

教育部人才培养模式改革和开放教育试点教材

C++

面向对象程序设计

王 萍 编著 邓俊辉 审



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

教育部人才培养模式改革和开放教育试点教材

C++ 面向对象程序设计

王 萍 编著 邓俊辉 审

清华大学出版社

(京)新登字 158 号

内 容 简 介

本教材是针对中央广播电视大学计算机专业编写的。本书围绕 C++ 面向对象程序设计,内容覆盖基本概念和方法,基本数据结构和面向对象的概念、方法和技巧。全书共有 14 章,其中第 1 章至第 6 章简要介绍了 C++ 语言的发展历史、特点和程序风格,C++ 语言程序设计的基本结构、步骤、方法和技巧;第 7 章至第 8 章重点介绍了数组、结构等数据结构以及指针和引用的概念和方法;第 9 章至第 14 章详细介绍了面向对象程序设计的方法和特点。

本书是中央广播电视大学开放教育计算机科学与技术专业的教材,也可供其他大专院校计算机专业的本科生以及从事计算机软件开发的科研人员使用。

版权所有,翻印必究

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: C++ 面向对象程序设计

作 者: 王萍 编著 邓俊辉 审

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 清华大学印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 27.5 字数: 629 千字

版 次: 2002 年 2 月第 1 版 2002 年 7 月第 4 次印刷

书 号: ISBN 7-302-05159-3/TP · 3026

印 数: 14001 ~ 19000

定 价: 35.00 元

序

我们正处在跨越世纪的门槛上,人类社会在一股股变革性力量的推动下发生着根本性的变化。知识经济时代的到来向我们显示,一个国家最重要的资源已经不再是土地、劳动力或资本,而是其国民的知识和创造力;国与国的竞争虽然常常表现为政治、经济或军事实力的较量,但归根到底已是一场教育和科技的竞争。换言之,国家的综合实力将主要由其国民的教育水平来决定。一时间,世界各国的校长们、跨国企业的巨头们乃至许多的政府首脑们都在纷纷议论21世纪的教育,以迎接知识经济的挑战。我们中华民族有着蜿蜒几千年的文明,为在世界民族之林重振雄风,再展辉煌,发出了时代的特强音:实施科教兴国,提高全民素质。从中央领导到广大群众,都对教育提出了更高的要求,寄予了更大的希望,同时也给予了更多的支持。人们在这方面的思想观念和实践探索正在以空前的速度发展着。

中国的高等教育已经走完了一个世纪的路程。已经过去的20世纪正是它从无到有、从小到大、由产生到发展的一段百年历史。中国人民在短短的数十年时间里构筑了资本主义国家好几百年来形成的高等教育体系,涌现出一批高水平的学校,培养了一大批高层次优秀人才,取得了辉煌的成就。但是在新时期,教育不适应现代化建设需要的矛盾不断显露,我国劳动者受教育水平普遍较低的现象无法面对新世纪的机遇和挑战,我国高等教育的发展现状也难以满足广大人民群众空前强烈的受教育愿望。一代伟人邓小平早在十年前就一针见血地指出,我们的最大失误是教育,一是放松了对青少年的思想道德教育,二是教育规模发展不够快。现在看来,这两个问题依然是症结所在。一个十二亿人口的泱泱大国,高等学校的毛入学率仅10%左右,实在很不相称。我国的高等教育已经面临着大力发展、高速发展、从根本上改变落后状态的紧迫问题。

令人欣慰和鼓舞的是中国有一所全世界最大的大学——中国广播电视大学,上百万的学生遍布在九百六十万平方公里的辽阔土地上。它突破传统教育在空间上的限制,不断减弱时间上的束缚,以覆盖面广、全方位为各类社会成员提供教育服务的优势,成为中国高等教育体系中的一个重要组成部分。二十多年来,它为实现高等教育大众化,为提高我国劳动者的整体素质,为变巨大的人口包袱为巨大的人力资源,以形成浩浩荡荡的高水平建设大军,发挥了不可磨灭的作用。最近,中央电大又有重大改革举措,进一步面向社会开展了“开放教育”等项试点工作,在教育思想、招生对象、培养模式、管理机制方面进行新的探索。尤其引人注目的是中央电大与国内的一些重点高校形成了紧密的合作关系,携手为我国现代远程教育开拓新路。重点高校有学科和教学上的优势,它们的加盟有利于电大提高教学质量、办出特色;而中央电大有很丰富的教育资源,有完整的办学系统,有一支富有经验的教学与管理队伍,特别是有较强的社会服务意识和人才市场意识,这对于需要进一步向社会开放的普通高校而言,又有许多值得学习和借鉴之处。我们完全有理由相信,中央电大和重点高校的结合,不仅可以在现阶段实现优势互补、资源共享,而且

有可能成长出一种符合我国国情发展教育的最具潜力的新型教育模式。

现在摆在我们面前的这套中央广播电视大学本科(专科起点)“计算机科学与技术”专业教材,就是中央电大和清华大学合作的产物。在开放教育试点启动之际,在计算机及其网络技术日新月异、其爆炸式发展和神话般应用使人们眼花缭乱、不知所措之时,在我国至少缺乏数十万计算机软件及网络技术人才的当口,这套教材像雪里送炭,像清风送爽,终于在人们的企盼和惊喜中问世了。它确实及时和解渴。教材的编者是清华大学计算机系一批学术水平高、教学经验丰富的教授,他们以知识、能力和素质的全面训练为目标,将教材的先进性、实用性和可读性融为一体。教材纲目清楚,重点突出,深入浅出,便于自学。书中每章有小结,章章有习题,有的还配有实验指导和习题解答,不仅对计算机专业学生适用,其他专业的学生也可以从此入门。清华大学的老师们还准备为这套教材制作多媒体导读光盘和网络辅导教材,指明教学基本要求,区分应该熟练掌握和只需一般了解的内容,并进行重点难点分析和讲解。这全套的教材称得上是难得的好书。

对于中国广播电视大学我是颇有感情的,不只是因为它过去的功绩和带给人们未来的曙光,还因为我本人二十年前也曾参与过中央电大《电子技术基础》课程的教学工作。那时我收到许多电大学生热情洋溢的来信,强烈感受到他们对知识与教育的渴求,感受到他们学习的艰辛和坚韧不拔的毅力,同时也感受到了广大学生对我的信任和鼓励。当年的电大学生如今多数已成为我国经济建设和社会发展中的骨干,一些人后来获得了博士学位,有的已成为我国重点大学的教授。中央电大的成功实践已在社会上赢得了很好的声誉,而当前扩大教育规模、构建终身学习体系的社会呼唤又给电大今后的发展提供了新的难得的机遇。近年来,信息网络与多媒体技术突飞猛进,也使电大的远程教育形式跃上了现代化的新台阶。这次中央电大和清华大学合作,共同在计算机专业开放教育改革试点中付出了辛勤的劳动,播下了希望的种子。我期待着中央电大有更多的创新,更大的发展,更加充满活力。我也殷切希望电大的学生们为中华民族的强盛而自强不息,学有所成。

努力吧,中国广播电视大学一定能成为中国教育界一颗璀璨的明珠。

清华大学副校长、教授 **胡东成**

二〇〇〇年八月于北京

前 言

作为 21 世纪信息产业的一个重要支柱,成熟的软件产业被公认为是反映一个国家信息技术发展水平的标志。尽快培养出一个高级软件人才梯队,是振兴我国信息产业的当务之急。

面向对象的程序设计思想和方法,是近几年来软件设计、开发和维护技术的一次革命。作为这种新技术的典型代表,C++语言以其灵活性、高效性和高可复用性得到广泛的认可、推崇和应用。

《C++面向对象程序设计》是为中央广播电视大学面向对象程序设计课程专门编写的教材。本书是作者在清华大学计算机系多次使用的讲义的基础上,依据中央广播电视大学教学大纲的要求,经过整理和完善而成的。在内容的取舍上,本书在力求系统全面的同时,根据课时要求,突出和强调了基本概念、基本原理、基本方法和技巧,使学生能够尽快地对C++程序设计形成整体的认识,为以后的进一步学习打下坚实的基础。

本书围绕C++面向对象程序设计,覆盖基本概念和方法(第1章至第6章)、基本数据结构(第7章至第8章)和面向对象的概念、方法和技巧(第9章至第14章)。针对教学的特点,书中包含大量例题和课后思考题,并提供了答案,极大地方便了学生的阅读和自学。教师在讲授时,可以根据学生基础、学时安排等做适当取舍和编排。

与其他的同类教材相比,本书根据学生在接受、领会C++语言过程中的特点,注重通过大量生动实际的例子,深入浅出地讲解和分析复杂的概念,力图使读者通过实际动手,领会和掌握C++的精髓,获得良好的学习效果,并能在实践中学以致用,获得良好的学习效果,为今后的进一步学习和应用开发打下坚实的基础。

本书初稿由王萍独立编著完成,由清华大学计算机系博士、副教授邓俊辉初审,并整理、最终完稿。

在本书的编写过程中,我们参考了不少国内外同行的有关资料。由于计算机技术的迅速发展,对面向对象程序设计的最新成果难以全面论及;限于作者的学识水平,书中尚有很多不足和错误之处,恳请读者惠予批评、指正。

作 者

2001年10月25日于清华园

目 录

序	I
前言	III
第1章 C++ 编程基础	1
1.1 程序语言的发展	1
1.1.1 机器语言	1
1.1.2 汇编语言	1
1.1.3 高级语言	2
1.1.4 C 语言	2
1.1.5 C++ 语言	2
1.2 C++ 语言简介	3
1.2.1 程序 = 数据 + 操作	3
1.2.2 数据类型	3
1.2.3 对数据的操作——表达式	3
1.2.4 数据操作的流程——语句控制	3
1.2.5 操作的复用——函数	3
1.2.6 数据和操作的封装——类	3
1.2.7 类的复用——派生和继承	3
1.2.8 软件模块	4
1.3 如何学习 C++ 语言	4
1.3.1 勤能生巧	4
1.3.2 风格与规范	4
1.4 用 Visual C++ 开发程序	4
1.4.1 程序——从概念到运行	4
1.4.2 Visual C++ 简介	5
1.4.3 建立应用程序	5
1.5 程序风格	7
1.5.1 效率与风格	7
1.5.2 注释——整理编程思路、增加程序的可读性	7
1.5.3 注释的形式	8
1.5.4 命名	13
1.5.5 编排	14

1.5.6	简单性原则	15
1.5.7	一致性原则	16
	小结	17
第2章	基本 C++ 程序结构	18
2.1	C++ 语言成分	18
2.1.1	字符	18
2.1.2	数字、标识符和表达式	18
2.1.3	语法	19
2.1.4	结构	19
2.1.5	模块	19
2.2	程序的基本结构	19
2.2.1	注释	20
2.2.2	编译预处理命令#include	20
2.2.3	main 函数	20
2.2.4	C++ 程序的执行过程	21
2.2.5	终端输出	22
2.3	变量和简单表达式	23
2.3.1	main 函数的返回值	24
2.3.2	常数和常量	24
2.3.3	变量	27
2.3.4	运算符与简单表达式	29
2.4	表达式和语句	31
2.4.1	返回值	31
2.4.2	嵌套	31
2.4.3	优先级	31
2.4.4	结合性	32
2.4.5	括号与次序	32
2.4.6	操作数	32
	小结	32
	习题	33
第3章	数据类型	36
3.1	数据及操作	36
3.2	C++ 数据类型	36
3.3	基本数据类型	37
3.3.1	数据在计算机中的存储	37
3.3.2	基本数据类型的修饰符号	38

3.3.3	基本数据类型	39
3.3.4	C++ 中的特殊字符	41
3.3.5	字符串	42
3.3.6	数据类型长度的确定	42
3.3.7	数据类型的选择	43
3.4	构造数据类型	44
3.4.1	数组	44
3.4.2	枚举	44
3.4.3	结构	46
3.5	数据输入	46
	小结	48
第4章	程序流程控制	50
4.1	复合语句	50
4.2	程序的执行流程	51
4.3	分支流程	51
4.3.1	if 语句	52
4.3.2	switch 语句	55
4.4	循环流程	57
4.4.1	while 语句	58
4.4.2	do...while 语句	62
4.4.3	for 循环	65
4.5	设计举例	68
4.6	其他控制语句	73
4.6.1	break 语句与多重循环	73
4.6.2	goto 语句	74
4.6.3	条件运算符?:	74
	小结	75
第5章	函数	77
5.1	函数机制	77
5.2	函数定义	78
5.2.1	函数名	79
5.2.2	函数参数	79
5.2.3	返回类型	80
5.2.4	函数体	80
5.3	函数调用	82
5.4	程序运行时的内存分布	83

5.5	函数调用的实现机制	84
5.6	函数的参数与返回值	88
5.7	函数参数的传递方式	88
5.7.1	指针参数	88
5.7.2	引用参数	90
5.7.3	数组参数	92
5.7.4	数组参数对实在参数的改变	94
5.7.5	字符串作为参数	94
5.7.6	多维数组作为参数	96
5.7.7	main()的参数	97
5.8	函数返回值	98
5.8.1	返回引用值	99
5.9	const 参数、const 返回值与 const 函数	100
5.9.1	const 参数	100
5.9.2	const 返回值	101
5.9.3	const 函数	101
5.10	作用域	102
5.10.1	局部作用域	103
5.10.2	文件作用域	104
5.10.3	覆盖问题	106
5.11	函数原型	109
5.11.1	函数作用域	109
5.11.2	函数原型	110
5.11.3	函数原型的作用	110
5.11.4	使用自己定义的函数原型	115
5.12	特殊的函数用法	115
5.12.1	内联函数	115
5.12.2	内联函数的作用	115
5.12.3	内联函数使用的限制	116
5.12.4	函数重载	116
5.12.5	使用函数重载的条件	117
5.12.6	重载函数的使用方法	117
5.12.7	函数的默认参数	118
5.12.8	递归函数	120
5.12.9	设计递归程序的方法	121
5.12.10	递归程序的优缺点	123
5.12.11	函数模板	123
5.12.12	模板的实例化	125

5.12.13 支持多种类型的模板	126
5.12.14 使用模板的注意事项	127
小结	127
第6章 简单程序设计	129
6.1 程序开发过程	129
6.2 文件之间的信息共享	130
6.2.1 外部变量	130
6.2.2 外部函数	132
6.2.3 静态全局变量与静态函数	133
6.3 头文件	135
6.3.1 #include 指令	135
6.3.2 头文件内容	136
6.3.3 预编译指令	138
6.4 生存期	141
6.4.1 生存期与内存	142
6.4.2 初始化问题	143
6.4.3 局部静态变量	143
小结	144
第7章 数组与结构	146
7.1 数组	146
7.1.1 数组的定义	146
7.1.2 数组元素的访问	147
7.1.3 数组复制	149
7.1.4 数组下标越界	149
7.1.5 数组元素的初始化	150
7.1.6 多维数组	151
7.1.7 多维数组元素的访问	151
7.1.8 多维数组元素的初始化	156
7.1.9 字符串数组	156
7.1.10 数组应用举例	158
7.2 结构	162
7.2.1 结构类型的定义	162
7.2.2 结构成员的访问	163
7.2.3 结构变量的初始化	163
7.2.4 结构的赋值	164
7.2.5 结构成员	166

7.2.6 结构嵌套	167
小结	167
第8章 指针和引用	169
8.1 指针的定义与初始化	169
8.2 指针的赋值与使用	171
8.3 指针的数据类型	174
8.3.1 void 指针 *	176
8.3.2 NULL 指针值	176
8.4 用 const 来限定指针	177
8.4.1 指向常量的指针	177
8.4.2 指针常量	178
8.4.3 指向常量的指针变量	178
8.5 指针与数组	179
8.5.1 指针运算	181
8.5.2 用指针对数组操作	183
8.5.3 指针与字符串	184
8.6 动态内存申请	185
8.6.1 new 与 delete	186
8.6.2 指针与动态数组	189
8.6.3 指针与动态结构	194
8.6.4 指针小结	201
8.7 引用的概念	201
8.7.1 引用的初始化	202
8.7.2 能够引用的数据类型	203
8.7.3 const 引用	204
8.8 指针和引用	205
8.9 引用的应用	206
小结	206
第9章 面向对象程序方法	207
9.1 面向对象方法概述	207
9.1.1 面向对象的概念	207
9.1.2 面向对象方法的形成	209
9.2 面向对象方法的基本概念	210
9.2.1 对象	211
9.2.2 程序设计语言中的对象	211
9.2.3 消息	212

9.2.4	程序设计语言中的消息	212
9.2.5	类	213
9.2.6	程序设计语言中的类	213
9.2.7	继承	214
9.2.8	程序设计语言中的继承	215
9.2.9	封装	215
9.2.10	程序设计语言中的封装	216
9.2.11	多态性	216
9.2.12	程序设计语言中的多态性	217
9.3	面向对象的意义	217
9.3.1	模块化——软件质量保证	217
9.3.2	复用——软件快速开发的必由之路	218
9.3.3	走面向对象式道路	218
9.3.4	结构化程序设计	219
9.3.5	面向对象程序设计	220
9.4	面向对象的分析与设计	221
9.4.1	面向对象开发方法的生命周期	221
9.4.2	面向对象分析	221
9.4.3	面向对象设计	222
9.5	面向对象方法与软件复用	222
9.5.1	复用级别	222
9.5.2	复用的好处	223
9.5.3	面向对象方法对复用的支持	223
9.5.4	复用技术对面向对象软件开发的支持	224
9.6	面向对象程序设计语言	224
	小结	225
第10章	类与对象	226
10.1	类的定义与使用	226
10.2	成员的访问控制	227
10.3	类的成员函数	231
10.3.1	成员函数的定义位置	231
10.3.2	类作用域	232
10.3.3	内联成员函数与非内联成员函数	235
10.3.4	使用内联函数的条件	236
10.3.5	const 成员函数	238
10.4	类定义与头文件	240
10.4.1	使用类库中的类	243

10.4.2 类与程序结构	244
10.5 对象的创建与使用	245
10.5.1 对象的创建	245
10.5.2 对象作为函数的参数与返回值	246
10.6 this 指针	247
10.7 类与结构的关系	250
10.8 举例	252
10.8.1 用数组实现的队列类	253
10.8.2 用链表实现的队列类	259
小结	265
第 11 章 构造函数与析构函数	267
11.1 构造函数和析构函数的意义	267
11.2 构造函数	270
11.2.1 重载构造函数	270
11.2.2 对象构造的几种方式	274
11.2.3 构造函数的默认参数	274
11.2.4 默认构造函数	275
11.2.5 拷贝构造函数	277
11.2.6 成员初始化参数表	280
11.3 析构函数	285
小结	295
第 12 章 静态成员、友元	297
12.1 静态成员	297
12.1.1 静态数据成员与静态函数成员	298
12.1.2 静态数据成员的初始化	299
12.2 友元	306
12.2.1 友元的说明与使用	306
12.2.2 使用友元的条件	311
12.2.3 使用友元的注意事项	311
小结	313
第 13 章 运算符重载	314
13.1 可以重载的运算符	314
13.2 运算符重载的规则	316
13.3 常用运算符重载举例	319
13.3.1 函数调用运算符的重载	320

13.3.2	赋值运算符的重载	323
13.3.3	双目算术运算符的重载	325
13.3.4	单目算术运算符的重载	327
13.3.5	自增自减运算符的重载	328
13.4	插入抽取运算符的重载	337
13.4.1	C++的I/O流库	338
13.4.2	插入运算符的重载	339
13.4.3	抽取运算符的重载	341
13.5	类型转换函数	345
13.5.1	构造函数充当类型转换函数	346
13.5.2	直接定义类型转换函数	347
13.5.3	隐式类型转换与显式类型转换	349
	小结	350
第14章	继承	352
14.1	单继承	352
14.1.1	公有、保护和私有继承	354
14.1.2	派生类对象的构造	361
14.1.3	继承应用举例	368
14.2	多继承	372
14.2.1	多继承的定义	373
14.2.2	多继承中的构造函数与析构函数	373
14.2.3	二义性	376
14.2.4	虚基类	379
14.2.5	虚基类的初始化	380
14.2.6	多继承的应用	381
14.3	继承与类库	381
14.4	Visual C++的类库——MFC	382
	小结	383
	自测题答案	384
	参考文献	421

第 1 章 C++ 编程基础

本章要点

(1) C++ 是一种面向对象的高级程序设计语言,具有良好的数据、功能封装性,可以提高软件的可复用性。

(2) Visual C++ Studio 是 C++ 语言的一种集成开发环境,提供了高效的编辑、编译、链接和调试功能。

(3) 软件书写应当遵循良好的风格,这包括注释、变量命名、编排、简单性和一致性等方面。

1.1 程序语言的发展

1.1.1 机器语言

计算机是人们的好帮手,它有能进行快速处理的 CPU,有能快速存取信息的内存,有能存放大量信息的硬盘。但是,计算机毕竟是一台机器,除非我们告诉它做什么,怎么去做,否则它什么也不会做。

要让计算机做事,必须给它“指令”。可是计算机很笨,它只认识 0 和 1,它的所有操作和数都用 0 和 1 来表示。如计算机中要表示数 5,6,7,8,9,必须分别用四位二进制数来表示:0101,0110,0111,1000,1001。而让计算机进行加、减、乘、除等操作,也必须用二进制数来表示。如果让计算机计算 $AL = 4 + 5$,则需要向计算机发出以下指令:

```
1011 0000 0000 0100
0000 0100 0000 0101
```

这是计算机可以理解的语言,我们称它为机器语言。

1.1.2 汇编语言

计算机用数来“思考”,而人不行。于是聪明的程序员想出了一种办法,用一些人易于理解的符号来代替上面的 0、1 序列,如求 $AL = 4 + 5$ 可以写成下面两句话:

```
MOV AL, 4
ADD AL, 5
```

上面的每一句话都代表一个指令。这样的语言很容易被翻译成机器语言,因为翻译是一种很机械的活,非常适合计算机进行,所以人们编写了汇编程序,专门负责将上面这

种符号语言(称为汇编语言)翻译成机器语言。程序员再也不用与繁琐的 0 和 1 打交道了。

1.1.3 高级语言

随着计算机硬件技术的迅速发展,计算机的处理能力不断提高。人们发现,可以用更自然的方式书写程序,如可以直接在程序中写“AL=4+5”来进行计算。然后将这些程序翻译成机器能理解的机器语言。于是产生了高级语言,翻译这些高级语言的程序称为“编译程序”。C、FORTRAN、PASCAL、COBOL 等都是高级语言。

为什么不用人类的自然语言来作为计算机语言呢?因为人类语言有一个很大的缺陷:它是不精确的。即使是像法律、合同这样精心精确书写的条文,也还是可能会存在二义性。而计算机需要的是准确而详尽的指示,自然语言显然无法胜任。计算机高级语言的发展趋势是向自然化发展,但是自然语言无法作为计算机语言。

1.1.4 C 语言

1970 年,两位程序员 Brian Kernighan 和 Dennis Ritchie 在 B 语言的基础上首创了一种新的程序设计语言,取名为 C 语言。设计 C 语言的最初目的是编写操作系统,它是 UNIX 操作系统的开发语言。C 语言有很多优点:与硬件无关,移植方便;语言简洁,使用方便;丰富的运算符和数据类型;可以直接访问内存地址,能进行位操作;生成的目标代码质量高,程序运行效率高。由于这些优点,C 语言很快就被用于编写各种不同类型的程序,从而成为世界上最流行的语言之一。

1.1.5 C++ 语言

C++ 源于 C 语言。随着 C 语言的应用,它的缺点也逐渐显示了出来:C 语言的类型检查机制弱,使得程序开发过程中的错误不能在编译时被发现;C 语言本身是面向过程的语言,没有支持代码复用的机制,因此所有的程序都需要从头开始编制,而且当程序规模达到一定程度时,程序员很难控制程序的复杂性。

20 世纪 80 年代初,美国 AT&T 贝尔实验室的 Bjarne Stroustrup 设计并实现了 C 语言的扩充、改进版本,C++ 语言由此诞生。C++ 语言改进了 C 的不足之处,增加了对面向对象的支持,在改进的同时,保持了 C 的简洁性和高效性。C++ 包含了 C 的所有语法,大多数 C 程序都可以简单地转化为 C++ 程序(只是里面不包含 C++ 新的特征),所以原来的一大批 C 程序员可以很容易地成为 C++ 语言的拥戴者。更重要的是,植根于 C 语言的 C++ 语言继承 C 的高效简洁的特点,可以使用比其他大多数语言更高效的方法组织信息。所以,C++ 语言获得了很大的成功。

时至今日,C++ 越来越受到重视并且已经得到了广泛的应用,许多软件公司如 Microsoft, Inprise 公司都为 C++ 设计编译系统,提供不同应用级别的类库和越来越方便的开发环境。利用 C++ 设计并实现应用系统已是日益简单和快捷了。