



普通高等教育“十二五”应用型本科规划教材

面向对象程序设计实践教学教程

中国高等教育学会 组织编写

主编 李延军

MIANXIANG DUIXIANG CHENGXU
SHEJI SHIJIAN JIAOCHENG



中国人民大学出版社



普通高等教育“十二五”应用型本科规划教材

面向对象程序设计实践教程

中国高等教育学会 组织编写

主编 李延军

中国人民大学出版社

• 北京 •

图书在版编目 (CIP) 数据

面向对象程序设计实践教程/李延军主编. —北京: 中国人民大学出版社, 2015. 1
普通高等教育“十二五”应用型本科规划教材
ISBN 978-7-300-20752-0

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 (2015) 第 024067 号

普通高等教育“十二五”应用型本科规划教材

面向对象程序设计实践教程

中国高等教育学会 组织编写

主编 李延军

Mianxiang Duixiang Chengxu Sheji Shijian Jiaocheng

出版发行 中国人民大学出版社

社 址 北京中关村大街 31 号

邮政编码 100080

电 话 010-62511242 (总编室)

010-62511770 (出版部)

010-82501766 (邮购部)

010-62514148 (门市部)

010-62515195 (发行公司)

010-62515275 (盗版举报)

网 址 <http://www.crup.com.cn>

<http://www.ttrnet.com> (人大教研网)

经 销 新华书店

印 刷 北京密兴印刷有限公司

规 格 185 mm×260 mm 16 开本

版 次 2015 年 3 月第 1 版

印 张 15.5

印 次 2015 年 3 月第 1 次印刷

字 数 356 000

定 价 32.00 元

版权所有 侵权必究 印装差错 负责调换

前言

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, C++ 和 OOP 是同一门课程吗? C++ 和 OOP 是什么关系? 这颇让读者感到困惑。首先, C++ 是面向对象的编程语言, 面向对象程序设计是设计方法。程序设计技术有四个层次, 从高到低分别是算法、方法学、语言和环境工具。语言是最基本、最起码的要求; 而掌握了程序设计方法学可以指导我们写出结构良好、易于维护的程序; 算法理论居于最高层次, 它着眼于高水平甚至有突破性的程序; 而环境工具则能使程序员快速高效地完成设计工作。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与之相比有许多优点, 更符合人类的思维习惯。就像我们都会唱的歌 “love you more than I can say”, 心里的想法不说别人又怎么会知道呢? 同理, 设计方法也要靠语言载体实现。面向对象程序设计语言很多, 如我们所熟知的 Smalltalk、Visual Basic、C++、Java 等。一般 Java 会单独设课, 于是 C++ 语言就成为最讨人喜欢的 OOP 教学工具了。

现在市面上存在的 C++ 或是 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, 不适合在时间有限的课堂教学上使用。自编教材逻辑性强、内容紧凑, 但存在以下四个方面的不足: 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, 大部分的教材的解释是迟后联编 (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, 分析不够深入, 浮于表面看问题, 或是能深入说明却用词晦涩, 使学生很难形象地去理解; 三是概念不统一、不明确, 对基类的私有成员在派生类中的访问控制就有多种提法, “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; 四是不能居高临下地把握类、继承等多态之间的脉络, 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、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，才是正确的学习方法。老师（或者教材）的作用是从整体上给学生以清晰的知识结构体系，培养学生的学习兴趣，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。

本教材的特点如下：

（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对整个课程的学习；继承是面向对象程序设计的关键；多态是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，则不能称为面向对象”，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 and 方便编程不可少的技术手段，也是编译时多态性的体现。

（2）突出重点，不求面面俱到。C++ 完全兼容 C，不但支持 SP 设计，而且能满足 OOP 的设计思想。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。教材应重点放在类和对象的概念上，理解继承是 OOP 中最重要机制，多态性的实现方式等方面，基础、重点和难点三者之间环环相扣。

（3）通俗易懂，兼具理论深度和实践强度。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机理时，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。此外，考虑将具有知识性、趣味性、综合性的课程设计内容选编进教材中（如五子棋游戏，图像处理 DIY 等），以期在理解基本概念、掌握各知识点上机验证的基础上，有一个综合性的理论联系实际的应用实例。

（4）由课堂讲授为主变为上机实践为主，期末考试为主变为过程考核为主。考虑将习题分为两个层次：验证性上机和设计性上机。要求读者独立完成并将实践过程计入技能成绩，力求提高学生的动手能力和独立思考的能力。

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。对在学习过程中须掌握的知识和容易出错的地方在书中进行了较详细的讲解。本书由李延军主编，第一、二、五章由晋帆编写，第三、四、十一章及附录由钟嘉玉编写，第六、七、八、九、十章由李延军编写。另外，李延军还负责了各章所有示例、习题与上机的设计和调试，以及全书的统稿工作。

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，特别感谢山东中医药大学曹慧教授、昆明理工大学的胡建华副教授对教材编写提供的指导，感谢张娟副教授对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、罗芳等同学在教学实践中提供的改进意见，同样感谢读者选择使用本书。由于作者时间、经验及水平有限，书中难免有错误、疏漏和不足之处，敬请专家、同行和读者朋友批评指正，谢谢！

李延军

2014 年 8 月

目录

第一章 程序设计与 C++ 基础	1
1.1 程序设计概述	1
1.2 C++ 语言简介	3
1.3 C++ 程序上机步骤	5
本章小结	8
习题与上机	9
第二章 从 C 到 C++	10
2.1 C++ 数据类型与输入输出	10
2.2 表达式和语句	11
2.3 数组	15
2.4 指针	18
2.5 内存的动态分配和释放	22
本章小结	23
习题与上机	24
第三章 函数	27
3.1 函数的定义与调用	27
3.2 函数的参数传递	29
3.3 内联函数和递归函数	32

3.4 具有默认参数的函数	34
3.5 函数重载	35
3.6 作用域和可见性	36
3.7 进阶示例—四则运算测试程序	38
本章小结	41
习题与上机	42
 第四章 类和对象	 46
4.1 类	46
4.2 对象	52
4.3 this 指针	60
本章小结	62
习题与上机	62
 第五章 类的其他特性	 69
5.1 静态成员	69
5.2 友元	71
5.3 常类型	76
5.4 类成员指针	79
5.5 类的组合	80
5.6 进阶示例—模拟数字时钟	82
本章小结	85
习题与上机	85
 第六章 类的继承和派生	 90
6.1 继承的概念	90
6.2 派生类的继承方式	93
6.3 派生类的构造和析构	100
6.4 调整基类成员在派生类中的访问属性	104
6.5 多重继承	109
6.6 赋值兼容性规则	118
6.7 进阶示例——多重继承技术	120
本章小结	123

习题与上机	124
第七章 多态性和虚函数	135
7.1 多态性概述	135
7.2 运算符重载	137
7.3 虚函数	142
7.4 抽象类	154
7.5 进阶示例——公司人员管理程序	156
本章小结	160
习题与上机	161
第八章 模板	166
8.1 模板的概念	166
8.2 函数模板与模板函数	167
8.3 类模板和模板类	170
本章小结	174
习题与上机	174
第九章 I/O 流	177
9.1 C++ 的流及流类库	177
9.2 预定义类型的输入输出	178
9.3 文件的输入输出	184
本章小结	186
习题与上机	187
第十章 异常处理	189
10.1 异常的概念	189
10.2 异常处理的实现	190
10.3 异常处理的规则	192
10.4 基于对象的异常处理	194
10.5 异常的重抛出	197
本章小结	198
习题与上机	199

第十一章 综合案例	203
11.1 五子棋游戏	203
11.2 图像处理 DIY	217
本章小结	229
附录 A C++语言中的关键字（保留字）	230
附录 B 运算符优先级表	232
附录 C 基本 ASCII 码字符	234
附录 D Windows 位图结构数据域的含义	236
参考文献	237

第一章 程序设计与 C++ 基础

程序设计是根据特定的问题,使用某种程序设计语言,设计出计算机能够执行的指令序列。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发展历程出发,介绍C++语言的产生、发展,C++语言的特点及C++程序的开发过程。通过一个简单例程的详细讲述,让学习者逐步进入C++的精彩世界。

学习要点

- 了解不同类型的编程语言
- 了解面向过程的程序设计语言及面向对象程序设计语言的特点
- 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

程序设计语言是计算机可以识别的语言,具有特定的词法与语法规则,由人编写,用于描述解决问题的方法,供计算机阅读和执行。



1.1.1 计算机程序设计语言和程序

1. 机器语言和汇编语言

机器语言是计算机自身的语言,是人类使用的第—种编程语言,其实际上就是由二进制0和1构成,它通过一系列的指令代码来完成操作,优点是直接在计算机硬件上执行,效率高。缺点也很明显,用机器语言编写的程序要求程序员必须记住机器指令的二进制代码和地址码,这样的程序不直观且编写难度大,不同型号的计算机其机器语言也不相同。

为了解决机器语言编程的困难,人们想到用一些助记符来代替二进制代码,如ADD表示加,MOV表示移动,LOAD表示读取等,优点是易于理解且效率较高。由于汇编语

言的抽象层次太低，一个简单的任务需要大量的语句实现，并且程序员还需考虑大量的机器细节，故编程难度大。

2. 高级程序设计语言

为了进一步方便编程，人们开发了更加接近人类自然语言习惯的高级语言，因为它对具体算法进行描述，所以又称为算法语言。在计算机上高级语言程序（源程序）不能直接执行，必须将它们翻译成具体的机器语言程序（目标程序），这种翻译由编译程序来完成。优点是面向问题，易于人们的理解和掌握，通过性和可移植性好。

3. 程序

“程序”有两种含义：一是由程序员编写的源代码，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。现在非常受重视的是易维护的代码。所谓易维护是指当用户改变时程序能够较容易地扩展和增强。



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。过程或函数被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。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，它是程序基础运行单位，具有一定的独立性，其他对象的操作不能访问该对象隐藏起来的数据。在面向对象程序设计中，将程序设计为一组相互协作的对象而不是一组相互协作的函数。所有的 OOP 语言（包括 C++）都具有三个共同的特性，即封装性、继承性和多态性。

1. 封装性

封装（encapsulation）是一种机制，它将代码和数据链接起来，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链接时一个对象（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

继承 (inheritance) 是一个进程。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

所谓多态 (polymorphism) 就是一个接口，多种方式。多态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++ 起源和特点

1. C++ 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 语言的基础上设计的，1972 年由 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，设计者于 1994 年制定了 ANSI C++ 标准草案。C++ 的发展如图 1—1 所示。

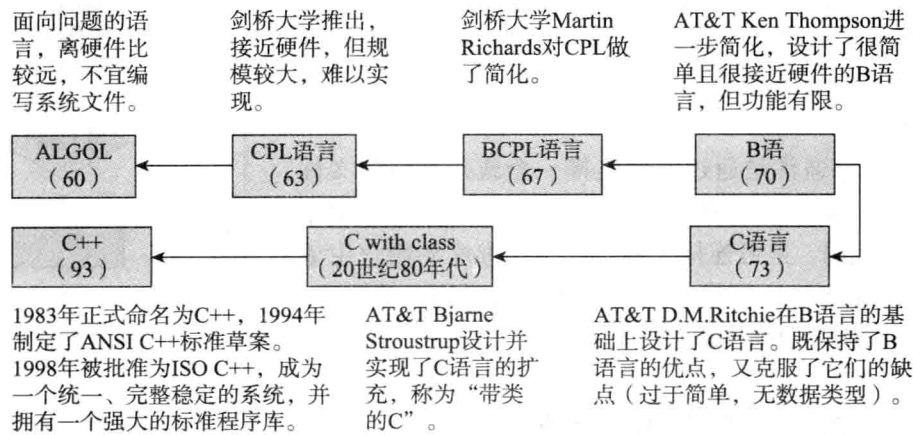


图 1—1 C++ 的发展

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++特点

C++现在得到了越来越广泛的应用，它继承了C语言的优点，并有自己独到的特点，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，可直接地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++的人正迅猛增加，优秀的C++版本和配套工具软件不断涌现。

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。文件与库连接的结果，即生成计算机可执行的程序。开发C++程序的步骤见图1—2。

1.2.3 最简单的程序

从一个最简单的程序来分析C++的程序构成。

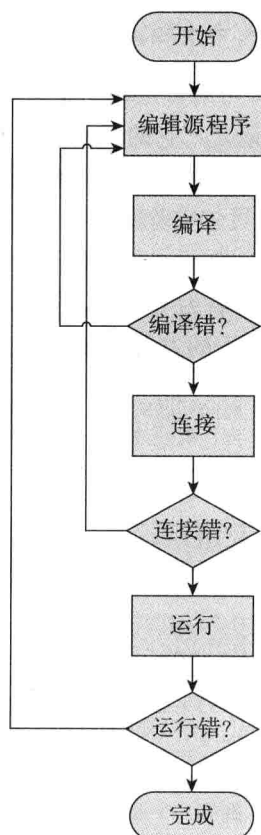


图1—2 开发C++程序的步骤

(1) 注释是程序员为读者作的说明, //或/**/。

(2) 每个以‘#’开头的行称为编译处理行。

iostream.h是系统定义的一个头文件,它设置了C++的I/O
相关环境,定义输入输出流对象cin与cout等。

//C++程序由注释、预处理命令、输入输出、函数和变量组成。

```
#include "iostream.h"
```

```
void main()
```

(3) main()表示主函数,每一个C++程序都必有一个main()函数。

```
{ cout<<"Let's learn to write a C++ program! ";
```

(4) “;”表示语句结束。

```
}
```

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,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 为环境进行介绍。



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。

(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”选项,在弹出的“新建”对话框(如图 1-4 所示)中选中“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,或单击右侧的“...”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。

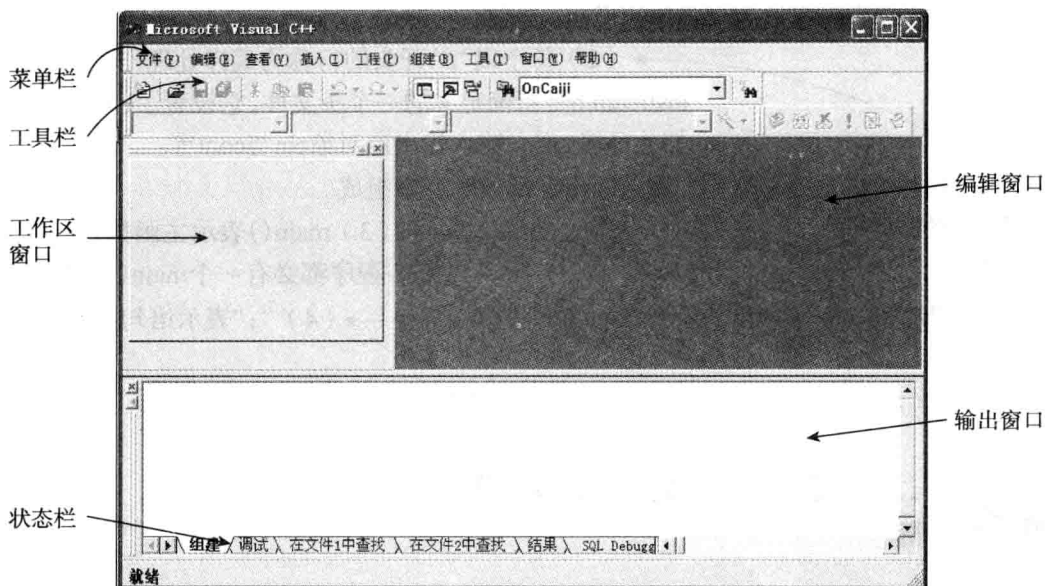


图 1—3 VC++6.0 的主窗口

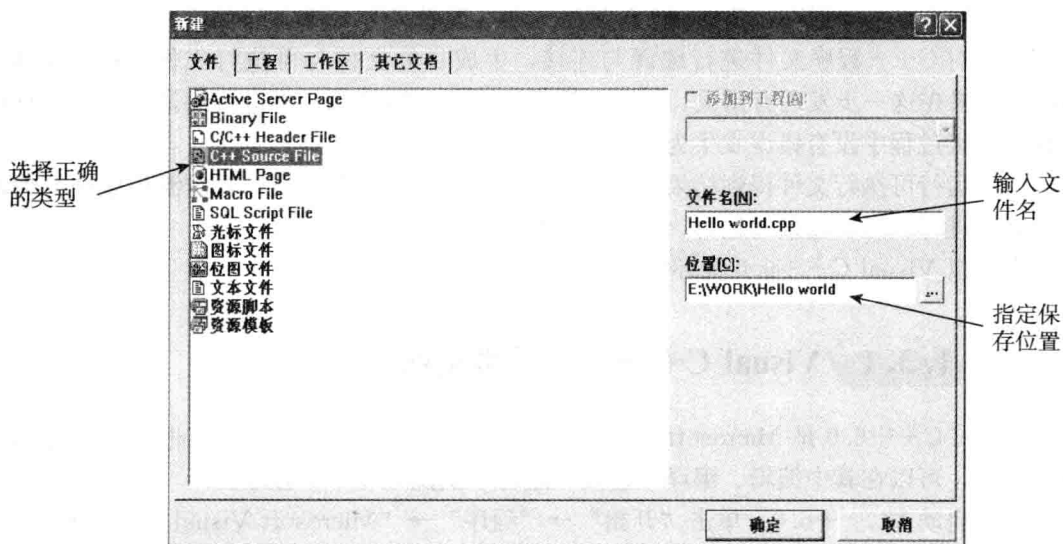


图 1—4 “新建”对话框

(3) 编辑文件。在文件编辑窗口输入程序代码以后，单击“文件”菜单的“保存”命令或工具栏上的“保存”按钮进行保存，如图 1—5 所示。

(4) 编译程序。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。或单击编译微型条的“编译”按钮或“组建”按钮。编译微型条的各工具功能如图 1—6 所示。(说明：组建=编译+连接。)

单击编译后，VC++6.0 会立即弹出提示信息框，如图 1—7 所示，意思是“需要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—5 编辑并保存文件

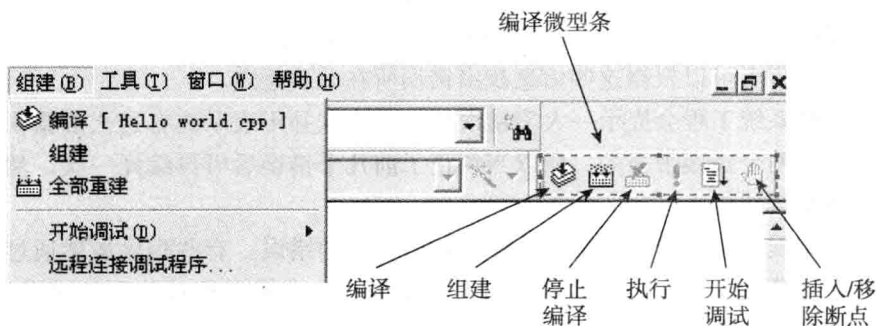


图 1—6 编译程序

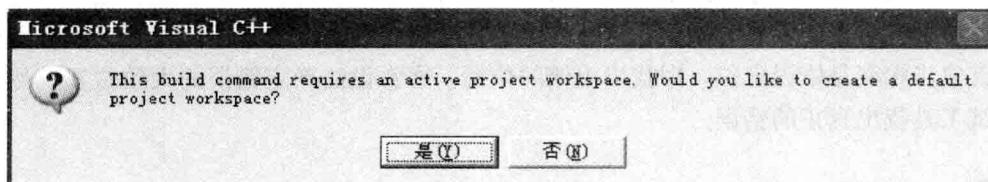


图 1—7 创建默认的项目工作区询问对话框

(5) 运行程序。选择“组建”菜单下的“执行”菜单项，或编译微型条的执行按钮即可运行当前程序。具体内容当在输出窗口出现“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，按任意键关闭该窗口。

(6) 调试程序。将在下一节中介绍。

(7) 关闭工作区。选择“文件”菜单中的“关闭工作空间”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。

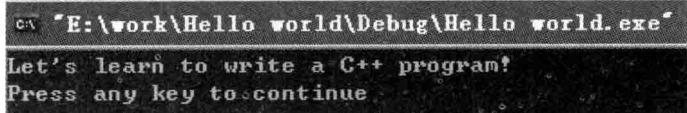


图 1—8 运行程序

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：

(1) 编译错误。编译错误是指程序编译时检查出来的语法错误。编译错误通常是编程者违反了 C++ 的语法规则，如大括号不匹配，语句少分号等。

(2) 连接错误。连接错误是指程序连接时出现的错误。连接错误一般由未定义或未指明要连接或包含的函数，或者函数调用不匹配等因素引起。

对编译错误和连接错误，C++ 语言系统会提供出错信息，包括出错位置（行号），出错提示信息。编程者可以根据这些信息找出错误所在并改正之。

说明：有时系统工程会提示一大串错误信息，但这并不表示真有这么多个错误。这往往是因为前面的一两个错误带来的。所以当纠正了前几个错误后可再编译一次，然后根据最新出错信息再继续纠正。

(3) 运行错误。运行错误是指导程序执行过程中的错误。有些程序虽然通过了编译连接，并能够在计算机上运行，但得到的结果不正确。这类错误相对前两种错误较难改正，所以要求程序设计者认真分析程序的执行过程。

错误的原因一部分是程序书写错误带来的，例如应该使用变量 x 的地方写成了变量 y ，虽然没有语法错误，但意思完全错了。另一部分可能是程序的算法不正确。还有一些程序的计算结果有时正确，有时错误，这往往是编程时对各种情况考虑不周所致。解决运行错误的首要步骤就是错误定位，即找出出错的位置，通常设法确定错误的大致位置，然后通过调试工具找出真正的错误。



本章小结

程序设计语言经历了从机器语言、汇编语言、高级语言、面向对象的语言的发展过程，使得编程语言与人的自然语言越来越接近，编程效率越来越高。

程序设计方法经历了面向过程的方法、结构化的方法、面向对象的方法的发展过程。

C++ 语言兼容 C 语言，具有面向对象的特点，支持面向对象程序设计方法。

面向对象的基本概念有对象、类、封装、继承、消息和多态。面向对象的软件开发包括面向对象的分析、面向对象的设计、面向对象的编程、面向对象的测试和面向对象的