

头 用 指南

面向对象程序设计语言

## 编者的话

本书介绍面向对象的程序设计语言 Turbo C++，由上下两篇组成。

Turbo C++ 功能强、性能高，它既和 ANSI C 兼容，也和 AT&T 的 C++ 兼容。它有利于 C 和 C++ 程序员获得一个快速和高效的编译器，有利于 Turbo Pascal 程序员学会具有所有“Turbo”优越性的 C 或 C++ 语言，有利于任何一个人学习 C 或 C++ 语言。Turbo C++ 的优越性之一在于，它能够使我们以适合于自己的速度，轻松地投入 C++ 的程序设计中。我们可以从学习 C 开始，然后在学习过程中，逐步尝试使用 C++ 的特性。当然，我们也可以从一开始就直接用 C++ 来编程。

面向对象的程序设计语言 C++ 是 C 语言自然发展的产物。C++ 的可移植性很强，我们可以很方便地将 C++ 应用程序从一个系统移植到另一个系统。因此，任何环境中几乎所有的编程任务，都可以用 C++ 来实现。

面向对象的程序设计方法以其新颖、独特、合理和自然的优点而深受计算机界广大同行的欢迎。该方法的特点是信息隐藏、数据抽象、动态链接以及接近于人的思维。面向对象的方法很有发展前途，愈来愈多的系统开发人员正在开始运用这种新型的方法。本书对这一方法进行了全面的论述，剖析了面向对象方法的实质和特性，在程序设计方法和技巧中融入了丰富的实践经验。

本书上篇首先介绍了 Turbo C 和 Turbo C++ 的基础知识，然后由浅入深地阐述了创建、运行 C 和 C++ 程序的方法，详细地讲述了 Turbo C++ 独特的集成开发环境 IDE 的全貌和使用方法，使读者能够快速地适应 Turbo C++ 的编程和开发环境。此外，这一篇还介绍了 Turbo C++ 进行工程和程序管理的各种工具设施。

上篇分为两个部分：第一部分提供了使用户能够快速地适应编程和运行环境等方面的内容，包括安装、定做、指导、入门和文档介绍等等；第二部分提供了有关 IDE 基础的内容，包括非常高级的编辑器、工程管理程序以及使用 MAKE 和 TLINK 实用程序、命令行编译器的细节。上篇共分十三章，各章内容分述如下：

第一章讲述了如何安装 Turbo C++，以及如何定做 Turbo C++。

第二章讲述了 Turbo C++ 的许多引人注目的特征，并告诉用户如何找到关于它们的更多信息。

第三章可使用户了解整个 IDE 环境，并且介绍了 Turbo C++ 的编辑器、鼠标支持和其他特性。这一章也是 Turbo C++ 系统的一个缩影，更为详细的内容可以参看本篇第八章、第九章和第十章。

第四章概述了 C 语言的基础知识，介绍了 C 程序的基本组成、数据和数据类型、运算符、函数、数组、结构以及其他方面的内容。

第五章介绍了面向对象的 C++ 程序设计的有关概念。

第六章讲述了快速使用 Turbo C++ 的内容。本篇第五、六两章实为一体，其中一章讲理论，另一章讲实践。

第七章介绍了集成调试器，并通过运行内部嵌有错误的范例程序，使用户可以充分了解调试的种种特性。

第八章概括了程序员平台的一些特征,提供了信息和范例,使用户可以多方面领略 IDE 的优越性.本章还包含了启动和退出 IDE 方面的内容.

第九章提供了编辑器的一个完整参考.

第十章讲述了怎样利用工程管理程序来管理多个文件的程序设计工程.

第十一章讲述了命令行编译器的使用方法,还说明了配置文件的有关内容.

第十二章描述了 MAKE 实用程序. MAKE 基本上等同于命令行的工程管理程序.

第十三章描述了 TLINK 实用程序. 同样, TLINK 也等同于命令行的集成连接程序.

下篇详尽地阐述了 Turbo C++ 的词法、语法、C++ 流、存储模式、浮点、覆盖及视频函数等内容,解释了 Turbo C++ 中体现面向对象方法的重要基础知识——类、对象、方法、重载和多态性等概念的含义和特点,从而使读者清楚地了解面向对象的程序设计方法的原理和本质.在此基础之上,本篇还着重叙述了运用 Turbo C++ 进行面向对象的程序设计的方法,提供了一些精湛的程序设计技巧,使读者能够在较高的层次上改善自己的实际工作能力和技术素质.

下篇共分十章,其中第一章到第四章讲述了 Turbo C++ 语言.这几章提供了 Turbo C++ 的 C 和 C++ 两方面的形式化语言定义、引用和语法.对于 ANSI C 标准的任何扩充,均在这几章中作了说明.各章内容分述如下:

第一章给出了 Turbo C++ 词法文法的描述.词法文法描述了一个语言可以识别的各种类似于词的单元,称为词法单元(token).

第二章提供了 Turbo C++ 短语结构文法的信息.短语结构文法详细地描述了由词法单元组成表达式、语句以及其他有意义的单元的合法途径.

第三章介绍了专用于 C++ 的内容.

第四章介绍了预处理器,其内容包括宏、头文件和编译指示,还包括其他一些浅显易懂但却非常有用的语言成分.

第五章描述了 main 函数.

第六章介绍了如何使用 C++ 2.0 版本的流库.

第七章介绍了存储模式、混合模式编程和覆盖.

第八章主要介绍了浮点函数和 BCD 函数.

第九章专门介绍了 Turbo C++ 中的正文和图形处理功能.

第十章列出并解释了全部的运行和编译时的出错信息与警告信息,并且给出了可能的解决办法.

附录 A 描述了 ANSI C 标准之中定义不严格或未曾定义的语言成分,它们将根据具体的实现而有所不同.该附录介绍了 Turbo C++ 处理这一类语言成分的做法.

附录 B 提供了运行库的一些源代码信息,列出并描述了头文件,同时还提供了运行库的交叉引用.例如,若读者希望找到有关图形处理的函数,则可在该附录中的题目“图形”之下,去查阅相应部分的内容.

参考书目中列出了一些与一般的 C, C++ 以及 Turbo C++ 相关的参考书.

本书的特点是结构清晰、先易后难,内容广泛、重点突出,注重实践,全面指导读者使用 Turbo C++. 另外,本书还详细地列出了程序设计技巧方面的有关源代码,使读者能够尽快掌握和熟练运用面向对象的程序设计方法.本书集原理知识、设计方

技巧

于一体,是指导程序设计人员运用面向对象的方法进行研究和开发的良好的辅助工具.

目前,各高等院校、企事业单位和科研机关,都在不同程度上用面向对象的方法进行软件的研究和开发工作,非常需要一本内容全面、通俗易懂、适用面广的有关 Turbo C++ 的参考书.本书将适应社会的这种需求,其指导上机实践的优越性将为各界人士所喜爱.这也正是广大 Turbo C++ 用户所关心的.可以预见,本书将适用于非常广泛的需求面.

本书可以作为程序员和计算机软件专业的本科生及研究生学习、使用 Turbo C 或 Turbo C++ 进行程序设计的参考教材,同时也是系统分析和系统设计人员在应用软件开发、系统软件开发和进行面向对象的方法研究中的一本实用参考书.

在本书的成稿过程中,我们曾得到朱文杰和蔡艳同志的热情帮助,还得了中国科学院软件研究所诸位同事、朋友的大力支持,在此,我们表示衷心的感谢.

限于水平,难免存在错误和不妥之处,请惠于批评指正.

# 目 录

## 上篇 用户指南

### 第一章 Turbo C++ 的安装

- 1.1 INSTALL 的用法 ..... (3)
- 1.2 README 文件 ..... (4)
- 1.3 HELPME!.DOC 文件 ..... (4)
- 1.4 Turbo Calc ..... (5)
- 1.5 IDE 的定做 ..... (5)

### 第二章 Turbo C++ 浏览

- 2.1 Turbo C++ 的特征 ..... (9)
- 2.2 本书的使用 ..... (10)
- 2.3 实用程序 ..... (10)

### 第三章 学习 IDE

- 3.1 IDE 简介 ..... (12)
- 3.2 第一部分:启动、装载及编辑 ..... (15)
- 3.3 第二部分:编译和运行 ..... (19)
- 3.4 第三部分:Turbo C++ 的退出 ..... (19)
- 3.5 怎样获得有关 IDE 的更多信息 ..... (20)

### 第四章 C 语言概述

- 4.1 历史的回顾 ..... (21)
- 4.2 程序设计的基本步骤 ..... (21)
- 4.3 C 程序的基本结构 ..... (23)
- 4.4 数 ..... (25)
- 4.5 变量 ..... (29)
- 4.6 表达式 ..... (41)
- 4.7 字符与字符串 ..... (43)
- 4.8 测试条件和选择 ..... (45)
- 4.9 循环 ..... (51)
- 4.10 运用函数和宏进行程序设计 ..... (58)
- 4.11 数据结构的建立 ..... (68)
- 4.12 数组的说明和初始化 ..... (69)
- 4.13 说明语句的构造 ..... (76)
- 4.14 指针 ..... (78)



4.15	系统资源的利用 .....	(84)
<b>第五章</b>	<b>C++ 入门</b>	
5.1	封装性 .....	(88)
5.2	继承性 .....	(90)
5.3	多态性 .....	(91)
5.4	重载 .....	(91)
5.5	用类构造现实世界的模型 .....	(91)
5.6	继承性 .....	(100)
5.7	虚函数 .....	(112)
5.8	动态的对象 .....	(122)
5.9	C++ 的灵活性 .....	(128)
5.10	用户定义的数据类型的 I/O .....	(141)
5.11	下一步的工作 .....	(142)
5.12	总结 .....	(142)
<b>第六章</b>	<b>C++ 进阶</b>	
6.1	更好的 C :从 C 到 C++ .....	(143)
6.2	支持面向对象的思想 .....	(146)
6.3	小结 .....	(156)
<b>第七章</b>	<b>IDE 中的程序调试</b>	
7.1	调试和程序开发 .....	(158)
7.2	一个范例程序的设计:PLOTEMP.C .....	(159)
7.3	原型程序的编写 .....	(161)
7.4	集成调试器的使用 .....	(164)
7.5	程序流的跟踪 .....	(164)
7.6	程序开发的继续 .....	(165)
7.7	断点的设置 .....	(166)
7.8	数据的检测 .....	(168)
7.9	变量的计算和改变 .....	(170)
7.10	设置监视来监控程序 .....	(173)
7.11	预防措施 .....	(176)
7.12	系统化的软件测试 .....	(177)
7.13	范例 PLOTEMP.C 的完成 .....	(178)
7.14	调试练习的答案 .....	(182)
<b>第八章</b>	<b>IDE 基础</b>	
8.1	Turbo C++ 的启动和退出 .....	(184)
8.2	IDE 部件 .....	(186)
8.3	帮助系统的使用 .....	(197)
<b>第九章</b>	<b>编辑器大全</b>	
9.1	新旧 Turbo C++ IDE .....	(199)

9.2	编辑器命令表 .....	(199)
<b>第十章</b>	<b>多文件工程管理</b>	
10.1	工程管理程序的用法 .....	(208)
10.2	出错的跟踪 .....	(211)
10.3	工程管理程序的功能 .....	(212)
10.4	多语言翻译器的使用 .....	(213)
10.5	库的抑制 .....	(214)
10.6	工程管理程序的其他特征 .....	(215)
<b>第十一章</b>	<b>命令行编译器</b>	
11.1	命令行编译器的使用 .....	(220)
11.2	编译器可选项 .....	(222)
11.3	连接程序可选项 .....	(230)
11.4	环境可选项 .....	(231)
11.5	一个范例 .....	(232)
<b>第十二章</b>	<b>程序管理工具 MAKE</b>	
12.1	MAKE 的工作过程 .....	(235)
12.2	MAKE 的启动 .....	(236)
12.3	MAKE 的一个简单运用 .....	(237)
12.4	制作文件的建立 .....	(239)
12.5	制作文件的组成 .....	(239)
12.6	注释 .....	(239)
12.7	命令列表 .....	(240)
12.8	显式规则 .....	(243)
12.9	隐式规则 .....	(245)
12.10	宏 .....	(247)
12.11	伪指令 .....	(250)
12.12	MAKE 的出错信息 .....	(254)
<b>第十三章</b>	<b>连接程序 TLINK</b>	
13.1	TLINK 的调用 .....	(258)
13.2	TLINK 的可选项 .....	(262)
13.3	TLINK 的限制 .....	(266)
13.4	TLINK 的出错信息 .....	(266)

## 下篇 程序员指南

<b>第一章</b>	<b>词法文法</b>	
1.1	空白 .....	(273)
1.2	词法符号 .....	(275)
1.3	运算符的描述 .....	(284)
1.4	说明 .....	(288)

1.5	说明的语法 .....	(293)
1.6	指针 .....	(308)
<b>第二章 短语结构文法</b>		
2.1	数组 .....	(312)
2.2	函数 .....	(312)
2.3	结构 .....	(316)
2.4	联合 .....	(320)
2.5	枚举 .....	(321)
2.6	表达式 .....	(323)
2.7	运算符的语义 .....	(326)
2.8	语句 .....	(336)
<b>第三章 C++</b>		
3.1	引用 .....	(342)
3.2	作用域访问运算符 .....	(344)
3.3	运算符 new 与 delete .....	(344)
3.4	类 .....	(345)
3.5	虚基类 .....	(352)
3.6	类的友员 .....	(353)
3.7	构造函数与析构函数 .....	(354)
3.8	构造函数 .....	(355)
3.9	析构函数 .....	(360)
3.10	运算符的重载 .....	(363)
3.11	运算符函数 .....	(364)
3.12	虚函数 .....	(367)
3.13	抽象类 .....	(368)
3.14	C++ 的作用域 .....	(369)
<b>第四章 预处理器</b>		
4.1	空伪指令 # .....	(372)
4.2	伪指令 #define 与 #undef .....	(372)
4.3	文件包含伪指令 #include .....	(377)
4.4	条件编译 .....	(378)
4.5	行控制伪指令 #line .....	(379)
4.6	伪指令 #error .....	(381)
4.7	伪指令 #pragma .....	(381)
4.8	预定义的宏 .....	(384)
<b>第五章 main 函数</b>		
5.1	main 的参数 .....	(386)
5.2	Pascal 调用习惯可选项 -p 的使用 .....	(388)
5.3	main 的返回值 .....	(389)



<b>第六章 C++ 流的使用</b>	
6.1 何谓流 .....	(390)
6.2 库 iostream .....	(390)
6.3 输出 .....	(391)
6.4 输入 .....	(395)
6.5 简单文件的 I/O .....	(396)
6.6 字符串流的处理 .....	(397)
6.7 流类的引用 .....	(398)
<b>第七章 存储管理</b>	
7.1 内存不够 .....	(409)
7.2 存储模式 .....	(409)
7.3 混合模式编程:地址修饰符 .....	(417)
7.4 覆盖(VROOMM) .....	(422)
<b>第八章 数学函数</b>	
8.1 浮点可选项 .....	(428)
8.2 复数数学函数的使用 .....	(430)
<b>第九章 视频函数</b>	
9.1 关于视频模式的几点说明 .....	(433)
9.2 关于窗口与视图区的几点说明 .....	(433)
9.3 在正文模式下的编程 .....	(434)
9.4 在图形模式下的编程 .....	(439)
<b>第十章 出错信息</b>	
10.1 运行出错信息 .....	(451)
10.2 编译出错信息 .....	(453)
<b>附录 A 专用于 ANSI 实现的标准</b> .....	(483)
<b>附录 B 运行库的交叉引用</b> .....	(491)
B.1 为何要访问运行库源代码 .....	(491)
B.2 Turbo C++头文件 .....	(491)
B.3 库例程的分类 .....	(493)
<b>参考书目</b> .....	(502)

上 篇  
用 户 指 南



# 第一章 Turbo C++ 的安装

由于我们采用了文件压缩技术,因此必须用自动安装程序 INSTALL 来安装 Turbo C++,而不只是把 Turbo C++ 文件拷贝到硬盘中. INSTALL 将自动拷贝并且把压缩了的 Turbo C++ 文件拆开,按照预定的目录结构装入. 安装盘中的 README 文件包含了一张文件分布的列表,以供参考.

注: Turbo C++ 软件包中含有两个不同的版本即 IDE 版本和 DOS 命令行运行版本. 如果你不熟悉 DOS 命令,则应在安装 Turbo C++ 之前,参考 DOS 手册.

我们假定你已熟悉 DOS 命令. 例如,你需要用 DISKCOPY 命令对所有的系统盘进行备份. 那么你就应对系统盘做一套完整的工作盘拷贝,然后将原始盘放到一个安全的地方保存起来.

本章包含下列内容:

- 在你的系统中安装 Turbo C++.
- 存取 RENAME 文件.
- 存取 HELPME! 文件.
- 一个指针指向有关 Borland Turbo Calc 程序的详细信息.
- 怎样定做 Turbo C++ (设置或改变缺省约定、颜色等等).

一旦你装入了 Turbo C++,就可以开始在 Turbo C++ 中进行深入研究了. 本篇中有些内容是为了适应某些程序设计的特殊需要而编写的. 第二章会告诉你在哪些地方可以找到有关 Turbo C++ 特征的详细信息.

## 1.1 INSTALL 的用法

INSTALL 可以检测你正在使用的硬件设备,同时恰当地配置好 Turbo C++. 它还创建必要的目录,再从系统盘(你所买的盘)中把文件传送到硬盘中. INSTALL 是能够自解释的. 下面是安装 Turbo C++ 的步骤:

(1) 在 A 驱动器中插入安装盘,键入下面的命令,然后按 Enter

```
A:INSTALL
```

(2) 在安装屏出现后按 Enter.

(3) 按照提示信息进行工作.

(4) 在安装结束之后,可能要将下面一行命令加到你的 CONFIG. SYS 文件中:

```
FILES = 20
```

并将 PATH=C:\TC\BIN 加到 AUTOEXEC. BAT 文件中去.

注意,安装完毕之后,INSTALL 将提示你阅读 README 文件中有关 Turbo C++ 的最新信息. HELPME!. DOC 文件也同样能够回答许多常见的有关技术支持的问题.

装入了 Turbo C++ 之后,你也许想运行一下 TCTOUR. TCTOUR 是有关新的 Turbo C++ 集成开发环境 IDE 中一些显著特征的指导手册,它在 TOUR 子目录中.

Turbo C++ 装入以后,如果你想开始运行,则可转到 Turbo C++ 目录下,键入 TC,再按 Enter. 否则继续阅读本章及下一章,以得到有关启动方面的重要信息.

注:若要退出 Turbo C++,则按 Alt-X.

当你运行了 IDE 后,也许希望永久地定做某些选项. TCINST 程序能够很容易地实现这一目的. 1.5 节对此有详细叙述.

如果你的系统是一个便携式计算机(带液晶显示器 LCD 或等离子显示器),则你除了要执行上述过程之外,在使用 Turbo C++ 之前,还需要设置屏幕参数. 在运行 Turbo C++ 之前,如果在 DOS 命令行上键入 MODE BW80,则会使 IDE 工作正确.

尽管你可能编了一个批处理文件来启动 Turbo C++,但是对于黑白屏幕来说,使用 Turbo C++ 的定做程序 TCINST 也能够很容易地安装 Turbo C++. 如果用了这个定做程序,就可在 Screen Modes 菜单中选择“Black and White”.

## 1.2 README 文件

README 文件包含了手册中可能没有的最新信息. 它还列出了系统盘中的所有文件以及每个文件内容的简短描述.

存取 README 文件时,按下述步骤进行:

(1) 如果你还未装入 Turbo C++,则将 Turbo C++ 盘插入 A 驱动器中;如果已装入,则直接转到第(3)步,或采用编辑窗口的方式来读 README 文件.

(2) 键入 A:,并按 Enter 键.

(3) 键入 README 并按 Enter 键,进入 README 文件以后,可以用 ↑ 和 ↓ 滚动这个文件内容.

(4) 按 Esc 键退出.

装入 Turbo C++ 后,可以在一个编辑窗口中打开 README 文件. 步骤如下:

(1) 通过在命令行键入 TC 启动 Turbo C++,按 Enter 键.

(2) 按 F10 键,选择 File | Open,键入 README 后按下 Enter 键,则 Turbo C++ 在一个编辑窗口内打开 README 文件.

(3) 阅读完 README 文件,选择 File | Quit (或继续在 IDE 中运行).

## 1.3 HELPME!.DOC 文件

在你的安装盘上包含有文件 HELPME!.DOC,该文件包含了对用户经常遇到的一些问题的解答,当你遇到困难时可以查询它. 如果用 README 程序来浏览 HELPME!.DOC,则在命令行键入:

```
README HELPME!.DOC
```

## 1.4 Turbo Calc

Turbo C++ 软件包中有一个报表程序 Turbo Calc 的源代码. 在编译这个程序之前, 阅读联机文档(TCALC.DOC)中的有关内容.

## 1.5 IDE 的定做

通过 TCINST, 可以改变 IDE 中的各种缺省设置, 例如屏幕大小、编辑模式、菜单颜色和缺省目录. TCINST 还允许指定和直接修改其他. EXE 及. PRJ 文件. 如果你未指定文件, 则 TCINST 就假定为 TC.EXE. 当 TCINST 找不到你指定的文件时, 就报告出错信息.

运用 TCINST 可以进行以下各种操作:

- (1) 在. PRJ 文件中改变编辑器可选项.
- (2) 设置路径到你的包含文件、库文件和输出文件所在的目录.
- (3) 为集成调试器选择缺省设置.
- (4) 为编译器和连接程序设置缺省约定.
- (5) 定做编辑器命令键.
- (6) 建立编辑器的缺省设置.
- (7) 设置缺省的视频显示模式.
- (8) 改变屏幕颜色.

注: Turbo C++ 随时都可以运行, 如果你不想运行 TCINST, 则不必运行它.

TCINST 的菜单与 IDE 中的菜单很相似. 用 TCINST 安装的任何选项设置, 如果在 TC.EXE 的菜单选项中出现, 那么在你装入了一个包含对该选项的一个不同设置的配置文件时, 或在你通过 IDE 的菜单系统改变了这个选项设置时, 原来的选项设置都将被抑制. 所以, 只有在没有装入任何配置文件时, 改变 TC.EXE 才有效. 由于这个原因, TCINST 允许你直接修改. PRJ 文件和 TCCONFIG.TC 文件.

### 1.5.1 TCINST 的运行

TCINST 的语法是:

```
TCINST [option] [exepath] [exename] | [configpath]
      TCCONFIG.TC | [prjpath] prjname. PRJ]
```

如果不给出路径和(或)文件名, TCINST 就在当前目录中寻找 TC.EXE. option 用于指定用彩色(/c)还是黑白(/b)来运行 TCINST. 通常, 如果 TCINST 检测出一个彩色模式下的彩色适配器, 它就以彩色方式运行. 例如, 如果你正在使用的是一个带彩色适配器的混合式监视器, 用/b 可选项就可以抑制上述缺省约定(即只是黑白).

利用 TCINST 可以修改 TCCONFIG.TC 和. PRJ 文件的局部备份. 以此方式, 还可以通过对各种工程目录下的不同配置文件, 定做系统中不同的 Turbo C++ 备份, 从而使用不同的编辑命令键、不同的菜单颜色等等.



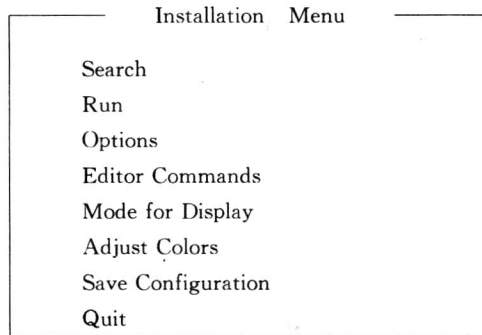
如果是在有一个 EGA 显示卡和 CGA 监视器的系统中运行 Turbo C++ ,就必须用 TCINST 来设置 Turbo C++ ,否则它就不能正常运行.参见本章 1.5.4 节有关的操作步骤.

### 1.5.2 TCINST 的安装菜单

尽管 TCINST 允许定做 Turbo C++ 中的很多组成部件,但它仅仅显示出那些你定做的菜单和选项.例如,当你在安装一个配置文件(.PRJ 或.TC)时,只有表示配置文件中值的菜单项被显示.因此,在使用 TCINST 修改 TCCONFIG.TC 文件时,编译器的可选项是不可用的;在安装一个.PRJ 文件时,颜色定做也是不可用的.

由于 TCINST 的安装菜单和选项选择与你正在修改的 IDE 中的那些部分相对应,所以,可以参考联机帮助以得到每个选项含义的详细信息.

出现于屏幕上的第一个菜单是 TCINST 安装菜单:



- Search 选项允许你存取有关查找的缺省设置.
  - Run 选项允许你设置缺省命令行参数,这些参数将传递给你的运行程序,就如同在 DOS 命令行键入的那样(但不支持重定向).这里只需给出参数就足够了,无需给出程序名.
  - Options 命令允许你存取适应于许多特性的缺省设置,包含存储模式、优化程度、出错信息显示、连接程序及环境设置,还包含头文件和库文件的目录路径名.
  - Editor Commands 选项用于定做交互式编辑器的键盘命令.如果要恢复缺省的 Editor Commands ,可以在 TCINST 主菜单中选择 E 选项,然后按下 R (Restore 缺省设置)及 Esc 键.
  - 利用 Mode for Display 就能够指明 Turbo C++ 所用的视频显示模式,还能够指明你的模式是否是一个“雪花状”视频适配器.
  - 用 Adjust Colors 菜单可以定做 IDE 中几乎每一部分的颜色.
  - Save Configuration 允许你选择是否保存对 TC.EXE 的修改;如果想对已有的修改再作一些改变,可以再次运行 TCINST.
  - Quit 选项询问是否希望退出,而不保存集成开发环境中所作的修改.
- 要选择一菜单项,只需键入给定选项的高亮度显示的大写字母.例如,按 A 以选择 Adjust Colors 选项,或者使用 ↑ 和 ↓ 键移至你要选择的亮条上,然后按下 Enter 键.

按下 Esc 键(必要时需按多次)就可返回到主菜单.

### 1.5.3 一些特别说明

虽然 TCINST 的大多数菜单项都是自解释的,但还是有一些菜单项需要作进一步的说明.

#### 1. 段名

用 Option | Compiler | Names 菜单中的菜单项,可以对 Code,Data 和 BSS 部分分别设置缺省的段名、组名和类名.当选取了其中的一个菜单项时,下一个菜单中出现的星号就告诉编译器要用缺省名.注意,不要轻易改变这些选项,除非你是一个专家.

#### 2. Debugger 菜单

Debugger 菜单中的各选项允许你对 Turbo C++ 的集成调试器设置确定的缺省设置.

在 Source 调试置成 On 的情况下,你既可用集成调试器,也可以用一个独立调试器(例如 Borland 公司的 Turbo Debugger)来对你的程序进行调试.当该选项被置成 Standalone 时,就只能使用独立调试器.当它被置成 None 时,在 .EXE 文件中就不放入任何调试信息.

Display Swapping 允许你把缺省层次设置成 None,Smart 或 Always.当用缺省设置 Smart 运行程序时,调试器就查看正在被执行的代码,以确定代码是否会影响屏幕,即是否有输出至屏幕.如果有代码输出到屏幕上(或调用了一个函数),屏幕就从 Editor 屏幕交换到 Execution 屏幕,并停留足够长的时间以显示输出,然后再交换回来;否则就不发生任何屏幕交换. Always 设置使屏幕在每次执行一条语句时都要被交换. None 设置则使调试器完全不交换屏幕.

Program Heap Size 以 K 字节为单位指明新程序的堆大小,可从 4K 至 640K 字节.

Inspector Options 有如下几种选择: Show Inherited (On 或 Off), Show Methods (On 或 Off) 和 Show Integers As. 其中, Show Integers As 给出了对 Decimal, Hex 或 Both 的选择.

#### 3. 编辑器命令

编辑器中的许多命令和按键都可以定做,它们大多数都是自解释的.定做的按键优先于初始所定义的按键,在定义按键序列时,还存在着一定的规则.其中,有些规则可应用于任何一个按键定义,而其他规则只有在一定按键模式下才起作用.

(1) 对每个给定的编辑命令,最多能按六次键.有些键组合等价于两次按键:包括 Alt (任何有效键)、光标移动键(↑, ↓, PgDn 和 Del 等等)以及所有功能键及其组合(F4, Shift-F7 和 Alt-F8 等等).

(2) 第一次按键必须是按一个控制键或特殊键,而不是字母、数字或标点符号.

(3) 如果将 Esc 键当作一命令键,则键入 Ctrl-[.

(4) 如果将 Backspace 键当作一命令键,则键入 Ctrl-H.

(5) 如果将 Enter 键当作一命令键,则键入 Ctrl-M.

(6) 预定义的 Help 功能键(F1 和 Alt-F1)不能再被置成编辑命令键.但是,其他任何功能键都可以.如果将 Turbo C++ 的一个热键作为某个编辑命令键序列的一部分,

TCINST 就会警告你正在抑制一个编辑器中的热键,并要确认你是否真希望如此.

第八章提供了 IDE 所预定义的热键的完整列表.

#### 1.5.4 视频模式的设置

通常, Turbo C++ 会正确地检测出你所用系统的视频模式. 如果以下几条中任意一条为真, 就应当只改变 Mode for Display 菜单:

- 想要选择一个和当前视频模式不同的模式.
- 有一个非“雪花状”的彩色/图形适配器.
- 认为 Turbo C++ 在错误地检测你的硬件.
- 有一个便携式计算机或一个带混合式屏幕的系统(它类似于单色的 CGA ). 在此情形下, 选择 Black and White.

如果选择 Default , 则 Turbo C++ 将始终以装入它时的那个活动模式运行.

如果选择 Color , 则 Turbo C++ 在检测出一个彩色适配器时, 不论你装入 IDE 时的活动模式是什么, 都将使用80列彩色模式; 退出时, Turbo C++ 将返回到以前的活动模式.

如果选择 Black and White , 则 Turbo C++ 在检测出一个彩色适配器时, 不论当前活动模式是什么, 都将使用80列黑白色模式; 退出时, 再切换回到原先的活动模式. 对便携式计算机和混合式监视器系统, 要用到这个选项.

如果选择 LCD or Composite, 功能完全同 Black and White .

如果选择 Monochrom , Turbo C++ 就在检测出一个单色适配器时, 不论当前活动模式如何, 都将使用单色模式.

当选择了前面四个选项中的一项时, 程序就对屏幕进行一次视频测试. 具体做法参见 Quick-Ref 行. 当选择了这些选项中的一项时, 状态行发出询问:

Conducting video test. Is your screen "snowy" now? Press any key to answer.

该状态行询问屏幕是否为“雪花状”, 要求按任一键以回答.

按下任一个键后, 可选下列之一来回答:

- Yes, 屏幕是“雪花状”.
- No , 始终关闭“雪花状”检测.
- Maybe, 始终检测硬件.

关于 Maybe 的详细说明可参阅 Quick-Ref 行. 按 Esc 返回到主菜单.