

面向21世纪高等院校计算机教材系列

Delphi

程序设计教程

● 刘瑞新 汪远征 李凤华 编著



机械工业出版社
China Machine Press

面向 21 世纪高等院校计算机教材系列

Delphi 程序设计教程

刘瑞新 汪远征 李凤华 编著



机械工业出版社

本书详细讲授 Delphi 6 的语法基础、可视化编程方法、程序控制结构（顺序、分支、循环、过程与函数）、数据结构（数组、记录、枚举、子界与集合、指针、文件）、菜单与工具栏、图形图像与多媒体、数据库编程、面向对象程序设计基础。由于 Delphi 是标准的 Pascal 语言，本书按照第一学习的程序设计语言来组织教材，采用经典的 Pascal 教材结构并结合可视化编程编写而成，可以作为数据结构等课程的前导课。本书概念准确、语法结构严格，可使学生养成良好的编程习惯，本书层次分明、例题丰富，适合教师课堂教学和学生自学。书中每章均附有典型例题和习题。为了使读者尽快掌握学习内容，本书的配套教材《Delphi 程序设计教程习题及习题解答》对书中习题做了详细解答。

本书适合作为大学、高职、高专等学校的教材，同样适合其他希望快速学会 Delphi 编程的初学者。

图书在版编目（CIP）数据

Delphi 程序设计教程 / 刘瑞新等编著.

—北京：机械工业出版社，2001.9

(面向 21 世纪高等院校计算机教材系列)

ISBN 7-111-09236-8

I. D... II. 刘... III. Delphi 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 053218 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑：胡毓坚 责任印制：何全君

北京铭成印刷有限公司印刷·新华书店北京发行所发行

2001 年 9 月第 1 版·第 1 次印刷

787 mm×1092mm^{1/16}·22.25 印张·549 千字

0 001—6 000 册

定价：31.00 元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话：(010) 68993821、68326677-2527

出版说明

随着计算机技术的飞速发展，计算机在经济与社会发展中的地位日益重要。在高等院校的培养目标中，都将计算机知识与应用能力作为其重要的组成部分。为此，国家教育部根据高等院校非计算机专业的计算机培养目标，提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次教育的课程体系。根据计算机科学发展迅速的学科特点，计算机教育应面向社会，面向潮流，与社会接轨，与时代同行。随着计算机软硬件的不断更新换代，计算机教学内容也必须随之不断更新。

为满足高等院校计算机教材的需求，机械工业出版社聘请了清华大学、北方交通大学、北京邮电大学等院校的老师，经过反复研讨，结合当前计算机发展需要和编者长期从事计算机教学的经验，精心编写出“面向 21 世纪高等院校计算机教材”。

本套教材理论教学和实践教学相结合，图文并茂、内容实用、层次分明、讲解清晰、系统全面，其中融入了教师大量的教学经验，是各类高等院校、高等职业学校及相关院校的最佳教材，也可作为培训班和自学使用。

前 言

现在国内计算机程序语言教学正处在从 DOS 编程向 Windows 编程的转变阶段,作为 Delphi 教科书,许多书都要求读者先学习过 Pascal 语言或其他程序设计语言,这样就势必造成学习周期的延长。在知识大爆炸的今天,这样的学习方法已经非常不适合现在的计算机教学。

本书采用一种全新的教学方法,在学习 Delphi 之前,不需要先学任何程序语言,使学生在 60~80 课时内,学会 Delphi 的语法基础、可视化编程方法、程序控制结构(顺序、分支、循环、过程与函数)、数据结构(数组、记录、枚举、子界与集合、指针、文件)、菜单与工具栏、图形图像与多媒体、数据库编程、面向对象程序设计基础。

由于 Delphi 是标准的 Pascal 语言,本书按照第一学习的程序设计语言来组织教材,采用经典的 Pascal 教材结构并结合可视化编程编写而成,可以作为数据结构等课程的前导课。本书概念准确、语法结构严格,可使学生养成良好的编程习惯。书中采用 Delphi 6 作为语言环境。

书中每章均附有典型例题和习题。为了使读者尽快掌握学习内容,本书的配套教材《Delphi 程序设计教程习题及习题解答》对书中习题做了详细解答。

本书由刘瑞新、汪远征、李风华、万朝阳、孙宏莉、方习年、杨巍纳、刘敏、赖永恒、陈琪、王学锋、桑继耀、吴翔胜、唐科敏、张六成编著。由于作者水平有限,书中错误之处欢迎读者提出宝贵意见和建议。

本书适合作为大学、高职、高专等学校的教材,同样适合其他希望快速学会 Delphi 编程的初学者。

编 者

目 录

出版说明

前言

第 1 章 Delphi 的基础知识	1
1.1 Delphi 简介	1
1.2 Delphi 可视化编程的基本概念	2
1.2.1 对象的属性、事件和方法	2
1.2.2 控件	3
1.3 Delphi 可视化编程的环境	6
1.3.1 进入 Delphi 6 环境	6
1.3.2 Delphi 6 的主菜单	7
1.3.3 Delphi 6 的工具栏	8
1.3.4 Delphi 6 的组件板	8
1.3.5 对象观察器	9
1.3.6 对象 TreeView	11
1.3.7 窗体设计器	12
1.3.8 代码编辑器	12
1.4 Delphi 可视化编程的步骤	14
1.4.1 创建一个工程	14
1.4.2 添加控件	14
1.4.3 设置属性	15
1.4.4 编写代码	16
1.4.5 保存工程	16
1.4.6 运行工程	17
1.4.7 关闭工程和关闭 Delphi	18
1.4.8 修改工程	18
1.4.9 Delphi 程序的基本组成	18
1.5 控件的画法	22
1.5.1 选定控件	22
1.5.2 控件的缩放和移动	23
1.5.3 控件的复制与删除	23
1.5.4 控件的对齐	23
1.6 窗体介绍	24
1.6.1 窗体的属性	24
1.6.2 窗体的事件	26

1.6.3	窗体的方法	27
1.6.4	创建新窗体	27
习题 1		29
第 2 章	Delphi 的语法基础	30
2.1	保留字与标识符	30
2.1.1	保留字	30
2.1.2	标识符	31
2.1.3	指令符	31
2.2	数据类型	32
2.2.1	数值型数据	33
2.2.2	字符数据	34
2.2.3	布尔型数据	35
2.3	常量与变量	35
2.3.1	常量	35
2.3.2	变量	36
2.4	运算符与表达式	37
2.4.1	算术运算符	38
2.4.2	位运算符	39
2.4.3	字符串运算符	39
2.5	常用函数与过程	40
2.5.1	数学运算函数	40
2.5.2	字符处理函数	42
2.5.3	日期时间函数	45
2.5.4	顺序类型函数	48
2.6	语句	48
2.6.1	语句的书写规则	49
2.6.2	语句格式中的符号约定	49
2.6.3	赋值语句	50
2.6.4	注释语句	50
习题 2		51
第 3 章	顺序结构	53
3.1	与输入 / 输出有关的控件	53
3.1.1	Label 控件	53
3.1.2	Edit 控件	56
3.1.3	Memo 控件	61
3.1.4	MaskEdit 控件	64
3.2	使用对话框函数与对话框过程	67
3.2.1	ShowMessage 与 ShowMessageFmt 过程	67
3.2.2	MessageDlg 与 MessageDlgPos 函数	68

3.2.3	CreateMessageDialog 函数	70
3.2.4	InputBox 与 InputQuery 函数	70
3.3	使用焦点	73
3.4	用于分组的控件	74
3.4.1	Panel 控件	74
3.4.2	GroupBox 控件	77
习题 3		79
第 4 章	选择结构	80
4.1	条件表达式	80
4.1.1	关系运算符与关系表达式	80
4.1.2	布尔运算符与布尔表达式	81
4.1.3	运算符的优先顺序	82
4.1.4	求值方式	83
4.2	IF 语句	83
4.2.1	简单条件语句	84
4.2.2	复合条件语句	88
4.3	Case 语句	92
4.3.1	多分支选择结构	92
4.3.2	Case 语句的应用	93
4.4	按钮类控件	96
4.4.1	Button 控件	96
4.4.2	BitBtn 控件	97
4.4.3	CheckBox 控件	98
4.4.4	RadioButton 控件	100
4.4.5	RadioGroup 控件	103
4.4.6	SpeedButton 控件	104
习题 4		107
第 5 章	循环结构	110
5.1	While 语句	110
5.1.1	While 语句的语法	110
5.1.2	While 语句的应用	111
5.2	Repeat 语句	113
5.2.1	Repeat 语句的语法	113
5.2.2	Repeat 语句的应用	114
5.3	For 语句	116
5.3.1	For 语句的语法	117
5.3.2	For 语句的应用	117
5.4	循环的嵌套	120
5.5	列表框与组合框	122

5.5.1	列表框.....	123
5.5.2	组合框.....	130
5.6	使用计时器控件.....	135
5.6.1	设计一个电子表.....	135
5.6.2	设计一个计时器.....	136
5.6.3	设计一个流动字幕板.....	139
习题 5		141
第 6 章	过程与函数.....	144
6.1	过程.....	144
6.1.1	事件过程的创建与调用.....	144
6.1.2	通用过程的语法格式.....	146
6.1.3	通用过程的创建与调用.....	147
6.2	函数.....	150
6.2.1	函数的语法格式.....	150
6.2.2	函数的创建与调用.....	151
6.3	向子程序传递参数.....	154
6.3.1	形式参数与实际参数.....	154
6.3.2	参数传递方式.....	155
6.3.3	使用缺省参数.....	156
6.3.4	函数与过程的重载.....	157
6.4	子程序的嵌套与递归.....	158
6.4.1	子程序的嵌套.....	158
6.4.2	子程序的递归.....	161
6.5	变量的作用域.....	162
6.5.1	公有变量与私有变量.....	162
6.5.2	全程变量与局部变量.....	163
6.5.3	全程变量的声明.....	163
6.6	Delphi 的程序结构.....	166
6.6.1	主程序.....	166
6.6.2	单元与作用域.....	166
6.6.3	程序段与作用域.....	167
习题 6		167
第 7 章	枚举、子界与集合类型.....	171
7.1	类型定义语句.....	171
7.2	枚举类型.....	171
7.2.1	枚举类型定义与变量的声明.....	171
7.2.2	枚举类型的运算.....	172
7.3	子界类型.....	175
7.3.1	子界类型定义与变量的声明.....	175

7.3.2	子界类型变量的运算.....	176
7.4	集合类型.....	178
7.4.1	集合与集合类型.....	178
7.4.2	集合类型定义与变量声明.....	178
7.4.3	集合类型数据的运算.....	180
习题 7		185
第 8 章	数组与记录类型.....	188
8.1	数组及其应用.....	188
8.1.1	静态数组.....	188
8.1.2	动态数组.....	194
8.2	记录类型.....	196
8.2.1	记录类型的声明.....	197
8.2.2	访问记录域与 With 语句.....	198
8.2.3	记录的变体部分.....	202
习题 8		204
第 9 章	指针类型.....	205
9.1	指针类型的声明.....	205
9.2	指针的运算.....	207
9.3	无类型指针.....	210
9.4	字符指针类型.....	210
9.5	动态存储结构的实现.....	211
习题 9		217
第 10 章	菜单与工具栏.....	218
10.1	菜单.....	218
10.1.1	主菜单.....	218
10.1.2	弹出式菜单.....	224
10.2	工具栏.....	224
10.2.1	如何设置工具栏.....	224
10.2.2	工具栏的改进.....	228
习题 10		229
第 11 章	文件管理.....	231
11.1	公共对话框.....	231
11.1.1	公共对话框控件.....	231
11.1.2	公共对话框的应用举例.....	232
11.2	数据文件.....	233
11.2.1	适合于各种文件的基本操作.....	233
11.2.2	适合于文本文件的基本操作.....	235
11.2.3	有类型文件.....	241
11.2.4	无类型文件.....	244

11.3 文本相关控件	245
11.3.1 Win3.1 页中的文件相关控件	245
11.3.2 Win3.2 页及 Sample 页中的文件相关控件	247
习题 11	247
第 12 章 图形图像与多媒体技术	249
12.1 几何图形控件 Shape	249
12.2 图象控件 Image	251
12.3 画布 (Canvas) 对象	251
12.3.1 像素操作	252
12.3.2 画笔操作	252
12.3.3 画刷操作	254
12.4 PaintBox 画板控件	255
12.5 多媒体程序设计	257
12.5.1 卡通控件	257
12.5.2 媒体播放器	259
习题 12	260
第 13 章 Delphi 数据库编程	262
13.1 数据库系统开发基础	262
13.1.1 数据库管理系统 (DBMS)	262
13.1.2 Delphi 的数据库特性及功能简介	263
13.1.3 数据访问部件	264
13.1.4 BDE 数据集部件	265
13.1.5 数据控制部件	268
13.2 Delphi 数据库应用程序的开发方法和步骤	269
13.2.1 数据库应用程序的开发步骤	269
13.2.2 简单数据库应用的创建	271
13.2.3 数据库的字段对象	276
13.3 基于 ADO 的数据库应用程序	284
13.3.1 基于 ADO 的体系结构	285
13.3.2 ADO 数据集组件	285
13.3.3 连接到 ADO 数据容器	286
13.3.4 ADO 应用程序举例	287
13.4 Delphi 中的 SQL 编程	290
13.4.1 SQL 查询语言简介	290
13.4.2 TQuery 部件在 SQL 编程中的运用	291
13.4.3 SQL 语言编程概述	294
13.4.4 动态 SQL 语句的编程	296
13.4.5 SQL 编程实例	299
习题 13	309

第 14 章 面向对象程序设计基础.....	310
14.1 对象的基本概念.....	310
14.1.1 对象的特性.....	311
14.1.2 从一个对象中继承数据和方法.....	313
14.1.3 对象的范围.....	313
14.1.4 对象公有域和私有域的说明.....	314
14.1.5 访问对象的域和方法.....	314
14.1.6 对象变量的赋值.....	316
14.1.7 建立非可视化对象.....	317
14.2 类类型和对象.....	319
14.3 类的方法.....	321
14.3.1 方法的声明.....	321
14.3.2 构造和析构.....	322
14.3.3 方法指令字.....	324
14.3.4 抽象方法.....	327
14.3.5 重载方法与重定义方法.....	327
14.4 类的特性.....	329
14.4.1 声明特性.....	329
14.4.2 特性限定符.....	330
14.4.3 数组特性.....	332
14.4.4 特性的重载和重定义.....	333
14.5 类成员的可见性.....	334
14.6 类类型的兼容性.....	337
14.7 VCL 类结构与 Tobject 类.....	338
14.7.1 VCL 类结构.....	338
14.7.2 Tobject 类.....	341
习题 14.....	343

第1章 Delphi的基础知识

Delphi 是由 Inprise 公司（前 Borland 公司）推出的可视化编程环境，它提供了一种方便、快捷的 Windows 应用程序开发工具。Delphi 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。对于广大的程序开发人员来讲，使用 Delphi 开发应用软件，无疑会大大地提高编程效率。

1.1 Delphi 简介

Delphi 到现在已经历了 6 代产品的发展历程，每一代产品都是伴随 Windows 操作平台的升级而升级。

Delphi 1 几乎与 Windows 95 同时发行，这对于习惯使用 Borland 公司的 Turbo Pascal 而又希望开发 Windows 应用程序的程序员来说，无疑是个福音。因为 Delphi 1 可以看作是 Turbo Pascal 的 Windows 版，它首次为使用 Pascal 语言的人提供了一个可视化的编程环境。虽然 Microsoft 公司的 Visual Basic（VB）出现的时间要比 Delphi 1 早些，但它产生的代码是在解释器的控制下运行的，而 Delphi 直接把源程序文本编译为本机代码指令。

Delphi 1 仍然属于 16 位的开发环境，它创建的应用程序可以运行在 Windows 3.1、Windows 95 和 Windows NT 下，直到 Delphi 的下一个版本。

Delphi 2 中的主要新特征是对 32 位编程的支持，其创建的应用程序不再运行在 Windows 3.1 下。除此之外，Delphi 2 还增加了如数据模块等多种高效的数据重用功能。

Delphi 3 大大巩固了 Delphi 作为 Windows 应用程序开发工具的地位，它改进了集成开发环境（IDE）和“代码洞察”、对其组件库进行了许多增强、还完全支持 ActiveX 控件及其创建。

Delphi 4 则提供了 Object Pascal 语言的扩展，使之更像 C++。另外，还改进了“工程管理器”、“模块资源管理器”的功能。

当然，Delphi 4 不是一个很成功的版本，由于它的 Bug 比较多，系统稳定性较差，许多程序员宁愿放弃 Delphi 4 提供的许多新特性而使用广泛受到好评的 Delphi 3。

1999 年推出的 Delphi 5 不仅继承了低版本 Delphi 的诸多优点，克服了 Delphi 4 中存在的缺陷，而且增加了许多新的功能，对开发环境也做了改进和完善。

2001 年 6 月推出的 Delphi 6 是目前的最新版本，Delphi 6 又在 Delphi 5 的基础上增加了许多新的功能，对开发环境也做了改进和完善，特别是增加了大量的新组件，以适于网络开发的需要。

1.2 Delphi 可视化编程的基本概念

在 Windows 环境下编程, 传统的编程方法已经难以胜任。这是由于传统的编程方法使用的是面向过程、按顺序进行的机制, 这种编程方法的缺点是写程序的人始终要关心什么时候发生什么事情, 面对 Windows 事件驱动的工作方式, 处理各种事件及其可能的组合将有惊人的工作量。

面向对象的程序设计 (Object-Oriented Programming, 简记为 OOP) 立意于创建软件重用代码, 具备更好地模拟现实世界环境的能力, 这使它被公认为是自上而下编程的优胜者。它通过给程序中加入扩展语句, 把函数“封装”进 Windows 编程所必需的“对象”中, 使得复杂的工作条理清晰、编写容易。

一些早期的具有 OOP 性能的程序语言如 C++, Pascal, Smalltalk 等, 虽然具有面向对象的特征, 但不能轻松地画出可视化对象, 与用户交互能力较差, 仍然要编写大量的代码。

Delphi 使用“可视化”的编程方法。程序员不必自己建立对象, 利用 Delphi 所提供的可视“控件”, 只要在提供的程序框架中加入完成功能的代码, 如选择命令、移动鼠标等, 而不必考虑按精确次序执行的每个步骤。在这种机制下, 不必编写一个大型的程序, 而是建立一个由若干微小程序组成的应用程序, 这些微小程序可以由用户启动的事件来激发。这样就可以快速创建强大的应用程序而毋需涉及不必要的细节。

简单地说, “可视化编程”就是使用 Delphi 的 Object Pascal 语言, 利用它所提供的可视“控件”来创建“对象”。这是一种编程方法的新概念。

1.2.1 对象的属性、事件和方法

对象 (Object) 在现实生活中是很常见的, 如: 一个人是一个对象, 一部汽车是一个对象。如果将一部汽车拆开来便看有“发动机、方向盘、转向轴、车轮...”, 每一个又都是一个对象, 即汽车对象是由多个“子”对象组成的。在可视化编程中, 常见的对象有: 窗体、编辑框、列表框等。

从可视化编程的角度来看, 对象是一个具有属性 (数据) 和行为方式 (方法) 的实体。简单地说, 属性用于描述对象, 方法让对象做一些动作, 而对象动作时常会引起事件。一个对象建立以后, 其操作就通过与该对象有关的属性、事件和方法来描述。

在可视化编程中, Delphi 的窗体与控件都是程序被操作的对象, 这些对象都有其自己的属性和方法。

1. 对象的属性

属性 (Property) 是对象的一项描述内容, 用以描述对象的一个特性, 不同的对象有不同的属性, 而每个对象都由若干属性来描述。在可视化编程中, 常见的属性有标题 (Caption)、名称 (Name)、字体 (Font)、是否可见 (Visible) 等。通过修改或设置某些属性便能有效地控制对象的外观和操作。

属性值的设置或修改可以通过对象观察器 (Object Inspector) 中的属性窗口来进行, 也

可以通过编程的方法在程序运行的时候来改变对象的属性。

在程序中设置属性的一般格式是：

〈对象名〉.〈属性名〉:=〈属性值〉；

例如，设置标签对象 Label1 的标题为“欢迎使用 Delphi 6”的命令是：

Label1.Caption := '欢迎使用 Delphi 6';

2. 对象的事件

所谓事件(Event)，是由 Delphi 预先定义好的、能够被对象识别的动作，如单击(OnClick)事件、双击(OnDblClick)事件、移动鼠标(OnMouseMove)事件等，不同的对象能识别的事件也不相同。

对象的事件是固定的，用户不能建立新的事件。为此，Delphi 提供了丰富的内部事件，这些事件足以应付 Windows 中的绝大部分操作需要。

事件过程(Event Procudure)是程序员为处理特定事件而编写的一段程序。当事件由用户触发(如 OnClick)或由系统触发(如 OnActivate)时，对象就会对该事件作出响应(Respond)。响应某个事件后所执行的程序代码就是事件过程。一个对象可以识别一个或多个事件，因此可以使用一个或多个事件过程对用户或系统的事件作出响应。虽然一个对象可以拥有多个事件过程，但在程序中要使用多少事件过程，则由程序员根据程序的具体要求来确定。对于必须响应的事件需编写该事件的事件过程，而不必理会的事件则不需要编写事件过程，只要交给 Delphi 的默认处理程序即可，例如命令按钮的 OnClick 事件是最重要的事件，而 OnMouseUp 事件则可有可无，全视设计人员的需要。

3. 对象的方法

方法(Method)是与对象相关联的过程与函数的统称，即在对象中说明的并且用户可以调用的公共函数和过程，由 Delphi 系统内部定义，而不用编写代码。方法用于完成某种特定的功能而不能响应某一事件，如对象移动(Move)、画线(Line)、显示(Show)等。每个方法可以完成某个功能，但其实现步骤和细节用户既看不到也不能修改，用户能做的工作就是按照约定直接使用(调用)它们。

方法也被“封装”在对象之中，不同的对象具有不同的内部方法。Delphi 提供了大量的方法供不同的对象调用。选中对象，按 F1 键激活帮助，选择 Methods，就能看到可以调用的方法列表。

1.2.2 控件

控件是建立程序界面的基本元素，是可视化编程的基础。控件的使用，充分体现了当今流行的面向对象的程序设计思想。

1. 类与控件

Delphi 的控件由可视化组件库——Visual Component Library (VCL) 提供。这个组件库

是一个类库，其中定义了众多的类。VCL 中的有些类被可视化地安排在组件板（参见 1.3.4）上，称之为组件（components）；有些则是不可见的，如 Tstream 类。VCL 中所有的类都是从 TObject 类继承而来，TObject 类称为基类。组件是从 TComponent 类继承而来，从 TComponent 类中又派生出可视化组件与非可视化组件等子类。可视化组件或称控件是从 TControl 类派生的，如 TButton、TEdit 等；非可视化组件是控件以外的所有组件，即从 TComponent 类中派生但不是从 TControl 类派生的组件。在设计时，非可视化组件以图标形式出现在窗体上，如 TOpenDialog（运行时可见）、TDataSource（运行时不可见）。

VCL 中主要的类及它们之间的层次关系如图 1-1 所示。

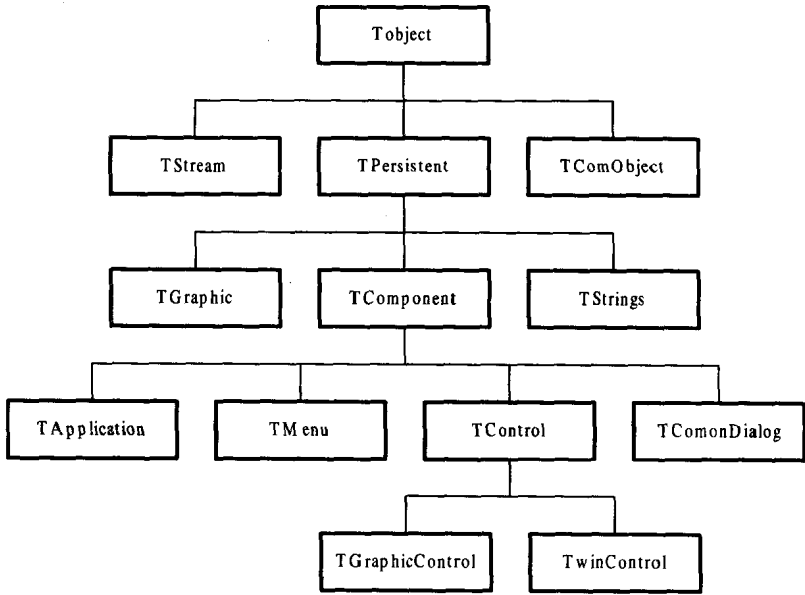


图1-1 VCL中主要的类之间的层次关系

为简单起见，本书将组件板中的组件（可视化组件与非可视化组件）都称为控件。

2. 常用控件介绍

Delphi 6 的组件板含有 27 个选项卡，总共包括 350 多个控件。表 1-1～表 1-3 给出前 3 个选项卡中最常用控件的简要说明。

表 1-1 Standard 选项卡中常用控件简介

组件图标	组件名称	说明
	MainMenu	用于设计和创建主菜单以及下拉式菜单，非可视控件
	PopupMenu	用于设计和创建弹出式菜单，可以在窗体或控件上使用，非可视控件
	Label	作为标签，用来显示不可变文本，可视控件
	Edit	用于设计供用户输入单行文本的编辑框，可视控件
	Memo	用于设计供用户输入多行文本的备注框，可视控件
	Button	按钮控件，可视控件

(续)

组件图标	组件名称	说明
	CheckBox	复选框控件，为用户提供选项，可视控件
	RadioButton	单选按钮，为用户提供多选一选择，可视控件
	ListBox	列表框，可视控件
	ComboBox	组合框，提供输入功能的列表框，可视控件
	ScrollBar	用于设计滚动条，可视控件
	GroupBox	组框，用于控件的分组，可视控件
	RadioGroup	用于设计单选按钮组，可视控件
	Panel	用于创建可以包含其他控件的面板，可视控件
	ActionList	用于收集应用程序对用户的回应，非可视控件

表 1-2 Additional 选项卡中常用控件简介

组件图标	组件名称	说明
	BitBtn	用于创建带位图的按钮，可视控件
	SpeedButton	与面板一起用于创建工具栏的按钮，可视控件
	MaskEdit	格式编辑框，用于控制数据的输入，可视控件
	StringGrid	用于简化显示字符串和与对象有关的数据，可视控件
	DrawGrid	用于以表格形式控制显示信息，可视控件
	Image	用于显示图标及图形对象，可视控件
	Shape	用于显示正方形、长方形、圆、椭圆等几何图形，可视控件
	Bevel	显示立体矩形，可视控件
	ScrollBar	可以滚动的显示框，可视控件
	CheckListBox	具有复选框的列表框，可视控件
	Splitter	用于在程序中创建用户可以改变大小的区域，可视控件
	StaticText	静态文本框，与 Label 相似，可视控件
	ControlBar	创建可以移动的工具栏，可视控件
	LabeledEdit	创建一个具有标签的编辑框，可视控件
	ColorBox	创建一个彩色枚举常量的下拉列表框，可视控件
	Chart	创建图标与图形，可视控件

表 1-3 Win32 选项卡中常用控件简介

组件图标	组件名称	说明
	TabControl	供用户选择的多标签控件，可视控件