

黄嘉光等 编写

计算机 C/C++ 语言系列丛书

Microsoft

C/C++

Microsoft

C/C++

7.0

百科全书

学苑出版社

希望

计算机 C/C++ 语言系列丛书

Microsoft C/C++ 7.0

百科全书

黄磊光	
程思乾	编写
李 津	
万 博	审校

学苑出版社

1994年·北京

(京)新登字 151 号

内 容 简 介

本书是针对利用 Microsoft C/C++ 7.0 进行 C 和 C++ 程序设计的人编写的,书中全面系统地介绍了 Microsoft C/C++ 7.0。首先介绍 Microsoft C/C++ 7.0 集成环境(PWB),说明集成编辑器和调试器的用法;然后介绍 C 和 C++ 语言的基础知识,包括操作符和表达式、说明、控制语句、函数、指针和其它高级数据结构,介绍文件输入/输出;接着介绍了类型兼容、宏、条件编译指令、预编译处理,讨论了面向对象的基本编程方法;再就是说明了处理屏幕文本和图形的技术,介绍了如何充分利用 Microsoft C/C++ 7.0 的库函数,描述了与汇编语言接口技术;还介绍了用 Microsoft C/C++ 7.0 进行 Windows 程序设计的技术;在附录部分给出了 Microsoft C/C++ 7.0 基础类库的快速参考。

欲购本书的用户可直接与北京 8721 信箱联系,电话 2562329, 邮编 100080。

计算机 C/C++ 语言系列丛书

Microsoft C/C++ 7.0 百科大全

编 写:黄磊光 程思乾 李 津
审 校:万 博
责任编辑:甄国宪
排 版:万博图书创作社
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:保定列电印刷厂
开 本:787×1092 1/16
印 张:49 字数:1134 千字
印 数:1~5000 册
版 次:1994 年 7 月北京第 1 版第 1 次
本册定价:63.00 元
ISBN7-5077-0875-6/TP·24

前 言

本书介绍 Microsoft C/C++ 提供的工具,包括 C/C++ 编译器、程序员工作台 (PWB)、调试工具、剖析工具和各种 Windows 开发工具,实际上是一个 Microsoft 工具手册,帮助读者很快熟悉软件包的每个组成部分。

使用每种设计语言时,程序员需要完整地去理解它。本书包含了 C、C++ 和 Windows 语言的所有重要的设计原理,包括新基础类库。对于初学者,前面的章节可以打好编写更复杂程序的坚实基础;对于高级设计者,这些章将作为一个手册,介绍 C++ 的原理。

本书还介绍了如何调试代码,并避免语法和逻辑设计错误,帮助理解过程设计与面向对象设计的不同及学会如何开发简单的 OOP 程序。

本书讨论了 Microsoft 窗口设计,有四章的篇幅为读者讲解 Windows 原理,并用示例解说如何编写简单的 Windows 程序。附录 C 列出了许多重要的 Windows 函数和基础类库函数和数据结构。在讲解时尽量选择了一些简单、完整的例子,以便可以研究、修改它,按自己的需要扩展在自己的程序中。

本书各章的主要内容是:

第一章至第四章介绍了包含于 Microsoft C/C++ 编译包中的设计工具。

第五章至第十四章讲述了 C/C++ 语言的基本程序设计原理。面向过程的一些章节讲述传统 C 和 C++ 程序设计原理。

第十五章至第十八章,给出了一个 C++ 面向对象设计的完整介绍。其中有定义和完整的例子,可以帮助读者完成面向对象程序的开发。

第十九章至第二十一章讲述了如何为 DOS 建立图形应用程序,如何使用重要的库函数。第十九章完整地介绍了 Microsoft 的 graphics 和 chart 库。第二十章包含了 C、C++ 库函数的详细讨论。第二十一章讲述了 C、C++ 的汇编语言接口;在第二十一章描述了如何在 C、C++ 与汇编语言之间进行参数传递,如何用汇编和低层函数与外部设备接口。

第二十二章至第二十三章介绍了 Microsoft Windows 原理,告诉读者如何使用 Microsoft C/C++ 编译器来开发 Windows 应用程序。这两章中的应用程序是传统的、基于消息用 C 编写的程序。

第二十四章和第二十五章致力于如何使用 Microsoft 基础类库去编程。通过使用强有力的 C++ 类、Microsoft 基础类库将大大缩短 Windows 应用程序的开发周期和源程序长度。

目 录

前 言

第一部分 快速启动 C/C++ 编程

第一章 MS C/C++ 编译器软件包

1.1	Windows 下的典型安装	3
1.2	Microsoft DOS 下的典型安装	4
1.3	建议	4
1.3.1	最小的软硬件要求	4
1.3.2	推荐的软硬件配置	4
1.3.3	好的安装计划	5
1.3.4	AUTOEXEC.BAT 和 CONFIG.SYS	5
1.3.5	DOS 和 Windows 下的工作	5
1.4	实用程序包	5
1.4.1	程序员工作台(PWB)	6
1.4.2	CL —— DOS 命令行编译器	6
1.4.3	PWB 集成编译器	6
1.4.4	LINK —— 连接器	6
1.4.5	EXEHDR —— EXE 文件头文件工具	6
1.4.6	NMAKE —— 工程维护	6
1.4.7	BUILD —— PWB 工程维护	7
1.4.8	LIB —— 库管理程序	7
1.4.9	HELPMAKE —— 帮助文件维护	7
1.4.10	BSCMAKE 和 SBRPACK —— 浏览器数据库维护	7
1.4.11	CVPACK —— 调试信息压缩器	7
1.4.12	IMPLIB —— 输入库管理程序	8
1.4.13	RM —— 文件转移	8
1.4.14	UNDEL —— 恢复删除	8
1.4.15	EXP —— 删除隐含子目录 DELETED	8
1.4.16	CodeView 调试器	8
1.4.17	Profiler	9
1.4.18	源浏览器(Source Browser)	9
1.4.19	Windows 开发工具	9
1.4.20	资源编译器	10
1.4.21	Spy	10
1.4.22	WXServer	10
1.4.23	QH Advisor	10

1.4.24	386MAX	10
1.5	C/C++和 Windows 编程特点	11
1.5.1	C++2.1适用范围	11
1.5.2	编译器优化	11
1.5.3	编译器存储模式	12
1.5.4	P 码	12
1.5.5	MOVE —— Microsoft 虚拟环境覆盖	13
1.5.6	基本类型库(MFC)	14
1.5.7	函数在线	15

第二章 启动程序员工作台

2.1	进入 PWB 环境	17
2.1.1	主菜单分类	18
2.1.2	书写新的应用程序	23
2.1.3	编辑应用程序	24
2.1.4	拷贝、剪切和粘贴(Copying ,Cutting 和 Pasting)	25
2.2	编译和连接	27
2.2.1	在 PWB 内工作	27
2.2.2	在 DOS 命令行下工作	29
2.3	使用 Microsoft Advisor 帮助系统	29
2.3.1	程序语句帮助	29
2.3.2	其他帮助特性	29
2.3.3	打印帮助信息	30
2.4	设置重要的 PWB 选项	32
2.4.1	键分配菜单命令	32
2.4.2	Editor Setings 菜单命令	32
2.4.3	Colors 菜单命令	35
2.4.4	TOOLS. INI 文件	36
2.5	改变 build 过程	36

第三章 启动 C/C++编译器

3.1	缺省编译器选项下的简单编译	37
3.1.1	在 PWB 下编译	38
3.1.2	在 DOS 命令行下编译	39
3.2	设置编译器和连接器开关	39
3.2.1	编译器开关	40
3.2.2	输出文件编译器选项	41
3.2.3	预处理器编译器选项	41
3.2.4	语言(Language)编译器选项	42
3.2.5	浮点编译器选项	42
3.2.6	源列表编译器选项	42
3.2.7	杂类编译器选项	43

3.2.8	宏汇编选项	43
3.2.9	连接编译器选项	44
3.2.10	优化编译器选项	44
3.2.11	存储模式编译器选项	45
3.2.12	连接器选项	45
3.3	用 Make 文件控制编译操作	47
3.3.1	PWB 工程文件	48
3.3.2	DOS NMake	52
3.4	特别编译选项	53
3.4.1	模块定义文件(.DEF)	53
3.4.2	资源编译器	54
3.5	排除错误	55
第四章	启动高级工具:CodeView, Browser 和 Profiler	
4.1	CodeView	57
4.1.1	一个简单的 DOS 例程	58
4.1.2	C++ 程序使用 CodeView	64
4.1.3	Windows 程序使用 CodeView	66
4.2	源浏览器	71
4.2.1	通用浏览器信息	71
4.2.2	符号定义查找	72
4.2.3	符号引用查找	73
4.2.4	创建调用树	74
4.3	Profiler	75
4.3.1	Profiler 批处理文件	75
4.4	代码书写形式	76

第二部分 C 和 C++ 编程基础

第五章 C 和 C++ 基础

5.1	C 的发展史	79
5.1.1	与其他语言的关系	80
5.1.2	C 语言的优势	81
5.1.3	C 的弱点	83
5.1.4	编程的规程	83
5.2	ANSI C 标准	83
5.3	C++ 的进化和面向对象的编程	85
5.4	C++ 的历史	85
5.4.1	使用 C++ 对象来进行流线代码设计	86
5.4.2	对 C 的提高	86
5.4.3	对 C 的主要提高	88
5.5	C 程序的基本元素	89

5.5.1	程序的五个基本成分	90
5.5.2	第一个C程序	90
5.5.3	第一个C++程序	91
5.5.4	第二个C程序	91
5.5.5	第二个C++程序	94
5.5.6	文件	96
第六章 数 据		
6.1	标识符	99
6.2	关键字	100
6.3	标准C和C++数据类型	101
6.3.1	字符	101
6.3.2	三种整数	103
6.3.3	unsigned 修饰符	103
6.3.4	浮点数	104
6.3.5	枚举型	105
6.4	访问修饰符	107
6.4.1	const 修饰符	107
6.4.2	#define 常量	108
6.4.3	Volatile 修饰符	108
6.4.4	const 和 volatile 一起使用	108
6.5	pascal, cdecl, near, far 和 huge 修饰符	109
6.5.1	pascal	109
6.5.2	cdecl	110
6.5.3	near, far 和 huge	111
6.6	数据类型转换	111
6.6.1	显示类型转换	112
6.7	存储类型	112
6.7.1	外层的变量说明	113
6.7.2	内层的变量说明	114
6.7.3	变量作用域	115
6.7.4	外层函数说明	116
6.8	操作符	116
6.8.1	位操作符	116
6.8.2	左移和右移	117
6.8.3	增量与减量	118
6.8.4	算术操作符	118
6.8.5	赋值操作符	119
6.8.6	复合赋值操作符	119
6.8.7	关系和逻辑操作符	121
6.8.8	条件操作符	123

6.8.9	逗号操作符	123
6.9	理解操作符优先顺序	124
6.10	标准 C 和 C++ 库函数	124
第七章 控制		
7.1	条件语句	127
7.1.1	if 语句	127
7.1.2	if - else 语句	128
7.1.3	嵌套 if - else	130
7.1.4	if - else - if 语句	131
7.1.5	? 条件语句	132
7.1.6	switch 语句	133
7.1.7	混合使用 if - else - if 和 switch 语句	139
7.2	循环语句	140
7.2.1	for 循环	141
7.2.2	while 循环	145
7.2.3	do - while 循环	147
7.2.4	break 语句	149
7.2.5	使用 CodeView	149
7.2.6	continue 语句	149
7.2.7	使用 CodeView	150
7.2.8	一起使用 break 和 continue	151
7.2.9	exit() 语句	152
7.2.10	atexit() 语句	155
第八章 编写和使用函数		
8.1	函数原型和风格	157
8.1.1	构造原型	157
8.1.2	通过值调用和通过引用调用	159
8.1.3	存储类型	161
8.1.4	作用域	161
8.1.5	递归	161
8.2	函数参数	162
8.2.1	形式参数和实际参数	162
8.2.2	void 类型的参数	163
8.2.3	字符参数	164
8.2.4	整数参数	164
8.2.5	浮点参数	165
8.2.6	双精度参数	166
8.2.7	数组参数	168
8.3	函数类型	169
8.3.1	void 函数类型	169

8.3.2	char 函数类型	171
8.3.3	int 函数类型	171
8.3.4	long 函数类型	172
8.3.5	float 函数类型	173
8.3.6	double 函数类型	174
8.4	main()的参数	175
8.4.1	字符串	175
8.4.2	整数	176
8.4.3	浮点数	177
8.5	重要的 C++ 特性	178
8.5.1	直接插入	178
8.5.2	重载	179
8.5.3	省略号(...)	180
8.6	作用域规则遇到的问题	181
8.6.1	C 程序中一个未定义的符号	182
8.6.2	使用具有文件作用域的变量	182
8.6.3	用局部变量屏蔽全局变量	183
8.6.4	C++ 的作用域问题	184
8.6.5	C++ 作用域限定操作符	185

第九章 数 组

9.1	什么是数组	187
9.2	数组和 C	187
9.3	数组说明	187
9.4	数组初始化	188
9.4.1	缺省初始化	188
9.4.2	显示初始化	189
9.4.3	不定长初始化	189
9.5	访问数组元素	190
9.6	计算数组大小(sizeof ())	192
9.7	数组下标越界	193
9.8	字符串输入输出	194
9.9	多维数组	196
9.10	数组作为函数参数	199
9.10.1	将数组传给 C 函数	199
9.10.2	数组传给 C++ 函数	201
9.11	字符串函数和字符数组	206
9.11.1	gets(), puts(), fgets(), fputs()和 sprintf()	206
9.11.2	strcpy, strcat, strncpy, strlen	208

第三部分 C++面向对象编程基础

第十章 指 针

10.1 定义指针变量	213
10.1.1 指针变量说明	213
10.1.2 简单语句使用指针变量	215
10.1.3 指针变量初始化	217
10.1.4 地址操作符的错误使用	219
10.1.5 数组指针	219
10.1.6 指针的指针	220
10.1.7 串的指针	221
10.1.8 指针运算	222
10.1.9 指针运算和数组	224
10.1.10 操作符++和--的有关事项	226
10.1.11 比较指针	226
10.1.12 可移植性	227
10.1.13 指针的大小	227
10.2 函数的指针	228
10.3 动态存储	231
10.3.1 使用空指针	233
10.4 指针和数组	236
10.4.1 串(字符型的数组)	236
10.4.2 指针数组	237
10.4.3 更多的有关指针的指针的内容	239
10.4.4 串指针的数组	244
10.5 C++引用类型	246
10.5.1 函数返回地址	247
10.5.2 使用 codeview	248
10.5.3 何时使用引用类型	248

第十一章 C语言中的 I/O

11.1 流函数	251
11.1.1 打开流	251
11.1.2 输入和输出的重定向	252
11.1.3 改变流缓冲区	253
11.1.4 关闭流	254
11.2 C中的低级输入和输出	255
11.3 字符输入和输出	255
11.3.1 使用 getc(), putc(), fgetc()和 fputc()	255
11.3.2 使用 getchar(), putchar(), fgetc()和 fputc()	256
11.3.3 使用 getch()和 putch()	256

11.4	串输入和输出.....	257
11.4.1	使用 gets(), puts(), fgetx() 和 fputs()	257
11.5	整数输入和输出.....	258
11.5.1	使用 getw(和 putw()	258
11.6	格式化输出.....	261
11.6.1	使用 printf()和 fprintf()	261
11.7	使用 fseek(), ftell()和 rewind()	265
11.7.1	使用 codeview	267
11.8	格式化输入.....	269
11.8.1	使用 scanf()、fscanf()和 sscanf().....	270
第十二章 C++的输入、输出介绍		
12.1	使用 C++精简 I/O	273
12.1.1	cin、cout 和 cerr	273
12.1.2	抽取(>>)和插入(<<)操作.....	273
12.2	从 stream.h 到 istream.h	280
12.2.1	操作符和成员函数.....	280
第十三章 结构、联合和其它数据结构		
13.1	C 和 C++结构	293
13.1.1	C 和 C++结构:语法和规则	293
13.1.2	C++结构:语法和规则的扩展	295
13.1.3	访问结构成员.....	295
13.1.4	构造一个简单的结构.....	296
13.1.5	将结构传递给函数.....	297
13.1.6	构造一个结构数组.....	299
13.1.7	使用结构指针.....	301
13.1.8	传递结构数组给函数指针.....	303
13.1.9	C++中结构的使用	305
13.1.10	对结构的附加操作	308
13.2	联合.....	310
13.2.1	联合:语法和规则	310
13.2.2	构造一个简单的联合.....	311
13.3	其它数据类型.....	312
13.3.1	使用 typedef	312
13.3.2	使用枚举类型.....	313
第十四章 高级 C 和 C++编程专题		
14.1	类型兼容.....	317
14.1.1	ANSI C 对于类型兼容的定义	317
14.1.2	什么是同一类型.....	317
14.1.3	枚举类型.....	318
14.1.4	数组类型.....	318

14.1.5	函数类型	319
14.1.6	结构和联合类型	319
14.1.7	指针类型	319
14.1.8	多源文件的相容性	319
14.2	宏	319
14.2.1	定义宏	320
14.2.2	宏和参数	320
14.2.3	宏扩展的问题	321
14.2.4	生成和使用自己的宏	322
14.2.5	编译器预置的宏	323
14.3	高级预处理语句	323
14.3.1	#ifdef 和 #endif	323
14.3.2	#undef	324
14.3.3	#ifndef	324
14.3.4	#if	324
14.3.5	#else 指令	325
14.3.6	#elif 指令	325
14.3.7	#line 指令	325
14.3.8	#error 指令	326
14.3.9	#pragma 指令	326
14.4	条件编译	326
14.5	高级预处理操作	327
14.5.1	#stringize operator(串化操作)	327
14.5.2	连接操作(##)	328
14.5.3	字符操作(#@)	328
14.6	头文件的正确使用	328
14.7	有效地使用头文件	329
14.8	预编译的头文件	330
14.8.1	产生预编译的头	330
14.8.2	在 PWB(程序员工作台)上使用预编译头文件	330
14.9	limits.h 和 float.h	330
14.10	错误处理:perror()	332
14.11	存储模式	333
14.11.1	Tiny(微)	333
14.11.2	small(小)	333
14.11.3	medium(中)	333
14.11.4	Compact(紧缩)	333
14.11.5	Large(大)	333
14.11.6	Huge(巨)	334
14.12	动态存储分配:链表	334

14.12.1	使用链表时需要考虑的事项	335
14.12.2	一个简单的链表	335
第十五章 面向对象的编程介绍		
15.1	并非是新内容	339
15.2	传统结构化设计	339
15.3	面向对象的程序设计	340
15.3.1	C++和面向对象设计	340
15.4	面向对象的术语	341
15.4.1	封装	341
15.4.2	类的层次性	342
15.5	认识C++类	343
15.5.1	一个作为私有类的结构	343
15.5.2	C++类的语法规则	347
15.5.3	一个简单的C++类	348
第十六章 C++类		
16.1	附加的类特性	353
16.1.1	一个简单的类	353
16.1.2	嵌套类	354
16.1.3	构造函数和析构函数	357
16.1.4	重载类的成员函数	363
16.1.5	使用友元函数来存取私有类变量	367
16.1.6	使用this指针	369
16.2	使用操作符重载	370
16.2.1	操作符重载和函数调用	370
16.2.2	重载的语法	370
16.3	派生类	373
16.3.1	派生类的语法	373
16.3.2	产生派生类	373
第十七章 C++的I/O		
17.1	枚举类型	379
17.2	引用变量	380
17.3	缺省参数	382
17.4	memset()	383
17.5	格式化的输出	383
17.6	C/C++ I/O 选择	387
17.7	iostream 类列表	388
17.7.1	输入流类	391
17.7.2	输出流类	392
17.7.3	缓冲流类	394
17.7.4	串流类	395

17.8	二进制文件	397
17.9	使用 extern "C" 来结合 C 和 C++ 代码	399
17.10	编写自己的操纵符	401
17.10.1	无参数的操纵符	401
17.10.2	带有一个参数的操作器	402
17.10.3	多参数的操纵符	403
第十八章 基于面向对象环境的工作		
18.1	C++ 中一个面向的对象的栈	407
18.2	C++ 中面向对象的连接表	409
18.2.1	父类的产生	410
18.2.2	派生子类	410
18.2.3	使用友元类	412
18.2.4	测试一个完整的程序	414
18.2.5	连接表输出	422
18.3	面向对象 C++ 的其它内容和汇编语言的混合编程	424

第四部分 DOS 图形、系统访问库及混合语言接口

第十九章 DOS 下的图形设计

19.1	文本和图形方式	427
19.2	使用简单的图形函数	431
19.2.1	画一个矩形	431
19.2.2	确定屏幕的分辨率	433
19.2.3	图形环境的信息	434
19.2.4	图形详解	436
19.2.5	使用基本图形函数	442
19.3	在图形模式下使用各种字体集	447
19.3.1	基本的字体属性	448
19.3.2	用不同的规格显示不同的字体	450
19.3.3	字体旋转	453
19.4	使用基本图形函数的科学和商业应用程序	457
19.4.1	画正弦曲线波形	458
19.4.2	傅立叶级数	462
19.4.3	由基本图形组成的饼图	466
19.5	表示图形库	472
19.5.1	交互式饼图	477
19.5.2	交互式直方图	480
19.5.3	交互式折线图	483
19.5.4	散点图	487
19.6	特殊的图形效果	489
19.6.1	使用 4 个视口	490

19.6.2	改变视口大小	492
19.6.3	简单的动画	494

第二十章 更具威力的编程:选择重要的 C 和 C++ 库

20.1	Microsoft C 和 C++ 头文件	501
20.2	标准库函数(stdlib. h)	502
20.2.1	实现数据转换	503
20.2.2	实现查找和排序	504
20.2.3	混合操作	506
20.3	字符函数 ctype. h)	509
20.3.1	相测是否是字母/数字、字母、ASCII 值	509
20.3.2	检测是否是控制、空白和标点字符	511
20.3.3	转换为 ASCII 码、小写字母、大写字母	512
20.4	字符串函数(string. h)	514
20.4.1	内存函数	515
20.4.2	使用字符串函数	517
20.5	数学函数(math. h)	520
20.5.1	建立三角函数值的表	521
20.6	时间函数(time. h)	523
20.6.1	时间和日期的结构和语法	523
20.7	依赖于系统的函数	528
20.7.1	bios. h 头文件	528
20.7.2	dos. h 头文件	530

第二十一章 Microsoft C/C++ 和汇编语言的混合编程

21.1	直接插入汇编语言代码	537
21.1.1	产生音响	538
21.1.2	并行端口作为通用 I/O 端口	539
21.1.3	控制 LED 发光管	540
21.2	创立 C/C++ 和汇编语言模块	542
21.2.1	传递函数参数	542
21.2.2	传递不同数据类型的参数	543
21.2.3	简单的 C 和汇编语言连接	545
21.2.4	用 C 和汇编语言连接硬件接口	548
21.2.5	由 C 向汇编语言传递数组	552

第五部分 Windows 程序设计

第二十二章 Windows 编程:概念和工具

22.1	Windows 程序设计语言	557
22.1.1	Windows 环境快速浏览	557
22.1.2	使用 Windows 的优越性	557
22.1.3	新的 Windows 可执行文件的格式	562

22.2	Windows 编程:概念和词汇	563
22.2.1	Windows 的窗口	563
22.2.2	Windows 布局	563
22.2.3	C/C++ 中的 Windows 类	565
22.2.4	OOP 及 Windows	565
22.2.5	发送和接收 Windows 消息	568
22.2.6	访问 Windows 函数	570
22.2.7	Windows 头文件:Windows.h	571
22.2.8	Windows 应用程序的组成成分	571
22.3	Microsoft C/C++ Windows 工具	572
22.3.1	使用 make 文件	572
22.3.2	创建资源	573
22.3.3	使用 Image 和 Dialog 编辑器	573
22.3.4	在命令行使用资源编译器(RC)	583
第二十三章 用 C 编写 Windows 应用程序		
23.1	应用程序框架	587
23.1.1	有效地使用句柄	587
23.1.2	考察 Windows 应用程序中的成分	587
23.2	一个简单的应用程序和示例	597
23.2.1	画椭圆	600
23.2.2	画弦	601
23.2.3	画饼图瓣	602
23.2.4	画矩形	603
23.3	使用 SWA 开发一个正弦波应用程序	605
23.4	创建 Windows 饼图应用程序	608
23.4.1	23PIE 和 23PIE.DEF 文件	608
23.4.2	23pie.h 头文件	609
23.4.3	23PIE.RC 资源文件	617
23.4.4	23PIE.C 程序	617
23.5	传统 C Windows 编程的更多信息	620
第二十四章 Windows 程序设计:基类库		
24.1	对基类库的需求	623
24.2	基类库设计的思考	624
24.3	基类库关键特性	624
24.4	一切从 CObject 开始	625
24.5	重要的基库类	626
24.6	一个简化的应用程序	627
24.6.1	用 24SIMPLE.CPP 建立一个窗口	628
24.6.2	运行 24SIMPLE.CPP	632
24.7	简化的设计保证容易维护	633