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三

面向对象的程序设计

Turbo C++ 程序员指南

叶　欣　童长忠
张国锋　唐晓菲　编译

海洋出版社

中国科学院希望高级电脑技术公司高级程序设计丛书之三

面向对象的程序设计

Turbo C++

程 序 员 指 南

叶欣　童长忠

张国锋　唐晓菲

编译

海 洋 出 版 社

1991 年 4 月 · 北京

内 容 提 要

Turbo C++1.0是在Turbo C的基础上推出的最新面向对象的程序设计软件包。该丛书共有五册，本书是其中的一册。本书共分七章，主要介绍了Turbo C++的语言标准、Turbo C++库函数分类，C++流、存储模型、浮点数和覆盖、视频函数、和汇编语言的接口及错误信息。最后的附录简单说明了ANSI的特殊实现标准。本书是该丛书中的高技术书，是引导读者进行面向对象程序设计的重要参考书。

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

Turb C++ 程序员指南

叶 欣 童长忠
张国锋 唐晓菲 编译

特约编辑：
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：650千数
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：13.00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。

《程序员指南》是本套书中的高级技术书，主要介绍 Turbo C++的语言标准、Turbo C++库函数分类，并讨论了 C++流、存储模型、浮点数、覆盖技术和视频函数，本书也介绍将 Turbo C++程序与汇编语言接口方面的知识，并列出 Turbo C++编程过程中可能出现的错误信息，最后简单说明了 ANSI 的特殊实现标准。

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。

本书在编译过程中，得到了许多同志的帮助和支持，其中张阳同志、吕幸生同志、薛梅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

目　录

引 言

本手册适合于高级程序员。它包括语言参考手册运行时间库的交叉引用以及有关C++流、存储模式、浮点、覆盖、视频函数、汇编语言接口和运行与编译时刻的出错信息等编程资料。

如果你:

1. 从未用任何语言编过程序;
2. 编过程序，但不是用C语言; 或者
3. 想了解有关安装Turbo C++的信息

请首先查看《入门》，它概括了全部Turbo C++的文档内容，其中引言和第二章介绍怎样最有效地使用Turbo C++手册。

与本手册有关的还有两本手册。一本是《用户手册》，它介绍了Turbo C++集成环境(包括编辑程序)、项目管理程序、命令行编译程序、Turbo C++的例程和Turbo编辑程序宏语言。

另一本是《库函数参考》，它按字每顺序介绍了全部Turbo C++函数和全局变量。

0.1 本手册的内容简介

第一章: “Turbo C++语言标准”讲述了Turbo C++语言，所有与ANSI C标准不同之处都在此标明。本章包括C与C++语言的语言参考及文法。

第二章: “运行时间库交叉参考”给出了一些关于运行时间库源代码方面的信息，列出并描述了头文件，同时提供了一个用对象来组织的运行时间库的交叉参考。例如，如果要找出哪些函数与图形有关，可查找本章的“图形”节。

第三章: “C++流”告诉你怎样使用C++流库。

第四章: “存储模型、浮点数和覆盖”介绍了存储模式、混合模式的程序设计、浮点问题和覆盖。

第五章: “视频函数”讨论怎样由Turbo C++处理正文与图形输出。

第六章: “与汇编语言的接口”说明怎样编写汇编语言程序，使之从Turbo C++程序调用时能正常工作。

第七章: “错误信息”列出并解释了全部运行时刻和编译程序产生的致命错误、一般错误和警告信息，同时建议了可能的解决方法。

附录A: “ANSI特殊实现标准”介绍了ANSI严格定义或未定义的ANSI C标准的一些方面，它们将随着实现的不同而不同。本附录给出了Turbo C++对这些方面的处理。

第一章　Turbo C++语言标准

本章为程序员提供了一个详细的 Turbo C++语言参考指南，它不是一语言的概述，而是 Turbo C++实现的 C 和 C++语言的形式描述。本章给出了词法和短语结构的语法以及所有预处理程序指令的细节。我们采用一种改版的 BNF 范式来表达文法。必要时通过粗略的解释和程序例子加以补充。

Turbo C++实现了 X3J11 技术委员会在 1983 年 6 月至 1988 年 9 月间制定的 ANSI C 标准，并在多处作了扩充。你可在编译时设置选择项以告知这些扩充是否存在。也可以让编译程序将 Turbo C++扩充关键字当作正常的标识符来处理(参见《用户手册》第四章“命令行编译程序”)。

你也可通过 ANSI C 提供的#pragma 指令来获得“一致的”扩充，以处理非标准的依赖于实现的特征。

Turbo C++完全实现了 AT&T 的 C++2.0 版，C++2.0 是由 AT&T 贝尔实验室 Bjarne Stroustrup 开发的 C 的面向对象超集，除了增加了许多新的特征和能力外，C++对 C 进行了或多或少的修改。我们在本章对这些不同之处作了注明。

1.1　语法与术语

语法定义由被定义的非终结符的名字后跟一个冒号(:)组成。备选通常每个占有一行，但若加有前缀短语“下述之一”，备选也可只放在一行上。例如:

外部定义:
　　函数定义
　　说明
八进制数字: 下述之一
　　0 1 2 3 4 5 6 7

定义结构中的选择项放在尖括号中:

整型后缀:
　　无符号后缀<长整型后缀>

全章中，单词“参数”用来系指通过函数调用传递的实际值，“参量”系指在函数头中定义的用来保存值的变量。

1.2　词法与短语结构的语法

词法关心的是一语言可识别的类似词的单元不同分类，这种单元也称为词法符号(Token)。短语语法详述词法符号组合成为表达式、语句和其它有意义单元的合法方式。

Turbo C++的词法符号由编译程序及其预处理程序在你的程序上进行一系列操作得到。

Turbo C++源代码由一 ASCII 字符序列组成，它可用一恰当的正文编辑程序(如 Turbo C++编辑程序)创建。Turbo C++的基本程序单元是文件，它通常对应于放在 RAM 中磁盘上的有名 DOS 文件，且带有扩展名.C 或.CPP。

预处理程序首先扫描程序正文以处理预处理程序指令。例如，指令#include<inc_file>在编译之前把文件 inc_file 的内容加到(或包含到)程序中。预处理程序还扩展在程序和包含文件中找到的任何宏。

1.3 空白

在编译的词法分析阶段，将把源代码分为词法符号和空白，空白是空格、水平制表、竖直制表、换行符和注释的合称。空白用来指示词法符号的开始和结束位置，但除了这一功能之外，其余的空白将被忽略。例如，两序列

```
int  i; float f;
```

和

```
int  i;
   float  f;
```

是词法等价的，通过词法分析都获得 6 个词法符号:

```
int i ; float f ;
```

空白字符可出现在串文字量里，这时它们将不作正常的词法分析过程，换句话说，它们将作为串的一部分，如

```
char name[] = "Borland International";
```

分析后将得到 7 个词法符号，包括串文字量符号"Borland International"。

1.3.1 行分隔符

若在最后的换行字符前加一反斜杠(\)，则反斜杠和换行符都将忽略，使得两正文物理行变为一行。

```
"Borland \
International"
```

分析后成为"Borland International"。

1.3.2 注释

注释为一用来注解程序的正文段。注释仅为程序员所用，它们在分析前将从源正文中去掉。

有两种方法来给出注释: 传统的 C 方法和 C++方法，两者 Turbo C++都支持，利用传统的选择项扩充，将允许嵌套的注释存在。你能随时采用任何一种。

1.3.2.1 C 注释

传统的 C 注释是放在符号对/*后的任一字符序列，它终止于跟在开始/*后的第一个符号对*/。整个序列，包括 4 个注释分隔符号，在宏扩展后由一个空格代替。注意有些 C 实现是消去注释而非代替空格。

Turbo C++不用/**/支持不可移植的词法符号传递策略，Turbo C++的符号传递由 ANSI 指定的##对进行，如:

```
#define VAR(i,j)  (i/**/ j)    /*编译出错*/
#define VAR(i,j)  (i/##/ j)    /*编译通过*/
#define VAR(i,j)  (i/##/ j)    /*编译通过*/
```

在 Turbo C++中，

```
int /*说明*/i /*计数器*/;
```

分析后成为

```
int i;
```

它有 3 个符号: int i ;

1.3.2.2 嵌套注释

ANSI C 不允许有嵌套注释，若将上一行注释为

```
/* int /*说明*/ ; /*计数器*/; */
```

将出错，因为和一个/*的作用域在第一个*/处终结。由此将得到

```
i ; */
```

这将产生一个语法错误。

在缺省状态下，Turbo C++是不允许嵌套注释的，但你能用-C 编译选择项或通过集成环境里的 O|C|Source|Option 选择项激活嵌套注释。

1.3.2.3 C++注释

C++采用两相邻的斜杠(//)来表示一行后面为注释。这种注释可开始于任一位置，而终止于行末:

```
class X { //这是一注释
...};
```

1.3.2.4 注释分隔符与空白

在极个别情况下，/*和//前及*/后的空白虽然语法上来说不是强制性的，但能避免移植性问题。例如，C++代码:

```
int  i = j//*除以 k * / k;
+m;
```

通过词法分析变为 i = j + m;而不是按传统 C 约定所期望的

```
int i = j/k;
+m;
```

更合法的形式

```
int i = j/ /*除以*/k;
+m;
```

能避免这一问题。

1.4 词法符号

Turbo C++识别 6 类词法符号: 关键字、标识符、常量、串文字量、操作符和标点符号(也称为分隔符)。词法符号的形式定义如下:

词法符号:

- 关键字
- 标识符
- 常量
- 串文字量
- 操作符

标点符号

对源代码进行分析时，词法符号以尽可能长的方式从字符序列中提取出来。例如，external 将分析为一个单标识符，而非关键字 extern 后跟标识符 al。

1.4.1 关键字

关键字保留为专用，一定不能用作通常的标识符名字。下面 4 张表列出了 Turbo C++的关键字。你可使用命令行编译选择项(或集成开发环境 IDE 的相应选择项)仅选用 ANSI 关键字等。有关这些选择项的说明参见《用户手册》的第一章“IDE 指南”和第四章“命令行编译程序”。

表 1.1 全部 Turbo C++关键字

asm	_ds	interrupt	short
auto	else	_loadds	signed
break	enum	long	sizeof
case	_es	near	_ss
catch	_export	new	static
cdecl	extern	operator	struct
char	far	pascal	switch
class	float	private	template
const	for	protected	this
continue	friend	public	typedef
_cs	goto	register	union
default	huge	_regparam	unsigned
delete	if	return	virtual
do	inline	_saveregs	void
double	int	_seg	volatile
			while

表 1.2 Turbo C++对 ANSI 的扩充

cdecl	_export	_loadds	_saveregs
_cs	far	near	_seg
_ds	huge	pascal	_ss
_es	interrupt	_regparam	

表 1.3　C++专用的关键字

catch	friend	operator	public
class	inline	private	template
delete	new	protected	this
			virtual

表 1.4　Turbo C++寄存器伪变量

_AH	_BL	_CL	_DL
_AL	_BP	_CX	_DX
_AX	_BX	_DH	_FLAGS
_BH	_CH	_DI	_SI
			_SP

1.4.2　标识符

标识符的形式定义如下:

标识符:

非数字

标识符　非数字

标识符　数字

非数字: 下述之一

a b c d e f g h i j k l m n o p q r s t u v w x y z _

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

数字: 下述之一

0 1 2 3 4 5 6 7 8 9

1.4.2.1　命名与长度限制

标识符是任一长度的类、对象、函数、变量和用户定义的数据类型等的名字。标识符可包含字母 A～Z、a～z、下划线_和数字 0～9，仅带有两条限制:

1. 首字符必须是字母或下划线。

2. 缺省时，Turbo C++只认为前 32 个字符有意义。有意义字符数可通过菜单与命令行选择项减少，但不能增加，使用 TCC 的-in 选择项或集成环境的 0|C|S|Identifier Length 选择项可修改该数。其中 1< =n< =32。

1.4.2.2　标识符与字母大小写

Turbo C++标识符的字母大小写是不同的，由此 Sun、sum 和 suM 是不同的标识符。

从其它模块输入的全局标识符与一般标识符一样，遵循相同的命令与有意义规则。不过，Turbo C++具有抑制大小写相关的选择，这样当与大小写无关的语言连接时能保持兼容性。通过检测连接程序对话框里的 Options|Linker|Case-Sensitive Link，或者利用 TLINK

的/c 命令行开关，你能确保全局标识符是大小写无关的。这时，全局量 Sum 与 sum 将认为是相同的，于是连接期间将可能导致“重复符号”这一警告信息。

这种规则的唯一例外是 Pascal 类型的标识符为了连接总是转换为大写。

1.4.2.3 唯一性与作用域

虽然标识符的名字相对上述规则来说带有任意性，但若同一名字在同一作用域里用多于一个的标识符且共享同一名字空间时，将会引起错误。重复名字若占不同的名字空间，则不管作用域是否相同，都总是合法的。

1.4.3 常量

常量是用来表示固定的数值和字符值的词法符号。Turbo C++支持 4 类常量:浮点型、整型、枚举型和字符型。

编译程序利用数值和代码中所用的格式等线索来判断常量的数据类型。常量的形式定义如下表所示。

表 1.5 常量的形式定义

常量:
    浮点常量
    整常量
    枚举常量
    字符常量
浮点常量:
    小数常量<指数部分><浮点后缀>
    数字序列 指数部分 <浮点后缀>
小数常量:
    <数字序列> . 数字序列
    数字序列 .
指数部分:
    e<符号>数字序列
    E<符号>数字序列
符号: 下述之一
    + -
数字序列:
    数字
    数字序列 数字
浮点后缀: 下述之一
    f l F L
整常量:
    十进制常量<整型后缀>
    八进制常量<整型后缀>