

潘金贵 尤晓东 陈世福等 编著

郑国梁 钱士均 主审

Turbo Pascal 程序设计教程

Turbo Pascal 程序设计教程

南京大学出版社

Turbo Pascal 程序设计教程

潘金贵 尤晓东 陈世福等 编著
郑国梁 钱士均 主审

南京大学出版社

1991·南京

(苏) 新登字第 011 号

Turbo Pascal 程序设计教程

潘金贵 尤晓东 陈世福等 编著

郑国梁 钱士均 主审

南京大学出版社出版

(南京大学校内)

江苏省新华书店发行 江苏丹阳练湖印刷厂印刷

开本 787×1092 毫米 1/16 印张: 31.5 字数: 780 千字

1991 年 9 月第 1 版 1991 年 9 月第 1 次印刷

印数 1-12000

ISBN 7-305-01343-9/TP·38

定价 12.60 元

责任编辑: 顾其兵

内 容 简 介

本书较系统地介绍了最新版本的 Turbo Pascal 语言及其程序设计技术。全书自然地分为三篇，共三十二章，主要内容包括：Turbo Pascal 语言的环境、各种数据类型、各种语句、Turbo Pascal 程序结构以及基本的和高级的编程技术，并给出了版本 6.0 的 Turbo Pascal 提供的所有库函数的完整信息。

本书叙述通俗易懂，内容循序渐进，示例丰富，便于自学，强调 Turbo Pascal 程序设计的特点，注重方法与技巧的训练，有助于读者良好程序设计风格的培养。

本书既可作为高等院校计算机专业的学生学习 Pascal 语言的教材或教学参考书，亦可供其它专业的学生以及从事计算机系统软件及应用软件开发的工程技术人员参考。

此外，由于近年来国内和国际青少年奥林匹克信息学竞赛都广泛采用 Pascal 语言，因此，本书对于广大的中学生中的计算机爱好者亦是一本极好的参考书。

关于与本书配套的例子盘和系统盘的信息可与编者联系。

前言

起初,沃思(N.Wirth)教授设计 Pascal 语言有两个非常明确的目的:

(1) 设计一个语言,它清楚而自然地反映某些基本概念,在这基础上,通过该语言的教学,能使程序设计成为一种系统训练。

(2) 定义一个语言,使得在一个可使用的计算机上,它的实现是可靠和高效的。

经过二十多年的研究、应用和发展,今天,不论是国内还是国外,几乎所有的(大、中、小、微)通用计算机上都有程序设计语言 Pascal 的编译系统。Pascal 语言已被公认为是定义严格、结构严谨、基于结构化程序设计原则的一种语言。它具有丰富的数据结构、简明灵活的控制结构,便于结构化设计程序,适用范围广,运行效率高等特点。1983 年原教育部颁布的教学大纲明确规定计算机专业的第一门程序设计课程采用 Pascal 语言为工具, Pascal 语言不仅是计算机专业的一门重要基础课,也是软件人员广泛使用的程序设计语言之一。

用 Pascal 语言书写的程序有许多优越性,如程序结构清晰,便于记忆,容易修改和易于移植。由于用 Pascal 语言书写的程序能清楚地反映数据结构和算法,所以非常便于交流。

因此,在 1983 年国际标准化组织(ISO)制订了 Pascal 语言的国际标准,即《计算机程序设计语言 Pascal》的国际标准文本(ISO7185-1983)。我国的《程序设计语言 Pascal》(国家标准文本)GB7591-87 也于 1987 年 12 月生效。

Turbo Pascal 是美国 Borland 国际公司推出的产品,它编译速度、运行效率高,可在 PC DOS、MS DOS 或 CP/M80、CP/M86、OS/2 等操作系统支持的 IBM PC/XT、AT、286、386、PS/2 及其兼容的个人计算机上运行。可见其应用之广泛。

其主要特点有:

一、与国际标准基本兼容,并作了若干扩充

尽管 Turbo Pascal 未明确指明,但按照 Pascal 的国际标准, Turbo Pascal 实现了 ISO7185-1983 标准的 0 级遵从。除此之外,它还扩充了许多其他功能,诸如与计算机硬件有关的绝对地址变量、机器字位及字节操作、中断处理、内存管理、外部子程序调用、程序内机器代码、内部数据格式、嵌入文件、整数逻辑操作和结构常量等等。另外,针对广泛的 IBM 微机用户,还提供了图形及色彩、窗口和声音等特殊功能,通常它们都是通过需求函数和过程来引用的。这些手段都充分发挥了微机自身的特点和功能,使得 Turbo Pascal 语言的表达能力更强、更实用。

二、用户界面好,编程效率高

用计算机解题,人们要做很多工作。在程序员按照逐步精化的原则用程序设计语言编制程序后,常规的方法是首先借助正文编辑程序编辑用户自己的程序,然后进行编译、连

接装配、运行等阶段的工作。这个过程常常循环往复，每一阶段都可能出错。一旦有错，又需要回到用编辑程序进行修改，再重复上述过程，直到得到满意的结果。这种传统的方法无统一的用户界面，多个工具分离，用户处理整个过程中在工具程序之间转来转去、调进退出，很花时间。Turbo Pascal 很好地解决了这个问题，它具有工具箱式的环境，有统一的界面。提供了集高性能文件管理、编辑、编译、调试运行为一体的集成开发环境 (IDE)。自 4.0 版本以后，这个界面是彩色多窗口的。现在的 6.0 版本，更是为之一新，给程序员带来编程效益的提高，它支持：

- 多重叠窗口；
- 鼠标、菜单、对话框；
- 多文件编辑器，可编辑的文件长度可达 1Mb；
- 增强性调试工具；
- 完整的工作面保存和恢复功能。

三、查错功能强，错误定位准

Turbo Pascal 的 IDE 交互性能好，对文件可以统一处理，直接支持编辑、编译、连接装配、运行和调试等各阶段工作。

值得特别指出的是，Turbo Pascal 在编译时的查错功能甚强。一旦发现有错，立即进入编辑状态进行修改，修改后直接转编译状态编译源程序。它的错误定位及错误性质的指示相当准确。这一点对程序员而言无疑是很受欢迎的。在 Turbo Pascal 中仅当修改了当前找到的第一个错误之后，才能继续编译下去。这一方面说明其编译算法设计得好，发挥了微机的特点，同时反映了 Pascal 语言的定义严谨。

四、单元可分别编译，有利于大型软件的开发

Turbo Pascal 中的一个单元(Unit)是常数、数据类型、变量、过程及函数的集合，每个单元很像一个独立的 Pascal 程序，它有必要的初始化代码和程序体，是允许程序分块独立编译的说明库。

单元设施可用于“软件工程”课程的教学和实践。例如，在概要设计阶段，可以利用单元的接口部分进行模块的功能描述、模块接口说明、数据的组织等；详细设计阶段，在实现部分进行各模块的细化工作；编码阶段，对单元进行分别编译；测试阶段，根据具体情况进行由底向上或自顶向下的测试组装工作。

五、带面向对象的扩充，支持面向对象程序设计(OOP)

面向对象的程序设计(简称 OOP)是在抽象数据类型的基础上发展起来的一种颇有前途的软件设计方法，它将数据抽象和类型继承融为一体，使人们在软件设计中普遍遵循的模块化、信息隐藏、抽象代码共享等软件工程思想得到充分的体现。

Turbo Pascal 自 5.5 版本起，扩充了支持面向对象程序设计的设施。在 Turbo Pascal 6.0 中又得到了进一步的改进和充实，使得 Turbo Pascal 也跻身于 OOP 的行列，成为面向对象程序设计的有力工具。

六、其它

- (1) 它规定了无类型变量，对于书写系统软件这是绝对必要的；
- (2) 说明部分的各个说明段次序可随意，十分灵活方便；
- (3) 自带的编辑器与 WordStar 命令几乎相同，广大用户不必专门为了掌握编辑程序

而花很多功夫。

此外, Turbo Pascal 6.0 还有以下一些新功能:

(4) 面向对象的应用工具 Turbo Vision, 可直接应用于用户程序(它提供了用以书写 IDE 的工具)

(5) 功能完备的嵌入式汇编器

(6) 对象说明中的私有域和私有方法

(7) 扩展的语法指令(\$X), 可将函数与过程等同对待(忽略函数的返回值)

(8) 286 代码生成

(9) 类型常量中的地址引用

(10) 远过程和近过程指令(\$G)

(11) 从目标文件中链接初始化数据(\$L)

(12) 一种新的堆管理器, 快速而且可以减小零碎的堆单元

(13) 增强的联机帮助工具, 每个库函数和过程均附有示例代码

由于 Turbo Pascal 具有如此多的优点, 它很容易满足各类用户的需要。交互式的环境使它充分体现了教学软件的特点; 功能优良、高效可靠使它成为有效的程序开发工具。因此, 它能广泛地用来书写系统软件和应用软件。

本书用通俗的语言系统地介绍了最新版本的 Turbo Pascal 语言。全书共分为三篇三十二章。

第一篇“Turbo Pascal 程序设计基础”, 它由四个部分二十六章组成, 详介了关于 Turbo Pascal 语言的基本成份及其在程序中的用法, 以及所有的标准库(System、DOS、Graph、Overlay 及 CRT 单元)和关于 8087 的信息, 介绍了内存管理、控制、输入/输出、自动优化、编译指令、工程管理、程序调试等高级编程技术, 讨论了 Turbo Pascal 与汇编语言接口的两种实现方法。

第二篇“Turbo Pascal 环境”由五章组成, 它提供了掌握新的 IDE 的速成材料和完整信息, 编辑程序、命令行编译程序的用法等。

第三篇“Turbo Pascal 6.0 的库程序”介绍了最新版的 Turbo Pascal 提供的所有库程序的详细信息。

两个附录列出了一些参考材料, 并列出了 Turbo Pascal 在编译和运行时刻产生的所有错误信息。

关于用 Turbo Pascal 语言进行面向对象程序设计以及面向对象的应用工具 Turbo Vision 的用法将作为本书的姐妹篇另册出版。

本书承蒙曾参加 Pascal 语言国家标准的制订、已从事 30 年程序设计教学、从 1977 年起就开始教授 Pascal 语言的南京大学郑国梁教授和钱士均副教授主审, 在此谨向他们深表谢意。

参加本书编著工作的人员有: 潘金贵(南京大学计算机系副教授、南京大学计算机应用研究所副所长)、陈世福(南京大学计算机系副教授, 南京大学人工智能教研室主任)、尤晓东(中国人民大学信息系教师, 工学硕士)、汪亚文(北京航空航天大学计算机系博士

)、吴卫华(江苏省统计局计算中心工程师)、冀惠刚(民航华北管理局计算机室工程师)、张勇(北京航空航天大学软件研究所教师,工学硕士)、汤建平(联想计算机公司评测中心工程师,工学硕士)。潘金贵和尤晓东同志对全书作了仔细的修改和统编。

由于水平、时间所限,不妥之处在所难免,欢迎读者批评指正。

编著者

一九九一年十月于南京

目 录

前言

上篇 Turbo Pascal 程序设计技术

第一部分 Turbo Pascal 程序设计基础

第一章 程序框架与语法描述	(1)
§ 1.1 Turbo Pascal 语言词汇集	(2)
§ 1.2 程序结构	(4)
§ 1.2.1 程序首部	(4)
§ 1.2.2 分程序	(4)
§ 1.2.3 程序中的注解	(5)
§ 1.3 语法描述	(6)
§ 1.3.1 语法	(6)
§ 1.3.2 语法图	(6)
第二章 Turbo Pascal 程序设计初步	(8)
§ 2.1 程序设计的要素	(8)
§ 2.2 数据类型	(9)
§ 2.3 操作符	(13)
§ 2.4 输出	(16)
§ 2.5 输入	(17)
§ 2.6 条件语句	(17)
§ 2.7 循环语句	(19)
§ 2.8 过程和函数	(21)
第三章 常量和常量定义	(24)
§ 3.1 常量	(24)
§ 3.2 字面常量	(24)
§ 3.3 常量定义	(26)
第四章 分程序和作用域	(28)
§ 4.1 语法	(28)
§ 4.2 作用域规则	(29)
§ 4.3 接口和标准标识符的作用域	(30)
第五章 类型和类型定义	(31)
§ 5.1 简单类型	(31)

§ 5.2	字符串类型	(35)
§ 5.3	结构类型	(35)
§ 5.4	指针类型	(43)
§ 5.5	过程类型	(44)
§ 5.6	类型的一致性和兼容性	(45)
§ 5.7	类型定义	(46)
第六章	变量与变量说明	(48)
§ 6.1	变量	(48)
§ 6.2	变量说明	(48)
§ 6.2.1	数据段	(49)
§ 6.2.2	堆栈段	(49)
§ 6.2.3	绝对变量	(49)
§ 6.3	变量引用	(50)
§ 6.4	限定词	(50)
§ 6.4.1	数组、字符串和下标	(50)
§ 6.4.2	记录和域指示符	(51)
§ 6.4.3	对象分量指示符	(51)
§ 6.4.4	指针和动态变量	(51)
§ 6.5	强制变量类型转换	(52)
第七章	类型常量	(54)
§ 7.1	简单类型常量	(54)
§ 7.2	字符串类型常量	(55)
§ 7.3	结构类型常量	(55)
§ 7.3.1	数组类型常量	(55)
§ 7.3.2	记录类型常量	(56)
§ 7.3.3	对象类型常量	(57)
§ 7.3.4	集合类型常量	(57)
§ 7.4	指针类型常量	(57)
§ 7.5	过程类型常量	(58)
第八章	表达式	(59)
§ 8.1	表达式语法	(59)
§ 8.2	操作符	(62)
§ 8.3	函数调用	(67)
§ 8.4	集合构造符	(68)
§ 8.5	值类型强制转换	(68)
§ 8.6	表达式中的过程类型	(69)
第九章	语句	(70)
§ 9.1	简单语句	(70)
§ 9.2	结构语句	(72)

第十章 过程和函数	(80)
§ 10.1 过程说明	(80)
§ 10.1.1 near、far 说明	(81)
§ 10.1.2 interrupt 说明	(81)
§ 10.1.3 forward 调用说明	(81)
§ 10.1.4 external 说明	(82)
§ 10.1.5 assembler 说明	(82)
§ 10.1.6 inline 说明	(82)
§ 10.2 函数说明	(83)
§ 10.3 方法说明	(84)
§ 10.3.1 构造函数和析构函数	(85)
§ 10.4 参数	(86)
§ 10.4.1 值参数	(87)
§ 10.4.2 变量参数	(87)
§ 10.4.3 无类型参数	(88)
§ 10.5 过程类型	(89)
§ 10.5.1 过程变量	(89)
§ 10.5.2 过程类型参数	(91)
第十一章 程序和单元	(93)
§ 11.1 程序	(93)
§ 11.1.1 程序语法	(93)
§ 11.1.2 程序头	(93)
§ 11.1.3 uses 子句	(93)
§ 11.2 单元	(94)
§ 11.3 什么是单元	(94)
§ 11.4 单元的结构	(95)
§ 11.4.1 接口部分	(96)
§ 11.4.2 实现部分	(96)
§ 11.4.3 初始化部分	(97)
§ 11.5 如何使用单元	(97)
§ 11.5.1 引用单元说明	(98)
§ 11.5.2 实现部分的 USES 子句	(101)
§ 11.6 标准单元	(103)
§ 11.7 编写用户单元	(104)
§ 11.7.1 编译单元	(104)
§ 11.7.2 示例	(105)
§ 11.7.3 单元和大程序	(106)
§ 11.7.4 用作覆盖的单元	(107)
§ 11.7.5 TPUMOVER 工具	(107)

第二部分 标准库

第十二章 System 单元	(108)
§ 12.1 标准过程和函数	(108)
§ 12.1.1 流控制过程	(108)
§ 12.1.2 动态分配过程	(108)
§ 12.1.3 动态分配函数	(108)
§ 12.1.4 转换函数	(109)
§ 12.1.5 算术函数	(109)
§ 12.1.6 序数过程	(109)
§ 12.1.7 序数函数	(109)
§ 12.1.8 串处理过程	(110)
§ 12.1.9 串处理函数	(110)
§ 12.1.10 指针和寻址函数	(110)
§ 12.1.11 其它过程	(110)
§ 12.1.12 其它函数	(110)
§ 12.2 文件的输入和输出	(111)
§ 12.2.1 文件 I/O 的简介	(111)
§ 12.2.2 文本文件	(112)
§ 12.2.3 无类型文件	(113)
§ 12.2.4 变量 FileMode	(113)
§ 12.3 Turbo Pascal 的设备	(114)
§ 12.3.1 DOS 设备	(114)
§ 12.3.2 文本文件设备	(115)
§ 12.4 预说明变量	(115)
§ 12.4.1 未初始化的变量	(115)
§ 12.4.2 初始化的变量	(116)
第十三章 DOS 单元	(118)
§ 13.1 有关的常量、类型和变量	(118)
§ 13.2 变量	(121)
§ 13.3 有关的过程和函数	(122)
§ 13.3.1 日期和时间过程	(122)
§ 13.3.2 中断支持过程	(122)
§ 13.3.3 磁盘状态函数	(122)
§ 13.3.4 文件处理过程	(122)
§ 13.3.5 文件处理函数	(123)
§ 13.3.6 进程处理过程	(123)
§ 13.3.7 进程处理函数	(123)
§ 13.3.8 环境处理函数	(123)

§ 13.3.9 其它过程和函数	(123)
第十四章 Graph 单元	(124)
§ 14.1 驱动程序	(124)
§ 14.2 当前指针(CP)	(126)
§ 14.3 正文	(126)
§ 14.4 图和属性	(127)
§ 14.4.1 视口和位象	(127)
§ 14.5 分页与颜色	(127)
§ 14.5.1 错误处理	(127)
§ 14.6 例子	(128)
§ 14.6.1 堆管理例程	(129)
§ 14.7 Graph 单元常量、类型和变量	(131)
第十五章 Overlay 单元	(141)
§ 15.1 覆盖管理	(141)
§ 15.2 常量和变量	(143)
§ 15.3 有关的过程和函数	(146)
§ 15.4 覆盖程序设计	(148)
§ 15.4.1 覆盖代码的产生	(148)
§ 15.4.2 Far 调用需求	(148)
§ 15.4.3 初始化覆盖管理模块	(149)
§ 15.4.4 覆盖单元的初始化部分	(151)
§ 15.4.5 不能用来覆盖的单元	(152)
§ 15.4.6 覆盖块的调试	(152)
§ 15.4.7 覆盖模块的外部例程	(152)
§ 15.5 在.EXE 文件中使用覆盖	(153)
第十六章 8087 的使用	(154)
§ 16.1 8087 数据类型	(155)
§ 16.2 扩展型精度运算	(155)
§ 16.3 实数的比较	(156)
§ 16.4 8087 的运算栈	(156)
§ 16.5 8087 的实数输出	(157)
§ 16.6 使用 8087 的单元	(158)
§ 16.6.1 8087 的检测	(158)
§ 16.6.2 用汇编语言仿真 8087	(159)
第十七章 CRT 单元	(160)
§ 17.1 输入和输出文件	(160)
§ 17.2 窗口	(160)
§ 17.2.1 特殊字符	(160)
§ 17.2.2 行输入	(161)

§ 17.3 有关的常量、类型和变量	(161)
§ 17.4 有关的过程和函数	(164)
第三部分 Turbo Pascal 高级编程技术	
第十八章 内存管理	(166)
§ 18.1 Turbo Pascal 内存映象	(166)
§ 18.2 堆管理程序	(166)
§ 18.2.1 析构函数	(167)
§ 18.2.2 空闲块表	(169)
§ 18.2.3 HeapError 变量	(170)
§ 18.3 内部数据格式	(171)
§ 18.4 直接内存存取	(175)
第十九章 程序控制	(176)
§ 19.1 调用约定	(176)
§ 19.1.1 变量参数	(176)
§ 19.1.2 值参数	(176)
§ 19.1.3 函数结果	(177)
§ 19.1.4 Near 和 Far 调用	(177)
§ 19.1.5 嵌套过程和函数	(177)
§ 19.1.6 入口和出口代码	(178)
§ 19.1.7 寄存器保存约定	(179)
§ 19.2 退出过程	(179)
§ 19.3 中断处理	(180)
§ 19.3.1 编写中断例程	(180)
第二十章 输入和输出	(182)
§ 20.1 正文文件设备驱动程序	(182)
§ 20.2 直接端口存取	(183)
第二十一章 自动优化	(185)
§ 21.1 常量预编译	(185)
§ 21.2 常量合并	(185)
§ 21.3 短路计算	(185)
§ 21.4 运算顺序	(185)
§ 21.5 范围检查	(186)
§ 21.6 用移位代替乘	(186)
§ 21.7 自动字边界对齐	(186)
§ 21.8 不执行代码的剔除和聪明的连接	(186)
第二十二章 编译指令	(188)
§ 22.1 开关命令	(188)
§ 22.1.1 数据对齐	(189)

§ 22.1.2	布尔运算	(189)
§ 22.1.3	调试信息	(189)
§ 22.1.4	浮点仿真	(190)
§ 22.1.5	强制 Far 调用	(190)
§ 22.1.6	产生 80286 代码	(191)
§ 22.1.7	输入 / 输出检查	(191)
§ 22.1.8	局部符号信息	(191)
§ 22.1.9	数字处理	(192)
§ 22.1.10	产生覆盖代码	(192)
§ 22.1.11	范围检查	(192)
§ 22.1.12	栈溢出检查	(193)
§ 22.1.13	变量串检查	(193)
§ 22.1.14	扩展语法的使用	(193)
§ 22.2	参数命令	(194)
§ 22.2.1	包含文件	(194)
§ 22.2.2	连接目标文件	(194)
§ 22.2.3	内存分配	(194)
§ 22.2.4	覆盖单元名	(195)
§ 22.3	条件编译	(195)
第二十三章	工程管理	(198)
§ 23.1	程序组织	(198)
§ 23.2	Build 和 Make 选项	(199)
§ 23.3	独立的实用程序 MAKE	(200)
§ 23.4	条件编译	(202)
§ 23.5	代码优化	(206)
第二十四章	调试 Turbo Pascal 程序	(208)
§ 24.1	程序错误类型	(208)
§ 24.2	Turbo Pascal 集成调试器	(209)
§ 24.2.1	调试器的功能	(209)
§ 24.2.2	进入和退出调试器	(210)
§ 24.2.3	跟踪程序	(211)
§ 24.2.4	单步执行程序	(213)
§ 24.2.5	使用断点	(214)
§ 24.2.6	监测值	(216)
§ 24.2.7	计算与修改	(220)
§ 24.2.8	游历	(222)
§ 24.3	面向对象的调试	(223)
§ 24.3.1	单步执行并跟踪方法调用	(223)
§ 24.3.2	在计算窗口中的对象	(223)

§ 24.3.3 Find Procedure 命令中的表达式	(224)
§ 24.4 有关问题	(224)
§ 24.4.1 如何写便于调试的程序	(224)
§ 24.4.2 内存问题	(225)
§ 24.4.3 递归子程序	(226)
§ 24.4.4 不能调试的代码	(227)
§ 24.4.5 常见错误	(227)
§ 24.5 错误处理	(228)
§ 24.5.1 输入 / 输出错误检测	(228)
§ 24.5.2 范围检查	(228)
§ 24.5.3 其他错误处理能力	(230)

第四部分 Turbo Pascal 与汇编语言的交互

第二十五章 直接嵌入汇编语言	(231)
§ 25.1 asm 语句	(231)
§ 25.1.1 寄存器的使用	(232)
§ 25.2 汇编语言语句的语法	(232)
§ 25.3 表达式	(238)
§ 25.3.1 Pascal 表达式与汇编表达式的区别	(238)
§ 25.3.2 表达式元素	(239)
§ 25.3.3 表达式的类	(243)
§ 25.3.4 表达式类型	(244)
§ 25.3.5 表达式操作符	(246)
§ 25.4 汇编语言过程和函数	(247)
第二十六章 连接汇编语言代码	(250)
§ 26.1 Turbo Assembler 和 Turbo Pascal	(251)
§ 26.2 汇编语言例程例子	(251)
§ 26.3 inline 机器码	(256)

中篇 Turbo Pascal 程序设计环境

第二十七章 安装和启动	(259)
§ 27.1 安装 Turbo Pascal	(259)
§ 27.2 启动 Turbo Pascal	(260)
第二十八章 熟悉新的 IDE	(261)
§ 28.1 IDE 的构成	(261)
§ 28.1.1 菜单条与菜单快速键	(261)
§ 28.1.2 Turbo Pascal 窗口	(264)
§ 28.1.3 状态行	(266)
§ 28.1.4 对话框	(267)

§ 28.1.5 编辑	(268)
§ 28.2 IDE 下程序开发举例	(269)
§ 28.2.1 例 1 程序的分析	(269)
§ 28.2.2 例 1 程序的存盘	(270)
§ 28.2.3 例 1 程序的编译	(270)
§ 28.2.4 例 1 程序的运行	(270)
§ 28.2.5 检查创建的文件	(271)
§ 28.3 修改例 1 程序	(271)
§ 28.3.1 调试修改后的程序	(272)
§ 28.3.2 纠正例 2 程序中的错误	(273)
§ 28.4 例 3.使用图形功能	(274)
第二十九章 IDE 参考指南	(277)
§ 29.1 启动 Turbo Pascal 的命令行选项	(277)
/C /D /E /G /L /N	
/O /P /S /T /W /X	
§ 29.2 ≡ (System) 菜单	(280)
About Refresh Display Clear Desktop	
§ 29.3 File 菜单	(280)
Open New Save Save As Save All	
Change Dir Print Get Info Dos Shell Exit	
§ 29.4 Edit 菜单	(284)
Restore Line Cut Copy Paste Copy Example	
Show Clipboard Clear	
§ 29.5 Search 菜单	(285)
Find(Options、Direction、Scope、Origin) Replace Search Again	
Go to Line Number Find Procedure Find Error	
§ 29.6 Run 菜单	(289)
Run Program Reset Go To Cursor Trace Into	
Step Over Parameters	
§ 29.7 Compile 菜单	(291)
Compile Make Build Destination Primary File	
§ 29.8 Debug 菜单	(292)
Evaluate / Modify Toggle Breakpoint Breakpoints	
Watches(Add Watch、Delete Watch、Edit Watch、Remove All Watches)	
§ 29.9 Options 菜单	(295)
Compiler(Code Generation、Run-time Errors、Syntax Options、	
Numeric Processing、Debugging、Conditional Defines) Memory Sizes	
Linker(Map File、Link Buffer) Debugger(Debugging、Display Swapping)	
Directories Environment(Preferences、Editor、Mouse、Startup、Colors)	