

北京希望
电脑公司

BORLAND[®] C++ 3.0

Borland C++ 3.0

程序员指南

5

徐颖 谢德刚 编译
希望 审校



海洋出版社

2-62

北京希望电脑公司 Borland C++ 3.0 系列丛书之五

Borland C++ 3.0

程序员指南

徐颖 谢德刚 编译
希望 审校

5

海洋出版社

1992 · 北京

内容简介

本系列书全面系统地介绍了 Borland 在 C++ 3.0 平台上所有产品的功能和用法。《Borland C++程序员手册》是利用 Borland C++进行编程，特别是进行 Windows 应用程序编程不可缺少的资料。

本书首先介绍了 Borland C++的词法元素、语言结构、C++的特色语法以及预处理程序的用法；接着说明了流和类库的概念、它们的用法和类库参考，并介绍了数学处理函数；为了优化 Borland C++ 3.0 的代码，对低层的硬件进行更深入的控制，讨论了怎样使用内部汇编语句、怎样与汇编语言进行接口；最后列出了 Borland C++与 ANSI 标准的差别。

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
程序员手册

徐颖 谢德刚 等编译
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：20.56 字数：464千字

1992年1月第一版 1992年1月第一次印刷

印数：1-3000

ISBN 7-5027-2609-8/TP·90 定价：17.00元

郑重说明

Borland(宝兰国际有限公司)授权我中科院九州计算机网络公司为 Borland 程序语言和应用软件之中国代理。

北京希望公司此中文手册的出版，已通过我公司与 Borland 的商讨，得到 Borland 的认可。

该手册仅限于和相应的 Borland 软件配套发行，严禁私自翻印和单独发行。

中科院九州计算机网络公司

1992年9月

地 址：北京海淀白石桥路25号

邮 编：100081

电 话：8311822 8420320

特此函告

中国科学院计算中心

九州计算机网络公司

1992.9.19

简介

本手册是一本高级程序员手册，它是为有经验的程序设计者编写的(无论在 C、C++ 或其它语言方面)。本手册提供了语言参考以及下列有关编程的信息：C++流、对象包含类、从 Microsoft C 的转换、Windows 应用程序、存贮模式、浮点运算、覆盖、视频函数、BASM、内部汇编和 ANSI 实现。

注意，代码例子都有一个 main 函数。EasyWin 使所有这些例子都工作在 Windows 下。因此，不必使用 WinMain 及其参数。

0.1 本书的内容

第一章~第四章：“词法”、“语言结构”、“C++特性”和“预处理器”，这几章描述了 C++语言。其中涉及到对 ANSI C 标准的扩充。这几章提供了规范语言的定义、引用以及 Borland C++ 中的 C 语法和 C++ 语法。本简介还介绍了第一章到第四章的其它信息。

第五章：“使用 C++流”，告诉用户如何使用 C++ 2.1 流库。

第六章：“Container 类库”，告诉用户如何在程序中使用 Borland C++ 对象包含类(包括模板)。

第七章：“从 Microsoft C 的转换”为从 Microsoft C 程序转换为 Borland C++ 程序提供了一些指导。

第八章：“构造 Windows 应用程序”，指导用户进行 Windows 编程。

第九章：“DOS 存贮管理”，讲述了存贮模式、覆盖和混合模式编程。

第十章：“数学运算”，讲述了浮点运算和 BCD 运算。

第十一章：“视频函数”，用来处理 Borland C++ 的正文和图形。

第十二章：“BASM 与内部汇编”，介绍了如何编写汇编语言程序，以使它们在 Borland C++ 程序中调用时正常执行。本章也介绍了有关 IDE 中的内部汇编器的信息。

附录 A：“ANSI 实现标准”，描述了 ANSI C 标准中定义不完善或者没有定义的一面。该附录介绍了 Borland C++ 是如何实现它们的。

0.2 规范定义介绍

第一章到第四章提供了对 Borland C++ 中的 C 语言和 C++ 语言的规范描述。这几章结合起来完整地描述了 Borland C++ 语言，提供了 Borland C++ 的 C 语言和 C++ 语言的规范语言定义、引用以及语法。但这几章没有说明如何使用这种语言。我们使用了改进的巴科斯—诺尔表示式来表示语法，在必要的地方提供了简洁的解释和程序例子。其组织方式如下：

- 第一章，“词法”，介绍了 Borland C++ 如何识别单词符号。词法涉及到类的单词的单元(称为单词符号，它由语言识别)的不同类别。

- 第二章，“语言结构”，解释了如何使用 Borland C++ 的单词符号。语言结构详细描述了单词符号合法的组合方式，单词符号可以组成表达式、语句和其它意义的单元。
- 第三章，“C++特性”，介绍了 C++ 所特有的一些方面。
- 第四章，“预处理器”，介绍了预处理器，包括宏、包含文件、条件编译及其它许多有用的成份。

Borland C++ 完全实现了 AT&T 公司的 C++ 2.1 版本，后者是由 AT&T Bell 实验室的 Bjarne Stroustrup 开发的面向对象的 C 语言子集。本手册将 AT&T 的早期版本当作 C++ 2.0 来看。除了提供许多新的功能和特征之外，C++ 还在某种程序上扩充了 C 语言。在这些章节中，我们标明了这些区别。第四章更详细地讨论了由 C++ 演变而来的 Borland C++ 的全部功能。

Borland C++ 也完全实现了 ANSI C 标准，如文中所述，其中也有某些扩充。用户可以设置编译选项，以便在遇到扩充的部分时可以得到警告。也可以通过设置编译选项，使编译程序象对待通常的标识符一样对待 Borland C++ 中的扩充关键字(见《用户指南》第五章“命令行编译器”)。

Borland C++ 中也扩充了由处理非标准的与具体实现细节无关的功能的 ANSI C 所提供的 #pragma 伪指令。

0.2.1 语法与术语

语法定义由所定义的非终结的单词符号的名字后跟冒号(:)组成。在其后面通常可以选择性地跟以若干行，但如果前面有短语“one of”，则可以将若干行内容放在一行上。例如：

external-definition:

function-definition

declaration

octal-digit: one of

0 1 2 3 4 5 6 7

某一结构中的可选元素被括在尖括号内：

integer-suffix:

unsigned-suffix <long-suffix>

在本书的各章节中，单词“argument”(实参)用于表示在函数调用过程中所传递的实际值，“parameter”(形参)用于表示定义于函数头部的存放值的变量。

目 录

简介	1
0.1 本书的内容	1
0.2 规范定义介绍	1
0.2.1 语法与术语	2
第一章 词法	3
1.1 空白符	3
1.1.1 用\续行	3
1.1.2 注释	4
1.1.2.1 C注释	4
1.1.2.2 嵌套注释	4
1.1.2.3 C++注释	4
1.1.2.4 注释分界符及空白符	4
1.2 单词符号	5
1.2.1 关键字	5
1.2.2 标识符	7
1.2.2.1 命名和长度限制	7
1.2.2.2 标识符及其大小写	7
1.2.2.3 唯一性和作用域	7
1.2.3 常量	8
1.2.3.1 整型常量	9
1.2.3.2 字符常量	10
1.2.3.3 浮点常量	12
1.2.3.4 文字串	14
1.2.3.5 常量及其内部表示	15
1.2.3.6 常量表达式	16
1.2.4 分隔符	17
1.2.4.1 方括号	17
1.2.4.2 圆括号	17
1.2.4.3 花括号	17
1.2.4.4 逗号	17
1.2.4.5 分号	18
1.2.4.6 冒号	18
1.2.4.7 省略号	18
1.2.4.8 星号(指针声明)	19
1.2.4.9 等号(初始化符号)	19

1.2.4.10 #号(预处理伪指令).....	19
第二章 语言结构.....	20
2.1 声明.....	20
2.1.1 对象.....	20
2.1.2 左值(Lvalues).....	20
2.1.2.1 右值.....	21
2.1.3 类型和存储类.....	21
2.1.4 作用域.....	21
2.1.4.1 块作用域.....	21
2.1.4.2 函数作用域.....	21
2.1.4.3 函数原型作用域.....	21
2.1.4.4 文件作用域.....	21
2.1.4.5 类作用域(C++).....	22
2.1.4.6 作用域和名空间.....	22
2.1.5 可见性.....	22
2.1.6 生存期.....	22
2.1.6.1 静态生存期.....	23
2.1.6.2 局部生存期.....	23
2.1.6.3 动态生存期.....	23
2.1.7 翻译单元.....	23
2.1.8 连接.....	24
2.1.8.1 名变换.....	24
2.2 声明语法.....	25
2.2.1 临时定义.....	25
2.2.2 可能的声明.....	25
2.2.3 外部声明和定义.....	29
2.2.4 类型说明符.....	31
2.2.5 类型分类.....	31
2.2.5.1 void 类型.....	32
2.2.6 基类型.....	32
2.2.6.1 整型.....	32
2.2.6.2 浮点类型.....	33
2.2.6.3 标准转换.....	33
2.2.6.4 特殊的 char、int 和 enum 间的转换.....	34
2.2.7 初始化.....	34
2.2.7.1 数组、结构与联合.....	35
2.2.8 简单声明.....	36
2.2.9 存储类说明符.....	36
2.2.9.1 存储类说明符 auto 的使用.....	36
2.2.9.2 存储类说明符 extern 的使用.....	36

2.2.9.3	存储类说明符 register 的使用	36
2.2.9.4	存储类说明符 static 的使用	37
2.2.9.5	存储类说明符 typedef 的使用	37
2.2.10	修饰符	37
2.2.10.1	const 修饰符	38
2.2.10.2	interrupt 函数修饰符	39
2.2.10.3	volatile 修饰符	39
2.2.10.4	cdecl 与 pascal 修饰符	40
2.2.10.5	指针修饰符	41
2.2.10.6	函数类型修饰符	41
2.2.11	复杂声明与声明符	41
2.3	指针	43
2.3.1	指向对象的指针	43
2.3.2	指向函数的指针	43
2.3.3	指针声明	43
2.3.4	指针与常量	44
2.3.5	指针算术运算	45
2.3.6	指针转换	45
2.3.7	C++ 引用声明	46
2.4	数组	46
2.5	函数	46
2.5.1	声明与定义	46
2.5.2	声明与原型	47
2.5.3	定义	48
2.5.4	形参声明	49
2.5.5	函数调用和参数转换	49
2.6	结构	49
2.6.1	无标结构与 typedef	50
2.6.2	结构成员声明	50
2.6.3	结构与函数	50
2.6.4	结构成员存取	51
2.6.5	结构字对齐	52
2.6.6	结构名空间	52
2.6.7	不完整声明	53
2.6.8	位字段	53
2.7	联合	54
2.7.1	无名联合(仅 C++)	55
2.7.2	联合声明	55
2.8	枚举	55
2.9	表达式	57

2.9.1	表达式与 C++	61
2.9.2	计算顺序	61
2.9.3	出错与溢出	62
2.10	运算符语义	62
2.11	运算符描述	62
2.11.1	单目运算符	63
2.11.2	二目运算符	63
2.11.2.1	加法类算符	63
2.11.2.2	乘法类算符	63
2.11.2.3	移位算符	63
2.11.2.4	按位算符	63
2.11.2.5	逻辑算符	63
2.11.2.6	赋值算符	63
2.11.2.7	关系算符	64
2.11.2.8	相等算符	64
2.11.2.9	成员选择算符	64
2.11.2.10	类成员算符	64
2.11.2.11	条件算符	64
2.11.2.12	逗号算符	64
2.11.3	后缀和前缀运算符	64
2.11.3.1	数组下标运算符[]	64
2.11.3.2	函数调用操作符()	65
2.11.3.3	结构/联合成员运算符.(圆点)	65
2.11.3.4	结构/联合指针运算符->	65
2.11.3.5	后缀增量运算符++	65
2.11.3.6	后缀减量运算符--	65
2.11.4	增量和减量运算符	65
2.11.4.1	前缀增量运算符	65
2.11.4.2	前缀减量运算符	66
2.11.5	单目运算符	66
2.11.5.1	取地址运算&	66
2.11.5.2	间接引用运算符*	66
2.11.5.3	单目加运算符+	66
2.11.5.4	单目减运算符-	67
2.11.5.5	按位补运算符~	67
2.11.5.6	逻辑非运算符!	67
2.11.6	sizeof 运算符	67
2.11.7	乘法类运算符	67
2.11.8	加法类运算符	68
2.11.8.1	加法运算符+	68

2.11.8.2	减法运算符	68
2.11.9	按位移位运算符	68
2.11.9.1	按位移位运算符(<<和>>)	69
2.11.10	关系运算符	69
2.11.10.1	小于运算符<	69
2.11.10.2	大于运算符>	69
2.11.10.3	小于等于运算符<=	70
2.11.10.4	大于等于运算符>=	70
2.11.11	相等类运算符	70
2.11.11.1	等于运算符==	70
2.11.11.2	不等于运算符!=	70
2.11.12	按位与运算符&	71
2.11.13	按位异或运算符^	71
2.11.14	按位同或运算符	71
2.11.15	逻辑与运算符&&	71
2.11.16	逻辑或运算符	72
2.11.17	条件运算符?:	72
2.11.18	赋值运算符	73
2.11.18.1	简单赋值运算符=	73
2.11.18.2	复合赋值运算符	73
2.11.19	逗号运算符	73
2.11.20	C++运算符	74
2.12	语句	74
2.12.1	块	76
2.12.2	标号语句	76
2.12.3	表达式语句	76
2.12.4	选择语句	76
2.12.4.1	if语句	76
2.12.4.2	switch语句	77
2.12.5	循环语句	77
2.12.5.1	While语句	77
2.12.5.2	do while语句	78
2.12.5.3	for语句	78
2.12.6	跳转语句	79
2.12.6.1	break语句	79
2.12.6.2	continue语句	79
2.12.6.3	goto语句	79
2.12.6.4	return语句	79
第三章	C++特性	81
3.1	引用	81

3.1.1	简单引用	81
3.1.2	引用参数	81
3.2	作用域访问运算符	82
3.3	new 与 delete 运算符	83
3.3.1	错误处理	83
3.3.2	带数组的运算符 new	84
3.3.3	带数组的运算符 delete	84
3.3.4	::operator new	84
3.3.5	带 new 运算符的初始值	84
3.4	类	85
3.4.1	类名	85
3.4.2	类类型	85
3.4.3	类名作用域	85
3.4.4	类对象	86
3.4.5	类成员表	86
3.4.6	成员函数	86
3.4.7	关键字 this	86
3.4.8	内部函数	87
3.4.9	静态成员	87
3.4.10	成员作用域	88
3.4.10.1	嵌套类型	89
3.4.10.2	成员访问控制	90
3.4.11	基类与派生类存取	91
3.5	虚基类	92
3.6	类的友元	93
3.7	构造函数(constructors)与析构函数(destructors)	94
3.8	构造函数	95
3.8.1	默认构造函数	95
3.8.2	拷贝构造函数	96
3.8.3	构造函数的重载	96
3.8.4	构造函数的调用次序	97
3.8.5	类初始化	98
3.9	析构函数	100
3.9.1	析构函数的调用	101
3.9.2	atexit、#pragma exit 与析构函数	101
3.9.3	exit 与析构函数	101
3.9.4	abort 与析构函数	101
3.9.5	虚析构函数	102
3.10	重载运算符	103
3.11	运算符函数	104

3.11.1	重载运算符与继承	104
3.11.2	重载 new 和 delete	104
3.11.3	重载单目运算符	105
3.11.4	重载二目运算符	106
3.11.5	重载赋值运算符=	106
3.11.6	重载函数调用运算符()	106
3.11.7	重载下标运算符	106
3.11.8	重载类成员存取运算符	107
3.12	虚函数	107
3.13	抽象类	108
3.14	C++作用域	109
3.14.1	类作用域	109
3.14.2	隐藏	109
3.14.3	C++作用域规则小结	109
3.15	模板	110
3.15.1	函数模板	111
3.15.1.1	重设模板函数	112
3.15.1.2	隐式的和显式的模板函数	112
3.15.2	类模板	113
3.15.2.1	参数	114
3.15.2.2	尖括号	114
3.15.2.3	类型安全的类属类	114
3.15.2.4	消去指针	115
3.15.3	模板编译开关	116
3.15.3.1	使用模板开关	116
第四章	预处理器	118
4.1	空指令#	120
4.2	#define 与#undef 指令	120
4.2.1	简单的#define 宏	120
4.2.2	#undef 指令	121
4.2.3	-D 与 U 选择	122
4.2.4	定义选择	122
4.2.5	关键字与保护字	122
4.2.6	带参数的宏	122
4.3	文件包含指令#include	124
4.3.1	<头文件名>形式的头文件搜索	125
4.3.2	“头文件名”形式的头文件搜索	125
4.4	条件编译	125
4.4.1	#if、#elif、#else、和#endif 条件指令	125
4.4.1.1	操作符 defined	126

4.4.2	#ifdef 和 #ifndef 条件指令	126
4.5	#line 行控制指令	127
4.6	#error 指令	128
4.7	#pragma 指令	128
4.7.1	#pragma argsused	129
4.7.2	#pragma exit 与 #pragma startup	129
4.7.3	#pragma hdrfile	130
4.7.4	#pragma hdrstop	130
4.7.5	#pragma inline	130
4.7.6	#pragma intrinsic	130
4.7.7	#pragma option	130
4.7.8	#pragma saveregs	131
4.7.9	#pragma warn	131
4.8	预定义的宏	132
4.8.1	__BCPLUSPLUS__	132
4.8.2	__BORLANDC__	132
4.8.3	__CDECL__	132
4.8.4	__CPLUSPLUS__	132
4.8.5	__DATE__	132
4.8.6	__DLL__	133
4.8.7	__FILE__	133
4.8.8	__LINE__	133
4.8.9	__MSDOS__	133
4.8.10	__OVERLAY__	133
4.8.11	__PASCAL__	133
4.8.12	__STDC__	133
4.8.13	__TCPLUSPLUS__	133
4.8.14	__TEMPLATES__	133
4.8.15	__TIME__	134
4.8.16	__TURBOC__	134
4.8.17	__Windows__	134
第五章	使用 C++ 流	135
5.1	什么是流?	135
5.2	iostream 库	135
5.2.1	streambuf 类	135
5.2.2	ios 类	135
5.3	输出	136
5.3.1	基本的类型	137
5.3.2	输出格式	137
5.3.3	操纵符	138

5.3.4	填充与补空	139
5.4	输入	140
5.5	用户定义类型的 I/O	140
5.6	简单文件 I/O	141
5.7	串流处理	142
5.8	屏幕输出流	143
5.9	流类参考	145
5.9.1	conbuf	145
5.9.1.1	成员函数	145
5.9.2	constream	146
5.9.2.1	成员函数	146
5.9.3	filebuf	146
5.9.3.1	成员函数	146
5.9.4	fstream	147
5.9.4.1	成员函数	147
5.9.5	fstreambase	147
5.9.5.1	成员函数	148
5.9.6	ifstream	148
5.9.6.1	成员函数	148
5.9.7	ios	148
5.9.7.1	数据成员	149
5.9.7.2	成员函数	149
5.9.8	iostream	150
5.9.9	iostream_withassign	150
5.9.9.1	成员函数	151
5.9.10	istream	151
5.9.10.1	成员函数	151
5.9.11	istream_withassign	152
5.9.11.1	成员函数	152
5.9.12	istrstream	152
5.9.13	ofstream	152
5.9.13.1	成员函数	152
5.9.14	ostream	152
5.9.14.1	成员函数	153
5.9.15	ostream_withassign	153
5.9.15.1	成员函数	153
5.9.16	ostrstream	153
5.9.16.1	成员函数	153
5.9.17	streambuf	153
5.9.17.1	成员函数	154

5.9.18	strstreambase	155
5.9.18.1	成员函数	155
5.9.19	strstreambuf	156
5.9.19.1	成员函数	156
5.9.20	strstream	156
5.9.20.1	成员函数	156
第六章	container 类库	157
6.1	与 2.0 比有哪些变化?	157
6.2	为何设置两种类库?	158
6.3	Container 基本概念	158
6.3.0.1	基于对象类其它类	160
6.3.1	类分类	160
6.3.2	Non-container 类	160
6.3.2.1	Error 类	160
6.3.2.2	Sortable 类	160
6.3.2.3	Association 类	161
6.3.3	Container 类	161
6.3.4	Container 和所有权	161
6.3.5	Container Iterators	162
6.3.6	Sequence class	163
6.3.7	Collections	163
6.3.7.1	无序集合	163
6.3.7.2	有序集合	164
6.4	BIDS 模板库	164
6.4.1	模板、类和容器	164
6.4.2	Container 实现	164
6.4.3	模板解决方法	165
6.4.3.1	ADT 和 FDS	165
6.4.3.2	类模板	165
6.4.4	Container 类兼容性	168
6.4.5	头文件	169
6.4.6	调节应用	169
6.4.7	FDS 的实现	170
6.4.8	ADT 的实现	173
6.5	类库目录	177
6.5.1	INCLUDE 目录	177
6.5.2	SOURCE 目录	177
6.5.3	LIB 目录	177
6.5.4	EXAMPLES 目录	178
6.6	先决条件和检查	178

6.7	Container 类参考	179
6.7.1	AbstractArray[抽象数组]	179
6.7.1.1	数据成员	179
6.7.1.2	成员函数	180
6.7.1.3	友元	181
6.7.2	Array	181
6.7.2.1	例子	181
6.7.2.2	成员函数	182
6.7.3	ArrayIteator	183
6.7.3.1	成员函数	183
6.7.4	Association	183
6.7.4.1	成员函数	184
6.7.4.2	例子	184
6.7.5	Bag	185
6.7.5.1	成员函数	185
6.7.6	BaseDate	187
6.7.6.1	成员函数	187
6.7.7	BaseTime	188
6.7.7.1	成员函数	188
6.7.8	Btree	189
6.7.8.1	成员函数	189
6.7.8.2	友元	190
6.7.9	Btreelteator	190
6.7.9.1	成员函数	190
6.7.10	Collection	191
6.7.10.1	成员函数	191
6.7.11	Container	192
6.7.11.1	成员函数	193
6.7.11.2	友元	194
6.7.12	ContainerIteator	194
6.7.12.1	成员函数	194
6.7.13	Date	195
6.7.13.1	成员函数	195
6.7.14	Deque	195
6.7.14.1	例子	196
6.7.14.2	成员函数	197
6.7.15	Dictionary	197
6.7.15.1	成员函数	198
6.7.16	DoubleList	198
6.7.16.1	成员函数	198