


Borland C++ & Turbo C++

# 程序员手册

 李振格 主编

 北京航空航天大学出版社

Borland C++ & Turbo C++

# 程序员手册

李 振 格 主 编

北京航空航天大学出版社

## 内 容 简 介

Borland C++和 Turbo C++支持多重覆盖窗口和自动覆盖管理,既是面向对象的程序设计语言,又是一个编辑、编译、调试、运行和剖视优化一体化的优秀的语言开发环境,且与 Turbo C 相兼容。

《用户手册》、《程序员手册》和《库函数参考手册》是上述两软件的最新系列参考书。《程序员手册》介绍了 C++ 语言的所有语法成分,提供了 C++ 的关于内存模式、混合模型编程、流 I/O、视频函数、浮点数处理、覆盖、编写 Windows 3.0 应用程序和错误信息的技术参考资料,用示例说明了利用 Turbo C、Borland C++ 和 Turbo C++ 先进技术进行程序设计的方法。在书中还通过丰富的范例由浅入深地讨论了类、对象、层次、重载、封装、继承、虚函数和多态性等概念,说明了利用 C++ 进行面向对象程序设计的方法。书中的所有程序都通过调试。

- 书 名: Borland C++ 及 Turbo C++ 程序员手册
- 主 编: 李 振 枋
- 责任编辑: 樊 友
- 出版者: 北京航空航天大学出版社出版
- 印刷者: 朝阳科印刷厂印刷
- 发 行 者: 新华书店总店科技发行所发行 全国各地书店经售
- 开 本: 78X1092 1/16
- 印 张: 9.5
- 字 数: 1011千字
- 印 数: 1-5000册
- 版 次: 1992年1月第一版
- 印 次: 1992年1月第一次印刷
- 书 号: ISBN 7-81012-298-3/TP·061
- 价 格: 30.35元

## 前 言

在 80 年代末期,一种新的程序设计方法开始形成,这就是所谓面向对象的程序设计。面向对象程序设计(Object Oriented Programming,简称 OOP)吸收了结构化程序设计的所有优点,并为程序设计者提供了分析和解决应用问题的一种新的方法。C++ 是 C 的面向对象版本, C 素以效率高、功能丰富及代码精练而著称。C++ 保留了 C 的这些重要性质,增加了面向对象程序设计的特征。

1988 年 Borland 公司在深受欢迎的 Turbo C 的基础上,推出最新的面向对象的程序设计软件包——Borland C++ 2.0 和 Turbo C++ 1.x。它们继承并发扬了原来 Turbo C 集成环境的优良特性,包含了面向对象的思想 and 设计方法,是目前国际上最受欢迎的面向对象程序设计软件包。Borland C++ 还可以为广为流行的 Windows 3.0 编写应用程序。

对编程者来说,从使用一般的 C 语言到使用 Turbo C 语言,是从一种单一的语言,转向编辑、编译、调试和运行一体化的具有联机帮助和热键触发的集成式的语言开发环境; Borland C++ 和 Turbo C++ 的环境支持鼠标器、支持多重覆盖窗口、支持自动管理覆盖、支持内部汇编码,集编辑、编译、调试、运行和剖视优化等功能于一体,并具有联机帮助和热键触发的特点。因此从使用 Turbo C 语言到使用 Borland C++ 或 Turbo C++ 语言,是从优良的环境转向优秀的环境,是从一般的模块化程序设计转向面向对象程序设计,标志着程序设计方法的一次飞跃。

由于 Borland C++ 和 Turbo C++ 从 Turbo C 发展而来, Borland C++ 和 Turbo C++ 都与 Turbo C 相兼容,因此用 Turbo C 编写的程序很容易移植到 Borland C++ 或 Turbo C++ 中来。Borland C++ 和 Turbo C++ 的库函数和集成环境都与 Turbo C 很相似或者说几乎相同,因此在本套书的编写过程中,许多章节都先介绍 Turbo C,然后再介绍 Borland C++ 和 Turbo C++。这样,一方面可以让 Turbo C 程序员能很快掌握 Borland C++ 或 Turbo C++, 另一方面读者也可把本套书作为 Turbo C 1.x 和 Turbo C 2.0 的《用户手册》、《库函数手册》和《程序员手册》。

《用户手册》、《程序员手册》和《库函数参考手册》是在参考 Borland C++ 和 Turbo C++ 两个软件的所有联机资料 and 许多关于 Borland C++ 和 Turbo C++ 程序设计方法的书籍后编写而成的,是掌握和使用 Borland C++ 和 Turbo C++, 编写 C++ 程序的不可缺少的书籍。

《用户手册》介绍了 Borland C++ 和 Turbo C++ 软件的安装步骤,说明了集成环境、实用程序和资源管理程序的使用方法,描述了编辑器宏语言,指出了 Microsoft 程

序员使用 Borland C++ 和 Turbo C++ 的注意事项。其中着重说明了集成环境编辑器、调试器和项目管理程序的用法。

《程序员手册》介绍了 C++ 语言的所有语法成分，提供了 C++ 的关于内存模式、混合模型编程、流 I/O、视频函数、浮点数处理、覆盖、编写 Windows 3.0 应用程序和错误信息的技术参考资料，用示例说明了利用 Turbo C、Borland C++ 和 Turbo C++ 先进技术进行程序设计的方法。在书中还通过丰富的范例由浅入深地讨论了类、对象、层次、重载、封装、继承、虚函数和多态性等概念，说明了利用 C++ 进行面向对象程序设计的方法。

《库函数参考手册》介绍了库函数的分类，C 的主函数(main)的设计和注意事项，并按字母顺序介绍了 Borland C++ 和 Turbo C++ 的每个库函数，它们的功能、用法、原型所在的头文件、返回值、相关函数和应用范例等信息，在第二章和第四章中介绍了类库和全局变量的参考和用法。书中每个库函数的说明都附有范例。

本套书中的所有程序都通过调试。

虽然在国外，特别在美国，面向对象程序设计方法已发展到实用的水平，许多软件公司都已采用 C++ 语言编写软件，但在国内，面向对象的概念还处于理论的初级阶段，许多概念和术语都没有统一，有些概念还没有明确，加上编者水平有限，书中不妥之处恳请广大读者指正。

本书在编写过程中，得到了许多同志的帮助和支持，朱海斌和刘晓东参加了本套书中部分章节的编写工作。章忆文、刘云、汪亚文、吕良双、李兵、黄磊光、杨金力、张军也帮助整理了一些文稿。

在出版的过程中，得到了北航蔡德林、杨昌竹等许多老师的许多有益的指导，在此表示衷心的感谢。

李振格

1991年6月于北航

# 目 录

## 前 言

## 第一部分 C++语言

<b>第一章 语言形式与词法符号</b> .....	<b>2</b>
1.1 语法形式.....	2
1.2 词法与短语结构的语法.....	2
1.3 空 白.....	3
1.3.1 行分隔符.....	3
1.3.2 注 释.....	3
1.4 词法符号.....	4
1.4.1 关键字.....	5
1.4.2 标识符.....	6
1.4.3 常 量.....	7
<b>第二章 操作符和表达式</b> .....	<b>15</b>
2.1 什么是操作符.....	15
2.2 单目操作符.....	16
2.3 双目操作符.....	16
2.3.1 加法类操作符.....	16
2.3.2 乘法类操作符.....	16
2.3.3 移位操作符.....	16
2.3.4 按位操作符.....	16
2.3.5 逻辑操作符.....	16
2.3.6 赋值操作符.....	17
2.3.7 关系操作符.....	17
2.3.8 相等类操作符.....	17
2.3.9 分量选择操作符.....	17
2.3.10 类成员操作符.....	17
2.4 三目操作符.....	17
2.5 标点符号.....	18
2.5.1 中括号.....	18
2.5.2 括 号.....	18
2.5.3 大括号.....	18
2.5.4 逗 号.....	18
2.5.5 分 号.....	19
2.5.6 冒 号.....	19
2.5.7 省略号.....	19
2.5.8 星号(指针说明).....	20

2.5.9	等号(赋值)	20
2.5.10	#号(预处理程序指令)	20
2.6	操作符语义	20
2.6.1	后缀和前缀操作符	20
2.6.2	单目操作符	22
2.6.3	sizeof 操作符	23
2.6.4	乘法类操作符	23
2.6.5	加法类操作符	24
2.6.6	按位移位操作符	24
2.6.7	关系操作符	25
2.6.8	相等类操作符	26
2.6.9	位运算操作符	26
2.6.10	逻辑运算符	27
2.6.11	条件操作符?:	28
2.6.12	赋值操作符	28
2.6.13	逗号操作符	29
2.7	高级运算符的使用实例	30
2.7.1	位运算符	30
2.7.2	移位运算符	31
2.7.3	?:运算符	35
2.7.4	C 和 C++ 语言的简写	37
2.7.5	逗号运算符	37
2.7.6	运算符优先级表	37
2.8	表达式	38
<b>第三章</b>	<b>说    明</b>	<b>41</b>
3.1	有关概念	41
3.1.1	对    象	41
3.1.2	左    值	41
3.1.3	右    值	42
3.1.4	类型与存储类	42
3.1.5	作用域	42
3.1.6	可见性	43
3.1.7	生存期	43
3.1.8	编译单元	44
3.1.9	连    接	44
3.2	说明的语法	45
3.2.1	暂时定义	45
3.2.2	可能的说明	46
3.3	类型说明	51
3.3.1	外部说明与定义	51

3.3.2	类型指明符	51
3.3.3	类型分类	51
3.3.4	基本类型	52
3.3.5	标准转换	56
3.3.6	特殊的 char、int 与 enum 间的转换	56
3.3.7	初始化	56
3.4	简单说明	59
3.5	存储类指明符	59
3.5.1	存储类指明符 auto 的使用	59
3.5.2	存储类指明符 extern 的使用	59
3.5.3	存储类指明符 register 的使用	60
3.5.4	存储类指明符 static 的使用	60
3.5.5	存储类指明符 typedef 的使用	60
3.6	修饰符	61
3.6.1	const 修饰符	61
3.6.2	中断函数修饰符	62
3.6.3	volatile 修饰符	62
3.6.4	cdecl 与 pascal 修饰符	62
3.6.5	指针修饰符	64
3.6.6	函数类型修饰符	64
3.7	复杂说明与说明符	64
第四章	程序控制语句	66
4.1	程序控制语句的语法	66
4.1.1	带标号语句	67
4.1.2	复合语句	68
4.1.3	表达式语句	68
4.1.4	选择语句	68
4.1.5	循环语句	69
4.1.6	跳转语句	70
4.2	if 语句	71
4.2.1	else 语句的用法	73
4.2.2	if-else-if 阶梯的用法	73
4.2.3	条件表达式	75
4.2.4	if 语句的嵌套结构	75
4.3	switch 语句	76
4.3.1	default 语句的用法	78
4.3.2	break 语句的用法	78
4.3.3	switch 语句的嵌套结构	79
4.4	循环	81
4.5	for 循环	82



4.5.1	for 循环的灵活用法	83
4.5.2	无穷的 for 循环	85
4.5.3	无穷 for 循环的中断	85
4.5.4	空循环的用法	85
4.6	while 循环语句	85
4.7	do while 循环	87
4.8	循环嵌套	88
4.9	循环中断	90
4.10	continue 语句	92
4.11	goto 语句	93
<b>第五章</b>	<b>指    针</b>	<b>95</b>
5.1	指针的语法规则	95
5.1.1	什么是指针	95
5.1.2	指针说明	96
5.1.3	指针与常量	96
5.1.4	指针算术运算	97
5.1.5	指针转换	98
5.1.6	C++ 引用说明	98
5.2	指针是地址	98
5.3	指针变量	98
5.4	指针操作符	98
5.5	指针表达式	100
5.5.1	指针赋值	100
5.5.2	指针运算	101
5.5.3	指针比较	102
5.6	指针和数组	102
5.6.1	索引指针	103
5.6.2	指针和字符串	104
5.6.3	如何得到一个数组元素的地址	105
5.6.4	指针数组	105
5.6.5	一个使用数组和指针的实例	107
5.7	指针的指针	111
5.8	指针的初始化	112
5.9	指针的一些问题	113
<b>第六章</b>	<b>高级数据类型——数组、结构、位域、联合和枚举</b>	<b>115</b>
6.1	高级数据类型的语法规则	115
6.1.1	数    组	115
6.1.2	结    构	115
6.1.3	位    域	119
6.1.4	联    合	120

6.1.5	枚 举	121
6.2	数 组	122
6.2.1	一维数组	122
6.2.2	字符串	124
6.2.3	二维数组	129
6.2.4	多维数组	131
6.2.5	数组初始化	131
6.2.6	一个水下搜索游戏	133
6.3	结 构	136
6.3.1	访问结构元素	137
6.3.2	结构数组	138
6.3.3	结构赋值	145
6.3.4	将结构传递给函数	146
6.3.5	结构指针	147
6.3.6	结构内部的数组和结构	151
6.4	位 域	151
6.5	联合(union)	154
6.6	枚 举	158
6.7	使用 sizeof 来确保可移植性	160
6.8	typedef	161
<b>第七章</b>	<b>函 数</b>	<b>163</b>
7.1	函数的初步概念	163
7.1.1	说明与定义	163
7.1.2	说明与原型	163
7.1.3	定 义	164
7.1.4	形参说明	165
7.1.5	函数调用与参数转换	165
7.2	return 语句	166
7.2.1	从一个函数中返回	166
7.2.2	返回值	167
7.2.3	函数返回非整型值	169
7.2.3.1	函数原型的使用	169
7.2.3.2	返回指针	170
7.2.3.3	函数返回 void 类型	171
7.3	有关函数原型的进一步说明	172
7.3.1	参数不匹配	172
7.3.2	使用头文件	173
7.3.3	无任何参数的函数原型	173
7.3.4	有关旧式 C 程序	174
7.4	作用域规则	174

7.4.1	局部变量	175
7.4.2	形式参数	176
7.4.3	全局变量	176
7.4.4	有关作用域的最后一个例子	177
7.5	有关函数的参数和自变量的更详尽说明	179
7.5.1	赋值调用和赋地址调用	179
7.5.2	一个赋地址调用的建立	180
7.5.3	数组与函数调用	180
7.6	argc,argv 和 env——main 中的参数	184
7.7	从 main() 中返回值	186
7.8	递归	187
7.9	参数说明的传统形式和现代形式的比较	188
7.10	对一些影响函数的效率和实用性问题的讨论	189
7.10.1	参数和通用函数	189
7.10.2	效率	190
<b>第八章</b>	<b>C++ 语言特性</b>	<b>191</b>
8.1	引用	191
8.1.1	引用参数	191
8.2	作用域存取操作符	193
8.3	new 与 delete 操作符	193
8.3.1	关于数组的 new 操作符	194
8.3.2	::operator new	194
8.3.3	带有 new 操作符的初始值	194
8.4	类	194
8.4.1	类名	194
8.4.2	类类型	194
8.4.3	类名作用域	195
8.4.4	类对象	195
8.4.5	类成员表	195
8.4.6	成员函数	196
8.4.7	关键字 this	196
8.4.8	内部函数	196
8.4.9	静态成员	197
8.4.10	成员作用域	198
8.4.11	基类与派生类存取	199
8.5	虚基类	201
8.6	类的友元	202
8.7	构造函数(constructors)与析构函数(destructors)简述	203
8.8	构造函数	204
8.8.1	缺省构造函数	204

8.8.2	拷贝构造函数	205
8.8.3	构造函数的重载	205
8.8.4	构造函数的调用次序	206
8.8.5	类的初始化	207
8.9	析构函数	209
8.9.1	析构函数的调用	209
8.9.2	atexit、#pragma exit 与析构函数	210
8.9.3	exit 与析构函数	210
8.9.4	abort 与析构函数	210
8.9.5	虚析构函数	211
8.10	重载操作符	211
8.11	操作符函数	212
8.11.1	重载操作符和继承	213
8.11.2	重载 new 和 delete	213
8.11.3	重载单目操作符	214
8.11.4	重载赋值操作符 =	214
8.11.5	重载函数调用操作符()	215
8.11.6	重载下标操作符[]	215
8.11.7	重载类成员存取操作符->	215
8.12	虚函数	215
8.13	抽象类	216
8.14	C++ 作用域	217
8.14.1	类作用域	218
8.14.2	隐藏	218
8.14.3	C++ 作用域规则小结	218
<b>第九章</b>	<b>Borland C++ &amp; Turbo C++ 预处理程序指令</b>	<b>219</b>
9.1	空指令#	221
9.2	#define 与#undef 指令	221
9.2.1	简单的#define 宏	221
9.2.2	#undef 指令	222
9.2.3	-D 与-U 选择项	223
9.2.4	关键字与保护字	223
9.2.5	带参宏	223
9.3	文件包含指令#include	225
9.3.1	<头名>形式的头文件搜索	226
9.3.2	“头名”形式的头文件搜索	226
9.4	条件编译	226
9.4.1	#if、#elif、#else 和#endif 条件指令	226
9.4.2	defined 运算符	227
9.4.3	#ifdef 和#ifndef 条件指令	227

9.5	#line 行控制指令	228
9.6	#error 指令	229
9.7	#pragma 指令	229
9.7.1	#pragma argsused	230
9.7.2	#pragma exit 与 #pragma startup	230
9.7.3	#pragma inline	231
9.7.4	#pragma option	231
9.7.5	#pragma saveregs	232
9.7.6	#pragma warn	232
9.8	预定义的宏	233
<b>第十章</b>	<b>C++要素</b>	<b>235</b>
10.1	封装(Encapsulation)	236
10.2	继承(inheritance)	238
10.3	多态性(Polymorphism)	239
10.4	重载(Overloading)	239
10.5	用类来模拟现实	240
10.5.1	建立类: 一个图形例子	240
10.5.2	说明对象	241
10.5.3	成员函数	241
10.5.4	调用一个成员函数	242
10.5.5	构造函数和析构函数	242
10.5.6	代码和数据相结合	244
10.5.7	成员访问权控制: 私有的(private)、 公有的(public)和保护的保护的(protected)	245
10.5.8	类: 缺省为私有的	245
10.5.9	运行一个 C++ 程序	246
10.6	继承	248
10.6.1	重新思考 Point 类	248
10.6.2	把类装入模块	251
10.6.3	扩充类	253
10.6.4	多重继承	256
10.7	虚函数	260
10.7.1	虚函数的作用	261
10.7.2	定义虚函数	262
10.7.3	开发一个完整的图形模块	263
10.7.4	普通成员函数与虚拟成员函数	269
10.8	动态对象	270
10.8.1	析构函数和 delete	271
10.8.2	分配动态对象的一个例子	271
10.9	C++ 中更多的灵活性	275

10.9.1	在类定义之外的内部函数	276
10.9.2	带有缺省参数的函数	277
10.9.3	有关重载函数的更多说明	277
10.9.4	重载运算符以提供新的含义	279
10.9.5	友元函数(Friend function)	282
10.9.6	C++流库	283
10.10	用户定义数据类型的I/O	288
10.11	如何进行一步?	289
10.12	小结	289
10.13	掌握C++	290
10.13.1	更好的C: 从C过渡	290
10.13.2	对象支持	293
10.13.3	小结	303

## 第二部分 面向对象程序设计

<b>第十一章</b>	<b>C++概述</b>	<b>306</b>
11.1	什么是面向对象程序设计	306
11.1.1	对象(object)	306
11.1.2	多态性(polymorphism)	307
11.1.3	继承(inheritance)	307
11.2	C++的一些基本原则	307
11.3	编译C++程序	309
11.4	类和对象	309
11.5	函数重载	312
11.6	操作符重载	314
11.7	再谈继承	314
11.8	构造函数与析构函数	318
11.9	C++关键字	320
<b>第十二章</b>	<b>对类的进一步考察</b>	<b>322</b>
12.1	参数化的构造函数	322
12.2	友元函数	325
12.3	缺省函数变元	329
12.4	正确使用缺省变元	331
12.5	类与结构的相关性	331
12.6	联合与类的相关性	333
12.7	内部函数	333
12.7.1	在类中建立一个内部函数	335
12.8	对继承的进一步讨论	336
12.9	多重继承	341
12.10	传递对象到函数	345

12.11	对象数组	345
12.12	对象指针	347
<b>第十三章</b>	<b>函数和操作符重载</b>	<b>349</b>
13.1	构造函数重载	349
13.2	C++中的局部变量	350
13.2.1	动态初始化	351
13.3	将动态初始化用于构造函数	352
13.4	关键字 this	354
13.5	操作符重载	355
13.5.1	友元操作符函数	359
13.6	引用	363
13.6.1	非参数的引用变量	364
13.6.2	使用引用来重载单目操作符	365
13.7	操作符重载的又一例子	368
<b>第十四章</b>	<b>继承、虚函数及多态性</b>	<b>373</b>
14.1	派生类指针	373
14.2	虚函数	375
14.3	为什么要使用虚函数	378
14.4	纯虚函数及抽象类型	382
14.5	先期联编与迟后联编	384
14.6	派生类中的构造函数及析构函数	385
14.7	多重基类	387

### 第三部分 C++技术参考

<b>第十五章</b>	<b>构造 WINDOWS 应用程序</b>	<b>390</b>
15.1	基本过程	390
15.2	在集成环境中编译和连接	391
15.2.1	资源文件描述	391
15.2.2	模块定义文件描述	391
15.2.3	编译和连接 WHELLO	391
15.3	从命令行编译和连接	393
15.3.1	从命令行进行编译	393
15.3.2	从命令进行连接	393
15.3.3	使用 MAKE 文件	394
15.3.4	对 Windows 的另一个 MAKE 文件	395
15.4	入口和出口	395
15.4.1	_export 关键字	397
15.4.2	入口、出口与输出：小结	397
15.5	内存模式	398
15.6	Windows 程序的连接	398

15.6.1	在集成环境中连接	398
15.6.2	用 TLINK 来连接	398
15.7	为 Windows 程序构造一个工程	400
15.8	WinMain	401
15.9	资源编译器	401
15.9.1	资源编译器调用语法	402
15.10	动态连接库	403
15.10.1	在集成环境中编译和连接动态连接库	403
15.10.2	从命令行编译连接一个动态连接库	403
15.10.3	创建动态连接库	404
<b>第十六章</b>	<b>Windows 和 C++ 入门</b>	<b>408</b>
16.1	面向对象的思想	408
16.2	Windows 的词汇	408
16.2.1	模块(module)和可执行性(executable)	408
16.2.2	应用程序(application)	409
16.2.3	动态连接库(DDL)	409
16.2.4	windows.h	409
16.2.5	承接程序 stub	409
16.2.6	Windows 下的函数	409
16.3	C 和 Windows	410
16.4	WHELLO 应用程序	415
16.4.1	伪指令和初始化	415
16.4.2	定义基类	417
16.4.3	定义派生类	418
16.4.4	指向 Windows 的指针	421
16.4.5	回调函数	421
16.5	总结 WinMain	423
<b>第十七章</b>	<b>文件输入输出</b>	<b>424</b>
17.1	传统的 Turbo C I/O	424
17.1.1	两个预处理指令	424
17.1.2	文件与流	426
17.1.3	流(streams)	426
17.1.4	概念和实际	427
17.1.5	控制台 I/O	428
17.1.6	控制台格式化 I/O	429
17.1.7	缓冲型 I/O 系统(ANSI 型 I/O 系统)	434
17.1.8	非缓冲型 I/O——UNIX 型文件系统	447
17.2	C++ I/O 流	452
17.2.1	对应原版本的新流	453
17.2.2	2.0 版流的用法	453



17.2.3	流的老版本用法	465
17.2.4	升级到 2.0 版流的指南	465
17.2.5	流类参考	466
	filebuf <fstream.h>	466
	fstream <fstream.h>	467
	fstreambase <fstream.h>	467
	ifstream <fstream.h>	467
	ios <iostream.h>	468
	iostream <iostream.h>	468
	istream <iostream.h>	469
	istream withassign <iostream.h>	469
	istrstream <strstream.h>	470
	ofstream <fstream.h>	470
	ostream <istream.h>	470
	ostream_withassign <iostream.h>	471
	ostrstream <strstream.h>	471
	streambuf <iostream.h>	471
	strstreambase <strstream.h>	472
	strstreambuf <strstream.h>	472
	strstream <strstream.h>	472
17.3	使用 C++ 的 I/O 类库	472
17.3.1	C++ 为何有自己的 I/O 系统	473
17.3.2	C++ 流	473
17.3.3	C++ 流类	474
17.3.4	建立自己的插入符和抽取符	474
17.3.5	格式化 I/O	478
17.3.6	文件 I/O	484
<b>第十八章</b>	<b>视频函数</b>	<b>491</b>
18.1	Borland C++ & Turbo C++ 图形系统和程序设计要素概述	491
18.1.1	关于视频模式的几点说明	491
18.1.2	有关窗口和视区的几点说明	491
18.1.3	在文本模式下编程	492
18.1.4	在图形模式下编程	497
18.2	Borland C++ & Turbo C++ 图形程序设计	507
18.2.1	基本正文模式函数	507
18.2.2	Borland C++ & Turbo C++ 的图形子系统简介	516
<b>第十九章</b>	<b>内存模型、浮点数和覆盖</b>	<b>525</b>
19.1	内存模型	525
19.1.1	8086 寄存器	525
19.1.2	内存分段	527