

面向21世纪高校教材

江苏省普通高校计算机等级考试系列教材

Visual C++ 程序设计

张岳新 编著

江苏省教育厅组织编写



24 V
713120-83
Z166

面向 21 世纪高校教材
江苏省教育厅组织编写

江苏省普通高校计算机等级考试系列教材

Visual C++程序设计

张岳新 编著

苏州大学出版社

内 容 简 介

C++ 是一种面向对象的程序设计语言。本书以没有学过程序设计语言、而直接学习 C++ 语言的读者为对象,重点介绍了 C++ 语言的基本概念、基本语法、程序设计的基本思想和面向对象的程序设计方法。为了便于读者学习和理解,本书提供了大量的例题,每一章后面备有相当数量的练习题和思考题。

本书分为两部分,前一部分讲述 VC++ 基础,后一部分介绍面向对象的程序设计方法,最后一章介绍了 MFC 程序设计的基本方法。本书通俗易懂,由浅入深,突出重点,侧重实用。本书可作为大专院校理工科类学生用的 C++ 语言课程的教材,也可作为计算机爱好者的自学教材。

图书在版编目(CIP)数据

Visual C++ 程序设计/张岳新编著;江苏省教育厅组
织编写. —苏州:苏州大学出版社,2002.1
面向 21 世纪高校教材 江苏省普通高校计算机等级
考试系列教材
ISBN 7-81037-931-3

I. V… II. ①张… ②江… III. C 语言—程序设计
—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 001729 号

Visual C++程序设计

张岳新 编著

责任编辑 周建兰

苏州大学出版社出版发行
(地址:苏州市王将东路 200 号 邮编:215021)
丹阳市教育印刷厂印装
(地址:丹阳市西门外 邮编:212300)

开本 787×1092 1/16 印张 25.75 字数 644 千
2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷
印数 1-15000 册

ISBN 7-81037-931-3/TP·48(课) 定价:26.00 元

苏州大学版图书若有印装错误,本社负责调换
苏州大学出版社营销部 电话:0512-7258815

前 言

C++ 语言是 C 语言的扩充,它保持了 C 语言的简洁、高效、源程序的可移植性好等特点;同时克服了 C 语言的类型检查机制薄弱和不适合开发大型程序的缺点。C++ 语言为程序设计者提供了良好的程序开发环境,能产生模块化程度高、重用性和可维护性好的程序。目前 C++ 语言已经在各个领域都得到了广泛的应用,它不仅应用于 C 语言应用的所有场合,效果比 C 语言更好,而且特别适合中等和大型程序项目的开发。

本书根据江苏省普通高校非计算机专业学生计算机基础知识和应用能力等级考试大纲(2001 年)对原书进行了修订再版。全书分为两部分:第一部分为 Visual C++ 基础,共有 8 章;第二部分为面向对象的程序设计,共有 7 章,合计 15 章。其中第 4 章(C++ 的流程控制语句)、第 8 章(指针和引用)、第 9 章(类和对象)、第 11 章(继承和派生类)和第 13 章(运算符重载)为本书的重点和难点。另外,在附录中编入了 ASCII 码表和常用的库函数。

本书较系统地介绍了 C++ 语言的基本概念和程序设计的基本方法。针对初学者在学习过程中遇到的困难和容易出现的问题,结合大量的例题进行了详细讲述。本书的特点是例题丰富、注重实用、通俗易懂、适合自学。本书虽然是以 VC++ 作为程序设计语言,但书中的内容基本上均适用于任一种 C++ 语言。

本书的所有例题均在 VC++ 6.0 版本的编译系统下运行过,在其他的 C++ 编译系统下也能正确运行。每一章后均提供了一定数量的练习题,可供读者复习时参考。

本书是江苏省普通高校计算机等级考试的指定教材,也可作为教师的教学参考书或自学者学习用书。书中加有“*”的内容和例题,对于一般学生可以不作考试或考查要求。若教学时数较少,第 15 章可以不讲授。为了便于教学,已另编一本与该本教材相配套的《Visual C++ 实验指导书》。

本书在江苏省普通高校计算机等级考试中心主任叶晓风的支持下,在多次教师研讨班上,对第一版广泛征求意见后修订定稿。修订稿经东南大学博士生导师孙志挥教授详细审阅,并提出了许多宝贵的修改意见。作者在此对孙志挥教授及所有曾经提供帮助和指导的教师们表示衷心的感谢。

由于编者的水平有限,错误和疏漏在所难免,敬请广大读者提出宝贵意见。

编 者

2001 年 10 月

目 录

Visual C++ 基础

第 1 章 C++ 概述	
1.1 C++ 语言的发展	3
1.2 C++ 的特点	4
1.3 简单的 C++ 程序介绍	4
1.4 面向对象的程序设计概述	6
1.5 VC++ 程序的上机过程	8
练习题	9
第 2 章 数据类型、运算符和表达式	
2.1 VC++ 的数据类型	10
2.2 基本运算符	17
2.3 表达式语句、空语句及自增、自减运算符	29
练习题	30
第 3 章 简单的输入/输出	
3.1 cin	32
3.2 cout	34
* 3.3 其他的输入/输出函数	37
练习题	37
第 4 章 C++ 的流程控制语句	
4.1 C++ 语言的语句和程序结构	39
4.2 选择结构语句	40
4.3 循环结构语句	48
4.4 控制执行顺序的语句	53
4.5 程序举例	57
练习题	60
第 5 章 函数和编译预处理	
5.1 函数的定义和调用	62
5.2 函数的形参、实参、返回值及函数的原型说明	64
5.3 函数的嵌套和递归调用	69
5.4 作用域和存储类	72
5.5 内联函数	80
5.6 具有缺省参数值和参数个数可变的函数	81
5.7 函数的重载	85

5.8 编译预处理	87
5.9 程序的多文件组织	94
练习题	96
第 6 章 数组	
6.1 数组的定义及应用	98
6.2 字符数组的定义及应用	111
6.3 字符串处理函数	114
练习题	118
第 7 章 结构体、共同体和枚举类型	
7.1 结构体的定义及应用	119
* 7.2 位域	127
7.3 共同体的定义及应用	128
7.4 枚举类型	130
练习题	135
第 8 章 指针和引用	
8.1 指针和指针变量	136
8.2 指针和数组	144
8.3 指针数组和指向指针的指针变量	151
8.4 指针和函数	157
8.5 new 和 delete 运算符	174
8.6 引用和其他类型的指针	178
8.7 简单链表及其应用	185
8.8 类型定义	194
练习题	196

面向对象的程序设计

第 9 章 类和对象	
9.1 概述	201
9.2 类	202
9.3 对象	207
9.4 成员函数的重载	212
9.5 this 指针	215
练习题	216
第 10 章 构造函数和析构函数	
10.1 构造函数	218
10.2 析构函数	226
10.3 实现类型转换和拷贝的构造函数	229

10.4 构造函数和对象成员	235
练习题	238
第 11 章 继承和派生类	
11.1 继承	241
11.2 初始化基类成员	247
11.3 冲突、支配规则和赋值兼容性	256
11.4 虚基类	263
练习题	267
第 12 章 类的其他特性	
12.1 友元函数	270
12.2 虚函数	275
12.3 静态成员	287
* 12.4 const、volatile 对象和成员函数	292
* 12.5 指向类成员的指针	296
练习题	300
第 13 章 运算符重载	
13.1 运算符重载	302
13.2 几个特殊运算符的重载	313
13.3 字符串类	325
练习题	331
第 14 章 输入/输出流类库	
14.1 概述	332
14.2 C++ 的基本流类体系	333
14.3 标准设备的输入/输出	339
14.4 文件流	344
练习题	357
第 15 章 MFC 程序设计基础	
15.1 Windows 和 MFC 编程	359
15.2 文档与视图结构	368
15.3 MFC 的数组类	373
15.4 鼠标使用实例	374
15.5 GDI 与文本处理	386
15.6 CString 类	390
15.7 文本处理	390
15.8 菜单的制作	396

附录 A ASCII 码表

附录 B 常用的库函数

Visual C++ 基础

- 第 1 章 C++ 概述
- 第 2 章 数据类型、运算符和表达式
- 第 3 章 简单的输入/输出
- 第 4 章 C++ 的流程控制语句
- 第 5 章 函数和编译预处理
- 第 6 章 数组
- 第 7 章 结构体、共同体和枚举类型
- 第 8 章 指针和引用

第 1 章

C++ 概述

本章介绍了 C++ 的起源、发展及其特点, C++ 程序的基本结构, 面向对象程序设计的基本概念, 上机操作过程。

1.1 C++ 语言的发展

自从 1946 年世界上第一台数字电子计算机 ENIAC 问世以来, 计算机应用领域不断扩大, 计算机技术高速发展, 尤其是近年来计算机的硬件和软件日新月异。作为应用计算机的一种工具——程序设计语言, 也得到了不断的充实和完善。几乎每年都有新的程序设计语言问世, 而原先的程序设计语言也不断地更新换代。

C++ 语言是在 C 语言的基础上逐步发展和完善起来的, 而 C 是吸收了其他语言的一些优点逐步成为实用性很强的一种语言。

20 世纪 60 年代, Martin Richards 为计算机软件人员在开发系统软件时, 作为记述语言使用而开发了 BCPL 语言(Basic Combined Programming Language)。1970 年, Ken Thompson 在继承 BCPL 语言的许多优点的基础上发明了实用的 B 语言。1972 年, 贝尔实验室的 Dennis Ritchie 和 Brian kernighan 在 B 语言的基础上, 作了进一步的充实和完善, 设计出了 C 语言。当时, 设计 C 语言是为了编写 UNIX 操作系统的, 以后, C 语言经过多次改进, 并开始流行。目前, 国际上标准的 C 是 87ANSI C, 常用的有 Microsoft C、Turbo C、Quick C 等。不同版本略有不同, 但基本的部分是兼容的。

C 语言主要含有如下特点。

1. C 语言是一种结构化的程序设计语言, 语言简洁, 使用灵活方便。它既适用于设计和编写大的系统程序, 又适用于编写小的控制程序, 同样也适用于科学计算。

2. 它既有高级语言的特点, 又具有汇编语言的特点。其运算符丰富, 除了能提供对数据的算术逻辑运算外, 还提供了二进制的位运算, 并且提供了灵活的数据结构。用 C 语言编写的程序表述灵活方便, 功能强大。用 C 语言开发的程序, 其结构性好, 目标程序质量高, 程序执行效率高。

3. 程序的可移植性好。在某一种型号的计算机上开发的用 C 语言编写的程序, 基本上可以不作修改, 而可直接移植到其他型号和不同档次的计算机上运行。

4. 程序的语法结构不够严密, 程序设计的自由度大。这对于比较精通 C 语言的程序设计者来说, 可以更方便地设计出高质量的非常通用的程序。但对于初学者来说, 要能比较熟练地运用 C 语言来编写程序, 并不是一件易事。与其他高级程序设计语言相比, 用 C 语言编写的程序调试较困难。往往是程序编好并输入计算机后, 编译时易通过, 而在执行时还会出错。但只要对 C 语言的语法规则真正领会, 编写程序及调试程序还是比较容易掌握的。

随着 C 语言应用的推广, C 语言存在的一些缺陷或不足也开始流露出来, 并受到大家的关注。比如: C 语言对数据类型检查的机制比较弱; 缺少支持代码重用的结构; 随着软件工工程规模的扩大, 难以适应开发特大型的程序等。

为克服 C 语言本身存在的缺点, 并保持 C 语言的简洁、高效, 且与汇编语言接近的特点, 1980 年贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充, 并把 Simula 67 中类的概念引入到 C 中。于 1983 年由 Rick Maseitti 提议正式命名为 C++ (C Plus Plus)。后来又把运算符的重载、引用、虚函数等功能加入到 C++ 中, 使 C++ 的功能日趋完善。

当前运用得较为广泛的 C++ 有 VC++ (Visual C Plus Plus)、BC++ (Borland C Plus Plus)、AT&T C++ 等。

1.2 C++ 的特点

C++ 并不是对 C 语言的功能作简单的改进和扩充, 而是一种本质性革新。C++ 之所以能得到广泛的应用, 除了 C++ 继承了 C 语言的一些特点之外, 还具有以下几方面的特点。

1. C++ 是 C 语言的一个超集, 大多数的 C 程序代码略作修改或不作修改就可在 C++ 的集成环境下运行或调试。这对于继承和开发当前已在广泛使用的软件是非常重要的, 可节省大量的人力和物力。

2. C++ 是一种面向对象的程序设计语言。它使得程序的各个模块的独立性更强, 程序的可读性和可理解性更好, 程序代码的结构性更加合理。这对于设计和调试一些大的软件尤为重要。

3. 用 C++ 设计的程序扩充性强。一方面, 在软件开发的前期, 对整个要解决的问题很难全部弄清楚, 开发人员只能根据自己的理解进行程序的结构设计; 而到软件开发的后期, 开发人员往往发现开始的理解并不正确或并不全面, 这时可能需要改变程序的结构或功能, 这就要求开发工具具有较强的可扩充性。另一方面, 已开发的软件, 随着时间的推移, 也还要求扩充新的功能, 或要改进某些功能。C++ 所具有的扩充性能的特点, 对于编写一些大的程序而言是非常重要的。

1.3 简单的 C++ 程序介绍

本书主要介绍 Visual C++, 但 C++ 的基本内容都是相同的, 除作特殊说明的章节外, 适用于任何一种 C++ 语言。本书上机实习的环境为 Visual C++ 6.0。

C++ 的集成环境不仅支持 C++ 程序的编译和调试, 而且也支持 C 程序的编译和调试。通常, C++ 的集成环境约定: 当源程序文件的扩展名为 .C 时, 则为 C 程序; 而当文件的扩展名为 .CPP 时, 则为 C++ 程序。本书中, 所有例子程序中的文件扩展名均为 .CPP。

1.3.1 一个简单的 C++ 程序

下面首先通过一个简单的例子来说明 C++ 程序的基本结构及其主要特点。

例 1.1 一个简单的 C++ 程序。

//源程序文件名为 EX1_1.cpp

```

/* C++ 程序的基本结构 */
#include <iostream.h>

void main( void)
{
    cout<<"i=";           //显示提示符
    int i;                 //说明变量 i
    cin>>i;                //从键盘上输入变量 i 的值
    cout<<"i 的值为:"<<i<<"\n"; // 输出变量 i 的值
}

```

该程序经编译和连接后,运行可执行程序时,在显示器上显示:

i =

此时等待用户输入一个整数。设从键盘上输入 100 时,则显示器上显示:

i 的值为: 100

下面简单介绍程序的基本结构和各语句的作用。

1. 注解或说明信息。

在 C++ 程序的任何位置处都可以插入注解信息。注解方法有两种:第一种方法是用“/*”和“*/”把注解信息括起来,这种注解可以出现在程序中的任何位置,如例 1.1 中的第二行;第二种方法是用两个连续的“/”字符,它表示从此开始到本行结束为注解,如例 1.1 中的第一行。

2. 包含文件或编译预处理指令。

以“#”开头的行称为编译预处理指令,如例 1.1 中的第三行。有关包含文件或编译指令的具体用法及规则将在后面介绍。由于程序中要用到输入/输出函数,故要包含文件 `iostream.h`,这是一个标准输入/输出流的头文件。

3. 主函数 main()。

任一 C++ 程序均要有一个且只能有一个主函数。一个 C++ 程序总是从 `main( )` 函数开始执行,而不管该函数在整个程序中的具体位置。

4. 花括号对 {}。

{ } 称为函数或语句括号。任一函数体均以“{”开始,并以“}”结束。注意,花括号要配对使用。

5. 任一 C++ 程序均由一个或多个函数组成。其中必须有一个主函数 `main( )`,其余函数可有可无。其余函数包括库函数和用户自定义的函数。在本例中, `cin`、`cout` 是库函数。任一函数可由若干个语句组成,每一个语句均以“;”结束。

6. 程序的书写规则。

对于 C++ 的编译器而言,一个语句可以写成若干行,一行内也可以写若干个语句。虽然 C++ 允许的书写格式是非常自由的,但是为了便于程序的阅读和相互交流,程序的书写必须符合以下基本规则。

(1) 对齐规则。同一层次的语句必须从同一列开始,同一层次的开花括号必须与对应的闭花括号在同一列上。

(2) 缩进规则。属于内一层次的语句,必须缩进几个字符,通常缩进两个、四个或八个

字符的位置。

(3) 任一函数的定义均从第一列开始书写。

7. C++ 语言没有专门的输入/输出语句。输入和输出是通过函数(对象)来实现的。

8. 严格区分大小写字母。

在某些高级语言中,不区分大小写字母,但在 C++ 中,是严格区分大小写字母的。如 A 与 a 表示两个不同的标识符。在书写程序或编辑程序时,要注意这一点。

1.3.2 程序的基本要求

用 C++ 语言进行程序设计并解决实际问题时,对程序是有质量要求的。通过对本课程的学习,对设计的程序要达到以下几方面的基本要求。

1. 程序的正确性。要求程序正确无误,即语法和语义正确,算法描述正确。这是对程序的最基本的要求。

2. 程序的可读性和可理解性好。当设计的程序被阅读时,要做到容易读懂,并且能容易理解程序的设计思想和设计方法。通常包括三个方面:首先是程序的结构性好,采用结构化的程序设计方法或采用软件工程的程序设计方法来设计程序;第二是在程序中增加注解,说明程序设计的思想和方法;第三是程序的书写格式规范。

3. 程序的可维护性好。程序易于修改,易于增加新的功能。这要求程序的结构性好,各模块的独立性强。

4. 程序的构思好,程序简短,执行速度快。

前面三点是最基本的要求,最后一点要求略高。要做到该点,仅学习本课程的知识是不够的,还需掌握数据结构、算法设计与分析、软件工程及程序设计方法学等知识,再通过大量的程序设计实践,就能编出高质量的程序。

1.3.3 C++ 程序的开发步骤

C++ 语言是一种编译性的语言,设计好一个 C++ 源程序后,需要经过编译、连接,生成可执行的程序文件,然后执行并调试程序。一个 C++ 程序的开发步骤可分为如下 5 个步骤。

1. 根据要解决的问题,分析需求,并用合适的方法描述之。

2. 编写 C++ 源程序,并利用一个编辑器将源程序输入到计算机中的某一个文件中。文件的扩展名为 .CPP。

3. 编译源程序,并产生目标程序。在 PC 机上,文件的扩展名为 .OBJ。

4. 连接。将一个或多个目标程序与库函数进行连接后,产生一个可执行文件。在 PC 机上,文件的扩展名为 .EXE。

5. 调试程序。运行可执行文件,分析运行结果。若结果不正确,则要修改源程序,并重复以上过程,直到得到正确的结果为止。

1.4 面向对象的程序设计概述

面向对象的程序设计(Object Oriented Programming,简称 OOP)方法是近年来十分流行的一种程序设计方法,它试图用客观世界中描述事物的方法来描述一个程序要解决的事情。

在 C++ 中通过引入类和对象的概念,增加了程序模块的独立性和可扩展性。

1.4.1 对象及面向对象的程序设计

对象(Object)是 OOP 中最重要的概念之一。简单地说,对象是一个抽象的概念,它是对一个客观实体的描述。它是既有数据又有对数据进行操作的代码的一个逻辑实体。

由此,面向对象的程序设计就是用 OOP 来描述客观世界中所需说明的有关事物。

1.4.2 面向对象程序设计的要素

C++ 是一种面向对象的程序设计语言,它支持面向对象程序设计的几个要素(封装性、继承和派生性、重载性以及多态性)。下面对这些要素分别作简单介绍。

1. 封装性

将描述对象的数据及对这些数据进行处理程序代码有机地组成一个整体,形成一个模块,对其数据及代码的存取权限加以限制后,模块完全独立,对象的这种特性称为封装性。这样就使得描述对象的数据只能通过对象中的程序代码来处理,而其他任何程序代码均不能访问对象中的数据。这种封装性可通过定义类来实现,对象是类的一个实例。

这种特性非常有利于程序的调试和维护。

2. 继承和派生性

一个类可以派生出新的类(原来的类称为基类),派生类可以全部或部分地继承基类的数据或程序代码,这种特性称为继承和派生性。派生类又可以作为其他类的基类,而派生出新的类,这样一层一层地继承和派生下去,可形成一棵树状的类结构。

利用这种特性,对于类同的问题或只有部分类同的问题,都可以从已定义的类中派生出来,省去重复性的工作。其作用是:一方面可减少程序设计的错误(利用原来已调试好的类);另一方面,可加快和简化程序设计,提高工作效率。

3. 重载性

一个函数名或一个运算符,根据不同的对象可以完成不同的功能或运算,这种特性称为重载性。例如,“+”运算符可以完成两个整数的求和运算,也可以完成两个实数的求和运算,还可以完成两个复数的求和运算,当然也可以完成两个字符串的拼接。在 C++ 中,有两种重载:一种是运算符的重载;另一种是函数的重载,如用相同的函数名 `abs()`,可以分别求整数、实数和双精度实数的绝对值。

这种特性为编程提供了极大的方便。

4. 多态性

通过系统提供的机构,实现对象之间的信息传递。按一定格式传递的信息称为消息。同一个消息为不同的对象所接收时,可以导致完全不同的行为,这种特性称为多态性。多态性的重要性在于允许一个类体系的不同对象,各自以不同的方式响应同一个消息,这就可以实现“同一接口,多种方法”。

这种特性不仅提高了程序设计的灵活性,而且也大大减轻了类体系使用者的记忆负担。

1.5 VC++ 程序的上机过程

VC++ 为用户开发 C 或 C++ 程序提供了一个集成环境,这个集成环境包括:源程序的输入、编辑和修改,源程序的编译和连接,程序运行期间的调试与跟踪,项目的自动管理,为程序的开发提供工具,窗口管理,联机帮助等。由于这个集成环境功能齐全,但又比较复杂,要能熟练运用集成环境中的各种工具需要经过较长时间的上机实习和体会。本节仅介绍最简单的上机操作过程,有关上机操作较为详细的说明,请参看与之配套的《Visual C++ 实验指导书》(苏州大学出版社,2001 年)。

在 Windows 95 或 98 下启动 VC++ 的集成环境,则产生如图 1-1 所示的一个组合窗口。

窗口最上面部分为标题。第二部分为菜单条,其中包括“文件(File)”菜单、“编辑(Edit)”菜单等。第三部分为功能按钮,类同于 Word,有“打开”文件、“剪切”、“复制”、“粘贴”按钮等。另有三个子窗口,左边的子窗口为程序结构窗;右边的子窗口为源程序编辑窗,用于输入或编辑源程序;下边的子窗口为信息输出窗,用来显示出错信息或调试程序的信息。

当要建立一个新的源程序文件时,选择“File”菜单中的“New”命令,这时弹出一个子窗口,在子窗口中选择“Files”按钮,在弹出的信息窗口中再选择“C++ Source File”,这时光标进入源程序编辑子窗口,就可输入源程序了。编辑源程序的方法与 Word 基本相同。

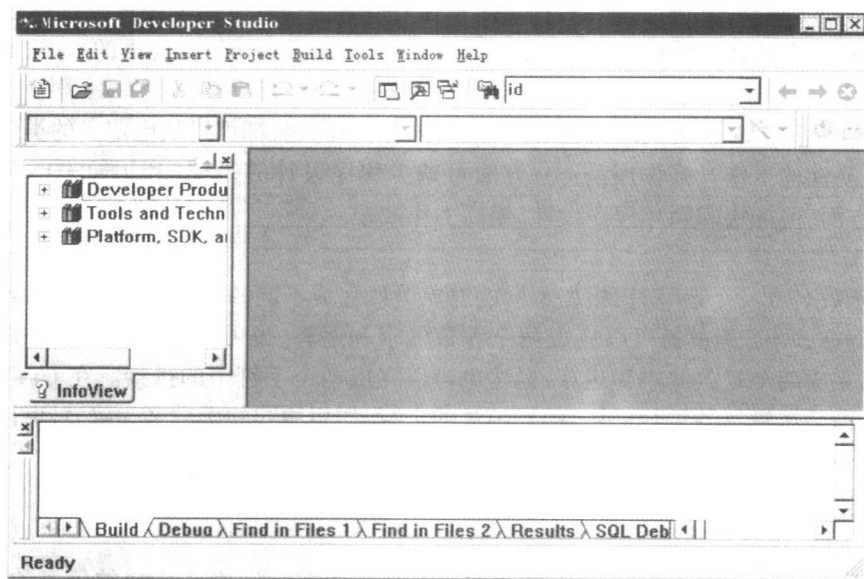


图 1-1 VC++ 集成环境

要打开一个已存在的源程序文件时,可选择“File”菜单中的“Open”命令,然后根据提示信息,选择相应的源程序文件名。由系统将指定的源程序文件调入源程序编辑子窗口,就可以对源程序文件进行编辑了。

当正确地输入源程序或编辑结束后,先存盘,然后可选择“Build”菜单中的“Build”命令来编译源程序和连接目标程序,最后选择“Build”菜单中的“Execute”命令来执行程序。

练 习 题

1. 程序的基本要求有哪些?
2. 面向对象程序设计有哪几个基本要素?
3. 在 VC++ 集成环境下,从输入源程序到得到正确的结果,要经过哪些步骤?
4. 将本章例题中的程序输入到源程序文件 EXAMPLE1.CPP 中,并在 VC++ 集成环境下编译、连接和运行。
5. 在 VC++ 中,有哪两种注解方法?每一种注解方法适用于什么场合?
6. 简要说明 C++ 程序开发的步骤。

数据类型、运算符和表达式

本章介绍 C++ 语言中保留字、标识符、常量、变量、基本的数据类型、基本运算符及其优先级、表达式及表达式的求值。

2.1 VC++ 的数据类型

本节介绍组成 C++ 程序的基本单位：关键字、标识符、标点符号及基本的数据类型。

2.1.1 关键字

在 C++ 语言中,已有特殊含义和用途的英文单词称为关键字(Keyword)或保留字。在程序中不得将它们另作它用。C++ 中共有 48 个关键字,其中有 5 个关键字不适用于 VC++。相反地,在 VC++ 中增加了 19 个专用的关键字,这些关键字也不适用于 C++。表 2-1 列出了 C++ 与 VC++ 兼容的 43 个关键字。

表 2-1 VC++ 与 C++ 兼容的 43 个关键字

关键字	类型	说明
auto	说明符	用于说明变量为自动类型
break	语句	用于循环语句或开关语句中,结束语句的执行
case	标号	用于开关语句中
char	类型说明符	用于说明字符类型的变量
class	类型说明符	用于说明类数据类型
const	类型说明符	用于说明常数类型变量
continue	语句	用于循环语句中,结束本次循环
default	标号	用于开关语句中,表示其他情况
delete	运算符	用于收回动态存储空间
do	语句	与 while 一起构成循环语句
double	类型说明符	用于说明双精度实型变量
else	语句	与 if 一起构成双向选择条件语句
enum	类型说明符	用于说明枚举类型数据
extern	说明符	用于说明外部类型变量或函数等
float	类型说明符	用于说明实型数据
for	语句	用于循环语句中
friend	访问说明符	用于说明友元成员
goto	语句	用于无条件转移语句中
if	语句	用于条件语句中
inline	说明符	用于说明内联函数