**实验10 lua环境和API**

**实验目的**：

（1）了解C API的概念；

（2）理解lua库函数的应用；

（3）掌握lua环境的创建；

（4）掌握lua的stack栈的使用。

**实验内容**：

# 1. 下载Lua[源码](https://so.csdn.net/so/search?q=%E6%BA%90%E7%A0%81&spm=1001.2101.3001.7020" \t "_blank)。

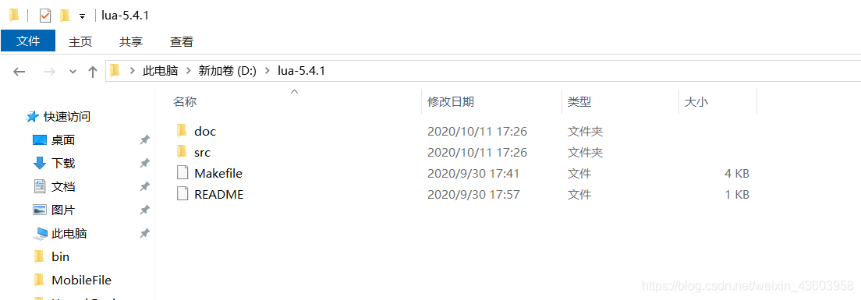
（1）首先进入lua官网<https://www.lua.org/>，下载最新版本的lua源码。



（2）点击下载图标即可下载。

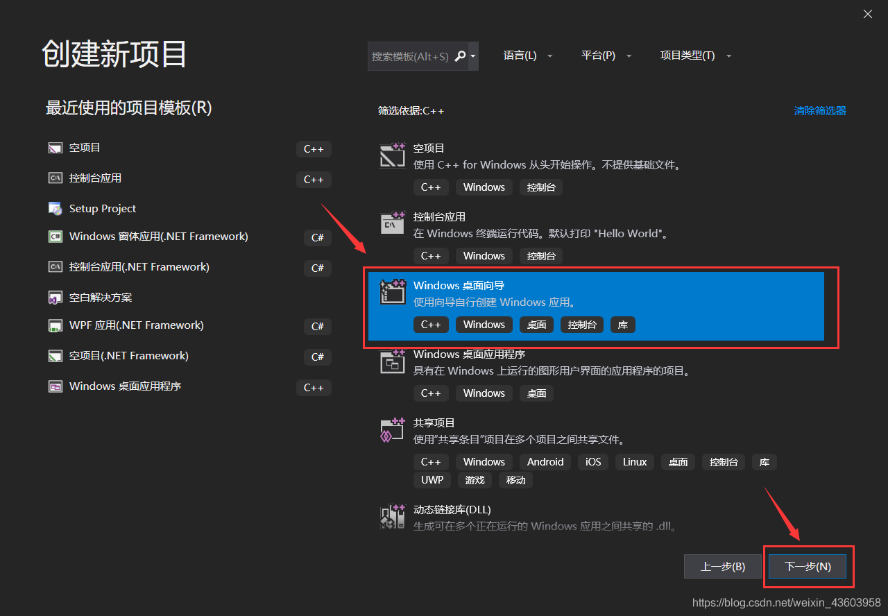


（3）下载压缩包后进行解压，我这里是解压到D盘。

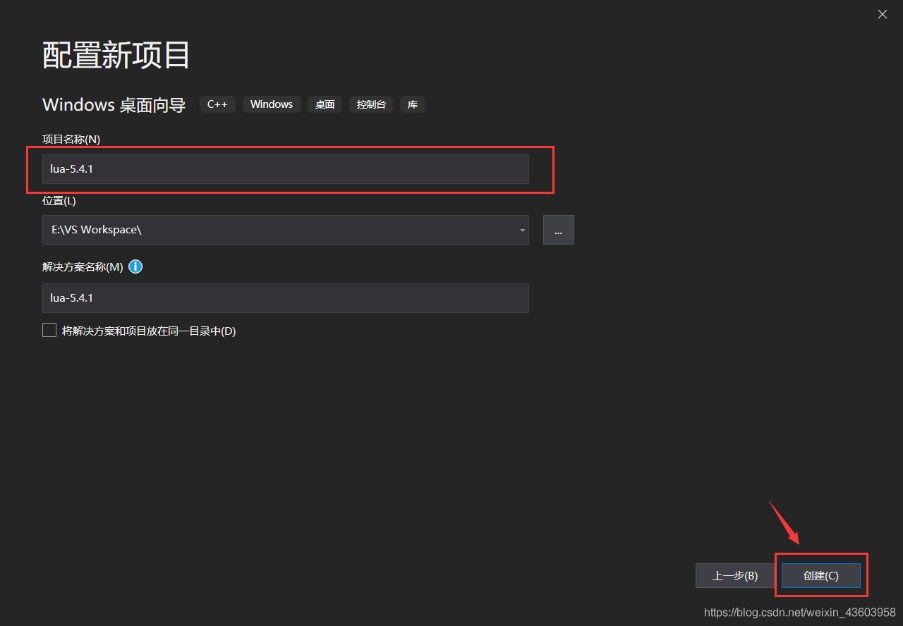


# 2. 用Visual Studio 2019编译Lua源码，生成静态库。

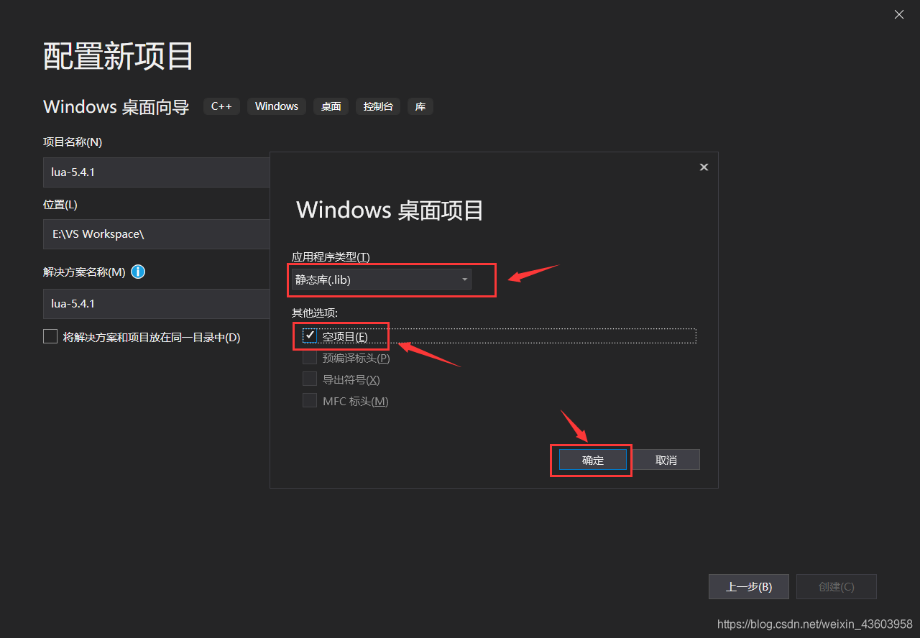
（1）启动VS2019，创建新项目，选择“Windows桌面向导”，点击“下一步”。



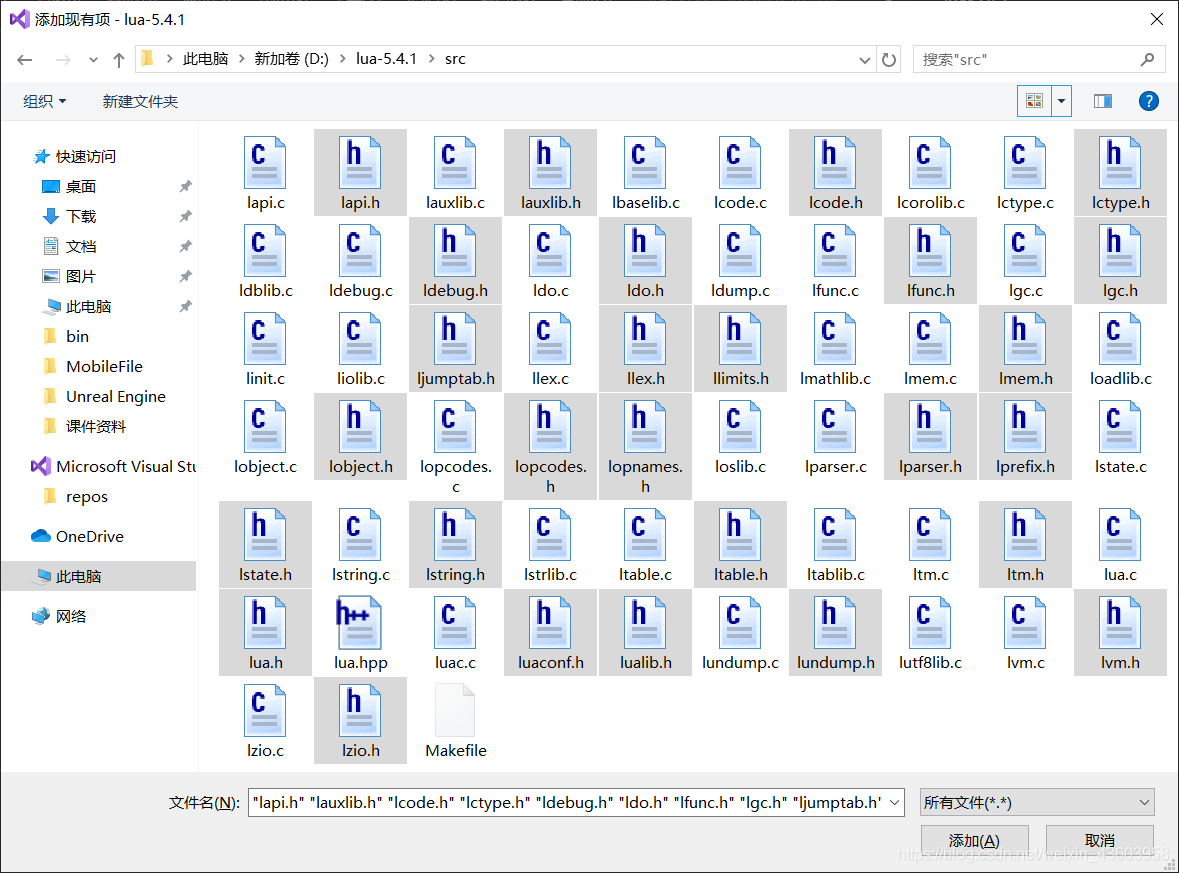
（2）设置项目名称，这里设置的名称是“lua-5.4.1”，点击“创建”。



（3）此时会弹出一个对话框，下拉应用程序类型选择“静态库”，选择“空项目”，点击“确认”，等待项目创建完毕。

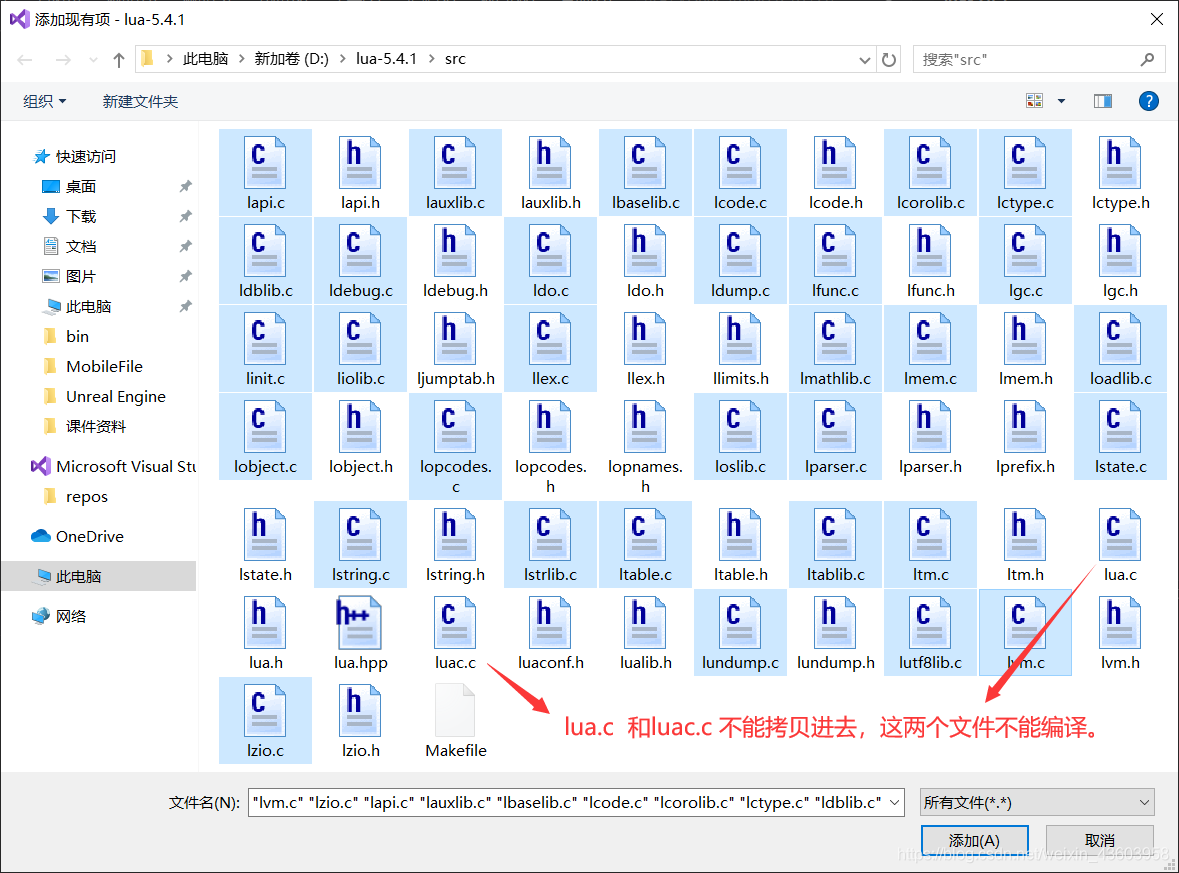


（4）创建好项目后，在【头文件】选项上点击鼠标右键，选择【添加】—>【现有项】，然后找到刚刚解压Lua压缩包的目录，选择Lua目录下的src文件夹，这里我的是D:\lua-5.4.1\src目录。然后把src下的所有.h文件添加到头文件。

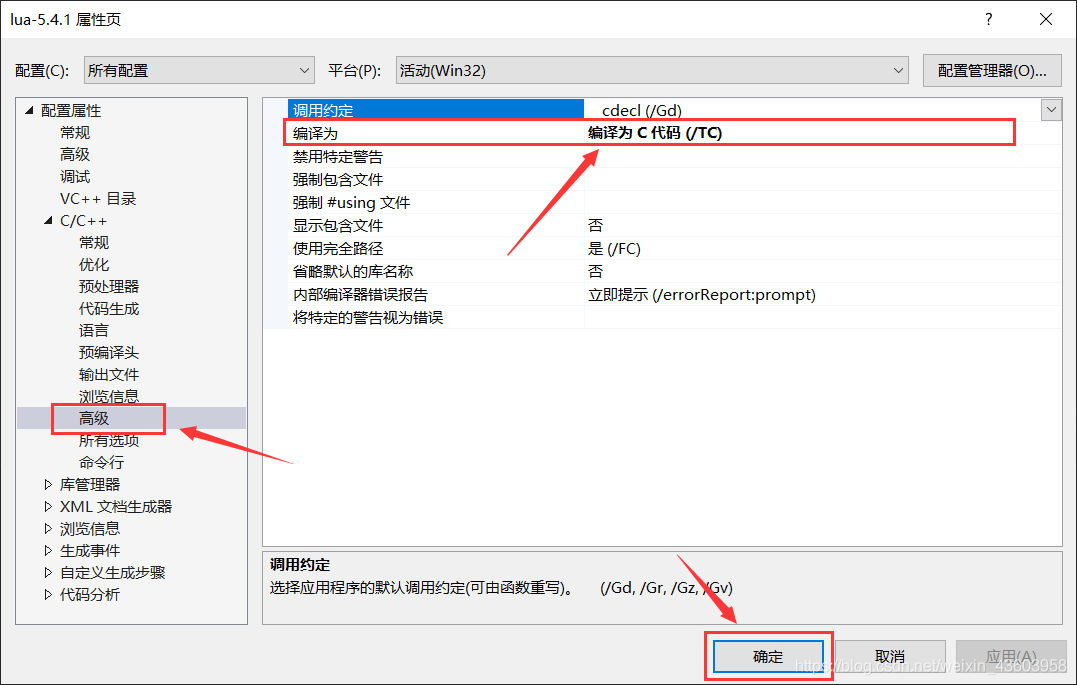


（5）同上操作，在【源文件】选项上点击鼠标右键，以此选择【添加】—>【现有项】，找到Lua目录。把src下除去“lua.c”和“luac.c”后的所有.c文件都添加到源文件。

注意： 需要注意的是 lua.c 和luac.c 不能拷贝进去，这两个文件不能编译。



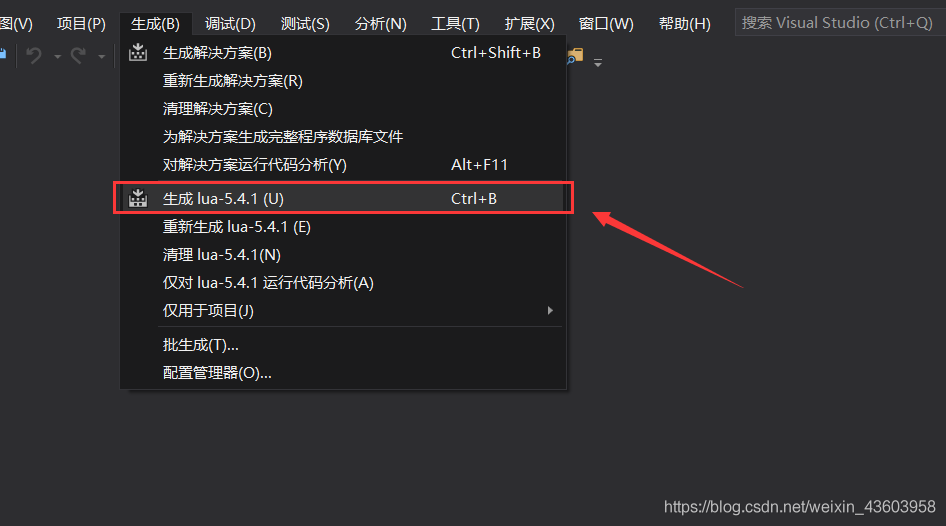
（6）导入头文件和源文件之后，右键【项目】—>【属性】，在界面中操作，【C/C++】—> 【高级】—> 【编译为】选择【编译为C代码(/TC)】，最后确定。



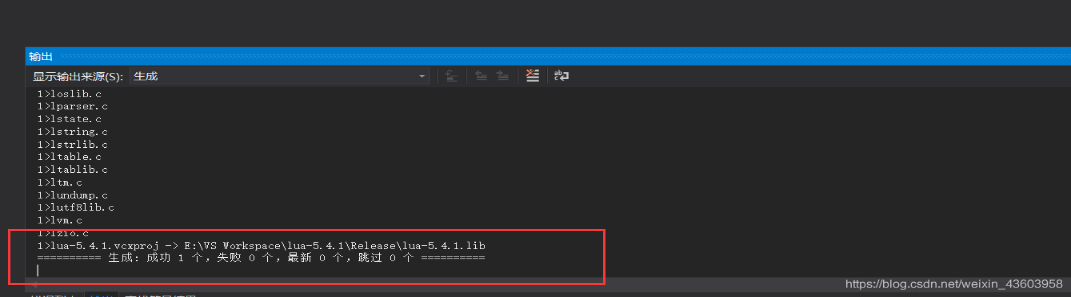
（7）把Debug替换成Release。



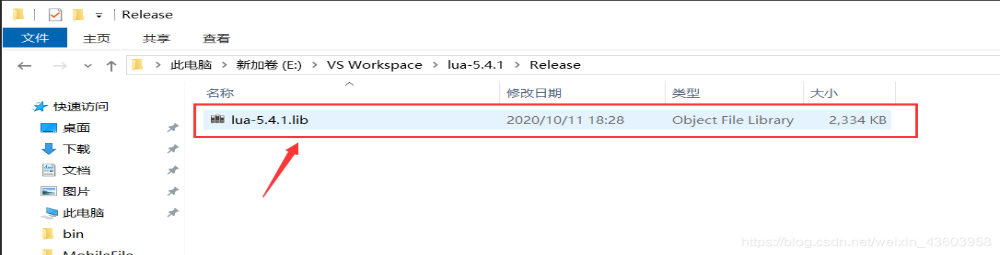
（8）生成静态库。



（9）显示生成成功。



（10）打开项目目录–找到release文件夹–找到lua5.3.lib（你自定义项目名称）。

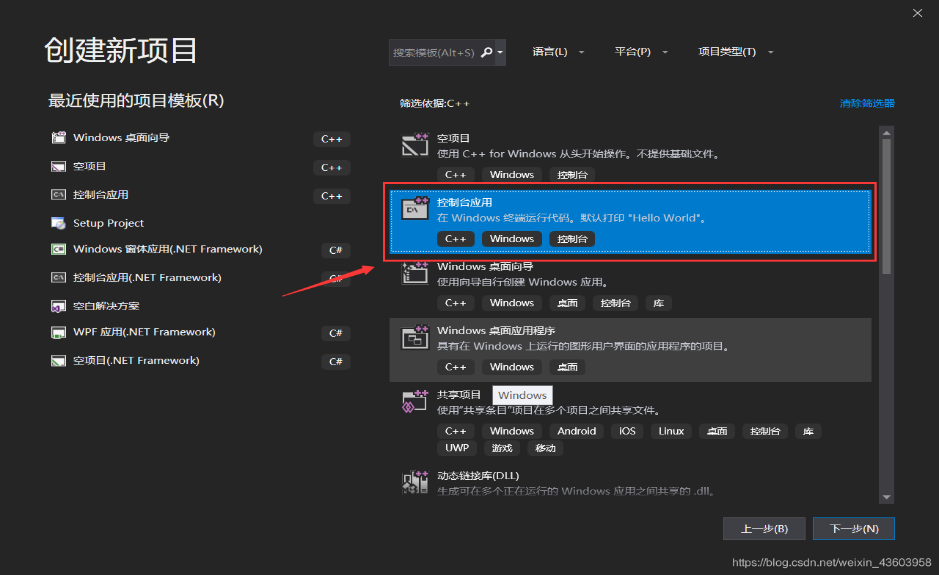


（11）到此为止，lua5.4.1就编译好了，为方便以后引用，可以把它拷贝到Lua解压目录下，如下图所示。

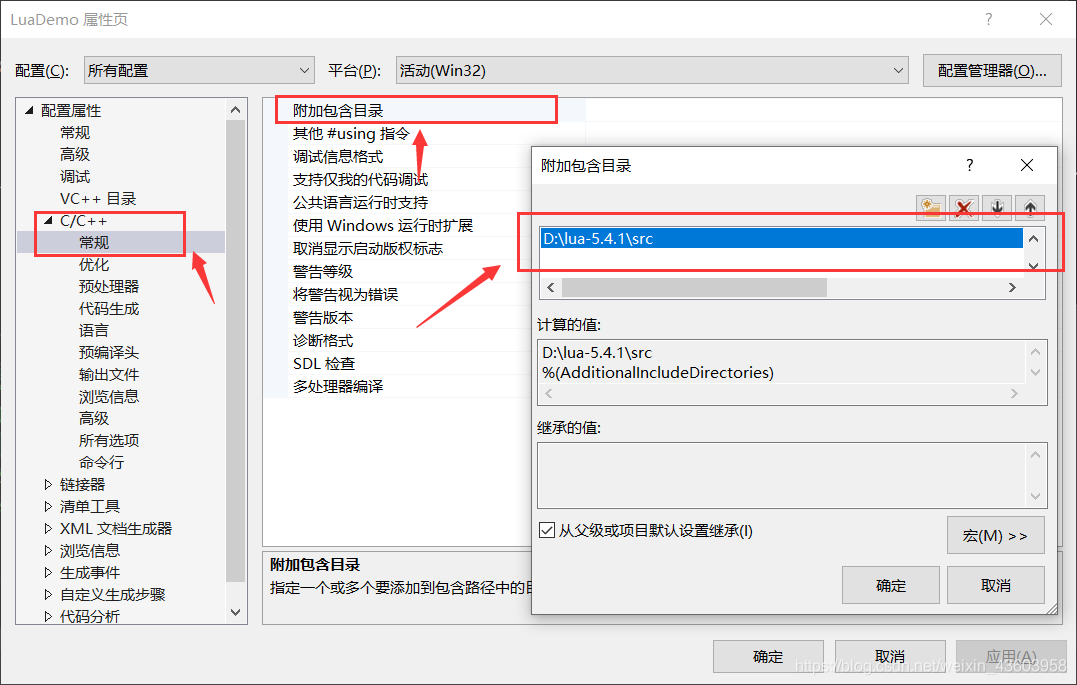


# 3. 在VS2019中，使用Lua库。

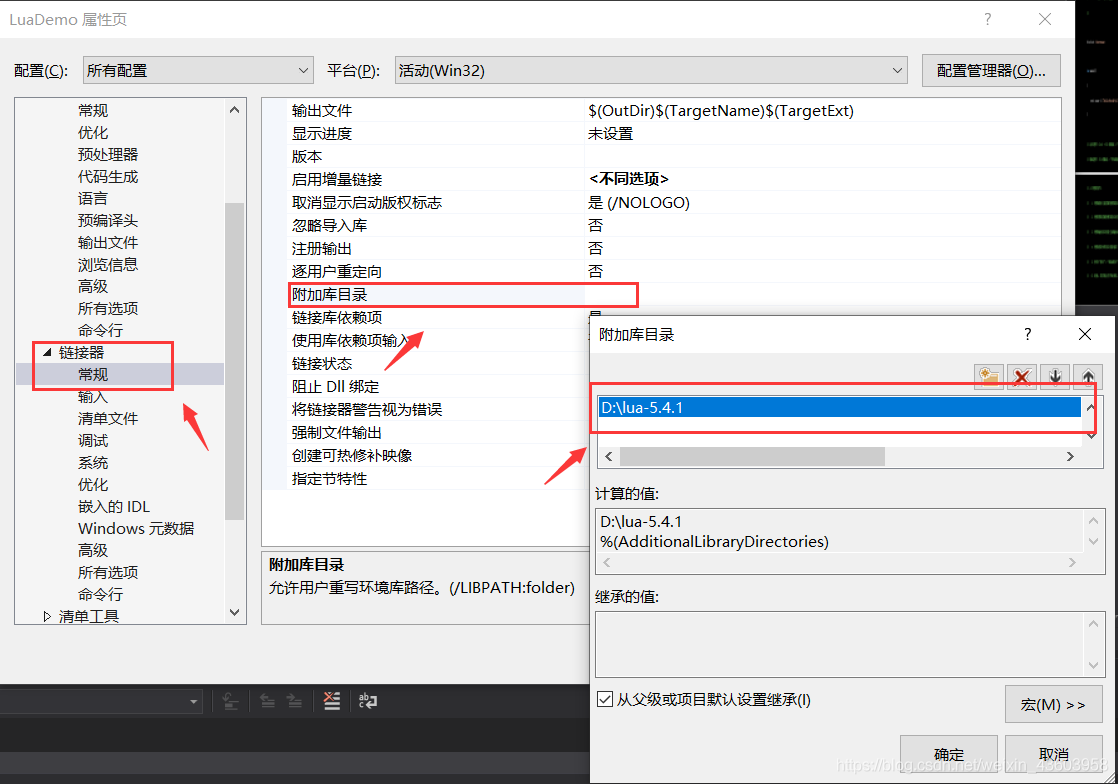
（1）创建一个控制台工程



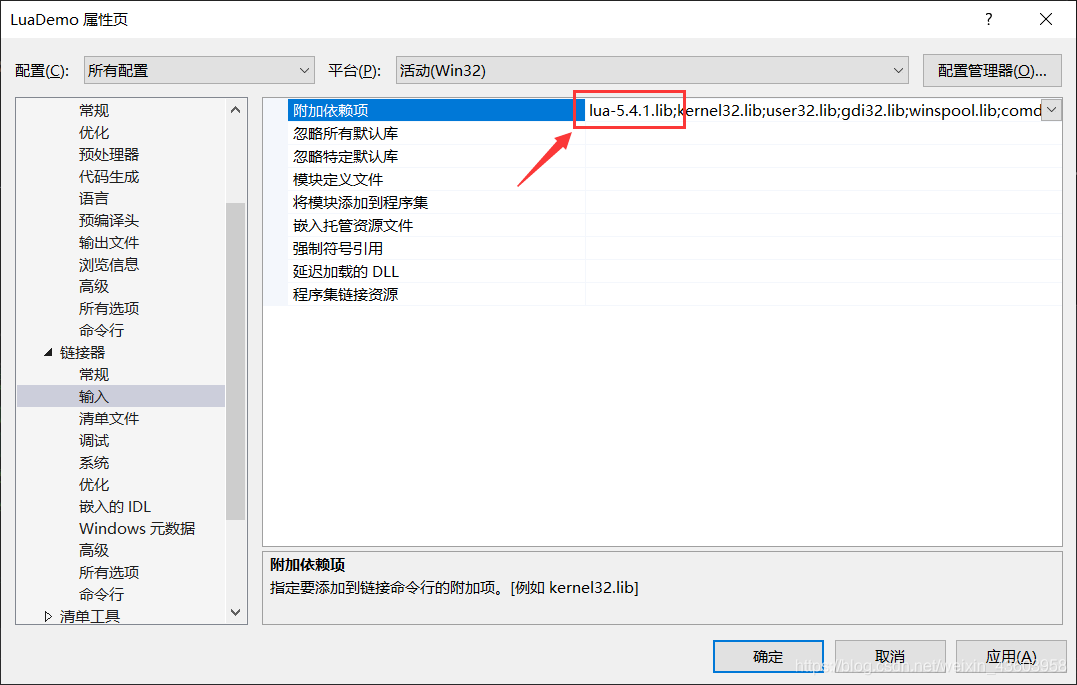
（2）右键单击项目–选择属性。

选择C/C++-常规–附加包含目录–选择lua源码目录的src文件夹。  


（3）选择链接器–常规–附加库目录–选择Lua库文件目录，这个目录为下载Lua源码后解压的那个目录。也就是Lua5.4.1的源码目录，D:\lua-5.4.1。如下图所示。



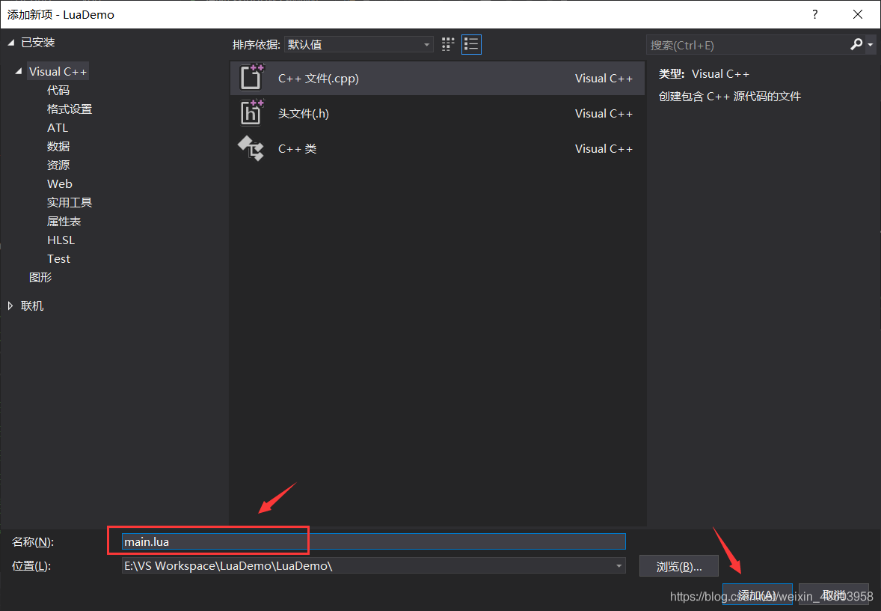
（4）选择链接器–输入–附加依赖项–输入我们所编译的lua5.4.1.lib库文件名称(刚才生成后拷贝到Lua源码目录下的静态链接库)。然后点击应用，确定就好了。



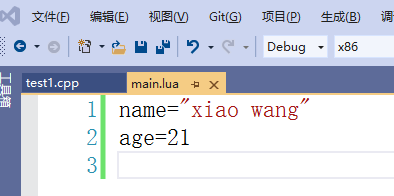
到此Lua开发环境就配置好了，下面测试搭建的Lua环境。

# 4. 测试搭建的Lua环境。

（1）在源代码里创建一个main.lua的文件，如图所示。



（2）打开main.lua，写以下语句



（3）在test1.cpp主函数（创建项目后自带的主函数，项目名称.cpp）里面，编写测试代码并运行:

#include <iostream>

#include <lua.hpp>

using namespace std;

void error(lua\_State\* L, const char\* fmt, ...)

{

va\_list argp;

va\_start(argp, fmt); //将字符串读入到参数列表

vfprintf(stderr, fmt, argp);

va\_end(argp); //参数列表清0

lua\_close(L); //关闭lua状态

system("pause");

exit(EXIT\_FAILURE);

}

int main()

{

lua\_State\* pluaState = luaL\_newstate(); //创建Lua虚拟栈

luaL\_openlibs(pluaState); //打开Lua标准库

//加载并运行文件

if (luaL\_loadfile(pluaState, "main.lua") || lua\_pcall(pluaState, 0, 0, 0))

{

error(pluaState, "omg read invalid ! "); //自定义的错误处理

}

//读取lua中的全局变量压入栈

lua\_getglobal(pluaState, "name");

lua\_getglobal(pluaState, "age");

//读取栈中的元素

cout << "姓名：" << lua\_tostring(pluaState, 1) << endl;

cout << "年龄：" << lua\_tostring(pluaState, -1) << endl;

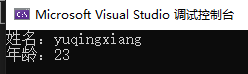
lua\_close(pluaState); //关闭lua虚拟栈

return 0;

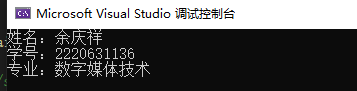
}

（4）运行项目看结果。





（5）将自己的姓名，学号，专业写入一个lua文件中。然后用C++写一个控制台应用程序，在其中调用该lua文件，并将姓名，学号、专业分别输出出来。



Cpp文件主要代码：

#include <lua.hpp>

using namespace std;

void error(lua\_State\* L, const char\* fmt, ...)

{

va\_list argp;

va\_start(argp, fmt);

vfprintf(stderr, fmt, argp);

va\_end(argp);

lua\_close(L);

system("pause");

exit(EXIT\_FAILURE);

}

lua文件代码：

lua\_State\* pluaState = luaL\_newstate();

luaL\_openlibs(pluaState);

if (luaL\_loadfile(pluaState, "main.lua") || lua\_pcall(pluaState, 0, 0, 0))

{

error(pluaState, "omg read invalid!");

}

lua\_getglobal(pluaState, "name");

lua\_getglobal(pluaState, "number");

lua\_getglobal(pluaState, "major");

cout << "姓名：" << lua\_tostring(pluaState, 1) << endl;

cout << "学号：" << lua\_tostring(pluaState, 2) << endl;

cout << "专业：" << lua\_tostring(pluaState, 3) << endl;

lua\_close(pluaState);

（6）请新建一个**Windows桌面应用程序**，根据lua文件中设置的长度和宽度设置窗口的大小。（窗口大小可以在createwindow函数的参数中调整）

HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, 0, **pw, ph**, nullptr, nullptr, hInstance, nullptr);

cpp文件关键代码：

#include "lua.hpp"

void error(lua\_State\* L, const char\* fmt, ...)

{

va\_list argp;

va\_start(argp, fmt);

vfprintf(stderr, fmt, argp);

va\_end(argp);

lua\_close(L);

system("pause");

exit(EXIT\_FAILURE);

}

lua\_State\* pluaState = luaL\_newstate();

luaL\_openlibs(pluaState);

if (luaL\_loadfile(pluaState, "main.lua") || lua\_pcall(pluaState, 0, 0, 0))

{

error(pluaState, "read invalid");

}

lua\_getglobal(pluaState, "width");

lua\_getglobal(pluaState, "height");

if (!lua\_isnumber(pluaState, 1))

error(pluaState, "'width' should be a number\n");

if (!lua\_isnumber(pluaState, 2))

error(pluaState, "'heigt' should be a number\n");

window\_width = lua\_tointeger(pluaState, 1);

window\_height = lua\_tointeger(pluaState, 2);

lua\_close(pluaState);

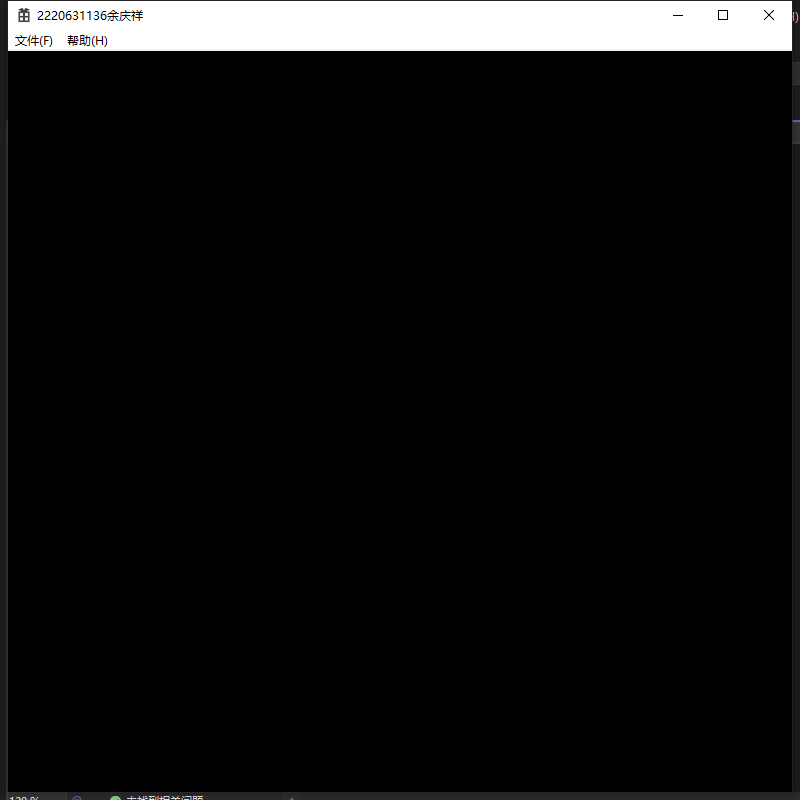
HWND hWnd = CreateWindowW(szWindowClass, L"2220631136余庆祥", WS\_OVERLAPPEDWINDOW,

0, 0, window\_width, window\_height, nullptr, nullptr, hInstance, nullptr);

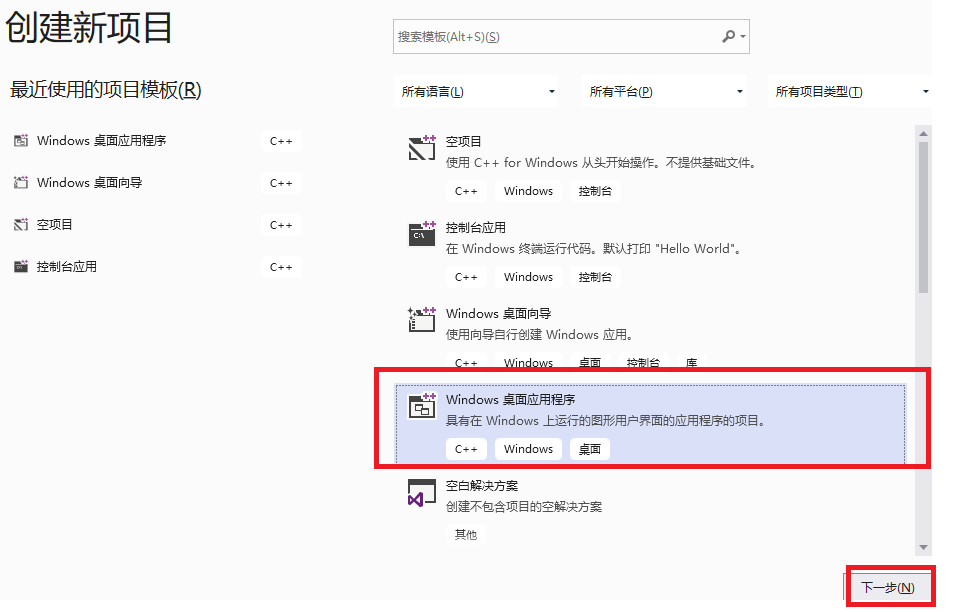
Lua文件代码：

width = 800

height = 800



提示:**首先创建一个lua文档**，存放窗口的长和宽的值，并把该文本文件**放在项目中**；本题目创建的是桌面应用程序，新建项目时是要选择“Windows桌面应用程序”，如下：



源文件如test2.cpp

**实验要求**：

（1）独立完成；

（2）win7以上兼容系统，lua解释器，VS2010以上版本；

（3）记录并分析实验结果。

**实验报告要求**：

（1）按照学院的实验报告格式编写实验报告(注意有封面的)，不能直接使用本指导书修改；

（2）指导书截图不用复制到实验报告，实验报告要附上自己的关键代码。最主要是要有合理的实验分析和总结；

（3）以“学号+姓名+实验\*”命名word文档，提交到ftp对应位置。