# Java通过jna调用C#

### **一 需求：**

Java使用jna调用经过C++ dll编译后的 C# dl库；

### 二 环境：

Windows 7 32位、VS2005、Eclipse，JDK 1.8 32位、微软常用运行库位数也要一直；

需要注意：window 位数需要和JDK、微软常用运行库位数保持一致，不能出现任何位数不一致！！！

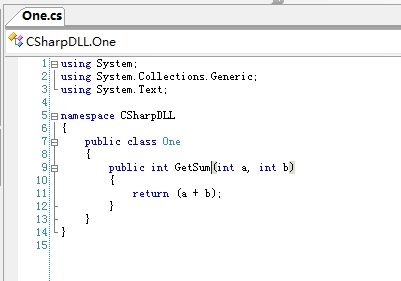
### **三 原理：**

Java调用C# 程序集，其中由于C# 属于托管程序，而Java 调用c# ，最好的方法就是通过使用jna 的jar包实现；通过jna调用由C++ 封装好的C# 动态程序集；

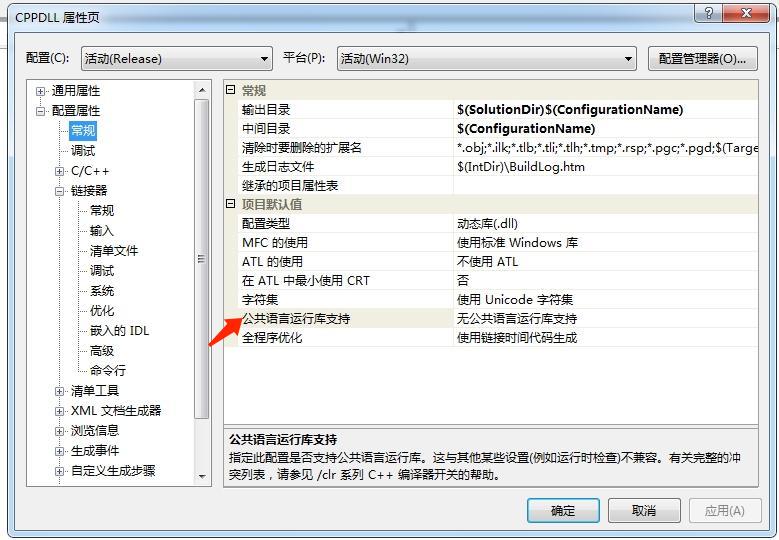
### **四 实现过程：**

1：创建C# 动态链接库DLL项目；

2：C# 实现类（创建One.cs类并实现简单的GetSum方法）：



3：C# 项目属性配置（生成时配置，红色必须选择“运行时需要clr”）：

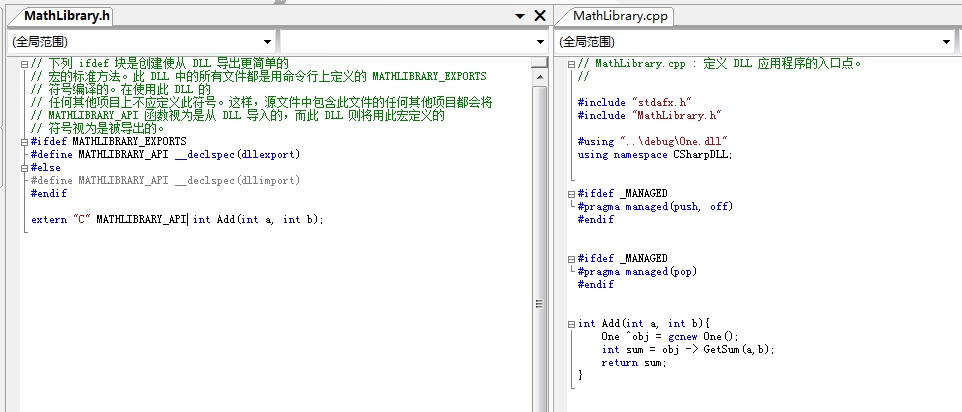


4：右键点击“生成”按钮即生成程序集，在配置管理里面我们也可以生成适用于32位和64位项目的dll文件，只需要设置为AnyCPU即可；

5：C++ 项目目录（win32项目，然后选择 DLL，点击完成即成功创建一个C# dl动态程序集项目，删除默认代码按下面即可实在不会可去MSDN Microsoft官网就能找到）：

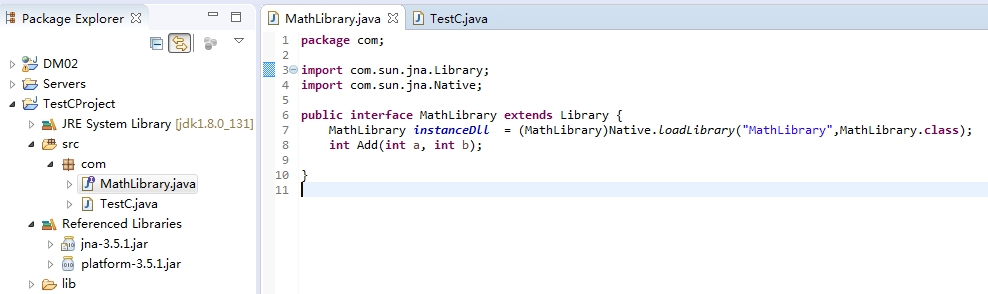
### **10**

6：C++ 添加代码（添加MathLibrary.h头文件、MathLibrary.cpp文件）：

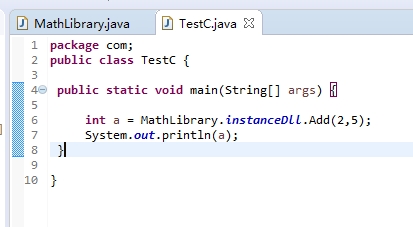


7：右键项目点击“生成” 按钮，注意配置管理里面的生成程序集的目标机器，这个不和C# 生成一个dll 32,64位都能用，C++ 需要针对不同位数生成不同的程序集，具体的可以在配置管理里面进行详细的配置；

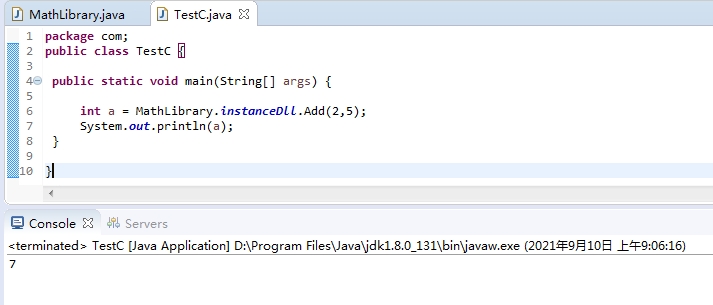
8：Java 目录结构及接口实现（可根据官网demo写）：



9：Java 代码调用（c# dll文件可放在java/bin中，C++可放在java项目根目录或者java/bin下）：



10：运行结果：



### **五 问题汇总与技巧（以本人测试代码为例）：**

**1：通过打开jna的调试信息定位错误原因：**

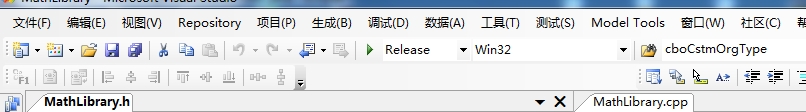
System.setProperty(“jna.debug\_load”,true);

查看DLL库的搜索路径：

System.getProperty(“jna.library.path”);

**2：报错：unable to load library ‘xxx’：找不到指定的模块**

解决：原因是C++ 程序集出现问题，例如：电脑位数和c++ dll不对应，c++程序集位置不对可以按照1中搜索位置放，修改Debug模式为Relase版本如下图：



**3：报错：%1 不是有效的 win32 程序**

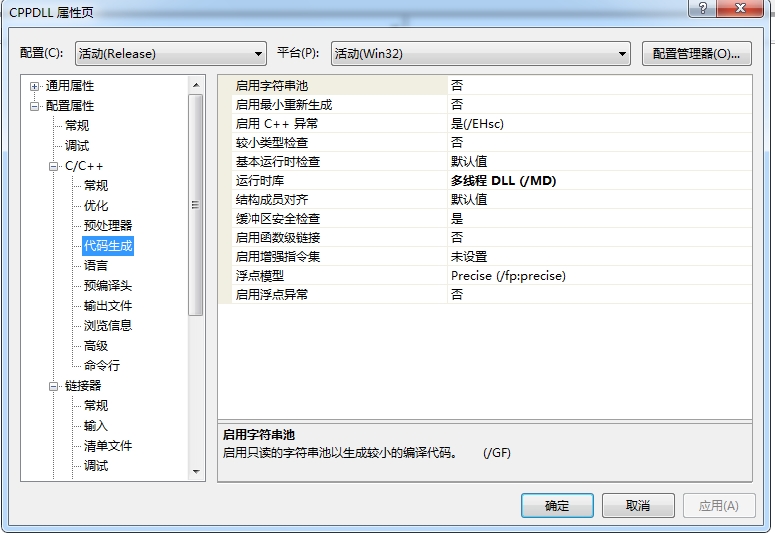
解决：系统和jdk的位数与dll位数不一致；

**4：报错：C++ 生成的 dll文件转移到别人电脑上提示2的错误**

解决：按照2的方法解决即可；

**5：报错：‘CL.exe’已退出，代码2**

解决：cl.exe这个是c++编辑器，这个原因是项目属性->配置属性，常规->公共语言运行库支持选择 这个选项为需要clr的选择和C/C++->代码生成->运行时库MT选错冲突，这里项目属性下的配置属性C/C++ 代码生成中的运行时库必须是多线程DLL(/MD),如下图



### **六 个人总结：**

综合整个项目来说挺感激领导的信任，在另一个项目组完成不了的情况下将这个难点交给自己一人来解决，并能够按时完成并交付。整个测试项目从需求确立到完成经历了将近一周的时间，其中针对自己接触过少的java、C++ 程序来说为了能够确保代码没问题，只能根据官网从创建项目开始一步一步来完成目标，以及针对jna的使用只能根据GitHub源码来开发；针对跨语言调用来讲要求还是非常苛刻的，不论是环境，版本还是数据类型的对应都是需要一一来确认的，当遇到问题从网上或者别人哪里得到答案时，自己需要保持清醒，对其问题能够有自己的分析，虽然不知道怎么做但是只要确认自己想法是对的就一定不要怀疑自己要敢于常识，最终相信你一定会成功。