

信息技术与管理职业教育系列丛书

C++

程序设计

王 永 王 锋◎主编

东南大学出版社

内 容 提 要

本书依据C++语言的两个特性,从面向过程化的程序设计的角度,系统地讲述了C++语言的基本概念、语法基础、简单数据类型和构造数据类型及应用、程序设计的基本方法,同时,针对C++语言面向对象程序设计的特点,对类和对象、继承性和多态性以及输入与输出流等知识进行了系统的阐述。

本书在编写过程中力求语言简练,通俗易懂。同时,本书通过大量的例题进行讲解,重点培养学生利用C++语言分析问题和解决问题的能力。本书中所有的实例都是 Visual C++ 6.0 集成开发环境下编辑、编译、调试通过的。为了加深对基础知识的学习,本书还按章节编写了对应的实验,读者在使用过程中可根据需要增加或减少相应的实验内容。

本书适合高职高专、大中专院校计算机及相关专业的学生作为“C++程序设计”教材及“程序设计语言”教材,同时,也可作为计算机爱好者及软件开发人员的参考用书。

图书在版编目(CIP)数据

C++程序设计/王永,王锋主编. —南京:东南大学出版社, 2007. 10

(信息技术与管理职业教育系列丛书)

ISBN 978-7-5641-0959-2

I. C… II. ①王… ②王… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 151602 号

C++ 程序设计

出版发行 东南大学出版社
社 址 南京市四牌楼 2 号(邮编 210096)
出版人 江 汉
经 销 江苏省新华书店
印 刷 南京玉河印刷厂
开 本 787 mm×1 092 mm 1/16
印 张 19.75
字 数 456 千字
版 印 次 2007 年 10 月第 1 版第 1 次印刷
印 数 1—4000 册
书 号 ISBN 978-7-5641-0959-2/TP·155
定 价 32.00 元

* 东大版图书若有印装质量问题,请直接联系读者服务部,电话:(025)83792328。

《信息技术与管理职业教育系列丛书》

编委会

主任：慕东周 张祖鹰

编委：(按丛书的出版顺序排名)

张建平 尚维来 陈 涵 吴 方 邹治军

张祖鹰 曾 浩 安珊珊 王 永 王 锋

张雪松 林 茂 陈详章 慕东周 王文捷

张明金

前 言

随着计算机技术的不断发展,计算机的应用领域也在不断扩展。作为计算机的应用工具——程序设计语言也在不断发展与进步。C++语言既有结构化程序设计的特点,又有面向对象程序设计的特色,因此,受到越来越多程序员的关注。

本书系统地介绍了C++语言的基本概念和程序设计的基本方法,同时,针对C++语言面向对象程序设计的特点进行了阐述。本书在编写过程中力求语言简练、通俗易懂,同时,通过大量的例题进行讲解,重点培养学生利用C++语言分析问题和解决问题的能力。

本书中所有的实例全部都在 Visual C++ 6.0 集成开发环境下编辑、编译、调试通过的。为了加深对基础知识的学习,本书还按章节编写了对应的实验,读者在使用过程中可根据需要增加或减少相应的实验内容。

本书由王永、王锋主编,张雪松、林茂、陈详章任副主编。其中,第1章、第2章、第3章、第4章由王永编写,第5章、第6章、第7章由林茂编写,第8章、第9章、第10章由张雪松编写,第11章及附录由王锋编写,全书由王锋、陈详章负责审阅。

在本书编写过程中,徐州工业职业技术学院信息与电子工程系慕东周主任提出了许多宝贵建议,并对本书的编写提供了很大帮助,在此表示衷心感谢。

由于编者水平有限,书中难免会出现错误或不妥之处,恳请广大读者和专家在使用过程中提出宝贵的意见和建议。

编 者
2007年7月

目 录

82	第1章 C++概述	1
84	1.1 C++的发展史	1
11	1.1.1 C++的起源	1
84	1.1.2 C++语言的特点	2
84	1.2 C++程序的结构	3
86	1.2.1 C++的词法与规则	3
86	1.2.2 C++程序示例	3
86	1.2.3 C++程序的基本结构	7
102	1.3 C++程序的开发与实现过程	9
88	1.4 C++语言的开发环境	10
88	1.4.1 Visual C++简介	10
88	1.4.2 C++上机操作示例	12
88	本章小结	17
88	习题1	18
88	第2章 数据类型、运算符与表达式	19
88	2.1 C++语言字符集	19
11	2.2 词法符号	19
87	2.2.1 标识符	20
11	2.2.2 关键字	20
11	2.2.3 分隔符	21
87	2.3 数据类型	21
87	2.3.1 基本数据类型	23
87	2.3.2 构造数据类型	24
87	2.4 常量与变量	25
88	2.4.1 常量	25
	2.4.2 变量	29
88	2.5 运算符与表达式	31
88	2.5.1 运算符	31
88	2.5.2 表达式	33
118	2.5.3 算术运算符和算术表达式	33

2.5.4	赋值运算符和赋值表达式	38
2.5.5	关系运算符和关系表达式	40
2.5.6	逻辑运算符和逻辑表达式	41
2.5.7	条件运算符和条件表达式	42
2.5.8	逗号运算符和逗号表达式	43
2.5.9	位运算符	44
2.5.10	数据类型长度运算符	47
1	本章小结	48
1	习题 2	49
1	第 3 章 程序结构与流程控制语句	50
8	3.1 程序的基本结构	50
8	3.1.1 C++程序的语句	50
8	3.1.2 程序的基本结构	52
7	3.2 顺序结构程序设计	53
9	3.2.1 数据输入	54
01	3.2.2 数据输出	55
01	3.2.3 数据输入/输出格式控制	56
81	3.3 分支结构程序设计	57
71	3.3.1 if 语句	58
81	3.3.2 switch 语句	63
	3.4 循环结构程序设计	66
01	3.4.1 for 语句	67
01	3.4.2 while 语句	69
01	3.4.3 do...while 语句	71
08	3.4.4 循环语句的嵌套	72
08	3.5 流程控制语句及函数	74
18	3.5.1 break 语句	74
18	3.5.2 continue 语句	75
88	3.5.3 语句标号与 goto 语句	77
18	3.5.4 exit()和 abort()函数	78
88	本章小结	79
88	习题 3	80
88	第 4 章 数组	83
18	4.1 一维数组	83
88	4.1.1 一维数组的定义	83
88	4.1.2 一维数组的初始化	84

4.1.3	一维数组元素的访问	85
4.1.4	一维数组应用举例	86
4.1.5	数组越界	93
4.2	二维数组	93
4.2.1	二维数组的定义	94
4.2.2	二维数组的初始化	94
4.2.3	二维数组元素的访问	95
4.2.4	应用举例	96
4.3	字符数组	100
4.3.1	字符数组的定义及初始化	100
4.3.2	字符数组的输入与输出操作	101
4.3.3	字符串处理函数	102
4.3.4	字符数组的应用	105
	本章小结	107
	习题 4	108
第 5 章 函数		110
5.1	函数概述	110
5.2	函数的定义与声明	111
5.2.1	函数的定义及格式	111
5.2.2	函数的声明	112
5.3	函数的调用	113
5.3.1	函数的声明调用方式	113
5.3.2	函数的值和类型	116
5.3.3	函数的嵌套调用	117
5.3.4	函数的递归调用	119
5.3.5	函数的参数	120
5.4	局部变量与全局变量	122
5.4.1	局部变量	122
5.4.2	全局变量	124
5.5	内联函数	125
5.6	函数重载	126
5.6.1	参数类型不同的重载函数	127
5.6.2	参数个数不同的重载函数	127
	本章小结	129
	习题 5	129

第 6 章 指针	131
6.1 指针	131
6.1.1 指针的概念	132
6.1.2 指针变量的定义与引用	134
6.1.3 指针的运算	136
6.2 指针与数组	139
6.2.1 指针的运算	139
6.2.2 多维数组与指针	142
6.3 指针与字符串	144
6.3.1 字符串的指针表示	144
6.3.2 指针数组	146
6.4 指针与函数	147
6.4.1 指针变量做函数的参数	147
6.4.2 函数的入口地址和函数指针	148
6.4.3 返回指针的函数	151
6.5 new 和 delete 操作符	151
6.5.1 new 和 delete 操作符的使用方法	152
6.5.2 使用 new 和 delete 操作符的注意事项	153
本章小结	155
习题 6	155
第 7 章 结构体与共用体	156
7.1 结构体	156
7.1.1 结构体的定义	157
7.1.2 结构体变量成员的引用	159
7.1.3 结构体数组	161
7.2 结构体与函数	163
7.2.1 结构体变量做函数参数	163
7.2.2 返回结构体类型值的函数	164
7.3 共用体	166
7.3.1 共用体及其定义	166
7.3.2 共用体成员的访问	167
7.4 枚举类型	168
7.4.1 枚举及其定义	168
7.4.2 枚举类型举例	170
本章小结	171
习题 7	171

第 8 章 类与对象	173
8.1 从结构体到类	173
8.1.1 类对结构体的扩展	173
8.1.2 类的定义	175
8.2 对象	179
8.2.1 对象的定义和使用	180
8.2.2 构造函数	183
8.2.3 析构函数	186
8.2.4 常数据成员与常成员函数	187
8.3 对象指针	189
8.3.1 一般对象指针	189
8.3.2 this 指针	191
8.4 静态成员	193
8.4.1 静态数据成员	193
8.4.2 静态成员函数	195
8.5 友元函数	197
8.6 类与对象编程实例	199
本章小结	209
习题 8	210
第 9 章 继承与多态	213
9.1 C++ 中的继承	214
9.1.1 单继承	214
9.1.2 多继承	224
9.2 多态性	227
9.2.1 静态多态性	228
9.2.2 动态多态性	232
9.3 综合实例	242
9.3.1 任务	243
9.3.2 salary.h 文件的源代码	243
9.3.3 salary.cpp 文件的源代码	245
9.3.4 manage.cpp 文件的源代码	250
本章小结	252
习题 9	252
第 10 章 C++ 常用的输入输出	258
10.1 流类库及其结构	259
10.2 键盘输入	261

10.2.1	使用预定义的提取符	261
10.2.2	使用成员函数 get()	262
10.2.3	使用成员函数 getline()	263
10.2.4	使用成员函数 read()	264
10.3	屏幕输出	265
10.3.1	使用预定义的插入符	265
10.3.2	使用成员函数 put()	266
10.3.3	使用成员函数 write()	267
10.4	字符串流	267
10.4.1	ostream 类的构造函数	268
10.4.2	istream 类的构造函数	269
10.5	磁盘文件的输入和输出	271
10.5.1	磁盘文件的打开和关闭操作	271
10.5.2	文本文件的读写操作	274
10.5.3	二进制文件的操作	277
	本章小结	279
	习题 10	279
第 11 章 实验指导		
实验 1	熟悉 Visual C++ 6.0 集成开发环境	281
实验 2	常量与变量、运算符与表达式	283
实验 3	程序结构和流程控制语句(1)	285
实验 4	程序结构和流程控制语句(2)	286
实验 5	数组及其应用	287
实验 6	函数的应用	289
实验 7	指针的应用	291
实验 8	结构体与共用体的应用	292
实验 9	类与对象	293
实验 10	继承与多态	294
实验 11	熟悉 C++ 常用的输入输出	295
	附录	297

第1章 C++概述

技能目标

- 了解C++语言的发展史
- 理解C++语言的特点
- 掌握C++程序的基本结构,会编写简单的C++程序
- 理解C++程序的开发与实现过程
- 掌握利用 Visual C++ 6.0 集成开发环境编译C++程序

C++语言是目前使用最为广泛的一种高效程序设计语言,它既可以进行过程化程序设计,也可以用于面向对象的程序设计。C++语言是由C语言发展而来的一种新的支持面向对象的语言,C++语言完全兼容C语言,不仅具有C语言灵活高效、功能强大的可移植性好等优点,而且由于引入面向对象的程序设计思想与机制,可以更大程度地提高编程能力,减少软件维护的开销,增强软件的可扩展性和可重用性。

1.1 C++的发展史

1.1.1 C++的起源

C++语言是在C语言的基础上逐步发展和完善起来的,而C语言是一种实用性很强的语言,因此,首先回顾一下C语言的发展,通过它来了解C++语言。

C语言的历史可以追溯到1969年。在1969年,美国贝尔实验室的Ken Thompson为DEC PDP-7计算机设计了一个操作系统软件,这就是最早的UNIX。接着,他又根据剑桥大学的Martin Richards设计的BCPL(Basic Combined Programming Language)语言在继承BCPL语言的许多优点的基础上为UNIX设计了一种便于编写系统软件的语言

言,命名为 B 语言。B 语言是一种无类型的语言,直接对机器字操作,这一点和后来的 C 语言有很大不同。作为系统软件编程语言的第一个应用,Ken Thompson 使用 B 语言重写了其自身的解释程序。1972~1973 年间,同在贝尔实验室的 Dennis Ritchie 改造了 B 语言,在 B 语言的基础上作了进一步的充实和完善,为其添加了数据类型的概念,并将原来的解释程序改写为可以直接生成机器代码的编译程序,然后将其命名为 C 语言。1973 年,Ken Thompson 小组在 DEC PDP-11 计算机上用 C 语言重新改写了 UNIX 的内核。与此同时,C 语言的编译程序也被移植到 IBM 360/370、VAX-11/780 等多种计算机上,迅速成为应用最广泛的系统程序设计语言。

C 语言具有许多优点,比如语言简洁灵活、运算符和数据结构丰富、具有结构化控制语句、程序执行效率高,同时具有高级语言和汇编语言的优点等。与其他高级语言相比,C 语言具有可以直接访问物理地址的优点,与汇编语言相比又具有良好的可读性和可移植性。因此,C 语言得到了极为广泛的应用。然而,C 语言也存在一些缺陷,例如类型检查机制相对较弱、缺少支持代码重用的语言结构等,造成了用 C 语言开发大程序比较困难。

为了克服 C 语言存在的缺点,贝尔实验室的 Stroustrup 博士及其同事开始对 C 语言进行改进和扩充,将“类”的概念引入了 C 语言,最初的 C++ 被称为“带类的 C”,1983 年正式命名为 C++(C Plus Plus)。后来,Stroustrup 博士和他的同事们又为 C++ 引进了运算符重载、引用、虚函数等许多特性,并使之更加精炼,于 1989 年后推出了 AT&T C++ 2.0 版。随后美国国家标准化协会 ANSI(American National Standard Institute)和国际标准化组织 ISO(International Standards Organization)一起进行了标准化工作,并于 1998 年正式发布了 C++ 语言的国际标准,即 ISO/IEC:98-14882。各软件厂商推出的 C++ 编译器都支持该标准,并有不同程度的拓展。

C++ 支持面向对象的程序设计方法,特别适合于中型和大型的软件开发项目,从开发时间、费用到软件的重用性、可扩充性、可维护性和可靠性等方面,C++ 均具有很大的优越性。同时,C++ 又是 C 语言的一个超集,这就使得许多 C 代码不经修改就可被 C++ 编译通过。

1.1.2 C++ 语言的特点

C++ 语言的特点主要表现在以下几个方面:

(1) 全面兼容 C 语言。

C++ 语言是在 C 语言的基础上扩展而成的,所以两种语言的基本语法和语义是相同的。C++ 是一个更好的 C,它保持了 C 语言的优点,大多数的 C 程序代码略做修改或不做修改就可在 C++ 的集成环境下调试和运行,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,可以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。C++ 中加入了面向对象程序设计(OOP)的基本特征。包括:

① 封装性:把一个数据结构同操作的函数(行为或方法)组合在一起。封装性是借助

于一种新的结构和数据类型机制——类实现的。

② 继承性:建立一个新的派生类,它从一个或多个先前定义的基类中继承函数和数据,而且可能重新定义或加进新的数据行为,这样就建立了类的层次。

③ 多态性:给行为取一个名字或符号,它共享一个类的层次,在这个层次中的每个类都以适合自己的方式实现这个行为。

(3) C++语言使得程序的各个模块的独立性更强,程序的可读性和可移植性更强,程序代码的结构更加合理,程序的扩充性更强,这对于设计、编制和调试一些大型的软件尤为重要。

(4) C++编写的程序代码可读性好,代码结构合理,对语法的限制较为宽松。

1.2 C++程序的结构

学习程序设计语言,首先应了解程序的基本结构。本节先从最简单的C++程序开始,接触C++的最基本结构,在以后各章节中再逐步深入学习C++的完整程序结构。

1.2.1 C++的词法与规则

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。一个程序只有严格按照语言规定的语法和表达方式编写,才能保证编写的程序在计算机中能正确地执行,同时也便于阅读和理解。C++是一门计算机语言,它规定了书写程序的语法。例如:如何使用变量、常量,如何撰写语句,如何让计算机作出判断,如何让某些特定的功能反复执行若干次(循环),如何把一些复杂的语句“打包”成一个简单的模块(函数/子程序)以便调用等等。详细的C++的词法与规则将在第2章中学习。

1.2.2 C++程序示例

为了更好地理解程序的基本结构,现在通过编制三个简单的程序示例来学习C++程序。

注:C++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++集成环境约定:当源程序文件的扩展名为.c时,则为C程序;而当源程序文件的扩展名为.cpp时,则为C++程序。因此,在本书中,所有例题程序的文件扩展名均为.cpp。

【例 1.1】编写程序,在屏幕上显示输出一个字符串“Hello world!”。文件名为 example1_1.cpp。

//example1_1.cpp:文本输出程序

```
#include <iostream.h>

void main()
{
    cout<<"Hello world!"<<endl;

    Hello world!
}
```

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:

Hello world!

显示结果如图 1.1 所示。

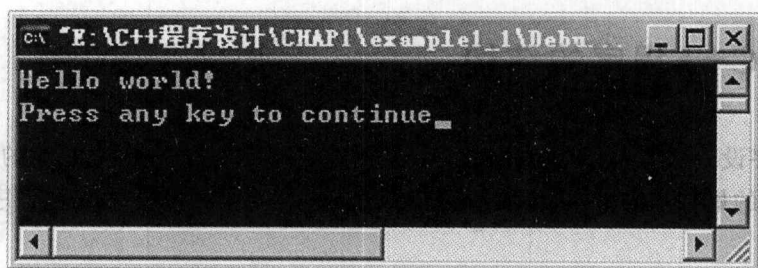


图 1.1 【例 1.1】程序运行结果

下面来分析程序 example1_1.cpp 的结构:

程序的第一行为: `//example1_1.cpp: 文本输出程序`。以“//”开头的是注释,注释是对程序的说明,用来提高程序的可读性,可以放在程序的任何位置,对程序的编译和运行不起作用。程序中的第五行也起到和第一行类似的注释效果。

程序的第二行为:

```
#include <iostream.h>
```

程序中的“#include”是 C++ 编译预处理中的文件包含命令,它用来告诉编译器,在当前程序编译时还要包括其他文件,这里包括的文件是“iostream.h”,文件名可以用双引号或尖括号括起来表示此文件处在一个特定的目录中。“iostream.h”是头文件,为了使用输出流 cout 和输入流 cin,程序开头必须用“#include”命令将文件“iostream.h”中的内容包含到本文件中来。本行也可写成:

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

程序的第三行为:

```
void main()
```

此行表示一个函数,“void”表示该函数没有返回值,main()表示主函数。每个 C++ 程序必须有且只能有一个主函数, C++ 程序总是从主函数开始执行的。通常用花括号括起来表示该函数的全部内容。在花括号内的部分是函数体,函数体由语句组成,每个语句以分号结束。格式如下:

```
void main()
{
    //在此位置写入程序体语句,每个语句以分号结束
}
```

程序的第六行为:

```
cout<<"Hello world!"<<endl;
```

“cout”表示是C++中的一个输出流,用来向标准输出设备输出,符号“<<”是输出操作符,用来输出紧随其后的元素。“cout”与符号“<<”结合使用可以输出常量、变量的值及原样输出双引号中的字符串。“endl”是输出换行符,功能是输出完本行后换下一行开始输出。

下面再举两例以加深对C++程序结构的理解与学习。

【例 1.2】 编写一个程序,求两个整数的和。文件名为 example1_2.cpp。

```
/* 求两个整数的和程序 */
#include <iostream.h>

void main()
{
    int x,y,sum;                //说明变量 x,y,sum 为整型数
    cout<<"请输入第一个整数:"<<endl; //显示提示信息
    cin>>x;                      //从键盘上输入变量 x 的值
    cout<<"请输入第二个整数:"<<endl; //显示提示信息
    cin>>y;                      //从键盘上输入变量 y 的值
    sum=x+y;                    //求和
    cout<<"两数和为:"<<sum<<"\n"; //输出结果
}
```

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:

请输入第一个整数:

5

请输入第二个整数:

4

两数和为:9

显示结果如图 1.2 所示。

程序 example1_2.cpp 的结构分析如下:

在“/*”和“*/”之间的部分也表示注释,功能等同于“//”,但此注释语句称为多行注释语句,“/*”和“*/”之间的部分可以是一行,也可以是多行,而“//”只注释单行语句。

程序中的语句:“int x,y,sum;”用来说明变量 x、y、sum 为 int(整型)变量。
 程序中的语句:“sum=x+y;”是一个赋值语句,表示将 x 和 y 的值相加,其结果输入变量 sum。
 程序中的语句:“cout<<”两数和为:“<<sum<<”\n”;”中“\n”是换行符,功能相当于“endl”。

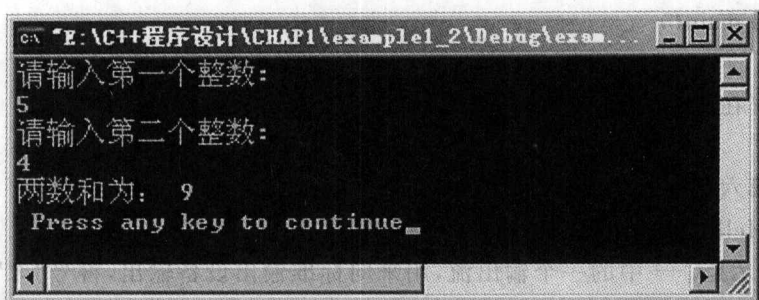


图 1.2 【例 1.2】程序运行结果

【例 1.3】 编写一程序,输入两个整数 a 和 b ,输出其中较大的一个数。文件名为 `example1_3.cpp`。

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
    int max(int x,int y);
```

```
    int a,b,m;
```

```
    cout<<"请输入两个数 a,b:";
```

```
    cin>>a>>b;
```

```
    m=max(a,b);
```

```
    cout<<"最大数为:"<<m<<endl;
```

```
}
```

```
int max(int x,int y)
```

```
{
```

```
    int z;
```

```
    if (x>y)
```

```
        z=x;
```

```
    else
```

```
        z=y;
```

```
    return(z);
```

```
}
```

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:

请输入两个数 a,b:3 5
最大数为:5

显示结果如图 1.3 所示。

该程序由两个函数组成:主函数 `main()` 和被调用函数 `max()`。函数 `max()` 的作用是找出 x 和 y 中的较大者,并通过 `return` 语句返回给主函数。主函数用来输入两个变量 a 和 b 的值,调用 `max()` 函数找出其中的较大者,并输出结果。此程序中用到的语句与程序结构形式在后面章节中将会学习,读者可在此先理解程序结构的组成。