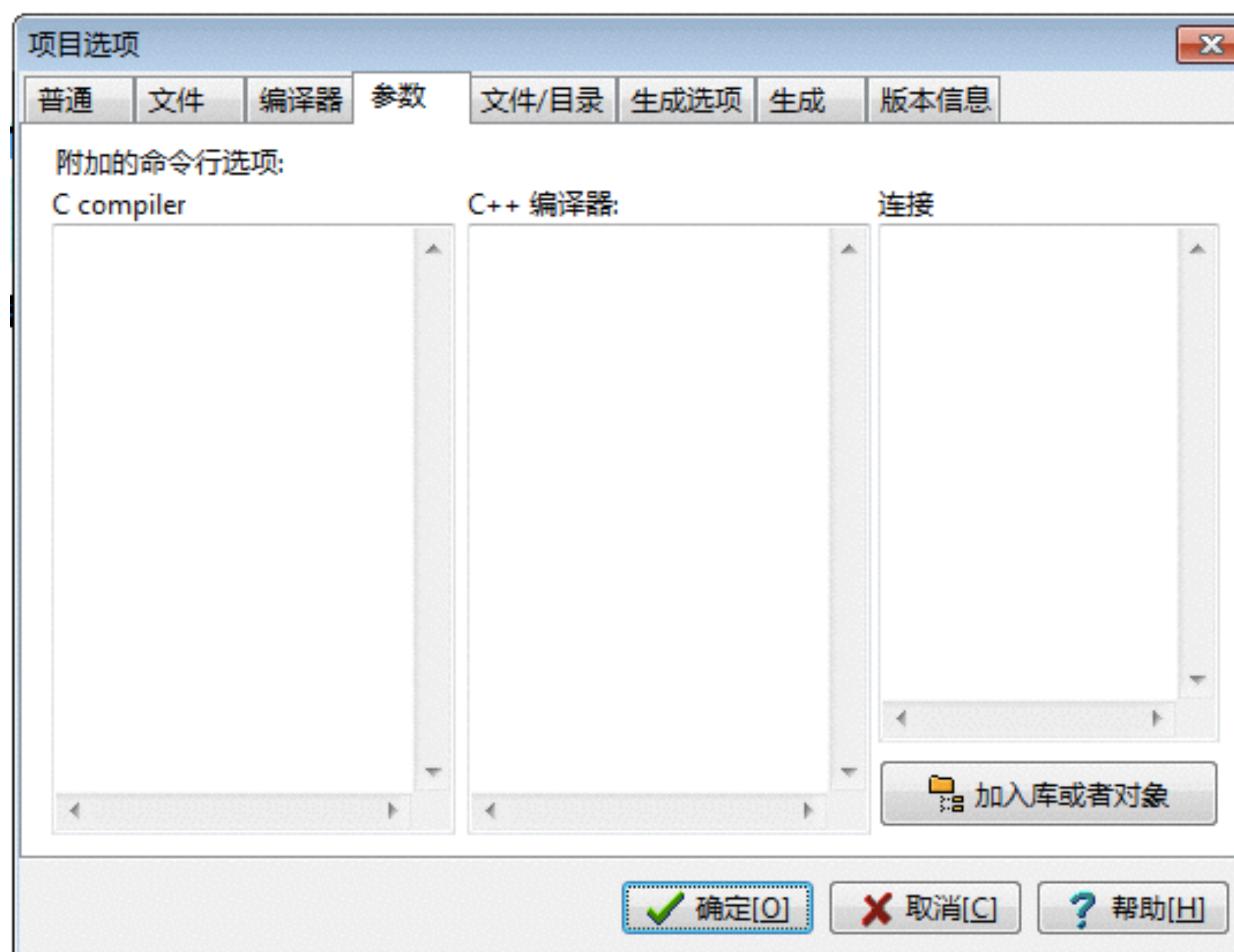
- 把这两个文件加入项目



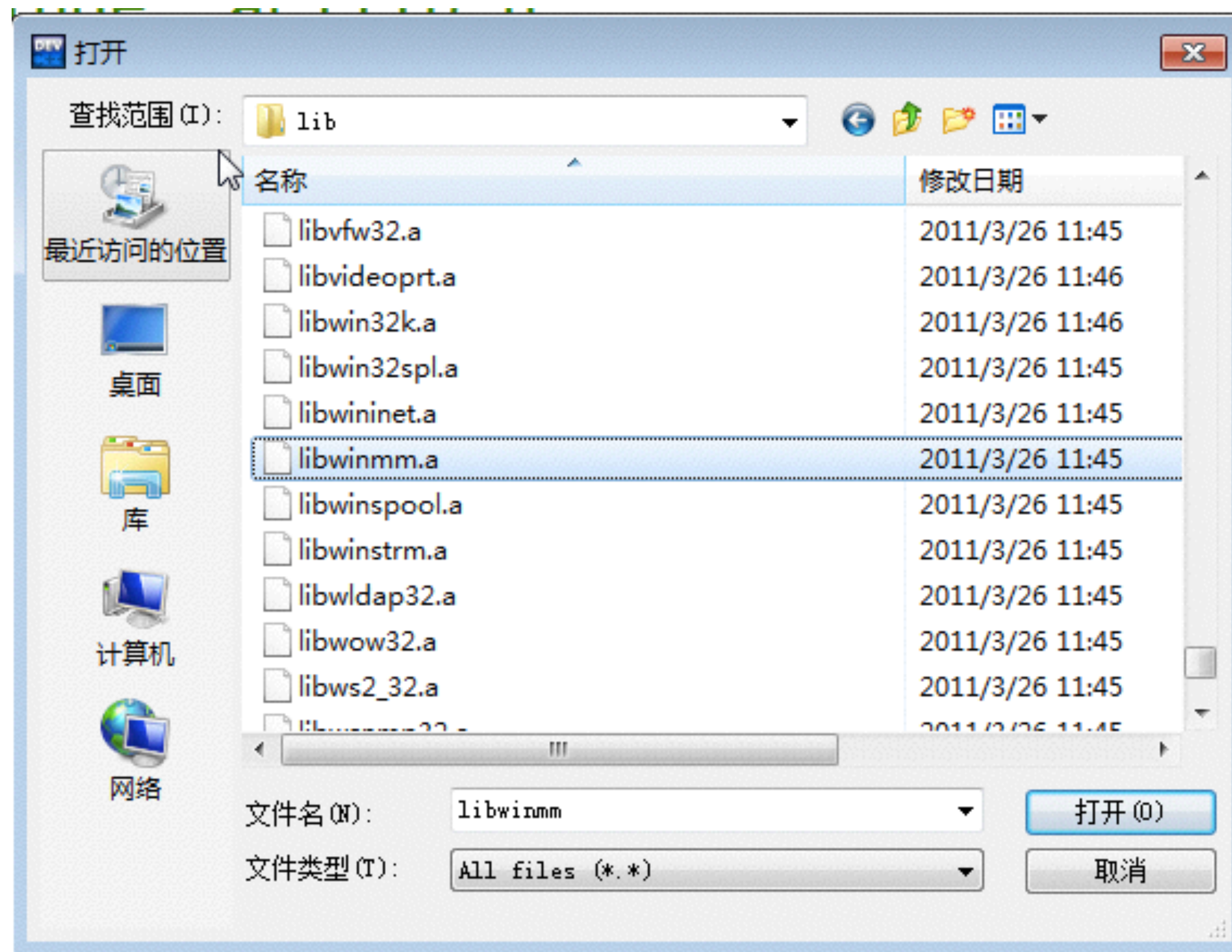
- 在“项目”菜单选择“项目属性”



- 打开“参数”页



- 点击“加入库或对象”按钮，找到Dev安装目录下的MinGW/lib目录，选择libwinmm.a



"C:/Program Files/Dev-Cpp/MinGW32/lib/libwinmm.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libmsimg32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libkernel32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libuser32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libgdi32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libole32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/liboleaut32.a"
"C:/Program Files/Dev-Cpp/MinGW32/lib/libuuid.a"

- 把main.c里的全部内容都删除，按下图输入：

```
#include "acllib.h"

int Setup()
{
    initWindow("Test",DEFAULT,DEFAULT,800,600);

    initConsole();
    printf("Hello\n");

    beginPaint();

    line(10,10,100,100);

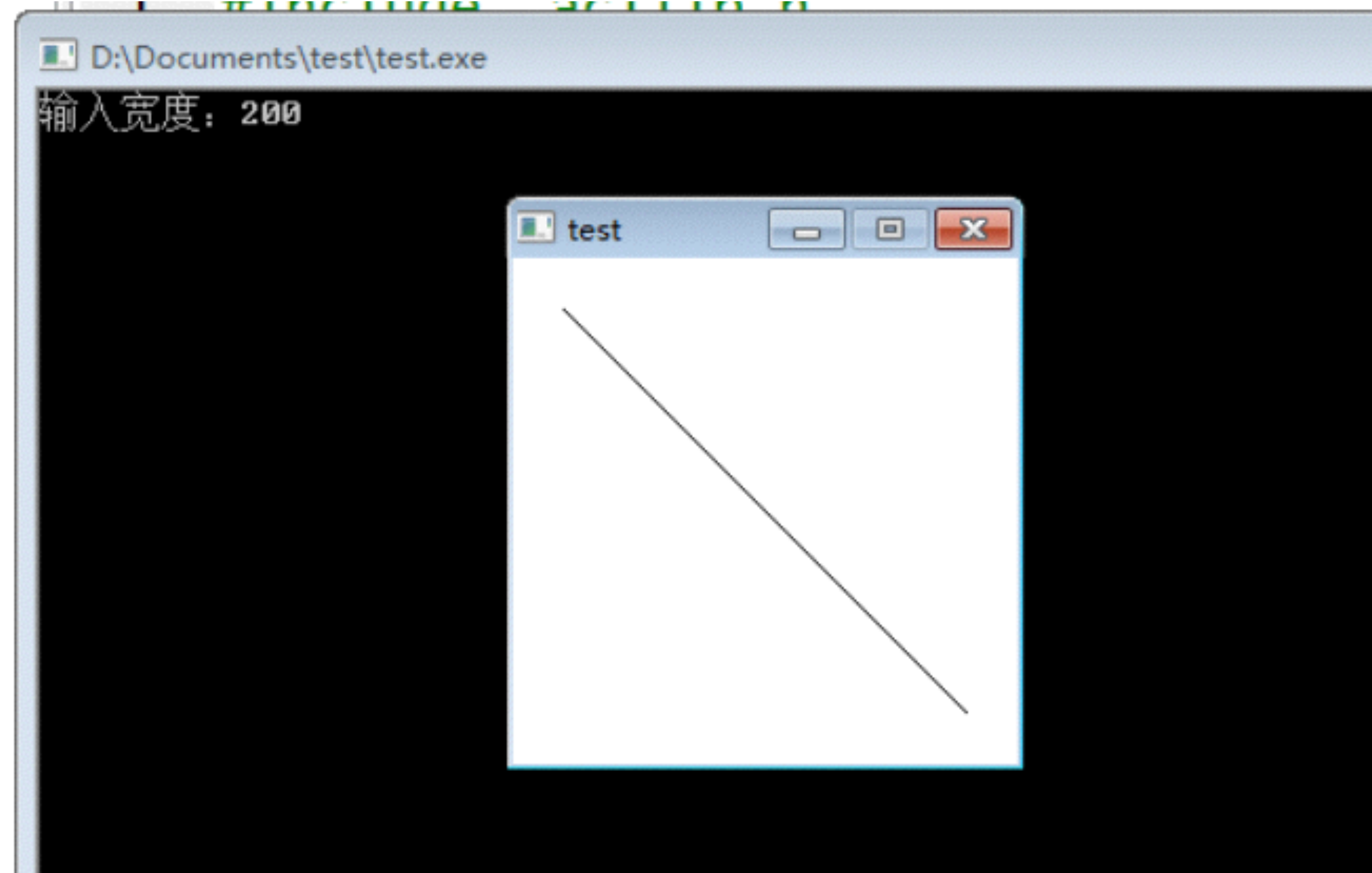
    endPaint();

    return 0;
}
```

ACLLib的基本绘图

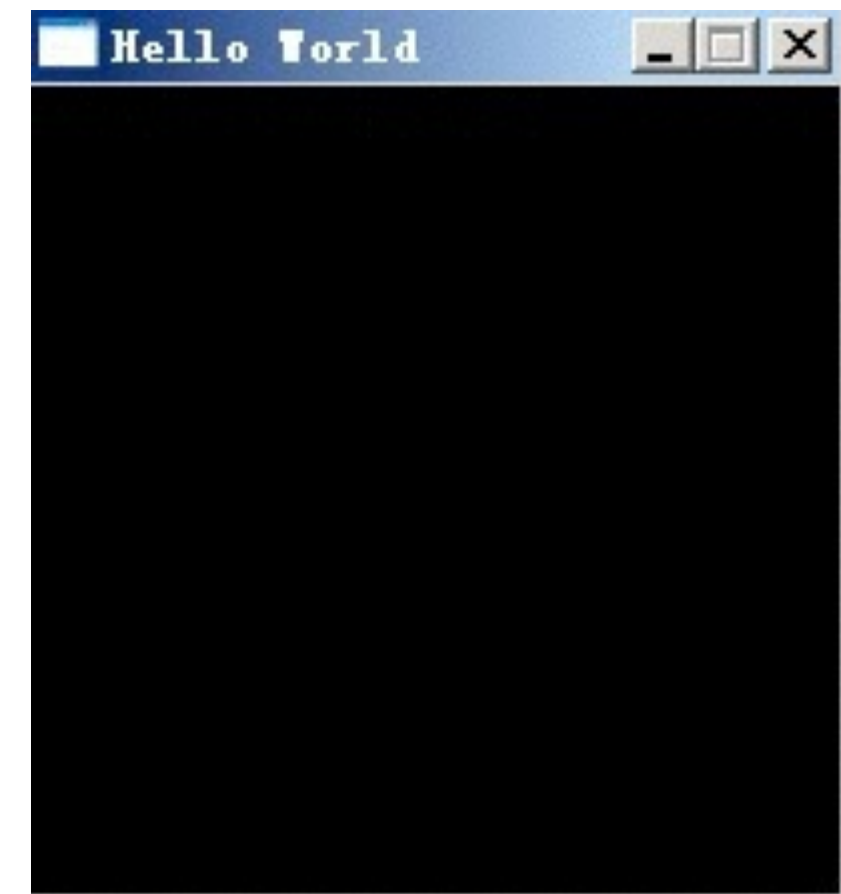
你的第一个程序

```
void initWindow(const char title[],int  
left, int top, int width, int height);  
void beginPaint();  
void endPaint();
```



创建图形窗口

- `void initWindow(const char *name, int left, int top, int width, int height);`
- `int Setup()`
`{`
`initWindow("Hello World",`
`100, 100, 200, 200);`
`return 0;`
`}`



坐标系

- 在Windows中，坐标是以像素点的数字来定义的。对于你创建出来的窗口，左上角是(0,0)，x轴自左向右增长，而y轴自上向下增长

$(0,0)$

$(639,479)$

终端窗口

- 如果需要用scanf和printf，则需要首先
 - `initConsole();`
- 然后就可以在那个窗口上使用scanf和printf了

启动/结束绘图

- `void beginPaint();`
- `void endPaint();`
- 任何绘图函数的调用必须在这一对函数调用之间



- `void putPixel(int x, int y, ACL_Color color);`
- `ACL_Color getPixel(int x, int y);`

颜色

- RGB(r,g,b)
- 红色 \rightarrow RGB(255,0,0)
- BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, WHITE

线

- `void moveTo(int x, int y);`
- `void moveRel(int dx, int dy);`
- `void line(int x0, int y0, int x1, int y1);`
- `void lineTo(int x, int y);`
- `void lineRel(int dx, int dy);`
- `void arc(int nLeftRect, int nTopRect, int nRightRect,
int nBottomRect, int nXStartArc, int nYStartArc,
int nXEndArc, int nYEndArc);`

画笔

- void setPenColor(ACL_Color color);
- void setPenWidth(int width);
- void setPenStyle(ACL_Pen_Style style);
 - PEN_STYLE_SOLID,
 - PEN_STYLE_DASH, /* ----- */
 - PEN_STYLE_DOT, /* */
 - PEN_STYLE_DASHDOT, /* _._._._ */
 - PEN_STYLE_DASHDOTDOT, /* _.._.._ */
 - PEN_STYLE_NULL

面

- `void chrod(int nLeftRect, int nTopRect, int nRightRect, int nBottomRect, int nXRadial1, int nYRadial1, int nXRadial2, int nYRadial2);`
- `void ellipse(int nLeftRect, int nTopRect, int nRightRect, int nBottomRect);`
- `void pie(int nLeftRect, int nTopRect, int nRightRect, int nBottomRect, int nXRadial1, int nYRadial1, int nXRadial2, int nYRadial2);`
- `void rectangle(int nLeftRect, int nTopRect, int nRightRect, int nBottomRect);`
- `void roundrect(int nLeftRect, int nTopRect, int nRightRect, int nBottomRect, int nWidth, int nHeight);`

刷子

- 画笔负责线及面的边缘，刷子负责面的内部
- `void setBrushColor(ACL_Color color);`
- `void setBrushStyle(ACL_Brush_Style style);`
 - `BRUSH_STYLE_SOLID = -1,`
 - `BRUSH_STYLE_HORIZONTAL, /* ----- */`
 - `BRUSH_STYLE_VERTICAL, /* ||||| */`
 - `BRUSH_STYLE_FDIAGONAL, /* \\\ \ \ */`
 - `BRUSH_STYLE_BDIAGONAL, /* /// \ \ */`
 - `BRUSH_STYLE_CROSS, /* + + + + + */`
 - `BRUSH_STYLE_DIAGCROSS, /* x x x x x */`

文字

- `void setTextColor(ACL_Color color);`
- `void setTextBkColor(ACL_Color color);`
- `void setTextSize(int size);`
- `void setFont(char *pFontName);`
- `void paintText(int x, int y, const char *pStr);`