

21世纪高等学校规划教材 | 计算机科学与技术



面向对象程序设计 (C++)实践教程

李延军 黄国勇 张娟 编著



清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

面向对象程序设计 (C++)实践教程

李延军 黄国勇 张娟 编著

清华大学出版社

北京

内 容 简 介

本书强调 C++ 程序设计的基本方法,在内容安排上循序渐进,突出重点。从 C 到 C++ 语言的基本数据类型与基本控制结构入手,逐步过渡到函数、类与对象、继承、模板、多态等复杂的 C++ 机制,使读者准确找到编程技术要领,能在较短的时间内掌握面向对象程序设计(OOP)的基本概念和技术方法。全书采用引入知识点、用例验证知识点、变化实例综合知识点的模式,由浅入深地展开对技术内容的讲述,将前后知识串联力求做到融会贯通。最后以五子棋游戏和基于 MFC 的图像处理 DIY 作为综合实例,在快乐学习的过程中帮助读者理解 OOP 的重要概念,提高设计、编写、调试程序的能力。

本书结构紧凑,内容丰富,可供高校信息类专业学生使用,也可作为 C++ 语言的培训教材和各类自学读者学习参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

面向对象程序设计(C++)实践教程/李延军等编著. —北京:清华大学出版社,2011.10
(21 世纪高等学校规划教材·计算机科学与技术)

ISBN 978-7-302-26522-1

I. ①面… II. ①李… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 173164 号

责任编辑:高买花

责任校对:胡伟民

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010 62770175

邮 购:010 62786544

投稿与读者服务:010-62795954,jsjtc@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:11.5 字 数:279 千字

版 次:2011 年 10 月第 1 版 印 次:2011 年 10 月第 1 次印刷

印 数:1~3000

定 价:20.00 元

产品编号:041565-01

随着计算机软件技术的发展,计算机软件应用范围越来越广,复杂程度越来越高,传统的结构化程序设计(SP)思想已越来越不能适应软件开发工作的需要。面向对象技术与传统的结构化程序设计思想相比有许多优点,因为采用面向对象的观点看待所要解决的问题并将其抽象为系统极其且更符合人类的思维习惯。现在书店里堆砌着大量的 C++ 或是面向对象程序设计(OOP)方面的书籍,大体分为两类:翻译教材和自编教材。翻译教材内容上较新,举例有特色,但国外名家娓娓道来的撰写风格往往使篇幅过长,不适合在有限的课堂教学中使用;自编教材逻辑性强、内容紧凑,但存在以下 4 个方面的不足:一是概念阐述不清,会对学生产生误导,如虚函数是运行时多态的基础,大部分教材中的解释是迟后联编(动态绑定),也有教材表述为“派生类中改写虚函数时,继承来的虚表所指函数会被覆盖”;二是对 OOP 的精华(封装性、继承性、多态性)部分的阐释有些片面,而面面俱到的分析却不够深入,浮于表面看问题或是能深入说明却又用词晦涩,很难形象地去理解;三是对基类的私有成员在派生类中的访问控制有多种提法,如“不能继承”、“能继承却不能访问”、“继承其位置”,这些不统一、不明确的提法要靠授课老师的理解予以解释说明;最后,不能居高临下地把握类、继承多态之间的脉络,对引用、友元、重载等就现象说现象,割裂了知识体系的完整性。

教育部计算机科学与技术教学指导委员会建议“加强学生实践和动手能力的培养”,本教材编写的初衷是在实施“厚基础、宽口径加精专业”的课程体系的过程中鼓励学生进行理论验证和动手实践,从而奠定学生扎实的专业基础。

学习程序设计是没有秘诀的,“编程是硬道理”。只有让学生真正动手编,上机调试,在成功中增强信心,在失败中积累经验才是正确的教学方法。老师(或者教材)的作用是像庖丁解牛一样从整体上给学生以清晰的知识结构体系,培育其学习的兴趣,以应用为中心,以初学者为对象,以提高编程能力带动知识学习为目标,本教材具有以下特点:

1. 强化重要概念

类是面向对象程序设计的首要概念,对封装性的理解会影响整个课程的学习;继承是面向对象程序设计的关键;多态是指一个名字可以有多种语义,是 OOP 的重要特征之一,“若语言不支持多态,则不能称其是面向对象的”,由此可见,多态之于 C++ 就像指针之于 C 一样;模板是 C++ 语言相对较新的一个重要特性,它使程序员能够快速建立具有类型安全的类库和函数集合;友元虽然“破坏”了类的封装性,却是提高效率和方便编程必不可少的技术手段,也是编译时多态性的体现。

2. 不求面面俱到,则求突出重点

C++ 完全兼容 C,不但支持 SP 设计,而且满足 OOP 设计思想。鉴于 C 语言程序设计是很多高校理工科的必修平台课,因此,没有必要将 SP 的内容再详细描述。教材应重点放在

理解类和对象的概念上,掌握继承机制,多态性的实现方式等内容,基础、重点和难点三者之间环环相扣。

3. 通俗易懂

面向对象技术有许多抽象的语法和较难理解的术语,教材编写在科学性、严谨性的基础上尽量做到简明扼要,通俗易懂。如在分析继承性和多态性时,尽量用图的方式展示程序运行变化过程,将仅能意会的流程深入浅出地表达出来,使读者准确找到编程的技术要领,洞穿 C++ 内部实现要害,只有这样才能让读者居高临下地把握知识体系,而不仅仅是记住几个概念。

4. 良好的编程风格的培养

很多培训机构采用机房教学,白板推演,逐个知识点验证的实战式方式。作为高校人才培养当然不能照搬这种培训模式,但其注重上机实践、重视培养良好编程风格的方法是值得借鉴的。

本书所有参编人员都是长期在高校从事专业教学与科研的一线教师,具有丰富的编程与教学经验,了解在学习过程中希望掌握的知识和容易出错的地方并在书中相应较详细的地方予以讲解。本书由李延军负责编写,具体分工为:第 1~3 章由张娟编写,第 4~5 章由黄国勇编写,第 6~9 章由李延军编写并负责各章所有实例及习题上机设计并调试,第 10~11 章及附录由李伟编写。全书由李延军老师统稿。

本教材在编写过程中得到了众多专家学者、同行、同事的帮助,特别感谢昆明理工大学的张力副教授、胡建华副教授对本教材编写提供的指导,感谢云南大学的代飞对教材内容选取和章节安排提出的建设性意见。感谢吴汉章,王晨纱、李云辉等在教学实践和教材编写中提供的教学反馈和改进意见,同样感谢读者选择使用本书,由于时间、经验及水平的原因,书中难免有疏漏和不足之处,敬请专家、同行和读者朋友批评指正,谢谢。

编 者

2010-9-16

第 1 章 概述	1
1.1 程序设计概述	1
1.1.1 计算机语言和程序	1
1.1.2 结构化程序设计	2
1.1.3 面向对象程序设计	2
1.2 C++ 语言简介	3
1.2.1 C++ 的起源和特点	3
1.2.2 开发过程	4
1.2.3 最简单的程序	4
1.3 C++ 程序上机步骤	5
1.3.1 Visual C++ 6.0 上机指南	5
1.3.2 程序调试	7
本章小结	8
习题与上机	8
第 2 章 从 C 到 C++	10
2.1 C++ 的数据类型与输入输出	10
2.1.1 C++ 的数据类型	10
2.1.2 输入输出	10
2.2 表达式和语句	11
2.2.1 运算符与表达式	11
2.2.2 流程控制语句	12
2.3 数组	13
2.4 指针	15
2.4.1 指针与数组	15
2.4.2 指针与函数	16
2.5 内存的动态分配和释放	17
本章小结	18
习题与上机	19



第3章 函数	21
3.1 函数概述	21
3.1.1 函数的定义与声明	21
3.1.2 函数原型	22
3.2 函数的参数传递	23
3.3 内联函数和递归函数	25
3.4 具有默认参数的函数	27
3.5 函数重载	27
3.6 作用域和可见性	28
本章小结	30
习题与上机	30
第4章 类和对象	34
4.1 类	34
4.1.1 结构与类	34
4.1.2 类的定义	35
4.1.3 类的成员函数	35
4.1.4 类成员的访问控制	36
4.1.5 类的作用域	37
4.2 对象	38
4.2.1 对象的创建和撤销	38
4.2.2 构造函数	39
4.2.3 复制构造函数	42
4.2.4 析构函数	43
4.3 this 指针	44
本章小结	46
习题与上机	46
第5章 类的其他特性	51
5.1 静态成员	51
5.1.1 静态数据成员	51
5.1.2 静态成员函数	52
5.2 友元	53
5.2.1 友元函数	53
5.2.2 友元类	54
5.3 常类型	56
5.4 类成员指针	58
5.5 类的组合	59

本章小结	61
习题与上机	61
第 6 章 类的继承和派生	65
6.1 继承的概念	65
6.2 派生类的继承方式	67
6.2.1 基类成员在派生类中的访问属性	67
6.2.2 派生类对基类成员的访问规则	68
6.3 派生类的构造和析构	72
6.4 调整基类成员在派生类中的访问属性	74
6.4.1 同名函数	74
6.4.2 访问声明	75
6.5 多重继承	78
6.5.1 多重继承的声明	78
6.5.2 多重继承的构造和析构	80
6.5.3 虚基类	80
6.6 赋值兼容性规则	85
本章小结	86
习题与上机	87
第 7 章 多态性和虚函数	96
7.1 多态性的基本概念	96
7.2 运算符重载	97
7.3 虚函数	101
7.3.1 虚函数的定义和使用	101
7.3.2 为什么虚函数可以做到动态绑定	102
7.3.3 虚函数的限制	106
7.4 纯虚函数和抽象类	107
7.4.1 纯虚函数	107
7.4.2 抽象类	108
本章小结	109
习题与上机	109
第 8 章 模板	114
8.1 模板的概念	114
8.2 函数模板与模板函数	115
8.2.1 函数模板声明和模板函数使用	115
8.2.2 重载函数模板	116
8.3 类模板和模板类	118

8.3.1 类模板的定义	118
8.3.2 使用类模板	119
本章小结	120
习题与上机	120
第9章 I/O 流	123
9.1 C++的流及流类库	123
9.1.1 C++的流	123
9.1.2 流类库	124
9.2 预定义类型的输入输出	125
9.2.1 无格式输入输出	125
9.2.2 输入输出的格式控制	126
9.3 文件的输入输出	129
9.3.1 文件的打开与关闭	129
9.3.2 文件的读写	130
本章小结	131
习题与上机	131
第10章 异常处理	134
10.1 异常的概念	134
10.2 异常处理的实现	135
10.3 异常处理的规则	136
10.4 基于对象的异常处理	138
10.5 异常的重抛出	141
本章小结	141
习题与上机	142
第11章 综合案例	145
11.1 五子棋游戏	145
11.1.1 功能描述	145
11.1.2 类的设计	145
11.1.3 数据结构设计	147
11.1.4 程序实现	148
11.2 图像处理DIY	156
11.2.1 预备知识	156
11.2.2 打开BMP格式图像文件的步骤	157
11.2.3 数字图像点处理	159
11.2.4 数字图像区域处理	162
本章小结	166

附录 A C++ 语言中的关键字(保留字)	168
附录 B 运算符优先级表	169
附录 C 基本 ASCII 码字符	170
附录 D Windows 位图结构数据域的含义	171
参考文献	172

概 述

内容提要

语言是人类创造的工具,它用来表达意思,交流思想。程序设计(编程)语言是人类与计算机交流的工具。C++是从C语言发展演变而来的一种新型的、以面向对象为特征的程序设计语言。本章从程序设计语言的发展历程出发,介绍C++语言的产生和发展;C++语言的特点及C++程序的开发过程。通过一个简单实例的详细讲述,引领读者逐步进入C++的精彩世界。

知识要点

- 不同类型的编程语言
- 面向对象程序设计的基本概念
- C++语言的特点
- C++程序开发的过程
- 源程序、头文件、可执行文件的概念

教学建议

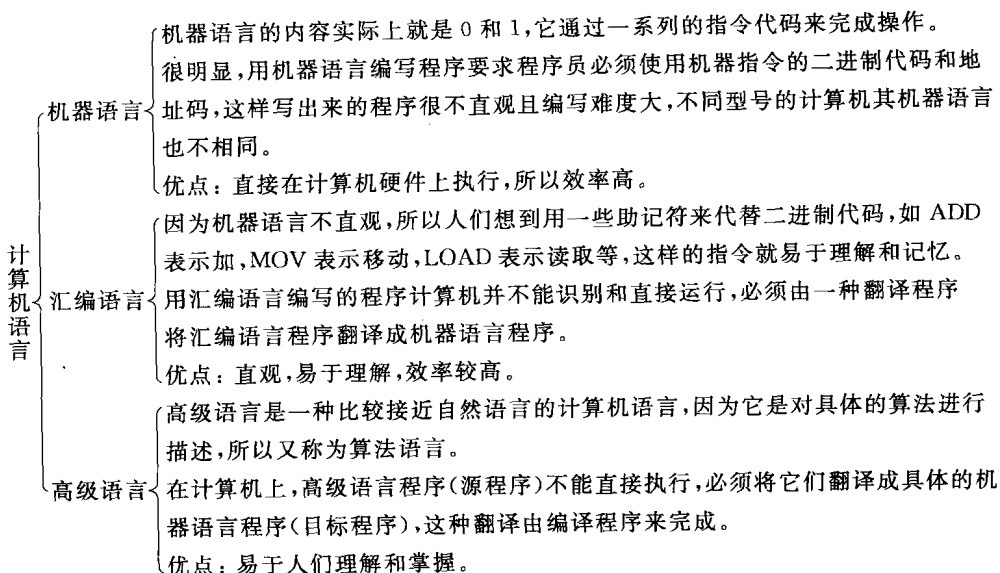
了解C++的来龙去脉和它的功能是十分必要的,同时还要知道编程是怎么回事。建议安排3学时。本章的教学基本要求是:了解不同类型的编程语言;了解面向对象程序设计语言中的基本概念;熟悉C++程序开发的过程;了解程序结构,理解源程序、头文件、可执行文件的概念;能够编写一个简单的程序,输入数据,输出结果。

1.1 程序设计概述

1.1.1 计算机语言和程序

1. 计算机语言

计算机语言是指程序设计人员和计算机都可以识别的程序代码规则,它是进行程序设计的工具,又称为程序设计语言。计算机语言一般可分为三类:



2. 程序

“程序”有两种含义:一是由程序员编写的源代码,其二是可执行的软件。任何语言编制的程序,最后一定都需要语言处理程序转换成机器语言程序才能被计算机执行。语言处理程序按其处理方式不同,可分为解释型程序和编译型程序两大类。现在人们非常重视的是易维护的代码。

1.1.2 结构化程序设计

以前,人们把程序看成是处理数据的一系列过程。过程或函数定义为一个接一个顺序执行的一组指令。数据与程序分开存储,编程的主要技巧在于追踪函数如何调用,哪些数据发生了变化。为了解决其中可能存在的问题,结构化编程应运而生。当一些任务十分复杂以至无法描述时,可以将它拆分为一系列较小的功能部件,直到这些自完备的子任务小到成为易于理解的程序。结构化程序设计的主要思想是功能分解并逐步求精。

结构化程序设计成功地处理复杂问题提供了有力的手段。然而,随着时间的推移,它的一些缺点也越来越突出。当数据量增大时,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。首先,对数据处理能力的需求越强,这种分离所造成的负面影响越显著;其次,结构化程序设计相对于老问题的新方法都要带来额外的开销,即重用性较差。对于软件工程师们来说,在面向对象程序设计出现之前,一直缺乏具备这种能力的工具。

1.1.3 面向对象程序设计

面向对象程序设计(Object_Oriented Programming, OOP)的本质是把数据和处理数据的过程当成一个整体——对象。所有的 OOP 语言(包括 C++)都具有三个共同的特性,即封装性、多态性和继承性。

1. 封装性(Encapsulation)

封装是一种机制,它将代码和数据连接起来,并使之安全,以防外界的干扰和误用。代码和数据用这种方式连接时一个对象(object)就产生了。换言之,对象就是支持封装的机

制。对象中的代码和数据可以是私有的(private),也可以是公有的(public),一个对象的公有部分为该项的私有部分提供了一个可以控制的接口。

2. 继承性(Inheritance)

继承是面向对象程序设计的重要特征。通过继承,一个对象可以获得另一个对象的属性,并往其中加入属于自己的属性。继承在 OOP 中扮演着十分重要的角色。

3. 多态性(Polymorphism)

所谓多态性就是一个接口,多种方式。多态性的优点在于通过提供一个相同的接口,可由通用类不同的动作来访问,从而降低了问题的复杂度。

1.2 C++语言简介

1.2.1 C++的起源和特点

1. C++的起源

C++语言是在 C 语言的基础上为支持面向对象的程序设计而研制的一个通用目的的程序设计语言,它的开发者是 AT&T 贝尔实验室的 Bjarne Stroustrup 博士。C 语言的前身是 B 语言,而 B 语言又是在继承和发展了 BCPL 基础上设计的。1972 年,Dennis Richie 在 AT&T 贝尔实验室设计了 C 语言。为了满足管理程序复杂性的要求,1980 年,贝尔实验室的 Bjarne Stroustrup 开始对 C 语言进行改进和扩充,最初的成果称为“C with class”。1983 年正式取名为 C++。在经历了三次修订后,于 1994 年制定了 ANSI C++ 标准草案。C++ 的发展如图 1-1 所示。

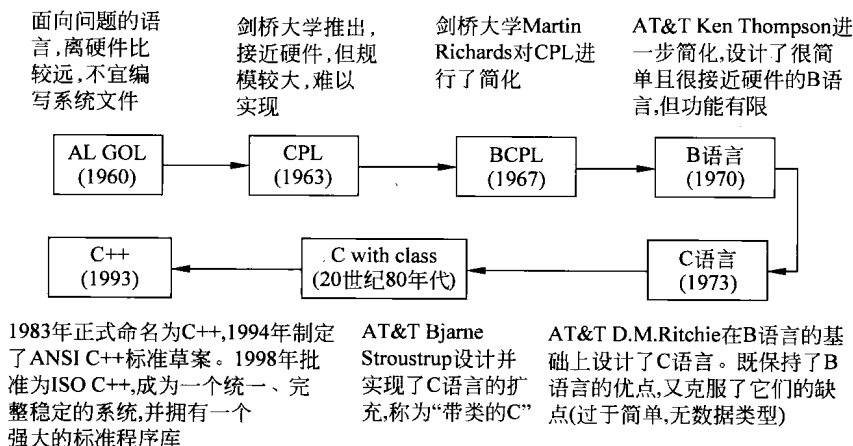


图 1-1 C++的发展

C++ 提出了把数据和在数据之上的操作封装在一起的类、对象和方法的机制,并通过派生、继承、重载和多态等特征,实现了人们期待已久的软件复用和程序的自动生成。

2. C++的特点

C++ 现在得到了越来越广泛的应用,它既继承了 C 语言的优点,又有自己独到的特点,最主要的有:

(1) C++保持与C兼容,这就使许多C代码不经修改就可以为C++所用,用C编写的众多的库函数和实用软件可以用于C++中。

(2) 用C++编写的程序可读性好,代码结构更为合理,可直接在程序中映射问题空间结构。

(3) 生成代码的质量高,运行效率仅比汇编语言代码段慢10%~20%。

(4) 从开发时间、费用到形成的软件的可重用性、可扩展性、可维护性和可靠性等方面有了很大的提高,使得大中型程序开发变得更加容易。

(5) 支持面向对象的机制,可方便地构造出模拟现实问题的实体和操作。

总之,目前人们对C++的兴趣越来越浓,它已经成为被广泛使用的通用程序设计语言。在国内外使用和研究C++的人数正迅猛增加,优秀的C++版本和配套工具软件也不断涌现。

1.2.2 开发过程

大多数现代编译程序都提供了一个集成开发环境。在这样的环境中,一般是从菜单中选定 compile 或 make 或 build 命令来生成可执行的计算机程序。程序员编制的源程序被编译(compile)后,会生成一个目标文件,通常以“.obj”作为文件后缀。该目标文件仍不是一个可执行的程序,因为目标代码只是一个个程序块,需要相互衔接成为一个适应一定的操作系统环境的程序整体。为了把它转换为可执行程序,必须进行连接(link)。

C++程序通常是通过同时连接一个或几个目标文件与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库两类。文件与库连接,生成计算机可执行的程序。开发C++程序的步骤如图1-2所示。

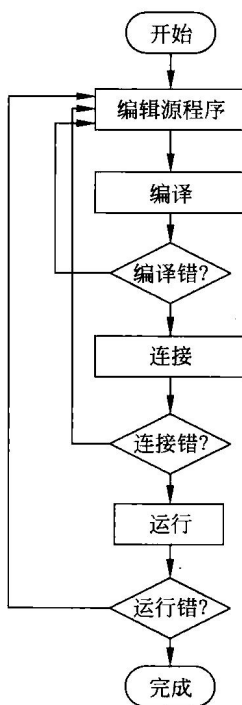
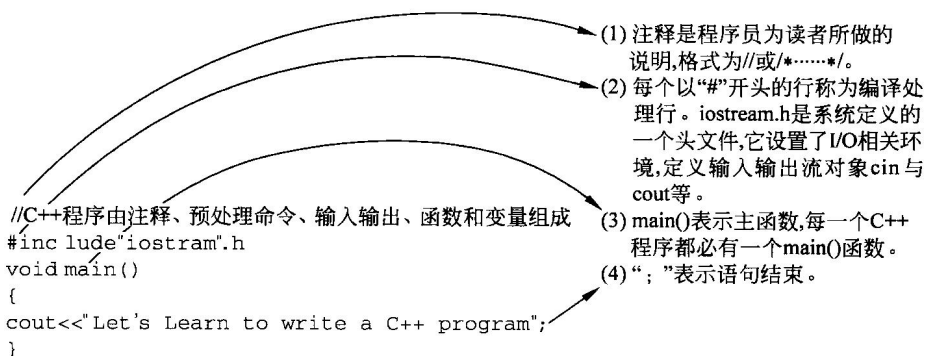


图 1-2 开发C++程序的步骤

1.2.3 最简单的程序

下面从一个最简单的程序来分析C++的程序构成。



程序输出: Let's learn to write a C++ program

1.3 C++ 程序上机步骤

一个 C++ 程序编写完成后,就可以进入输入、编译连接与运行的过程。C++ 程序的上机过程一般分为以下几个步骤:

(1) 输入 C++ 程序,建立 C++ 程序文件,扩展名为 .cpp。

(2) 对 C++ 程序文件进行编译与连接,生成目标文件与可执行文件(扩展名为 .exe)。如果在这一步发现有错误,则要对源程序进行编辑修改,再进行编译与连接,直到在编译连接过程中没有错误发生为止。

(3) 运行可执行文件得到结果。如果在运行过程中发现有错误,则要对源程序进行编辑修改,再进行编译、连接与运行,直到没有错误发生为止。

下面以 Visual C++ 6.0 为例进行介绍。

1.3.1 Visual C++ 6.0 上机指南

Visual C++ 6.0 是 Microsoft 公司开发的基于 Windows 平台的 C/C++ 可视化集成开发工具,可以在其中编辑、编译、连接、运行、调试 C/C++ 程序。

1. 启动 VC++ 6.0

单击“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0;也可以在桌面上创建 VC++ 6.0 的快捷方式,双击该图标也可启动 VC++ 6.0,启动后可看到 VC++ 6.0 的主窗口,如图 1-3 所示。

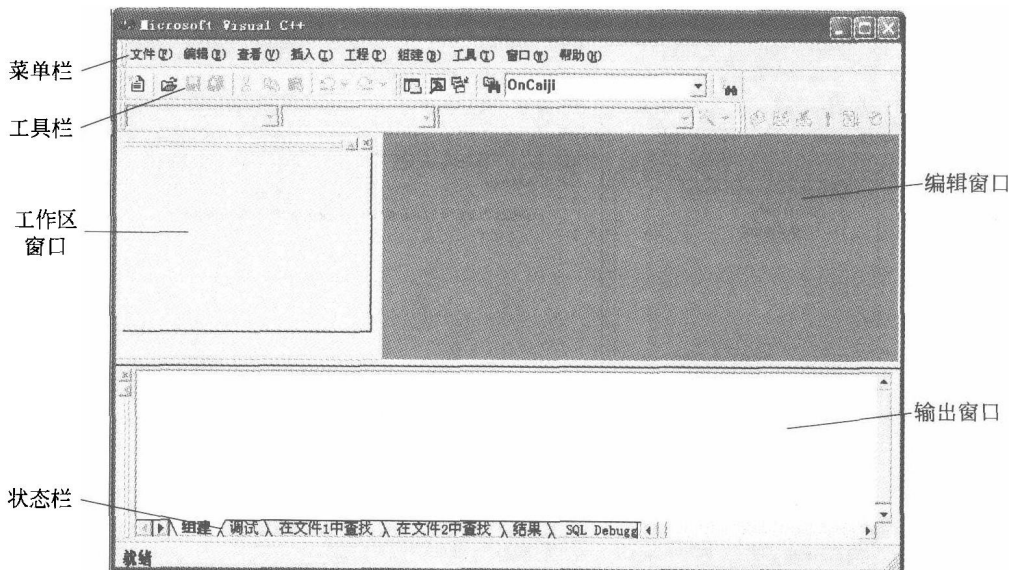


图 1-3 VC++ 6.0 的主窗口

2. 新建 C++ 源程序

首先在本地硬盘上新建一个工作目录“E: \work”。简单的 C++ 程序只包含一个源文件。选择“文件”→“新建”命令,单击如图 1-4 所示的“文件”选项卡,选中 C++ Source File。

然后在右边“文件名”文本框中输入文件名,例如 Hello world.cpp(.cpp 后缀可省略);在“位置”文本框中输入文件存放位置,或单击右边的“...”按钮,进行位置选择,例如“E: \WORK\Hello world”。最后单击“确定”按钮,打开一个编辑窗口,如图 1-4 所示。

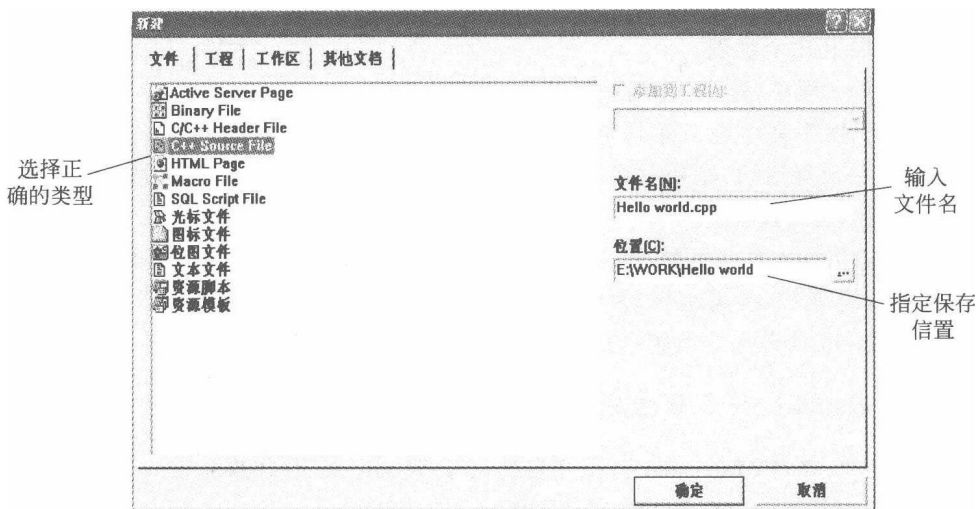


图 1-4 “新建”对话框

3. 编辑文件

在文件编辑窗口中输入程序代码后,选择“文件”菜单中的“保存”命令或单击工具栏上的“保存”按钮进行保存,如图 1-5 所示。

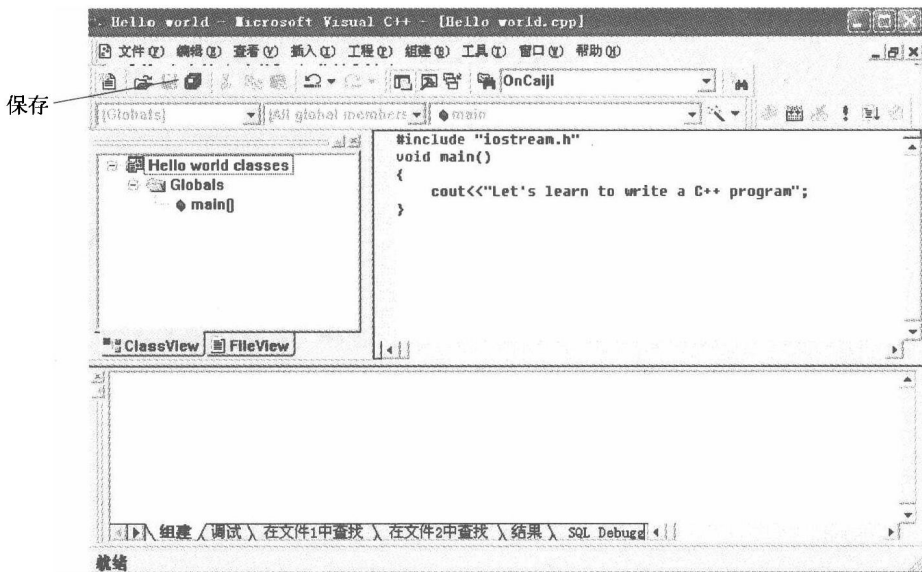


图 1-5 编辑文件

4. 编译程序

如图 1-5 所示,打开“组建”菜单,选择“编译”菜单项或“组建”菜单项对 Hello world.

cpp 进行编译。或单击编译工具条的“编译”按钮或“组建”按钮。编译工具条的各工具功能如图 1-6 所示。

注意：组建=编译+连接

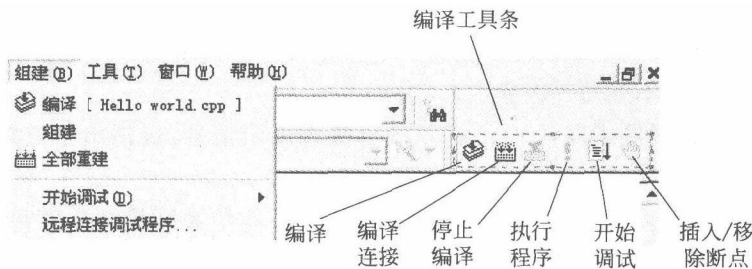


图 1-6 编译程序

单击“编译”按钮后,VC++ 6.0 会立即弹出一个询问对话框,如图 1-7 所示,意思是“需要有一个活动项目的工作区才可以执行编译命令,是否要创建一个默认的项目工作区?”单击“是”按钮。

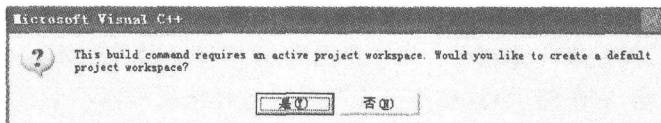


图 1-7 创建默认的项目工作区询问对话框

5. 运行程序

选择“组建”菜单下的“执行”菜单项,或单击编译工具条上的“!”按钮即可运行当前程序。当在输出窗口中出现“0 error(s), 0 warning(s)”的提示信息后,VC++ 6.0 将打开一个控制台窗口,在其中运行该项程序,如图 1-8 所示,按任意键将关闭该窗口。

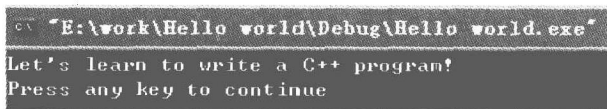


图 1-8 运行程序

6. 调试程序

7. 关闭工作区

选择“文件”菜单中的“关闭工作区”命令关闭当前工作区,此时会关闭工作区中所有已打开的文件。

1.3.2 程序调试

除了较简单的情况,一般程序都很难一次就正确。在上机过程中,根据出错现象找出错误并将改正的过程称为程序调试。在学习程序设计的过程中,逐步培养调试程序的能力是非常重要的,要靠实践积累经验,并不断总结。程序中的错误大致分为三类: