

趣味
程序

导学

Delphi

杨正华 吕跃春 编著



92 . TP311.56
Y296

科海电脑技术丛书

趣味程序导学 Delphi

杨正华 吕跃春 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书通过编写趣味游戏程序来引导读者学习 Delphi 6 的编程方法和技巧,形式新颖活泼,别具一格。

全书从 Delphi 语言基础知识和编写简单程序入手,将 Delphi 编程的知识点有机地分散在“幸运 52”、“速算 24”、“华容道”、“趣味作文”、“俄罗斯方块”、“我的 ICQ”等多个趣味游戏的程序设计示例中,引导读者轻松学习 Delphi 6 的相关知识、编程技术及技巧,其中包括 Delphi 6 中文本、多媒体、图形图像处理、数据库以及网络编程等内容。

本书以示例导学方式来组织内容,集趣味性、直观性和可操作性于一体,适用于 Delphi 6 初学者及对游戏程序感兴趣的电脑爱好者。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:趣味程序导学 Delphi

作 者:杨正华 吕跃春

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

印刷者:北京市耀华印刷有限公司

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:20.875 字数:516 千字

版 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数:0001~5000

盘 号:ISBN 7-900643-31-1

定 价:32.00 元(1CD)

目 录

第1章 初识Delphi.....	1
1.1 Delphi 6 概述.....	1
1.1.1 Delphi历史回顾.....	1
1.1.2 Delphi 6的新特性.....	2
1.2 Delphi 6集成开发环境.....	4
1.3 Delphi 6编程语言.....	6
1.3.1 代码书写规则.....	6
1.3.2 数据类型.....	8
1.3.3 语句.....	12
1.3.4 过程与函数.....	16
1.3.5 类与对象.....	18
1.3.6 单元.....	19
1.3.7 组件.....	21
1.4 本章知识点回顾.....	22
第2章 “幸运52”模拟游戏——Delphi初步应用.....	23
2.1 游戏效果说明.....	23
2.2 布置程序界面.....	25
2.2.1 认识Delphi中的组件.....	25
2.2.2 组件的布局.....	29
2.2.3 组件属性设置.....	31
2.3 初始化代码.....	34
2.3.1 查看代码.....	34
2.3.2 添加变量.....	35
2.3.3 事件处理的概念.....	36
2.3.4 窗体创建事件.....	36
2.3.5 按钮响应事件.....	39
2.3.6 字符输入事件.....	41
2.4 与用户的交互——引入消息框.....	42
2.5 程序的编译、运行和调试.....	43
2.5.1 编译和运行.....	43
2.5.2 调试.....	45
2.6 本章知识点回顾.....	51

第3章 “速算24”扑克游戏——单元、异常、逻辑	53
3.1 “速算24”扑克游戏效果说明	53
3.2 生成和建立程序	55
3.2.1 Image组件	55
3.2.2 Timer组件	56
3.2.3 设计初始界面	58
3.2.4 事件处理	60
3.3 单元间的互相引用	65
3.3.1 单元引用的概念	65
3.3.2 uses语句	65
3.3.3 创建另一个单元	65
3.4 异常处理	67
3.4.1 异常处理的概念	67
3.4.2 资源保护方式	67
3.4.3 异常响应方式	68
3.4.4 提供默认响应	68
3.5 数学逻辑单元	69
3.5.1 算法设计	69
3.5.2 字符串的相关函数	69
3.5.3 算法的代码编写	70
3.5.4 处理用户的错误输入	79
3.6 本章知识点回顾	84
第4章 华容道——图形图像编程	87
4.1 游戏效果说明	87
4.2 创建初始界面	88
4.3 Delphi绘图相关对象	91
4.3.1 TCanvas——画布对象	92
4.3.2 TPen——画笔对象	94
4.3.3 TBrush——画刷对象	94
4.3.4 TColor——颜色对象	95
4.4 Delphi图像相关对象	95
4.4.1 Windows位图的基本结构	95
4.4.2 图像对象	96
4.5 游戏内部逻辑	97
4.5.1 数据初始化	97
4.5.2 逻辑模块划分	99
4.5.3 游戏功能完善	113
4.5.4 游戏封面设计	120

4.6 本章知识点回顾.....	129
第5章 媒体播放器——多媒体程序设计	131
5.1 效果说明	131
5.2 媒体播放组件	132
5.2.1 Delphi对媒体播放的封装	132
5.2.2 窗体初步布置	139
5.2.3 实现组件相关操作	140
5.3 菜单和对话框	142
5.3.1 菜单组件	142
5.3.2 对话框组件	148
5.3.3 音量控制	150
5.4 本章知识点回顾	157
第6章 趣味作文——多文档界面	159
6.1 程序效果说明	159
6.2 多文档界面的生成	161
6.2.1 创建多文档界面的父窗体	162
6.2.2 设计菜单	162
6.2.3 创建多文档界面的子窗体	163
6.2.4 加入对话框组件	166
6.3 添加事件处理	167
6.3.1 父窗体菜单响应事件的建立	167
6.3.2 子窗口菜单响应事件的建立	170
6.3.3 添加自定义函数Open	176
6.4 游戏功能的完善	177
6.4.1 程序启动时不显示子窗体	177
6.4.2 正常关闭子窗体	178
6.4.3 在关闭子窗体时询问用户是否保存文件	179
6.4.4 更换应用程序的图标	180
6.4.5 在“窗口”菜单中显示子窗口列表	181
6.5 Windows API 应用	182
6.5.1 在多文档界面应用程序的父窗口上绘制背景	183
6.5.2 实现拖放文件	187
6.6 本章知识点回顾	189
第7章 北京公交车查询系统——数据库编程基础	191
7.1 系统使用说明	191
7.2 数据库系统基础知识	192
7.2.1 数据库简介	192

7.2.2 数据库系统的历史和发展趋势	194
7.2.3 数据库管理系统	195
7.2.4 数据库应用程序	196
7.3 Delphi的数据库应用程序开发体系	197
7.3.1 概述	197
7.3.2 DBD	200
7.3.3 Borland数据库引擎组件	203
7.3.4 Delphi数据控制组件	205
7.3.5 简单的数据库应用程序	207
7.4 数据集组件	210
7.4.1 数据集组件的几种状态	210
7.4.2 数据集组件的打开和关闭	211
7.4.3 数据集的导航	212
7.4.4 数据集中的数据维护	213
7.4.5 数据集组件与数据控制组件的连接	216
7.4.6 数据集组件的事件	216
7.5 TTable组件	217
7.5.1 TTable组件的主要属性	217
7.5.2 TTable组件的方法及应用	218
7.6 TQuery组件	222
7.6.1 SQL语言简介	222
7.6.2 TQuery组件的使用	224
7.7 程序编码	225
7.7.1 程序初始界面	225
7.7.2 程序内部编码	227
7.8 本章知识点回顾	229
第8章 俄罗斯方块（上）——自定义类与组件	232
8.1 游戏效果说明	232
8.2 Delphi面向对象编程	233
8.2.1 类的相关知识回顾	233
8.2.2 Delphi中类的实现	235
8.3 用户自定义类——TBlock	244
8.3.1 程序功能分析	244
8.3.2 TBlock类设计	245
8.3.3 方块类的具体实现	251
8.4 Delphi自定义组件	257
8.4.1 组件基础知识	257
8.4.2 新组件的定义	258

8.4.3 搭建组件框架.....	261
8.4.4 组件功能设计.....	266
8.4.5 组件注册.....	274
8.5 自定义组件——圆形按钮.....	276
8.5.1 组件框架搭建.....	276
8.5.2 组件具体实现.....	277
8.6 本章知识点回顾.....	283
第9章 俄罗斯方块（下）——Delphi综合编程.....	286
9.1 项目整体框架.....	286
9.1.1 功能分析.....	286
9.1.2 模块划分.....	287
9.2 程序编制实现.....	287
9.2.1 分页界面实现.....	287
9.2.2 主界面设计.....	289
9.2.3 排行榜设计.....	290
9.2.4 游戏设置页面.....	291
9.2.5 关于和帮助.....	292
9.2.6 功能实现.....	294
9.2.7 封面.....	299
9.2.8 流畅的动画.....	300
9.2.9 圆形按钮.....	301
9.3 本章知识点回顾.....	303
第10章 属于你的ICQ——网络编程.....	304
10.1 程序效果说明.....	304
10.2 网络基础知识.....	305
10.2.1 Internet的历史与现状.....	305
10.2.2 网络编程基础.....	306
10.3 Delphi 6中的网络组件.....	310
10.3.1 FastNet组件.....	310
10.3.2 消息收发组件.....	311
10.4 建立游戏界面.....	314
10.5 功 能 实 现.....	316
10.5.1 客户端编程.....	316
10.5.2 服务器端编程.....	318
10.5.3 程序分发.....	319
10.6 本章知识点回顾.....	320

第 1 章 初识 Delphi

Delphi是Borland公司推出的基于Windows环境的快速程序开发工具。Delphi的命名，来源于古希腊城市，这个城市因为拥有阿波罗神殿而著名。本章将简要介绍Delphi语言的发展历史、语言特点、编程环境等相关知识。通过本章的学习，读者将能初步了解Delphi。

1.1 Delphi 6 概述

1.1.1 Delphi历史回顾

Delphi是第四代编程语言，是RAD（Rapid Application Develop，快速应用开发）工具的代表。

从核心上说，Delphi其实是一个Pascal编译器。1995年，Borland推出了Delphi 1。Delphi 1提供了一种完全不同的开发Windows程序的方式：可视化的开发环境、编译后的可执行软件、DDL、数据库，以及可以毫无限制地给可视环境命名。而且Delphi 1是第一个综合了可视化开发环境、优化的源代码编译器、可扩展的数据库访问引擎的Windows开发工具，它奠定了RAD的概念。

一年后的Delphi 2在32位的操作系统Windows 95 和Windows NT 下实现了原有的一切功能。另外，Delphi 2还增加了许多Delphi 1没有的功能，例如32位的编译器能生成速度更快的应用程序，对象库得到进一步的丰富和扩展，完善了数据库支持，改进了字符串处理，支持OLE对象连接与嵌入和可视化窗体继承以及与16位的Delphi兼容等。Delphi 2 成为衡量其他RAD工具的标准。为了满足IT产业的需要，Delphi 2增强了数据库和客户端/服务器的功能。

1997年，Borland公司推出了Delphi 3。Delphi 3使本来极其复杂的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等技术变得非常容易。虽然Delphi 3与Delphi 1编写应用程序的基本方法大都相同，但Delphi 3的Code Insight（代码内视）技术却简化了代码编写过程。

1998年推出的Delphi 4致力于使Delphi更易于使用。Module Explore（模块导航）技术的引入使程序员能够以一致的图形界面浏览和编辑代码。代码导航和类自动生成的功能使程序员只需关注应用程序本身，而不必在输入代码上花费太多精力。IDE经过重新设计可支持浮动或可停靠的工具栏和窗口，调试器也做了改进。Delphi 4不愧为一个领先的开发工具，它的MIDAS、DCOM和CORBA等技术使Delphi 4 的应用范围扩展到企业级用户。

1999年, Delphi 5推出。Delphi 5通过增加更多的功能使程序的编写更简单。新功能包括: 进一步增强了IDE和调试器的功能、提供了Team Source小组开发软件和转换工具等。Delphi 5也为简化Internet的开发增加了许多新功能, 包括: Active Server Object Wizard用于创建ASP, Internet Express组件用于支持XML和MIDAS功能, 使Delphi成为Internet的一个通用数据平台。

1.1.2 Delphi 6的新特性

和以往Delphi的每一个新版本一样, Delphi 6继续改善了开发环境, 帮助程序员提高效率。Delphi 6在集成开发环境中加入了很多让程序员爱不释手的新功能, 下面是对它们的一个简单介绍。

1. Code Complete (代码完成)

Code Complete的功能在Delphi 5中就已经加入了, 这项功能非常受程序员的欢迎, 因为Code Complete可以大幅度地减少程序员需要键入的程序代码, 并且减少键入错误。Delphi 6的Code Complete功能在原有的基础上继续改善, 新的Code Complete窗口不但可以由程序员自行调整大小, 而且可以使用不同的颜色代表不同的对象, 例如变量、方法和特性等。

Delphi 6的Code Complete窗口加入了色彩分析, 以及对对象分门别类的能力。此外, 新的Code Complete窗口也会更聪明地过滤对象能够使用的PME, 避免了以往Delphi 5的Code Complete窗口无法显示一些PME的困窘。另外, 程序员在新的Code Complete窗口中选择使用某个方法之后, Code Complete会自动地在方法名称之后加上“() ; ”字符, 把光标停在圆括号之中, 并且自动显示这个方法需要的所有参数, Delphi的确比以前更聪明更方便了。

2. Object TreeView (对象树视图)

另外一个新的功能就是Delphi 6的Object TreeView窗口。这个新的窗口除了可以显示窗体中所有的对象之外, 也可以以分层方式表达组件之间的关系。当程序员在Object TreeView窗口中单击了一个组件之后, 这个组件会立刻出现在对象查看器之中, 这时程序员可以改变这个对象的特性值和事件处理程序。

当窗体复杂, 拥有大量的组件时, Object TreeView窗口可以让用户迅速找到这个组件, 并且知道与当前组件相关的组件, 非常方便。

此外Delphi 6的编辑器也强化了Class Complete的功能, 让程序员在定义类别程序代码时拥有更大的弹性, 并且加入了Interface Complete的功能, 让程序员在编写接口程序代码时也能够享有Code Complete的功能。

3. 可定制的 View 窗口

Delphi 6最强劲的集成开发环境功能应该要算是新的可定制的View窗口了。在Delphi 6中, 编辑器不但可以让程序员观看和编辑Object Pascal的程序代码, 而且如果程序员编辑

的是Web应用程序,那么还可以直接在编辑器中查看HTML程序代码、脚本语言程序代码,预览Web应用程序执行结果的画面以及Web应用程序产生的HTML程序代码。

4. 新一代数据存取引擎——DBExpress

Borland为了让Windows平台上的Delphi以及Linux平台上的Kylix拥有共同的数据存取引擎,因此决定开发新一代的数据存取引擎,这就是DBExpress。DBExpress是一组存取各种不同关联数据库的原生驱动程序,以及一组基于这些驱动程序而开发的统一的组件。由于原生驱动程序和组件的开发都考虑了跨平台,因此Delphi和Kylix都可以使用这组驱动程序和组件。程序员通过统一的组件来存取不同的数据库,以方便开发数据库应用系统。

5. 开发 Internet/Intranet 应用系统的 WebSnap 组件组

Delphi 6中最重要的新增功能之一应该要算是能够让程序员开发Internet/Intranet应用系统的新架构和组件,这个新的功能称为WebSnap。Delphi 6的WebSnap允许程序员使用组件和可视化的方式快速开发复杂的Internet/Intranet应用系统。WebSnap的特点是允许程序员直接在这些组件之中加入脚本语言,并且能够直接在Delphi的集成开发环境中撰写脚本语言,预览Web应用程序的输出结果以及Web应用程序产生的HTML程序代码。

6. MIDAS 的进化版本——DataSnap

在Delphi 6中,MIDAS被改名为DataSnap。DataSnap不但强化了MIDAS原有的功能,更加入了许多新的组件,让程序员可以使用它开发出更为强劲的应用系统。此外DataSnap也改善了MIDAS的执行效率,让使用它的应用程序能够执行得更快。

7. 对 XML/XSL 的支持

Delphi 6的另外一个重要的改良就是对于XML/XSL的支持。除了增加许多新的VCL组件,让程序员不必使用低阶的COM接口或是复杂的API就可以轻易的处理XML/XSL文件和资料之外,MIDAS的资料更可以直接输出成XML,或是把XML的资料直接导入为MIDAS的Data Packet,Delphi的程序员再也不需要为复杂的XML程序设计伤脑筋了。

8. 开发 Web Service 的向导和组件

Web Service该是现在最热门的Web技术之一了,不管是IBM或是Microsoft都在发展Web Service的解决方案。Delphi 6在这方面也不落后,Delphi 6提供了SOAP和Web Service向导,让程序员能够快速地开发SOAP应用系统,也可以汇入WSDL自动产生骨架程序代码。由于SOAP是未来Web和分布式应用系统的主流技术,而Web Service更是未来Web应用系统的主要软件架构,因此对于程序员来说,掌握这些技术非常重要。

Delphi 6是Borland支持SOAP的第一个产品,由于SOAP是标准的协议,因此通过Delphi 6开发的SOAP应用系统可以和COM+、EJB或是Microsoft的.NET沟通,Delphi的程序员现在就可以使用它开发未来的应用系统。在Delphi 6中也提供了6个VCL组件让程序员开发SOAP和Web Service的应用系统。Delphi 6 允许程序员快速建立SOAP应用程序和Web Service。

总的来说, Delphi 6最主要的新功能是在XML/XSL, Internet/Intranet以及SOAP和Web Service方面的加强。同时为了跨平台的目的, 在数据存取引擎、可视化组件架构和运行时期函数库方面也进行了大幅度的改善。至于在BDE, ADO和COM+方面则只有小幅的进步。

当Microsoft不再推出VB/VC新一代版本的Windows开发工具, 而Power Builder等其他Windows开发工具也逐渐被淘汰之际, Delphi 6可以说是惟一最新的Windows开发工具, 也是最好的Windows开发工具。如果你仍然需要在Windows下开发应用系统, 或是需要使用/整合最新的信息技术, 那么Delphi 6仍然是非常值得购买和升级的开发工具。

1.2 Delphi 6集成开发环境

1. 启动 Delphi

这里, 假定你已经在计算机上安装了Delphi 6。

选择Windows的开始菜单的程序命令, 再选择Borland Delphi 6子菜单命令, 最后选择Delphi 6命令, 就会打开Delphi的初始窗口, 如图1.1所示。

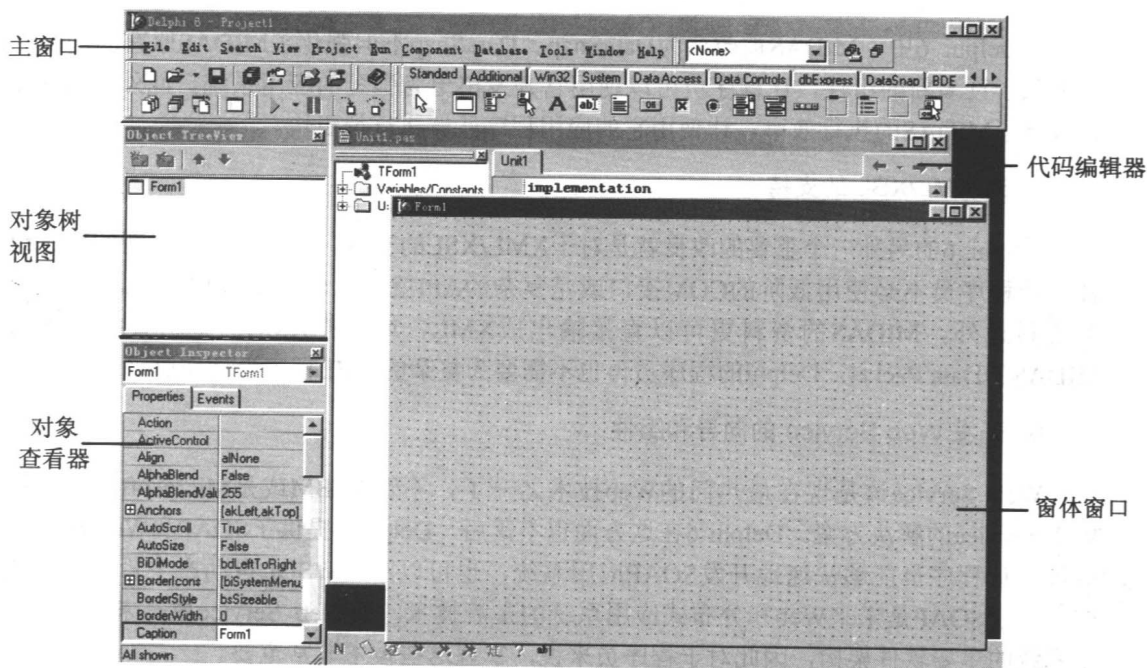


图 1.1 Delphi 6 初始窗口

在上图中, 已经标示出了Delphi初始窗口中的一些主要部分, 下面分别进行介绍。

2. 主窗口

Delphi的主窗口位于屏幕的上端,包括菜单(Menu)、加速条(Speed Bar)和组件面板(Component Panel)。Menu是下拉式主菜单;Speed Bar位于主窗口的左下端,由两排加速按钮组成,这些按钮是菜单命令的快捷方式,各种图标直观地表示了它能执行的动作;Component Panel由一行、若干页对象按钮组成,利用它来选择需要的组件并将它放到窗体中去。

3. 窗体窗口

窗体窗口是开展大部分设计工作的区域。首次启动Delphi 6时显示的是系统默认窗体Form1。可以把组件放在窗体中,通过移动位置、改变尺寸等操作随心所欲地安排它们,以此来开发应用程序的用户界面。可以把窗体想象成一个可以放置其他组件的容器。窗体上有栅格(Grids),供放置组件时对齐位置用,在程序运行时Grids是不可见的。

4. 代码编辑器

可以在代码编辑器中书写Delphi应用程序的源代码。当程序中含有不止一个窗口时,会有几个单元的源程序出现在代码编辑器中。代码编辑器的标题条中显示了当前正在编辑的单元文件名。要查看某一特定程序的源代码,只需用鼠标单击写有该单元文件名的页标签,就可以对该单元进行编辑。

5. 对象查看器

对象查看器(Object Inspector)窗口含有两个选项卡:Properties选项卡显示窗体中当前被选择组件的属性信息,并允许改变对象的属性;Events选项卡列出了当前组件可以响应的事件。单击Object Inspector下端的Events标签,使得Events选项卡可见。在一定的事件后边的空白处,可以定义对象接受到相应事件时执行的动作。

首次启动时,Object Inspector窗口显示的是当前窗体Form1的属性。Object Inspector根据对象属性的多少,决定是否有滚动显示。移动滚动条,可以查看当前对象的全部属性。

此外,Object Inspector上还有Object Selector(对象选择器),它位于Object Inspector上方的下拉式菜单中。Object Selector显示了窗体上所有组件的名称和类型,也包含窗体本身。用Object Selector可以很容易地在窗体的各个组件之间切换,也可以快速地回到窗体本身。当窗体中含有较多的对象时,这是切换对象尤其是回到窗体的最快捷途径。

想使Object Inspector一直可见,可将鼠标移到Object Inspector上,右击以启动Object Inspector的弹出式菜单,将其设置为Stay On Top。这对初学者常是一个很重要的设置方式。

6. 对象树视图

Delphi 6的一个新的功能就是对象树视图(Object TreeView)窗口。这个新的窗口除了可以显示窗体中所有的对象之外,也以树形方式表达组件之间拥有者的关系。当程序员在Object TreeView窗口中选择一个组件之后,这个组件会立刻出现在对象查看器之中,程序员可以改变这个对象的特性值和事件处理程序。

当窗体很复杂、拥有大量的组件时，Object TreeView窗口可以让你立刻找到这个组件，并且知道和它相关的组件，非常方便。

开始时，Object TreeView中只有窗体本身，随着组件的加入，这个树状视图的内容会越来越丰富。

1.3 Delphi 6编程语言

Delphi的编程语言是以Pascal为基础的。Pascal语言具有可读性好、编写容易的特点，这使得它很适合作为基础的开发语言。同时，使用编译器创建的应用程序只生成单个可执行文件（.exe），正是上述的这些优点，使得Pascal成为Delphi这种先进开发环境的编程语言。

1.3.1 代码书写规则

在编写自己的Object Pascal程序时，要注意程序的可读性。Pascal语言是英式结构语言，在程序中选择合适的缩排、大小写风格，并在需要时将程序代码分行，使得程序代码能够很容易地被自己和他人读懂。

1. 缩排

缩排是指在每一级有两个空格。不要在源代码中保留tab字符，因为tab字符会随着不同用户的不同设置和不同的资源管理工具（打印、文档、版本控制等）而代表不同的宽度。

2. 不区分大小写

Delphi是不区分大小写的语言，就是说，avar、aVar、Avar、AVAR在Delphi中代表的都是同一个变量，Delphi无法区分。基于这一个特性，编程时我们就不要试图通过大小写来区分不同的变量。

由于这种不敏感，为了使程序清晰易读，我们就要在编程过程中养成良好的命名习惯。

过程的名字应该以大写字母开头，名字中的其他单词也应以大写字母开头，以加强可读性。下面是一个格式不合常规的过程名称：

```
procedure thisfunctionwilldosomething;
```

下面是一个合适的区分大小写的名称：

```
procedure ThisFunctionWillDoSomething;
```

过程的名称应该同它的内容相符，一个引发其他行为的过程应以动词开头，例如：

```
procedure ReadDataFromFile;
```

一个用于设置输入参数的过程应以单词Set作为前缀，例如：

```
procedure SetFormCaption;
```

一个用来接收某个值的过程应以单词**Get**作为前缀，例如：

```
procedure GetFormCaption: string;
```

另外，我们也可以通过在各个单词间加入下划线达到清晰易读的效果。

3. 注释

给程序及时地加上注释是良好的编程习惯。Delphi有两种注释方式：

(1) 在大段的程序上加注“{}”，编辑器会把它们处理成空白。Delphi保留了Borland Pascal编辑器的风格，关键字采用黑体字，被注释的部分会变暗，这使得编程风格良好，程序易读易写。

(2) 也可以利用符号“//”在一行上加入注释。

4. 标识符

标识符是Delphi应用程序中一些量的名称，这些量包括**var**（变量）、**const**（常量）、**type**（类型）、**procedure**（过程）、**method**（方法）及其他。Object Pascal 在应用标识符时，必须首先声明它们。Object Pascal是强类型语言，要求相互赋值的变量类型必须相同，这样，编译器可以确保赋给变量或属性的值是正确的类型，若出现错误，也便于改正。

5. 变量

变量是程序代码中代表一个内存地址的标识符，而此地址的内存内容在程序代码执行时可以被改变。在使用变量前必须对它进行声明，即对它进行命名，并说明它的类型。在所有变量声明前加上保留字**var**。变量声明左边是变量的名称，右边则是该变量的类型，中间用“:”隔开。

下面声明了几个变量：

```
var  
    name      : string;  
    length    : integer;
```

6. 常量

常量在声明时就被赋予了一个值，在程序执行过程中是不可改变的。下面的例子声明了3个常量：

```
const  
    Pi = 3.14159;  
    Answer = 342;  
    ProductName = "Delphi";
```

像变量一样，常量也有类型。不同的是，常量假设其类型就是常量声明中它的值的类型。上面声明的3个常量的类型分别是实数型、整数型、字符串型。常量用“=”表示两边

的值是相等的。

1.3.2 数据类型

Object Pascal有多个预定义的数据类型，用户可以随意声明任何这些类型的变量。图1.2表示了Delphi中包括的数据类型。

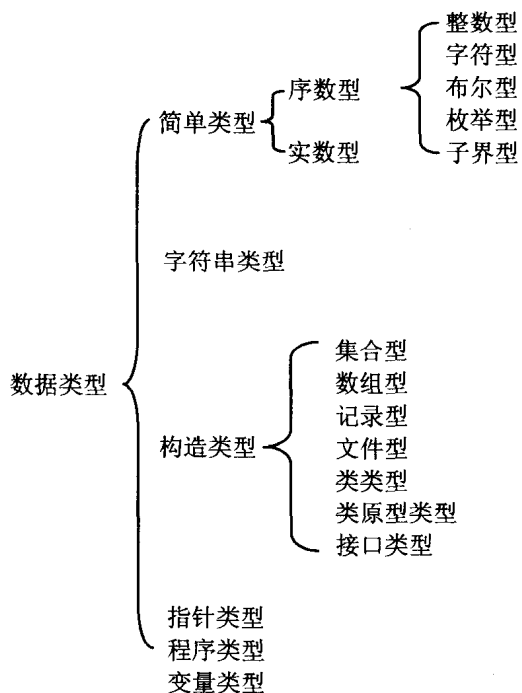


图 1.2 Delphi 预定义的数据类型

下面针对一些重要且常见的数据类型进行介绍。

1. 简单类型

简单类型包括序数型和实数型。这些类型的特点就是不可以再进行分割。下面对常用数据类型进行介绍。

(1) 整数型

整数型是整数的集合，包括多种不同的整数类别。表1.1说明了这些不同的类型。

表1.1 整数型说明

类型	表示范围	字节
ShortInt	-128~127	8位，有正负符号标志
SmallInt	-32 768~32 767	16位，有正负符号标志
LongInt	-2 147 443 648~2 147 483 647	32位，有正负符号标志
Int64	-2 ⁶³ ~2 ⁶³	64位，有正负符号标志
Byte	0~255	8位，无正负符号标志

(续表)

类型	表示范围	字节
Word	0~65 535	16位, 无正负符号标志
LongWord	0~4 294 967 295	32位, 无正负符号标志

(2) 字符型

字符型是ASCII字符的集合, 同样包括3种类型。

Char表示一个ASCII字符; AnsiChar用于保存扩展的ANSI字符; WideChar用于保存单个Unicode字符。

(3) 布尔型

布尔型只包含True或False两个值。

(4) 实数型

实数型是实数的集合, 包括多种不同的类型。表1.2说明各个类型的特点。需要说明的是, Real类型只有在和以前Borland Pascal兼容的情况下才使用, 否则应使用Double或Extended类型。

表1.2 实数型说明

类型	范围	小数位	存储字节
Real	$2.9 \times 10^9 \sim 1.7 \times 10^{38}$	11~12	6位
Single	$1.5 \times 10^{45} \sim 3.4 \times 10^{38}$	7~8	4位
Double	$5.0 \times 10^{-324} \sim 1.7 \times 10^{308}$	15~16	8位
Extended	$3.6 \times 10^{-4951} \sim 1.1 \times 10^{4932}$	19~20	10位
Comp	$-2^{63}+1 \sim 2^{63}-1$	19~20	8位
Currency	-922 337 203 685 477.5 808~ 922 337 203 685 477.5 807	19~20	8位

2. 构造类型

构造类型是由一个或多个已经实现定义的类型按一定的方法构造而成的新数据类型。Pascal定义了4种基本构造方法, 对应4种构造类型: 数组型、集合型、记录型、文件型。

另外, 作为面向对象的编程语言, Delphi还定义了3种特殊的构造类型: 类、类原型、接口。关于这3种类型, 我们将在后续的章节进行详细介绍。

(1) 数组型

数组是由固定个数的多个元素组成的有序结构, 其中所有元素的类型相同。

数组型的定义如下:

```
type 标识符=array [下标类型] of 数据类型
```

其中下标类型应该是顺序类型。在Delphi中, 数组的下标类型部分可以省略, 这样这个数组就是一个长度可变的数组。

一般, 只允许通过数组变量名与相应下标访问数组变量中的一个元素。但当两个数组