

21世纪高等学校计算机教育实用规划教材

# C++程序设计教程 (第二版)

朱红 赵琦 王庆宝 编著



清华大学出版社



21世纪高等学校计算机教育实用规划教材

# C++程序设计教程 (第二版)

朱红 赵琦 王庆宝 编著

清华大学出版社  
北京

## 内 容 简 介

C++是一种高效、实用的程序设计语言,它既可以进行过程化程序设计,也可以进行面向对象程序设计,是编程人员最广泛使用的工具。本书是为高等院校程序设计课程编写的教材,全书共分为两个部分,第一部分是C++语言基础,包括第1~8章,主要讲述C++的数据类型、运算规则,顺序、选择、循环结构的程序设计,以及数组、函数、指针、结构体等内容,侧重于基本概念、基本语法及常规算法;第二部分是面向对象程序设计基础,包括第9~16章,主要介绍类和对象、类的构造和析构、友元、运算符重载、继承和派生及输入输出流等内容,侧重于对编程的训练,强调模块化、规范化的程序设计,引导读者逐步体会面向对象程序设计的特点。本书有配套的教学参考书《C++程序设计上机实践与学习辅导》,以方便读者复习和上机操作。

本书所列举的例题、习题均在VC++ 6.0下调试通过。

本书面向计算机及相关专业本、专科学生,是学习C++语言的基础教材。

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

C++程序设计教程/朱红等编著.--2版.--北京:清华大学出版社,2016  
21世纪高等学校计算机教育实用规划教材  
ISBN 978-7-302-42283-9

I. ①C… II. ①朱… 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283736号

责任编辑:闫红梅 王冰飞

封面设计:常雪影

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.75 字 数:653千字

版 次:2009年1月第1版 2016年9月第2版 印 次:2016年9月第1次印刷

印 数:1~3000

定 价:49.50元

# 出版说明

---

随着我国高等教育规模的扩大以及产业结构调整的不断深入,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和帮助下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

---

C++语言是从C语言继承、发展而来的一种高效、实用的程序设计语言,一方面C++语言全面兼容C语言,另一方面C++支持面向对象的方法,实现了类的封装、数据隐藏、继承及多态性,使得代码容易维护且高度可重用。

本书作为C++语言的入门教材,不仅详细介绍了C++语言本身,还深入地讲述了面向对象的程序设计方法。本书的主要特色如下:

(1) 强调基本概念、基本语法、基本结构,不深究语法的细节,从宏观上把握程序的结构。

C++的许多概念(如函数、指针、类等)是掌握C++语言的重要基础。本书在对概念的讲解上注重强调这些概念在编程中的作用及其所实现的功能,而不去罗列一些具体的语法细节和特例。

(2) 注重模块化的程序设计,注重模仿,强调规范化的程序结构,不提倡过多编程技巧和个人风格。

发明C++语言的初衷是为了应对软件危机,解决开发大型软件时遇到的问题,提高软件的开发效率。虽然本书的读者对象是没有编程基础的初学者,通过本书所能接触到的也只是相对简单的程序,但程序结构的设计和编程习惯的培养却正是从初学时开始形成的,因此本书所涉及的概念、算法、语法包括例题的讲解都强调规范化、结构化,引导读者适当地模仿,从基本程序的学习开始就养成规范编程的习惯。

(3) 通俗易懂。

众所周知,C++语言概念众多、叙述复杂、语法灵活,很难用浅显、直白的语言去诠释这些内容。本书利用了大量的图示说明,把复杂的概念、算法用图形的形式描述出来,使读者有一个形象、直观的认识。

全书共分两个部分。第一部分是C++语言基础,共有8章。第1章C++基础知识介绍C++的发展历史、C++程序的基本要素和C++程序的开发步骤;第2章基本数据类型与表达式介绍C++语言的基本数据类型、运算符与表达式以及数据类型转换,简单输入/输出语句等;第3章基本流程控制结构介绍结构化程序的设计方法,以及与3种控制结构——顺序、选择和循环结构相关联的语法知识及其控制语句;第4章函数介绍函数的定义与声明,函数的调用,内联函数、重载函数以及函数的作用域等内容;第5章编译预处理介绍编译预处理的知识以及3种预处理指令——宏、文件包含和条件编译;第6章数组介绍一维数组和

二维数组的定义与使用,数组名作为函数参数的应用,字符数组与字符串的应用;第7章指针介绍指针与指针变量的概念,以及指针运算、指针数组、函数指针、指针的指针等;第8章结构体和共用体介绍了C++的构造数据类型,有结构体、共用体和枚举类型,对单向链表的各种操作也做了详细的说明。第二部分是面向对象程序设计基础,共有8章。第9章类和对象介绍面向对象程序设计的基本要素、类和对象的定义方法、对象的初始化、this指针等;第10章构造函数和析构函数介绍默认的构造函数、构造函数的重载、复制构造函数和显式定义的析构函数;第11章静态成员与友元介绍了静态数据成员和静态成员函数、友元函数和友元类;第12章运算符重载介绍单目与双目运算符的重载,包括重载成为成员函数或友元函数在定义格式及应用中的区别;第13章继承和派生介绍基类和派生类,单继承、多继承和虚基类,继承中冲突的解决和支配规则等;第14章虚函数介绍了虚函数与运行时的多态性的概念,包括虚函数的作用和功能、虚函数的应用、纯虚函数和抽象类等;第15章输入输出流介绍ios类、键盘输入和屏幕输出、磁盘文件的输入输出等;第16章C++工具介绍模板的概念和异常处理,该章内容是选学内容。

本书所列举的例题、习题均在VC++ 6.0下调试通过。

本书有配套的教学参考书《C++程序设计上机实践与学习辅导》,内有16个单元的上机实验内容,与书中的章节相对应,同时针对学习中的难点补充了大量的例题讲解和各种典型的习题,并有6套模拟试卷及习题解答。

在本书的编写过程中,编者参阅了许多C++的参考书和有关资料,谨向这些书的作者表示衷心的感谢!

本书由朱红、赵琦、王庆宝编著,在本书的编写过程中,闫玉德、王芳、钱芸生、陈文建、朱近、刘明、刘永、张微、俞虹、蔡骅、靳从等老师对本书的内容及修订提出了很多宝贵意见,在此一并表示衷心的感谢。

由于作者水平有限,书中难免有错误之处,恳请读者批评指正。

编 者

2016年5月

# 目 录

---

<b>第 1 章 C++ 基础知识</b> .....	1
1.1 C++ 与 Visual C++ 语言 .....	1
1.2 C++ 程序简介 .....	2
1.2.1 一个简单的程序 .....	2
1.2.2 C++ 程序的基本要素 .....	2
1.2.3 C++ 程序的书写规则 .....	3
1.3 C++ 程序的开发步骤与上机实践 .....	4
1.3.1 C++ 程序的开发步骤 .....	4
1.3.2 C++ 程序的上机实践 .....	4
1.4 信息在计算机中的表示 .....	5
1.4.1 进位记数制 .....	5
1.4.2 带符号数在计算机中的表示 .....	9
1.4.3 字符在计算机中的表示 .....	13
练习题 .....	14
<b>第 2 章 基本数据类型与表达式</b> .....	16
2.1 数据类型 .....	16
2.2 常量和变量 .....	17
2.2.1 常量 .....	17
2.2.2 变量 .....	18
2.3 整型数据 .....	20
2.3.1 整型常量 .....	20
2.3.2 整型变量 .....	21
2.4 浮点型数据 .....	23
2.4.1 浮点型常量 .....	23
2.4.2 浮点型变量 .....	23
2.5 字符型数据 .....	24
2.5.1 字符型常量 .....	24
2.5.2 字符型变量 .....	25
2.5.3 字符串常量 .....	27



2.6	类型转换	27
2.6.1	不同类型数据间的混合算术运算	27
2.6.2	赋值时的类型转换	28
2.6.3	强制类型转换符	31
2.7	运算符与表达式	32
2.7.1	算术运算符与算术表达式	32
2.7.2	关系运算符与关系表达式	32
2.7.3	逻辑运算符与逻辑表达式	33
2.7.4	赋值运算符与赋值表达式	35
2.7.5	自增运算符与自减运算符	36
2.7.6	逗号运算符与逗号表达式	38
2.7.7	sizeof()运算符及表达式	38
2.7.8	表达式中运算符的运算顺序	38
2.8	简单的输入/输出语句	39
2.8.1	十进制整数、实数与字符数据的输入	40
2.8.2	十六进制或八进制数据的输入	42
2.8.3	数据的输出	43
	练习题	44
<b>第3章</b>	<b>基本流程控制结构</b>	<b>47</b>
3.1	结构化程序设计基础	47
3.1.1	结构化程序设计	47
3.1.2	基本控制结构	48
3.1.3	C++语言中语句的分类	49
3.2	选择结构语句	50
3.2.1	if语句	50
3.2.2	条件运算符“?:”	55
3.2.3	switch语句	57
3.3	循环结构语句	60
3.3.1	while语句	60
3.3.2	do...while语句	62
3.3.3	for语句	64
3.3.4	3种循环的比较及适用场合	66
3.3.5	多重循环	67
3.4	控制执行顺序的语句	68
3.4.1	break语句	68
3.4.2	continue语句	70
*3.4.3	goto语句	71
*3.4.4	exit()和abort()函数	71

3.5 综合应用举例 .....	72
练习题 .....	75
<b>第4章 函数 .....</b>	<b>84</b>
4.1 函数概述 .....	84
4.2 函数的定义与调用 .....	85
4.2.1 函数的定义 .....	85
4.2.2 函数的调用 .....	88
4.2.3 引用作为函数参数 .....	92
4.3 函数的嵌套调用 .....	93
4.4 函数的递归调用 .....	94
4.5 内联函数 .....	97
4.6 重载函数 .....	98
4.7 带有默认参数的函数 .....	99
4.8 局部变量和全局变量 .....	100
4.8.1 局部变量 .....	101
4.8.2 全局变量 .....	102
4.9 变量的存储类别 .....	105
4.9.1 变量的生存期和存储类别 .....	105
4.9.2 auto 型变量 .....	105
4.9.3 static 局部变量 .....	106
4.9.4 register 型变量 .....	107
4.9.5 extern 型变量 .....	108
4.9.6 用 static 声明全局变量 .....	109
4.10 内部函数和外部函数 .....	110
4.10.1 内部函数 .....	110
4.10.2 外部函数 .....	111
练习题 .....	111
<b>第5章 编译预处理 .....</b>	<b>121</b>
5.1 宏定义 .....	121
5.1.1 不带参数的宏定义 .....	121
5.1.2 带参数的宏定义 .....	123
5.2 文件包含 .....	125
5.3 条件编译 .....	126
练习题 .....	128
<b>第6章 数组 .....</b>	<b>133</b>
6.1 数组的定义和引用 .....	133

6.1.1	一维数组的定义 .....	133
6.1.2	一维数组的引用 .....	134
6.1.3	一维数组的初始化 .....	135
6.1.4	一维数组程序举例 .....	136
6.2	二维数组的定义和引用 .....	139
6.2.1	二维数组的定义 .....	139
6.2.2	二维数组的引用 .....	139
6.2.3	二维数组的初始化 .....	139
6.2.4	二维数组程序举例 .....	140
6.3	数组作为函数的参数 .....	142
6.3.1	数组元素作为函数的实参 .....	142
6.3.2	数组名作为函数的实参 .....	143
6.3.3	二维数组作为函数的参数 .....	147
6.4	字符数组 .....	148
6.4.1	字符数组的定义 .....	148
6.4.2	字符数组的初始化 .....	149
6.4.3	字符串和字符串结束标志 .....	150
6.4.4	字符数组的输入/输出 .....	150
6.4.5	字符串处理函数 .....	152
6.4.6	字符数组的应用举例 .....	156
	练习题 .....	161
<b>第7章</b>	<b>指针</b> .....	<b>172</b>
7.1	指针与地址 .....	172
7.1.1	指针的概念 .....	172
7.1.2	指针变量的定义 .....	173
7.1.3	指针变量作为函数参数 .....	176
7.2	指针与一维数组 .....	177
7.2.1	通过指针引用一维数组中的元素 .....	177
7.2.2	通过指针在函数间传递一维数组 .....	179
7.3	字符指针与字符串 .....	180
7.3.1	字符数组与字符指针 .....	180
7.3.2	字符串操作的特点 .....	181
7.4	指针与函数 .....	183
7.4.1	函数指针变量 .....	183
7.4.2	指针型函数 .....	186
7.5	指针与二维数组 .....	187
7.5.1	二维数组的地址 .....	187
7.5.2	通过指针引用二维数组中的元素 .....	190

7.6	指针数组与指向指针的指针 .....	192
7.6.1	指针数组 .....	192
7.6.2	指针的指针 .....	194
7.6.3	多级指针小结 .....	196
7.7	动态存储分配 .....	197
	练习题 .....	199
<b>第8章</b>	<b>结构体和共用体 .....</b>	<b>210</b>
8.1	结构体 .....	210
8.1.1	结构体与结构体类型的声明 .....	210
8.1.2	结构体类型变量 .....	211
8.1.3	结构体数组和指针 .....	214
8.2	链表 .....	218
8.2.1	正向链表 .....	218
8.2.2	建立反向链表 .....	229
8.3	共用体 .....	235
8.3.1	共用体类型与共用体变量 .....	235
8.3.2	共用体变量的引用 .....	236
8.4	枚举 .....	237
8.5	用 typedef 声明类型 .....	239
	练习题 .....	240
<b>第9章</b>	<b>类和对象 .....</b>	<b>247</b>
9.1	面向对象程序设计概述 .....	247
9.1.1	面向对象 .....	247
9.1.2	面向对象中的主要概念 .....	248
9.1.3	面向对象的程序设计 .....	249
9.2	类的声明和对象的定义 .....	250
9.2.1	类的声明 .....	250
9.2.2	对象的定义 .....	252
9.2.3	对象成员的访问 .....	253
9.2.4	用类的指针访问对象成员 .....	255
9.2.5	用对象的引用变量访问对象成员 .....	256
9.3	成员函数 .....	256
9.3.1	重载类的成员函数 .....	256
9.3.2	默认参数的类的成员函数 .....	257
9.4	this 指针及在成员函数中访问成员函数 .....	258
9.4.1	this 指针 .....	258
9.4.2	在成员函数中访问成员函数 .....	261

9.5 类和对象的应用举例 .....	262
练习题 .....	266
<b>第 10 章 构造函数和析构函数 .....</b>	<b>271</b>
10.1 构造函数 .....	271
10.1.1 构造函数的作用 .....	271
10.1.2 构造函数的重载 .....	272
10.1.3 默认的构造函数 .....	273
10.2 析构函数 .....	275
10.3 构造对象的顺序 .....	278
10.4 对象的动态建立和释放 .....	279
10.5 复制构造函数 .....	280
10.6 临时对象 .....	285
10.7 面向对象的程序设计应用举例 .....	286
练习题 .....	292
<b>第 11 章 静态成员与友元 .....</b>	<b>301</b>
11.1 静态成员 .....	301
11.2 静态成员函数 .....	302
11.3 友元函数 .....	303
11.3.1 将普通函数声明为友元函数 .....	303
11.3.2 将另一类的成员函数声明为类的友元函数 .....	304
11.4 友元类 .....	306
练习题 .....	307
<b>第 12 章 运算符重载 .....</b>	<b>313</b>
12.1 运算符重载的概念 .....	313
12.2 将运算符重载为成员函数 .....	315
12.2.1 将双目运算符重载为成员函数 .....	315
12.2.2 将单目运算符重载为成员函数 .....	317
12.2.3 将赋值运算符重载为成员函数 .....	320
12.3 将运算符重载为友元函数 .....	326
12.3.1 将双目运算符重载为友元函数 .....	326
12.3.2 将单目运算符重载为友元函数 .....	328
12.4 类型转换运算符函数 .....	329
12.5 重载流插入和流提取运算符 .....	331
12.6 综合实例 .....	334
练习题 .....	337

<b>第 13 章 继承和派生</b> .....	343
13.1 继承与派生基础 .....	343
13.1.1 继承与派生的概念 .....	343
13.1.2 派生类的定义 .....	344
13.1.3 派生类的 3 种继承方式 .....	345
13.2 派生类的构造函数和析构函数 .....	349
13.2.1 单继承派生类的构造函数 .....	349
13.2.2 多继承派生类的构造函数 .....	350
13.2.3 有子对象的派生类的构造函数 .....	352
13.3 继承的冲突与支配 .....	354
13.3.1 冲突 .....	354
13.3.2 支配 .....	356
13.3.3 赋值兼容规则 .....	357
13.4 虚基类 .....	358
13.4.1 虚基类的定义 .....	358
13.4.2 虚基类的初始化 .....	359
练习题 .....	361
<b>第 14 章 虚函数</b> .....	367
14.1 多态性的概念 .....	367
14.2 虚函数简介 .....	367
14.3 纯虚函数与抽象类 .....	372
练习题 .....	376
<b>第 15 章 输入输出流</b> .....	380
15.1 C++ 的输入输出流 .....	380
15.1.1 ios 类的结构 .....	380
15.1.2 输入流 .....	381
15.1.3 输出流 .....	383
15.2 格式化输入与输出 .....	384
15.2.1 输入的格式化 .....	384
15.2.2 输出的格式化 .....	384
15.3 文件流 .....	386
15.3.1 文件的概念 .....	386
15.3.2 文件的打开与关闭 .....	387
15.3.3 对文本文件的操作 .....	389
15.3.4 对二进制文件的操作 .....	391
练习题 .....	394

<b>第 16 章 C++ 工具</b> .....	398
16.1 模板 .....	398
16.1.1 模板的概念 .....	398
16.1.2 函数模板 .....	399
16.1.3 类模板 .....	400
16.2 异常处理 .....	401
16.2.1 异常的概念 .....	401
16.2.2 异常处理的机制 .....	402
练习题 .....	405
<b>附录 A ASCII 字符集及其代码对照表</b> .....	407
<b>附录 B 库函数</b> .....	409
<b>参考文献</b> .....	413

## 1.1 C++ 与 Visual C++ 语言

计算机作为信息处理的重要工具正在影响和改变着人们的工作、学习和生活方式。使用计算机进行信息处理要通过相应的应用软件进行,如使用 Word 进行文本编辑、使用 MATLAB 进行科学计算等,这些软件均由专业软件开发人员设计编程。一般来说,在日常工作中遇到的任务大多数可借助现成的应用软件完成,但有时仍需为具体问题开发相应的软件。特别是要解决工程应用领域中遇到的大量具体问题,使用通用的软件不仅效率低,还可能无法完成任务,在这种情况下,自行编写具有针对性的相应软件可能是唯一的解决方法。

编写计算机软件需要使用程序设计语言,目前可用的程序设计语言很多,各有特点。随着 Windows 操作系统的崛起,由传统的面向控制台的字符软件开发转向面向窗口的可视化编程已成为必然趋势,Visual C++ 正是 Windows 环境下最强大、最流行的程序设计语言之一。

Visual C++ 语言是在 C 语言的基础上逐步发展和完善起来的。1969 年, 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 语言的基础上做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, 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 87ANSIC, 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。随着 C 语言应用的推广, C 语言表现出一些缺陷或不足。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, 后于 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 Visual C++ (Visual C Plus Plus)、BC++ (Borland C Plus Plus)、AT&TC++ 等。

Visual C++ 支持面向对象的程序设计方法 (Object-Oriented Programming, OOP), 支持 MFC (Microsoft Foundation Class) 类库编程, 有强大的集成开发环境 Developer Studio, 用来开发各种类型、不同规模和复杂程度的应用程序, 开发效率很高, 生成的应用软件代码品质优良, 这一切使得 Visual C++ 成为许多专业程序开发人员的首选。在本书中主要介绍 Visual C++ 语言的基础知识与使用 Visual C++ 开发应用程序的基本方法。



## 1.2 C++ 程序简介

2

本书主要介绍 Visual C++, 但 C++ 的基本内容都是相同的, 除做特殊说明的章节外, 适用于任何一种 C++ 语言。本书上机实践的环境为 Developer Studio Visual C++ 6.0。该集成开发环境不仅支持 C++ 程序的编译和调试, 而且支持 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; 而当源程序文件的扩展名为 .cpp 时, 则为 C++ 程序。对于使用 C++ 程序设计语言编程的初学者来说, 了解和掌握 C++ 程序的基本要素是必需的。下面通过一个简单的 C++ 程序介绍 C++ 程序的基本要素。

### 1.2.1 一个简单的程序

**【例 1.1】** 求两个整数  $a$  和  $b$  之和  $sum$ 。

```
//源程序文件名为 Ex1_1.cpp,用于计算两个整数之和
#include<iostream.h>
void main()

{
    int a,b,sum;           //定义变量
    cin>>a>>b;           //从键盘上输入变量 a,b 的值
    sum = a + b;          //计算 a + b,并将结果传给 sum
    cout<<"sum = "<<sum;  //输出变量 sum 的值,即将 sum 的值显示在显示屏上
    cout<<"\n";          //使光标在显示器上换行
}

```

该程序经编译、连接后,在运行可执行程序时从键盘输入两个整数,中间用空格分隔,如 12 \_18 ↵ (符号 \_ 表示空格,↵ 表示 Enter 键,下同),则显示器上显示:

```
sum = 30
```

该程序虽然短小,但包含了 C++ 程序的基本元素。

### 1.2.2 C++ 程序的基本要素

C++ 程序的基本结构如图 1.1 所示。

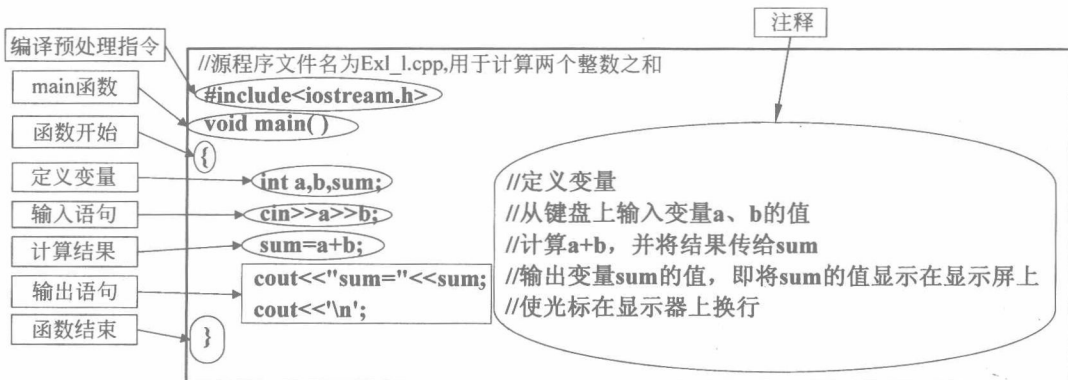


图 1.1 C++ 程序的基本结构