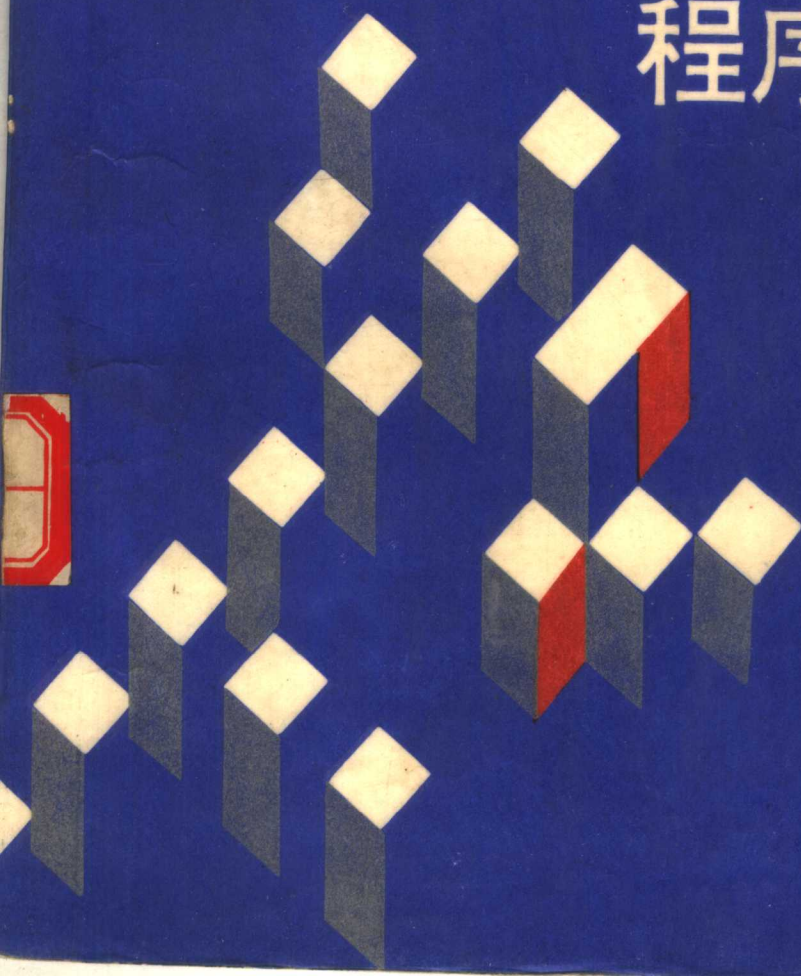


潘金贵 吴卫华 潘柔等 编著

郑国梁 费翔林 主审

Turbo Pascal 面向对象 程序设计技术 与工具



南京大学出版社

Turbo Pascal 面向对象 程序设计技术与工具

潘金贵 吴卫华 萧柔等 编著
郑国梁 费翔林 主审

南 京 大 学 出 版 社
1992 · 南京

(苏) 新登字第 011 号

内 容 简 介

本书系统地介绍了最新版本的 Turbo Pascal 语言的面向对象程序设计原理及用面向对象方法进行程序设计的工具箱 Turbo Vision。全书共分为两篇十八章。

上篇详细介绍了面向对象程序设计的概念、原理和 Turbo Pascal (6.0 版) 语言中的相应设施及其使用, 并给出一个面向对象程序设计的实例, 以及如何调试面向对象程序的技术, 还涉及了一些面向对象的高级话题。

下篇包含了关于 Turbo Vision 的大量解释材料, 以及一个完整的参考手册。第一部分介绍 Turbo Vision 的基本背景性原理, 并说明用 Turbo Vision 开发应用程序的全过程; 第二部分详细介绍 Turbo Vision 的基本组成部分, 包括对 Turbo Vision 对象层次成分的解释和开发高质量的应用程序的技巧; 第三部分是 Turbo Vision 的所有对象和其他组成部分的一个综合的参考。

本书叙述通俗易懂, 内容循序渐进, 示例丰富, 便于自学, 强调用 Turbo Pascal 进行面向对象程序设计的特点, 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面向对象程序设计原理与方法的教材或教学参考书, 亦可供从事计算机系统软件及应用软件开发的工程技术人员作为 Turbo Vision 工具箱的参考手册。

关于与本书配套的例子盘和系统盘的信息可与编者联系。

Turbo Pascal 面向对象程序设计技术与工具

潘金贵 吴卫华 潘柔等 编著

郑国梁 费翔林 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 江苏建潮印刷厂印刷

*

开本: 787×1092 1/16 印张: 26.5 字数: 660 千

1992 年 5 月第 1 版 1992 年 5 月第 1 次印刷

印数 1-10, 000

ISBN 7-305-01538-5/TP·45

定价: 12.00 元

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,它编译速度快、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、其他

- (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。

Borland 公司在开发了一系列窗口、会话窗和菜单式程序之后, 就考虑用面向对象的程序设计理论作指导, 把所有这些功能组合成为一个可以重复使用的工具箱。于是, 导致了 Turbo Vision 产品的推出。

Turbo Vision 是一个窗口式的、面向对象的程序骨架, 可以在这种程序骨架的基础上构造应用程序, 而不必重复编写程序的基本骨架, 从而节省大量的程序开发时间。

Turbo Vision 特别适合开发高效的、灵活的、有一致用户界面的基于字符的应用软件。例如, 用 Turbo Vision 只需极短的时间就可写出一个新的 Turbo Pascal 集成环境(IDE)。用 Turbo Vision 来开发用户的应用软件更是得心应手。

Turbo Vision 是一个完整的面向对象的函数库, 包括:

- 多层次的、可定义大小的覆盖式窗口
- 下拉菜单
- 鼠标支持
- 会话窗
- 内部颜色设置
- 按钮、滚行条、输入窗口、检查方框及单选按钮
- 标准的击键输入处理和鼠标器输入处理
- 其他

用 Turbo Vision 开发的应用软件在外观上具有上述这些现代程序的技术特性和功能特性, 而开发工作却是非常轻松的。

用 Turbo Vision 开发应用软件扩展了面向对象的程序设计技术, 包括继承性和多态性等。为此, 首先需要了解面向对象程序设计的原理, 这在本书的上篇中作了详尽的阐述。

由于 Turbo Pascal 及其面向对象程序设计工具 Turbo Vision 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 语言的面向对象程序设计原理及用面向对象方法进行程序设计的工具箱 Turbo Vision。全书共分为两篇十章。

上篇由四章组成，详细介绍了面向对象程序设计的概念、原理和 Turbo Pascal (6.0 版) 语言中的相应设施及其使用；并给出一个面向对象程序设计的实例，以及如何调试面向对象程序的技术，还涉及了一些面向对象的高级话题。

由于 Turbo Vision 是一个全新的工具箱，许多程序开发人员可能还不太熟悉它所采用的一些技术，因此本书下篇中包含了大量的解释材料，以及一个完整的参考手册。

下篇共分三个部分：

- 第一部分介绍 Turbo Vision 的基本的背景性原理，并说明用 Turbo Vision 开发应用程序的全过程。
- 第二部分详细介绍 Turbo Vision 的基本组成部分，包括对 Turbo Vision 对象层次成分的解釋和开发高质量的应用程序的技巧。
- 第三部分是 Turbo Vision 的所有对象和其他组成部分的一个综合的参考。

在学习下篇之前要熟悉指针类型和动态内存分配，几乎所有 Turbo Vision 的对象都是通过堆进行动态内存分配的。还需要回顾一下 New 函数的扩展语法，New 函数的扩展语法允许将构造方法作为 New 函数的参数，大多数 Turbo Vision 的对象是通过这种方式创建的。

关于用 Turbo Pascal 语言进行程序设计的原理、技术及练习请阅读南京大学出版社出版的本书姐妹篇《Turbo Pascal 程序设计教程》和《Pascal 程序设计习题精选与题解》。

参加本书编写工作的人员还有欣超、陈斌黎、冀惠刚、潘旭燕、王勇、李东、尹刚等同志，潘金贵和潘柔同志对全书进行了仔细的修改和统编。

本书承蒙曾参加 Pascal 语言国家标准的制订、已从事 30 多年程序设计教学、从 1977 年起就开始教授 Pascal 语言、目前正在从事 OOP 方法研究的南京大学郑国梁教授和费翔林副教授主审，在此谨向他们深表谢意。

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

一九九一年五月于南京

目 录

上篇 Turbo Pascal 面向对象程序设计

第一章 面向对象程序设计概述	(3)
1.1 什么是对象	(3)
1.2 继承	(4)
1.3 对象与记录的主要区别	(5)
1.3.1 对象类型实例	(6)
1.3.2 对象域	(6)
1.3.3 避免直接访问对象的数据域	(7)
1.4 方法	(7)
1.4.1 代码/数据封装	(8)
1.4.2 方法定义	(9)
1.4.3 方法的作用域与 self 参数	(10)
1.4.4 对象的数据域与方法的形式参数	(11)
1.4.5 在单元中定义对象	(11)
1.4.6 充满活力的编程方法	(14)
1.4.7 封装	(15)
1.4.8 增加方法不会降低程序性能	(16)
1.4.9 扩展对象	(16)
1.4.10 继承静态方法	(18)
1.4.11 虚方法及其多态性	(19)
1.4.12 前期约束与后期约束	(20)
1.4.13 对象类型的兼容性	(21)
1.4.14 多态对象	(22)
1.4.15 虚方法	(23)
1.4.16 后期约束示例	(25)
1.4.17 使用过程和方法的选择	(26)
1.4.18 对象的扩展性	(33)
1.4.19 使用静态方法或虚方法的选择	(35)
1.4.20 动态对象	(36)
1.4.21 用 NEW 分配和调用对象	(36)
1.4.22 释放动态对象	(37)

1.4.23 析构函数	(37)
1.4.24 动态对象分配示例	(39)
1.5 几点建议	(45)
1.6 结论	(46)
第二章 Turbo Pascal 面向对象程序设计实例	(47)
2.1 窗口对象	(47)
2.1.1 屏幕类	(47)
2.1.2 屏幕窗口	(48)
2.1.3 镶边窗口	(49)
2.1.4 转换类	(58)
2.2 屏幕对象及屏幕类编码	(59)
第三章 面向对象的调试	(70)
3.1 IDE 中面向对象的调试	(70)
3.1.1 单步执行并跟踪方法调用	(70)
3.1.2 在计算窗口中的对象	(70)
3.1.3 Find Procedure 命令中的表达式	(71)
3.2 在 Turbo Debugger 中调试面向对象的功能	(71)
3.2.1 步进和跟踪方法调用	(71)
3.2.2 作用域	(71)
3.2.3 Evaluate 窗口	(72)
3.2.4 Watch 窗口	(73)
3.2.5 Hierarchy 窗口	(73)
3.2.6 对象类型/类考察窗口	(74)
3.2.7 对象实例考察窗口	(75)
第四章 关于对象的进一步讨论	(78)
4.1 对象的内部数据格式	(78)
4.1.1 虚方法表	(79)
4.1.2 SizeOf 函数	(80)
4.1.3 TypeOf 函数	(80)
4.1.4 虚方法调用	(81)
4.2 方法调用约定	(81)
4.2.1 构造函数和析构函数	(82)
4.2.2 New 和 Dispose 的扩充	(82)
4.3 汇编语言方法	(83)
4.4 构造函数的纠错	(84)

下篇 面向对象程序设计工具 Turbo Vision

第一部分 Turbo Vision 入门

第五章 继承程序骨架	(95)
5.1 窗口程序的骨架	(95)
5.2 开发应用软件的一种全新的观点	(95)
5.3 Turbo Vision 应用程序的组成	(96)
5.3.1 名词	(96)
5.3.2 一致的外观和感觉	(97)
5.4 Turbo Vision 风格的“Hello, World!”演示程序	(98)
5.4.1 运行 HELLO.PAS	(101)
5.4.2 下拉菜单	(102)
5.4.3 会话框	(102)
5.4.4 按钮	(103)
5.4.5 退出	(103)
5.5 HELLO.PAS 程序	(103)
5.5.1 应用程序对象	(103)
5.5.2 会话框对象	(104)
5.5.3 程序的运行和调试	(105)
5.5.4 HELLO.PAS 的主程序	(105)
5.5.5 Init 方法	(105)
5.5.6 Run 方法	(106)
5.5.7 Done 方法	(106)
5.6 小结	(106)
第六章 编写 Turbo Vision 应用程序	(107)
6.1 第一个 Turbo Vision 程序	(107)
6.2 工作面、菜单条和状态行	(108)
6.2.1 工作面	(109)
6.2.2 状态行	(109)
6.2.3 菜单条	(111)
6.2.4 结构注释	(113)
6.3 打开一个窗口	(113)
6.3.1 标准窗口设备	(113)
6.3.2 窗口初始化	(115)
6.3.3 窗口操作	(116)
6.3.4 浏览窗口	(116)
6.3.5 带缓冲区的显示	(121)

6.3.6	上下滚动	(123)
6.3.7	一个窗口内的多个视图	(125)
6.3.8	把功能放进视图	(127)
6.4	创建会话框	(127)
6.4.1	执行模式会话框	(130)
6.4.2	获得控制	(130)
6.4.3	选取	(132)
6.4.4	标识控制	(134)
6.4.5	输入行对象	(134)
6.4.6	设置和读取数据	(135)
6.4.7	简捷键及冲突	(137)
6.4.8	结束会话	(138)
6.5	其他会话框控制	(139)
6.5.1	静态文本	(139)
6.5.2	表显示器	(139)
6.5.3	表框	(139)
6.5.4	历史输入	(139)
6.6	标准会话框	(139)

第二部分 Turbo Vision 程序设计

第七章	对象层次	(141)
7.1	对象类型学	(142)
7.1.1	抽象对象	(142)
7.1.2	抽象方法	(143)
7.2	对象的例化和衍生	(143)
7.2.1	例化	(143)
7.2.2	衍生	(143)
7.3	Turbo Vision 方法	(144)
7.3.1	抽象方法(Abstract methods)	(144)
7.3.2	伪抽象方法(Pseudo-abstract methods)	(144)
7.3.3	虚方法(Virtual Methods)	(144)
7.3.4	静态方法(Static Methods)	(144)
7.4	Turbo Vision 数据域	(144)
7.5	原始对象类型	(145)
7.5.1	TPoint	(145)
7.5.2	TRect	(146)
7.5.3	TObject	(146)
7.6	视图	(146)
7.6.1	视图概述	(146)

7.6.2 视图组	(146)
7.6.3 终端视图	(148)
7.7 不可见成分	(150)
7.7.1 流式文件	(150)
7.7.2 资源(Resource)	(151)
7.7.3 群(Collections)	(151)
7.7.4 字符串表	(152)
第八章 视图	(153)
8.1 控制 Turbo Vision	(153)
8.2 简单视图对象	(153)
8.2.1 调整视角	(154)
8.2.2 TRect 对象	(154)
8.2.3 显示视图	(155)
8.2.4 添加最好的功效	(156)
8.3 复杂视图	(156)
8.3.1 视图组和子视图	(156)
8.3.2 视图组的进一步讨论	(157)
8.3.3 视图组画像	(159)
8.3.4 视图之间的联系	(159)
8.3.5 子视图和视图树	(161)
8.4 受选视图和聚焦视图	(164)
8.4.1 找出聚焦视图	(164)
8.4.2 视图如何接受聚焦	(165)
8.4.3 聚焦链	(165)
8.5 模式视图	(165)
8.6 修改默认情况	(166)
8.6.1 Options 标志字	(166)
8.6.2 GrowMode 标志字节	(168)
8.6.3 DragMode 标志字节	(169)
8.6.4 State 标志字和 SetState 方法	(170)
8.7 视图的颜色	(172)
8.7.1 调色板	(172)
8.7.2 调色板内部	(173)
8.7.3 GetColor 方法	(174)
8.7.4 重载默认颜色值	(174)
8.7.5 增加新颜色	(175)
第九章 事件驱动程序设计	(176)
9.1 使用 Turbo Vision	(176)

9.2 事件的本来面目	(177)
9.2.1 事件的种类	(177)
9.2.2 事件和命令	(179)
9.3 事件的传送	(179)
9.3.1 事件的来源	(179)
9.3.2 事件的去向	(179)
9.3.3 屏蔽事件	(181)
9.3.4 相位(Phase)	(181)
9.4 命令	(183)
9.4.1 定义命令	(183)
9.4.2 联结命令	(184)
9.4.3 使命令“可执行”或“不可执行”	(184)
9.5 处理事件	(184)
9.6 事件记录	(185)
9.6.1 事件的清除	(186)
9.6.2 事件的重载	(186)
9.7 事件修改机制	(187)
9.7.1 中央控制的事件采集	(187)
9.7.2 重载 GetEvent 方法	(187)
9.7.3 使用 Idle 方法	(188)
9.8 视图间的通讯	(188)
9.8.1 媒介视图(Intermediaries)	(188)
9.8.2 视图之间的消息	(189)
9.8.3 谁处理广播事件	(190)
9.8.4 HandleEvent 方法的调用	(191)
9.8.5 求助上下文	(191)
第十章 编写安全的程序	(192)
10.1 原子操作程序设计	(192)
10.1.1 安全区	(192)
10.1.2 非内存出错	(194)
10.1.3 “消费大户”视图	(195)
第十一章 群	(204)
11.1 群对象	(204)
11.1.1 群的动态定长特性	(204)
11.1.2 群的多态性	(205)
11.1.3 类型检查与群	(205)
11.2 创建群	(205)
11.3 “重复”方法	(207)

11.3.1	ForEach 方法	(207)
11.3.2	FirstThat 方法和 LastThat 方法	(207)
11.4	排序群	(208)
11.5	字符串群及“重复”方法的进一步讨论	(210)
11.6	多态性群	(211)
11.7	群与内存管理	(214)
第十二章	流	(215)
12.1	问题：对象输入 / 输出	(215)
12.2	解决办法：流	(215)
12.2.1	流的多态性	(216)
12.2.2	流处理对象	(216)
12.3	流的基本应用	(216)
12.3.1	创建流	(216)
12.3.2	读写流	(217)
12.3.3	关闭流	(218)
12.4	一起使用对象和流	(218)
12.4.1	Load 方法和 Store 方法	(218)
12.4.2	定义流注册	(219)
12.4.3	流注册	(220)
12.5	流机制	(220)
12.5.1	Put 的操作流程	(220)
12.5.2	Get 的操作流程	(220)
12.5.3	处理空对象指针	(221)
12.6	流上的群：一个完整的例子	(221)
12.6.1	添加 Store 方法	(221)
12.6.2	注册记录	(222)
12.6.3	注册	(223)
12.6.4	写进流	(223)
12.7	确定存流的结构	(224)
12.7.1	子视图实例	(224)
12.7.2	同等视图实例	(225)
12.8	存储和加载工作面	(226)
12.9	拷贝流	(226)
12.10	随机存取流	(227)
12.11	流中的非对象数据	(227)
12.12	设计自己的流、流出错处理	(227)
第十三章	资源文件	(228)
13.1	为什么要使用资源文件	(228)

13.2	资源文件中有些什么	(228)
13.3	创建一个资源文件	(229)
13.4	读取资源	(230)
13.5	字符串表及其建立	(231)
第十四章	Turbo Vision 程序的调试和移植技巧	(232)
14.1	调试 Turbo Vision 程序	(232)
14.1.1	在 HandleEvent 方法中设置断点	(232)
14.1.2	在调用的程序中设置断点	(233)
14.1.3	挂起	(233)
14.2	移植成 Turbo Vision 程序	(234)
14.2.1	净化原来的程序代码	(234)
14.2.2	重新考虑程序的构造	(234)
14.3	使用位映像域	(235)
14.3.1	标志值	(235)
14.3.2	位掩码	(235)
14.3.3	位操作	(235)
14.4	小结	(236)

第三部分 Turbo Vision 参考

第十五章	如何使用“Turbo Vision 参考”	(237)
15.1	如何找到需要的内容	(237)
15.2	对象概述	(237)
15.3	取名约定	(237)

第十六章	单元交叉参考	(239)
16.1	Objects 单元	(239)
16.2	Views 单元	(241)
16.3	Dialogs 单元	(244)
16.4	App 单元	(245)
16.5	Menus 单元	(245)
16.6	Drivers 单元	(246)
16.7	TextView 单元	(250)
16.8	Memory 单元	(250)
16.9	HistList 单元	(251)

第十七章	对象参考	(252)
-------------	-------------------	----------------

第十八章	综合参考	(364)
-------------	-------------------	----------------

上 篇

Turbo Pascal 面向对象程序设计

