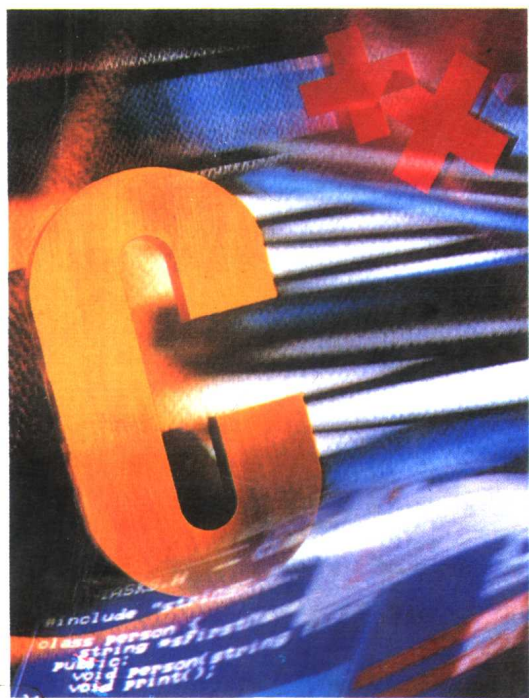


计算机 C/C++ 语言系列丛书

BORLAND C++

郝阿朋 编写
陈材保

程序设计基础教程



学苑出版社

计算机 C/C++ 语言系列丛书

BORLAND C++ 程序设计基础教程

郝阿朋 陈材保 编写
希 望 审校

学苑出版社
1993

(京)新登字 151 号

内 容 提 要

本书是一本全面介绍 C++ 语法元素与编程技巧的基础教程。本书提供了大量的程序示例及习题,这些程序在 Borland C++ 3.0 环境下运行过,程序不要修改就可以在 Microsoft C++ 环境下运行。本书可作为大专院校的教材,也可以作为 C++ 程序员的自学教程和参考手册。

欲购本书的用户可直接与北京 8721 信箱资料部联系,邮码:100080,电话:2562328。

计算机 C/C++ 语言系列丛书

Borland C++ 程序设计基础教程

编 写:郝阿朋 陈材保
审 校:希 望
责任编辑:徐建军
出版发行:学苑出版社 邮政编码:100032
社 址:北京市西城区成方街 33 号
印 刷:兰空印刷厂印刷
开 本:787×1092 1/16
印 张:20.375 字 数:464 千字
印 数:1~5000 册
版 次:1993 年 11 月北京第 1 版第 1 次
ISBN7-5077-0875-6/TP·24
本册定价:23.00 元

学苑版图书印、装错误可随时退换

《C++程序设计基础教程》介绍

Borland C++是 Turbo C 与 C++有机结合的一种高效而且支持面向对象程序设计的新型通用程序设计语言,适合于编制各种系统软件和应用软件。它是当今世界上最有影响的程序设计语言之一。目前在我国软件开发领域,它也是十分流行的。

本书就是专门为 C++的初学者编写的教学型书籍。

本书在内容上,全面而系统地介绍了 Borland C++程序设计的基本概念和方法。较全面地说明了 C++的运算符、数据结构、程序结构、用户函数与库函数、文件输入输出和预处理程序,尤其是重点介绍了对象、类、继承性、多态性等概念及其编程机制,此外,还介绍了 C++的图形处理功能。全书共分十五章,为便于读者检验和巩固所学的内容,每章后附有习题,书的最后附有习题参考答案。

本书中的例题及习题参考答案全部在 IBM—PC 机上,用 Borland C++3.0 编译运行过。除个别之处,程序不必修改就可在 Microsoft—C++上编译运行。

本书在写法上,根据初学者特点由浅入深,循序渐进。对一些概念和规则的说明简单明了。为增强直观性,给出了大量完整的带有运行结果和程序分析的例子(104个),这样便于读者对概念和规则的理解,同时也方便了读者机上操作练习。所有例子均不涉及复杂的物理和数学背景,算法也很简单,主要用于说明语法规则和使用特性。

本书可以作为各类使用计算机人员的培训教材或计算机爱好者的自学教材。

C++是一种专业性较强的计算机程序设计语言。本书作为一种普及性教材,尽量避免过多的介绍和引用专业概念和术语,它特别注重于实用性,总是试图用最简洁直观的方式介绍语言的特性,使读者能很快地掌握这种语言,使之在使用的过程中提高对 C++的兴趣和加强对概念的理解。通过本书的学习之后,可使读者掌握 C++的基本特性,为进一步学习 C++高级编程打下基础。

本书的第十三章“类”和第十四章“输入/输出”是由陈材保同志编写,其余部分是由郝阿朋同志编写。由于水平和经验所限,本书肯定有许多缺点和不足之处,请读者批评指正。

编者

一九九三年九月

目 录

第一章 Turbo C++概述	(1)
第一节 C++的特点	(1)
一、C语言的出现与发展	(1)
二、面向对象的程序设计(OOP)	(1)
三、Turbo C++的出现	(2)
四、Turbo C++的运行环境	(2)
五、C++的特点	(3)
第二节 C++程序的基本组成结构	(4)
一、程序设计中的几个基本概念	(4)
二、一个简单的例子	(5)
三、C++程序的基本结构	(6)
第三节 Turbo C++的集成开发环境(IDE)	(7)
一、进入和退出 Turbo C++集成开发环境(IDE)	(7)
二、建立、编辑源程序文件	(8)
三、运行程序(编译、链接、运行)	(9)
四、源程序的存盘和取盘	(9)
第四节 C++程序的汉字处理	(10)
一、编辑和修改汉字程序	(10)
二、汉字程序的编译和链接	(11)
三、在汉字系统下运行汉字程序	(11)
四、在西文系统下运行汉字程序	(11)
第二章 词法	(14)
第一节 基本符号	(14)
一、基本符号	(14)
第二节 标识符与关键字	(14)
一、标识符	(14)
二、关键字	(15)
第三节 空白符与注释行	(15)
一、空白	(15)
二、行分隔符	(15)
三、注释	(16)
第四节 常量、变量	(17)
一、常量	(17)
二、变量	(18)
第三章 C++的基本数据类型	(20)
第一节 C++类型系统的基本原则	(20)

第二节	整型(int)	(20)
第三节	浮点类型(float)	(22)
第四节	字符类型(char)	(23)
第五节	类型转换	(24)
一、	自动类型转换	(24)
二、	赋值转换	(24)
三、	强制类型转换	(24)
第六节	C++输入/输出简介	(25)
一、	C++显示输出	(25)
二、	C++键盘输入	(26)
第四章	运算符与表达式	(28)
第一节	算术运算符、赋值运算符及其表达式	(28)
一、	二元运算	(28)
二、	一元运算	(29)
三、	赋值运算	(30)
第二节	关系运算符与关系表达式	(31)
一、	关系运算符	(31)
第三节	逻辑运算符与逻辑表达式	(32)
一、	逻辑运算	(32)
第四节	位运算符及其表达式	(33)
一、	运算符及其表达式	(33)
第五节	一元加一、减一运算及其表达式	(37)
一、	一元加1、减1运算	(37)
第六节	自反运算符及其表达式	(40)
一、	自反运算	(40)
第七节	其它运算符及其表达式	(42)
一、	sizeof运算	(42)
二、	条件运算	(43)
三、	顺序运算	(44)
第八节	运算符的优先级	(44)
一、	优先级	(44)
二、	结合性	(44)
三、	C++运算符属性表	(45)
第五章	C++的程序结构	(47)
第一节	顺序结构及其语句	(47)
一、	变量说明语句	(47)
二、	表达式语句	(48)
三、	输入输出语句	(48)
四、	函数调用语句	(49)

五、顺序结构举例.....	(50)
第二节 选择结构及其语句	(50)
一、条件语句(if)	(50)
二、多路分支语句(switch)	(54)
第三节 循环结构及其语句	(56)
一、while 循环语句	(56)
二、do while 循环语句	(57)
三、for 循环语句	(58)
第四节 辅助控制语句(break、continue、goto)	(59)
一、无条件转向语句(goto)和标号	(59)
二、跳出语句(break)	(61)
三、继续语句(continue)	(62)
第六章 数组与字符串	(64)
第一节 一维数组	(64)
一、数组的说明.....	(64)
二、数组的操作.....	(65)
第二节 字符串数组	(68)
一、字符数组的说明.....	(69)
二、字符数组的操作.....	(69)
第三节 二维数组及多维数组	(72)
一、二维数组的说明.....	(72)
二、二维数组的操作.....	(73)
第七章 指针	(79)
第一节 指针与地址的概念	(79)
一、指针与对象变量.....	(79)
二、指针说明.....	(79)
第二节 指针的取地址和取内容运算	(80)
一、取地址运算.....	(80)
二、取内容运算.....	(81)
三、举例.....	(81)
第三节 指针与数组	(82)
一、指针与一维数组.....	(83)
二、指针与多维数组.....	(85)
第四节 数组指针的运算	(87)
一、关系运算.....	(87)
二、算术运算.....	(87)
第五节 字符串指针	(89)
一、字符串指针的说明.....	(90)
二、字符串指针的操作.....	(90)

第六节 指针数组	(92)
一、指针数组的说明	(92)
二、应用举例	(92)
第八章 结 构	(95)
第一节 结构类型与结构变量	(95)
一、结构类型的定义	(95)
二、结构变量的说明	(96)
第二节 结构变量的使用	(97)
一、结构成员的引用	(97)
二、结构成员的运算及输入/输出	(98)
三、举例	(98)
第三节 结构数组	(99)
一、结构数组的说明	(99)
二、结构数组元素成员的引用	(100)
三、结构数组元素成员的运算及输入/输出	(100)
四、举例	(101)
第四节 结构指针	(102)
一、结构指针的说明	(102)
二、结构指针对结构变量成员的引用	(102)
三、结构指针与结构数组	(104)
第五节 结构的递归与链表	(105)
一、递归结构	(105)
二、内存分配函数(malloc 和 free)	(106)
三、new 和 delete 运算	(107)
四、链表	(107)
第九章 联合、枚举	(112)
第一节 联合	(112)
一、联合类型	(112)
二、联合变量	(113)
三、联合的使用	(113)
四、联合与结构比较	(114)
五、无名联合(C++专用)	(116)
第二节 枚举类型	(117)
一、枚举类型的定义	(117)
二、枚举变量的说明	(118)
三、枚举变量的操作	(119)
第十章 函 数	(121)
第一节 函数的定义与调用	(121)
一、函数的定义	(121)

二、函数的调用	(122)
三、返回语句(return)与返回值	(125)
第二节 参数传递	(126)
一、传递值	(127)
二、传递指针	(128)
三、使用缺省值的函数参数	(131)
第三节 void 类型	(132)
一、void 返回类型	(132)
二、void 参数类型	(132)
第四节 函数的递归调用	(133)
第五节 函数的重载	(134)
第六节 库函数介绍	(135)
一、数学计算函数(MATH. H)	(135)
二、字符分类函数(CTYPE. H)	(137)
三、字符串操作函数(STRING. H)	(138)
四、图形处理函数(GRAPHICS. H)	(138)
第十一章 存储类别	(142)
第一节 变量的存在性和可见性	(142)
第二节 变量的存储类别	(143)
一、自动变量(auto)	(143)
二、寄存器变量(register)	(145)
三、静态变量(static)	(146)
四、外部变量(extern)	(147)
五、只读变量(const)	(148)
第三节 函数的存储类别	(149)
第十二章 C++的预处理程序	(152)
第一节 宏替换命令(#define、#undef)	(152)
一、简单宏替换	(152)
二、带参数宏替换	(153)
三、取消宏替换	(154)
第二节 文件嵌入命令(#include)	(155)
第三节 条件编译命令(#if、#elif、#else、#endif 等)	(156)
一、格式与功能	(156)
二、举例	(157)
第十三章 类	(159)
第一节 程序语言的发展趋势	(159)
第二节 类与对象的建立	(159)
一、C++的类	(159)
二、C++的对象	(160)

三、关于栈的一个程序例题	(160)
第三节 类中成份的内、外联系	(162)
一、内联码(inline)	(162)
二、类内成份的可访性	(163)
三、成员函数与友元函数	(163)
四、构造函数和析构函数	(167)
五、类的继承	(170)
六、多继承	(173)
七、对象与指针	(177)
八、对象向外部的信息传递	(180)
九、C++的其它聚集类型	(184)
第四节 类与多态性.....	(185)
一、局部化和动态初始化	(185)
二、函数的重载	(188)
三、运算符的重载	(190)
四、用友元函数重载运算符	(196)
第五节 继承与多态性.....	(200)
一、虚函数	(200)
二、纯虚函数与抽象类	(204)
第十四章 输入和输出	(209)
第一节 流类与预定义流.....	(209)
第二节 插入运算符<<和提取运算符>>的重载.....	(210)
一、提取运算符>>的重载	(210)
二、插入运算符<<的重载	(211)
三、矩阵乘法的例题	(211)
第三节 输入和输出的格式化控制.....	(214)
一、用 ios 类中的成员函数控制 I/O 格式.....	(214)
二、用控制器函数控制 I/O 格式	(218)
三、建立自己的控制器函数	(220)
第四节 文件的输入和输出.....	(222)
一、文件的打开和关闭	(222)
二、设备文件的使用	(226)
三、二进制文件	(227)
四、到达文件尾的判断	(231)
五、随机读写	(233)
第十五章 C++程序的图形处理	(237)
第一节 显示器的工作模式.....	(237)
一、显示卡(显示适配器)	(237)
二、文本模式	(237)

三、图形模式	(237)
第二节 窗口和视口	(238)
一、窗口	(238)
二、视口	(238)
第三节 文本模式下的程序设计	(238)
一、文本模式下程序设计的基本结构	(238)
二、文本输出及管理	(240)
三、窗口及模式控制	(240)
四、属性控制	(240)
五、状态查询	(242)
第四节 图形模式下的程序设计	(243)
一、图形模式下程序设计的基本结构	(243)
二、图形系统控制	(245)
三、绘图及填充	(246)
四、管理屏幕及视口	(250)
五、正文输出	(251)
六、颜色控制	(251)
习题参考答案	(254)
附录 A ASCII 码表	(281)
附录 B Turbo C++ 库函数	(283)
一、分类函数	(283)
二、数学函数	(284)
三、串和内存操作函数	(289)
四、输入输出函数	(293)
五、图形函数	(301)

第一章 Turbo C++ 概述

第一节 C++ 的特点

C++是在C程序设计语言的基础上建立起来的。一方面,C++包含了C语言的全部功能和特性,它是C语言的超集。另一方面,C++增加了一些新的概念和功能,例如对象、类等,这为体现当今最先进的程序设计思想即面向对象的程序设计提供了编程机制。下面就简单介绍一下C++的发展情况。

一、C语言的出现与发展

C语言是七十年代初出现的一种新型通用程序设计语言,适合于编制各种软件,包括系统软件和应用软件。它是当今世界上最有影响的程序设计语言之一。目前我国软件开发领域,C语言十分流行,主要原因是,一方面它具有语言简洁、表达力强、代码质量高、可移植性好等特点,它的独到之处是兼具“高级语言”的特性与“低级处理”的能力,它能代替汇编语言直接与机器硬件打交道,这是众多其它高级语言所不能的。另一方面,C语言在各类机(大、中、小、微)上均能使用,尤其在各类微机上十分普及。此外,C语言有极其雄厚的后源支撑,众多实用的C语言程序库和程序包为用户提供了良好的开发环境。因此它深受广大程序员的喜爱。但是,C语言也有不足的一面,例如,其语法检查很不严格,因此,程序的可靠性要由程序员自己负责。

进入90年代,计算机的应用成倍增长。人们对计算机的软硬件期望之高也是空前的。这就使得当今的软件设计要面对三方面的难题。

1. 难度急速上升:在软件开发前和开发过程中,系统需求或者不甚明确,或者一直在变化。随着计算机应用深入到不同专业,一个成功的软件必定会涉及到很多领域的专业知识。因此,在整个软件系统生命周期(从设计开发一直到被淘汰)中,要不断对其进行修改和完善。这就要求软件系统的可读性和可维护性要好。

2. 软件大型化:开发者和用户都趋向于把系统的功能设置得完善,用户界面要漂亮友好。因此,这使得软件的规模越来越大。这就要求能够有效地对大型软件进行控制。

3. 软件需求规模化:软件的“生产“已面临着不是手工作坊”单件”生产,而逐步转向流水线式的批量生产。

由于上述原因和硬件费用的不断下降,使得软件费用不断上升,软件的可靠性不断下降,管理和控制软件的开发过程变得十分困难,这就是所谓的软件危机。解决软件危机的途径,主要是通过采用先进的程序设计方法和能够体现先进程序设计方法的程序设计语言。

在这种形势下,C语言就显得程序的可靠性不高、可维护性不好和对开发大型软件(十几万行以上的源程序)难以管理和控制等不足。

二、面向对象的程序设计(OOP)

面向对象的程序设计(Object Oriented Programming)是一种比较先进的程序设计方法。它

的最大特点是让程序员能够理解和控制更大、更复杂的程序,它特别支持自顶向下地进行层次化、模块化程序设计。其关键是能够确立对象,并将对象成员即所包含的数据及对这些数据进行操作的函数(代码)封装起来。对象的成员又分为私有成员和公共成员,对象的私有成员只能由对象中的函数所访问,对象的公共成员可以由程序中的任何部分访问。这样就便于数据隐藏和保护。

有关 Turbo C++ 的面向对象设计的程序机制,在第十三章中有详细介绍。

三、Turbo C++ 的出现

Turbo C++ 是美国 BORLAND 公司在 Turbo C 的基础上,扩充引进了面向对象的概念和功能而发展成功的,并于 1983 年正式向外界公开推出。C++ 是一种面向对象开发方法的程序设计语言。由于 C++ 提供了把数据和在数据上的操作封装在一起的类、对象和方法机制,并通过派生、继承、重载和多态性等特性,实现了人们所希望的软件重用技术,使得软件,特别是大型复杂软件的设计和与维护变得更有效和容易,并使软件开发能更自然的反映事物的本来面貌。此外,C++ 又与 C 语言完全兼容,因次,它在软件开发领域受到广泛的欢迎。

四、Turbo C++ 的运行环境

不同版本的 Turbo C++ 对环境的要求不同。现以 Turbo C++ 1.0 和 Turbo C 3.0 为例说明。

1. Turbo C++ 1.0

1) 硬件环境

Turbo C++ 1.0 运行于 IBM 系列 PC 机及其兼容机,其中包括 XT(8088、8086)、AT(286)、386、486、PS/2 机等。

并要求机器配有一个硬盘驱动器,一个软盘驱动器,至少需要 640K 内存,还需要 80 列的单色或彩色显示器。

Turbo C++ 1.0 包含有浮点运算子程序,从而使你的程序能利用 80x87 协处理器。若没有协处理器,它可以进行仿真运算,但运算速度要大为降低。

2) 软件环境

Turbo C++ 1.0 需要 DOS 3.30 以上版本。

Turbo C++ 支持鼠标器,它不是必需的,但 Resource Workshop(资源工作库)是需要鼠标器的。若配置有鼠标器,你就必需用下列实用程序之一,才能做到完全兼容。

- Microsoft Mouse 6.1 版本或更高版本。
- Logitech Mouse 3.4 版或更高版本。
- Mouse Systems 的 PCMouse 6.22 版或更高版本。
- IMSI Mouse 6.11 版或更高版本。

2. Turbo C++ 3.0

1) 硬件环境

Turbo C++ 3.0 运行于 IBM 系列 PC 机及其兼容机,其中包括 AT(286)、386、486、PS/2 机等。

并要求机器配有一个硬盘驱动器,一个软盘驱动器,至少需要 640K 内存,另外再加上 1MB 扩充内存,还需要 80 列的单色或彩色显示器。

Turbo C++ for Windows 的集成开发环境(IDE)至少需要 2MB 的扩充内存。

Turbo C++ 3.0 包含有浮点运算子程序,从而使你的程序能利用 80x87 协处理器。若没有协处理器,它可以进行仿真运算,但运算速度要大为降低。

2) 软件环境

Turbo C++ 3.0 需要 DOS 3.30 以上版本。

Turbo C++ for Windows 的 IDE 需要保护模式 Windows 3.0 或更高版本。

Turbo C++ 支持鼠标器,它不是必须的,但 Resource Workshop(资源工作库)是需要鼠标器的。若配置有鼠标器,你就必须用下列实用程序之一,才能做到完全兼容。

- Microsoft Mouse 6.1 版本或更高版本。
- Logitech Mouse 3.4 版或更高版本。
- Mouse Systems 的 PCMouse 6.22 版或更高版本。
- IMSI Mouse 6.11 版或更高版本。

五、C++ 的特点

由于 C++ 包含了 C 语言的全部特性,并体现了面向对象的程序设计思想,所以,从使用者的角度看,C++ 语言主要具有以下特点:

1. C++ 是一种高效率的编译型语言

C++ 语言适当地考虑了背景机,因此,其生成的目标程序质量较高。对于较大的程序,C++ 的源程序可分开存放在若干个文件中,待分别编译以后再连接在一起,形成一个可执行文件。

2. 语言简洁、结构清晰

C++ 程序通常是由若干个函数组成。C++ 语言强大的函数功能为程序的模块化、结构化提供了手段。因而使得源程序显得简洁清晰、可读性较强。

3. 表达能力强

C++ 语言提供了丰富的运算符以及可进行低级处理的位操作、地址操作。此外,还提供了先进的程序控制结构和数据结构。

4. 强大的库函数

C++语言系统为方便用户提供了丰富的库函数,如1987年由美国Borland公司推出的Turbo C++具有十几类300多个函数子程序。另外,C++语言也便于用户定义和使用自己的函数。

5. 支持面向对象的程序设计方法

C++通过类和对象等概念,提供了面向对象的程序设计机制,增强了管理和控制开发大型软件的能力。

C++语言不仅适合开发各类应用软件,而且适合开发系统软件。因此,C++深受广大计算机软件人员及计算机爱好者的欢迎。

第二节 C++程序的基本组成结构

一、程序设计中的几个基本概念

在介绍C++之前,先说明几个程序设计中要用到的基本概念。

1. 源程序

用汇编语言或高级语言编写的程序称为源程序。机器硬件不能直接识别和执行源程序。源程序是通过编辑软件(如:Wordstar)编辑生成的,它是以文件形式存在盘上。

2. 目标程序

将源程序翻译后,生成的二进制代码程序。目标程序机器能够识别,但还不具备运行条件。目标程序是通过语言系统提供的翻译程序对源程序翻译后生成的。目标程序也是以文件形式存在盘上的。

3. 可执行程序

计算机能直接识别和执行的程序。是将目标程序与所需要的库文件等链接后生成的程序。可执行程序是由语言系统提供的链接程序对目标程序进行链接所生成的,它也是以可执行文件的形式存在盘上的。

4. 编译程序

它是编译型高级语言系统所提供的一种翻译程序,其作用是,将高级语言编写的源程序翻译成二进制代码程序。例如,FORTRAN语言、C语言为编译型语言。因此,它们的程序设计必须经过编辑、编译、链接、运行等,几个步骤来进行。

5. 汇编程序

它是汇编语言所提供的一种翻译程序,其作用是,将汇编语言编写的源程序翻译成二进制代码程序。

6. 链接程序

它是语言系统所提供的一种装配程序,其作用是,将若干个目标程序和指定的库文件链接起来,产生一个可执行的程序文件。

7. 解释程序

它是解释型语言系统提供的一种翻译程序,其与编译程序的区别在于,它对源程序中的语句,翻译一句,执行一句。而不是全部翻译完后,再链接生成执行文件,才能执行。例如,BASIC 是一种解释型语言。BASIC 解释程序对 BASIC 源程序解释一句,执行一句。进入 BASIC 状态,也就进入了解释、运行状态。这时,就可对输入的 BASIC 源程序进行解释执行。

8. 编译时间

编译过程所用的时间。编译过程所出现的错误称为编译错误。

9. 运行时间

程序执行所用的时间。执行过程所出现的错误称为运行错误。

二、一个简单的例子

下面让我们通过一个完整的例子说明C++程序的组成结构。

例 1.2.1:从键盘上输入任一整数 N,计算其平方,然后在显示器上输出结果。

```
1      /* The example is print the square of N */
2      #include <stdio.h>
3      #include <iostream.h>
4      main()
5      int sqr(int x);
6      {
7          int n,s;
8
9          cout <<< "n=";
10         cin >>> n;
11         s=sqr(n);
12         printf("\ns= %d",s);
13     }
14     // square the integer
```



```

15     int sqr(int x)
16     {
17         return(x * x);
18     }

```

执行结果:

n=5(键入)

s=25

说明:

1. C++源程序中并无行号,本书中的C++程序给出行号只为说明程序方便。

2. 第1行是注释行。注释只用来对程序进行说明,编译时不产生代码,因此注释不影响程序的运行。C++的注释可以由“/*”和“*/”括起来的一系列字符,注释可出现在语句之后,也可出现在语句之前。当注释要连续占几行时,每行要以“/*”开头,以“*/”结束。

3. 第2、3行是预处理命令。C++程序中若调用了C++的某个库函数,就必须用(#include)预处理命令,将包含该函数说明的“头文件”嵌入源程序文件中。因本例中调用了包含在“stdio. h”和“iostream. h”头文件中说明的输出函数 printf)和C++输入/输出流(cout、cin),所以本行命令进行了嵌入说明。

4. 从第4到13行是名为 main() 的无参数函数。main() 被称为主函数,它是任何一个完整的C++程序中必不可少的,它是能直接运行的函数。在C++语言中,程序编译、运行的基本单位是函数。每个函数都有一个名字。

5. 第4行是对sqr函数的一个简单说明,因为在主函数(main())中要调用此函数。

6. 第6行的花括号“{”和第13行的花括号“}”将函数的语句部分即函数体括起来,这是说明C++函数所要求的形式符号。

7. 第7行为说明语句,说明n,s为整型变量。在C++中,要求对所使用的变量必须进行说明。“;”为语句终止符,C++中,语句一般以“;”号为结束标志。

8. 第8行是空行,表明C++程序中语句间可以有空行。这是因为C++程序的书写格式是自由的。

9. 第9行的语句功能是在显示器上输出一串“n=”,这是C++输出的特殊形式。

10. 第10行的作用是从键盘上为整型变量n输入值。这是C++输入的特殊形式。

11. 第11行是调用sqr函数计算变量n的平方值,然后赋给变量s。

12. 第12行是利用格式输出函数 printf)在显示器上先换一个新行,并显示整数变量s的值。

13. 第14到18行是定义了名为sqr的求平方的函数。此函数在主函main()中被调用。本例中用到的各种语句,将在后面章节中陆续介绍。

三、C++程序的基本结构

通过上面的例子,可以看出C++程序的基本结构为:

预处理命令 ←嵌入命令、宏定义等