

HOPE

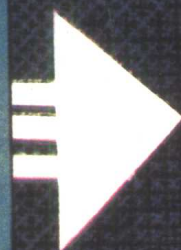


中国科学院希望高级电脑技术公司高级程序设计丛书之五

面向对象的程序设计

Turbo C++

程序设计方法



海洋出版社

叶欣 晓飞 薛梅
魏群 皇甫广宇 编译

968743

TP311
6477

中国科学院希望高级电脑技术公司高级程序设计丛书之五

面向对象的程序设计

Turbo C++

程 序 设 计 方 法

海 洋 出 版 社

1991 年 4 月 · 北京

内 容 提 要

Turbo C++ 1.0是在Turbo C基础上推出的最新面向对象的程序设计软件包。该丛书共五册,本书是其中的一册。主要内容分三部分,共二十一章。第一部分综述了Turbo C++,介绍了其集成环境,讨论了编辑器。第二部分介绍C语言的基本知识,并说明了C++中类化的性质。第三部分完整地描述了C的C++扩充及许多库函数。书中几乎对Turbo C++的各个主要性质都给出了编译及运行的典型程序。书末的六个附录,列出了常用C库函数, Turbo C++ 调试器使用、存储模式、VROOMM覆盖技术、使用命令行编译程序、编译多文件程序等资料。

本书内容丰富,结构合理,它使已熟悉C语言的读者不必再重复阅读熟知的内容,而又可使初次接触C和C++的读者方便地获得充分的背景材料。

Turbo C++ 程序设计方法

叶欣 晓飞 薛梅
魏群 皇甫广宇 编译

审 校 刘莉蕾

责任编辑: 阎世尊

海洋出版社出版(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: 787×1092 1/16 印张: 字数 560千字

1991年4月第一版 1991年4月第一次印刷

印数: 1—3000

ISBN 7-5027-1392-1/TP·21

定价: 15.00元

前 言

随着计算机软件设计方法的不断提高和改进,面向对象的程序设计在我国也逐渐流行起来并大有迅猛发展的趋势,它提出了一种全新的程序设计思想,把数据和对数据进行的操作融为一体。

美国 BORLAND 公司在 Turbo C 的基础上,推出最新面向对象的程序设计软件包——Turbo C++ 1.0 版。它继承并发挥了原来 Turbo C 集成环境的优良特性,并包含了面向对象的基本思想和设计方法,是目前国际上最受欢迎的面向对象程序设计软件包。

我们在长期从事 Turbo C 程序设计的基础上,加上近几年对面向对象的研究,根据 BORLAND 公司的 Turbo C++ 软件和资料,特编译了 Turbo C++ 程序设计丛书。

本丛书共有五册,分别为《入门》、《用户手册》、《程序员指南》、《库函数参考》和《程序设计方法》,全面地介绍了 Turbo C++ 的基础知识和高级技术,是一套引导读者进行面向对象程序设计的系统参考书。

《程序设计方法》是一本介绍如何利用 Turbo C++ 进行程序设计的教科书,本书从低级到高级、从简单到复杂、从 C 到 C++,循序渐进地介绍了程序设计的一般方法和注意事项。

由于面向对象的概念在国内还处于初步阶段,加上我们水平有限,书中难免会有错误和缺点,敬请广大读者批评指正,以备再版时修订。来信请寄中央财政金融学院信息系叶欣收。

本书在编译过程中,得到了许多同志的帮助和支持,刘学功同志、姜涛同志为本书的成稿做了有益的工作,在此表示谢意。

本丛书出版过程中,得到海洋出版社的编辑同志以及中国科学院希望高级电脑技术公司资料部秦人华经理、杨淑新老师的大力帮助和支持,在此表示衷心的感谢。

编译者

1991 年 4 月于北京

25348/03

目 录

引言	1
第一部分 Turbo C++入门	3
第一章 Turbo C++综述	4
1.1 C++的起源	4
1.2 什么是面向对象程序设计?	4
1.3 C++忠实于C之精华	5
1.4 C++的用途	5
1.5 编译程序及解释程序	6
1.6 关于C程序员	6
第二章 Turbo C++集成开发环境	7
2.1 运行 Turbo C++	7
2.2 使用鼠标	8
2.3 主菜单	8
2.3.1 对话框	9
2.4 打开完整菜单	10
2.5 对主菜单的考察	10
2.5.1 系统菜单	10
2.5.2 File	11
2.5.3 Edit	12
2.5.4 Search	12
2.5.5 Run	12
2.5.6 Compile	12
2.5.7 Debug	13
2.5.8 Project	13
2.5.9 Options	13
2.5.10 Window	14
2.5.11 Help	15
2.6 热键	16
2.7 使用 Turbo C++的上下文敏感帮助	17
2.8 理解窗口	17
2.8.1 窗口的大小设置及移动	18
2.9 编辑窗口	19
2.10 信息窗口	19
2.11 状态行	19

第三章 使用 Turbo C++ 编辑器	20
3.1 编辑器命令	20
3.2 激活编辑器并键入文本	20
3.3 删除字符、字和行	22
3.4 文本的移动、拷贝和块移动	22
3.5 剪裁板的使用	23
3.6 光标移动的进一步说明	24
3.7 搜索和替换	24
3.8 设置和搜索位置标识	26
3.9 文件的存储和装入	27
3.10 自动缩格	27
3.11 文本块移入和移出磁盘文件	27
3.12 对匹配	28
3.13 其它命令	28
3.14 命令综述	29
3.15 修改编辑器缺省	30
3.16 用文件名激活 Turbo C++	31
第二部分 C 语言	32
第四章 C 语言要素	33
4.1 准备 IDE	33
4.2 C 对大小写敏感	33
4.3 一个简单的 C 程序	33
4.3.1 进一步讨论	34
4.4 错误处理	35
4.5 错误与警告比较	36
4.6 第二个程序	36
4.6.1 一种变化	37
4.7 快速复习	38
4.8 什么是 C 函数?	38
4.8.1 函数和自变量	38
4.8.2 返回值的函数	39
4.8.3 函数的一般形式	40
4.9 两个简单命令	40
4.9.1 if 命令	40
4.9.2 for 循环命令	41
4.10 代码块	41
4.11 字符和字符串	41
4.11.1 字符串	42
4.12 printf(): 快速复习	43

4.13	scanf()快速回顾	44
4.14	分号、括号和注释	44
4.15	缩排练习	45
4.16	C 库	45
4.17	C 的关键字	45
4.18	术语复习	46
	第五章 变量、常量、操作符和表达式	47
5.1	标识符名称	47
5.2	数据类型	47
5.2.1	类型修饰符	48
5.3	变量说明	50
5.3.1	变量说明的位置	50
5.4	常量	51
5.4.1	十六进制和八进制常量	52
5.4.2	字符串常量	52
5.4.3	转义字符常量	52
5.4.4	变量初始化	53
5.5	操作符	54
5.5.1	算术操作符	54
5.5.2	增量和减量	55
5.5.3	关系和逻辑操作符	55
5.5.4	赋值操作符	58
5.6	表达式	59
5.6.1	表达式中的类型转换	59
5.6.2	空格和圆括号	60
	第六章 程序控制语句	61
6.1	if 语句	61
6.1.1	else 语句的使用	62
6.1.2	if-else-if 阶梯	63
6.1.3	条件表达式	64
6.1.4	嵌套 if	64
6.2	switch 语句	65
6.2.1	default 语句	67
6.2.2	break 语句的进一步讨论	67
6.2.3	嵌套的 switch 语句	68
6.3	循环	70
6.4	for 循环	70
6.4.1	for 循环的基本知识	70
6.4.2	for 循环的变化	71
6.4.3	无穷循环	73

6.4.4	for 循环的中断	73
6.4.5	无循环体循环的使用	73
6.5	while 循环	73
6.6	do_while 循环	75
6.7	嵌套循环	76
6.8	循环中断	78
6.9	continue 语句	79
6.10	标号和 goto	80
第七章	数组和字符串	82
7.1	一维数组	82
7.1.1	无界检测	83
7.1.2	一维数组是一个表	83
7.2	字符串	83
7.2.1	从键盘上读字符串	84
7.2.2	一些字符串库函数	84
7.2.3	空终止符的使用	87
7.2.4	printf()的一种变化	88
7.3	二维数组	88
7.3.1	字符串数组	90
7.4	多维数组	91
7.5	数值初始化	91
7.5.1	变界数组初始化	92
7.6	一个水下搜索游戏	92
第八章	指针	97
8.1	指针是地址	97
8.2	指针变量	97
8.3	指针操作符	98
8.4	指针表达式	99
8.4.1	指针赋值	99
8.4.2	指针运算	100
8.4.3	指针比较	101
8.5	指针和数组	101
8.5.1	索引指针	102
8.5.2	指针和字符串	102
8.5.3	如何得到一个数组元素的地址	103
8.5.4	指针数组	104
8.5.5	一个使用数组和指针的有趣实例	105
8.6	指针的指针	109
8.7	初始化指针	110
8.8	指针的一些问题	111

第九章 函数：更详尽的说明	113
9.1 函数的一般形式	113
9.2 return 语句	113
9.2.1 从一个函数中返回	113
9.2.2 返回值	114
9.3 函数返回非整型值	116
9.3.1 使用函数原型	116
9.3.2 返回指针	118
9.3.3 void 类型函数	119
9.4 更多的有关原型的知识	119
9.4.1 参数不匹配	119
9.4.2 头文件：更详尽的说明	120
9.4.3 无任何参数的原型函数	121
9.4.4 有关旧式 C 程序	121
9.5 作用域规则	122
9.5.1 局部变量	122
9.5.2 形式参数	124
9.5.3 全局变量	124
9.5.4 作用域最后的例子	125
9.6 函数的参数和自变量：更详尽说明	125
9.6.1 赋值调用和赋地址调用	125
9.6.2 建立一个赋地址调用	127
9.6.3 函数调用与数组	128
9.7 argc、argv 和 env——main 中的参数	131
9.8 从 main() 中返回值	133
9.9 递归	134
9.10 参数说明的传统风格和现代风格	135
9.11 补充问题	136
9.11.1 参数和通用函数	136
9.11.2 效率	137
第十章 输入、输出和磁盘文件	138
10.1 两个预处理指令	138
10.1.1 #define 指令	138
10.1.2 #include 指令	140
10.2 流与文件	140
10.3 流(streams)	140
10.3.1 文本流	141
10.3.2 二进制流	141
10.3.3 文件	141
10.4 概念和实际	141

10.5 控制台 I/O	142
10.5.1 字符读写	142
10.5.2 字符串读写	143
10.6 控制台格式化 I/O	143
10.6.1 printf() 函数	144
10.6.2 scanf() 函数	145
10.7 缓冲型 I/O 系统(ANSI 型 I/O 系统)	148
10.7.1 文件指针	149
10.7.2 打开文件	149
10.7.3 写字符	150
10.7.4 读字符	151
10.7.5 fgetc() 的使用	151
10.7.6 关闭文件	151
10.7.7 ferror() 和 rewind() 函数	152
10.7.8 fopen(), getc(), putc() 和 fclose() 函数的用法	152
10.7.9 getw() 和 putw() 函数的使用	154
10.7.10 fgetc() 和 fputc() 函数	154
10.7.11 fread() 和 fwrite() 函数	154
10.7.12 fseek() 函数和随机访问 I/O	156
10.7.13 标准流	158
10.7.14 fprintf() 和 fscanf() 函数	159
10.7.15 删除文件	161
10.8 非缓冲型 I/O——UNIX 型文件系统	161
10.8.1 open(), creat() 和 close() 函数	162
10.8.2 read() 和 write() 函数	163
10.8.3 unlink() 函数	165
10.8.4 随机访问文件和 lseek() 函数	165
第十一章 高级数据类型	167
11.1 访问修饰符	167
11.1.1 const 常量	167
11.1.2 volatile 易变量	168
11.2 存储类型说明符	169
11.2.1 auto(自动变量)	169
11.2.2 extern(外部变量)	169
11.2.3 static variables(静态变量)	170
11.2.4 static local variables(静态局部变量)	170
11.2.5 static global variables(静态全局变量)	171
11.2.6 register variables(寄存器变量)	172
11.3 赋值语句中的类型转换	173
11.4 函数类型修饰符	175

11.4.1	pascal	175
11.4.2	cdecl	175
11.4.3	interrupt	175
11.5	指向函数的指针	175
11.6	动态分配	177
11.7	分配和释放内存	178
第十二章	用户定义的数据类型	180
12.1	结构	180
12.1.1	访问结构元素	181
12.2	结构数组	182
12.2.1	通讯录实例	182
12.3	结构赋值	188
12.4	将结构传递给函数	188
12.4.1	将结构元素传递给函数	188
12.4.2	将整个结构传递给函数	189
12.5	结构指针	190
12.5.1	结构指针说明	190
12.5.2	使用结构指针	190
12.6	结构内部的数组和结构	193
12.7	位域	194
12.8	联合(union)	197
12.9	枚举	200
12.10	使用 sizeof 来确保可移植性	202
12.11	typedef	203
第十三章	高级运算符	205
13.1	按位运算符	205
13.2	?运算符	211
13.3	C 语言的简写	212
13.4	逗号运算符	213
13.5	方括号和圆括号	213
13.6	运算符优先级表	213
第十四章	屏幕控制函数	214
14.1	基本正文模式函数	214
14.1.1	正文窗口	214
14.1.2	清除窗口	215
14.1.3	光标定位	215
14.1.4	清除到行末	215
14.1.5	删除和插入行	216
14.1.6	建立窗口	217
14.1.7	一些窗口 I/O 函数	219

14.1.8	正文模式	221
14.1.9	用彩色输出正文	221
14.2	Turbo C++的图形子系统介绍	223
14.2.1	一个有别名的窗口	223
14.2.2	初始化显示器适配器	223
14.2.3	退出图形模式	225
14.2.4	颜色和调色板	225
14.2.5	基本图形函数	227
14.2.6	改变绘图色	228
14.2.7	区域填充	228
14.2.8	rectangle()函数	229
14.2.9	创建视口	230
第十五章	C预处理指令	232
15.1	C预处理指令	232
15.2	#define 指令	232
15.3	#error 指令	233
15.4	#include 指令	234
15.5	条件编译指令	234
15.5.1	#if、#else、#elif 和 #endif	234
15.5.2	#ifdef 和 #ifndef 指令	236
15.6	#undef 指令	237
15.7	#line 指令	237
15.8	#pragma 指令	237
15.9	预定义的宏替换名	239
第三部分	使用 Turbo C++的面向对象性质	241
第十六章	C++概述	242
16.1	什么是面向对象程序设计?	242
16.1.1	对象(object)	242
16.1.2	多态性(polymorphism)	243
16.1.3	继承(inheritance)	243
16.2	C++的一些基本原则	243
16.3	编译 C++程序	245
16.4	类及对象的引入	245
16.5	函数重载	247
16.6	操作符重载	249
16.7	再谈继承	250
16.8	构造函数与析构函数	252
16.9	C++关键字	255

第十七章 对类的进一步考察	256
17.1 参数化的构造函数	256
17.2 友元函数	258
17.3 缺省函数变元	262
17.4 正确地使用缺省变元	264
17.5 类与结构之相关性	264
17.6 联合与类之相关性	265
17.7 内部函数	266
17.7.1 在类中建立一个内部函数	267
17.8 对继承的进一步讨论	268
17.9 多重继承	272
17.10 传递对象到函数	276
17.11 对象数组	276
17.12 对象指针	277
第十八章 函数和操作符重载	280
18.1 构造函数重载	280
18.2 C++ 中的局部变量	281
18.2.1 动态初始化	282
18.3 将动态初始化用于构造函数	283
18.4 关键字 this	284
18.5 操作符重载	285
18.5.1 友元操作符函数	289
18.6 引用	292
18.6.1 非参数的引用变量	294
18.6.2 使用引用来重载单目操作符	295
18.7 操作符重载的又一例子	297
第十九章 继承、虚函数及多态性	301
19.1 派生类指针	301
19.2 虚函数	303
19.3 为什么使用虚函数	305
19.4 纯虚函数及抽象类型	308
19.5 先期联编与迟后联编	310
19.6 派生类中的构造函数及析构函数	311
19.7 多重基类	313
第二十章 使用 C++ 的 I/O 类库	314
20.1 C++ 为何有自己的 I/O 系统	314
20.2 C++ 流	315
20.2.1 C++ 预定义流	315
20.3 C++ 流类	315
20.4 建立自己的插入符和抽取符	315

20.4.1	建立插入函数	315
20.4.2	重载抽取函数	317
20.5	格式化 I/O	319
20.5.1	用 ios 成员函数格式化	319
20.5.2	使用操纵函数	322
20.5.3	建立自己的操纵函数	323
20.6	文件 I/O	325
20.6.1	打开和关闭文件	325
20.6.2	读写文本文件	326
20.6.3	二进制 I/O	327
20.6.4	检测 EOF	329
20.6.5	随机访问	329
第二十一章	其它 C++ 内容	332
21.1	用 new 与 delete 进行动态分配	332
21.1.1	重载 new 和 delete	335
21.2	静态类成员	338
21.3	C 与 C++ 的区别	341
21.4	Turbo C++ 的复数及 BCD 类	341
21.5	基于消息的哲学	344
21.6	最后的话	347
附录 A	常用的一些 C 库函数	348
A.1	串和字符函数	348
A.2	数学函数	356
A.3	操作系统相关函数	362
A.4	其它函数	366
附录 B	使用 Turbo C++ 调试器	375
B.1	为调试准备程序	375
B.2	什么是源级调试器	375
B.3	调试器的基本内容	375
B.3.1	单步调试	376
B.3.2	设置断点	376
B.3.3	监视变量	377
B.3.4	监视表达式格式码	377
B.3.5	监视栈	378
B.3.6	计算一个表达式	379
B.3.7	检测一个变量	379
B.3.8	使用寄存器窗口	379
附录 C	Turbo C++ 的存储模式	380
C.1	8086 处理器系列	380
C.2	地址计算	380

C.3	近指针及远指针·····	381
C.4	存储模式·····	381
C.4.1	微模式(Tiny Model)·····	381
C.4.2	小模式(Small Model)·····	381
C.4.3	中模式(Medium Model)·····	381
C.4.4	紧缩模式(Compact Model)·····	381
C.4.5	大模式(Large Model)·····	381
C.4.6	巨模式(Huge Model)·····	381
C.4.7	模式选择·····	381
C.4.8	存储模式编译选项·····	381
C.5	强制转换存储模式·····	381
C.5.1	far·····	381
C.5.2	near·····	381
C.5.3	huge·····	381
C.6	Turbo C++的段指示符·····	381
附录 D	使用 VROOMM 覆盖技术·····	385
附录 E	使用命令行编译程序·····	385
附录 F	编译多文件程序·····	385

引 言

在 80 年代末期,一种新的程序设计方法开始形成,这就是所谓的面向对象的程序设计。面向对象程序设计(Object oriented Programming, 以下简称 OOP)融合了结构化程序设计(及其前身)的所有性质,并为程序员提供了分析及解决程序设计任务的一种令人激动的新方法,在本书中,你将会学到 OOP 为什么是重要的,以及怎样用 Turbo C++来实现 OOP。

C++是 C 的面向对象版本。在 80 年代, C 是最流行的语言,并且在很长时期内, C 仍将被广泛使用。C 是以其效率、功能及精致而著称的。C++保留了这些重要的性质,但另增加了对 OOP 的支持。正是这种组合,才使 C++成为当前仅有的最重要的面向对象程序设计语言。

1988 年,美国 Borland 公司开始了其对 C++的秘密工作。他们以其极为有力的 Turbo C 作为起步,在之上加入了 C++的 OOP 扩充。这不是一项轻而易举的任务。虽然 C++易于为程序员所使用,但对编译器的构造却相当困难。事实上, Turbo C++是 Borland 曾经进行过的最大的语言工程。为创建 Turbo C++, Borland 汇集了当今一些最优秀的编译器设计人员。在 Turbo C++的开发期间, Borland 还改进了与其交互式开发环境的用户接口。

正如你将从本书中所看到的那样, Turbo C++提供了一种有力而灵活的程序设计环境。它产生紧凑而有效的代码,并且支持几百种库函数和类。

关于本书

本书以基础语言 C 开始,讲解 Turbo C++程序设计语言。由于 C++是 C 语言的一个超集,所以在学习 C++这前,应当了解 C。但是,即使你尚未知晓 C,仍可以使用本书,因为它含有你需要的所有背景知识。本书还描述了 Turbo C++的集成环境、编辑器及许多库函数。在读完本书后,你将能称自己为一名 Turbo C++程序员。

本书基于如下哲学:最好的学习方法是实践。为此,本书含有许多将对其进行编译及运行的典型程序。事实上,对 Turbo C++的几乎每个主要性质,都有一典型程序来加以说明。

本书的组成

本书的组成是基于如下考虑的:它应能为两种类型的程序员有效使用,即已经了解 C 但想要学习 C++ 扩充的程序员 和对 C 及 C++都是第一次接触的程序员。

第一部分综述了 Turbo C++, 介绍了 Turbo C++的集成环境,并且讨论了编辑器。第二部分说明了 C++中的类似 C 的性质。如果你尚未了解 C,一定要仔细阅读这部分。但是,如果你对 C 很熟练,就可以跳到第三部分,在那里,完整地描述了 C 的 C++扩充。

以上述方式组织本书使得有经验的 C 程序员不必再次面对已经熟知的东西,而另一方面,又使得初学者能方便地获得充分的背景材料。

为何本书适合于你

如果你想要学习用 Turbo C++ 编程，并且想了解所谓面向对象程序的设计思想，那么，本书就很适合于你，不管你是程序设计的新手还是老手。如你所知，C++ 是基于 C 语言的。事实上，了解 C 是学习 C++ 的前提。由于本书的组成方式，如果你已熟悉 C，就可以不重温已了解的内容而很快进入处理 C 向 C++ 扩充的章节。如果你对 C 和 C++ 都是初次接触，那么就可从头开始，一次一章地直到本书结束。

本书除了讲解 Turbo C++ 语言外，还描述了 Turbo C++ 的集成编程环境、Turbo C++ 编辑器以及各种编译器选项。