

北京希望电脑公司Borland C++3.0系列丛书

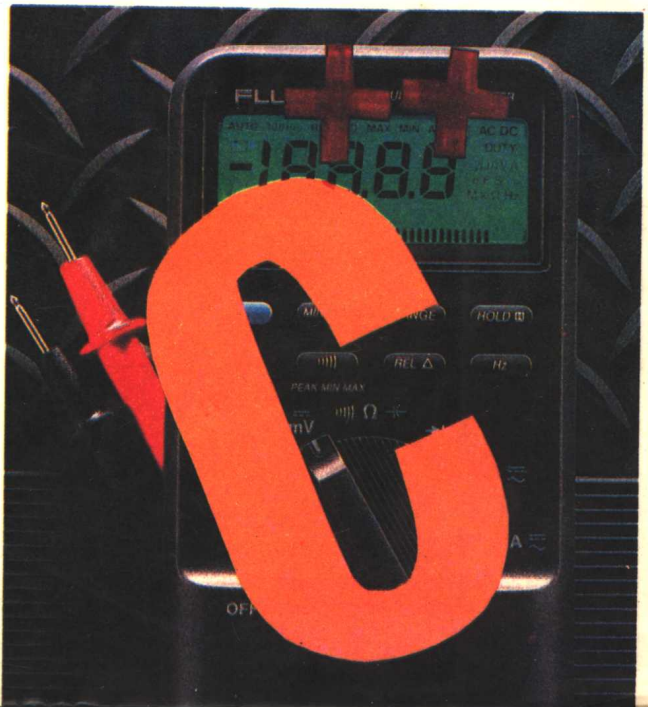
北京希望
电脑公司

Borland C++3.0

DOS和Windows编程技巧与实例

周枫 编

海洋出版社



北京希望电脑公司 Borland C++ 3.0 系列丛书

Borland C++ 3.0

DOS 和 Windows 编程技巧与实例

周 枫 编

海洋出版社

1992 · 2 北京

内容简介

本书分为三部分。第一部分包括基本的 Borland C++ 编程环境和 C(非面向对象)语言;第二部分包括 C++(面向对象)语言;第三部分对 Windows 应用程序编程作了基本的介绍,其中包括高版本 Borland C++ 3.0 的使用。每章分成若干节,各个小节都经过精心设计,包含适量的内容和学习用例,易于消化吸收。

本书是 Borland C++ 3.0 系列丛书之一,需要本书和整套丛书者请与北京 8721 信箱联系,邮编 100080,电话 2562329。

(京)新登字 087 号

责任编辑:阎世尊

Borland C++ 3.0
DOS 和 Windows 编程技巧与实例

周 枫 编
希 望 审校

海洋出版社(北京市复兴门外大街1号)
海洋出版社发行 双青印刷厂印刷
开本: 787x1092 1/16 印张: 54.5 字数: 1300 千字
1992年1月第一版 1992年1月第一次印刷
印数: 1-3000册
ISBN 7-5027-2609-8/TP.90 定价: 40.00 元

目 录

第零章	前 言	1
0.1	本书适合的读者对象	1
0.2	本书的内容	1
0.3	本书的结构安排	2
0.4	本书的使用方法	2
0.5	C和C++编程练习注意事项	2
0.6	选择 Borland 编译器	3
第一部分 使用 Borland C++的基本特征		
第一章	启动 Borland C++	5
1.1	运行安装程序	5
1.1.1	安装 Borland C++	5
1.2	集成开发环境(IDE)的使用	7
1.2.1	启动 IDE	7
1.2.2	使用 IDE 的菜单和窗口	9
1.2.2.1	菜单系统	9
1.2.2.2	窗口系统	10
1.3	配置 Borland C++	13
1.3.1	设置 IDE 选择项	13
1.3.2	使用 BCINST	15
1.4	编写第一个 C 语言程序	18
1.4.1	理解 C 程序的结构	18
1.4.1.1	预处理器指令	20
1.4.1.2	全局说明	21
1.4.1.2	main()函数	22
1.4.1.3	用户定义的函数	22
1.4.2	使用编辑器编写程序	23
1.5	介绍库函数	26
1.5.1	使用一些基本的输入函数	26
1.5.1.1	get()函数	27
1.5.1.2	scanf()函数	28
1.5.2	使用一些基本的输出函数	30
1.5.2.1	put()函数	30
1.5.2.2	printf()函数	31
1.5.3	使用一些格式转换函数	31

1.5.3.1	atoi() 函数	31
1.5.3.2	toupper() 函数	32
1.6	练习	33
1.7	总结	34
第二章	理解 C 语言的基础	36
2.1	预备知识	36
2.1.1	理解源文件、目标文件和装载模块	36
2.1.2	理解程序的逻辑和执行流程	37
2.1.2.1	使用条件指令	37
2.1.2.2	使用循环指令	41
2.2	使用基本数据类型	43
2.2.1	理解 C 语言的基本数据类型	43
2.2.1.1	整数	43
2.2.1.2	浮点数	44
2.2.1.3	字符型数据	45
2.2.2	在何处定义数据对象	46
2.3	编写 C 语言表达式和语句	48
2.3.1	理解表达式和语句	49
2.3.1.1	C 语言的操作符集	50
2.4	控制类型转换	53
2.4.1	理解隐式类型转换	53
2.5	显式类型转换的使用	55
2.6	使用 C 的宏	56
2.6.1	定义类似对象的宏	56
2.6.2	定义类似函数的宏	59
2.7	练习	62
2.8	总结	62
第三章	使用 C 语言函数	64
3.1	理解 main() 函数和库函数	64
3.1.1	编写 main() 函数	64
3.1.2	使用库函数	66
3.1.2.1	哪些库函数可以利用	66
3.1.2.2	在程序中加入库函数	67
3.2	编写用户自己的函数	69
3.2.1	编写函数原型	69
3.2.2	把参数传送给函数	71
3.3	函数返回值	74
3.3.1	定义和使用函数类型	74
3.3.2	象数据项一样使用函数	76
3.4	理解存储类别	76

3.4.1	确定变量的作用域	77
3.4.2	确定变量的生存期	78
3.4.3	确定变量的连接	79
3.5	使用高级的程序控制逻辑	80
3.5.1	编写循环控制语句	80
3.5.1.1	使用 goto 语句	80
3.5.1.2	使用 break 语句	81
3.5.1.3	使用 continue 语句	81
3.5.2	改变程序的执行流程	82
3.5.2.1	使用 exit() 和 abort() 函数	82
3.5.2.2	使用 system(), exec...(), 和 spawn() 函数	83
3.6	使用可变参数表	84
3.6.1	设计可变参数表	84
3.6.2	使用 va...() 函数	85
3.7	练习	89
3.8	总结	89
第四章	使用指针和派生类型	91
4.1	理解标准 C 语言的派生类型	91
4.1.1	理解 C 的类型划分方案	91
4.1.2	从旧类型中创建新类型	92
4.1.3	什么是不完全类型	92
4.2	理解 C 语言的指针	94
4.2.1	理解间接寻址	95
4.2.2	RAM 地址和系统装入程序	95
4.2.2.1	使用 C 语言的间接操作符和取址操作符	100
4.2.3	使用数组和串	102
4.2.3.1	声明和使用对象数组	102
4.2.3.2	理解 C 语言的串	105
4.2.4	使用结构和联合类型	108
4.2.4.1	从不同类型中建立结构类型	108
4.2.4.2	联合是结构对象的变化形式	111
4.3	使用指向函数的指针	113
4.3.1	指向函数的指针声明和初始化	113
4.3.2	利用指针引用某函数	114
4.4	在动态内存中使用指针	117
4.4.1	C 语言程序和动态内存	117
4.4.2	使用动态存储	119
4.5	练习	125
4.6	总结	125
第五章	建立、编译和测试 Borland C++ 程序	127

5.1	在一个程序中使用几个源文件	127
5.1.1	确定把哪些函数放在源文件中	127
5.1.2	设置 Borland C++ 工程文件	143
5.2	理解外部引用	145
5.2.1	使用关键字 extern	145
5.2.2	使用外部函数	146
5.3	为外部模块编写头文件	147
5.3.1	确定头文件中的内容	147
5.3.2	包含用户提供的头文件	148
5.4	在头文件中使用带条件的编译程序说明	149
5.5	用 IDE(集成开发环境)编译和运行程序	151
5.5.1	编译和运行简单的程序	151
5.5.2	编译和运行复杂的程序	152
5.6	介绍集成 Debugger	154
5.6.1	使用 step over 命令缩小搜索范围	154
5.6.2	使用 Trace into 命令来对问题进行准确定位	155
5.6.3	设置和删除断点	156
5.7	练习	157
5.8	总结	158
第六章	使用 Borland C++ 输入/输出函数库	159
6.1	理解 I/O 概念	159
6.1.1	理解文件与设备	159
6.1.2	理解文件与流	160
6.1.3	文本流和二进制流	162
6.2	利用标准流进行 I/O	162
6.2.1	使用格式化 I/O 函数	163
6.2.2	理解 scanf() 函数	167
6.2.3	理解 printf() 函数	170
6.2.4	使用字符 I/O 函数	173
6.3	使用文件控制函数	177
6.3.1	打开文件、关闭文件和控制文件	177
6.3.2	控制文件缓冲区	180
6.4	使用直接文件 I/O 函数	182
6.4.1	理解直接 I/O 的概念	182
6.4.2	读写直接文件	183
6.5	使用文件定位函数	189
6.5.1	得到当前文件位置	189
6.5.2	设定一个新文件位置	190
6.6	处理文件 I/O 错误	192
6.6.1	检测文件 I/O 错误	192

6.6.2	显示和清除文件 I/O 错误	192
6.7	练习	193
6.8	小结	193
第七章	存储模式与 VROOMM 技术	195
7.1	80×86 的体系结构	195
7.1.1	段(Segment)、节(Paragraph)以及偏移地址(Offset)	195
7.1.2	CPU 的地址寄存器	197
7.2	near 指针、far 指针和 huge 指针	199
7.2.1	选择想要的指针大小	199
7.2.2	near、far 和 huge 说明符	200
7.3	六个 Borland C++ 存储模式	203
7.3.1	决定使用哪种存储模式	203
7.3.2	以混合模式编程	204
7.4	创建 COM 型的可执行程序文件	206
7.4.1	使用 COM 文件	206
7.5	覆盖管理程序	208
7.5.1	覆盖管理程序会做什么	209
7.5.2	Borland 的 VROOMM	209
7.6	设计和创建覆盖程序	211
7.6.1	决定哪些模块要覆盖	211
7.6.2	覆盖程序的编译和链接	212
7.7	练习	214
7.8	小结	215
第八章	使用 Borland C++ 的视频函数	217
8.1	IBM/PC 的文本方式	217
8.1.1	PC 显示器适配器和屏幕的概述	217
8.1.2	掌握存储变换屏幕 I/O	218
8.2	控制文本屏幕	219
8.2.1	使用文本方式控制函数	219
8.2.2	使用直接控制台 I/O 以获得高性能	223
8.3	使用窗口函数	223
8.4	了解 IBM-PC 的图形方式	227
8.4.1	象素点与调色板	227
8.4.2	控制图形屏幕	231
8.5	介绍 BGI 图形库	233
8.5.1	使用画图 and 填充函数	233
8.5.2	控制屏幕和视口	238
8.6	在图形方式下显示文本	238
8.6.1	BGI 字体	239
8.6.2	使用图形方式下的文本函数	239

8.7	练习	242
8.8	小结	242
第九章	使用 Turbo Debugger 和 Turbo Profiler	244
9.1	启动 Turbo Debugger	244
9.1.1	理解错误和调试程序的概念	244
9.1.2	准备在 IDE 里运行 Turbo Debugger	246
9.2	使用 Turbo Debugger	246
9.2.1	用 Turbo Debugger 运行程序	246
9.2.2	观察数据	250
9.3	使用高级调试功能	252
9.3.1	改变程序中变量或表达式的值	252
9.3.2	设置断点	253
9.4	使用 Turbo Profiler	254
9.4.1	什么是“剖析”(profiling)	254
9.4.2	启动 Turbo Profiler	255
9.4.3	了解基本的剖析信息	257
9.5	使用高级剖析功能	260
9.5.1	选择剖析区域	261
9.5.2	显示剖析统计数据	263
9.5.3	积极剖析与消极剖析	265
9.6	练习	266
9.7	小结	266
第十章	使用 Borland C++ 的高级功能	268
10.1	使用直接插入(inline)汇编语言	268
10.1.1	直接插入式汇编环境	269
10.1.2	使用 asm 关键字	271
10.2	与汇编语言子程序的接口	274
10.2.1	在 C 程序里调用汇编子程序	274
10.2.2	在汇编子程序中调用 C 函数	285
10.3	使用中断功能	293
10.3.1	80×86 的中断结构	293
10.3.2	使用 Borland 的中断接口	295
10.4	使用中断处理程序	298
10.4.1	声明中断处理程序函数	298
10.4.2	实现一个时钟中断处理程序	300
10.5	练习	304
10.6	小结	305

第二部分 Borland C++ 面向对象的功能

第十一章	使用 C++ 的类	306
-------------	------------------	------------

11.1	C 与 C++ 派生类型的比较	306
11.1.1	在 C++ 中重新定义“派生类型”	307
11.1.2	理解 C++ 的封装(Encapsulation)概念	308
11.1.3	使用 struct 说明类	309
11.2	说明 C++ 的类	313
11.2.1	理解类的说明	315
11.2.2	使用类说明的语法规则	316
11.2.3	说明类成员	319
11.2.4	建立后进先出的堆栈类	320
11.2.5	使用 this 指针	326
11.2.6	使用关键字 public, private 以及 protected	332
11.3	编写类的成员函数	332
11.3.1	联系成员函数和类	333
11.3.1.1	分别编译成员函数	333
11.3.1.2	定义直接插入的成员函数	334
11.3.1.3	指定成员函数的缺省参数	335
11.3.2	提供构造函数和析构函数	336
11.4	使用友元函数	340
11.4.1	在类中包括友元函数	340
11.4.2	决定何时使用友元函数	341
11.5	练习	342
11.6	小结	342
第十二章	创建 C++ 的对象	345
12.1	定义 C++ 的对象	345
12.1.1	为类对象指定存储类	345
12.1.2	定义具有动态生存期的类对象	347
12.1.3	定义局部(auto)类对象	362
12.1.4	定义全局(静态)的类对象	370
12.2	初始化类对象	378
12.2.1	使用构造函数来初始化类对象	379
12.2.2	使用初始化表	381
11.3	练习	382
11.4	总结	382
第十三章	访问 C++ 的对象	384
13.1	使用作用域限制操作符	384
13.1.1	作用域限制的一般用法	384
13.1.2	使用作用域限制进行语法控制	386
13.1.3	利用作用域限制来控制二义性	387
13.2	理解 C++ 的作用域规则	391
13.2.1	理解 C 和 C++ 作用域的不同	392

13.2.2	C++的作用域规则	393
13.3	同C++的对象交流信息	396
13.3.1	向对象传送信息	396
13.3.2	理解*this	420
13.4	使用引用操作符	421
13.4.1	由地址操作符进化而来的引用操作符	421
13.4.2	理解引用操作符	421
13.5	使用对象作为函数的参数	422
13.5.1	通过值和通过引用来传送对象	422
13.5.2	从成员函数中访问其它对象	424
13.6	使用指向对象的指针	425
13.6.1	理解何时需要指针	425
13.6.2	定义指向对象的指针和数组	426
13.7	练习	428
13.8	小结	428
第十四章	在C++中使用重载函数和操作符	430
14.1	重载成员函数	430
14.1.1	理解C++的重载	430
14.1.2	声明重载的成员函数	433
14.2	重载友元函数和非成员函数	435
14.2.1	重载类的友元函数	435
14.2.2	重载非成员函数	440
14.3	理解类型安全的连接	441
14.3.1	理解函数合并	441
14.3.2	利用标准C的包含文件来控制连接问题	443
14.4	在C++中重载操作符	445
14.4.1	理解操作符重载	446
14.4.2	声明重载操作符函数	451
14.4.2.1	定性地讨论操作符函数	451
14.4.2.2	理解重载操作符函数的语法规则	452
14.4.2.3	指定重载操作符函数的参数	454
14.4.2.4	指定操作符函数的返回类型	455
14.4.2.5	利用友元函数来重载操作符	456
14.4.2.6	定义强制类型转换操作符函数	459
14.4.3	重载双目操作符和单目操作符	460
14.5	重载下标操作符和函数调用操作符	461
14.5.1	使用重载的下标操作符	462
14.5.2	使用重载函数调用操作符	466
14.6	练习	470
14.7	小结	471

第十五章 使用 C++ 的构造函数和析构函数	472
15.1 理解构造函数和析构函数调用.....	472
15.1.1 声明构造函数和析构函数.....	472
15.1.1.1 声明构造函数.....	473
15.1.1.2 声明析构函数.....	477
15.1.2 使用构造函数初始化.....	479
15.1.3 何时调用构造函数.....	482
15.1.4 何时调用析构函数.....	488
15.2 重载构造函数.....	492
15.3 写一个缺省构造函数.....	492
15.3.1 编写其它构造函数.....	494
15.3.2 决定何时需要一个拷贝构造函数.....	496
15.4 使用 new() 操作符和 delete() 操作符.....	497
15.4.1 new 和 delete 的使用.....	497
15.4.2 动态地创建和删除类对象.....	499
15.5 重载操作符 new() 和操作符 delete().....	501
15.5.1 重载全局操作符.....	501
15.5.2 重载类的操作符.....	507
15.6 练习.....	509
15.7 总结.....	509
第十六章 使用 C++ 流	511
16.1 C++ 流介绍.....	511
16.1.1 C++ 流与标准流的比较.....	512
16.1.2 使用 C++ 流进行标准 I/O.....	514
16.2 处理 C++ 流的错误.....	522
16.2.1 检测 C++ 流错误状态.....	523
16.2.2 使用流状态成员函数.....	525
16.3 用 C++ 流控制数据格式.....	527
16.3.1 在内部类型中使用插入符和抽取符.....	527
16.3.2 重载 << 和 >> 操作符.....	532
16.4 使用 C++ 流操纵符(manipulators).....	535
16.4.1 理解 C++ 的操纵符.....	535
16.4.2 使用操纵符改变状态和属性.....	536
16.5 使用 C++ 文件 I/O 流.....	544
16.5.1 读和写 fstream 文件.....	548
16.5.2 用 C++ 流进行文件定位.....	549
16.6 使用和转换旧式 C++ 流.....	551
16.6.1 新旧 C++ 流的区别.....	552
16.6.2 转化为新的 C++ 流.....	553
16.7 练习.....	554

16.8 总 结	555
第十七章 使用 C++ 派生类	557
17.1 无继承性的代码重用	557
17.1.1 理解代码的重用性	557
17.1.2 通过组合重新使用代码	558
17.2 使用单个基类	561
17.2.1 理解继承	561
17.2.2 声明基本和派生类	561
17.3 一个实践项目：扩展内存支持	569
17.3.1 理解扩展内存规范(XMS)	569
17.3.2 建立 XMS 管理类	570
17.3.3 派生一个交换缓冲类	583
17.4 使用虚拟函数	588
17.4.1 使用迟后联编和虚拟函数	588
17.4.2 使用域限定控制符来控制成员函数的存取权限	592
17.5 使用多基类	592
17.5.1 从多个基类中派生	593
17.5.2 声明和使用虚拟基类	596
17.6 从抽象类中派生类	597
17.6.1 理解纯虚拟函数	598
17.6.2 纯虚拟函数的实现	598
17.7 在继承时使用构造函数和析构函数	599
17.7.1 初始化代码不能继承	599
17.7.2 理解在继承时调用构造函数和析构函数的顺序	600
17.7.3 使用虚拟析构函数	600
17.8 练 习	602
17.9 小 结	603
第十八章 对象控制、操作及将来发展方向	604
18.1 用户自定义类型转换	604
18.1.1 使用类的构造函数来进行类型转换	604
18.1.2 类型操作符的重载	606
18.2 使用 generic 类	609
18.2.1 理解抽象和 generic 类的设计	609
18.2.2 构造 generic 类	613
18.3 控制对象的行为与性能	620
18.3.1 使用友元函数以提高效率	620
18.3.2 使用友元函数来控制对象句法	621
18.3.3 使用静态存储类来避免重复创建	623
18.3.4 使用可引用量和指针	626
18.3.4 使用直接插入函数来消除函数调用	628

18.4	类对象控制中今后的发展方向	629
18.4.1	类和函数的模板	629
18.4.2	异常处理	630
18.5	练习	631
18.6	总结	631

第三部分 编写 Windows 应用程序

第十九章	使用 Turbo Vision	632
19.1	Turbo Vision 概述	632
19.1.1	何谓应用程序框架	632
19.1.2	为何要使用 Turbo Vision 的应用程序框架	633
19.1.3	使用面向对象的代码的优点	633
19.1.4	事件驱动编程的优点	634
19.2	Turbo Vision 应用程序概述	635
19.2.1	事件	635
19.2.2	视口	635
19.2.2.1	组	635
19.2.3	哑对象	636
19.3	编写第一个 Turbo Vision 应用程序	636
19.3.1	编写一个基本的 Turbo Vision 应用程序	636
19.3.2	增加菜单支持	638
19.4	使用 Turbo Vision 创建窗口	643
19.4.1	在 Turbo Vision 应用程序中增加窗口	643
19.4.2	在 Turbo Vision 窗口中增加文本	648
19.4.3	使用对话框窗口	652
19.5	练习	660
19.6	小结	660
第二十章	Borland C++ 高级编译器介绍	661
20.1	新的特征和功能	661
20.2	运行的程序员平台	663
20.2.1	运行在 MS-Windows 下的程序员平台	663
20.2.2	使用 Borland C++ 的选项	665
20.3	Turbo Debugger 2.5 & 3.0	666
20.4	使用 Turbo Debugger 2.5 和 3.0	667
20.4.1	影响 TASM 的 C 编译器的差异	667
20.4.2	支持 Windows 3.0 的新 MODEL 标识符	670
20.4.3	虚段支持	671
20.4.4	486 指令集支持	671
20.4.5	RETCODE 指令	672
20.4.6	TASM.EXE 和 Windows DMPI	672

20.4.7	使用 Whitewater 资源开发工具	672
20.5	使用 Resource Workshop	675
20.6	总 结	687
第二十一章	使用 Borland C++ 进行 Windows 程序设计	688
21.1	了解 Windows 程序设计环境	688
21.1.1	Windows 是多任务环境	689
21.1.2	Windows 环境是面向对象的环境	691
21.2	设置 Borland C++ 以支持 Windows 应用程序设计	698
21.2.1	设置选择项以支持资源编译器和输入库管理器	698
21.2.1.1	只运行集成环境(IDE)	699
21.2.1.2	在 Windows 下运行集成环境(IDE)	699
21.2.2	为入口和出口代码设置编译器选项	700
21.3	使用 WhiteWater 资源工具箱(RT)	702
21.3.1	设置 RT 作为一个 Windows 程序项	703
21.3.2	fcwin.c 程序需要的资源	703
21.3.3	运行 RT	704
21.3.3.1	使用 RT 的主窗口	704
21.3.3.2	使用资源浏览器	705
21.3.3.3	建立和编辑图符	706
21.3.3.4	生成和编辑菜单	706
21.3.3.5	生成和编辑对话框	708
21.5	利用 Borland 的 Resource Workshop 绘制 FCWIN 资源	713
21.6	练 习	721
21.7	总 结	722
第二十二章	设计 Windows 应用程序	723
22.1	建立 Windows 应用程序环境	723
22.1.1	建立 Windows 承接程序可执行文件	723
22.1.2	了解 Windows 目录的用途	724
22.1.1.3	通过实践学习: 设计 fcwin.c	725
22.2	为 Windows 应用程序生成源文件	756
22.2.1	了解 Windows 3.0 程序设计环境	757
22.2.2	生成模块定义文件	757
22.2.3	设计程序的头文件	759
22.2.4	为 FCWIN 生成项目文件	760
22.3	练 习	761
22.4	总 结	761
第二十三章	用 Borland C++ 2.0 写 Windows 应用程序	763
23.1	设计 Windows 界面	763
23.1.2	注册窗口类	763
23.1.2	设置主消息循环	766

23.1.3	编写 WndProc()函数	767
23.1.4	为对话框设置回调函数	768
23.1.4.1	生成对话函数	769
23.1.4.2	控制对话框	770
23.1.5	为弹出帮助和出错信息使用 MessageBox()	771
23.1.6	Windows 打印管理器的 spooling 硬拷贝	772
23.2	使用动态连接库	780
23.2.1	理解 DLL	780
23.2.2	编写一个 DLL 应用程序	781
23.3	练习	785
23.4	总结	785
第二十四章 使用 ObjectWindows		786
24.1	C++ ObjectWindows 简介	786
24.1.1	在 Windows 程序中使用 C++ 类	787
24.1.2	了解 ObjectWindows 库	800
24.1.3	编写第一个 ObjectWindows 程序	801
24.2	与基本 Windows API 函数和元素的接口	805
24.2.1	控制应用程序	806
24.2.2	控制窗口对象	807
24.2.3	控制 Windows 消息	809
24.3	练习	810
24.4	小结	811
附录 A	ASCII 字符	812
附录 B	使用 printf()和 scanf()的说明	815
附录 C	使用 exec...()和 spawn...()的详细说明	826
附录 D	quad 类的程序清单	830
附录 E	CWIN 资源的完整清单	843

第零章 前 言

欢迎你加入到令人激动的、具有挑战性的 C 和 C++ 语言编程世界里来。

C 语言是具有挑战性的——它迅速发展成为如今最普遍、最强有力的编程语言之一。在普及 C 和 C++ 方面，Borland International 公司起着尤为重要的作用。Borland C++ 的编译器和开发环境使得曾经只有开发者才能购买得起的单个的复杂编程工具变得对用户也很方便实用。《Borland C++ 3.0 编程技巧与实例》这本书就是根据这个产品而来的。

Borland C++ 3.0 的特色包括 C 和 C++ 编译器(新版本)、一个独立的调试器、一个程序剖视器和 Turbo 汇编程序，另外还支持建立 Windows 3.0 下可执行的动态链接库(DLL)模式。另外，Borland C++ 还含有创建和维护 Windows 资源(图标、菜单、对话框及其它)的 Whitewater Group 资源工具箱。本书讨论了以上所有这些特色。

从 UNIX 系统程序员首次使用 C 语言到现在，C 语言得到了迅速的普及，C 的后继者 C++ 使这种势头得到延续，C 和 C++ 语言在各种 PC 机上得到了广泛的应用。C 语言正迅速成为认真的程序员所选择的一种语言，而不管他是业余的还是专业软件开发人员。C++ 也在很快流行起来，虽然 C++ 与 C 关系密切，然而作为两个不同的语言 C 和 C++ 之间仍有很大的差别，本书将讨论这两者的差异。

0.1 本书适合的读者对象

本书适合于有下述要求的读者：

- 试图学习更多有关 C 语言编程的人员。这里面既包括对 C 或 C++ 一无所知的用户，也包括有一些 C 或 C++ 编程经验的用户。为了方便初学者学习，本书有丰富的入门知识，对有一定知识和经验的程序员来说，本书提供了大量的中高级编程知识和技巧。
- 确实想使用 Borland 公司的 C++ 编译器的人员。Borland C++ 有它自己独特的编程环境和扩展了的 ANSI 标准 C 集合，本书介绍了这些内容。

0.2 本书的内容

本书有两个目的：一是帮助学习 C 和 C++，二是说明如何使用 Borland 公司的 C++ 编程和开发环境。

这本书的中心内容是 C 和 C++ 语言，特别是它们在 Borland C++ 中的实现方法。读者可能对编制 C 和 C++ 程序比阅读有关文本编辑器、调试器或者程序剖视器的内容(尽管本书中也包括这些内容)更感兴趣，所以，本书的最主要任务是对语言进行指导性介绍。

当然，读者也会对 Borland C++ 配套的工具和实用程序发生兴趣，这些工具和实用程序就是这本书的第二个中心内容。本书也介绍了如何使用这个软件包中主要组成部分的知识。