

中国高等院校计算机基础教育课程体系规划教材

谭浩强 编著

C++程序设计

C++ Programming

医学影像学系电子技术教研室

祝元仲 副教授

Phone: 13350669768

QQ: 31958396, 519788625

2017-2-18



群名称：生物医学工程2014级
群 号：519788625

什么是生物医学工程？



- 生物医学工程（**Biomedical Engineering, BME**）的内涵是将工程科学的原理和方法与生命科学的原理和方法相结合，认识生命运动的规律（定量），并用以维持、促进人的健康。
 - （**NIH (National Institutes of Health, 美国国立卫生研究院)**对生物医学工程学科的定义，中国生物医学工程学会生物医学工程学科发展研究报告**2012**）

什么是生物医学工程？



- 生物医学工程综合运用工程科学、物理学、化学、数学和计算科学的原理研究生物、医学、人体行为与健康。
- 生物医学工程的发展方向是从分子水平到器官水平，发展新的概念，建立新的知识，创立新的技术，开发新的材料和器械，为人类的健康事业服务。

培养目标



- 专业领域为医疗仪器，重点即大型医疗设备
 - 操作使用-影像技师
 - 维护管理-临床工程师、医疗装备管理师
 - 销售维修-销售、售后售前工程师
 - 生产制造-工程师
 - 研发设计-工程师
- 医学物理师
- 医院信息化建设与管理

课程性质、目的和任务



- 本课程是生物医学工程专业的一门重要的专业基础课，具有较强的理论性和实践性。
- 目的：理解C++的基本概念，建立面向对象的程序设计思想，掌握基本的面向对象程序设计方法，能够利用C++语言解决一般应用问题，为后继专业课程的学习与研究奠定程序设计基础。
- 本课程主要讲授C++的语法、算法、面向对象程序设计的基本方法。

课程教学要求



- 1. 掌握C++程序设计语言的基本语法与规则。
- 2. 重点掌握程序结构及算法、函数、指针及引用、类与对象、继承与派生、多态性等概念。
- 3. 灵活运用面向过程编程方法解决一些简单的问题。
- 4. 能够分析面向对象程序的程序结构，阅读和理解面向对象的程序。
- 5. 能够模仿并上机实现面向对象程序。



关于学习方法的一般要求

- 课前预习
 - 课程内容多，一定要预习，尤其找出难点疑惑处。
- 课堂注意
 - 有问题及时提；适当记笔记；
- 课后复习
 - 仔细阅读和理解课本的相关内容、细节和例题，掌握最常用的一些基本语法、语义、语用，在应用中深入逐步掌握。
- 课后作业
 - 可以相互讨论、相互借鉴，要“知其然，知其所以然”。
- 上机实践
 - 结合问题，上机编写、调试程序。 ***Do it yourself!***

请按要求完成作业、
阅读资料



关于C++方法的观点

- 实践、理解、再实践，是精通计算机编程的唯一途径；
- 程序调试(**DeBug**)是计算机编程最重要的技术之一；
- 不要求死记硬背规则！重要的是编程思路、**debug**、见识；
多练习！多犯错误！多积累经验！
- 尽可能完成课后每个练习题，并上机调试程序。
- 借鉴前人的成果、多学习多交流，是提高编程水平的捷径。**MSDN**；**CSDN**；
- 培养良好的项目管理意识；
管理自己的作业和作品

上机出问题是正常现象！不要着急！

参考书目



1. **The C++ Programming Language**, (高等教育出版社, 原版影印**Special Edition**)
2. **C++程序设计语言(特别版)**, Bjarne Stroustrup 著
裘宗燕 译 机械工业出版社
3. **C++程序设计教程(第二版)**, 钱能 主编 清华大学出版社
4. **C++语言程序设计(第三版)**, 郑莉 董渊 编著 清华大学出版社
5. **C++ Primer (第三版) 中文版**, Stanley B.Lippman 等著 潘爱民 等译 中国电力出版社
6. **C++编程思想 Thinking in C++**, Bruce Eckel 著
刘宗田 等译 机械工业出版社
7. <http://msdn.microsoft.com/>
8. <http://www.csdn.net/>



教学及实验安排

- 成绩评定

- 平时 **10%**，分组检查
- 实验 **20%**，实验考试
- 考试 **70%**，补考 **≠** 通过

- 实验安排

- 时间：周二下午
- 地点：**11**楼生物医学工程实验室
- 要求：预习报告，实验报告

总 目 录



第1篇 基 本 知 识

第1章 C++的初步知识

第2章 数据类型与表达式

第2篇 面向过程的程序设计

第3章 程序设计初步

第4章 函数

第5章 数组

第6章 指针

第7章 自定义数据类型



第3篇 基于对象的程序设计

第8章 类和对象的特性

第9章 使用类和对象

第10章 运算符重载

第4篇 面向对象的程序设计

第11章 继承与派生

第12章 多态性与虚函数

第13章 输入输出流

第14章 C++工具

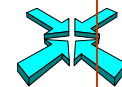
前言



- 不应当把面向过程和面向对象的程序设计对立起来，任何程序设计都需要用到面向过程的知识。
- C++既可以编写面向过程的程序，也可以编写面向对象的程序。
- C++程序设计牵涉到面向对象程序设计的理论、C++语言的语法以及算法等3个方面的内容。

第1章 C++的初步知识

Preliminary Knowledge of C++



什么是计算机程序？程序有什么用？

如何编写一个C++程序？

1.1 从C到C++

1.2 最简单的C++程序

1.3 C++程序的构成和书写形式

1.4 C++程序的编写和实现

1.5 关于C++上机实践

* 1.1 从C到C++



机器语言--汇编语言--高级语言；

FORTRAN (IBM) -1954，科学计算

Ada (Ada Augusta Byron) -1983，通用程序设计

Cobol (Common Business Oriented Language) -1959，最早的高级编程语言之一，数据处理

BASIC-1964，在**FORTRAN**语言的基础上简化而成的，它是为初学者设计的小型高级语言。

C-1972，由美国贝尔实验室**Dennis.M.Ritchie**研制成功。是为计算机专业人员设计的。大多数系统软件和许多应用软件都是用**C**编写的。

http://blog.sina.com.cn/s/blog_4bddce2b0102dwf2.html

Ada Augusta Byron



- 爱达·奥古斯塔·拜伦(**Ada Augusta Byron**)1815年生于英国伦敦。她的父亲是英国著名诗人乔治·拜伦，母亲是数学家安娜·伊莎贝拉 (“**the Princess of Parallelograms**”，平行四边形公主)。
- 曾对现代计算机技术之父查尔斯·巴贝奇(**Charles Babage**)的笔记、手稿进行了整理和修正。被称为世界上第一位计算机程序员。



Grace Murray Hopper



- **Grace Murray Hopper, 格蕾丝·赫柏 (1906—1992), 美国海军少将。是共享代码库、编译器及验证软件以及编译器标准的使用的早期倡导者。促进了计算机科学的发展,促成了COBOL的产生。**
- **被称为当代第一个程序员, 她为Mark I计算机写了第一个程序。**
- **1943 年1 月:Mark I 自动顺序控制计算机在美国研制成功。整个机器有51 英尺长、5 吨重、75万个零部件。该机使用了3304 个继电器, 60 个开关作为机械只读存储器。程序存储在纸带上, 数据可以来自纸带或卡片阅读器。Mark I 被用来为美国海军计算弹道火力表。**
- **A ship in port is safe, but that is not what ships are built for.**
- **(船泊在港湾是安全的, 但那并不是我们造船的目的)**

Dennis.M.Ritchie



- **C语言之父，UNIX之父。40年如一日在贝尔实验室/朗讯科技公司（原AT&T实验室）工作，现在是计算科学研究中心系统软件研究部主任。**
- **美国计算机协会(ACM)授予的系统及语言杰出论文奖(1974)；电气和电子工程师协会(IEEE)的Emmanuel Piore 奖(1982)；美国计算机协会(ACM)的图灵奖(1983)；等等。**
- **1978年Brian W.Kernighian和Dennis M.Ritchie出版了名著《C程序设计语言（The C Programming Language）》，是C语言方面最权威的教材之一。**



C++是由AT&T Bell(贝尔)实验室的Bjarne Stroustrup博士及其同事于20世纪80年代初在C的基础上开发成功的。C++保留了C原有的所有优点，增加了面向对象的机制。

面向对象程序设计，是针对开发较大规模的程序而提出来的，目的是提高软件开发的效率。

面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

C++既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的混合型的程序设计语言。

这是计算机悬案。很多作者谈到C++中的“undefined”错误时都说“就象Stroustrup的名字一样“,呵呵

bi ya ni. si chu si chua p



Bjarne Stroustrup, C++ • 语言的设计者和实现者，世人称之为“C++之父”。

剑桥大学计算机科学博士，得克萨斯州A&M大学计算机系教授。

- 1990年荣获《财富》杂志评选的“美国12位最年轻的科学家”称号。
- 1993年，由于在C++领域的重大贡献，Bjarne获得了ACM该年度的 Grace Murray Hopper大奖并成为ACM院士（Association for Computer Machinery，目前世界上最大的教育和科学计算协会，成为ACM院士是个人成就的里程碑）。
- 1995年，BYTE杂志颁予他“近20年来计算机工业最具影响力的20人”的称号。

*1.2 最简单的C++程序



例1.1 输出一行字符。

```
#include <iostream>           //包含头文件iostream
using namespace std;          //使用命名空间std
int main( )
{
    cout<<"This is a C++ program.";
    return 0;
}
```



例1.2 求a和b两个数之和。

```
// 求两数之和
#include <iostream>
using namespace std;
int main( )
{
    int a,b,sum;
    cin>>a>>b;
    sum=a+b;
    //输出语句
    cout<<"a+b="<<sum<<endl;
    return 0;
}
```

(本行是注释行)
//预处理命令
//命名空间std
//主函数首部
//函数体开始
//定义变量
//输入语句
//赋值语句
//返回一个零值
//函数结束



例1.3 给两个数x和y，求两数中的大者。

```
#include <iostream> //预处理命令
using namespace std; //命名空间std
int main( ) //主函数
{ //主函数体开始
    int max(int x,int y); //对max函数作声明
    int a,b,m; //变量声明
    cin>>a>>b; //输入变量a和b的值
    m=max(a,b); //调用max函数
    cout<<"max="
    <<m<<'\\n'; //输出大数m的值
    return 0; //程序正常返回零值
} //主函数结束
```



```
int max(int x,int y) //定义max函数，形参为整型
{
    //max函数体开始
    int z;           //变量声明，定义变量z为整型
    if(x>y) z=x;      //如果x>y，则将x的值赋给z
    else z=y;         //否则，将y的值赋给z
    return(z);        //将z的值返回调用处
}                    //max函数结束
```

1.3 C++程序的构成和书写形式



- (1) 源程序文件是后缀为.cpp的文本文件。
- (2) 源程序通常包括以下几个部分：
 - ①注释部分。注释部分以//或/*作为标记。
 - ②预处理命令。
 - ③全局声明部分(在函数外的声明部分)。包括对用户自己定义的数据类型的声明和程序中所用到的变量的定义。
 - ④函数，是实现操作的部分，是程序中必须有的和最基本的组成部分。必须有一个(而且只能有一个)主函数(main函数)。



- **(3) 一个函数由两部分组成:**

- ① 函数首部，即函数的第一行。包括函数名、函数类型、函数属性、函数参数(形参)名、参数类型。

int max(int x,int y)

- ② 函数体，即函数首部下面的大括号内的部分。

函数体一般包括:

- 局部声明部分 (在函数内的声明部分)。包括对本函数中所用到的类型、函数的声明和变量的定义。
- 执行部分。由若干个执行语句组成，用来进行有关的操作，以实现函数的功能。

- **(4) 语句包括两类。一类是声明语句，另一类是执行语句。语句是实现操作的基本成分。**



- **(5) C++程序总是从main函数开始执行。**
- **(6) C++程序书写格式自由，每个语句以分号作为结束标志。**
 - 用大括号对{ }表示程序的层次范围，一个完整的程序模块要用一对{ }包括。
- **(7) 源程序必须加上必要的注释，以增加程序的可读性。**
 - 用“//”作注释时，有效范围只有一行。
 - 用“/*.....*/”作注释时有效范围为多行。
- **(8) C++程序中的名字（标识符）是大小写敏感的。**

编程风格之程序风格



- **1. 缩进格式:**

- 各层次缩进的分格采用**VC**的缺省风格，即每层次缩进为**4格/8格**；对继行则要求再缩进**4格/8格**。

- **2. 大括号的位置**

- 将开始的大括号放在一行的最后，而将结束大括号放在一行的第一位，
- 大括号位于下一行，要求相匹配的大括号在同一列，例如：



```
• int main()  
• {  
•     long lI;           // 循环变量  
•     long lSum;         // 用来记录和  
•     float fAvg;        // 用来求平均值  
•     .....  
•     for( lI=0;lI<10;lI++)    // 对数进行累加  
•     {  
•         lSum=lSum+lI;  
•         .....  
•     }  
•     fAvg=lSum/10.0;        // 求平均值  
•     .....  
• }
```



编程风格之注释规范

- 1、文件注释：
- 文件应该在开头加入以下注释：
- ```
////////////////////////////////////
```
- ```
// 项目: 文件所在的项目名。
```
- ```
// 作者: **, 修改者: **
```
- ```
// 描述: 说明文件的功能。
```
- ```
// 主要函数:
```
- ```
// 版本: 说明文件的版本, 完成日期。
```
- ```
// 修改: 说明对文件的修改内容、原因以及日期。
```
- ```
// 参考文献: .....
```
- ```
////////////////////////////////////
```
- 文件注释可以炫耀你的成就, 也是为了捅了篓子别人可以找你!O(∩\_∩)O~

# 编程风格之注释规范



- 2、函数注释
- 对于函数，应该从“功能”，“参数”，“返回值”、“主要思路”、“调用方法”、“日期”六个方面用如下格式注释：
- ```
//=====//
```
- ```
// 功能： 从一个String 中删除另一个String
```
- ```
// 参数： strByDelete,strToDelete
```
- ```
// 返回： 找到并删除返回1， 否则返回0.....列出错误编码
```
- ```
// 主要思路： 本算法主要采用循环比较的方法来.....
```
- ```
// 调用方法：
```
- ```
// 日期： 2000/8/21.9:40--2000/8/23.21:45
```
- ```
//=====//
```
- ```
函数名(.....)           // 功能.....
```

编程风格之注释规范



- **2、变量的注释：**

- 对于变量的注释紧跟在变量的后面说明变量的作用。原则上对于每个变量应该注释，但对于意义非常明显的变量，如：**i,j**等循环变量可以不注释。例如：

- **long lI;** // 循环变量
- **long lSum;** // 总和
- **float fAvg;** // 分数平均值
- **long lLineCount** // 数据线的根数



编程风格之注释规范

- 4、其他注释：
- 在各功能模块的每一主要部分之前添加块注释。
- 在循环、条件、选择等位置尽可能多加以注释。
- 对于前后顺序不能颠倒的情况，在注释中增加序号。

• 例如：

```
• //1、.....注释
• for (.....)
• {
•     if(.....)                //.....注释
•     {.....                }
•     else                    //.....注释
•     {.....                }
• }
```

1.4 C++程序的编写和实现



一个程序从编写到运行的步骤：

1. 用C++编写源程序

C++的源程序 (*source program*) 是以.cpp (c plus plus)作为后缀的文本文件。

2. 对源程序进行编译

编译是以源程序文件为单位编译的。

为了使计算机能执行高级语言源程序，必须先用编译器(*compiler*，也称编译程序或编译系统)，把源程序翻译成二进制形式的目标程序(*object program*)。



目标程序一般以`.obj`或`.o`作为后缀(**object** 的缩写)。编译时对源程序进行词法检查和语法检查。编译结束后会显示出所有的编译出错信息。一般编译系统给出的出错信息分为两种，一种是错误(**error**)；一种是警告(**warning**)。

3. 将目标文件连接

连接程序(**linker**)将一个程序的所有目标文件和系统的库文件以及系统提供的其他信息连接起来，最终形成可执行的二进制文件（***.exe**）。

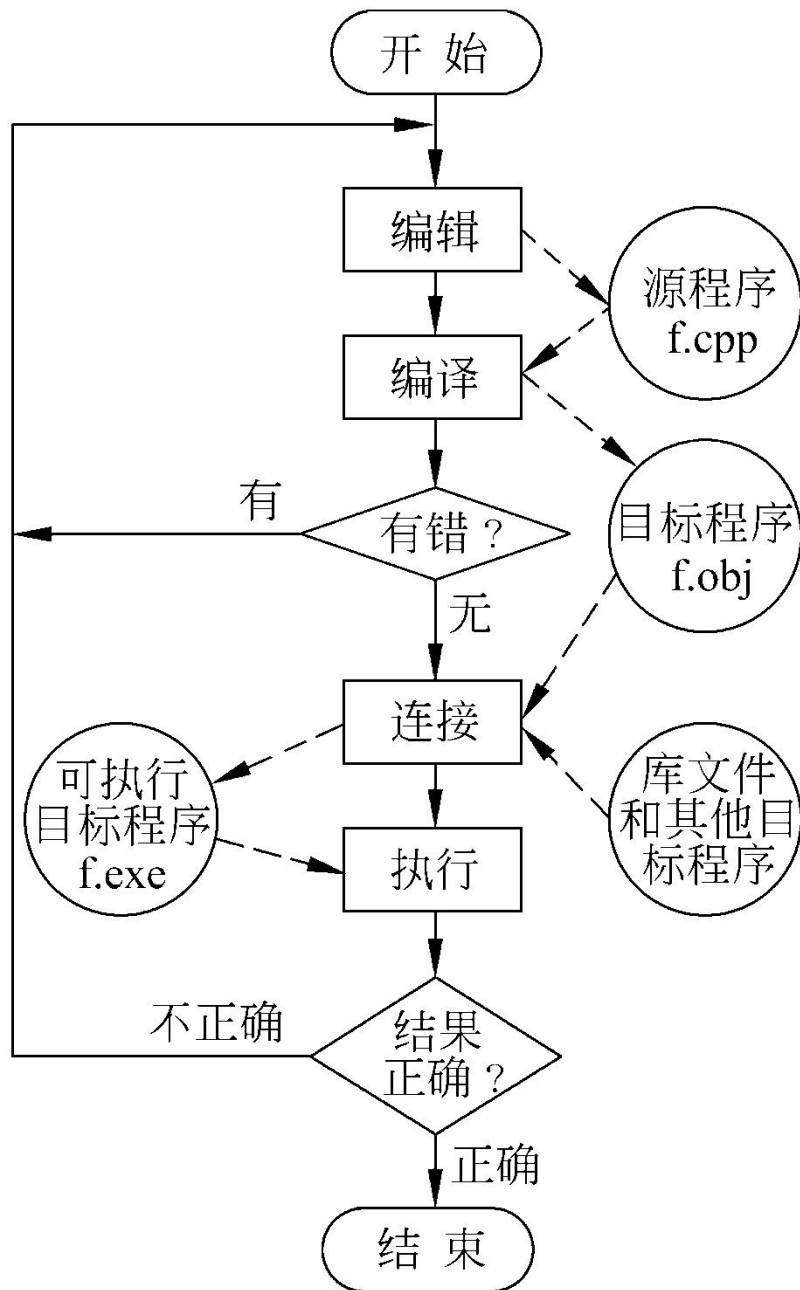


4. 运行程序

运行最终形成的可执行的二进制文件(.exe文件)，得到运行结果。

5. 分析运行结果

如果运行结果不正确，应检查程序或算法是否有问题。





1.5 关于C++上机实践

- 在了解了C++语言的初步知识后，尽快在计算机上编译和运行C++程序，以加深对C++程序的认识，并初步掌握C++的上机操作。
- 可以使用不同的C++编译系统，在不同的环境下编译和运行一个C++程序。应当能在不同的C++环境下运行自己的程序，并且了解不同的C++编译系统的特点和使用方法，在需要时能将自己的程序方便地移植到不同的平台上。
- 在教材的参考书《C++程序设计题解与上机指导》一书中简单介绍了在Visual C++ 6.0和GCC两种典型的环境下运行C++程序的方法。

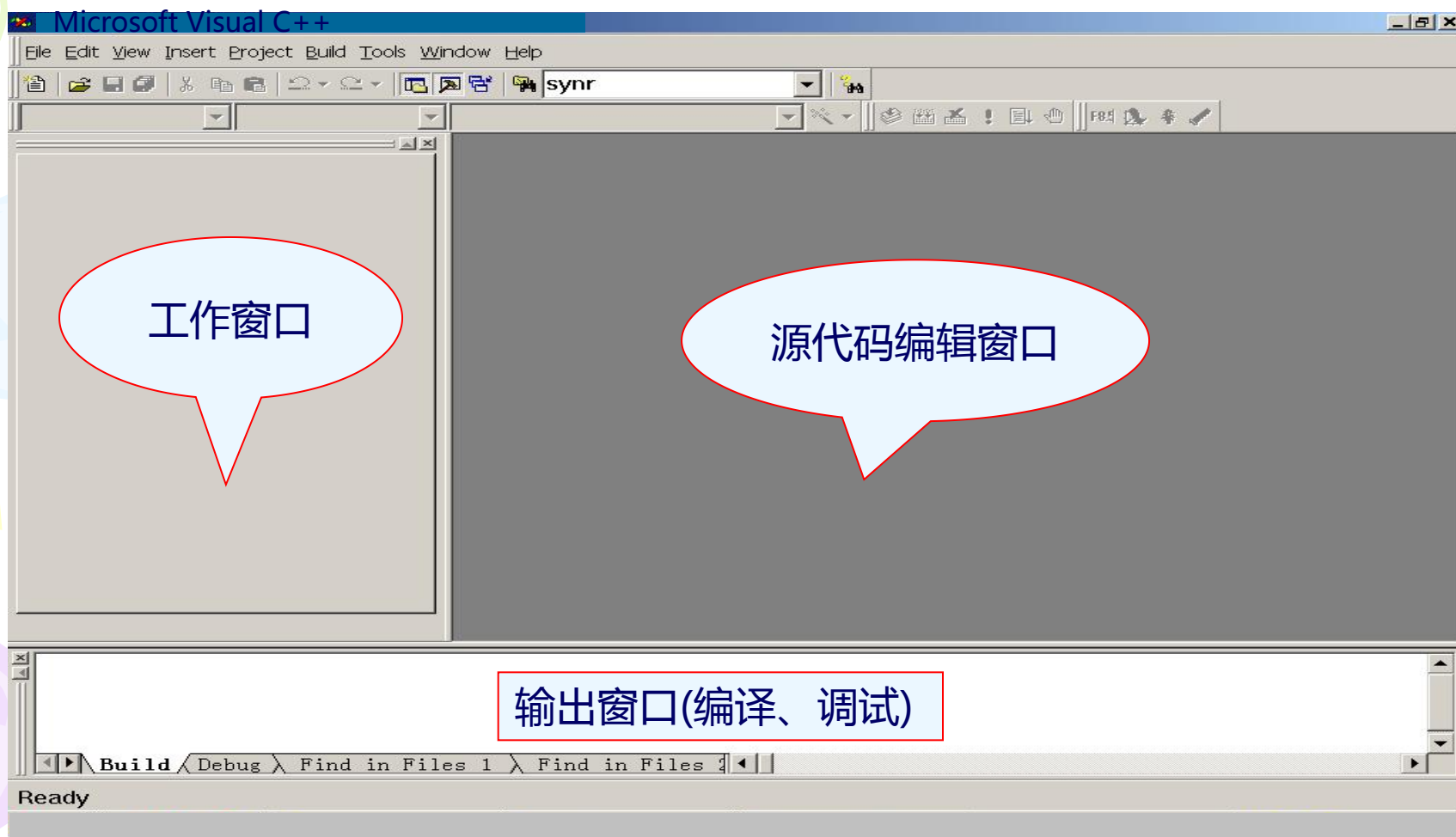


- 思考题：
- C和C++的优缺点。
- 学习C++程序设计的捷径是什么？
- 课本习题
- P16： 7、 9、 10
- 下一章：
- 什么是数据结构和数据类型？
- 数据在计算机中是如何存储的？

Visual C++集成环境使用



1、启动界面



2、工作窗口

打开方法：

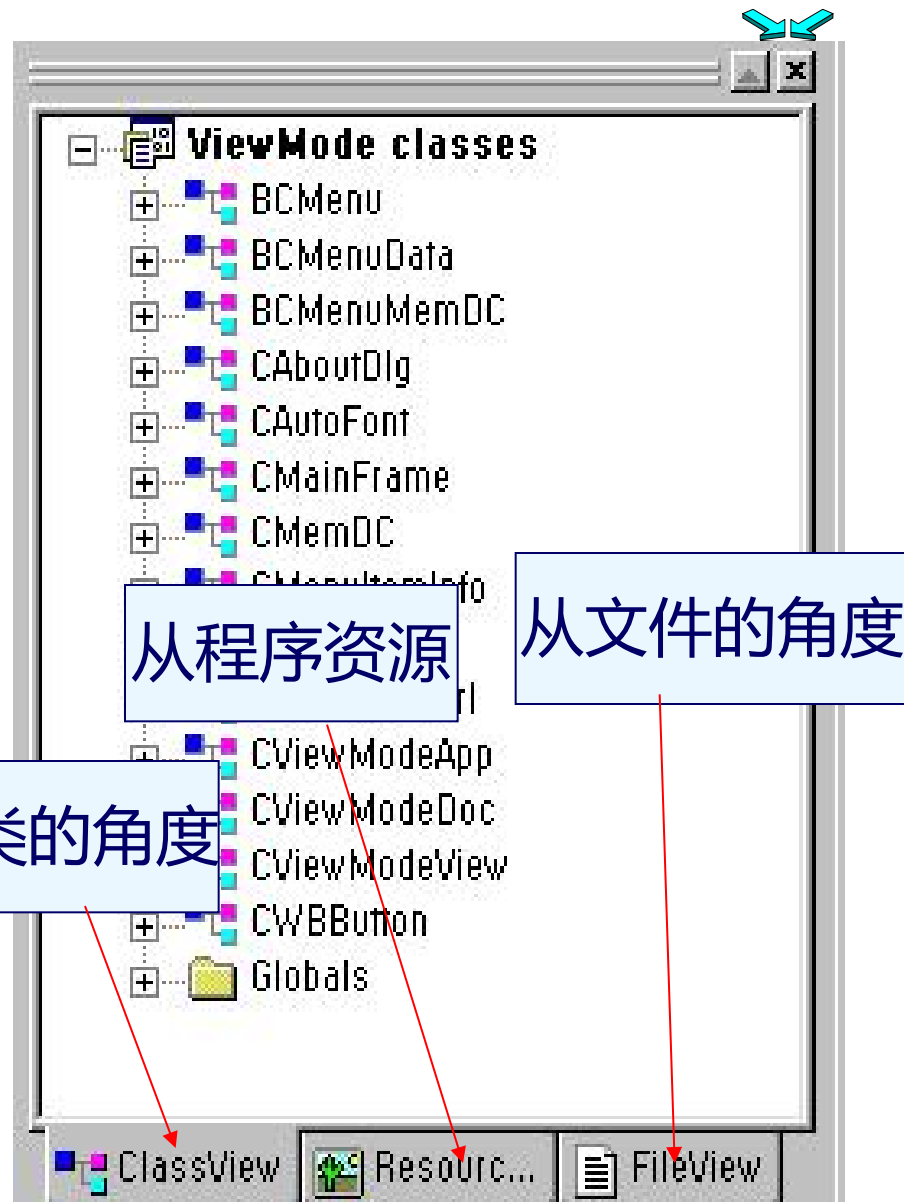
View→Workspace
是显示程序框架的地方，

可从三种角度观察程序：

ClassView—类视图

Resources—资源视图

FileView—文件视图





3、源代码编辑窗口

程序文件名

用户编辑程序、资源的全屏幕编辑器。在选择“工作窗口中的”项目时，会出现右图。

```
#include <iostream>
using namespace std;

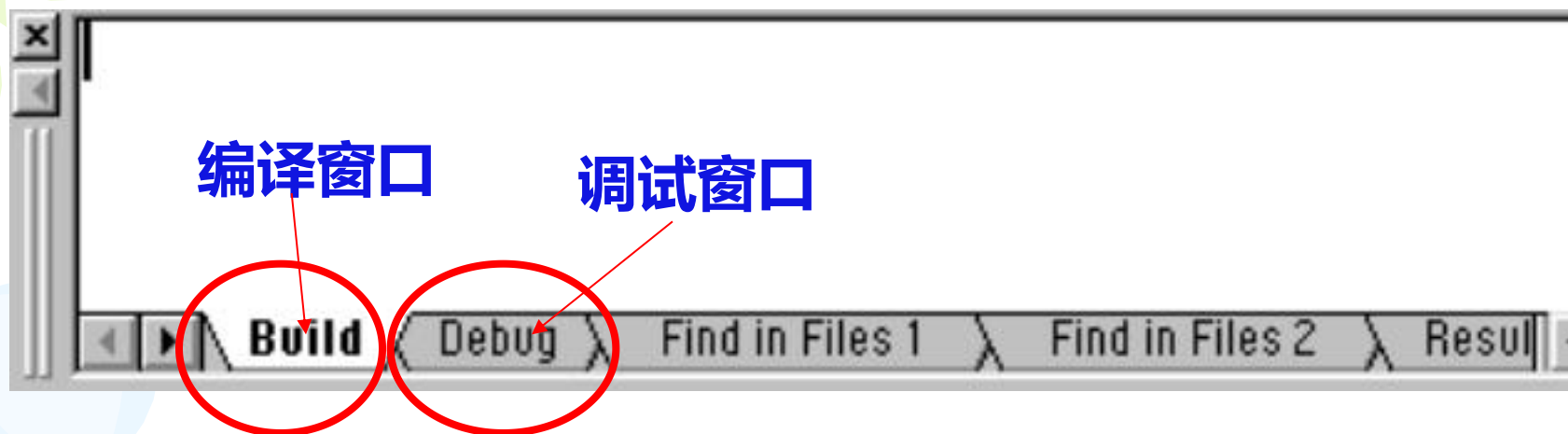
int main()
{
    int max(int x,int y);
    int a,b,c;

    cin>>a>>b;
    c=max(a,b);
    cout<<"max="<<c<<endl;

    return 0;
}

int max(int x,int y)
{
    if(x>y)
        return x;
    else
        return y;
}
```

4、输出窗口



输出窗口可以显示：

编译错误信息

(Error、
Warning)

调试变量状态

文件查找结果等信

息。

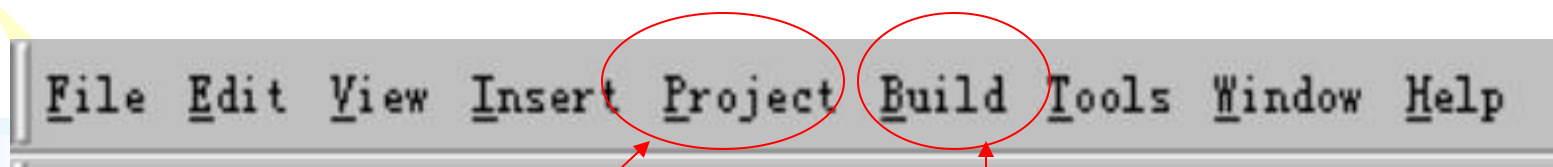
输出窗口，可关闭或打开：

组合键：Alt+2

菜单：view→Output



5、VC++菜单



工程管理
菜单项

编译，运
行程序菜
单项



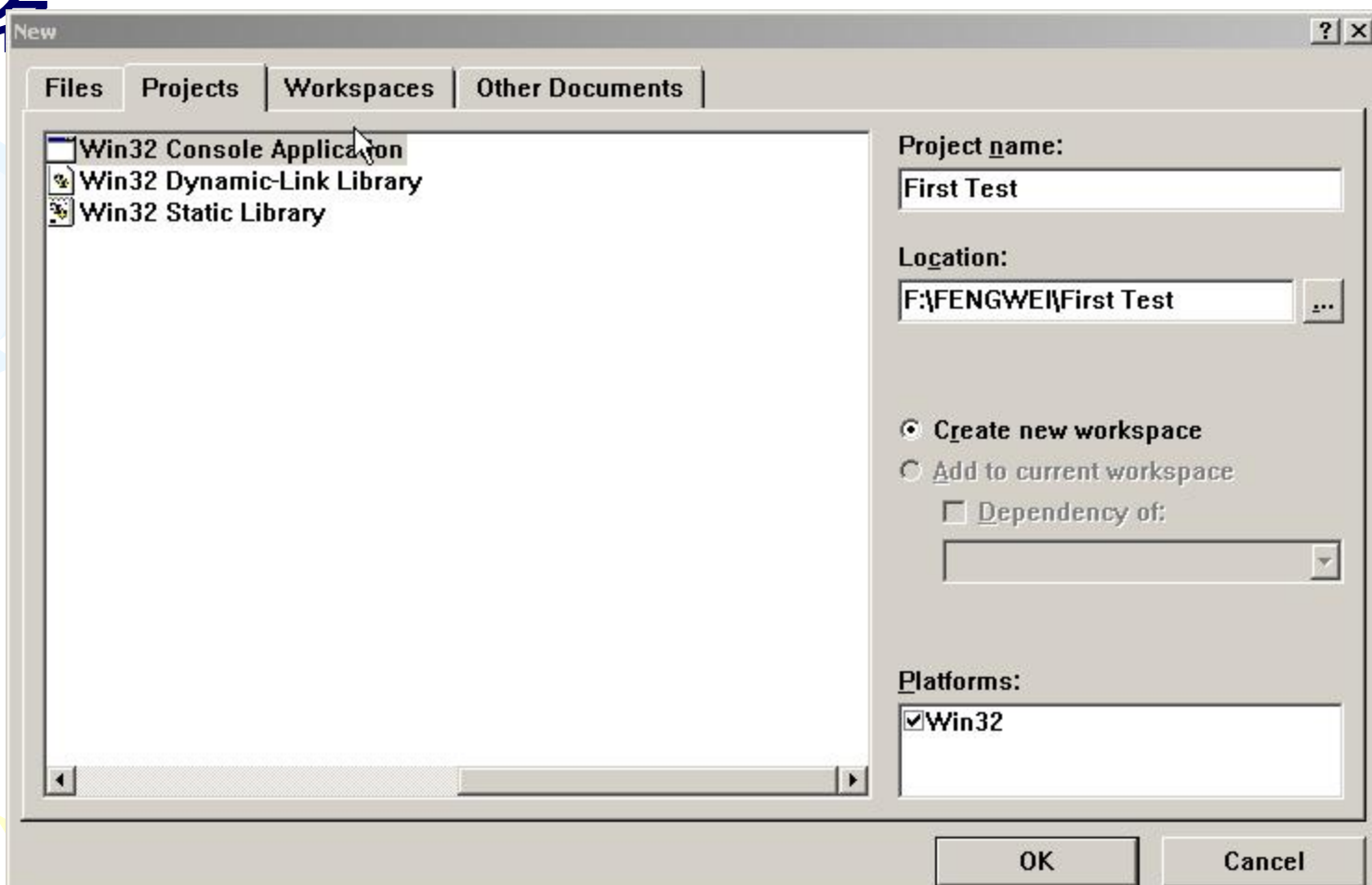
6、程序上机过程

- (1) 新建程序(File→New)
- (2) 输入或编辑程序
- (3) 保存程序(File→Save – Ctrl+s)
- (4) 编译程序 (Compile – Ctrl+F7)
- (5) 建立程序 (Build – F7)
- (6) 运行程序 (Execute – Ctrl+F5)



(1) 新建程序—第一步

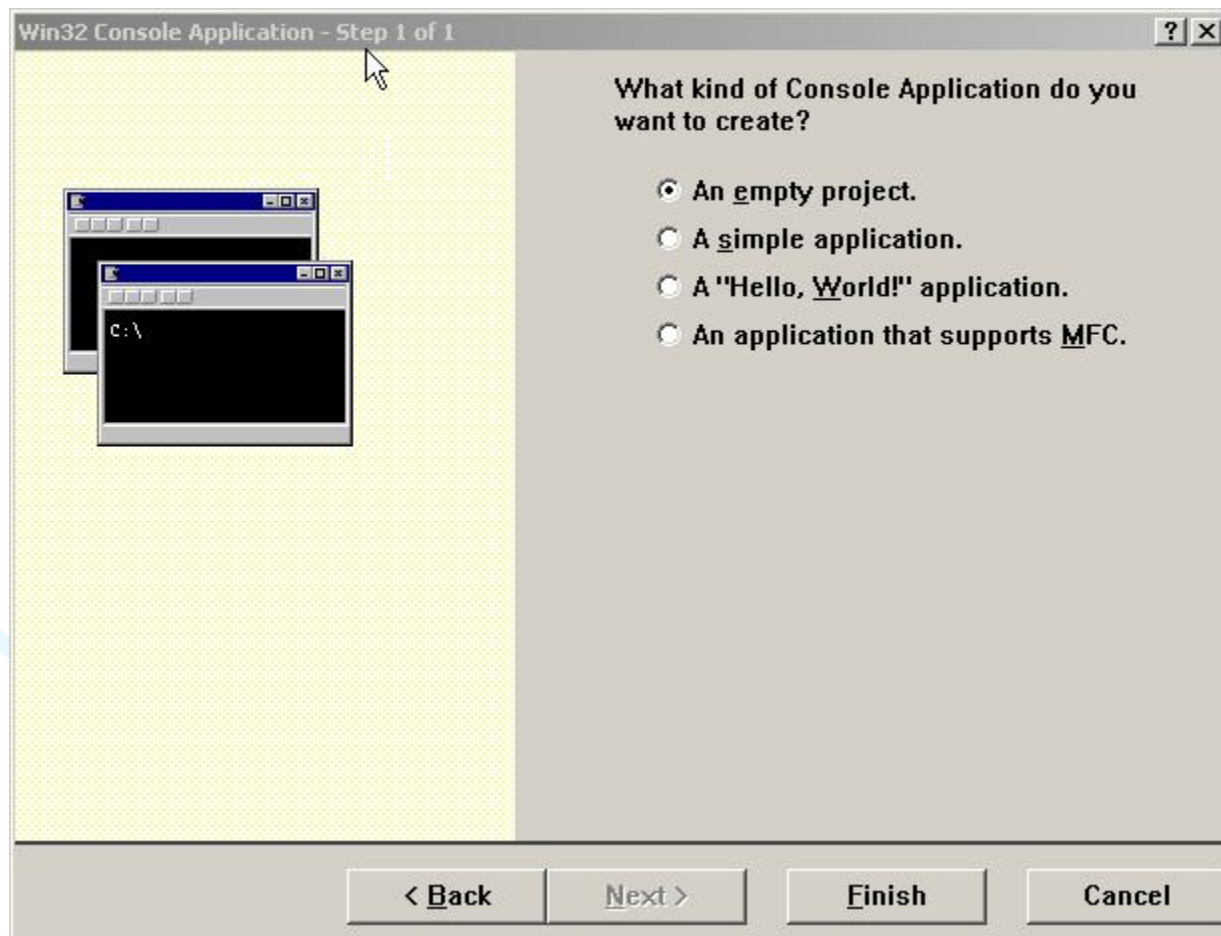
首先，在Windows资源管理器的F盘符下建立个人目录。
而后，单击File→New，在“新建项目”对话框中输入项目名





(1) 新建程序—第二步

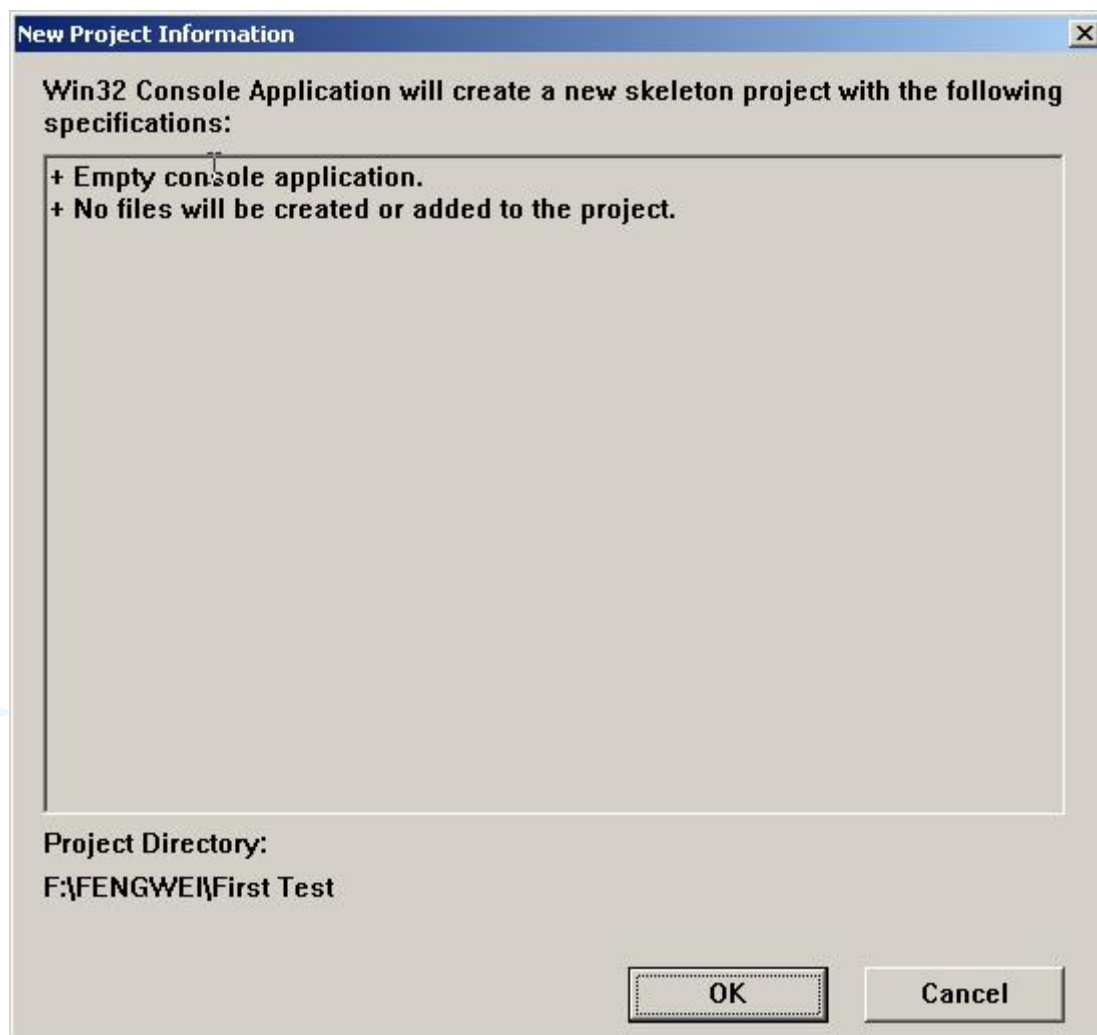
接上步，当按下“OK”按钮后，





(1) 新建程序—第三步

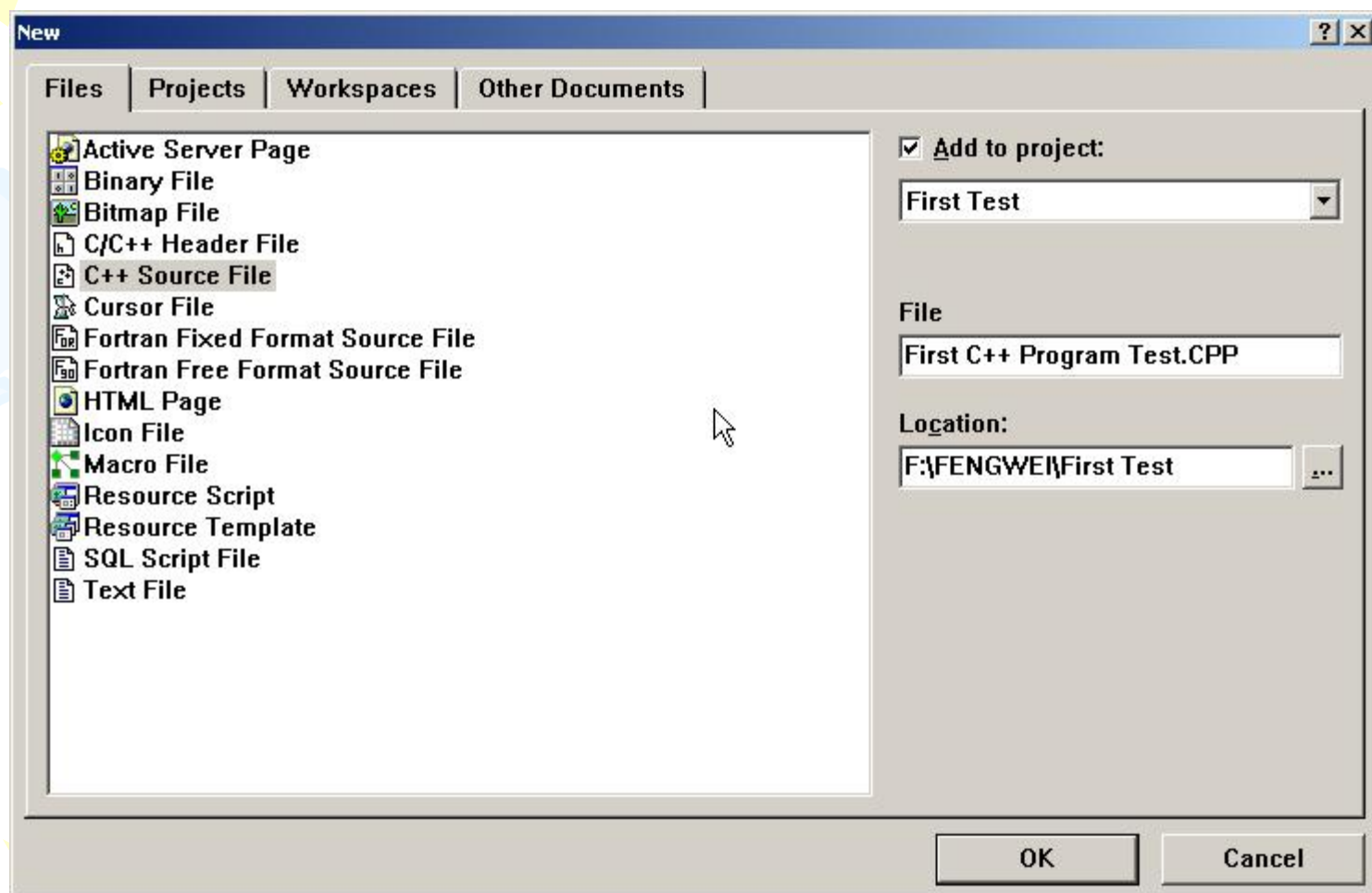
接上步，按下“Finish”按钮，按下“OK”按钮。





(1) 新建程序—第四步

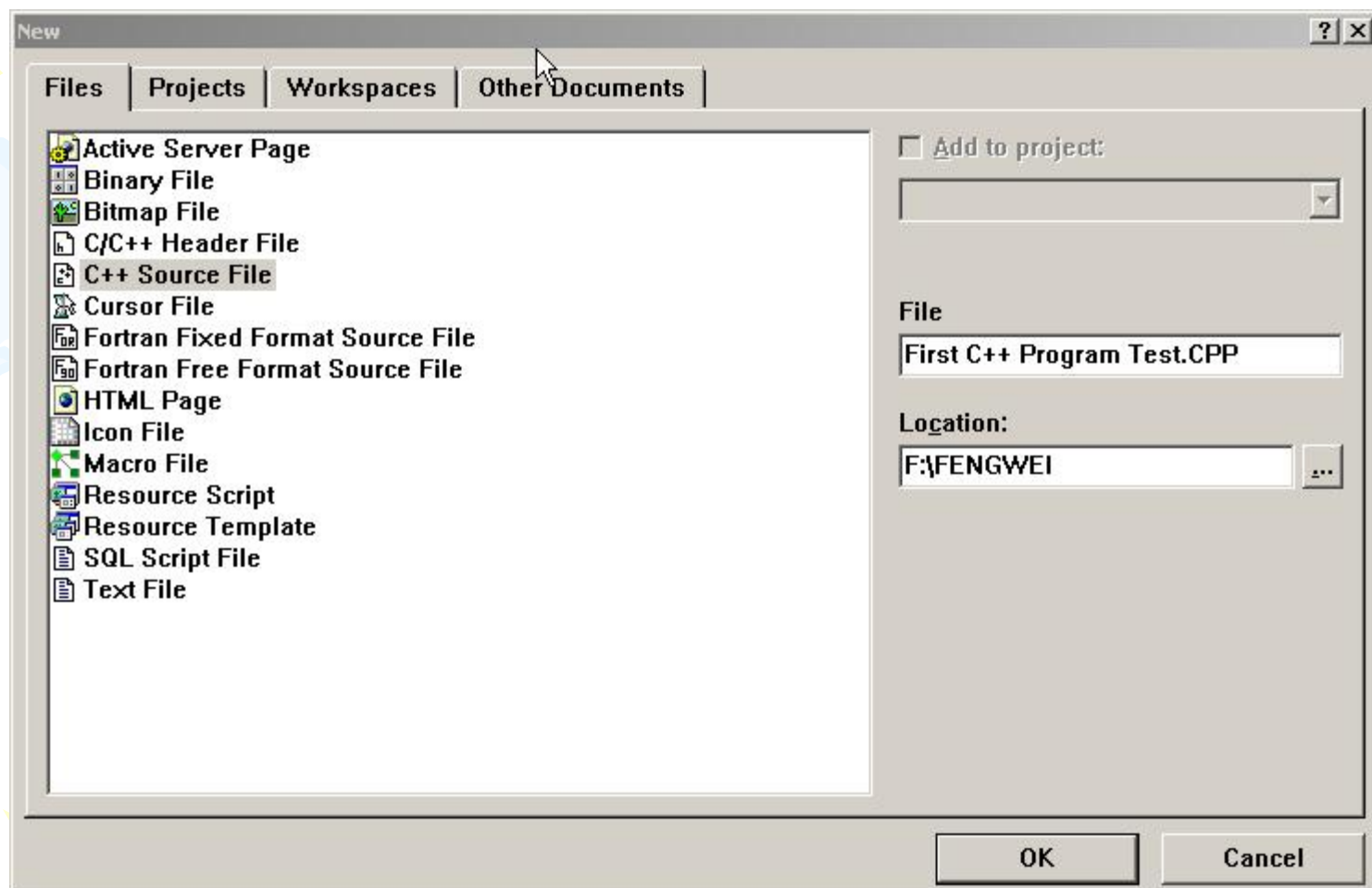
单击File→New，在“新建文件”对话框中输入后缀为.cpp的文件名，





(1) 新建程序—简化步骤

首先，在Windows资源管理器的F盘符下建立个人名字目录以及子目录名。而后，单击File→New，在“新建文件”对话框输入后缀为.cpp的文件名，





(2) 编辑程序

接上步，按下“OK”按钮后，输入、编辑程序

```
#include <iostream>
using namespace std;

int main()
{
    int max(int x,int y);
    int a,b,c;

    cin>>a>>b;
    c=max(a,b);
    cout<<"max="<<c<<endl;

    return 0;
}

int max(int x,int y)
{
```



(3) 保存程序

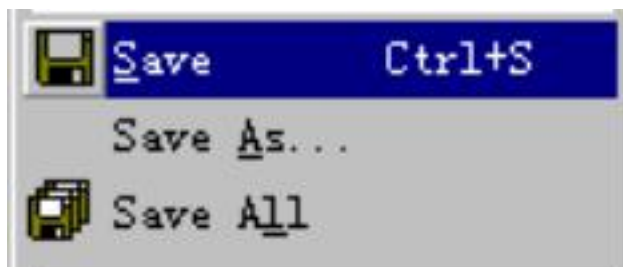
接上步，保存程序

方法一：按工具栏上“保存”或“保存所有”所有图标



方法二：按Ctrl+S

方法三：File→Save 或 File→Save All





(4、5、6) 编译、建立、运行程序

编译程序 : Compile (Ctrl+F7)。无错时的 “Output”窗口:

```
First_c_program_test.obj - 0 error(s), 0 warning(s)
```

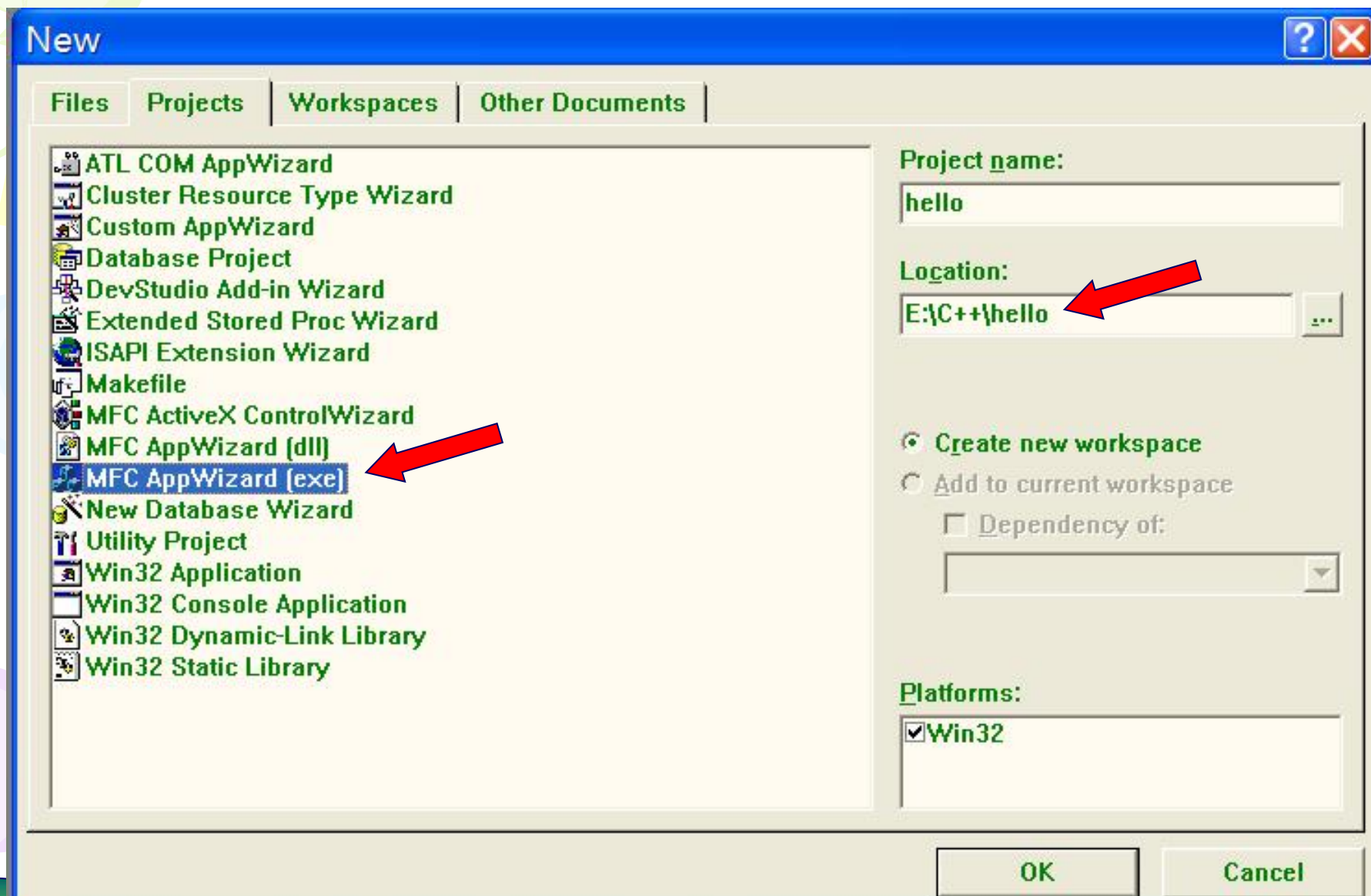
建立程序 : Build (F7)。无错时的 “Output”窗口:

```
Fisrt Test.exe - 0 error(s), 0 warning(s)
```

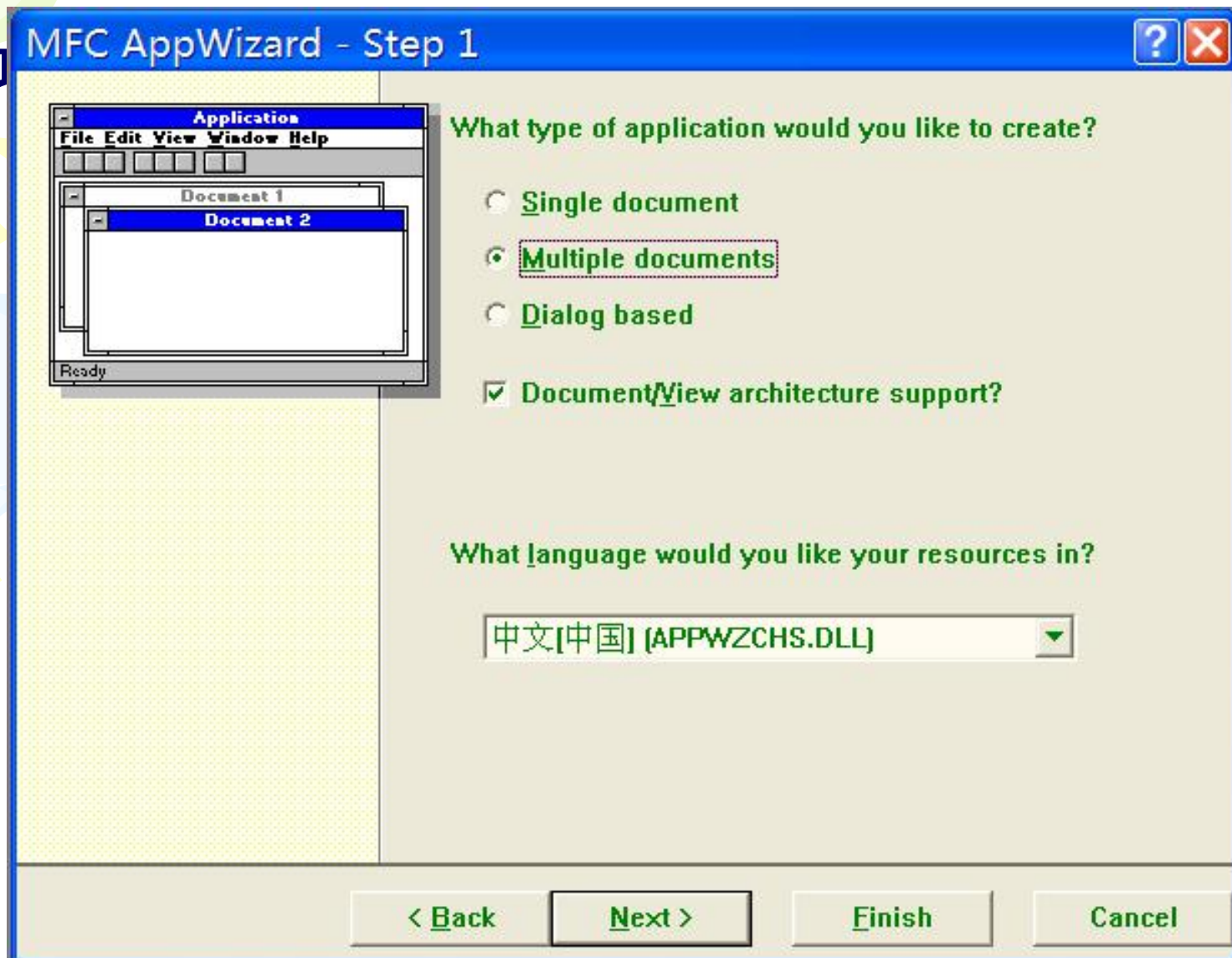
执行程序 : Execute (Ctrl+F5)。执行结果:

```
"F:\fengwei\Fisrt Test\Debug\Fisrt Test.exe"  
Input two numbers from keyboard:  
1,2  
a=1  
b=2  
max=2  
Press any key to continue_
```

用AppWizard创建一个MFC应用项目

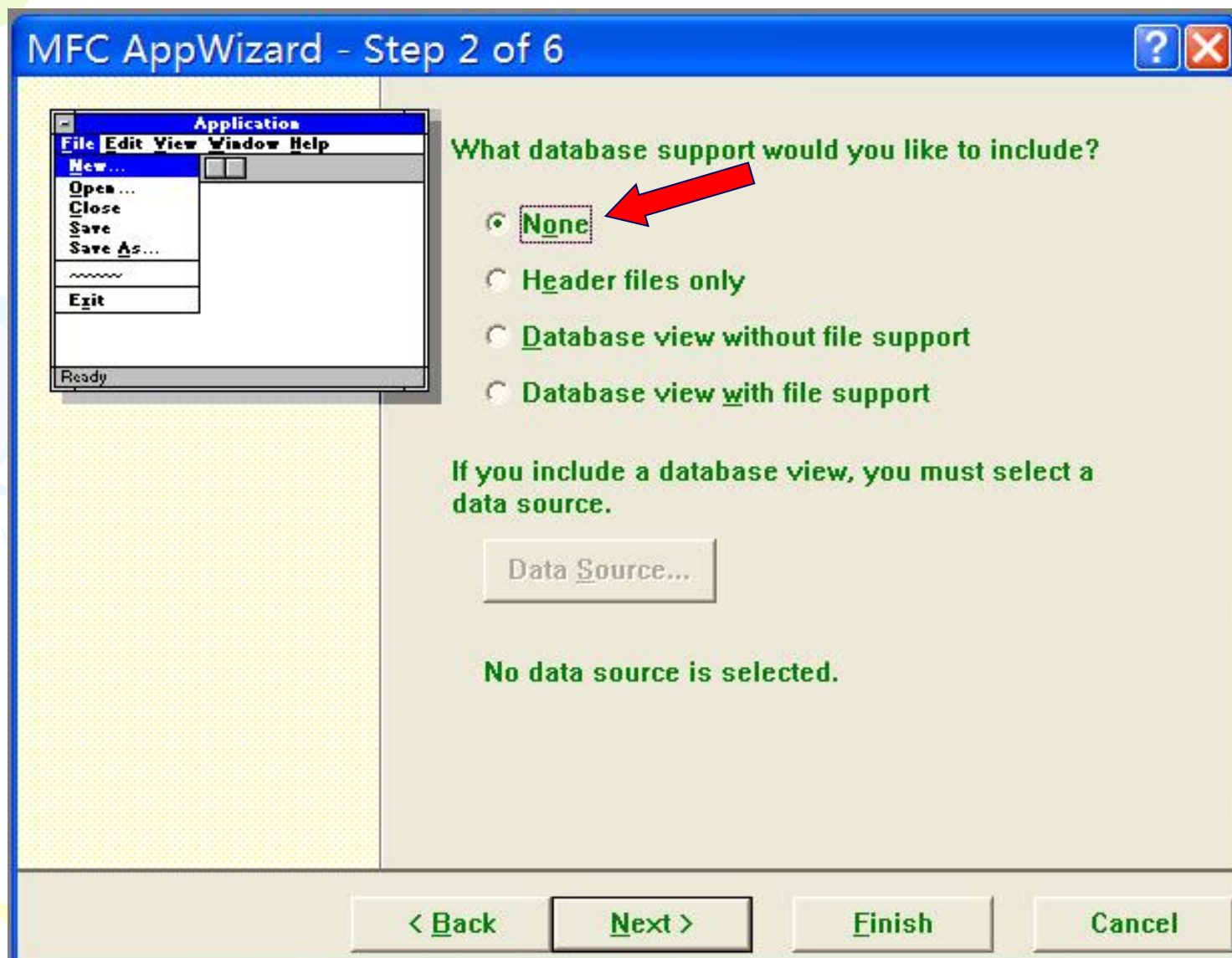


2. 指定应用项目的类型并选择语言（向导步骤1）✕✕





3. 选择数据库支持方式（向导步骤2）





4. 选择复合文档支持方式（向导步骤3）

MFC AppWizard - Step 3 of 6

What compound document support would you like to include?

☒ None

☐ Container

☐ Mini-server

☐ Full-server

☐ Both container and server

☐ Active document server

☐ Active document container

Would you like support for compound files?

☐ Yes, please

☒ No, thank you

What other support would you like to include?

☐ Automation

☒ ActiveX Controls

< Back Next > Finish Cancel

5. 指定窗口特征及工具条的风格等（向导步骤4）✂✂

MFC AppWizard - Step 4 of 6

What features would you like to include?

- ☒ Docking toolbar
- ☒ Initial status bar
- ☒ Printing and print preview
- ☐ Context-sensitive Help
- ☒ 3D controls
- ☐ MAPI (Messaging API)
- ☐ Windows Sockets

How do you want your toolbars to look?

- ☒ Normal
- ☐ Internet Explorer ReBars

How many files would you like on your recent file list?

4

Advanced...

Application

File Edit View Window Help

Print...
Print Preview
Print Setup...

Exit

Ready

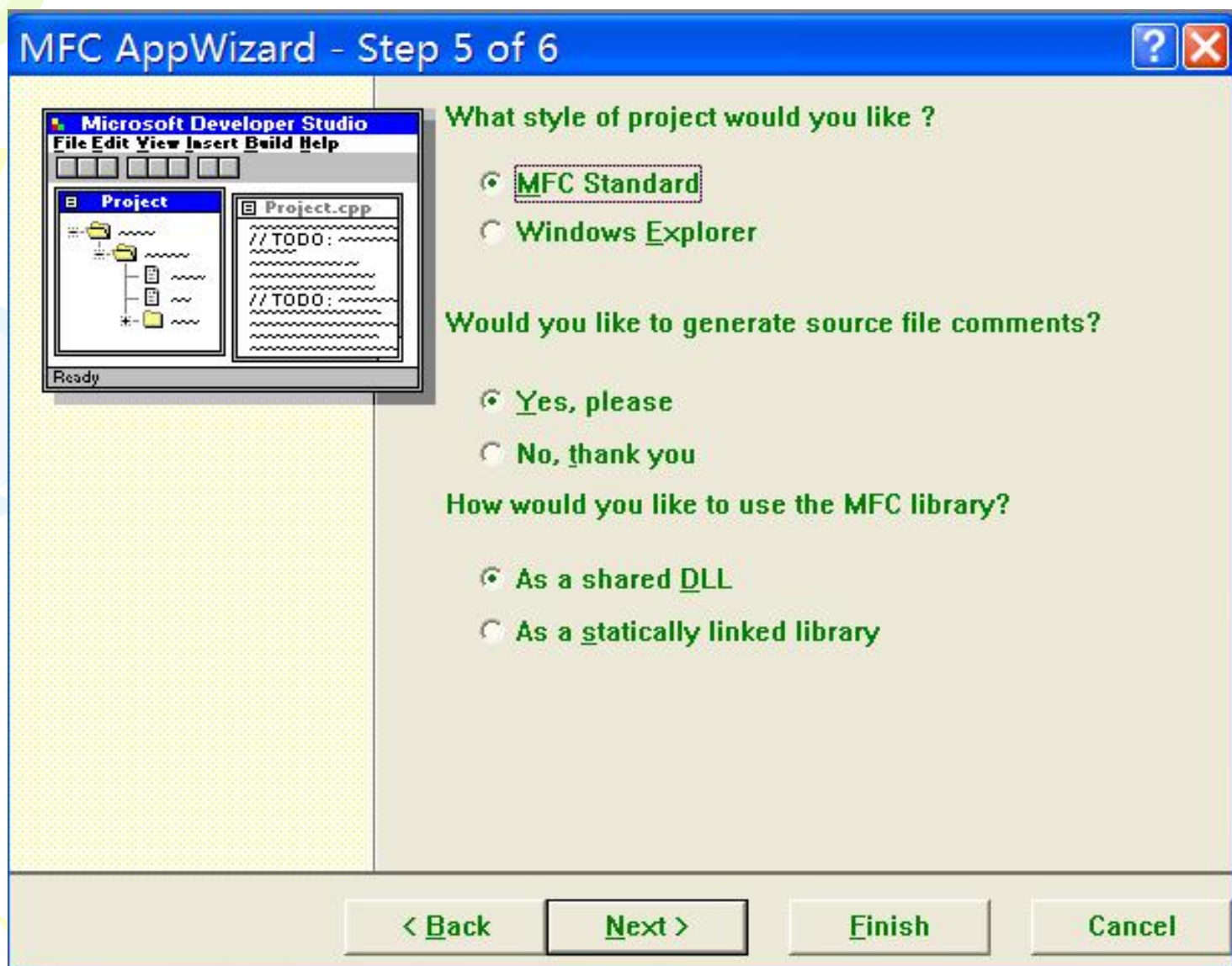
Editing Control: Record

☒ Check Box ☐ Radio Button
☐ Radio Button

< Back Next > Finish Cancel



6. 指定项目风格等（向导步骤5）



7. 第5步 New Project Information



步骤6)



AppWizard will create a new skeleton project with the following specifications:

Application type of hello:

Multiple Document Interface Application targeting:
Win32

Classes to be created:

Application: CHelloApp in hello.h and hello.cpp
Frame: CMainFrame in MainFrm.h and MainFrm.cpp
MDIChildFrame: CChildFrame in ChildFrm.h and ChildFrm.cpp
Document: CHelloDoc in helloDoc.h and helloDoc.cpp
View: CHelloView in helloView.h and helloView.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:
中文[中国]

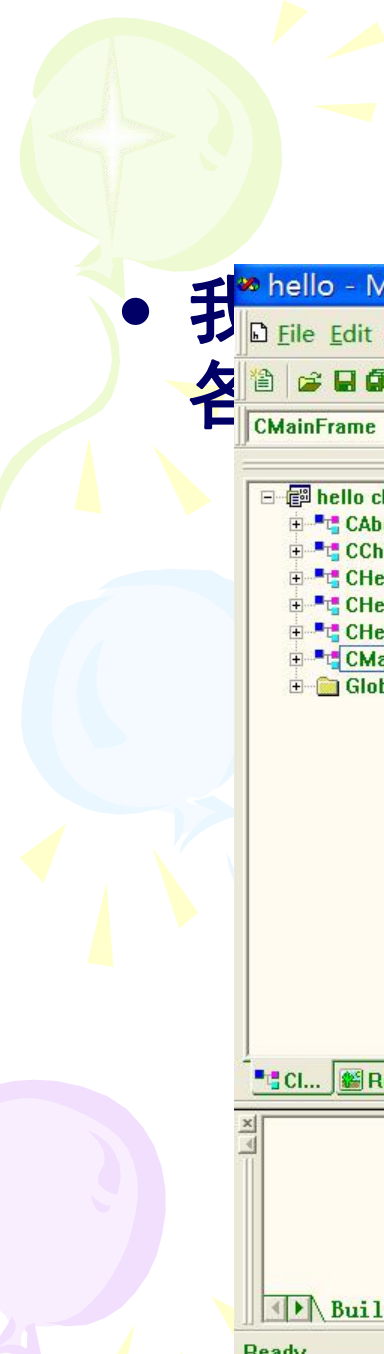
Project Directory:

E:\C++\hello



OK

Cancel

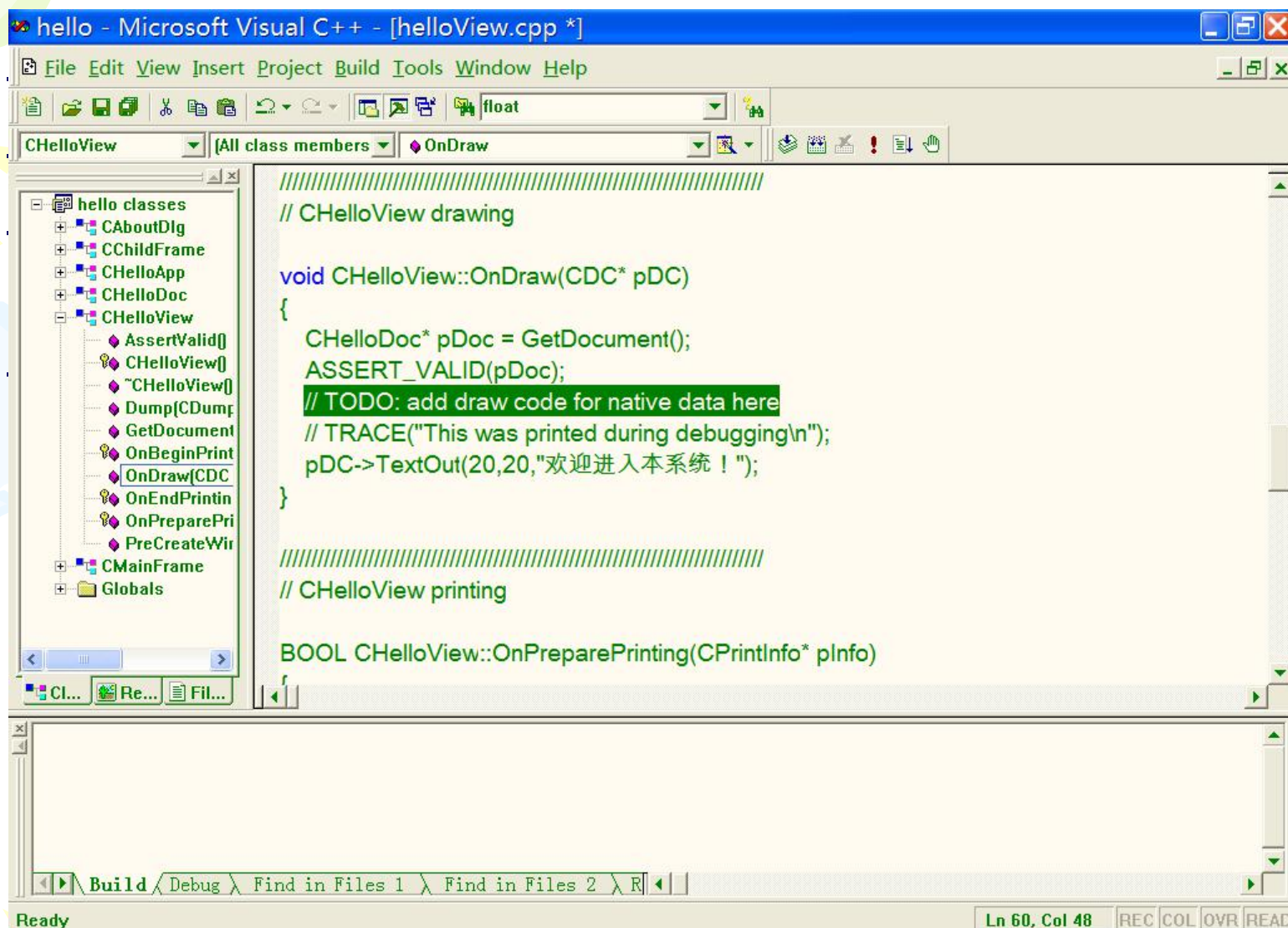


● 我答





8. 项目的编辑



类;

9&10. 创建及运行

