

全国高职高专规划教材

Delphi 程序设计

Programming Delphi

涂金德 主 编
牛立成 金林樵 副主编



科学出版社

www.sciencep.com



全国高职高专规划教材

Delphi 程序设计

涂金德 主 编

牛立成 金林樵 副主编

科学出版社

北 京

内 容 简 介

Delphi 简单易学、功能强大、编译速度快,特别是在数据库编程中表现突出,因而在众多的软件开发工具中脱颖而出。

本书以通俗、实用为主,系统地介绍了 Delphi 的语法、常用组件、数据库编程、Help 文件制作及使用、文件和流操作、Internet 程序开发等内容,并在每章节后安排了典型的操作实例及其详细的实现步骤,使初学者能够快速入门。

本书每章均给出要点、难点及习题,并有配套的《Delphi 程序设计实训教程》,希望学习过程中结合习题练习和实训内容,达到快速掌握 Delphi 应用程序开发的目的。

本书可作为大专院校计算机专业和非计算机专业学生学习 Delphi 编程的教材,尤其适合作高职高专院校计算机专业学生的教材,也可供其他读者自学参考。

图书在版编目(CIP)数据

Delphi 程序设计/涂金德主编. —北京:科学出版社,2003

(全国高职高专规划教材)

ISBN 7-03-012073-6

I. D… II. 涂… III. 软件工具—程序设计—高等学校:技术学校—教材 IV. TP311.56

中国版本图书馆CIP数据核字(2003)第071964号

策划编辑:李振格/责任编辑:舒立

责任印制:吕春珉/封面设计:东方人华平面设计部

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003年8月第 一 版 开本:787×1092 1/16

2003年8月第一次印刷 印张:18

印数:1—4 000 字数:423 000

定价:24.00元

(如有印装质量问题,我社负责调换〈路通〉)

全国高职高专规划教材编委会名单

主 任 俞瑞钊

副主任 陈庆章 蒋联海 周必水 刘加海

委 员 (以姓氏笔画为序)

王 雷 王筱慧 方 程 方锦明 卢菊洪 代绍庆

吕何新 朱 炜 刘向荣 江爱民 江锦祥 孙光弟

李天真 李永平 李良财 李明钧 李益明 余根墀

汪志达 沈凤池 沈安衢 张 元 张学辉 张锦祥

张德发 陈月波 陈晓燕 邵应珍 范剑波 欧阳江林

周国民 周建阳 赵小明 胡海影 秦学礼 徐文杰

凌 彦 曹哲新 戚海燕 龚祥国 章剑林 蒋黎红

董方武 鲁俊生 谢 川 谢晓飞 楼 丰 楼程伟

鞠洪尧

秘书长 熊盛新

本书编写人员名单

主 编 涂金德

副主编 牛立成 金林樵

撰稿人 涂金德 牛立成 金林樵 杨端容 杨 鹏 梁 方

前 言

Delphi 是 Borland 公司推出的基于 Windows 操作系统的可视化开发平台,它以 Object Pascal 语言为核心,是一个快速开发应用程序。Delphi 提供了灵活的可视化设计工具,将可视化界面与面向对象技术完美、紧密地结合起来。目前 Delphi 因简单易学、功能强大、编译速度快已在众多的软件开发工具中脱颖而出。它的优势之一在于开发数据库应用程序。在这方面,它不仅提供了大量的数据库组件,并配有数据库引擎 BDE,能通过 SQL Links 和 ODBC 访问多种数据库,同时提供了强大的开发网络数据库的能力。

为了突出 Delphi 在数据库开发方面的应用,兼顾 Internet 编程,同时考虑 Delphi 程序设计的系统性,本教程编写内容安排如下:

第 1 章介绍 Delphi 的编程语言——Object Pascal,第 2 章介绍 Delphi 常用组件,第 3 章介绍 Delphi 数据库基础知识,第 4 章介绍开发单层数据库应用程序,第 5 章介绍 Help 文件的制作及使用,第 6 章介绍文件和流操作,第 7 章介绍 Internet 程序开发。

本书没有将太多的篇幅用于操作性知识介绍,而将偏重于操作的内容,如“Delphi 集成开发环境”、“Delphi 程序调试”及课题设计安排在与本书配套使用的《Delphi 程序设计实训教程》中。

本教程由从事 Delphi 教学多年,并有丰富 Delphi 应用程序开发经验的教师执笔。从教学实践出发,以通俗、实用为主,使本书真正成为一本能够快速入门的实用的教程。

编者认为编写操作类教程的关键在于如何巧妙地结合理论与实际操作,如何选用和制作典型的操作实例。本书能够兼顾理论与操作,对于理论要求较高的内容,先系统简要地介绍理论,然后再给出典型的操作实例,既避免因缺乏理论基础而不能理解和制作实例,又避免在操作实例中穿插过多理论而干扰实例制作;对理论要求不高或理论与操作紧密相关的内容,把理论融入操作实例,并且在每一操作实例中按一般制作顺序给出详尽的操作步骤,使初学者也容易制作出实例,能够达到快速入门的目的。

本书中选用的操作实例典型、实用,从简单到复杂,并且每章均配有习题,主要是涉及理论方面的,能够很好地引起读者对一些重点难点问题的思考,并加深对它们的理解。至于更多的操作实例可以参考与本书配套的《Delphi 程序设计实训教程》。

本书第 1 章由杨端容编写,第 3 章由梁方编写,第 4 章由金林樵编写,第 6 章由杨鹏编写,第 7 章由牛立成编写,第 2 章、第 5 章由涂金德编写,最后由涂金德修改定稿。

本书可作为大专院校计算机专业和非计算机专业学生学习 Delphi 编程的教材,尤

其适合作高职高专院校计算机专业的教材，也可供其他读者自学参考。

当然，如何编写一本好的操作类教程还有许多问题值得专家们探讨，限于作者水平，加之时间仓促，书中不足之处在所难免，敬请广大读者批评指正。

编 者

2003 年 6 月

目 录

第 1 章 Delphi 的编程语言 ——Object Pascal	1
1.1 Delphi 概述.....	1
1.2 Object Pascal	2
1.2.1 注释、保留字和标识符	2
1.2.2 常量、变量和数据类型	3
1.2.3 运算符与表达式	12
1.2.4 语句	13
1.2.5 过程与函数	17
1.2.6 程序文件与单元文件	20
1.2.7 面向对象程序设计	23
1.2.8 异常与异常处理	26
小结.....	28
习题.....	28
第 2 章 Delphi 常用组件.....	31
2.1 窗体、按钮和分组组件.....	31
2.1.1 窗体	31
2.1.2 按钮和分组组件	35
2.1.3 操作实例	38
2.2 文本组件和图形组件	42
2.2.1 文本组件	42
2.2.2 图形组件	45
2.2.3 操作实例	47
2.3 菜单、工具栏、状态栏组件	50
2.3.1 菜单组件	50
2.3.2 工具栏组件	53
2.3.3 StatusBar (状态栏) 组件	54
2.3.4 ActionList (行为列表) 组件	55
2.3.5 操作实例	55
2.4 对话框函数和对话框组件	61
2.4.1 标准对话框函数	61
2.4.2 标准对话框组件	63
2.4.3 操作实例	64

2.5 其他常用组件	68
2.5.1 Timer (定时器) 组件	68
2.5.2 Animate (动画) 组件	68
2.5.3 ProgressBar (进程条) 组件	69
2.5.4 滚动组件	70
2.5.5 DateTimePicker 组件和 MonthCalendar 组件	72
2.5.6 操作实例	72
小结	76
习题	77
第 3 章 Delphi 数据库基础	81
3.1 数据库基本知识	81
3.1.1 数据库及数据库系统概述	81
3.1.2 关系数据库系统	82
3.2 Delphi 数据库工具	84
3.2.1 Database Desktop	84
3.2.2 BDE Administrator	92
3.2.3 SQL Explorer	99
3.2.4 Datapump	101
小结	104
习题	105
第 4 章 开发单层数据库应用程序	104
4.1 基于 Table 组件的数据库编程	104
4.1.1 数据表组件 Table	104
4.1.3 数据控制组件	129
4.1.4 操作实例	132
4.2 基于 Query 组件的数据库编程	135
4.2.1 SQL 语言简介	135
4.2.2 Query 组件	144
4.2.3 操作实例	149
4.3 数据库报表制作	152
4.3.1 报表的组成	153
4.3.2 Quick Report 报表组件	153
4.3.3 操作实例	156
4.4 数据统计图表的制作	159
4.4.1 统计图表组件	159
4.4.2 操作实例	165
4.5 单层数据库应用程序开发实例	167
4.5.1 功能总体设计	167

4.5.2 数据表的设计及创建	168
4.5.3 各功能模块的设计	170
4.5.4 系统的实现	171
4.5.5 系统的使用	203
4.6 制作安装程序	206
4.6.1 规划安装内容 (Organize Your Setup)	208
4.6.2 指定应用程序内容 (Specify Application Data)	210
4.6.3 配置目标系统 (Configure the Target System)	214
4.6.4 定制安装界面 (Customize the Setup Appearance)	216
4.6.5 定义安装系统要求和操作	218
4.6.6 准备发布安装 (Prepare for Release)	218
小结	219
习题	220
第 5 章 Help 文件的制作及使用	221
5.1 在 Delphi 中使用提示文本	221
5.1.1 使用 Hint 窗口显示提示文本	221
5.1.2 使用状态栏显示提示文本	222
5.2 制作 Help 文件	223
5.2.1 建立 Help 的 RTF 文件	223
5.2.2 建立 Help 目录文件	226
5.2.3 建立 Help 工程文件	227
5.2.4 生成 Help 文件	229
5.3 在 Delphi 程序中使用 Help 文件	230
5.3.1 利用 TControl 类组件的 HelpContext 属性	230
5.3.2 利用 Application 变量的属性和方法	230
小结	231
习题	231
第 6 章 文件和流	233
6.1 传统的文件操作	233
6.1.1 文本文件	233
6.1.2 类型文件	237
6.1.3 无类型文件	240
6.2 流操作	242
6.2.1 Delphi 中流的概念	242
6.2.2 TStream 类	243
6.2.3 常用的流类及其操作	245
6.2.4 操作实例	248

小结.....	251
习题.....	251
第7章 Internet 程序开发.....	252
7.1 文件传输服务程序开发.....	252
7.1.1 FTP 与 NMFTP 组件.....	252
7.1.2 操作实例.....	256
7.2 Web 服务器端程序开发.....	259
7.2.1 Web 服务器端程序开发基础.....	259
7.2.2 Web 服务器程序开发环境.....	260
7.2.3 操作实例.....	261
7.3 Socket 编程.....	265
7.3.1 Socket 与 Winsock.....	265
7.3.2 Socket 连接类型.....	265
7.3.3 Socket 对象.....	266
7.3.4 Socket 间数据传输与连接方式.....	266
7.3.5 Socket 组件.....	267
7.3.6 操作实例.....	268
小结.....	273
习题.....	273
主要参考文献.....	275

第 1 章 Delphi 的编程语言

——Object Pascal

本章要点

- Object Pascal 语法
- Delphi 的面向对象程序设计
- 异常及异常处理

本章难点

- Object Pascal 语法
- 面向对象的概念

1.1 Delphi 概述

Delphi 是著名的 Borland 公司开发的功能强大的 Windows 可视化应用程序开发工具。现在广为流传的一句话：“真正的程序员用 C，聪明的程序员用 Delphi”，这是对 Delphi 最实在的描述，它具有简单、高效、功能强大的特点，被称为第四代编程语言。和 VC 相比，Delphi 更简单、更易于掌握，而在功能上却丝毫不逊色；和 VB 相比，Delphi 则功能更强、更实用。可以说 Delphi 兼备了 VC 功能强大和 VB 简单易学的特点。

Delphi 提供了各种开发工具，包括集成开发环境、图像编辑（Image Editor），以及各种开发数据库的应用程序，如 DataBase Desktop 等。除此之外，还允许用户挂接其他的开发工具，如 Borland 公司的资源编辑器（Resource Workshop）。在 Delphi 众多的优势当中，它在数据库方面的特长显得尤为突出：适应于多种数据库结构，从客户机 / 服务器模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和大量的企业组件。

Delphi 发展至今，从 Delphi 1.0、Delphi 2.0 到现在的 Delphi 7.0，不断添加和改进各种特性，功能越来越强大。其中 Delphi 6 目前应用较为广泛。Delphi 6 是 Inprise 公司于 2001 年 6 月发布，是当前 Windows 平台上第一个全面支持 WEB 服务的快速开发工具，它支持所有新发布的工业标准和 RAD 环境，包括 XML/XSL，SOAP 和 WSDL 等。

Delphi 6 的新特性：

- Delphi 6 通过完全基于 SOAP 的网络服务和 XML 数据交换支持，从根本上简化了在互联网上构筑下一代电子商务和程序。
- Web Snap 使得 Delphi 能够无缝地集成到网站和现今流行的 HTML 开发环境的网站开发团队中。

- 构建单一源代码的 Windows/Linux 程序。

1.2 Object Pascal

Pascal 语言是一种语法严谨、条理清晰的结构化程序设计语言。使用 Pascal 语言开发的程序易于阅读、编译迅速、生成的代码简练。

Delphi 采用 Object Pascal 作为程序设计语言。Object Pascal 在 Pascal 语言的基础上引入了面向对象技术。它不仅具有 Pascal 语言的所有优点，还具有便于开发应用程序的面向对象的特征。

1.2.1 注释、保留字和标识符

1. 注释

编写一个 Object Pascal 程序时，为了增强程序的可读性，在程序中应选择合适的缩排、大小写风格，并在需要时将程序代码分行，并给程序加上一些必要的注释。Object Pascal 的注释语句有三种：大括号注释、带星号的小括号注释和 C 语言风格的双斜杠注释。

例如：

{ 大括号注释 }

(* 带星号的小括号注释 *)

// C 语言风格的双斜杠注释

其中“{”和“}”、“*”和“*”必须成对使用，它们可以用来注释多行内容。而“//”只能用来注释单行内容。

2. 保留字

保留字是 Object Pascal 预定义的具有特殊意义和用途的字符串。Object Pascal 中的保留字如下：

And	Array	as	asm	begin
Case	Class	const	constructor	destructor
Dispinterface	Div	do	downto	else
End	Except	exports	file	finalization
Finally	For	function	goto	if
implementation	In	inherited	initialization	inline
Interface	Is	label	library	mod
Nil	Not	object	to	for
Out	Packed	procedure	program	property
Raise	Record	repeat	resourcestring	set
Shl	Shr	string	then	threadvar
To	Try	type	until	unit
Uses	Var	while	with	xor

3. 标识符

标识符用来标识常量、变量、类型、属性、域、过程、函数、程序、单元、库以及包等。标识符的命名要注意以下几点：

- 标识符必须由字母（A~Z、a~z）、数字（0~9）或下划线（_）组成，它不区分字母的大小写。
- 标识符的第一个字符必须是字母或下划线。
- 标识符的长度不应超过 255 个字符，如果超过 255 个字符就只有前 255 个有效。
- 不能将保留字用作标识符，如 IF、AND、DO 等不能用作标识符。
- 尽量不要重新定义已预定义的标识符，如一些标准的数学函数 Sqrt、Sin 等。

例如，下面的标识符是合法的：

Student, main_form, edit1

下面的标识符是不合法的：

abc\$12 // \$ 不能用在标识符中

and // 不能将保留字用作标识符

1.2.2 常量、变量和数据类型

1. 常量

常量是在说明时就被赋予一个值，程序执行过程中，其值是不能被改变的。

Object Pascal 中的常量用 const 说明后才能使用。其说明格式为：

const 常量名=常量表达式

例如：

```
const
Pi=3.14159;
r=342;
str='Delphi';
```

常量也有自己的数据类型，它的类型由常量说明语句中等号左边的常量表达式值的类型所决定。上例中的三个常量的类型分别是实型、整型、字符串型。

2. 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执行时可以被改变。在使用变量前必须对它进行说明，对变量进行说明时要使用保留字 var。变量说明格式为：

var 变量名 1, 变量名 2..., 变量名 n: 类型;

例如：

```
var
  I, J, K: Integer;                //说明整型变量
  Ok: Boolean;                    //说明布尔类型变量
```

```
s:string;
```

```
//说明字符串类型变量
```

3. 数据类型

数据类型描述了变量和常量使用的情况，能够存储数据的形式以及能存储数据的多少。在使用变量前必须确定其类型，在声明常量时，常量值就代表它的类型。Object Pascal 中有丰富的数据类型，如图 1.1 所示。下面分别介绍这些数据类型。

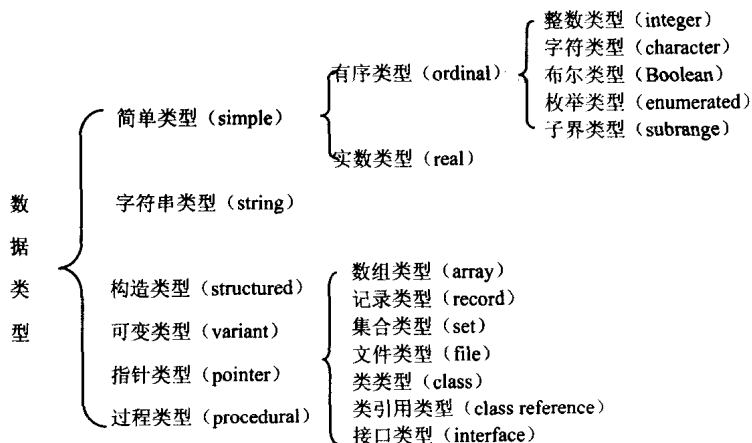


图 1.1 Object Pascal 中的数据类型

(1) 简单类型

简单类型是最基本的数据类型，它包括有序类型和实数类型。

1) 有序类型

有序类型是建立在概念“顺序”或“序列”基础上的数据类型。该数据类型中的每一取值均对应有一个序号。它不仅比较两个有序值的大小，而且可以求取给定有序值的前驱及后继，或者计算它们的最大或最小值。有序类型包括整型、字符类型、布尔类型、枚举类型和子界类型。

① 整数类型

Object Pascal 的整数类型如表 1.1 所示。

表 1.1 Object Pascal 中的各种整数类型

整数类型	取值范围	字节数	说 明
Integer	-2147483648~2147483647	4	有符号
Cardinal	0~4294967295	4	无符号
Shortint	-128~127	1	有符号
Smallint	-32768~32767	2	有符号
Longint	-2147483648~2147483647	4	有符号
Int64	$2^{63} \sim 2^{63}-1$	8	有符号
Byte	0~255	1	无符号
Word	0~65535	2	无符号
Longword	0~4294967295	4	无符号

② 布尔型

布尔型只包含 true 或 false 两个值, 可以进行逻辑运算。True 对应的序号为 1, false 对应的序号为 0。Object Pascal 中常用的布尔型为 Boolean(占 1 个字节)。为了保持与其他语言的兼容性, Object Pascal 还提供 Bytebool(占 1 个字节)、Wordbool(占 2 个字节)、Longbool(占 4 个字节)等布尔型。

③ 字符型

在 Object Pascal 中字符型数据的取值只能是一个字符, 不能是多个字符或空字符。

常用字符型是 Char, 它是 ANSI 字符集, 即通常所说的 ASCII 字符, 它占用 1 个字节, 其字符对应的序号为 ASCII 值。使用时单个字符前面要加单引号或用 # 号后紧跟对应字符的 ASCII 值。

例如: 'a' 或 #97 均表示字符 a。

但 # 号常用来表示不可显示字符, 如回车表示为: #13。

除此之外, 还有字符型 AnsiChar, 它也是 ANSI 字符集; WideChar, 它是 Unicode 字符集(前 256 个是 ANSI 字符集), 占用 2 个字节。

④ 枚举类型

枚举类型是用一组数量有限的标识符来表示一组连续的整数常数, 说明一个枚举类型的语法如下:

```
type 类型名 = (标识符 1, 标识符 2, ...标识符 n);
```

各标识符对应的序号为 0, 1, 2...n-1。

其中, 标识符列表中的标识符彼此之间用逗号隔开, 它列出了枚举类型所能取值的范围。也就是说, 一个枚举类型的变量, 其值总是所列的标识符列表中的一个。

例如:

```
Type
```

```
week = (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday);
```

在一个枚举类型被定义后, 在程序中就可以像使用 Object Pascal 预定义数据类型一样来使用该枚举类型, 如:

```
SomeDay: week;
```

声明枚举类型和枚举变量时要注意:

- 枚举的元素只能是标识符。
- 同一个枚举元素不能同时出现在多个枚举中。

例如:

```
Type Colors = (Red, Blue, Green, White, Black);
```

```
Another Colors = (Yellow, Line, Silver, Green);
```

上面的两个类型中都有 Green 元素, 这是不允许的。

不能直接用枚举元素参加运算, 以枚举类型变量 SomeDay 为例, 程序中如果出现

$X := \text{Friday} * 8$; 这是错误的, 只能用某个枚举元素对枚举变量赋值, 例如, $\text{SomeDays} := \text{Friday}$;

Object Pascal 中提供了几个标准的函数用于对枚举变量进行操作, 其中 Pred 函数用于返回指定元素的前一个元素, Succ 函数用于返回指定元素的后一个元素。

例如: $\text{Pred}(\text{Friday})$ 的返回值为 Thursday , $\text{Succ}(\text{Sunday})$ 的返回值为 Monday 。

函数 Ord 用于返回指定的枚举元素的值 (序号), 序号从 0 开始。

例如: $\text{Ord}(\text{Wednesday})$ 的返回值为 3。

⑤ 子界类型

子界类型一般用于某个变量变动范围的限制, 这个变量可以是以下几种数据类型: 整型、布尔型、字符型和枚举型, 说明子界类型的语法如下:

Type 类型名 = 下界常数..上界常数

例如:

Type

Thour = 0..23;

Caps = 'A'..'Z';

声明和使用子界类型时, 要注意以下几点:

- 上界常数和下界常数的类型必须一致, 且都是有序类型。
- 子界类型变量的取值是有一个范围的, 如果运算过程中超出这个范围将导致程序出错。

2) 实数类型

实数类型代表不同格式的浮点数, 主要是为了满足各方面对较高精度或者较大数值的计算。在 Object Pascal 中有几种不同实数类型, 如表 1.2 所示。

表 1.2 Object Pascal 中的各种实数类型

实数类型	取值范围	有效位数	字节数
Real	$5.0 \times 10^{-324} \sim 1.7 \times 10^{308}$	15~16	8
Real48	$2.9 \times 10^{-39} \sim 1.7 \times 10^{38}$	11~12	6
Single	$1.5 \times 10^{-45} \sim 7.4 \times 10^{38}$	7~8	4
Double	$5.0 \times 10^{-324} \sim 1.7 \times 10^{308}$	15~16	8
Extended	$7.4 \times 10^{-4932} \sim 1.1 \times 10^{4932}$	19~20	10
Comp	$-2^{63}+1 \sim 2^{63}-1$	19~20	8
Currency	-922337203685475.5808~922337203685475.5807	19~20	8

(2) 字符串类型

Object Pascal 的字符串类型分为短字符串和长字符串两种。短字符串是指字符串长度最大不超过 255 个字符的字符序列。当编译开关的状态为 $\{\$H-\}$ 时, 用保留字 String 声明的是一个短字符串。不管编译开关是 $\{\$H+\}$ 还是 $\{\$H-\}$, 用 ShortString 声明的变量总是短字符串类型。

例如: