

轻松学习

Delphi 3.0

编程

人民邮电出版社

内 容 提 要

Delphi3.0 是 Borland 公司推出的新一代软件开发工具。它是运行在 Windows3.X、Windows 95 或 Windows NT 环境下面面向对象可视化前端开发工具。

本书深入浅出地介绍了 Delphi3.0 编程的主要特点及新技术。包括如何安装 Delphi3.0 以及 Delphi3.0 的集成开发环境、语言基础、菜单程序设计以及多文档窗口设计等。特别是 Delphi3.0 中的对象向导机制(Object Wizard)、Object Pascal 语言、程序自动生成技术、可视化的数据库、多媒体技术更是其他可视化开发工具无法比拟的。

本书是一本关于 Delphi3.0 的循序提高的图书，既有浅显的讲解，又有大量的示例程序。读者通过本书的学习，可以快速地成为一名优秀的 Delphi3.0 程序开发人员。

本书适合计算机爱好者或程序开发人员学习参考。

轻松学习 Delphi 3.0 编程

- ◆ 编 著 李晓华
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:25.25
字数:632 千字 1998 年 10 月第 1 版
印数:1-5 000 册 1998 年 10 月北京第 1 次印刷

ISBN 7-115-07160-8/TP·704

定价:35.00 元

前 言

在人类文明社会即将跨入二十一世纪之际计算机技术和现代化通信技术,推动着人类社会的高速发展。计算机技术中 Delphi3.0 的出现,给原本已经火热的可视化(Visual)、面向对象(Object)编程技术环境注入了新的活力。Delphi 是 Borland 公司推出的面向对象的可视化开发工具,一经推出便引起全球的关注。它是 Borland 公司积累十余年制作编译器的经验,结合视觉化与阶段式数据库的技术所推出的新一代产品。它拥有世界上最快的编译器,并提供了功能多样的构件、灵活的代码自动生成器、功能强大的数据开发和多媒体技术等,从而使所开发的应用程序令人耳目一新。

Delphi 作为比较成熟的商品化软件,2.0 版于 1996 年正式进入市场。它是继 Windows95 宣布后不久推出的全新的 32 位编译器,之后为适应 Internet 技术发展需要,Borland 公司于 1996 年 6 月宣布了 Delphi2.01 版本,这个版本提供了新的构件,并支持 Internet 技术,它是 Delphi3.0 的前身。

Borland 公司于 1997 年 7 月正式提出 Delphi3.0 版本。在这一新版本里,它提供了功能更多、更强大的开发工具。

Delphi3.0 是目前 Windows 环境下最好的开发工具。其主要优点主要体现在以下几个方面:

- 保持了可视化的、面向对象的集成开发环境。
- 为开发人员提供了代码智能化和函数参数提示

(prompting)的功能。

- 所生成的可执行代码，其运行速度要比 Visual Basic 5.0 版的编译代码大约快 3~6 倍，可达到最快的 C++ 编译器的 70%。

- 提出了一个新的技术——程序包 (Package)。通过这种“程序包”的技术，在生成 *.exe 文件时，可将其从附带的 DLL 文件中分离开来，从而生成小的客户机应用程序。

- 全面支持 ActiveX，可以创建高效的 ActiveX 构件、ActiveX 服务器和 ActiveX 文档。

- 全面支持 Internet 技术。

- 强大的数据库支持。

- 增加了 Qusoft 公司提供的 Quick Report 2.0 构件。

因此，尽快了解和掌握 Delphi 3.0 的开发步骤和方法是广大编程爱好者的共同心愿。本书全面系统地介绍了 Delphi 3.0 的开发过程、使用技巧、新的对象向导工具、数据库管理、多媒体技术、OLE 对象链接与嵌套技术等。

由于作者水平有限及写作时间仓促，书中疏漏之处敬请广大读者批评指正。

作 者

目 录



第一章

面向对象的基本概念及 Windows 编程

1.1	面向对象编程的概念 -	2
1.2	Windows 平台的主要优势 -	3
1.3	Windows 应用程序机制及其有关概念	3
1.4	Windows 环境下应用程序的主要特点	6
1.5	Windows 应用程序的基本框架结构	8
1.5.1	Windows 应用程序的基本框架结构	8
1.5.2	多任务处理	9
1.5.3	Winmain()函数的框架流程	9
1.5.4	Windows 库	9
1.6	Windows 应用程序开发环境	10
1.7	Windows 应用程序的开发规则及其过程	12
1.8	Windows 的消息循环	13
1.9	Windows 应用程序开发常用的数据类型及其相关函数 ..	15
1.9.1	MC7.0 对 Windows 编程常用的数据类型和结构	15
1.9.2	MC 7.0 对 Windows 编程的相关函数介绍	15
1.10	一个 Windows 应用程序开发实例的具体分析	17
1.11	如何创建一个 Windows 应用程序窗口	22
1.12	如何制作 Windows 应用程序的菜单	24
1.13	Windows 中的对话框	27

第二章

Delphi3.0 —— 新一代面向对象的可视化开发工具

2.1	Delphi 面向对象程序开发	32
2.1.1	Delphi3.0 全面支持面向对象的基本特征	32
2.1.2	事件驱动、事件处理及事件模拟	33
2.2	Visual 可视化编程的出现	34
2.3	各种可视化开发工具比较	34

2.3.1	Visual Basic	35
2.3.2	Visual FoxPro	35
2.3.3	Visual C++ 和 Visual Java	35
2.3.4	Delphi	38
2.4	Delphi 的基本特性	39

第三章

Delphi3.0 的工作环境及其安装过程

3.1	Delphi3.0 的工作环境	46
3.2	Delphi3.0 的安装过程	46
3.3	Delphi3.0 的启动方式	62

第四章

Delphi3.0 集成开发环境

4.1	Delphi3.0 集成开发环境 (IDE)	66
4.2	主要开发工具和窗体功能	66
4.2.1	主窗口	66
4.2.2	快速图标栏 (SpeedBar)	68
4.2.3	构件选择板 (Component Palette)	70
4.2.4	窗体 (Form)	71
4.2.5	对象观察器 (Object Inspector)	71
4.2.6	智能化的代码程序编辑器 (Coder Editor)	72
4.3	其他重要工具	73
4.3.1	工程管理器 (Project Manager)	73
4.3.2	浏览对象 (Browse Objects)	76
4.3.3	图像编辑器 (Image Editor)	77
4.3.4	数据库程序向导 (Database Form Wizard)	78
4.3.5	数据库桌面 (Database Desktop)	79
4.3.6	数据库“探险”工具——Database/Explorer	80
4.3.7	报表生成器 (Report Smith3.0)	81
4.3.8	Windows32 位探测器 WinSight32	81

第五章

Delphi3.0 应用程序开发初步

5.1	Delphi3.0 程序的基本结构	84
5.2	Delphi 应用程序开发初步	85
5.3	一个不用录入程序代码的 Delphi 应用程序	86
5.4	一个快速开发 Delphi 应用程序实例	91

第六章

Delphi3.0 语言基础及程序设计

6.1 数据存储器——变量	96
6.1.1 声明变量	96
6.1.2 为变量赋值	97
6.2 常数	98
6.3 数据类型	99
6.3.1 标准类型	100
6.3.2 数组类型	101
6.3.3 记录类型	103
6.4 运算符及其表达式	104
6.5 Delphi3.0 中的控制结构	104
6.5.1 控制结构	104
6.5.2 控制语句的应用	105
6.6 函数、过程、子程序	110
6.6.1 系统内部子程序	110
6.6.2 VCL 附带的子程序的主要功能及其附属程序	129
6.7 事件	131
6.8 Delphi3.0 与 Windows API 的使用	133

第七章

Delphi 菜单程序设计

7.1 普通菜单的制作	136
7.1.1 加入主菜单项	137
7.1.2 制作子菜单项	138
7.1.3 对菜单项作修改	139
7.1.4 加入快捷键	141
7.1.5 为菜单加入事件处理程序	142
7.1.6 执行菜单程序	145
7.2 弹出式菜单	145
7.3 标准对话框	148
7.4 使用工程专家建立菜单	150

第八章

Delphi3.0 中的主要构件及其使用

8.1 Delphi3.0 中新增加的功能及构件	160
8.2 Delphi3.0 中的构件选用区	161
8.3 主要构件功能及其使用	161
8.4 构件使用的基本技巧	174
8.5 举例	175

第九章

Delphi3.0 中多文档窗口的设计

9.1 MDI 的几个重要属性	180
9.2 MDI 多文档应用举例	181
9.2.1 创建主窗体	181
9.2.2 在窗体中加入菜单	182
9.2.3 创建“打开”文件对话框	185
9.2.4 子窗体的设计	186
9.2.5 子窗体中菜单设计	187
9.2.6 编写事件过程	189
9.2.7 将子窗体菜单作用到主窗体上	191
9.2.8 创建工具栏	191
9.2.9 创建状态栏	193
9.2.10 编写打印文本及设置打印机事件	195
9.3 利用 Delphi 中提供的工程专家建立 MDI 文档	198

第十章

Delphi3.0 的程序生成器

10.1 “New Item”新项目对话框	210
10.1.1 新建 Application(应用程序)	211
10.1.2 新建 Component(构件)	213
10.2 “Active X”页及“Forms”页	215
10.2.1 “Active X”页	215
10.2.2 “Forms”页	229
10.3 “Dialogs”页	231
10.4 “Projects”页及“Data Modules”页	234
10.4.1 “Projects”页	234
10.4.2 “Data Modules”页	236
10.5 “Business”页	236
10.5.1 使用 Tee Chart Wizard 图表向导	237
10.5.2 使用“Decision Cube Sample”决策例子	241
10.5.3 快速报表设计例子	250
10.6 Delphi 中的构件与对象的区别及其关系	252

第十一章

Delphi3.0 中的绘图功能

11.1 绘图的基本概念	256
11.1.1 笛卡尔坐标与 Delphi 画布	256
11.1.2 画布对象	257
11.2 基本绘图元素	258

11.2.1	像素点 pixels	258
11.2.2	线和点移动	259
11.2.3	矩形 Rectangle()	260
11.2.4	画线段 polyline()和 polygon	261
11.2.5	圆角矩形	261
11.2.6	椭圆 Ellipse()	261
11.2.7	在画布上输出文本	264
11.3	图形打印输出	266

第十二章

Delphi3.0 与数据库

12.1	数据库的基本概念	268
12.1.1	数据库管理技术的发展	268
12.1.2	数据库系统的基本特点	269
12.1.3	数据库系统的组成与结构	269
12.1.4	数据库及关系数据库	272
12.1.5	数据库的现状	273
12.2	Delphi3.0 数据库引擎及其数据库机制	274
12.3	有关数据库控件使用技巧	277
12.3.1	DBNavigator 控件使用	277
12.3.2	Bevel 控件使用	280
12.3.3	DBGrid 控件	280
12.4	数据库引擎管理器	281
12.4.1	增加新的数据库别名	282
12.4.2	数据库设置	282
12.4.3	System :显示有关数据库系统的信息	284
12.4.4	使用中文数据库的用户的注意事项	285
12.5	数据库桌面工具 DBD(DataBase Desktop)	285
12.5.1	DBD 的操作使用	285
12.5.2	使用 DBD 创建数据库的例子	290
12.6	数据库程序向导(Database Form Wizard)的使用	298
12.7	一个开发 Delphi 数据库程序的实例	302

第十三章

Delphi3.0 中多媒体功能设计

13.1	多媒体及其设备	310
13.1.1	多媒体及其构成	310
13.1.2	多媒体控制设备 MCI	310
13.1.3	多媒体软件开发途径	312
13.2	多媒体 MediaPlayer 构件	312

13.2.1	MediaPlayer 构件的选取及其功用	312
13.2.2	MediaPlayer 构件的主要属性	313
13.2.3	主要事件和方法	315
13.3	MCI 控制播放原理及其时间格式	316
13.3.1	播放时间位置控制	316
13.3.2	MCI 控制播放原理	317
13.4	多媒体在 Delphi 中的应用	317
13.4.1	利用图形、图像构件显示图形文件	317
13.4.2	利用 MediaPlayer 构件开发一个具有 CD 唱盘、AVI 影片、WAV 声音、MIDI 音乐等的播放器	319

第十四章

构件技术的发展及其应用

14.1	构件技术的基本概念	326
14.2	OLE、DDE、剪贴板	326
14.2.1	对象链接与嵌入 OLE	326
14.2.2	动态数据交换 DDE	327
14.3	COM(Common Object Mode)构件对象模型与 OLE2.0	329
14.4	DCOM 模型与 ActiveX	330
14.5	OLE 的应用	331
14.5.1	Delphi 中的 OLE 控件	331
14.5.2	在 Delphi 中如何使用 OLE 控件	331
14.5.3	OLE 应用举例	331
14.6	DDE 的应用	333
14.6.1	Delphi 中的 DDE 控件	333
14.6.2	一个 DDE 动态数据交换程序的具体应用	335
14.7	Delphi3.0 中构件制作的方法	338
14.7.1	构件制作的基本方法	338
14.7.2	制作构件实例	340

第十五章

创建 Delphi3.0 应用程序的联机帮助系统

15.1	联机帮助系统开发的基本方法	352
15.2	Windows 帮助系统的主要功能及其特点	353
15.3	制作帮助源文件*.RTF	355
15.3.1	帮助源文件的基本结构	355
15.3.2	专题页中的基本编码信息	356
15.3.3	RTF 文件的基本制作方法	356
15.3.4	在 Word7.0 编辑器中录入 Help 帮助源文件	358
15.3.5	Help 文件中控制代码的具体含义	361

15.3.6	创建不可滚动的标题区域	362
15.3.7	使用图形和超图形	363
15.4	创建帮助工程项目文件 HPJ	365
15.4.1	[Options]——指定选择段	366
15.4.2	[Files]——文件段	368
15.4.3	[Config]——配置段	369
15.4.4	[Map]——指明专题标识符的上下文关联号	369
15.4.5	[BITMAPS]——位图文件名列表	369
15.4.6	[Alias]——指明替换的上下文串	370
15.4.7	[Windows]——Help 帮助窗口段	370
15.5	使用帮助宏 WinHelp	373
15.6	使用 Help 生成软件开发帮助文件	376
15.7	从 Delphi3.0 应用程序中调用 Help 帮助文件	386

第一章

面向对象的基本概念及 Windows 编程

本章介绍面向对象的基本概念和 Windows 环境下的编程知识。面向对象的编程技术是目前最先进的程序设计方法之一，面向对象的程序设计方法(OOP)与编程技术不同于标准的过程化程序设计。

Windows 是一个用于 P C 机上的图形用户平台，它为应用程序提供了一个由一致的窗口和菜单结构构成的多任务环境。

- 从用户的角度看，Windows 最大的特点是从 DOS 的字符操作方式迈进了图文并茂的图形窗口操作环境。

- 从程序开发人员的角度看，Windows 则带来了全新的程序风格,如一致的用户界面、多任务、与设备无关的图形操作、高效的内存管理、事件驱动式程序设计等。

通过本章的内容，我们可以学习面向对象编程的初步知识，并通过 C 语言展示开发 Windows 应用程序的一般步骤，这为我们学习面向对象的、可视化的开发工具打下了良好的基础。从中我们可以看出使用 C 语言编写 Windows 环境下的应用程序太难了。因此，选择 Delphi3.0 实现 Windows 环境下编程是一个明智的选择。

从 1983 年 Microsoft 公司推出的 Windows1.0 到目前广泛使用的 Windows95,Windows 平台不断地被广大用户所喜爱。在 Windows 环境下开发的应用程序将越来越具有吸引力。由于它采用了统一的图形界面和菜单,因此,倍受用户的青睐。

如今 Windows 平台正风靡全球,围绕着 Windows 环境下形形色色的开发工具正不断涌现,各种面向对象的可视化软件正日趋成熟。可以说,当今 P C 机的软件就是窗口软件的时代,因此,我们开发出的产品必须具有 Windows 的特征。只有这样,软件产品才具有强大的生命力。

目前,Windows 环境下的编程方法很多。如用 Msc8.0、Borland C 5.0、Visual Basic 5.0、Visual C ++ 5.0、FoxPro5.0 for Windows、Delphi3.0 for Windows、Access for Windows 等语言。这些面向对象的开发工具都具有各自特点,要么在编程方面具有特色;要么在数据库管理方面具有优势;要么在进行多媒体创意时具有长处。

实际上这些众多的、面向对象的程序开发工具只有三种类型:一是程序设计语言;二是数据库管理系统;三是多媒体创作工具。根据需要用户可自行选择相应的开发工具,但作为软件开发的必备知识,了解和掌握 Windows 环境下程序设计的方法是最基础的,也是必要的。因此,必须首先了解 Windows 应用程序的机制,如多任务、消息传递机制、以及面向对象的编程方法等。下面让我们先看一看有关面向对象的概念以及 Windows 应用程序的开发环境和方法。

1.1 面向对象编程的概念

面向对象的程序设计方法(OOP)与编程技术不同于标准的过程化程序设计。程序开发人员进行面向对象的程序开发时,不再是单纯地从代码的第一行一直编到最后一行,而是考虑如何创建对象,利用对象来简化程序设计,提供代码的可重用性。对象可以是应用程序的一个自包含组件,一方面具有私有的功能,供自己使用;另一方面又提供公用的功能,供其他用户使用。

面向对象的编程技术是目前最先进的程序开发方法之一。它是基于消息循环、事件驱动和抽象数据类型的概念,并将系统中所有的数据、模块、窗体及其窗体中的各种元素(如按钮、文本框、菜单等),看成“对象”(object)。程序开发人员对这些“对象”进行操作设计的过程就是面向对象的编程。

对象是一个客观存在的实体,它具有广泛的含义。声音、数据、窗体、图像、图标、字体、字串、菜单都可以看成对象。以往的程序设计是面向过程的,程序的运行只能严格按程序开发人员的设计思想而执行;而面向对象的编程,彻底打破了传统的程序设计概念,引入了面向对象的新的程序设计风格,从而极大地扩展了编程的范畴。

Windows 为我们提供了一个面向对象(如窗口、菜单、对话框、控制窗口等为对象)的开发环境。开发 Windows 的应用程序,实际上就是要对这些对象进行操作。

1.2 Windows 平台的主要优势

Windows 平台是一个用于 PC 机上的图形用户界面，它为应用程序提供了一个由一致的窗口和菜单结构构成的多任务环境，如图 1-1 所示。

从图 1-1 中可以看出，Windows 的窗口环境是基于图形的。从用户的角度看，Windows 最大的特点是从 DOS 的字符操作方式迈进到图文并茂的图形窗口操作环境；而从程序开发人员的角度看，Windows 则带来了全新的程序风格，如一致的用户界面、多任务、与设备无关的图形操作、高效的内存管理、事件驱动式程序设计等。

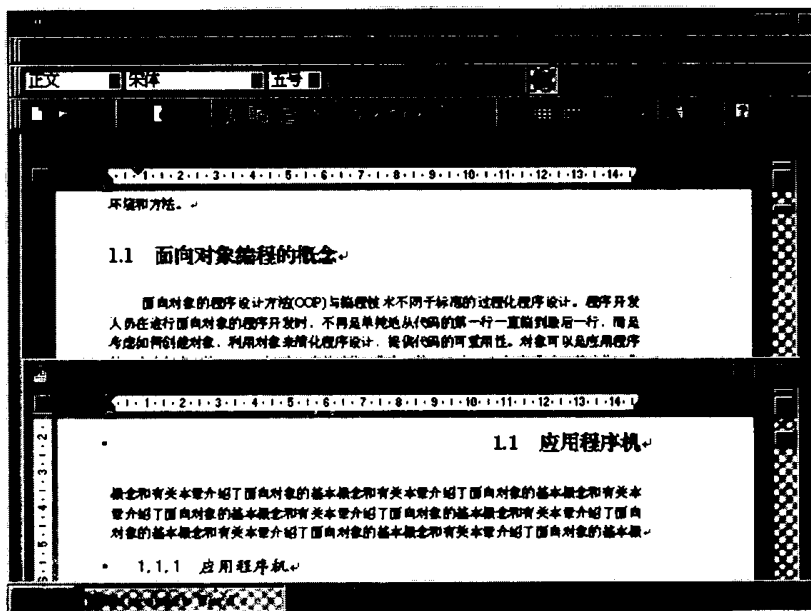


图 1-1 Windows 环境下的窗口及菜单

Windows 的主要设计目标之一就是能够同时访问多个应用程序，共享系统资源和多任务运行环境。它不像 DOS 环境那样，独立占用计算机系统的存储和输入输出设备，而是所有应用程序都能够共享计算机的资源，这样对程序开发人员来说，编程就困难些了，幸好 Windows 为用户提供了一个高效的软件开发工具包 SDK，它为在 Windows 环境下开发出具有窗口特点和功能的应用软件，提供了各类工具、资源、函数库和数据结构，从而减轻了程序开发人员的负担，使得编程更加容易、高效。

1.3 Windows 应用程序机制及其有关概念

在了解 Windows 环境下应用程序编程模式之前，让我们先对与 Windows 环境下编程有关的基本概念作一些介绍。这些概念对于刚刚涉及到 Windows 编程的人员来说是必须要理解和掌握的。只有对这些基本概念有了深刻的理解，才能对 Windows 环境下应用程序的编程模式有所帮助。

一、对象

对象是客观存在的一个实体。它具有广泛的含义，声音、图像、数据、一段文字都可以称为对象。一段应用程序也可以称之为对象，应用程序中的窗口、菜单、对话框、图像、图标等，都可统称为对象。

二、面向对象的编程

在应用程序中,对对象的某种操作可以用一段具体的代码来实现，面向对象的编程打破了传统的编程模式，大大地扩展了编程的范围。

三、句柄

句柄用户识别对象而为对象赋予的唯一名称。给一个对象赋予句柄后，对该对象的调用就可以通过句柄来实现了。

四、事件驱动

Windows 环境不像 DOS 那样，具有中断的概念,它是采用事件驱动程序机制。传统的程序设计多数是采用过程方式驱动；而 Windows 是基于事件驱动，它不需要由事先编好的顺序来控制，而由事件的发生来控制。Windows 的这种事件驱动的机制，特别适合于应用程序与用户的交互。

五、消息

通过某种方式传递给应用程序的信息。消息和 DOS 应用程序的输入有一定的相似之处，它们的作用都在于引发应用程序做某种动作。Windows 下消息所包含的信息要比 DOS 下的输入多得多。

前面我们谈过 Windows 是采用事件驱动程序设计的，而事件驱动程序的设计是紧紧围绕着消息的产生与处理而展开的。Windows 程序设计的主要工作，就是对应用程序所要发出或要接收的消息进行排序管理，它可以同时为多个程序接受和分配输入消息，它为每个运行着的程序维持一个消息队列，同时监控着所有的输入设备。通常情况下，Windows 维持着两个消息队列，即系统消息队列和应用程序消息队列。它将输入的消息首先放入系统消息队列中，然后再将系统队列中的输入消息分配到应用程序相应的消息队列中。

六、实例

在 Windows 中不仅可以同时运行多个应用程序，而且可以运行一个应用程序的多个副本，其中一个副本就称为一个实例,如图 1-2 所示。

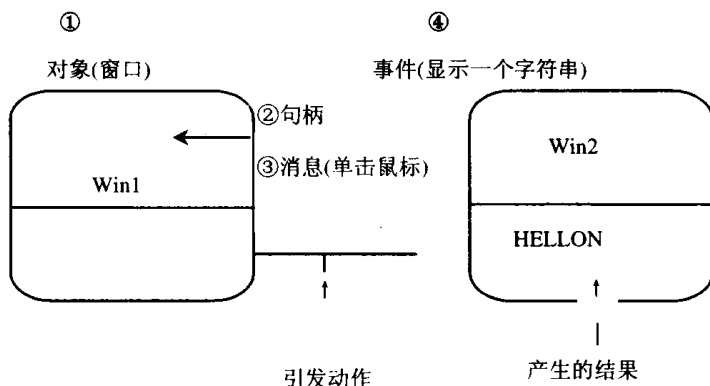


图 1-2 实例

七、窗口函数

窗口函数是接收和处理消息的函数，在 Windows 的程序设计中，消息循环负责从消息队列中检索消息并将消息发送给相应的窗口函数，由窗口函数决定一个消息被接受时所采用的处理过程。实际上窗口函数就是消息处理的过程。下面是一个消息处理过程 WndProc 例子：

```
long FAR PASCAL WndProc(hWnd,message,wParam,lParam)
HWND hWnd;  窗口句柄
unsigned message;  消息类型
WORD wParam;  附加信息
LONG lParam;  附加信息

    PAINTSTRUCT ps;
    switch(message);
    case WM_DESTROY:
        psotQuitMessage(0);
        break;
    case WM_PAINT:
        BeginPaint(hWnd,(LPPAINTSTRUCT)&ps);
        EditPaint(ps.hdc);
        EndPaint(hWnd,(LPPAINTSTRUCT)&ps);
        break;
    .....

    default:
        return DEFWindowsProc(hWnd,message,wParam,lParam);
        break;
```

八、资源及资源文件

资源是存在于 Windows 应用程序的 EXE 文件或 DLL 文件中的只读数据。在 Windows 的应用程序中使用了儿种不同类型的资源,这些资源包括光标、菜单、字符串、位图、对话框、图标、快捷键等。一个资源创建后存放在资源文件中,下面分别加以说明:

字符串 是一个应用程序用于菜单项名、对话框名和错误信息提示等场合的正文字符串,下面定义一个字符串表的例子:

```
STRINGTABLE
BEGIN
    IDSTR_NAME,"EDIE"
    IDSTR_TITLE,"program Editor"
```

快捷键 是用于选择一个菜单项或引用一个命令的键或键的组合,通常称为热键。使用快捷键是为了快速访问一些最常用的操作,可以代替鼠标的操作。以下是一些快捷键的定义:


```

the Accel Table   Accele Rators
BEGIN
    VK_BACK, MN_UNDO, VIRTKEY,SHIFT
    VK_DELETE,MN_CUT,  VIRTKEY,SHIFT
    "^F",      MN_FIND
    "^c",      MN_COPY
END

```

光标 是一个用于指定鼠标运行的 32×32 像素的特殊位图。它的主要作用是将当前状态告知用户，以说明当前的运行方式、接受的命令、错误警告、跟踪一个操作的进程等。因此 Windows 中有几个预定义的光标类型供用户调用。

位图 是表示一个图像图标的二进制数据，使用它能很快地将预定义的物体画到 CRT 上。这些位又叫像素。Windows 中有两种类型的位图，其一是设备相关位图(DDB)：它与内存中的位图和 CRT 上的像素有紧密的相关性；其二是设备无关位图(DIB)：它只是描述图像的“外貌”，而不是指定位和像素之间的一一对应关系。这种外观上定义使得应用程序能够在任何输出设备上显示 DIB。Windows 把图像经过转换，再输出到任一输出设备上，就不要求这某一特定的设备编写独特的代码。

对话框 是一个弹出窗口，它是应用程序与用户交互的界面。在 Windows 中有下列几种类型的对话框：固定的和可移动的对话框、模式、无模式和半模式对话框等。

1.4 Windows 环境下应用程序的主要特点

一、直观的用户图形界面

Windows 环境下应用程序具有直观、友好的用户界面，它是由图形接口来实现的，图形用户接口主要由窗口、菜单、会话框等组成。

二、队列式输入

Windows 环境下应用程序将输入放入队列中，当 Windows 应用程序准备好接收输入时，则从队列中取出这一输入。

三、与设备无关的图形输出

Windows 为用户提供了丰富的、与设备类型无关的图形，它是通过以下方式实现与输出类型无关的，见图 1-3。

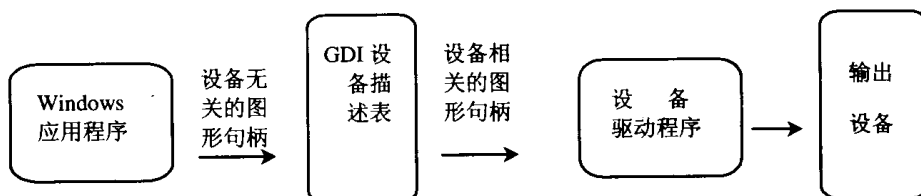


图 1-3 与设备无关的图形输出