

万水 Delphi 技术丛书

曹洋 等编著

Delphi 7

经典问题 解析



中国水利水电出版社
www.waterpub.com.cn

万水 Delphi 技术丛书

Delphi 7 经典问题解析

曹洋 等编著

中国水利水电出版社

内 容 提 要

本书共分为 9 章, 内容包括: Delphi 开发语言与 Windows 程序设计开发、Object Pascal 的相关问题及其解决方案、操作系统编程的相关问题及其解决方案、文件系统编程的相关问题及其解决方案、Win32 API 编程的相关问题及其解决方案、Delphi 中的字符串处理问题及其解决方案、注册表与 INI 文件编程问题及其解决方案、Delphi 下的多媒体编程问题及其解决方案、Delphi 网络编程问题及其解决方案。本书内容翔实, 衔接紧密。通读本书的读者会发现, 任何两章都是紧密衔接的, 每一章都会或多或少地用到前面章节的解决方案。本书的宗旨就是要为中高级用户补上其他参考书中缺少的内容, 因此有的领域没有赘述。这些内容都是 Delphi 经典问题中的经典问题, 希望读者能够认真研读本书。相信通过本书的学习, 您的编程水平一定会有质的飞跃。

图书在版编目 (CIP) 数据

Delphi 7 经典问题解析/曹洋等编著. —北京: 中国水利水电出版社, 2003

(万水 Delphi 技术丛书)

ISBN 7-5084-1599-X

I. D… II. 曹… III. 软件工具—程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字 (2003) 第 055416 号

书 名	Delphi 7 经典问题解析
作 者	曹洋 等编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1000 毫米 16 开本 20.25 印张 446 千字
版 次	2003 年 8 月第一版 2003 年 8 月北京第一次印刷
印 数	0001—5000 册
定 价	32.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

Delphi 软件开发工具是 Borland 公司推出的一个完全导向的可视化系统开发环境的工具。它具有功能强大、运行速度快、易于学习和使用以及开发迅速等特点，所以一经推出就受到广大用户的喜爱。Delphi 结合了可视化技术、面向对象技术、数据库技术以及网络技术等多种先进的软件编程技术和思想，并使用了世界上最快的编译器，使其成为创建功能丰富、界面友好的 Windows 应用软件的工具之一。

Delphi 可以使你用最少的程序量来得到最高的开发效率，从而快速地开发一个 Windows 的应用程序。Delphi 提供了易于使用的图形化工具供软件开发人员创建程序中的可视化部分。开发者可以按照自己的喜好及用户要求，随心所欲地安排窗口的外观，这相对于基于 C 或者 C++ 语言的开发环境而言，软件界面设计开发周期将会大大缩短。

Delphi 所采用的底层编程语言是 Borland (现为 Inprise) 公司的 Object Pascal。Object Pascal 语言是一种相当简练的编程语言，其语法严格、结构清晰、可读性好。Delphi 内嵌的 Object Pascal 不是传统意义上的 Pascal，但它保留了传统 Pascal 语言的精髓并增加了面向对象的概念，从而使其语言的风格更适合于可视化的面向对象程序设计。

Delphi 是以面向对象的方式设计程序代码的。在 Delphi 中，一切都被视为对象。编写面向对象的程序能够重复使用开发者自己的工作成果，并使程序维护起来更为容易，同时程序的扩展也变得更加方便。

面向对象技术的四大特性，即封装性、可维护性、继承性和多态性，都在 Delphi 中得到了全面和充分的体现。① Delphi 利用控件封装了大部分的 Windows API 函数，从而使得程序员可以简单地通过对象的属性和方法来安全调用这些 API 函数，避免了直接调用的复杂和烦琐；② 由于 Delphi 利用控件封装了大量的 API 函数，基于 Windows 操作系统向上兼容的特点，从而使得应用程序获得了更好的可移植性；③ Delphi 中的继承性可使设计人员在已有对象类型（父类）的基础上创建新的对象（子类），这个新对象将继承其先前对象的所有事件、属性及方法，并且可以派生出属于自己的新的过程、属性或者方法；④ Delphi 中的多态性是指允许将父对象设置成为和一个或更多的其子对象相等的技术，赋值之后，父对象就可以根据当前赋值给它的子对象的特性以不同的方式工作。简单地说，就是允许将子类类型的指针赋值给父类类型的指针。多态性在 Object Pascal 和 C++ 中都是通过虚函数（Virtual Function）实现的。

封装可以隐藏实现细节，使得代码模块化；继承可以扩展已存在的代码模块（类），二者的目的都是为了代码重用（Code Reuse）。而多态则是为了实现另一个目的，即接口重用（Interface Reuse）。而现实往往是，要有效重用代码很难，而真正最具有价值的重用是接口重

用，因为“接口是最有价值的资源。设计接口比用一堆类来实现这个接口更要花费时间，而且耗费的是更昂贵的人力的时间。”（摘自《设计模式》可复用面向对象软件设计的基础。）

应用多态的例子非常普遍，在 Delphi 的 VCL 类库中，最典型的就是：TObject 类有一个虚拟的 Destroy 虚构函数和一个非虚拟的 Free 函数。Free 函数是调用 Destroy 函数的。因此，当对任何对象（都是 TObject 的子类对象）调用 .Free() 之后，都会执行 TObject.Free() 方法，它会调用所用对象的析构函数 Destroy()。这就保证了任何类型的对象都可以正确地析构。

Delphi 还对目前流行的各种技术，例如：Internet、OLE（对象链接嵌入）、DDE（动态数据交换）、COM（组件对象模型）等提供了强有力的支持，它们同样通过控件的形式供程序员使用，这使得程序员可以避开这些技术复杂的底层实现，而直接利用其高层应用部分。

Delphi 开发工具从诞生以来，就一直受到广大编程爱好者和程序的青睐。从 16 位 Delphi 1.0 版本逐渐升级完善，直到今天的 Delphi 7.0 版本，Borland（Inprise）公司不断在用新技术、新机制吸引着世界各地的程序员。可以预见，即将推出的 Delphi.Net 版本会拥有广阔的前景和美好的未来，它将使得越来越多的程序员和编程爱好者加入到 Delphi 开发阵营。

本书内容

本书共分为 9 章，具体介绍如下：

第 1 章：Delphi 开发语言与 Windows 程序设计开发。本章主要介绍 Delphi 开发工具的历史和 Windows 下程序设计的过程，这一章是全书的基础，非常重要。因为只有对 Windows 程序开发和 Windows 的机制有了一定的了解，并通晓 Delphi 各个版本的异同，才能有的放矢地学习本书后面的章节。

第 2 章：Object Pascal 的相关问题及其解决方案。本章主要介绍 Delphi 的核心动力之源——Object Pascal 语言的一些疑难问题与解决方案，侧重于讲解 Object Pascal 中的精髓之处和容易混淆的概念。

第 3 章：操作系统编程的相关问题及其解决方案。本章主要介绍 Delphi 在针对操作系统编程过程中读者经常碰到的一些疑难问题，并介绍了 Windows 的钩子（Hook）和消息机制等内容。

第 4 章：文件系统编程的相关问题及其解决方案。本章主要介绍 Delphi 在针对文件系统对象编程的过程中读者感到困惑和难以处理的问题，从传统文件处理方式开始，然后介绍流对象，并引导读者探索了很多 Windows 的内部机制。

第 5 章：Win32 API 编程的相关问题及其解决方案。本章主要介绍利用一些鲜为人知的 Windows API 函数来实现特殊的程序特技，并结合当前很多软件中读者比较感兴趣的功能和效果，剖析了其实现机理，并给出了相应的解决方案。

第 6 章：Delphi 中的字符串处理问题及其解决方案。字符串处理问题是任何编程语言都大量涉及的问题。Delphi 对字符串的处理做得比较完善，它提供了相当丰富的处理函数和过程，但读者对这些函数和过程的使用场合和技巧了解得并不多。作者通过剖析日常使用的几

个软件中的特定功能，让读者充分领会如何巧妙地处理字符串。本章最后还介绍了编程中经常使用的 TStringList 和 TStrings。

第 7 章：注册表与 INI 文件编程问题及其解决方案。注册表和 INI 文件可以作为存取数据的载体，它们对于读者来说并不陌生。但如何巧妙有效地利用它们来更好地实现程序特效，则恐怕很难有人说得清楚。本章将通过几个具体的应用帮助读者掌握这两类对象处理方法的同时，开阔处理问题的思路。

第 8 章：Delphi 下的多媒体编程问题及其解决方案。多媒体编程在 Delphi 中占有很大的比例。Delphi 为我们提供了大量的多媒体控件来实现多媒体操纵，但它远远不能满足中高级用户控制多媒体的需要。本章将抛弃这些传统的控件，利用一些 Windows Multimedia API 函数来实现这些需求，其中分为两个部分：图形图像和音频视频。

第 9 章：Delphi 网络编程问题及其解决方案。Delphi 在为我们提供大量的网络编程控件的同时，其各个版本附带的 Demo(演示)程序也详尽细致地对这些控件的使用做了深入讲解。而本书面向的读者对象主要是中高级用户，因此本章的重点将放在如何在不利用这些标准控件的情况下更高效地解决某些非常规问题。本章循序渐进，带领读者从简单的问题入手，不断发现和提出新的问题，然后动手解决问题。相信读者一定会在这里对困惑已久的网络编程问题找到合适的解决方案。

本书虽然章节不多，但内容翔实，衔接紧密，通读本书的读者会发现，任何两章的内容都是紧密衔接的，每一章都会或多或少地用到前面章节的解决方案。至于 Delphi 编程的其他方面的介绍，别的参考书介绍得已经很多，很详细了。本书的宗旨就是要为中高级用户补上其他参考书中都缺少的内容，因此有的领域没有赘述。这些内容都是 Delphi 经典问题中的经典问题，希望读者能够认真研读本书，相信通过本书的学习，您的编程水平一定会有质的飞跃。

本书特点

本书是一本介绍 Delphi 高级应用和经典疑难问题解决方案的指导性书籍，本书适用于对 Delphi 开发比较熟悉的中高级用户，全书内容详实，例程丰富，汇集了作者多年的实际开发经验和研究心得，是一本真正面向高级应用的 Delphi 力作，可为 Delphi 程序员挑战发展空间提供帮助。

本书所介绍的内容涉及到很多 Windows 操作系统底层的机理和内幕，在指导读者解决具体的编程疑难问题的同时，让读者进一步深入学习探讨相关的问题，帮助读者逐渐形成发现问题、分析问题、独立设计解决方案、编码实现直至解决问题的编程和思考习惯。

目前介绍 Delphi 开发的参考书越出越快、越出越厚、越出越多，但真正有新内容、有新意、有价值的却越来越少，面向中高级用户的则更是凤毛麟角。本书主要介绍的内容是目前出版的 Delphi 类参考书中几乎都没有涉及到的或者介绍相对不全面（甚至错误）的内容。其中很多都是作者自己潜心研究的成果，现在拿出来介绍给广大的编程爱好者，希望能够帮助

大家尽快提高编程水平。

相信读者通过学习本书，一定会深刻理解程序设计的艺术之美。当前我们迫切需要的不仅仅是某个问题的处理方法，更重要的是掌握对新发现问题的独立思考，深入剖析，设计解决方案的能力。这也是本书全篇编写的宗旨所在，本书将通过启发式的教学方式，一方面帮助读者摒弃以往不好的编程习惯和风格，另一方面培养读者敏锐的观察力和想象力。

参加本书编写工作的除封面署名外，还有童剑、张勇、张红中、马健兵、余成武、朱亮、张雷、孙莉、王军平、李毅鹏、肖兵亮、孙仁毅、朱健涛、钱辉、马聪、郭润鸿、郭文斌、魏国梁、黄卓、张雨思、王平、高建中等。

由于时间仓促和编写水平有限，书中的不妥或错误之处在所难免，恳请广大读者批评指正。

编者

2003年6月

目 录

前言

第 1 章 Delphi 开发语言与 Windows 程序设计开发	1
1.1 Windows 下的程序设计与开发	1
1.1.1 Windows 操作系统概述	1
1.1.2 Windows 程序设计与开发工具	2
1.1.3 Delphi 开发工具的发展历史回顾	5
1.2 本章小结	7
第 2 章 Object Pascal 的相关问题及其解决方案	8
2.1 Object Pascal 本身的深入剖析与探讨	8
2.1.1 Object Pascal 数据类型的辨析和说明	8
2.1.2 Object Pascal 中的类型转换问题探讨	17
2.1.3 Object Pascal 中 With 语句的使用问题	19
2.1.4 Object Pascal 中例程传递参数问题的探讨	22
2.1.5 Object Pascal 中的动态数组	23
2.1.6 Object Pascal 中的 Forward 声明释疑	27
2.1.7 Object Pascal 中的高级功能——过程类型	29
2.1.8 Object Pascal 中的默认参数问题与函数重载	33
2.1.9 Object Pascal 中的 Handle (句柄) 数据类型与 Windows 编程	35
2.1.10 类定义中 Public、Private、Protected、Published 与 Automated 辨析	39
2.1.11 Object Pascal 中的回调函数问题解析	44
2.1.12 Object Pascal 中的 Variant 数据类型与 OLE 探秘	48
2.1.13 如何不使用 VCL 创建 Windows 应用程序	54
2.2 本章小结	58
第 3 章 操作系统编程的相关问题及其解决方案	59
3.1 Delphi 中的操作系统编程概述	59
3.1.1 如何编程实现检测 Windows 操作系统的运行模式	59
3.1.2 如何编程检测 Windows 操作系统类型	60
3.1.3 如何编程实现系统关机/重新启动/注销/待机等操作	62
3.1.4 如何编程实现禁止系统功能热键	68
3.1.5 如何编程启动/关闭外部的应用程序以实现 Delphi 程序调用其他程序	78

3.1.6	如何编程实现 Windows 任务栏和桌面图标的显示/隐藏	87
3.1.7	如何编程获取系统相关信息	89
3.1.8	如何编程将文件删除到回收站及实现清空回收站操作	105
3.1.9	如何编程实现让程序只运行一个实例而防止程序被重复执行	112
3.1.10	如何编程启动控制面板程序及各个设置功能	114
3.1.11	如何编程实现应用程序任务栏图标的显示/隐藏	116
3.1.12	如何在任务栏中添加图标及菜单响应	117
3.2	本章小结	126
第 4 章	文件系统编程的相关问题及其解决方案	127
4.1	Delphi 中的文件系统编程概述	127
4.1.1	Delphi 中的传统文件处理方式与解决方案	127
4.1.2	Delphi 中的流处理方式与文件系统问题的解决	141
4.1.3	TFileStream 流对象与文件处理问题	144
4.1.4	如何在任意目录下查找指定类型或者名称的文件	146
4.1.5	如何将一个目录下的所有文件 Copy 到一个新目录下	154
4.1.6	有关目录创建问题的解决方案	158
4.2	本章小结	164
第 5 章	Win32 API 编程的相关问题及其解决方案	165
5.1	Delphi 下的 Win32 API 编程概述	165
5.1.1	如何实现当鼠标离开窗口的时候窗体自动隐藏	166
5.1.2	如何在抓取屏幕的时候连同鼠标一起抓获	175
5.1.3	如何利用 Windows API 实现应用程序标题栏的闪动	182
5.1.4	如何使你的窗口以特技的形式显示给使用者	183
5.1.5	如何编程实现程序间的数据传递和共享	184
5.2	本章小结	192
第 6 章	字符串处理问题及其解决方案	194
6.1	Delphi 对字符串操作的支持概述	194
6.1.1	如何分解固定格式的字符串 (Foxmail 和 FlashGet 中信息自动填充的实现原理)	196
6.1.2	如何获得字符串中汉字和英文字母及其他字符的个数	205
6.1.3	如何判断 E-mail 地址的合法性	208
6.1.4	字符串里包含特殊符号怎样处理	210
6.1.5	如何对字符串进行简单加密	211
6.1.6	如何使用字符串列表 TStringList 和 TStringList	214
6.2	本章小结	216

第 7 章 注册表与 INI 文件编程问题及其解决方案	217
7.1 注册表及 INI 文件编程概述	217
7.1.1 注册表及 INI 文件常用操作函数介绍	220
7.1.2 如何编程让某个程序在系统启动时自动加载	227
7.1.3 如何编程实现在 IE 标准按钮栏加入自己的图标和响应事件	230
7.1.4 如何编程向注册表中写入二进制和 DWORD 类型值	233
7.1.5 如何编程修改注册表键下的“(默认)”项值	236
7.1.6 如何编程获取注册表某个键下面的所有项和对应的项值	237
7.1.7 INI 文件编程举例(利用 INI 文件保存窗体的设置)	239
7.2 本章小结	244
第 8 章 多媒体编程问题及其解决方案	245
8.1 Delphi 下的多媒体编程概述	245
8.1.1 如何编程实现 Bmp 位图文件与矢量图形文件 (wmf 或 emf 格式)的转换	246
8.1.2 如何编程实现 BMP 格式和 JPEG 格式相互转换	251
8.1.3 如何编程实现将一幅图像转换成灰度图 (类似于 Word 中的图片处理工具栏中的“灰度”)	254
8.1.4 如何编程实现将文本或图像旋转特定的角度	257
8.1.5 如何将资源文件添加到 EXE 程序中并由程序调用	260
8.1.6 如何检测声卡状态及编程控制声卡音量	266
8.1.7 CD-ROM 相关问题及其解决方案	270
8.1.8 显示屏幕分辨率的相关问题及其解决方案	276
8.2 本章小结	276
第 9 章 网络编程问题及其解决方案	278
9.1 Delphi 下的网络编程概述	278
9.1.1 如何编程获取/修改本机的局域网 IP 地址	280
9.1.2 如何编程实现 IP 地址与主机名称(包括域名情况)互换	285
9.1.3 如何编程获得局域网所有工作组的名称	288
9.1.4 如何编程获得局域网工作组中的所有主机名称	291
9.1.5 如何编程获取局域网中指定主机的共享资源列表	293
9.1.6 如何编程检测计算机是否接入 Internet 及当前接入方式	295
9.1.7 如何编程实现设置/取消文件夹的共享	297
9.1.8 如何编程切断当前所有的网络共享连接	303
9.1.9 如何编程检测本机同远程主机之间的连接状态	306
9.2 本章小结	314

第 1 章 Delphi 开发语言与 Windows 程序设计开发

本章主要介绍 Delphi 程序设计语言的历史与 Windows 程序设计开发的一些基础理论，这些知识将是您掌握 Delphi 并利用它实现高效开发的坚实基础。

1.1 Windows 下的程序设计与开发

1.1.1 Windows 操作系统概述

操作系统是管理计算机软硬件资源的一个平台，没有它，任何计算机都无法正常运行。在个人电脑发展史上，出现过许多不同的操作系统，其中最常用的有 5 种：DOS、Windows、Linux、Unix/Xenix、OS/2。这些操作系统都各有自己的特色和特定的应用场合，其中最常见和最常使用的是 Windows 操作系统。

Windows 操作系统是美国 Microsoft 公司在 1985 年 11 月发布的第一代窗口式多任务系统，它使 PC 机开始进入了所谓的图形用户界面（GUI，Graphic User Interface）时代。这种界面方式为用户提供了很大的方便，它带来了一种全新的概念并且把计算机操作系统的易用性提高到了一个新的阶段。

在相继研发了 Windows 操作系统 1.X，2.X 的基础上，Microsoft 公司于 1990 年又推出了 Windows 操作系统 3.0 版。相对于先前的版本而言，它的功能进一步加强，内存管理方面的改进尤为明显，且提供了数量相当多的 Windows 应用软件，因此成为 386、486 微机新的操作系统标准。随后，Windows 3.1 版很快发布，而且针对中国市场推出了相应的中文版本。Windows 3.1 版较之 Windows 3.0 版的功能更为强大，它占领了微机操作系统市场绝大多数的份额，成为当时最流行的操作系统。

1995 年，Microsoft 公司推出了 Windows 95 操作系统。在此之前的 Windows 操作系统都是由 DOS 引导的，而 Windows 95 是一个完全独立的 32 位操作系统，它在很多方面作了进一步的改进，还集成了网络功能和即插即用（Plug and Play）功能。这个版本的操作系统性能更加稳定，易用性更强。微软开发操作系统的脚步并没有因为 Windows 95 的推出而停止。Microsoft 公司于 1998 年又推出了 Windows 95 的改进版 Windows 98。这个版本的操作系统整合了浏览器技术，实现了与浏览器的无缝链接，操作系统附带了微软开发的 IE（Internet Explorer）软件，使用户访问 Internet 资源就像访问本地硬盘一样方便，从而更好地满足了人们越来越多的访问 Internet 资源的需要。集众多优点于一身的 Windows 98 操作系统可以称得上是 Microsoft 公司个人用户（家用）操作系统领域发展的一个里程碑，它的诞生具有划时代

的意义。

商用领域,主要的操作系统有 Windows NT, Windows 2000 等,这些操作系统与个人用户操作系统相比有更好的稳定性和健壮性,一般都作为服务器的主流操作系统。

近期 Microsoft 公司又推出了 Windows XP 操作系统,这个操作系统的图形用户界面更加美观和易于操作,它还增加了很多对当今流行的标准的支持,例如对 IEEE 1394 的支持等,使得其正逐渐取代 Windows 98 而成为个人 PC 机的首选操作系统。

1.1.2 Windows 程序设计与开发工具

1.1.2.1 Windows 程序设计的过程概述

Windows 操作系统的大行其道,使得众多的软件开发商把大部分的精力都集中在 Windows 平台上,但是 Windows 操作系统本身也并非面面俱到,它同样需要第三方开发的程序来完善和充实自己从而满足用户不断提出的需求。

开发 Windows 应用程序的过程主要包括如下 5 个方面:①需求分析→②结构流程设计→③编码实现→④调试→⑤发布。其中①和②两个环节尤其重要,它关系到最终开发出来的软件产品的功能和体系结构,以及将来维护升级的余地。需求分析做得不全面会造成软件的功能不健全,这将直接导致软件没有销路;体系结构设计的不合理会造成后续版本的升级或开发都要全部重新做,从而将浪费很大的人力和物力。设计的重用性不高对整个软件而言将是致命的。做好了前面两个方面,后面的工作相对而言就简单多了。采用某一种开发语言实现设计所要体现的思想就是编码实现的过程,而后针对特定的数据或者环境进行模拟演示称作调试。编码和调试这两个环节循环往复进行,直到准确无误地实现了设计的全部内涵为止。最后一个环节是软件的发布,因为错误总是难免的,所以在最终发布 Final 版本之前,软件厂商会提供给客户一些测试版本来进行最后的排错,测试版本一般分为 α 和 β 两种类型。厂商会在一定的时间期限内收集客户的反馈来改进软件的不足,从而充分地保证最终版本的健壮性。

1.1.2.2 Windows 程序设计的主流开发工具简介

随着计算机技术的不断发展,各种编程工具也随之发展,这就使得当今的大多数程序开发人员可以摆脱使用枯燥无味的计算机指令或汇编语言开发软件,而是利用一系列高效的、具有良好可视化功能的编程工具去开发各种软件,从而达到事半功倍的效果。下面我们就来介绍一下当今 Windows 程序设计的几种主流开发工具。

1. Visual Basic

Visual Basic 是 Microsoft 公司推出的以 Basic 语言为基本语言的一种可视化编程工具。它作为一种较早出现的开发程序,以其容易学习,开发效率较高,具有完善的帮助系统等优点影响了好几代编程人员。Visual Basic 作为一种可视化的开发工具存在很大的局限性,主要体现在:① Visual Basic 不具备跨平台特性;② 对组件技术的支持基于 COM 和 ActiveX,在组件技术不断完善发展的今天,这种落后性是很明显的;③ Visual Basic 在进行系统底层开发的

时候相对复杂,不能嵌套汇编;④面向对象的特性差,网络功能和数据库功能也没有非常特殊的表现。这些局限性导致了它在未来软件开发中必定逐步被其他工具替代。

2. PowerBuilder

PowerBuilder 是 SyBase 公司的产品,它是开发 MIS (Management Information System, 管理信息系统) 系统和数据库相关软件产品的首选开发工具,具有使用简单,容易学习,容易掌握的特点。它在代码执行效率上也有相当出色的表现。它对数据库的支持相当好,可以直接在语句中嵌套 SQL 语句,其特有的 Data Window (数据窗口) 技术使得数据的操纵更加直观和随心所欲。但是它对系统底层和网络方面的开发支持不是很好,面向对象的特性也不太理想。所以相对而言,它仍然不是我们心目中最理想的开发工具。

3. Visual C++

Visual C++ 是 Microsoft 公司推出的以 C++ 为开发语言的基于 MFC 库的可视化开发工具,从总体上说,它是一个功能强大但是不便使用的开发工具。它在网络开发和多媒体开发等方面都有不俗的表现,帮助系统也做得相对完善。更值得一提的是,对于要求时空效率非常高的底层开发领域则更是非它莫属。但 C++ 语言的繁杂导致了其难以被初学者短时期内快速掌握和熟练应用,这在很大程度上限制了这种开发工具的效率。所以虽然使用这种工具开发出的代码效率要相对高一些,但是开发周期的冗长是其无法掩盖的“痼疾”,从总体角度来看,其同样不能称之为一种“高效”的开发工具。

4. Java 语言

1995 年 SUN 公司推出 Java 语言之后,全世界的目光都被这种神奇的语言所吸引。在 Java 出现以前,Internet 上的信息都是一些乏味死板的 HTML 文档,因此人们迫切希望能在 Web 中看到一些交互式的内容,开发人员也很希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序,同时这些程序还应有极大的安全保障。对于用户的这种要求,传统的编程语言显得无能为力,而 Java 的出现恰好弥补了这个缺陷。

Java 是一种简单的面向对象语言,其特点是:分布式,多线程动态,健壮安全,结构中立,可移植性强。Java 语言本身的这些特点使得其在网络开发应用方面具有高人一等的表现,而且其面向对象性高,支持的组件技术多,平台无关的特性也决定了它在现在和未来的开发中将会占据越来越重要的地位。但是 Java 在系统底层开发和多媒体开发方面却表现得并不让人非常满意,其解释执行的机制也使得其效率不是很高。

1.1.2.3 Delphi —— 近乎完美的开发工具

以上介绍的 4 种开发语言都有各自的优点,但在某些方面存在着致命的弱点,这就决定了它们不能成为一种高效的 RAD (Rapid Application Development) 工具。在这种情况下, Borland 公司审时度势,顺应潮流地推出了 Delphi 开发工具,从而引发了程序开发领域的重大变革,越来越多的程序员选择了 Delphi 阵营。也许读者经常会有这样的疑问:“到底什么使得 Delphi 如此优秀”、“与别的编程工具相比, Delphi 究竟优秀在哪里”等。下面我们就来看看这种开发语言的魔力所在。

首先, Delphi 是全新的可视化编程环境, 它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想, 采用了弹性可重复利用的完整的面向对象程序语言 (Object-Oriented Language)、当今世界上最快的编辑器、最为领先的数据库技术。与其他开发工具相比较而言, Delphi 的优势主要体现在如下几个方面:

1. 开发的高效

Delphi 是一个 RAD (Rapid Application Development, 快速开发工具) 工具, 它具有可视化的开发环境, 是真正面向对象的。它基于面向对象 (Object Oriented) 技术构建的 VCL 库中的所有组件都可以被继承以创建新的组件, 包括窗体类 TForm; Delphi 的 Code Insight 技术 (即代码自动完成功能) 省去了许多人工输入的麻烦, 而且这一技术建立在编译器信息之上, 而不是基于像 Visual Basic 等使用的类型库机制, 因此应用范围更加广泛。

对 RAD 工具 (如 Delphi、Visual Basic、C++ Builder 和 PowerBuilder 等) 来说, 窗体设计器是一项独特的功能。一些更为经典的开发环境, 如 Visual C++ 和 Borland C++ 等, 虽然都提供了对话编辑器, 但却没有将窗体设计器集成到开发流程中, 这必然导致开发效率的降低。

2. 语言的高效

Delphi 内嵌的程序开发语言是 Object Pascal (面向对象的 Pascal), 这是一种真正支持面向对象而又优雅美观、易于掌握的语言, 其在功能上毫不逊色于任何其它的面向对象开发语言。并且非常值得推崇的是, Object Pascal 在使得开发者运用各种模式 (Pattern) 进行的系统结构设计都能被完善支持的情况下, 没有过多地增加实现过程的复杂度, 这使得模式在实施时也不用考虑太多语言 / 编译器细节, 从而大大减轻了设计人员的负担。

3. 编译/链接的高效

1983 年 Borland 公司推出了世界闻名的 Pascal 编译器——Turbo Pascal, 实现了詹森和沃斯 (Jensen & Wirth) 在 “Pascal User Manual and Report” 中提出的思想。由于具有简洁和功能强的特点, Turbo Pascal 成为当时最畅销的编译器之一。Pascal 编译器最著名的特点就是速度快, 而 Delphi 正是建立在这种编译器的基础之上。事实上, 它可能是到目前为止, 最快的高级语言本地代码编译器。快速的编译器可以让你频繁地在修改代码和编译运行的状态间切换, 从而大大提高了程序开发的效率。

4. 程序执行的高效

Delphi 不但编译速度快, 而且生成的目标代码的执行效率也非常高。Delphi 编译 / 链接后生成完全本地代码 (Native Code), 因此可执行文件可以被独立执行和分发 (注: 对于 “绿色软件” 的开发, 这一点十分重要), 并不一定需要其他运行库 (Runtime Library) 的支持。或者, 你也可以选择动态链接编译 (Dynamic Link/Compile), 这样可以大大减小可执行文件的长度, 不过在这种情况下发布最终的程序时, 必须同时附带必要的运行库文件才能保证程序的正常运行, 也就是说必须要将可执行文件与其运行所需要的库文件一起打包成安装程序进

行分发。一般而言,动态链接编译虽然繁杂麻烦一些,但是对于软件的维护和二次升级开发,却有着相当大的优势。开发人员也许不需要修改可执行程序的任何部分,而只要相应更新运行库就可以实现软件故障的排除或者软件的升级换代了。

以上简单介绍了 Delphi 开发工具的优势,对于广大的程序开发人员来讲,使用 Delphi 开发 Windows 应用程序,将会大大地提高编程效率和降低软件维护成本。更重要的是,这种“以人为本”的设计思想将会使您随着应用开发的深入,慢慢发现编程的乐趣和魅力所在,编程将不再会是一件枯燥无味的机械工作了。

1.1.3 Delphi 开发工具的发展历史回顾

从核心上说,Delphi 其实是一个 Pascal 编译器。自从 15 年前 Anders Hejlsberg 写下第一个 Turbo Pascal 编译器以来,Borland 公司就一直在推动着 Pascal 编译器向前发展。Pascal 具有稳定、优雅以及编译速度快等特点。Delphi 系列的开发工具也不例外,随着版本的不断提升,编译器的功能和性能有了显著的增加和提高,但它的速度却减慢了很少。下面就让我们循着记忆的足迹,回过头去看一看 Delphi 以前的各个版本以及每一版本发行的背景吧。

1. Delphi 1.0

在使用 DOS 操作系统的年代,程序员只有两种选择:要么使用易于使用但速度慢的 Basic 语言,要么使用效率高但却复杂得难以驾驭的汇编语言。Turbo Pascal 以其结构化语言的简练和真编译器(True Compile)的性能,综合了以上两种语言的优势。因此 Borland 公司在 1995 年率先推出了 Delphi 1.0 版本,它是一个 16 位的开发工具,这个版本的 Delphi 提供了一种完全不同的开发 Windows 程序的方法,或者也可以称之为一种全新的理念,即可可视化的开发环境、编译后的可执行软件、动态链接技术、可扩展的数据库访问引擎(Extend DataBase Engine)。因此可以说,Delphi 1.0 奠定了快速应用程序开发(RAD)的基础。

2. Delphi 2.0

Borland 公司在 1996 年发布了 Delphi 2.0 版本,这个版本的 Delphi 增加了许多 Delphi 1.0 版本没有的功能,其中最重要的是:① 这个版本的 Delphi 已经是 32 位的编译器,因此能生成速度更快的应用程序;② 对象库也得到进一步的丰富和扩展,同时还完善了数据库支持功能,改进了字符串处理;③ 支持 OLE(对象链接嵌入)和可视化窗体继承;④ 与 16 位的 Delphi 1.0 兼容,这意味着用 Delphi 1.0 编写的程序不需要做任何修改,程序员只需要把源代码在 Delphi 2.0 的集成开发环境下重新编译一下,就可以生成 32 位的 Windows 95 应用程序,这使得软件的升级变得异常轻松和方便。综合以上优点,这个版本的 Delphi 当之无愧地成为了衡量其他 RAD 工具性能优劣的标准。

3. Delphi 3.0

Borland 公司一直没有停下开发更优秀的 Delphi 工具的步伐,1997 年,Delphi 3.0 版本面世了。相比较 Delphi 2.0 的开发目的而言,这个版本的 Delphi 除了继续兼容 16 位的 Delphi 之外,其更多的精力放在了对当时非常流行技术的支持上,例如:Com、ActiveX、WWW 应用

程序开发、“瘦”客户应用程序、多层数据库系统体系结构 (MIDAS) 等。同时它还首次采用了内视 (Code Insight) 技术, 大大减轻了代码编写的沉重负担。

4. Delphi 4.0

到了 1998 年, 在有了前面三个版本的经验积累之后, Borland 公司推出了可以算得上 Delphi 发展史上的里程碑的旗舰产品 Delphi 4.0, 它的兼容性更好, 对各种新技术的支持也越来越完美, 同时 MIDAS、DCOM 和 CORBRA 等技术也使 Delphi 4.0 的应用范围扩展到了企业级。Delphi 4.0 在长期的 Delphi 用户中赢得了广泛的信誉, 借助于 Delphi 4.0, Borland 公司有效地巩固了它在同 Microsoft 竞争中的地位。

5. Delphi 5.0

在 1999 年的下半年, Inprise 公司 (前身是 Borland) 再次让世人为之瞩目。Delphi 5.0 版本的发布让 Microsoft 刚刚巩固的 Visual Basic 阵地重新感到了压力, 这个版本的 Delphi 进一步增强了 IDE 集成开发环境和调试器的功能、提供了 TeamSource 小组开发软件和转换工具等。同时 Delphi 5.0 也为简化 Internet 的开发增加了许多新的功能, 包括: Active Server Object Wizard 用于创建 ASP; Internet Express 组件用于支持 XML 和新的 MIDAS 功能, 从而使 Delphi 成为 Internet 的一个通用数据平台。Delphi 5.0 进一步巩固了 Delphi 的市场占有率, 越来越多的程序员开始转向 Delphi 阵营。

6. Delphi 6.0

2001 年 5 月, Inprise 公司的又一力作 Delphi 6.0 问世了, Delphi 6.0 为 Microsoft.NET、Oracle.NOW 以及 Sun One 引入了一系列创建 Web 服务的新特性, 以扩展其在数据库中间件方面的能力, 其编译器具有直接支持 XML 和 SOAP 的能力。Delphi 6.0 的另一个新特性是 WebSnap, 它是一个 Web 页面设计器, 它的出现将简化创建和调试 Web 应用的工作。通过 Delphi 6.0, Borland 的 MIDAS 中间件开发框架将被 DataSnap 所替代。DataSnap 是一个具有激活 Web 服务功能的中间件环境, 用作数据库存取。这个版本的 Delphi 是继 Delphi 3.0 版本之后的又一标志性产品。

7. Delphi 7.0

这是迄今为止最新的 Delphi 版本了, 这个版本的 Delphi 功能更为强大, 不但继承了前面所有版本的优良特性, 而且纠正了很多错误, 它对 Web 服务的支持更加完善, 尤其是其附带的 Indy 系列组件, 几乎囊括了目前所有的 Web 功能, 用户无须了解太多的基础知识, 就可以在很短的时间内编写出自己的 Web 应用程序。智能化的编译器和帮助系统, 更给这个版本的 Delphi 增色不少。

虽然各个版本的 Delphi 发布时间彼此相隔不长, 但每个版本带给用户的却是质的变化。可以这样讲, 在 RAD 工具群雄并起的年代, Delphi 的确是一个非常理想的首选开发工具。看到这里, 读者朋友们应该明白为什么会有那么多的 Delphi 程序员和爱好者始终不渝地支持和使用了 Delphi 了吧。虽然外界好评如潮, 但 Inprise 公司却没有就此不前, 目前该公司正在积极筹措 Delphi.Net 开发工具的研发工作。

1.2 本章小结

通过本章的学习，您一定对 Windows 下的程序开发过程和开发工具有了很深刻的印象，会充分理解为什么 Delphi 能够战胜其他开发工具而成为 RAD 中的佼佼者，会对各个版本的 Delphi 特性有充分的认识，这将会帮助您在后面章节的学习中更加深刻地领会作者的意图，从而领略 Delphi 魔幻般的魅力所在。在下面的章节中，作者将带领您解决一个又一个的 Delphi 经典编程问题。通过学习本书，相信您一定会感到自己发现问题，思考问题和解决问题的过程是非常美妙的一件事情。