

高职高专计算机类专业“十二五”规划教材

C++面向对象程序设计

(第二版)

主 编 曹 静
副主编 董 宁 陈 丹



结构合理、有效

内容讲解由浅入深、循序渐进，章节编写方式层层展开、环环相套

内容充实、实用

涵盖C++主要知识点，各章重要知识点后面都附有典型的实训案例

语言严谨、精练

结合实训案例，清楚、准确地解释C++中众多且比较抽象的概念

快速掌握项目开发

项目开发与理论教学紧密结合，每章的项目设计都作为大型软件项目的一部分出现



中国水利水电出版社
www.waterpub.com.cn

高职高专计算机类专业“十二五”规划教材

C++面向对象程序设计

（第二版）

主 编 曹 静

副主编 董 宁 陈 丹



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书系统地介绍了面向对象程序设计的概念和 C++语言的基础知识。全书将编程语言和编程思想融为一体,使读者在一步步学习 C++语言的同时,可以一层层掌握面向对象的编程原理。为方便读者理解 C++面向对象程序设计的相关知识,书中采用通俗易懂的方式介绍了有关理论和原理,并辅以大量实例和插图。每章均配有相关内容的项目设计,便于读者进行实际操作和练习。

本书内容翔实、叙述清晰、注重实践、习题丰富,可作为高职高专各专业的计算机入门教材,也适合各领域想在计算机方面进修提高的人员自学使用。

本书配有导学、【课堂练习】参考代码、电子教案和源代码文件,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目 (C I P) 数据

C++面向对象程序设计 / 曹静主编. -- 2版. -- 北
京: 中国水利水电出版社, 2011.1
高职高专计算机类专业“十二五”规划教材
ISBN 978-7-5084-8125-8

I. ①C… II. ①曹… III. ①
C语言—程序设计—高等学校: 技术学校—教材 IV.
①TP312

中国版本图书馆CIP数据核字(2010)第232876号

策划编辑: 杨庆川 责任编辑: 李 炎 加工编辑: 李 皓 封面设计: 李 佳

书 名	高职高专计算机类专业“十二五”规划教材 C++面向对象程序设计 (第二版)
作 者	主 编 曹 静 副主编 董 宁 陈 丹
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16开本 19.25印张 488千字
版 次	2006年8月第1版 2011年1月第2版 2011年1月第3次印刷
印 数	7001—11000册
定 价	30.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

C++语言是由AT&T贝尔实验室的Bjarne Stroustrup建立的一种功能强大的高级编程语言，是最早被用于商业开发的面向对象编程语言。使用面向对象编程方法后，我们可以高效地开发极为复杂的软件系统并且可以保证软件的可靠性和可维护性，同时能提高软件的重用性、灵活性和扩展性。本书的目标是把C++面向对象程序设计作为大学生学习编程的第一门课程，而且设计成适合所有专业的学生。作者假定他们预先没有学过任何编程语言，而所用的数学与理科知识也仅限于一年级大学生的水平。

本书是全国示范性软件职业学院计算机及其相关专业指定教材，针对全国示范性软件职业学院特点，淡化理论，以够用为度，强化技能，重在实际操作，以项目设计贯穿全书，再辅以大量实例，让学生在实践中掌握C++语言和面向对象的软件开发方法。

本书是作者在多年的教学实践和科学研究的基础上，并参阅了大量国内外相关教材后，几经修改而成。主要特点如下：

1. 知识点全面，语言严谨、精练

C++中概念众多且比较抽象，本书对这些概念进行了清楚准确的解释并结合实例说明，让读者能全面掌握每一个知识点。

2. 实际项目开发与理论教学紧密结合

为了使读者快速掌握C++相关知识的使用方法，本书在各个章节的重要知识点后都附有典型的实训案例和单元练习，最后一章更结合各章的实训内容给出一个完整的项目。

3. 结构合理、有效

本书按照由浅入深的顺序，循序渐进地介绍了C++面向对象程序设计的相关知识。各个章节在编写的时候都是层层展开，环环相套的。

4. 内容充实、实用

本书不再使用传统教材中那些最多只有数百行，且在实践中极少使用的示例程序，各章实例都是实用性很强的程序或程序模块，并且都是作为一个大型软件项目的一部分出现，实例代码可维护性强，重用率高，真正体现了面向对象程序开发的思想。

5. 本书配有全部的程序源文件和电子教案

为方便读者使用，本书配有导学、【课堂练习】参考代码、电子教案和源代码文件，读者可以从中国水利水电出版社网站和万水书苑免费下载，网址为：<http://www.waterpub.com.cn/softdown/>和<http://www.wsbookshow.com>。

本书共计四部分，其中第一部分（包括第1章至第3章）主要介绍C++基本语法概念和程序结构；第二部分（包括第4章至第6章）主要介绍类、类的封装、继承和多态等；第三部分（包括第7章至第9章）主要介绍I/O流、模板和异常处理；第四部分（包括第10章）是一个综合全书内容的实训项目。

本书由曹静担任主编，董宁、陈丹担任副主编，张宇、张松慧、文凡、刘嵩、刘洁、肖英参加编写，董宁、陈丹、库波统编全稿。

读者朋友在阅读本书的过程中，如果觉得有疑问或不妥之处，请与编者联系，帮助我们改正提高，编者将不胜感激。作者E-mail: jingcao_2002@163.com。

编 者

2010年10月

目 录

前言

第1章 面向对象程序设计概述	1	2.4.2 函数的参数传递	40
1.1 面向对象的基本概念	1	2.4.3 函数参数的引用传递	41
1.2 C++面向对象程序设计	2	2.4.4 带默认形参值的函数	43
1.3 C++程序的基本组成	3	2.4.5 函数的嵌套与递归调用	43
1.4 Microsoft Visual Studio 2008 开发环境	6	2.4.6 内联函数	46
【实训1】简单 C++程序实例	11	2.4.7 函数的重载	47
本章小结	12	【实训3】函数的应用	48
习题	12	2.5 构造类型	50
第2章 C++语言基础	13	2.5.1 数组与字符串	50
2.1 数据类型和基本输入输出	13	2.5.2 结构体	55
2.1.1 基本数据类型	13	2.5.3 共用体	59
2.1.2 常量	14	2.5.4 枚举类型	59
2.1.3 变量	17	2.5.5 用 typedef 定义类型	61
2.1.4 基本输入、输出	18	2.6 指针	62
2.2 运算符和表达式	19	2.6.1 指针和指针变量	62
2.2.1 算术运算符和算术表达式	19	2.6.2 &和*运算符	63
2.2.2 赋值运算符和赋值表达式	20	2.6.3 指针的赋值与运算	63
2.2.3 逗号运算符和逗号表达式	21	2.6.4 指针和数组	65
2.2.4 关系运算符和关系表达式	21	2.6.5 指针和结构体	69
2.2.5 逻辑运算符和逻辑表达式	21	2.6.6 指针和函数	70
2.2.6 ++、--与 sizeof 运算符	22	2.6.7 动态内存分配 new 和 delete	73
2.2.7 位运算符	23	【实训4】指针的综合应用	74
2.2.8 三目运算符	25	项目设计1 控制台程序的人机界面设计	80
2.2.9 运算符的优先顺序	25	本章小结	84
2.2.10 运算式中的类型转换	26	习题	85
2.3 基本控制结构	27	第3章 C++程序的结构	88
2.3.1 表达式语句、空语句和复合语句	28	3.1 作用域和可见性	88
2.3.2 选择控制语句	29	3.1.1 作用域	88
2.3.3 循环控制语句	32	3.1.2 可见性	89
2.3.4 break 和 continue 语句	36	3.2 生存期	90
【实训2】程序控制结构的应用	37	3.2.1 静态生存期	90
2.4 函数	39	3.2.2 动态生存期	90
2.4.1 函数的定义和调用	39	3.3 变量的存储类型	91

3.3.1 局部变量	91	4.7.4 浅拷贝和深拷贝	148
3.3.2 全局变量	92	4.7.5 this 指针	152
3.3.3 外部变量	93	【实训 7】对象数组与指针的综合应用	152
3.4 编译预处理和多文件结构	95	4.8 类的组合	156
3.4.1 编译预处理命令	95	4.8.1 类的组合	156
3.4.2 多文件组织结构	97	4.8.2 前向引用声明	160
项目设计 2 预处理及多文件结构	99	4.9 友元	161
本章小结	103	4.9.1 友元函数	161
习题	103	4.9.2 友元类	163
第 4 章 类与对象	105	项目设计 3 面向对象程序设计中类的应用	164
4.1 面向对象程序设计的基本特点	105	本章小结	171
4.1.1 抽象	105	习题	172
4.1.2 封装	106	第 5 章 类的继承与派生	173
4.1.3 继承	107	5.1 继承与派生	173
4.1.4 多态	108	5.1.1 基类与派生类	173
4.2 建立类与对象	109	5.1.2 派生类的声明	174
4.2.1 类的声明与实现	109	5.1.3 派生类生成过程	175
4.2.2 类成员的访问控制	110	5.2 继承的三种方式	176
4.2.3 类的成员函数	112	5.2.1 公有继承	177
4.2.4 对象的建立与使用	115	5.2.2 私有继承	179
【实训 5】类的定义与对象的使用	118	5.2.3 保护继承	182
4.3 构造函数和析构函数	120	5.3 派生类的构造函数和析构函数	184
4.3.1 构造函数	120	5.3.1 构造函数	184
4.3.2 析构函数	126	5.3.2 析构函数	187
4.4 类的作用域和对象的生存期	128	5.4 多继承和虚基类	190
4.4.1 类的作用域	128	5.4.1 派生类成员的标识与访问	190
4.4.2 对象的生存期	129	5.4.2 作用域标识	190
4.5 类的静态成员	131	5.4.3 多继承和虚基类	191
4.5.1 静态数据成员	131	5.4.4 虚基类及其派生类的构造函数	193
4.5.2 静态函数成员	133	5.5 赋值兼容规则	194
【实训 6】对象与类成员的应用	137	【实训 8】类继承与派生的应用	197
4.6 常类型	138	项目设计 4 类的组织结构在校园信息	
4.6.1 常引用	138	管理系统中的应用	200
4.6.2 常对象	139	本章小结	208
4.6.3 用 const 修饰的类成员	139	习题	208
4.7 数组、指针与对象	143	第 6 章 多态性与虚函数	209
4.7.1 对象数组	143	6.1 多态性概述	209
4.7.2 对象指针	145	6.1.1 多态的类型	209
4.7.3 动态配置对象内存	146	6.1.2 多态的实现	209

6.2 重载与再定义	210	8.2.1 数据流的格式控制	254
6.2.1 函数的重载	210	8.2.2 常用的输入/输出函数	259
6.2.2 函数的再定义	211	8.2.3 数据流的错误侦测	260
6.3 运算符重载	212	8.3 文件的输入/输出	261
6.3.1 运算符重载的规则	213	8.3.1 fstream 类	261
6.3.2 运算符重载为成员函数	214	8.3.2 格式化文字模式的文件 I/O	263
6.3.3 运算符重载为友元函数	219	8.3.3 二进制模式的文件 I/O	265
6.4 虚函数	221	8.3.4 文件指针的使用	266
6.4.1 一般虚函数成员	222	【实训 10】利用文件存储对象	269
6.4.2 虚析构函数	224	项目设计 7 校园信息管理系统中	
6.5 抽象类	225	的数据存储	271
6.5.1 纯虚函数	226	本章小结	278
6.5.2 抽象类	226	习题	278
【实训 9】类的多态性综合应用	228	第 9 章 异常处理	279
项目设计 5 多态在校园信息管理系统中		9.1 异常处理的基本思想	279
的应用	232	9.2 C++异常处理的实现	279
本章小结	240	9.2.1 异常处理的机制	279
习题	240	9.2.2 异常处理的规则	281
第 7 章 模板	242	9.3 异常处理中的构造与析构	283
7.1 模板的概念	242	本章小结	284
7.2 函数模板	242	习题	284
7.3 类模板	244	第 10 章 项目设计——校园信息管理系统	285
项目设计 6 顺序表模板	248	10.1 系统分析与设计	285
本章小结	252	10.2 程序设计分析	285
习题	252	10.3 源程序及说明	287
第 8 章 数据流与文件的输入/输出	253	10.4 运行结果与分析	298
8.1 I/O 流的概念	253	参考文献	302
8.2 数据流的控制	254		

第 1 章 面向对象程序设计概述

教学要求

- 掌握: 面向对象的基本概念, 面向对象程序设计和结构化程序设计的区别。
- 理解: C++面向对象程序设计的特点, 一个基本 C++程序的组成, Microsoft Visual Studio 2008 的开发环境。
- 了解: 计算机程序设计语言的发展过程。

1.1 面向对象的基本概念

计算机程序设计语言是计算机可以识别的语言, 供计算机阅读和执行, 用于描述解决问题的方法。计算机程序就是用计算机程序设计语言所要求的规范书写的一系列动作, 它表达了程序员要求计算机执行的操作。

最早使用的计算机程序设计语言是机器语言, 它是由计算机硬件系统可以识别的二进制指令组成的语言。不久出现了汇编语言, 它是一种用符号表示, 面向某一特定机型的程序设计语言, 它的指令与机器指令一一对应。为了提高编制程序的效率和程序的可读性, FORTRAN, COBOL、Pascal、C、C++等高级语言相继出现。

高级程序设计语言的出现屏蔽了计算机硬件的细节, 提高了语言的抽象层次, 程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这就使软件设计师在书写程序时可以联系到程序所描述的具体事物, 从而大大提高了编写程序的效率。

传统软件开发应用的是结构化程序设计方法, 把程序看成是处理数据的一系列过程, 采用了模块分解与功能抽象以及自顶向下、分而治之的方法, 从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务, 便于开发和维护, 成功地处理复杂问题提供了有力手段。

但是当程序需要处理的数据量增大时, 数据与处理这些数据的方法之间的分离使程序变得越来越难以理解。当对某段程序进行修改或删除时, 整个程序中所有与之相关的部分都要进行相应的修改, 从而使程序代码的维护变得困难。对数据处理能力的需求越强, 这种分离所造成的负作用越显著。为了克服结构化程序模式在设计系统软件和大型应用软件时所存在的缺陷, 面向对象模式应运而生。

结构化程序设计是以功能为中心描述系统, 而面向对象程序设计是围绕现实世界的概念组织模块, 以数据为中心描述系统。面向对象程序设计把一个复杂的问题分解成多个能够完成独立功能的对象(类), 用对象来描述问题空间的实体, 用程序代码模拟现实世界中真实或抽象的对象, 使程序设计过程更自然、更直观。

面向对象程序设计的本质是把数据及对数据的操作方法放在一起, 作为一个相互依存、不可分离的整体——对象。对同类型对象抽象出其共性, 形成“类”。“类”中的大多数数据, 只能用本类的方法进行处理。“类”通过一个简单的外部接口与外界发生关系, 对象与对象之

间通过消息进行通信。这样,程序模块间的关系更为简单,程序模块的独立性、数据的安全性就有了良好的保障。

面向对象程序设计方法具有4个基本特征:抽象、封装、继承和多态性。

抽象(abstract)就是忽略事物的非本质特征,只注意那些与当前目标有关的本质特征,从而找出事物的共性,把具有共同性质的事物划分为一类,得出一个抽象的概念。面向对象方法中的“类”为具有相同属性和行为的一组对象提供抽象的描述,一个属于某类的对象称为该类的一个实例。

封装(encapsulation)就是把每个对象的数据(属性)和操作(行为)包装在一个类中,并尽可能隐藏对象的内部细节。一般限制直接访问对象的属性,而应通过操作接口访问,这样使程序中模块之间关系更简单、数据更安全。对程序的修改也仅限于类的内部,使得由于修改程序所带来的影响局部化。

继承(inheritance)是指一个新类可以从现有的类派生而来。新类继承了现有类的特性,包括一些属性和行为,并且可以修改或增加新的属性和行为,使之适合具体的需要。继承是面向对象技术能够提高软件开发效率的重要原因之一,它很好地解决了软件的可重用性问题。

多态性(polymorphism)是指允许不同类的对象对同一消息作出的响应不相同。一般类中定义的属性或行为,被特殊类继承后,可以具有不同的数据类型或者表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。

通过使用面向对象程序设计的抽象、封装、继承和多态性等机制,程序更易于维护、更新和升级。利用代码可重用性,程序员可以在程序中大量使用成熟的类库,从而缩短程序的开发时间,提高程序员的工作效率和程序的可靠性。

1.2 C++面向对象程序设计

面向对象程序设计语言很多,如Smalltalk、Ada、Eiffel、Object Pascal、Visual Basic、C++等。作为C语言的继承者,C++语言首先继承了C语言的以下特点:丰富的运算符和数据类型、结构化的程序设计方法、高效的机器代码、良好的可移植性。同时,由于C是C++的子集,这使得许多C语言代码不需要修改就可以被C++编译器编译通过。C++的错误检查机制强,它提供了专门的机制检查类,更适合大、中型程序的开发。

C++语言扩展了C语言的功能,它引入了一些新特性,强调代码的有效性和紧凑性。在程序设计方法方面,C++是一种混合型程序设计语言,它既支持结构化程序设计,又支持面向对象程序设计。C++既适合作为系统软件描述语言,也适于编写应用软件。作为程序设计者,C++的目标是为程序员的程序开发提供一个优良的设计工具,以编写模块化程度高、可重用性和可维护性俱佳的程序。因此,可以说C++是程序员必须掌握的语言。

本书主要介绍使用最为广泛的C++语言和使用C++语言进行面向对象程序设计的方法。在学习编程之前,我们首先简单了解一下C++程序的开发过程。在后续章节的学习和编程实践中,读者将对它们有更深入的理解。

程序的开发通常要经过编辑、编译、连接、运行调试等步骤。编辑是将源程序输入到计算机中,生成后缀为.cpp的磁盘文件。编译(compile)是将程序的源代码转换成机器语言代码。程序员编制的源程序被编译后,会生成一个目标文件,通常以.obj作为文件扩展名,该目标文件为源程序的目标代码。但是编译后的程序还不能由计算机执行,因为目标代码只是一个

个的程序块,需要相互衔接成为一个适应一定操作系统环境的程序整体。为了把它转换成可执行程序,必须进行连接(link)。C++程序通常是通过同时连接一个或多个目标文件以及一个或几个库而创建的。库(.lib)是一组由机器指令构成的程序代码,是可连接文件。库有标准库和用户生成的库。标准库是由C++提供的,用户生成的库是由软件开发商或程序员提供的。文件与库连接的结果,即生成一个扩展名为.exe的计算机可执行程序。

程序员首先在集成开发环境中编辑源程序,或在其他编辑器中输入源程序,然后在集成环境中启动编译程序将源程序转化为目标文件。在编译和连接时,都会对程序中的错误进行检查,并将查出的错误显示在屏幕上。编译阶段查出的错误称为语法错误,连接时查出的错误称为连接错误。如果出现编译错误或连接错误,程序员必须回到编辑状态下重新开始编辑程序和编译、连接的过程,直到排除所有错误,成功生成可执行文件为止。

1.3 C++程序的基本组成

为了能够更加直观地认识C++语言,我们就从最简单的例子分析C++程序的构成。

例1-1的程序代码就是一个完整的可运行的C++程序,它可以输出两个句子。

【例1-1】第一个C++程序

```
//Ch01_01.cpp
#include <iostream>
using namespace std;
void main()
{
    cout<< "Hello,World!\n ";
    cout<< "I'm a student.\n ";
}
```

程序运行结果为:

```
Hello,World!
I'm a student.
```

一个C++程序由三部分组成,分别是注释部分、编译预处理和程序主体。

程序的注释部分是程序员为读者作的说明,是提高程序可读性的一种方法。一般将注释分为序言注释和注解性注释两种。前者位于程序的开始部分,用以说明程序或文件的名称、功能、编写的时间、作者以及输入输出说明等,而后者位于程序主体中,对难以理解的语句作说明。

C++程序中,有两种给出注释的方法。这两种注释方法与C语言相同,一种是使用“/*”和“*/”括起注释文字,它可以占多行。例如:

```
/* This is my first program.
This is comment.
*/
```

这里“/*”和“*/”之间的内容都会作为注释处理。而另一种方法就是例1-1中的注释,它是从“//”开始直至换行,中间的部分就是注释。

注释部分仅仅用于说明,提高程序的阅读质量,并不是程序的必要部分,它是可选的。在生成可执行程序之前,C++的编译系统在对源程序进行编译的时候不处理注释部分,因此注释部分对程序的功能实现不起任何作用。由于编译系统忽略注释部分,所以注释部分也不会增加最终产生的可执行程序的文件大小。

每个以符号“#”开头的行，称为编译预处理行。如本例中“#include”称为编译指令。而“#include <iostream>”指示编译器在对程序进行预处理时，将文件 iostream.h 中的代码嵌入到程序中该指令所在的地方。文件 iostream.h 中设置了 C++ 的 I/O 相关环境，定义了输入输出流对象 cin 和 cout 等。由于这类文件常被嵌入在程序的开始处，所以称之为头文件。在 C++ 程序中如果使用了系统中提供的一些功能，就必须嵌入相关的头文件。using namespace std; 表示使用 C++ 标准中定义的 std 类库中的头文件。

在程序接下来的部分中，main 是主函数名，函数是 C++ 程序中最小的功能单位，每一个 C++ 程序都必须有一个 main() 函数，并且 main() 表明了程序执行的开始点。main() 函数之前的 void 表示 main() 函数没有返回值。函数体用一对大括号 {} 包含。用程序语句来描述一个函数所执行算法的过程称为函数定义。在上述例子中，main() 函数头和函数体构成了一个完整的函数定义。

程序由语句组成，每条语句都以分号作为语句的结束符。在 main() 函数体中，cout 是一个代表标准输出的流设备，它是 C++ 系统预定义的对象，其中包含了用于输出的功能。当程序要在设备上输出时，就需要在程序中指定该对象。输出操作由操作符“<<”来表达，其作用是将紧随其后的双引号中的字符串输出到标准输出设备上。

程序中用双引号括起的部分“Hello,World!\n”称为字符串常量。其中的字符“\n”表示一个换行控制符。

下面我们再看一个进行简单计算的程序。

【例 1-2】一个进行加法计算的 C++ 程序

```
// Ch01_02.cpp
#include <iostream>
using namespace std;
void main()
{
    int m,n,t;
    cout<<" Please input two integers: \n";
    cin>>m>>n;
    t=m+n;
    cout<<"The result is "<<t<<endl;
}
```

程序运行结果为：

Please input two integers:

35 62 ✓

The result is 97

同第一个程序一样，例 1-2 也是从 main() 开始运行。在 C++ 程序中，每一个变量都需要在声明后才可以使⽤，所以我们首先在程序中定义将会用到的变量，int m,n,t; 表示分别定义了 m,n,t 三个整型（int）变量，int 是 C++ 语言提供的标准数据类型。定义变量时，需要在变量前声明变量的类型。

接下来，输出“Please input two integers: ”到显示器，执行语句 cin>>m>>n; 是从键盘接收用户输入的两个整型值，即 m 和 n。用户输入 35、一个空格和 62，敲回车键程序继续执行。

“✓”在本书中表示用户敲回车键。程序将两个数分别赋值给 m 和 n。

t=m+n; 是将 m 和 n 相加的值赋值给变量 t，得到的结果是 97。在接下来的语句 cout<<"The result is"<<t<<endl; 中，用连续的三个“<<”符号，将各项内容连续输出，“endl”表示输出一个换行符，与“\n”等价。

从上面两个例子可以看出,每一种编程语言都使用一组字符来构造编程语句。用 C++ 语言编写程序时,除了字符型数据以外,其他所有成分都只能由字符集中的字符构成。C++ 语言的字符集由下列字符构成:

26 个大写英文字母: ABCDEFGHIJKLMNOPQRSTUVWXYZ

26 个小写英文字母: abcdefghijklmnopqrstuvwxyz

10 个数字字符: 0 1 2 3 4 5 6 7 8 9

特殊字符: + - * / = , . _ : ; ? \ " ' ~ | !
% & () [] { } ^ < > 空格

词法记号是最小的词法单元,接下来介绍 C++ 的关键字、标识符、文字、运算符、分隔符、空白符。

1. 关键字

C++ 中,关键字又称保留字,它是预先定义好的标识符,这些标识符对 C++ 编译程序有着特殊的含义。下面列举出 C++ 语言中的关键字:

auto	bool	break	case
catch	char	class	const
const_cast	continue	default	delete
do	double	dynamic_cast	else
enum	explicit	extern	false
float	for	friend	goto
if	inline	int	long
mutable	namespace	new	operator
private	protected	public	register
reinterpret_cast	return	short	signed
sizeof	static	static_cast	struct
switch	template	this	throw
true	try	typedef	typeid
typename	union	unsigned	using
virtual	void	volatile	while

2. 标识符

标识符是程序员声明的单词,用它命名程序正文中的一些实体,如函数名、变量名、类名、对象名等。C++ 标识符的构成规则如下:

以大写英文字母、小写英文字母或下划线开始;可以由大写英文字母、小写英文字母、下划线或数字 0~9 组成;大写英文字母和小写英文字母代表不同的标识符;不能使用 C++ 关键字作为标识符。

3. 文字(常量)

文字是在程序中直接使用符号表示的数据,包括数字、字符、字符串和布尔文字。

4. 操作符(运算符)

操作符是用于实现各种运算的符号,例如: + - * / 。

5. 分隔符

分隔符用于分隔各个词法记号或程序正文,C++ 中使用的分隔符有: () { } , : ; 等,

这些分隔符不表示任何实际的操作，仅用于构造程序。

6. 空白符

在程序编译时的词法分析阶段将程序正文分解为词法记号和空白符。空白符是空格、制表符、换行符和注释的总称。

空白符用于指示词法记号的开始和结束，但除了这个功能以外，其余的空白符都将被忽略。因此，C++程序可以不必严格地按行书写，凡是可以出现空格的地方，都可以出现换行。例如，在声明一个变量的时候，以下几种情况都是等价的：

```
int age ;
与
int          age ;
或与
int
age
;
```

尽管以上三种情况达到的目的是一样的，我们还是要求大家在书写程序的时候，要尽量做到思路清晰、易于阅读。因为程序编写好之后，不仅要提供给计算机执行，也要给其他软件设计师阅读，以便于修改和维护。

1.4 Microsoft Visual Studio 2008 开发环境

Microsoft Visual Studio 2008 是微软公司推出的一个功能强大的可视化软件开发环境，其中包含 C++集成开发环境 Microsoft Visual C++。Microsoft Visual C++中提供了 C++编译器、程序编辑环境、调试器以及程序向导等组件，用它可以方便的编写各类 C++程序。要使用 Microsoft Visual C++编写 C++程序，首先要了解它的集成开发环境。本节主要介绍 Microsoft Visual C++集成开发环境的特点、界面风格和简单工具的使用。

安装完 Microsoft Visual Studio 2008 后，就可以使用该集成开发环境创建 C++项目，编写 C++程序了。下面通过例 1-1 说明如何利用 Microsoft Visual Studio 2008 创建一个 C++程序项目。

编程说明与实现：

(1) 启动 Microsoft Visual Studio 2008，打开“文件”菜单，然后打开“新建”子菜单，执行“项目”命令，出现“新建项目”对话框，如图 1-1 所示。在“项目类型”列表框中选中“其他语言”下的“Visual C++”选项，在“模板”列表框中选中“Win32 控制台应用程序”。在“名称”文本框中输入项目名称“Hello”，在“位置”文本框中输入保存项目的路径。单击“确定”按钮，进入“Win32 应用程序向导”的第一步界面。

(2) 在“Win32 应用程序向导”的第一步，显示了当前项目的基本概况，如图 1-2 所示。然后单击“下一步”按钮进入“Win32 应用程序向导”的第二步界面。

(3) 在“Win32 应用程序向导”的第二步，可以对当前即将建立的项目做一些基本配置，如图 1-3 所示。在“附加选项”部分中，需要勾选“空项目”复选框，然后单击“完成”按钮，完成 C++项目的创建。项目创建完成后，会自动进入该项目管理界面，如图 1-4 所示。

(4) 在项目管理界面中向项目 Hello 中添加源程序文件。在项目管理界面中的“解决方案资源管理器”中找到我们创建的解决方案“Hello”，然后在其下的“源文件”图标上单击鼠

标右键,在弹出菜单中选择“添加”菜单项,并执行其下的“新建项”命令,此时会弹出如图1-5所示的“添加新项”对话框。在该对话框中的“模板”列表框中,选中“C++文件(.cpp)”,并在“名称”文本框中输入“Hello”,然后再单击“添加”按钮,此时会在当前项目中添加一个名为“Hello.cpp”的C++源代码文件,并自动打开该文件的编辑窗口,最后在编辑窗口中输入例1-1的源程序代码,如图1-6所示。



图 1-1 创建 Win32 控制台应用程序

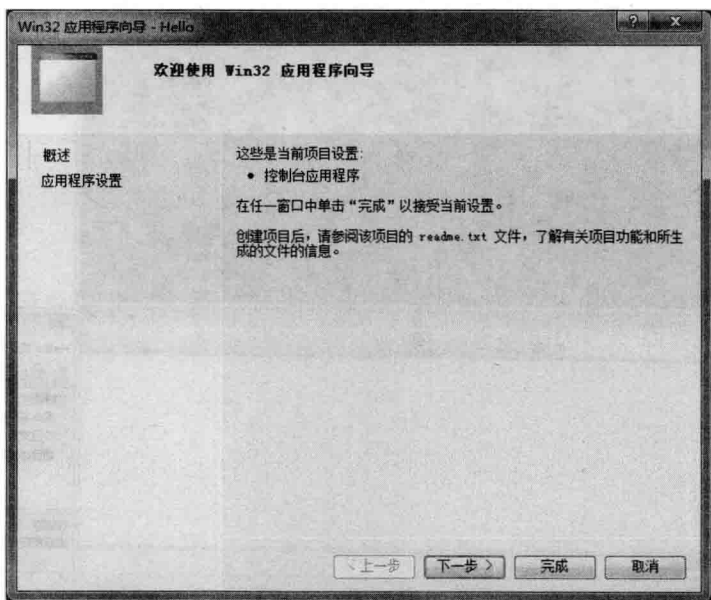


图 1-2 “Win32 应用程序向导”第一步

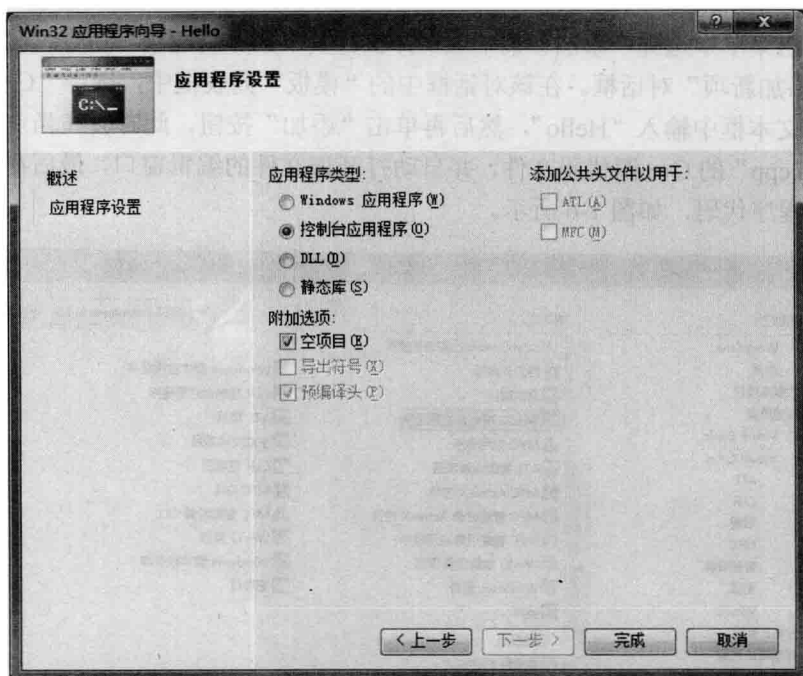


图 1-3 “Win32 应用程序向导” 第二步

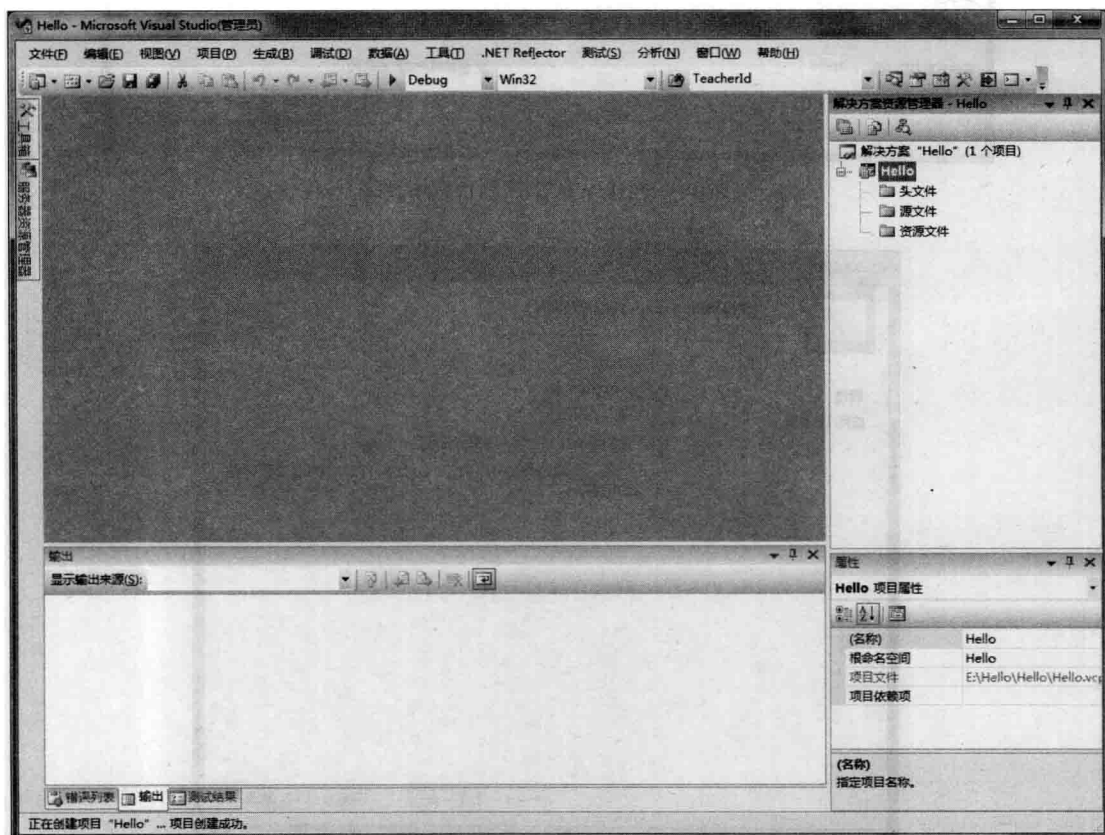


图 1-4 项目管理界面

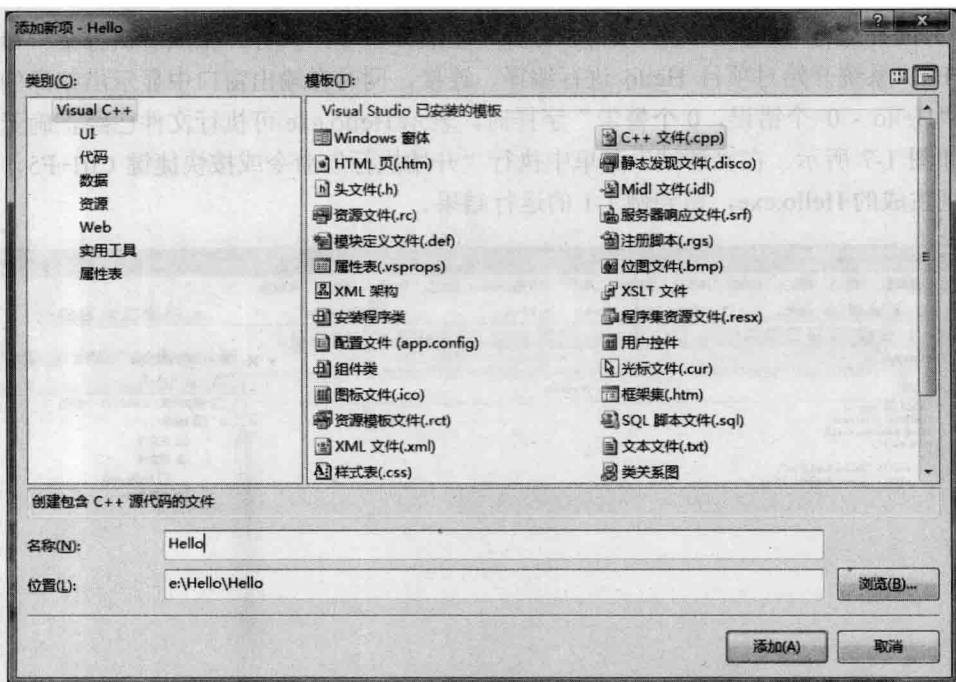


图 1-5 “添加新项”对话框

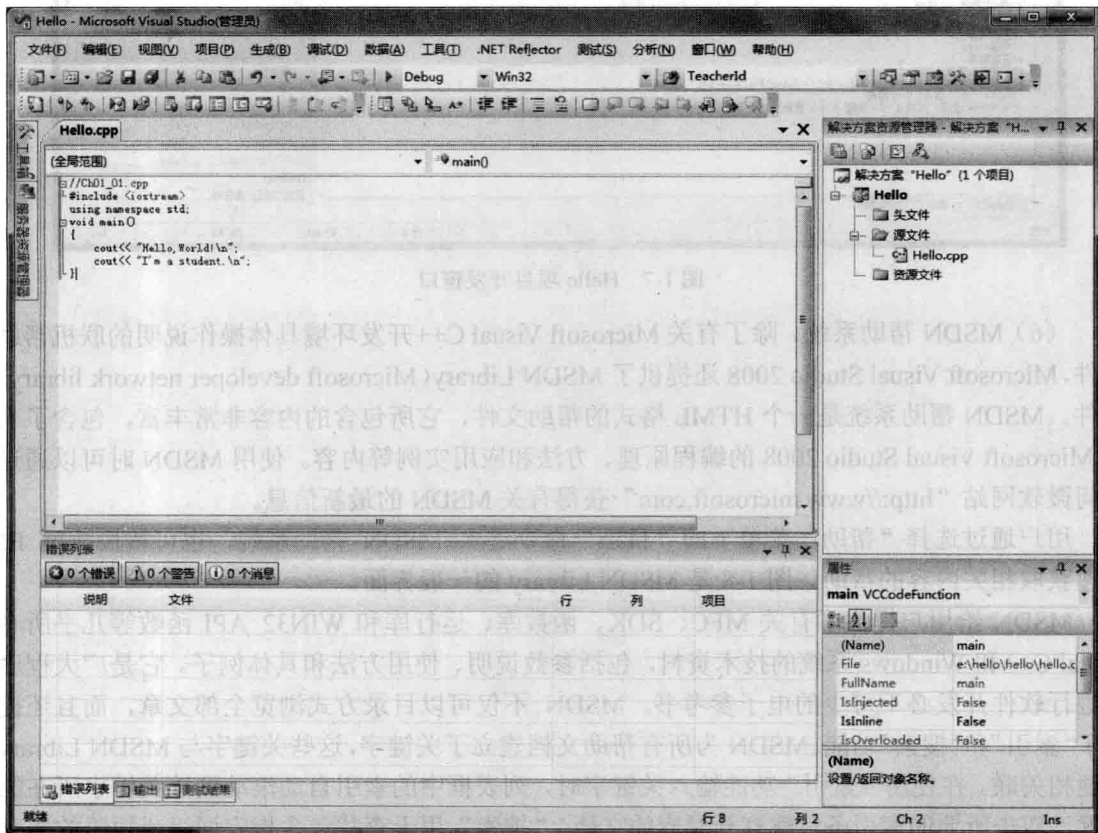


图 1-6 源文件编辑窗口

(5) 编译运行“Hello”C++项目。打开“生成”菜单,执行“生成解决方案”命令或按快捷键 F6,系统开始对项目 Hello 进行编译、连接,同时在输出窗口中显示出编译的内容,当出现“Hello - 0 个错误,0 个警告”字样时,表示 Hello.exe 可执行文件已经正确无误地生成了,如图 1-7 所示。在“调试”菜单中执行“开始执行”命令或按快捷键 Ctrl+F5,就可以运行刚刚生成的 Hello.exe,得到例 1-1 的运行结果。

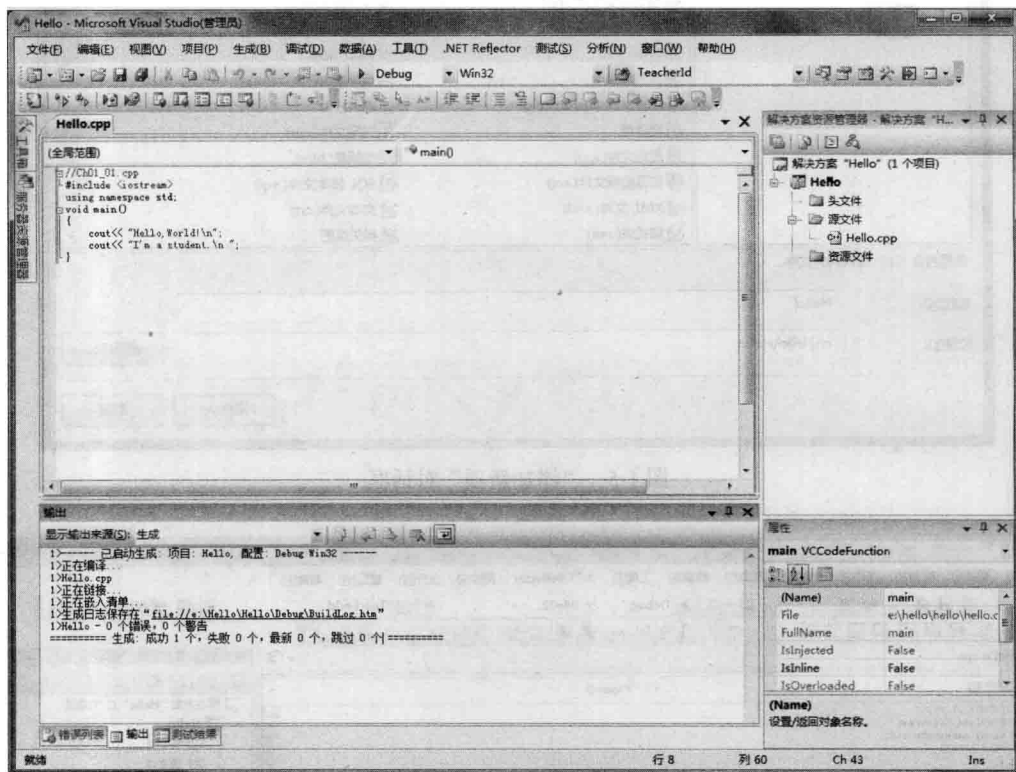


图 1-7 Hello 项目开发窗口

(6) MSDN 帮助系统。除了有关 Microsoft Visual C++开发环境具体操作说明的联机帮助文件,Microsoft Visual Studio 2008 还提供了 MSDN Library(Microsoft developer network library)组件。MSDN 帮助系统是一个 HTML 格式的帮助用户,它所包含的内容非常丰富,包含了有关 Microsoft Visual Studio 2008 的编程原理、方法和应用实例等内容。使用 MSDN 时可以通过访问微软网站“<http://www.microsoft.com>”获得有关 MSDN 的最新信息。

用户通过选择“帮助”菜单下的“目录”命令进入 MSDN 帮助系统,也可按快捷键 F1 快速获取相关内容的帮助。图 1-8 是 MSDN Library 的一般界面。

MSDN 给用户提供了有关 MFC、SDK、函数库、运行库和 WIN32 API 函数等几乎所有 Visual C++和 Windows 系统的技术资料,包括参数说明、使用方法和具体例子。它是广大程序员进行软件开发必不可少的电子参考书。MSDN 不仅可以目录方式浏览全部文章,而且还提供了“索引”和“搜索”功能。MSDN 为所有帮助文档建立了关键字,这些关键字与 MSDN Library 主题相关联。在使用“索引”功能输入关键字时,列表框中的索引自动滚动到该关键字所在的位置,双击所要的索引条目就打开对应的文档。“搜索”用于查找包含指定词组或短语的所有文档。