

 普通高等教育计算机规划教材

Delphi

程序设计教程

第 2 版

刘瑞新 万朝阳 董淑娟 等编著



提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

Delphi 程序设计教程

第2版

刘瑞新 万朝阳 董淑娟 等编著



机械工业出版社

本教材采用任务驱动、案例教学的编写方式,详细讲授 Delphi 程序设计的方法和技巧,内容涵盖了基本控件使用和数据库开发等,包括 Delphi 的语法基础、基本控件编程方法、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、工具栏与多选项卡、图形图像与多媒体、数据库工具、数据库编程基础。本书从程序整体设计思路入手,按照从整体布局到具体实现的方式,结合实例,详细介绍了程序设计实现的整个过程。通过本教材的学习,使学生能快速入门并掌握 Delphi 程序设计的基本技能,轻松地创建出数据库应用程序并领略编程的成就感。

为了使读者尽快掌握学习内容,本书的配套教材《Delphi 程序设计教程上机指导及习题解答(第2版)》对书中习题作了详细解答,并为每章增加了上机试验项目,配套使用将使学习效果更佳。

本书适合作为普通高等院校、高职高专院校的教材,同样适合其他希望快速学会 Delphi 数据库应用程序的初学者。

图书在版编目(CIP)数据

Delphi 程序设计教程/刘瑞新等编著. —2版. —北京:机械工业出版社, 2006.3 (2012.2重印)

(普通高等教育计算机规划教材)

ISBN 978-7-111-09236-0

I. D... II. 刘... III. 软件工具—程序设计—高等学校—教材
IV. TP311.56

中国版本图书馆 CIP 数据核字(2006)第 003398 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:孙 业

责任印制:乔 宇

北京机工印刷厂印刷(三河市南杨庄国丰装订厂装订)

2012 年 2 月第 2 版·第 8 次印刷

184mm×260mm·21.5 印张·530 千字

70 001—72 000 册

标准书号:ISBN 978-7-111-09236-0

定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销 售 一 部:(010)68326294

教材网:<http://www.cmpedu.com>

销 售 二 部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- (4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- (5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前言

本教材的第1版得到许多高等院校师生的认可。为使教材更加完善,我们对原教材进行了一些调整和充实,使之更加符合当前高等院校对程序设计课程教学的新要求。在编写本教材的过程中,我们以程序设计作为主线,把难点分散到各章节中,对重点、难点分析透彻,注重知识内容的连贯性,取材深浅适宜。

本教材具有如下主要特点:

1. 任务驱动案例教学

本教材采用案例驱动的方法,以程序设计和窗体设计的过程为主线,以实例为引导,把介绍知识与实例设计、分析融于一体。

每个章节均以具有代表性、实用性、趣味性的实例引导,即先给出实例的最终结果,在分析实例的基础上,再详细介绍具体的实现过程。在实例的设计、制作过程中,把本章节的知识点融于实例之中,使学生能够快速掌握概念和操作方法。

本书每章均有习题,以方便学生练习。但对于初学程序设计的学生,在学习中往往会出现“上课听得懂,课后不会做”的现象。为此我们编写了本书的配套教材《Delphi 程序设计教程上机指导及习题解答(第2版)》,其中提供了本书全部的习题解答,并每章增加一个上机实验,以方便教师组织教学和学生的上机操作,同时增加一个综合应用实例,意在使学生通过实例学习应用系统的开发过程。

书中所有程序都在 Delphi 6、Delphi 7、Delphi 8 下调试通过。

2. 编写体系与结构合理

编写数据库应用程序的方法主要有两种:编写代码法、编写代码法与可视化工具软件法的联合使用。由于可视化编程比较直观,并且容易入门,所以越来越受到青睐。

本教材是基于可视化编程思想来编排的,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,主要从两个大方面进行介绍:一方面是基于基本语法与基本控件的使用进行介绍,从而能让读者快速入门;另一方面是基于数据库的开发,能让读者轻轻松松掌握基本数据库应用程序的开发。

3. 案例设计精巧

由于案例是教材的主体,案例设计的好坏是教材成败的关键。采用任务驱动教学法的难点就在于案例的设计,即在编写教材时,怎样在各章节中设计一个个相互关联而又相对独立的案例(即一个任务),并且案例的难度又要呈现阶梯式递增。

我们在处理本书时,把教材目标的全部内容细分为一系列知识点,通过“由简到繁、由易到难、循序渐进、深入浅出、承前启后”的案例,使学生真正学会解决实际应用问题的方法和能力。

在设计案例时,本书注重每个知识点之间的统筹兼顾,为教师和学生设计、构造出一系列典型的可操作的案例。所有案例都源于实际设计过程中最常见、最实用的方法,让学生在完成“任务”中掌握知识与方法。

4. 内容取舍得当

由于受到课时和软件版本等限制,不可能也不需要要求学生不分轻重缓急地掌握所谓的全面技能,而应该紧紧围绕实际“任务”介绍必要的、常用的、重要的基础知识和技能,同时给学生独立思考、探索和自我开拓的余地。为此,我们对 Delphi 程序设计的内容进行了合理取舍,精选出最本质、最实用的内容介绍给学生,以引导学生掌握基本的开发方法并从中获得编程的成就感,进一步激发学习 Delphi 程序设计的热情。

5. 配备教学资源

为了便于教师教学,本教材配有电子教案,可从机械工业出版社的网站下载。网址是:
<http://www.cmpbook.com/>。为了方便学生练习,本教材的源代码等也可从网站下载。

本教材主要由刘瑞新、万朝阳、董淑娟编写,参加编写的作者还有汪远征、秦国防、张二峰、孙培民、李自豪、王海翔、赵秀英、郭晓燕、李慧、马亚平、巩义云、刘婕、李晶、彭春艳、葛立峰、孙洪玲。作为新的教学方法的尝试,书中难免有错误和疏漏之处,敬请师生批评指正,以便完善。

本书涵盖内容较广,具有指导性强、示例典型、技术新颖及内容丰富等特点。本书适合作为普通高等院校、高职高专院校计算机及相关专业的教学用书,同时也可作为数据库应用程序爱好者或相关从业人员的自学用书。

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	1
1.2.1 对象的属性、事件和方法	1
1.2.2 控件	2
1.3 Delphi 可视化编程的环境	5
1.3.1 进入 Delphi 环境	5
1.3.2 Delphi 的主菜单	6
1.3.3 Delphi 的工具栏	7
1.3.4 Delphi 的组件板	7
1.3.5 对象观察器	8
1.3.6 对象 TreeView	10
1.3.7 窗体设计器	10
1.3.8 代码编辑器	11
1.4 Delphi 可视化编程的步骤	12
1.4.1 创建一个工程	13
1.4.2 添加控件	13
1.4.3 设置属性	13
1.4.4 编写代码	15
1.4.5 保存工程	15
1.4.6 运行工程	16
1.4.7 关闭工程和关闭 Delphi	17
1.4.8 修改工程	17
1.4.9 Delphi 程序的基本组成	17
1.5 控件的画法	20
1.5.1 选定控件	21
1.5.2 控件的缩放和移动	21
1.5.3 控件的复制与删除	22
1.5.4 控件的对齐	22
1.6 窗体介绍	23
1.6.1 窗体的属性	23
1.6.2 窗体的事件	25
1.6.3 窗体的方法	25

1.6.4	创建新窗体	26
1.7	习题	28
第2章 Delphi 的语法基础		29
2.1	保留字与标识符	29
2.1.1	保留字	29
2.1.2	标识符	29
2.1.3	指令符	30
2.2	数据类型	31
2.2.1	数值型数据	31
2.2.2	字符数据	32
2.2.3	布尔型数据	33
2.3	常量与变量	34
2.3.1	常量	34
2.3.2	变量	35
2.4	运算符与表达式	36
2.4.1	算术运算符与算术表达式	36
2.4.2	位运算符	37
2.4.3	字符串运算符	38
2.5	常用函数与过程	38
2.5.1	数学运算函数	38
2.5.2	字符处理函数	40
2.5.3	日期时间函数	43
2.5.4	顺序类型函数	45
2.6	语句	46
2.6.1	语句的书写规则	46
2.6.2	语句格式中的符号约定	47
2.6.3	赋值语句	47
2.6.4	注释语句	48
2.7	习题	48
第3章 常用基本控件		51
3.1	与输入/输出有关的控件	51
3.1.1	Label 控件	51
3.1.2	Edit 控件	53
3.1.3	Memo 控件	55
3.1.4	MaskEdit 控件	58
3.2	使用对话框函数与对话框过程	61
3.2.1	ShowMessage 与 ShowMessageFmt 过程	61
3.2.2	MessageDlg 与 MessageDlgPos 函数	62
3.2.3	CreateMessageDialog 函数	64

3.2.4	InputBox 与 InputQuery 函数	64
3.3	使用焦点	66
3.4	用于分组的控件	67
3.4.1	Panel 控件	67
3.4.2	GroupBox 控件	70
3.5	按钮类控件	72
3.5.1	Button 控件	72
3.5.2	BitBtn 控件	73
3.5.3	CheckBox 控件	74
3.5.4	RadioButton 控件	76
3.5.5	RadioGroup 控件	79
3.5.6	SpeedButton 控件	80
3.6	列表框与组合框	81
3.6.1	列表框	81
3.6.2	组合框	86
3.7	使用计时器控件	91
3.7.1	设计一个电子表	91
3.7.2	设计一个计时器	93
3.7.3	设计一个流动字幕板	96
3.8	习题	97
第 4 章	基本语句结构	99
4.1	顺序结构	99
4.2	选择结构	99
4.2.1	关系运算符和关系表达式	100
4.2.2	布尔运算符与布尔表达式	101
4.2.3	运算符的优先顺序	101
4.2.4	求值方式	102
4.2.5	If 语句	102
4.2.6	Case 语句	107
4.3	循环结构	111
4.3.1	While 语句	111
4.3.2	Repeat 语句	113
4.3.3	For 语句	115
4.3.4	循环的嵌套	117
4.4	习题	118
第 5 章	过程与函数	122
5.1	过程	122
5.1.1	事件过程的创建与调用	122
5.1.2	通用过程的语法格式	124

5.1.3	通用过程的创建与调用	125
5.2	函数	128
5.2.1	函数的语法格式	128
5.2.2	函数的创建与调用	129
5.3	向子程序传递参数	132
5.3.1	形式参数与实际参数	132
5.3.2	参数传递方式	133
5.3.3	使用默认参数	134
5.3.4	函数与过程的重载	135
5.4	子程序的嵌套与递归	136
5.4.1	子程序的嵌套	136
5.4.2	子程序的递归	139
5.5	变量的作用域	140
5.5.1	公有变量与私有变量	141
5.5.2	全程变量与局部变量	141
5.5.3	全程变量的声明	141
5.6	Delphi 的程序结构	144
5.6.1	主程序	144
5.6.2	单元与作用域	144
5.6.3	程序段与作用域	145
5.7	习题	146
第 6 章	枚举、子界与集合类型	149
6.1	类型定义语句	149
6.2	枚举类型	149
6.2.1	枚举类型定义与变量的声明	149
6.2.2	枚举类型的运算	150
6.3	子界类型	152
6.3.1	子界类型定义与变量的声明	153
6.3.2	子界类型变量的运算	153
6.4	集合类型	155
6.4.1	集合与集合类型	155
6.4.2	集合类型定义与变量声明	156
6.4.3	集合类型数据的运算	157
6.5	习题	162
第 7 章	数组与记录类型	167
7.1	数组及其应用	167
7.1.1	静态数组	167
7.1.2	动态数组	173
7.2	记录类型	175

7.2.1	记录类型的声明	176
7.2.2	访问记录域与 With 语句	177
7.2.3	记录的变体部分	181
7.3	习题	182
第 8 章	指针类型	183
8.1	指针类型的声明	183
8.2	指针的运算	185
8.3	无类型指针	187
8.4	字符指针类型	188
8.5	动态存储结构的实现	189
8.6	习题	194
第 9 章	菜单、工具栏与多选项卡	195
9.1	菜单	195
9.1.1	主菜单	195
9.1.2	弹出式菜单	201
9.2	工具栏	201
9.2.1	如何设置工具栏	201
9.2.2	工具栏的改进	205
9.3	多选项卡	206
9.4	习题	208
第 10 章	文件管理	210
10.1	公共对话框	210
10.1.1	公共对话框控件	210
10.1.2	公共对话框的应用举例	211
10.2	数据文件	212
10.2.1	适合于各种文件的基本操作	213
10.2.2	适合于文本文件的基本操作	214
10.2.3	有类型文件	220
10.2.4	无类型文件	223
10.3	文本相关控件	223
10.3.1	Win3.1 页中的文件相关控件	223
10.3.2	Win3.2 页及 Sample 页中的文件相关控件	225
10.4	习题	226
第 11 章	图形图像与多媒体技术	228
11.1	几何图形控件 Shape	228
11.2	图像控件 Image	229
11.3	画布 (Canvas) 对象	230
11.3.1	像素操作	231
11.3.2	画笔操作	231

11.3.3	画刷操作	232
11.4	PaintBox 画板控件	233
11.5	多媒体程序设计	235
11.5.1	卡通控件	235
11.5.2	媒体播放器	237
11.6	习题	239
第 12 章	Delphi 数据库开发工具	241
12.1	数据库系统概述	241
12.1.1	数据库管理系统 (DBMS)	241
12.1.2	数据库应用程序	242
12.1.3	Delphi 的数据库辅助工具	243
12.2	Database Desktop	243
12.2.1	Database Desktop 概述	243
12.2.2	设定工作目录和私有目录	244
12.2.3	用 Database Desktop 建立数据库表	244
12.2.4	编辑数据	246
12.2.5	数据库表的各种属性	246
12.2.6	建立其他型数据库表	248
12.2.7	使用 SQL 查询和 QBE 查询	249
12.3	BDE Administrator	250
12.4	ODBC (开放数据库链路)	251
12.5	SQL Explorer (数据库管理器)	252
12.5.1	数据库的管理	252
12.5.2	数据库的访问	253
12.6	Data Pump (数据转移)	254
12.7	数据库工具的综合实例	255
12.8	习题	259
第 13 章	BDE 数据库应用程序开发概述	260
13.1	数据访问控件	260
13.1.1	TTable 控件	262
13.1.2	TQuery 控件	272
13.1.3	数据集控件的事件	279
13.1.4	TDataSource 控件	279
13.2	数据库的字段对象	281
13.3	数据浏览控件	285
13.3.1	数据浏览控件的基本属性	286
13.3.2	TDBGrid 控件	287
13.3.3	TDBNavigator 控件	290
13.3.4	其他数据浏览控件	294

13.4 习题	300
第 14 章 ADO 数据库开发基础	303
14.1 ADO 控件页介绍	303
14.2 TADOConnection 控件	304
14.2.1 TADOConnection 的常用属性	304
14.2.2 TADOConnection 的主要方法	307
14.2.3 TADOConnection 的主要事件	308
14.3 TADOCommand 控件	309
14.3.1 TADOCommand 的主要属性	309
14.3.2 TADOCommand 的主要方法	310
14.4 TADODataset 控件	313
14.4.1 TADODataset 的常用属性	313
14.4.2 TADODataset 的常用方法	314
14.5 TADOTable 控件	316
14.6 TADOQuery 控件	318
14.7 TADOStoredProc 控件	320
14.8 使用 ADO 开发数据库的综合实例	321
14.9 习题	330

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。程序开发人员使用 Delphi 开发应用软件，无疑会大大提高编程效率。

1.1 Delphi 简介

Delphi 到 2005 年已经历了 9 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。

本书采用 Delphi 6 作为平台，讲述 Delphi 程序开发的基本知识。

1.2 Delphi 可视化编程的基本概念

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，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，只要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无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” 的命令是:

```
Label1.Caption := '欢迎使用 Delphi';
```

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 Procedure) 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, 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的形式出现在窗体上，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。

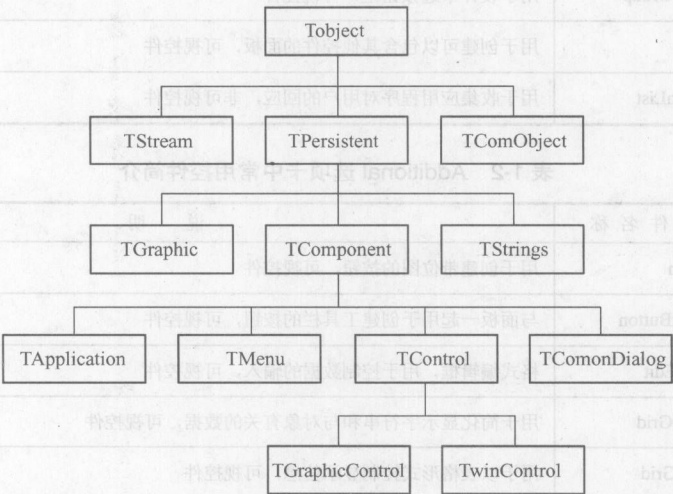


图 1-1 VCL 中主要的类之间的层次关系

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

组件图标	组件名称	说明
	Button	按钮控件，可视控件
	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