

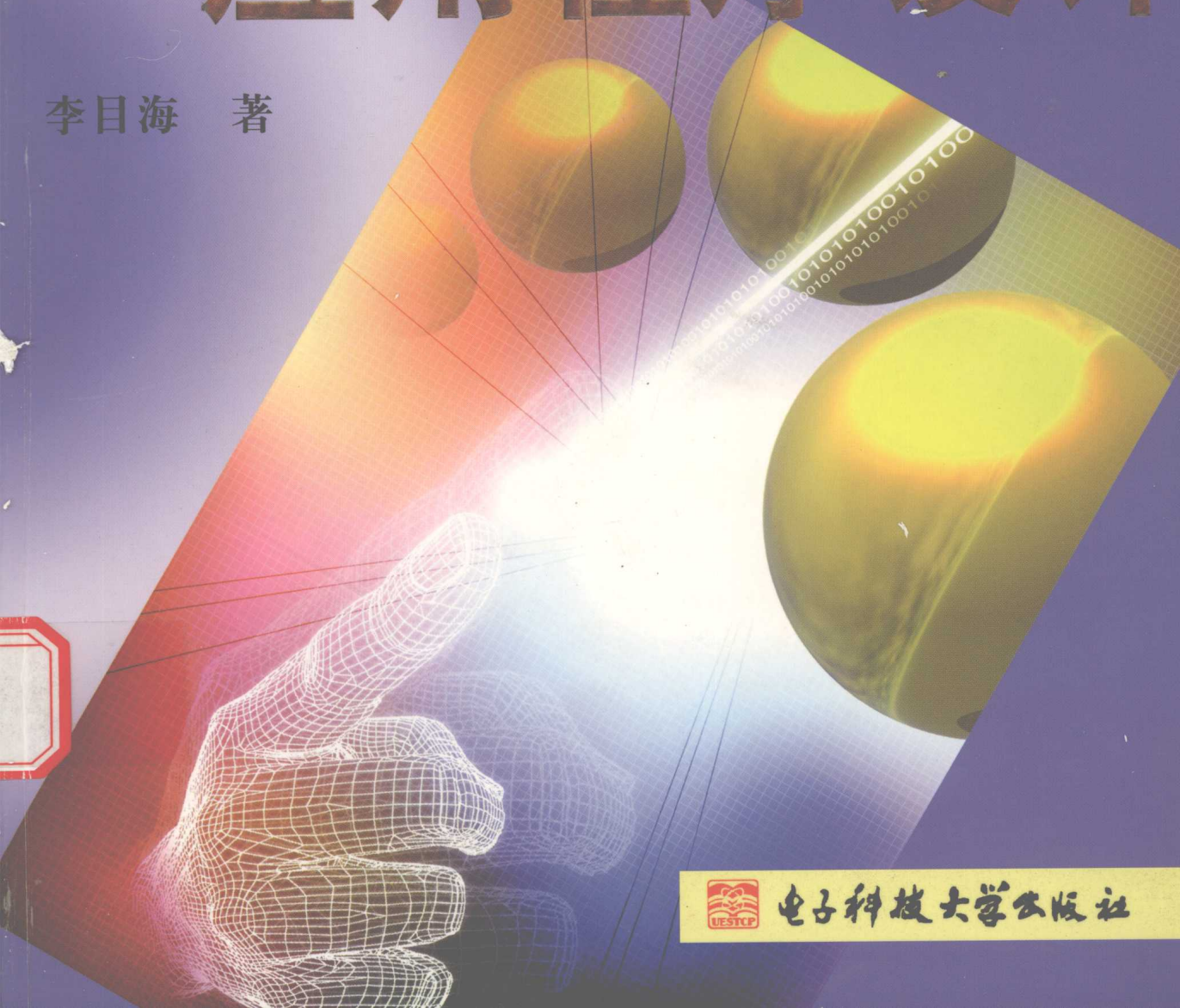


21世纪计算机科学与技术系列教材

Delphi

应用程序设计

李目海 著



电子科技大学出版社

21 世纪计算机科学与技术系列教材

Delphi 应用程序设计

李目海 著

电子科技大学出版社

内 容 简 介

本书采用“树式”教学模式思想,向读者全面讲解基于 Delphi 平台的应用程序开发技术。全书包括四个部分:第一部分介绍 Delphi 的集成开发环境和程序设计的基础知识,第二部分讲述应用程序的组成结构和建立方法,第三部分讲述如何根据用户需求调查、分析问题以及建立系统模型的方法,第四部分重点介绍开发应用程序的关键技术。

全书围绕应用开发这一主题,通过浅显易懂的语言、大量直观的图示和贴近现实的丰富实例,较全面地剖析了开发应用程序的全过程。本书可作为高校理工科本、专科学生教材,也可作为程序员业务学习用书。

图书在版编目(CIP)数据

Delphi 应用程序设计/李目海著. —成都:电子科技大学出版社,2005.8

(21 世纪计算机科学与技术系列教材)

ISBN 7-81094-873-3

I. D… II. 李… 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(2005)第 075832 号

21 世纪计算机科学与技术系列教材

Delphi 应用程序设计

李目海 著

出 版	电子科技大学出版社(成都市建设北路二段四号 邮编 610054)
责任编辑	张致强
发 行	新华书店
印 刷	安徽省蚌埠广达印务有限公司
开 本	787×1092 1/16 印张 26 字数 648 千字
版 次	2005 年 8 月第 1 版
印 次	2005 年 8 月第 1 次印刷
书 号	ISBN 7-81094-873-3 / TP·464
印 数	1—5000 册
定 价	36.00 元

前 言

Delphi 是 Borland 公司研制的可视化软件开发平台,它是目前设计 Windows 应用程序最方便、最快捷以及功能最强大的语言之一。

Delphi 自 1995 年诞生之日起就号称“VB 杀手”,在程序员中也一直流传着“真正的程序员学 C,聪明的程序员学 Delphi”的说法,目前越来越多的程序员将 Delphi 作为开发应用软件的首选平台,之所以产生这一现象,是因为 Delphi 具有以下特点:

(一)简单易学的开发环境

这里所说的“简单易学”包括两个方面的含义:一方面,Delphi 使用面向对象的 Pascal 语言为基础语言,而 Pascal 语言是计算机界公认的教学语言,它结构严谨、层次清晰,特别适合于初学者学习。Pascal 的这些特点,使基于 Delphi 开发的应用程序具有同样的特征。另一方面,Delphi 支持可视化的程序设计,使软件开发像搭积木那样简单、直观;同时,其简洁明快的开发环境、方便快捷的功能菜单,使编程者一旦使用,就会爱不释手。

(二)功能强大的可视化组件(VCL)

越来越多的程序员将 Delphi 作为首选开发平台,主要原因就在于 Delphi 为程序员提供了数百个功能强大的可视化组件,同时,程序员还能根据需要开发自己的组件产品,并安装到 Delphi 系统中,这一功能极大地提高了软件开发效率。例如,利用本书作者提供的组件,读者可以在不到 1 分钟的时间内,不写任何代码,就能开发出一个具有浏览、编辑、插入、删除、查询、另存和打印等多种功能的数据库应用软件。可以说,选择 Delphi 开发平台就意味着高效和成功。

(三)灵活的数据库支持

Delphi 没有专用数据库,然而,它却支持几乎所有的数据库,如 Access、VFoxpro、dBase、Paradox 等桌面数据库和 MS SQL Server、Oracle、Sybase、InterBase 等大型数据库。利用 Delphi 提供的 DataSnap、BDE、ADO 和 dbExpress 等组件以及上述数据库可以很容易地开发出数据库应用软件,同时,Delphi 较先进的多层数据库技术(MIDAS)使用户开发的应用软件更能适应当前社会的需要。

(四)对 MS Office 软件的访问

Delphi 支持对 MS Office 系列软件的访问,使程序员能够编写可以直接处理 Word、Excel 等文档的应用程序,进而可以开发针对 MS Office 软件操作的自动阅卷系统。笔者已利用这一技术开发出中小学使用的“‘信息技术’考试系统”和高

校使用的“‘计算机文化基础’考试及自动阅卷系统”。此外,利用这一技术,还能将应用软件中难以实现的打印、排版等功能转移到 MS Office 软件中执行,从而在大大降低编程难度的同时,提高应用软件的性能。

基于 Delphi 的上述特点,笔者从 1998 年至今一直从事 Delphi 的教学研究与应用软件的开发工作;然而,多年来一直受到 Delphi 教材内容松散、应用性能差等问题的困扰。为了使读者能够尽快掌握利用 Delphi 开发应用程序的方法,笔者提出了“树式”教学模式思想,即把学生要学的课程看成是解决某一较复杂的实际问题,并把这一问题形象地称为“树”。此模式思想是:把用于解决问题的基础知识看成“树根”;为解决问题,将较复杂问题分解成多个相对较简单的子问题,把这些子问题称为“树枝”;将用于解决子问题的方法或知识看成为“树叶”;将已解决的子问题结合到一起并最终解决较复杂实际问题的部分称为“树干”。这样就把所要解决的较复杂的实际问题看成了一棵“树”。笔者认为,教师向学生传授的是如何把复杂问题分解成一棵“问题树”、然后从最容易解决的“树叶”入手逐步解决问题的方法。笔者在这种思想指导下对计算机教学探索了多年,总想写一本书来阐述这种思想,从萌发这一想法至今已近四年了,一直时断时续地写作,虽然其间对书中内容进行了多次较大规模的修改,但仍感与自己的初始愿望有较大的差距,现在想来,写出一本能让自己满意的图书确实不像开始所想的那样容易。

本书结构采用了前面所说的“树式”教学思想模式。第 1 章和第 2 章是基础,主要讲解 Delphi 语言的工作环境和面向对象的 Pascal 语言,可以将它看成“树根”,若读者对 Delphi 的工作环境和面向对象的 Pascal 语言比较熟悉,可跳过此部分。第 3 章是“树干”,主要向读者讲解 Delphi 中的项目文件的建立、设置以及如何将解决子问题的各个程序模块结合起来构成应用系统的方法。第 4 章主要讲解如何将复杂问题分解成子问题的方法,此章讲述了如何根据需求来调查、分析问题,并最终建立解决问题的模型,同时讲解如何根据问题模型建立系统使用的数据库。第 5 章以及第 5 章以后所讲的内容就是为解决子问题而准备的,本书将开发应用程序需要解决的问题分为以下几个部分:窗体设计(第 5 章)、菜单设计(第 6 章)、图像与文字处理(第 7 章)、开发数据库应用程序(第 8 章)、图表与打印(第 9 章)、多层数据库应用程序设计(第 10 章)和 Delphi 在 MS Office 软件上的应用(第 11 章)。解决了上述问题后,读者再通过第 3 章中所学到的方法,就可以很容易地构建出完整的应用软件系统,这就是本书写作的灵魂。

为了方便读者学习,本书将习题安排在各相应章节之后,它们是笔者在实际开发应用程序时遇到的一些问题,有些还有相当的难度,请读者在学习时认真编写程序代码并调试完成。

本书在编写过程中力求从实际应用出发,通过大量实例向读者介绍 Delphi 开发应用程序的全过程,可以说,本书是笔者多年教学与研究的结晶,有些内容在同类书籍中较少涉及,有些则是笔者从实际出发总结出来的基本规则,如窗体布局和菜单设计准则等。特别是“第 9 章图表与打印程序设计”,该章介绍了由本书著者研究开发的打印组件的使用方法,读者通过此章的学习,可以基本上解决应用程序中数据表的打印问题。

由于 Delphi 以 Pascal 为基础语言,因此,读者在学习本书之前最好有一些 Pascal 语言基础,如果确实没有学过 Pascal 语言,请读者仔细阅读本书第 2 章,即使读者已经学过 Pascal 语言,从第 2 章中也能学到 Pascal 语言课程中很少涉及的内容。

经审定,本书可作为大中专学校教材或教师参考书,同时也适合开发 Windows 应用程序的程序员使用。

曾德藩和陈增昌等老师参加了本书的校对工作,笔者在此表示衷心感谢!

由于时间仓促,加之作者水平有限,书中错误和疏漏之处在所难免,敬请广大读者不吝批评指正,以便不断修订完善。

李目海

2005 年 8 月

E-mail:LEMAIL236@126.COM

目 录

第 1 章 Delphi 可视化开发环境	(1)
1.1 Delphi 简介	(1)
1.2 Delphi 7 的集成开发环境(IDE)	(2)
1.3 创建简单应用程序	(22)
第 2 章 面向对象的 Pascal 语言	(26)
2.1 注 释	(26)
2.2 运算符	(27)
2.3 Object Pascal 的数据类型	(30)
2.4 强制类型转换和类型约定	(48)
2.5 常 量	(49)
2.6 变 量	(52)
2.7 Object Pascal 中的过程与函数	(53)
2.8 字符串资源	(59)
2.9 条件判断	(61)
2.10 循 环	(62)
2.11 作用域	(67)
2.12 单元(Unit)	(68)
2.13 包(Packages)	(71)
2.14 面向对象编程(Object Oriental Program)	(72)
第 3 章 应用程序的框架和设计	(86)
3.1 应用程序的基本结构	(86)
3.2 开发应用程序前的系统设置	(87)
3.3 项目文件	(90)
3.4 项目文件的管理与优化	(98)
3.5 编译并运行项目	(99)
3.6 项目中的 Application 对象	(100)
3.7 项目中的 Screen 对象	(108)
3.8 程序设计中的常用基本技巧	(108)
第 4 章 应用项目的系统分析与数据库设计	(121)
4.1 深入实际调查	(122)

4.2	分析问题的复杂性,确定解决方案	(127)
4.3	分析数据,确定数据表结构	(130)
4.4	建立数据表	(131)
第 5 章	窗体设计	(139)
5.1	窗体分类	(139)
5.2	窗体显示	(140)
5.3	设计 MDI 程序	(149)
5.4	窗体继承	(153)
5.5	提示窗口的设计	(155)
5.6	窗体设计的基本技巧	(167)
5.7	窗体的布局	(171)
第 6 章	菜单设计	(181)
6.1	菜单设计的一般原则	(181)
6.2	菜单的种类	(182)
6.3	下拉菜单设计	(182)
6.4	动态菜单设计	(189)
6.5	弹出菜单	(194)
6.6	主菜单合并	(195)
6.7	按钮菜单	(196)
6.8	多页标签菜单	(198)
6.9	工具栏组件(ToolBar)	(200)
第 7 章	图像与文字处理	(203)
7.1	画布的属性(Canvas)	(203)
7.2	画布的方法	(216)
7.3	坐标系统和映射模式	(231)
第 8 章	开发 BDE 数据库应用程序	(237)
8.1	使用 BDE 连接数据库	(237)
8.2	BDE 组件简介	(243)
8.3	Table 组件	(245)
8.4	Query 组件	(279)
8.5	使用查询表组件动态创建数据表	(284)
第 9 章	图表与报表	(288)
9.1	图表组件(Chart)	(288)
9.2	打印数据表	(297)

第 10 章 多层数据库应用系统设计	(320)
10.1 客户/服务器结构	(320)
10.2 建立数据库	(322)
10.3 编写应用程序服务器程序	(349)
10.4 建立客户端程序	(353)
10.5 修改联合查询表	(355)
10.6 撤销修改	(359)
10.7 应用程序服务器运行客户端 SQL 语句	(365)
10.8 设置主 / 细表	(367)
10.9 设置用户权限	(370)
10.10 使用代码建立数据表	(376)
第 11 章 Delphi 在 Word 及 Excel 中的应用	(380)
11.1 Delphi 对 MS Office 系列软件的支持	(380)
11.2 在 Word 软件上的应用	(381)
11.3 在 Excel 中的应用	(395)

第 1 章 Delphi 可视化开发环境

本章重点：

- Delphi 的特点
- Delphi 的集成开发环境

本章主要介绍了 Delphi 的特点，对 Delphi 的集成开发环境作了较为详细的讲解，读者只有掌握并熟练应用 Delphi 集成开发环境，才能高效率地开发应用程序。此外，本章还介绍了一些一般程序员很少了解的功能。因此认真阅读本章内容，将有助于读者今后快速编写高效的程序代码。

1.1 Delphi 简介

Delphi 自从 1995 年发布以来，以其强大的开发功能、快捷方便的开发手段和高效稳定的程序代码已越来越受到程序员的青睐，并逐渐成为开发 Windows 和 Linux 操作系统下的主流产品，目前有一句流行的话叫“真正的程序员学 C，聪明的程序员学 Delphi”，从一个侧面反映了 Delphi 语言的特点。

那么是什么使得 Delphi 如此优秀？是什么使得众多程序员愿意选择 Delphi 语言？由 Delphi 下列五个特点可以得到答案。

1.1.1 可视化的开发环境

支持可视化编程是 Delphi 之所以快速流行的关键，与目前可视化开发工具一样，Delphi 的可视开发环境也是由相互协同工作的编辑器、调试器和窗体设计器三部分组成。用户可在窗体上设计应用程序的界面，由于窗体设计器“所见即所得”的设计特点，使程序员开发应用程序的难度大大降低；在设计窗体时，Delphi 还具有能根据窗体上设置的组件情况自动生成相关代码的功能。在编辑器窗口中，不仅可以输入程序代码，而且能自由设置调试程序的断点和监控点，在编写程序代码时，Delphi 还提供了能提高编程速度的 Code Insight 技术。Delphi 语言的这些优良特性极大地提高了开发应用程序的效率，大大减少了手工输入及设计的麻烦。

1.1.2 快捷高效的编译器

众所周知，Pascal 编译器最著名的特点就是编译速度快，而 Delphi 正是建立在这种编译器基础之上的高级编程语言，这使得它成为 Windows 操作系统上最快的高级语言编译器。

1.1.3 编程语言规范，易于调试

Delphi 使用 Object Pascal 语言作为基础语言，而 Object Pascal 语言是目前最规范、最严谨的语言之一，这些特征使得编写 Delphi 程序结构清晰，从而使程序更加容易调试。另外，它还采取限制某些可能容易被滥用的面向对象的标准概念，如限制使用对象的多重继承等，虽然面向对象的有些概念加了某些限制，但却大大降低了用户编程的随意性，进而大大降低了程序的调试难度。

1.1.4 采用灵活的数据库结构

虽然 Delphi 不是专业的数据库产品，但其强大的数据库应用程序的开发能力，使得程序员一旦使用，就爱不释手。由于 Delphi 没有自己的数据库，因此使得 Delphi 具有使用几乎所有其他数据库的能力，这是目前其他开发语言所不能具备的。Delphi 不仅可使用 Paradox、FoxPro、Access 等小型桌面数据库，而且还支持 Oracle、MS SQL Server、Sybase、InterBase 和 Informix 等大型数据库开发数据库应用程序，Delphi 提供了多种数据开发解决方案和专业组件，若用户不满意使用 BDE 组件，可使用 ADO、dbExpress 或 InterBase 等组件来开发数据库产品。此外，Delphi 提出的 MIDAS 技术使开发多层数据库应用程序更加方便，这一特点，使得在其他平台上开发数据库产品的程序员可以毫无障碍地转移到 Delphi 开发平台上来。

1.1.5 组件功能的灵活扩充

组件功能是否具有灵活扩充功能是一项常被其他软件设计工具忽略了的重要功能。由于可视组件库 VCL (Visual Component Library) 是 Delphi 的核心，因此，组件是否具有使用灵活、简单、易于扩充等特点就决定了开发工具的效率，在使用 Delphi 开发应用程序中，开发人员可根据 Delphi 提供的对象继承方法较容易地实现组件功能的再扩充，甚至开发出适用于自己的专用组件，相比之下，其他基于组件的开发语言却因为结构过于死板或过于复杂，很难实现组件的灵活扩充功能。

除了上述特点外，Delphi 还在文档管理、第三方组件支持等方面都有很优良的特性，而且使用 Delphi 越久，越能体会到它的功能强大。

1.2 Delphi 7 的集成开发环境 (IDE)

Delphi 7 的集成开发环境是快速开发应用程序最方便的工具，它可以让你用最少的代码编写出运行效率高、性能稳定的 Windows 应用程序，Delphi 所有的应用程序快速开发工具都集成在如图 1-1 所示的 Delphi 的 IDE (Integrated Development Environment) 内。按照功能来分，Delphi 7 的集成开发环境由九个部分组成，分别为：主菜单 (Main Menu)、工具栏 (Tool Bar)、组件面板 (Component Palette)、窗体设计器 (Form)、对象观察器 (Object Inspector)、代码编辑器 (Code Editor)、代码浏览器 (Code Explorer)、对象树视图 (Object Tree View) 和调试器 (Debugger)。

下面分别对这些部分详加说明：

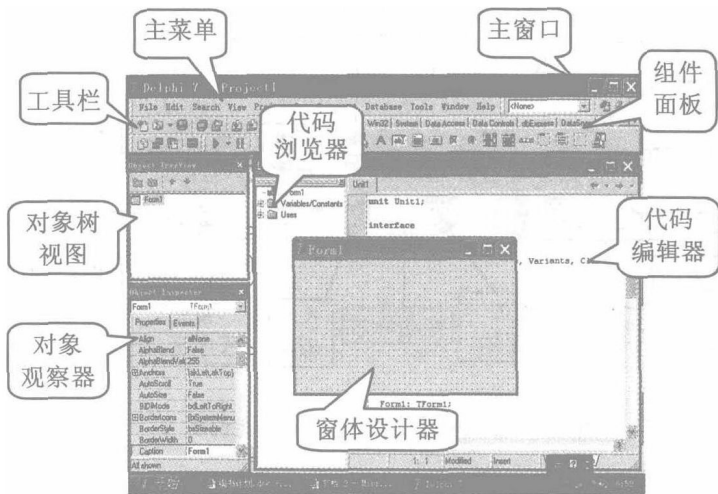


图 1-1 Delphi 7 的集成开发环境

1.2.1 主菜单 (Main Menu)

Delphi 7 中的主菜单继承了前几个版本的优点，并采用 Windows XP 菜单技术，在界面外观上更加美观，使用也更加方便，它几乎集成了 Delphi 中的所有操作功能，用户只要单击指定的主菜单，系统就会弹出如图 1-2 所示的下拉功能菜单，然后用鼠标选择所需功能即可。



图 1-2 主菜单的下拉菜单

1.2.2 工具栏 (Tools Bar)

为了提高用户程序开发的速度和效率，Delphi 还提供了功能较为齐全的工具栏（如图 1-3 所示），这些工具栏上提供的功能与主菜单中的相应功能是一致的，用户只要单击工具栏上相应按钮，即可完成菜单上的相应功能，因此使用工具栏比使用菜单更加快捷方便。

图 1-3 中显示的快捷按钮是 Delphi 7 的默认方式，也可以通过选择的方法设置工具栏。设置方法是：用鼠标右击工具栏，则出现如图 1-4 所示的弹出式菜单，通过选择，使菜单项前显示对号图标，则表示使用相应的工具栏。

注意：这是一个开关式选择，即选择与取消是交互的，用户一旦取消了某个工具栏的显示，可使用此方法来恢复。

此外，我们还可以通过鼠标“拖动”方式来设置工具栏，例如，拖动图 1-3 中的“拖动标志”，将工具栏拖出，则拖出的工具栏将变成一个浮动窗口，如图 1-5 所示。

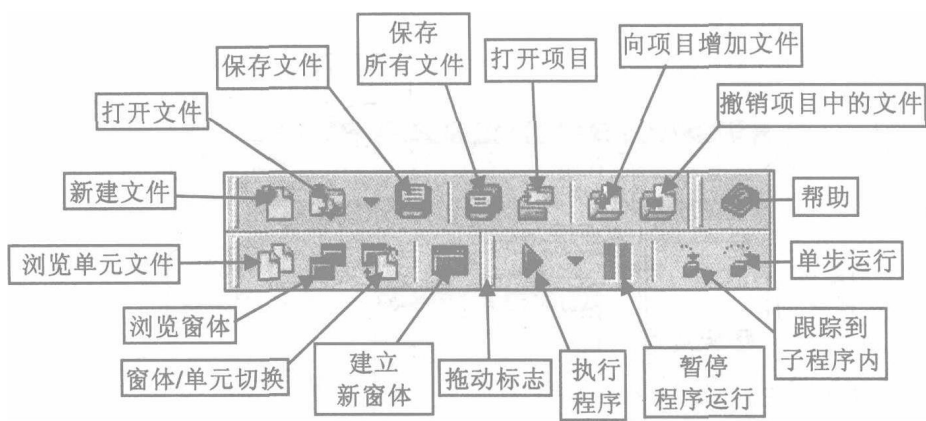


图 1-3 工具栏上的快捷按钮功能说明

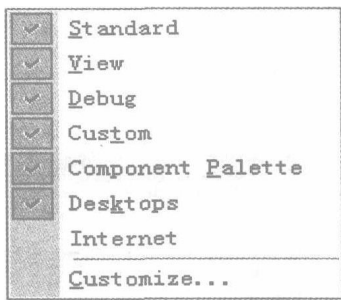


图 1-4 工具栏弹出式菜单



图 1-5 被拖出的工具栏

1.2.3 组件面板 (Component Palette)

组件是构成 Delphi 开发系统的重要组成部分，Delphi 7 有两种标准组件格式：一种格式为可视化组件类型库 VCL (Visual Component Library)，这种格式的组件用来开发运行在 Windows 操作系统上的应用程序，Delphi 5 及以前的版本中的所有组件都属于该格式；另一种格式是跨平台的组件库 CLX (Borland Component Library for Cross Platform)，这是自 Delphi 6 以来新增加的组件库，用于建立可以同时运行在 Windows 和 Linux 操作系统上的应用程序。Borland 公司发布的 Kylix (Delphi 的 Linux 版本) 就是使用 CLX 组件库来开发跨平台应用程序的。

在通常情况下，在设计期就可看到显示效果的组件，我们称之为可视组件，例如我们常用的 Data Controls 面板上的组件；而只能在程序运行时 (又叫运行期) 才能看到显示效果的组件被称为非可视组件，如 Dialogs 和 Data Access 等面板上的组件均为非可视组件。为了以后叙述方便，我们通常把可容纳其他组件的组件称为“容器” (Container)，如我们常用的窗体 (Form) 和 Panel 等就是容器。

Delphi 为我们提供了大量丰富而且实用、方便的组件。为了方便使用，系统将组件按功能分类，每一种类别的组件放置在特定的组件面板上，如图 1-6 所示，组件面板为我们开发

应用程序提供了方便快捷的手段。



图 1-6 组件面板

使用组件面板非常方便，用户只要用鼠标单击面板上的分类标签，将打开该标签下的组件，用鼠标单击所需组件，然后在窗体上再单击一下鼠标左键，则所选组件就自动加入到点击的窗体中，同时与组件相对应的程序代码也自动加入到程序中。这种搭积木式的编程方式，有时用户不用写一程序代码就可以开发出功能强大的 Windows 应用程序。此外，Delphi 也允许用户开发自己的组件并注册到系统中，以进一步加快应用系统的开发，所以，有人说“真正的程序员用 C，聪明的程序员用 Delphi”。

若用户想一次选择多个相同组件，可按住“Shift”键，再用鼠标在组件面板上选择所需的组件，当组件图标周围有一蓝线框时，表示已设置组件为多选状态，用户只要在窗体上多次单击鼠标，就可达到选择多个组件的目的；若用户想取消选择方式，则单击一下组件面板最左边的“箭头”按钮（见图 1-6）即可。

除上述方法外，用户也可通过菜单功能选择组件，具体方法是：选择“View”主菜单下的“Component List”菜单，则显示如图 1-7 所示的组件搜索窗口，用户可通过输入组件名或浏览两种方式查找所需组件，找到并选中后，单击“Add To Form”按钮，则选择的组件就自动增加到当前窗体上。

编程时，需要经常修改组件的属性或事件，而要完成这些操作，必须首先选中组件后再对其修改，那么如何选择窗体上的组件呢？一般来说有两种方法：一种是“单选”，就是用鼠标直接点击所要选的组件，若组件上出现选择框，表示组件选中；另一种是多选，就是按住“Shift”键后，用鼠标连续点击要选的组件。

设计程序窗体时，需要经常移动组件在窗体上的位置，移动窗体上的组件有两种方式：一种是直接拖动，即用鼠标选择组件后直接将其拖动到所需位置处；另一种是细小移动，即所谓的“微调”，方法是：选择好组件后，按住“Ctrl”键再按方向键。

读者可能注意到，当用鼠标移动组件时，每次移动的距离都是网格的倍数，即每次最少移动一个网格（窗体上用虚线构成的网格），用户可通过菜单方式调整网格的大小。方法是：单击主菜单“Tools”→“Environment Options”→“Debugger”（如图 1-8 所示），用户可

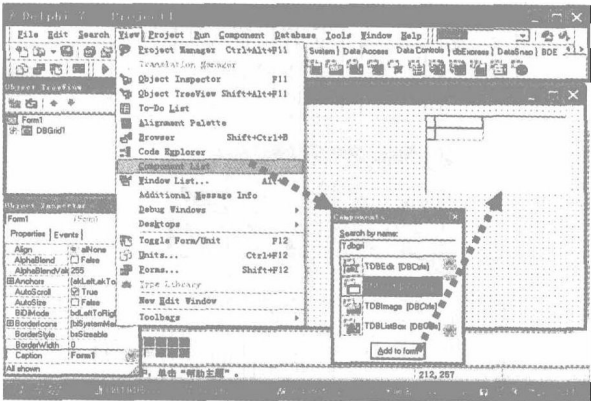


图 1-7 组件搜索窗口

自由调整网格大小。同时，当用户选定“Display Grid”时，窗体上显示网格，否则不显示网格。当用户选定“Snap To Grid”时，则在窗体上移动组件时，移动距离以网格来计算。

在开发较复杂的应用系统时，同一个窗体上常常放置大量的组件，此时，组件排列又成为开发者比较头痛的工作。为此，Delphi 为编程人员提供了功能较齐全的组件自动排列功能。其中主要有两种方法可供用户使用：一种是工具栏方式，另一种是弹出菜单方式。

使用工具栏自动排列组件的方法为：单击主菜单“View”→“Alignment Palette”，则系统显示对齐工具栏窗口（见图 1-9），选择要对齐的组件（可按前面所述用“Shift”键进行多个组件选择），然后，只要单击“对齐”工具栏窗口上相应按钮，即可完成组件的自动排列。

工具栏方式虽然操作方便，但有含义表达不太明瞭的缺点，因此，很多开发人员喜欢使用弹出式菜单方式排列组件。其基本方法是：首先选择要排列的组件，然后单击右键，则出现如图 1-10 所示的弹出菜单，再选择“Position”→“Align”后，则显示对齐设置窗口，设置完成后再次单击“OK”按钮即可实现组件的自动排列功能。

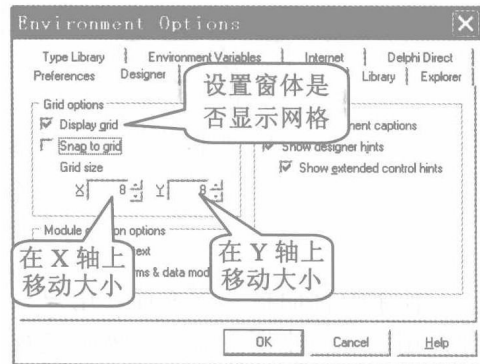


图 1-8 网格设置

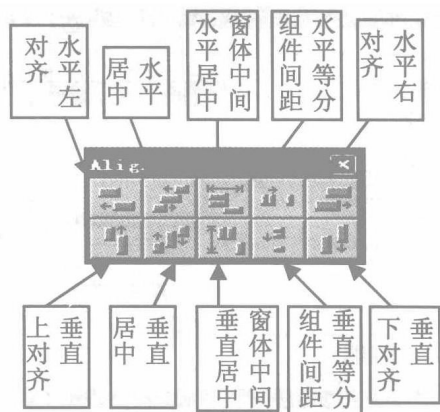


图 1-9 对齐工具栏窗口

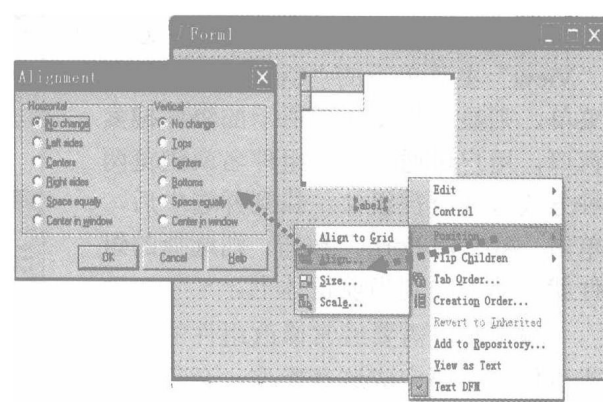


图 1-10 使用弹出菜单对齐组件

1.2.4 窗体设计器（Form Designer）

刚创建窗体时，窗体设计器是一个空白的窗口。有时我们可以把窗体设计器看成是艺术家的画布，在这块画布上用户任意放置 Delphi 中的组件，并设计出丰富多彩的画面，应用程序的用户界面就是由窗体设计器来实现的。

建立窗体的方法主要有两个：一是通过工具栏上的“新建窗体”按钮，二是使用菜单方法。

使用快捷按钮建立：用鼠标在如图 1-11 所示的工具栏的“新建窗体”按钮上点击一下，则系统就自动建立一个如图 1-12 所示的新窗体。

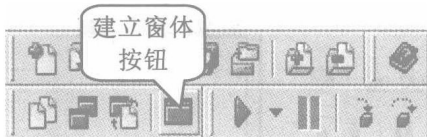


图 1-11 工具栏

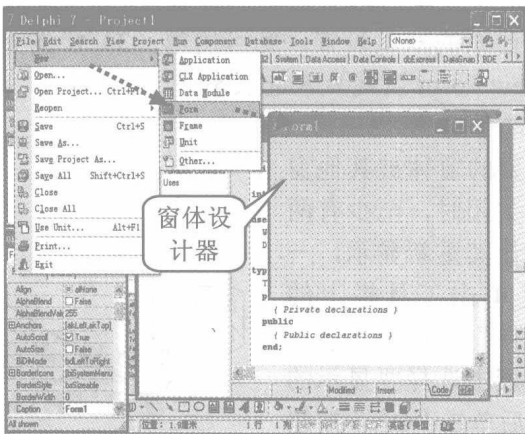


图 1-12 新窗体

使用菜单方法：执行菜单“File” → “New” → “Form”，系统也会自动创建一个如图 1-12 所示的新窗体。

技巧：

● 设置默认窗体

用上述两种方法建立的窗体都是空白的，它们都是 TForm 直接继承过来的默认窗体，有时用户想让系统自动添加一个设计好的标准窗体，而不是空窗体，即将用户设计的窗体设置为默认窗体。如何设置呢？本小节给出以下方法：

- (1) 建立要设置为默认窗体的窗体。
- (2) 将设计好的窗体添加到对象库中。

在设计好的窗体上单击鼠标右键，从下拉菜单中选择“Add To Repository”功能项，则显示如图 1-13 所示的窗口，输入“MyForm”标题信息，并设置 Pages 的值为“Forms”，然后单击“OK”按钮。

- (3) 设置用户默认窗体。

选择菜单“Tools” → “Repository”，则显示如图 1-14 所示窗口，在“Pages”列表中选择“Forms”，然后在“Objects”列表中选择用户定义的窗体对象“MyForm”，再选中“New Form”，再后单击“OK”，完成设置。

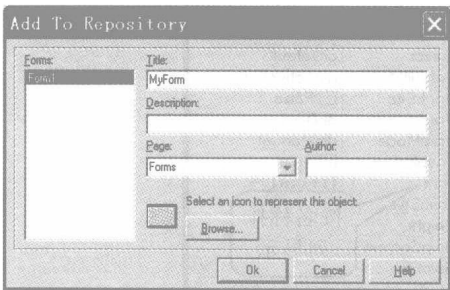


图 1-13 加入窗体到对象库

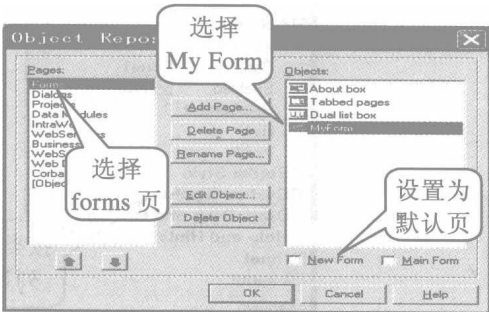


图 1-14 设置默认窗体

完成上述三步后再建立一个新窗体，将会发现，新窗体具有与默认窗体完全一样的界面。若想去掉已做的默认设置，则可用第（3）步的方法，在“Objects”列表中删除设置的对象名或将 New Form 设置为未选状态。

● 选择窗体

当需要的窗体被其他窗体掩盖或关闭时，至少有两种方法可让其重新显示，一种方法是单击工具栏上的“浏览窗体”按钮，在显示窗体的列表中选择要显示的窗体，然后单击“OK”按钮确认即可，如图 1-15 所示。

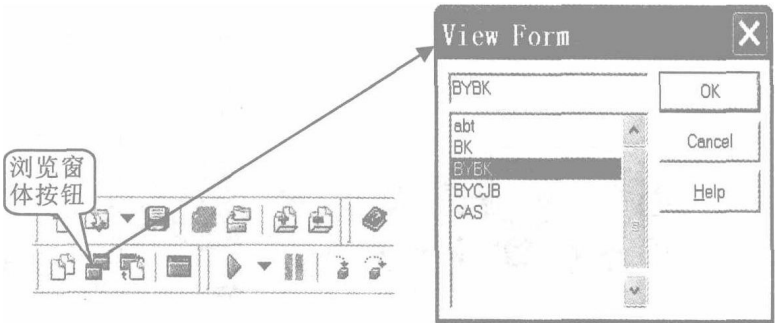


图 1-15 使用工具栏上的“浏览窗体”按钮显示窗体

另一种方法是直接按“F12”键，进入代码编辑窗口，然后选择要显示窗体对应的单元文件，再按“F12”键，即可完成所需窗体的显示（注意：“F12”功能键是单元程序与窗体显示的开关键）。

1.2.5 对象观察器（Object Inspector）

利用 Object Inspector，可以修改窗体或组件的属性（Property），或者使它们能够响应不同的事件（Event）。这里的属性指的是描述窗体或组件特性的数据，如高度、颜色、字体等，它们决定了组件在屏幕上的大小、颜色等特征。事件则是一种消息处理机制，它能够捕捉某种情况的发生并作出反应，如鼠标单击和窗口重画就是两种典型的事件。

Object Inspector 设置窗口类似一个带标签的多页笔记本，它包括属性页（Properties）和事件页（Events），如图 1-16 所示，在设置属性或事件时，可用鼠标单击“属性”或“事件”标签进行切换。需要注意的是，在 Object Inspector 中显示的是在窗体设计器中选定组件的属性和事件，也就是说，若想修改窗体设计器中组件的属性或事件，则必须首先选择它，然后再到 Object Inspector 中修改。

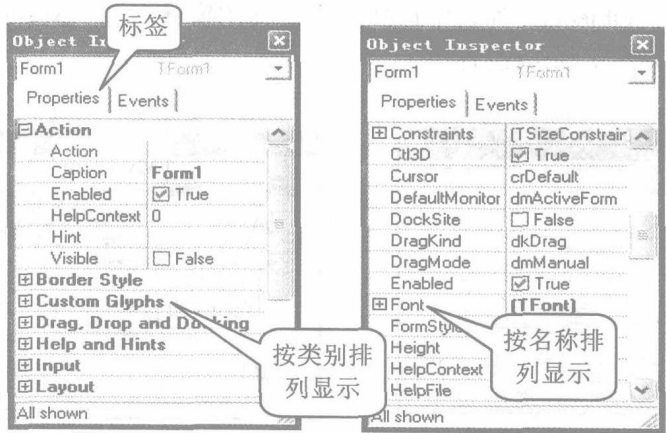


图 1-16 Object Inspector 两种排列方式显示的对象观察器