LEGEND

最新 Turbo Pascal 6.0

Turbo Vision 指南

译·红艳　校·润华

北京联想计算机集团公司　1991·5

最新 Turbo Pascal 6.0

Turbo Vision 指南

译 妙生 叶舟 校 润华

北京联想计算机集团公司 1991.3

译者序

随着面向对象的程序设计技术的日益成熟，其思想已逐渐应用到程序设计语言中。一九年九0年底，Borland公司隆重推出了Turbo C++和Turbo Pascal 6.0，这些语言都溶进了面向对象的程序设计思想。

Turbo Pascal 6.0与以前的版本完全兼容，但增加了许多新的特征。这些特征包括：

- ☐ 在5.5版的基础上，进一步扩展了面向对象的程序设计；
- ☐ 提供了一个功能强大的、面向对象的软件开发工具Turbo Vision，利用它可以开发出用户界面一致的交互式应用程序，大大地节省了开发应用程序界面所需的时间；
- ☐ 提供了一个用Turbo Vision开发的全新Turbo Pascal集成开发环境(IDE)，它支持多窗口、鼠标、菜单和对话，提供了多文件编辑器、增强型调试工具和磁盘管理工具；
- ☐ 提供了功能完备的嵌入式(inline)汇编器；
- ☐ 支持对象中的私有域和私有方法；
- ☐ 增加了扩展伪指令$X，可等同对待函数和过程(忽略函数的返回值)；
- ☐ 支持286代码的生成；
- ☐ 提供了增强型联机帮助工具，可以查阅每个库函数和过程。

为了满足广大用户的需要，我们编译了这套Turbo Pascal 6.0丛书。这套丛书包括《最新Turbo Pascal 6.0用户指南》、《最新Turbo Pascal 6.0程序员指南》、《最新Turbo Pascal 6.0库函数参考手册》和《最新Turbo Vision指南》。其中：

《用户指南》讲述了如何安装、掌握和使用Turbo Pascal的集成开发环境及命令行编译器，其中包括Turbo Pascal程序设计的基本知识和某些高级程序设计技术，如面向对象的程序设计和大项目管理等。

《程序员指南》讨论Turbo Pascal程序设计技术，详细描述了语言定义、标准库函数，高级程序设计技术，以及汇编语言接口，列出并说明了编译和运行时所产生的所有错误信息。

《库函数参考手册》按字母顺序列出了Turbo Pascal 6.0的所有过程与函数，并给出了每个过程和函数的语法、说明、返回值、相关的例程，以及演示该例程的例子。

《Turbo Vision指南》介绍了Turbo Pascal 6.0提供的应用程序开发工具Turbo Vision的基本概念，循序渐进地指导用户利用Turbo Vision来开发出交互式的多窗口应用软件。该书还列出并详细描述了Turbo Vision中的所有对象、过程、函数和类型。

北京联想计算机集团公司资料部陈淑华经理对本套资料的出版进行了周密的组织工作，在此表示真诚的谢意。

一九九一年三月于北京

目 录

第一部分　学习 Turbo Vision

第一章 继 承

您最后编写的应用程序中有多少是“肉”？多少是“骨头”？

应用程序的“肉”是求解该程序所要解决问题的部分，如计算、数据库操作等。另一方面，应用程序的“骨头”是指将程序中所有东西结合到一起的部分，如菜单，编辑区，错误报告，鼠标器处理等。如果您的应用程序和大多数应用程序一样，您在编写“骨头”比起编写“肉”时会花费同样或更多的时间。虽然这种程序内部结构通常能用于任何程序，但出于习惯，大多数程序员在开始每一个新的项目时总是编写差别很小的新编辑器、菜单管理器和事件处理器等。

也许经常有人告诫您避免重新开发同样的旧程序，在这里 Turbo Vision 给您提供了停止重复开发的机会，并开始继承。

1.1 窗口应用程序的骨架

Turbo Vision 是事件驱动、窗口应用程序的框架。它没有前面所说的“肉”，只是一个强壮和柔性的骨架。您可以用 Turbo Pascal 面向对象编程的扩展特性给骨架加上“肉”。Turbo Vision 给您提供了一个骨架应用程序对象：TApplication，您可以生成一个 TApplication 的后裔对象作为您的应用程序，如叫做 MyApplication。然后在 MyApplication 上加上完成工作所需要的部分。

在最高层次上，这样做就行了。应用程序中整个 begin…end 块有如：

```
begin
  MyApplication. tnit;                    {建立应用程序}
  MyApplication. Run;                     {运行}
  MyApplication. Done                     {运行完后放弃}
end.
```

1.2 应用程序开发的一种新观点

您以前可能用过过程或函数库，乍看起来，Turbo Vision 很象传统的库。毕竟可以买到提供菜单、窗口和鼠标器处理等的库程序。但在这种表面现象下有着截然的区别，为了不至于在一些高水平和高难度的概念上碰壁，这一点是值得理解的。

要提醒您自己的第一件事情就是您现在处在一个对象的领域。在传统的结构化编程中，当一个工具（如菜单管理器）不完全适合您的需求时，您会修改该工具的源代码，直至适合为止。除非您记下代码原来的确切形状，否则进入并修改工具的源代码是一个难以恢复的冒失步骤。而且改变已证实过的源代码（特别是别人编写的源代码）很容易将讨厌的新错误带进已证实过的子系统，错误可以超出您原来涉及的区域传播到很远的地方。

利用 Turbo Vision，您大可不必修改实际的源代码。您靠扩展来“改变”它。TApplication 应用程序骨架在 APP. TPU 内保持不变。您可以用派生新对象类型的方法给它加上内容，用您为新对象写出的新方法取代被继承的旧方法。

Turbo Vision 又是一个层次结构，它并不只是一个互不衔接、充满工具的工具箱。如果您

要使用其中之一，您必须使用全体。在 Turbo Vision 的每一个组件后只有一种结构的视觉，它们用很多精巧和联锁的方式一起工作。您不应只“抽出”鼠标器支持并使用它——“抽出”工作会比您自己从头开始写鼠标器支持程序还要多。

全部使用面向对象的技术和将整个 Turbo Vision 包容在自身术语上这两项提议是 Turbo Vision 开发方式的基础。这意味着按 Turbo Vision“规则”运行，需要时就使用它的组件对象类。我们开发 Turbo Vision 以节省您大量不必要和重复的工作，并为您提供一个证实了的应用程序框架。为了获得最大的收益，让我们使用 Turbo Vision 吧！

1.3 Turbo Vision 应用程序的要素

在看到 Turbo Vision 应用程序怎样运行之前，先让我们看看“工具箱中有什么”——Turbo Vision 给你提供什么工具去建立您自己的应用程序。

1.3.1 组件的命名

一个 Turbo Vision 应用程序是一组协调的视口、事件和哑对象。

1.3.1.1 视口

一个视口是在屏幕上可见的任一程序要素，所有这样的要素都是对象。在 Turbo Vision 的上下文中，你所能看到的一个要素就是一个视口。域、域标题，窗口边界，卷滚条，菜单条和对话框都是视口。多个视口可以结合成为更复杂的要素，如窗口和对话框。这些视口集合称为组，它们象单个视口一样一起操作。概念上可以将组作为视口考虑。

视口通常是长方形的。它包括包含单个字符的长方形，及一个字符高或一个字符宽的行或列。

第四章将详细说明视口。

1.3.1.2 事件

一个事件是某种应用程序必须响应的发生事物。事件来于键盘，鼠标器或 Turbo Vision 别的部分。例如击键是一个事件，按鼠标器按钮也是一个事件。当事件发生时，Turbo Vision 应用程序骨架将把它们排入队列，然后由事队处理器按顺序处理。TApplication 对象是应用程序主体，它包含一个事件处理器。通过一个机构，TApplication 不服务的事件被传送到程序拥有的其它视口，直到找到一个视口处理该事件或出现“废弃事件”错误为止。

例如，击 F1 键启动求助系统。击 F1 键将由主程序的事件处理器处理，除非每一个视口有通向求助系统的入口（在上下文相关求助系统中可能发生）。相反，一般的字符数字键或行编辑键须由当前激活的视口处理，即由当前与用户交互的视口处理。

第五章将详细解释事件。

1.3.1.3 哑对象

哑对象是程序中任一其它不是视口的对象。它们是“哑”对象的原因在于它们自己不对屏幕讲话。哑对象执行计算和外设通讯以及通常做应用程序方面的工作。当哑对象需要将一些输出结果显示到屏幕上时，它必须通过和某个视口的协作。这个概念对于应用程序保持

正常秩序是非常重要的，只有视口可以通向显示。

哑对象用 Turbo Pascal 的 write 和 writeln 语句对屏幕写操作时不会受到任何阻碍。然而，如果您按自己方法对屏幕进行写操作时，您所写的正文将破坏 Turbo Vision 所写的正文，并且 Turbo Vision 所写的正文（例如通过移动窗口或改变窗口大小）将抹去这种“叛逆”正文。

1.3.2 共同的外观与感觉

因为设计 Turbo Vision 的目的是给屏幕设计带来一种标准和理性的方法，所以您的应用程序会获得一种熟悉的外观与感觉。这种外观与感觉等同于 Turbo 语言系列本身的外观与感觉，并且它是基于多年的经验和可用性测试得来的。应用程序有一个共同的和好理解的外观对于用户和您本人都是一个突出的优点。不管应用程序所做的工作多么神密，使用它的方法却是熟悉的，学习曲线也会上升得快。

图 1.1 表明了一组可能作为 Turbo Vision 成份出现的普通对象。桌面为阴影背景，在其上有其它的应用出现。象 Turbo Vision 中其它别的任何东西一样，桌面也是一个对象。顶部显示的菜单条和底部显示的状态行也同样是对象。菜单条上的单词表示菜单，菜单由鼠标移到单词上加按钮或按热键“拉下”。

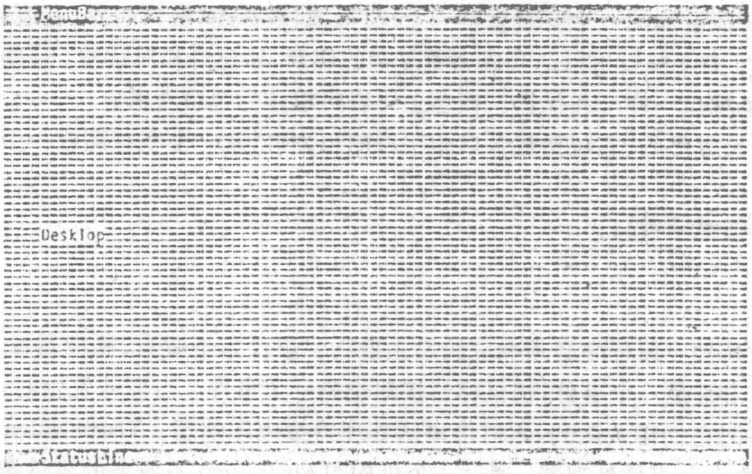

图 1.1 Turbo Vision 屏幕对象

状态行中出现的正文由您决定，但它通常用于显示应用程序当前状态的信息，拥有的热键或用户当前可用的命令提示。

当菜单被拉下时，根据鼠标或光标键的运动，一个醒目条在菜单的一列选择项上上下滑动。当按下 Enter 键或按鼠标器左按键时，选中醒目的选择项。选择一个菜单项会将一个命令传送到应用程序的某一部分。

应用程序一般通过一个或多个窗口及对话框和用户通讯，这些窗口或对话框根据鼠标器或键盘命令出现或消失。Turbo Vision 为进入和显示信息提供各种各样的窗口机构。窗口

内部可以设置为可滚动的，这使窗口能作为象文档文件这样的大型数据显示的门面。在数据上滚动窗口由移动窗口底部、右部或这两个部分的卷滚条来完成。卷滚条表示窗口相对于被显示数据整体的相对位置。

对话框经常包含按钮，它们是能由鼠标选取的醒目显示的单词（或按制表键 Tab 移动醒目显示的单词后按下空格键）。随着鼠标器按钮的按下，显示的单词也象被压下一样，并可以设置为向应用程序传送一个命令。

1.4 “Hello，world”的 Turbo Vision 风格

演示怎样使用一种新语言或用户接口工具箱的传统方法是给出一个用该工具编写的“Hello，world”程序。这个程序通常只由足以在屏幕上显示字符串“Hello，world”和返回 DOS 的代码组成。

Turbo Vision 将用一种不同的方法说“Hello world！”。

经典的“Hello，world”程序不是交互式的（它只“说”不“听”），然而 Turbo Vision 是一种生成交互程序的工具。

Turbo Vision 最简单的应用程序比在 begin 和 end 之间夹一条 writeln 所涉及的内容要多得多。比较传统的“Hello，world”程序，Turbo Vision 的 HELLO. PAS 程序要做大量的工作，它们包括：

- ☐ 清除桌面，使它成为半色调图案
- ☐ 在屏幕的上部和下部分别显示菜单条和状态行
- ☐ 为击键和鼠标器事件建立一个处理器
- ☐ 在幕后建立一个菜单对象，并把它连接到菜单条上
- ☐ 建立一个幕后对话框
- ☐ 将对话区连接到菜单上
- ☐ 等待鼠标器或键盘的动作

以上没有向屏幕上写正文的任何内容。虽然已准备了一些正文，但都在幕后等待发命令来调用。在学用 Turbo Vision 时必须牢记：用 Turbo Vision 编程的实质是设计一个惯用的视口，并当它接收到命令时教它做什么。Turbo Vision（框架）所做的是为视口取得适当的命令。您只要决定击键鼠标器按钮按下或菜单命令进入视口代码时做什么。

您所编程序的“肉”是根据用户命令执行有意义工作的代码，这些“肉质”代码包含在你生成的对象中。

“Hello，world”源代码在 Turbo Vision 盘的 HELLO. PAS 文件中。

1.4.1 运行 HELLO. PAS

在详细剖析 HELLO. PAS 之前，最好装载该程序，对其进行编译并观察其执行过程。

运行时，Hello 清除屏幕，生一个如图 1.2 所示的桌面。没有打开的窗口，屏幕顶部的菜单条上只有一项：Hello 命令。注意 Hello 中的“H”和“ello”设置为不同的颜色，状态行包含一条信息 Alt－X Exit。

这里是指出在用户环境中编程的两条普通规则的最佳时机：不要让用户不知道下一步要做什么和给用户提供进退的方法。在做其它事情以前，Hello 有两种明确的选择：选择菜单