



21 世纪高职高专信息技术教材

Delphi 程序设计教程

北京希望电子出版社 总策划

刘 毅 编 著



科学出版社

www.sciencep.com



21 世纪高职高专信息技术教材

Delphi 程序设计教程

北京希望电子出版社 总策划

刘 毅 编 著

 科学出版社
www.sciencep.com

内 容 简 介

本书是21世纪高职高专信息技术教材中的一本,本书力求使读者能够快速、轻松地学会用Delphi编程,能充分理解面向对象程序设计的内涵,能用本书提供的实例解决实际问题。

本书以Delphi 6为开发环境,主要介绍了构成Delphi应用程序的多种基本元素和必备知识,包括Delphi的可视化开发环境、Object Pascal语言、窗体和各种功能的组件、面向对象程序设计的基本概念、MDI、图形图像处理、目录和文件系统、数据库编程、多媒体开发和Internet编程。

本书的主要特色为:内容新颖,理论与实践并重,系统全面,图文并茂,通俗易懂。

本书可作为高等职业院校、高等专科学校、成人高等学校、软件职业技术学院专业学生学习用书,也可以供继续教育学校以及程序爱好者使用。

本书相关程序代码,请到 www.b-xr.com 免费下载。

需要本书或技术支持的读者,请与北京中关村 083 信箱(邮编:100080)发行部联系,电话:010-62528991, 62524940, 62521921, 62521724, 82610344, 62978181(总机)传真:010-62520573, E-mail: yanmc@bhp.com.cn。

图书在版编目(CIP)数据

Delphi 程序设计教程/刘毅 编著. —北京:科学出版社, 2004.8

21 世纪高职高专信息技术教材

ISBN 7-03-013260-2

I. D... II. 刘... III. Delphi 语言—程序设计—高等院校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 030921 号

责任编辑:王玉玲

/ 责任校对:佳宜

责任印刷:媛明

/ 封面设计:梁运丽

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>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2004 年 8 月第 一 版

开本:787×1092 1/16

2004 年 8 月第一次印刷

印张:18 7/8

印数:1-5000 册

字数:413 000

定价:22.00 元

21世纪高职高专信息技术教材编委会名单

(排名不分先后)

主 任 高 林 教授 (联合大学)

副主任 谢玉声 教授 (钟山学院) 袁启昌 教授 (钟山学院)

胡伏湘 副教授 (长沙民政学院) 陆卫民

委 员

阮东波 (宁波高专)	徐 萍 (钟山学院)
周文革 (宁波高专)	李 淼 (重庆工业职业技术学院)
程 刚 (宁波高专)	李超燕 (宁波职业技术学院)
连晋平 (广东肇庆学院)	景鹏森 (钟山学院)
唐伟奇 (长沙民政学院)	陈孟建 (杭州经贸学院)
慕东周 (徐州工职学院)	宗小羽 (钟山学院)
朱作付 (徐州工职学院)	韦 伟 (钟山学院)
杨旭东 (徐州工职学院)	曹冬梅 (钟山学院)
米 昶 (青岛高职)	杨章静 (钟山学院)
孙 杰 (青岛高职)	尹 静 (钟山学院)
唐燕青 (青岛高职)	田 更 (青岛大学)
刘 毅 (重庆工职学院)	吴 军 (钟山学院)
邱建国 (青岛高职)	杨金龙 (青岛高职)
侯晓华 (重庆工职学院)	崔俊杰 (承德民族职业技术学院)
蒋建强 (苏州经贸学院)	陈翠娥 (长沙民政学院)
王趾成 (河北高职)	陈 春 (四川师范)
龙 超 (永州职业学院)	徐建华
罗映峰 (广东技师学院)	郑明红
冯矢勇 (苏州高职)	韩素华

序

高等职业教育目前已成为我国高等教育的重要组成部分,对于推动我国社会主义现代化建设起着不可忽视的作用。计算机教育在整个高职教育中有着举足轻重的地位,因为计算机的普及已经涉及到各个行业。对于传统的学习计算机知识的方法即理论为主、应用为辅的教学模式,相对高职教育来说有些不太适合,针对这种情况,就需要一些符合高职教育特点的教材来满足这种需求。

为解决教材供需不平衡的矛盾,北京希望电子出版社与全国高等学校计算机基础教育研究会高职高专专业委员会联合组织国内十几所高职院校,聘请“双师”型教师共同编写针对高职特点的教材 30 多种,以及实训类教材 10 多种,并请专家论证了本套教材的体系、风格、结构、内容等方面的可行性与可操作性。该系列教材体现“重在能力素质培养”的目标,结合教育部的教学大纲要求,在实用性、新颖性、可读性几个方面都有所突破。

高职教材建设是教学改革重要的环节,高等职业技术教育专业设置要与劳动力市场需求相结合,教学内容与国家职业标准相衔接。采取“订单教学”的校企合作培养模式,实行学业文凭和职业资格两种证书制度,使一线技术人才培养实现教学与市场“零距离”、毕业生上岗“零适应期”。这种以市场为导向实行的订单教学,能够直接为用人单位培养实用型人才,是一条富有特色的职教之路,可以保证同学们将来在就业和升学两条渠道上有最大的发展空间。所以,高校就要突出应用技能培养的办学特色,按照人才市场供求信号进行学科、专业和教学内容的调整,以适应社会需要。在培养学生的知识、能力、技能方面都要与其他综合性本科院校有所区别。

本系列教材就是遵循这种订单式教学的需要,一方面是设定系统理论知识的教材,这种教材的内容按照“必需、够用”的原则,构筑坚实的具有高职特色的理论体系基础;另一方面是训练职业动手能力的实训教材,按照“切实、实用”的原则,培养动手能力强的人才。以上两种教材相互配合,既可以单独使用,也可以配套使用。

高职教材建设还在探索中,如何能满足企业对人才的需求,跟上时代发展的步伐,这些都是亟需解决的问题。本丛书旨在抛砖引玉,希望更多的优秀教师参与到教材建设中来,真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议,为下一次的修订与改版做准备,使本丛书日臻完美。

若有投稿或建议,请发至本丛书出版者电子邮件: hansuhua@163bj.com

21 世纪高职高专信息技术教材编委会

前 言

Delphi被称为第四代编程语言,它以基于窗口和面向对象的编程方法与Windows操作系统紧密地结合,它具有强大的数据库技术支持、迅捷的编译速度,同时兼备Visual C++强大的功能和Visual Basic易学灵活的特点,是程序员们至爱的编程工具。现在广泛流传的一句话:“真正的程序员用C,聪明的程序员用Delphi”就可以说明这一点。

Delphi继承了Pascal语言的严谨结构和优雅风格,以开放式的环境,完全地支持对象程序设计等诸多特性,成为Windows开发工具中首选的教学环境。

本书是Delphi程序设计的入门书,适用于有一定编程语言基础的读者。

本书以Delphi 6为开发环境,主要介绍了构成Delphi应用程序的多种基本元素和必备知识,包括Delphi的可视化开发环境、Object Pascal语言、窗体和各种功能的组件、面向对象程序设计的基本概念、MDI、图形图像处理、目录和文件系统、数据库编程、多媒体开发和Internet编程。

本书力求使读者能够快速、轻松地学会用Delphi编程,能充分理解面向对象程序设计的内涵,能用本书提供的实例解决实际问题。

根据应用型人才的培养目标,本书注重理论和实际相结合,注重基础知识的理解与基本技能的培养。在理论“必需、够用为度”的前提下,加强实用性,加强实践性环节。为此,我们几乎每一章都有方法与技巧及实例。我们每一章都配有习题,而且结合作者多年的开发经验和教学经验,编写了一个图书馆管理信息系统。

本书由刘毅主编,刘妮娅、俞正跃参编。其中,刘毅编写了第4、5、6、7、8、9、10、11章,刘妮娅编写了第2、3章,俞正跃编写了第1章。全书由刘毅制定编写大纲并负责统稿工作。李霞老师在百忙之中抽出宝贵时间认真细致地审阅了全书,刘欣在本书的实例编写当中提供了一定的技术支持,刘作勋、熊渝萍、李洪波在本书的校对中给予了极大的帮助,并提出了许多宝贵意见,作者深表感谢。由于作者水平有限,时间紧迫,书中难免有疏漏之处、恳请广大读者批评指正,不胜感激。

作者E-mail地址: lyi61@vip.sina.com

编 者

目 录

第 1 部分 Delphi 编程基础	1	2.3.1 算术运算符	30
第 1 章 Delphi 集成开发环境	2	2.3.2 逻辑运算符	30
1.1 Delphi 简介	2	2.3.3 位运算符	30
1.1.1 Delphi 的历史	2	2.3.4 字符串运算符	31
1.1.2 Delphi 中的有关概念	2	2.3.5 指针运算符	31
1.2 Delphi 的集成开发环境	3	2.3.6 集合运算符	31
1.2.1 主窗口	4	2.3.7 关系运算符	32
1.2.2 窗体设计器	11	2.3.8 类运算符	32
1.2.3 Object Inspector (对象查看器)	12	2.3.9 @运算符	32
1.2.4 代码编辑器	12	2.4 Object Pascal 语言的语句	32
1.2.5 代码浏览器	13	2.4.1 声明语句	32
1.2.6 工程管理器	13	2.4.2 赋值语句	33
1.2.7 创建一个简单的应用程序	13	2.4.3 复合语句	33
1.3 开发环境的优化	16	2.4.4 goto 语句	34
1.3.1 自定义工具栏	16	2.4.5 with 语句	34
1.3.2 编程环境设置	17	2.4.6 分支语句	34
1.4 方法与技巧	18	2.4.7 循环语句	36
1.4.1 使用代码浏览器	18	2.5 过程与函数	38
1.4.2 使用帮助系统	18	2.5.1 过程的定义与调用	38
1.4.3 设置 IDE 桌面	19	2.5.2 函数的定义与调用	39
1.5 习题	19	2.5.3 参数的传递	40
第 2 章 Object Pascal 语言简介	20	2.5.4 规范化命名	42
2.1 Object Pascal 语言基础	20	2.6 Object Pascal 语言的面向对象技术 ..	43
2.1.1 保留字、指令字和标识符	20	2.6.1 对象和类的概念	43
2.1.2 注释	21	2.6.2 Object Pascal 中类的定义	45
2.1.3 常量与变量	22	2.6.3 多态性	49
2.2 Object Pascal 语言的数据类型	23	2.6.4 类运算符	50
2.2.1 简单数据类型	23	2.7 单元文件的结构	50
2.2.2 字符串类型	25	2.7.1 标准的单元文件结构	50
2.2.3 结构类型	26	2.7.2 单元文件代码分析	51
2.2.4 指针类型	28	2.7.3 循环单元引用	53
2.2.5 过程与函数类型 (procedural)	29	2.8 异常与异常处理	53
2.2.6 可变类型 (variant)	29	2.8.1 try...except 语句	54
2.3 Object Pascal 语言的运算符	29	2.8.2 try...finally 语句	54
		2.8.3 raise 语句	56
		2.9 方法和技巧	56

2.9.1	命名规则.....	56	3.6.2	RadioGroup 组件.....	77
2.9.2	设置代码模板.....	57	3.6.3	Panel 组件.....	77
2.9.3	设置提示信息.....	57	3.6.4	ScrollBar 组件.....	77
2.9.4	获取和使用命令行参数.....	58	3.6.5	PageControl 组件.....	78
2.10	习题.....	58	3.7	菜单的使用.....	78
第 2 部分 Delphi 编程的核心技能.....		59	3.7.1	菜单的使用.....	78
第 3 章 常见组件编程.....		60	3.7.2	主菜单.....	79
3.1	窗体和组件.....	60	3.7.3	弹出式菜单.....	80
3.1.1	概述.....	60	3.8	工具栏、状态栏的使用.....	81
3.1.2	窗体.....	61	3.8.1	CoolBar 组件.....	81
3.1.3	组件(Component).....	63	3.8.2	ToolBar 组件.....	82
3.1.4	在窗体中使用组件.....	64	3.8.3	ControlBar 组件.....	82
3.2	文本组件的使用.....	67	3.8.4	状态栏的使用.....	83
3.2.1	Edit 组件.....	68	3.9	编程实例.....	83
3.2.2	Label 组件.....	68	小型计算器实例.....		83
3.2.3	Memo 组件.....	69	3.10	方法与技巧.....	87
3.2.4	MaskEdit 组件.....	69	3.10.1	窗体、组件的使用原则.....	87
3.2.5	RichEdit.....	70	3.10.2	使用信息对话框.....	88
3.2.6	几点说明.....	70	3.11	习题.....	92
3.3	按钮组件的使用.....	70	第 4 章 键盘和鼠标.....		94
3.3.1	Button 组件.....	71	4.1	事件概述.....	94
3.3.2	BitBtn 组件.....	71	4.2	键盘事件.....	95
3.3.3	SpeedButton 组件.....	72	常用的键盘事件.....		95
3.3.4	CheckBox 组件.....	72	4.3	鼠标事件.....	96
3.3.5	RadioButton 组件.....	72	4.3.1	常用鼠标事件.....	96
3.4	列表类组件的使用.....	73	4.3.2	拖曳事件.....	96
3.4.1	ListBox 组件.....	73	4.4	方法与技巧.....	98
3.4.2	ComboBox 组件.....	73	4.4.1	模拟按下键盘上的某个键.....	98
3.4.3	CheckListBox 组件.....	74	4.4.2	检测功能键.....	99
3.4.4	ListView 组件和 TreeView 组件.....	74	4.4.3	限制鼠标移动的范围.....	100
3.4.5	ImageList 组件.....	76	4.5	习题.....	101
3.4.6	DateTimePicker 组件.....	76	第 5 章 文件、目录和驱动器.....		102
3.5	表格类组件的使用.....	76	5.1	文件.....	102
3.5.1	DrawGrid 组件.....	76	5.2	文件对话框组件.....	103
3.5.2	StringGrid 组件.....	76	5.3	Win3.1 相关组件.....	104
3.5.3	DBGrid 组件.....	77	5.4	常用函数.....	104
3.6	分类组件的使用.....	77	5.4.1	文件操作常用函数.....	104
3.6.1	GroupBox 组件.....	77	5.4.2	目录操作常用函数.....	110
			5.4.3	驱动器操作常用函数.....	112

5.4.4	文件名操作常用函数	113
5.5	例子程序	114
5.5.1	文件操作	114
5.5.2	菜单加工具栏的文字 编辑器	116
5.6	方法与技巧	119
5.6.1	获取驱动器类型信息	119
5.6.2	将文件删除到回收站中	120
5.7	习题	121
第 6 章	打印	122
6.1	TPrinter 对象	122
6.2	打印操作常用函数	124
6.3	打印操作	125
6.3.1	打印文本	125
6.3.2	打印位图	126
6.3.3	打印 TMemo 组件中的内容	127
6.3.4	打印 RTF 格式的文本	128
6.4	打印技巧	128
6.4.1	获取显示当前打印机的 分辨率	128
6.4.2	用打印机的点数做度量 单位	129
6.4.3	将打印结果直接送到 打印机	129
6.4.4	获取默认打印机的信息	129
6.4.5	打印窗体	130
6.5	习题	130
第 3 部分	Delphi 高级编程技术	131
第 7 章	Delphi 数据库基础	132
7.1	关系数据库	132
7.1.1	关系数据库	132
7.1.2	关系数据库的组成	132
7.1.3	关键字	133
7.1.4	表之间的约束	133
7.2	SQL 语言	134
7.2.1	数据库标准命令	134
7.2.2	SELECT 命令详解	137
7.3	Delphi 的数据库开发工具	142
7.3.1	使用 SQL Explorer	142

7.3.2	使用 BDE Administrator	143
7.3.3	使用 Database Desktop	144
7.4	习题	146
第 8 章	开发桌面型数据库应用程序	147
8.1	数据访问组件简介	148
8.1.1	TTable 组件介绍	149
8.1.2	TQuery 组件介绍	152
8.1.3	TQuery 组件和 TTable 组件 的比较	154
8.1.4	TStoredProc 组件介绍	154
8.1.5	TDataSource 组件介绍	159
8.1.6	TDatabase 组件介绍	162
8.1.7	TSession 组件介绍	167
8.2	数据控制组件简介	168
8.2.1	数据控制组件简介	169
8.2.2	TDBGrid 组件介绍	170
8.2.3	TDBNavigator 组件介绍	171
8.2.4	TDBText 组件介绍	174
8.2.5	TDBEdit 组件介绍	174
8.2.6	TDBMemo 组件介绍	174
8.2.7	TDBImage 组件介绍	176
8.2.8	其他数据库控制组件介绍	176
8.3	数据报表的设计	177
8.3.1	QReport 组件的介绍	178
8.3.2	快速生成报表	186
8.4	程序举例	187
8.4.1	例子一: 从已建数据源中 查询数据	187
8.4.2	用 Table 控件开发数据库 应用程序	190
8.4.3	制作一个报表	193
8.5	习题	199
第 9 章	桌面型数据库应用程序实例	200
9.1	程序实现的功能及总体结构设计	200
9.1.1	应用程序的功能划分	200
9.1.2	应用程序的结构设计	202
9.2	应用程序中数据模块的设计	203
9.3	用户登录功能的实现	203
9.4	图书馆管理系统功能的实现	206

9.4.1 修改图书记录	206	10.6 音频和视频媒体的程序开发	246
9.4.2 修改学生记录	208	10.7 媒体播放器的实现	248
9.4.3 修改密码设置	209	10.8 方法和技巧	252
9.5 图书馆服务系统功能的实现	212	10.8.1 Delphi 实现图象灰度处理 的 3 种方法	252
9.5.1 查询图书、借阅图书及 预约图书	212	10.8.2 Delphi 图形编辑技巧二则 ..	254
9.5.2 查询个人借阅情况及 归还图书	219	10.9 习题	255
9.6 习题	225	第 11 章 Internet 程序开发	257
第 10 章 多媒体程序开发	226	11.1 制作 HTML 页面	257
10.1 多媒体	226	11.1.1 HTML 语言格式	257
10.1.1 概述	226	11.1.2 制作静态网页	257
10.1.2 多媒体应用领域	227	11.1.3 制作动态网页	260
10.1.3 多媒体软件开发工具	228	11.2 制作浏览器	261
10.2 Delphi 与多媒体程序设计	230	11.3 文件传输服务程序开发	263
10.2.1 Windows 中的多媒体	230	11.3.1 FTP 与 NMFTP 组件	263
10.2.2 Delphi 中的多媒体组件	231	11.3.2 文件传输服务程序实例	264
10.3 图像格式的处理	232	11.4 Web 服务器端程序开发	266
10.3.1 位图	232	11.4.1 Web 服务器应用 程序的结构	267
10.3.2 JPEG 文件	234	11.4.2 WebDispatcher 组件	267
10.4 特殊显示效果	236	11.4.3 TPageProducer 组件	270
10.4.1 基本原理	236	11.4.4 TDataSetPageProducer 组件 ..	271
10.4.2 调用 BitBlt	236	11.4.5 查看表数据	272
10.4.3 调用 CopyRect	238	11.4.6 TQueryTableProducer 组件	274
10.4.4 Delphi 图形显示技巧	238	11.5 Socket 编程	276
10.5 利用图像控件实现动画效果	244	聊天工具	276
10.5.1 TImage 组件变换法	244	11.6 习题	279
10.5.2 TPanel 组件变换法	245		
10.5.3 Canvas 画面变换法	245		

第一部分

Delphi 编程基础

第一部分将介绍以下内容：

- Delphi 简介
- Delphi 集成开发环境
- Windows 环境下应用程序的开发
- 集成调试器的使用
- 集成开发环境的优化
- Object Pascal 语言

这一部分的内容是 Delphi 编程的基础，如果读者在 Delphi 编程方面已经有了一定经验，可以有选择地阅读感兴趣的章节。

另外，在阅读这一部分的时候，暂时会有一些印象不深的地方。所以建议在学习了后面的内容之后，回过头来再看看这一部分的内容，一定会有所收获。

第 1 章

Delphi 集成开发环境

本章重点内容:

- Delphi 简介
- Delphi 的集成开发环境
- 开发环境的优化

1.1 Delphi 简介

1.1.1 Delphi 的历史

Delphi 是 Borland 公司 (Inprise 公司的前身) 推出的一种可视化的、方便快捷的 Windows 应用程序开发工具。Delphi 的快速流行与它的既有 VC 的强大功能和高效性、同时也具有 VB 编程的方便性是分不开的,这也就决定了使用 Delphi 开发程序的用户数量迅速的增长。

作为一种工具,Delphi 建立在成熟的面向对象语言 Object Pascal 之上。使用 Delphi,您可以做到能用 C++完成的任何事情但避免 C++的大部分麻烦;使用 Delphi,您可以创建 Visual Basic 应用程序,反过来则是做不到的。在 C++中,程序员可能会在内存管理、模板、操作符重载时遇到麻烦,而在 Delphi 中则不会出现这种问题。Delphi 既具有 C++的强大性又具有 Visual Basic 的易用性。

1.1.2 Delphi 中的有关概念

1. 面向对象程序设计

面向对象的程序设计(Object-Oriented Programming,简记为 OOP)是 Delphi 诞生的基础。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。面向对象的编程语言使得复杂的工作条理清晰、编写容易。说它是一场革命,不是对对象本身而言,而是对它们处理工作的能力而言。对象并不与传统程序设计和编程方法兼容,只是部分面向对象反而会使情形更糟。除非整个开发环境都是面向对象的,否则对象产生的好处还没有带来的麻烦多。而 Delphi 是完全面向对象的,这就使得 Delphi 成为一种触手可及的促进软件重用的开发工具,从而具有很强大的吸引力。

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,程序员仍然要编写大量的代码。Delphi 的推出,填补了这项空白。您不必自己建立对象,只要在提供的程序框架中加入完成功能的代码,其余的都交给 Delphi 去做。欲生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁,Delphi 将帮助您轻松地完成。它允许在一个具有真正 OOP 扩展的可视化编程环境中,使用它的 Object Pascal 语言。这种革命性的组合,使得可视化编程与面向对象的开发框架结合起来。

2. 可视化组件库

Delphi 中的所有组件是在可视组件库(Visual Component Library, 缩写为 VCL)中封装的。使用这个 VCL, 程序员可以通过少量的代码就能够创建出复杂的应用程序。

3. Delphi 中的窗体和组件

Delphi 的 VCL 中提供了大量的组件, 每个组件就是一个类, 正是由于这些组件的存在, 才使得 Delphi 成为一个方便快捷的应用程序快速开发工具。

Delphi 中的组件按照分类以标签页的形式在 Delphi 窗体上的组件面板中出现, 这些标签页在 Delphi 窗体上的顺序和编号可能因为 Delphi 的版本和种类以及安装的选项和附加组件而有所不同。

在 Windows 操作系统中, 人-机交互的界面主要是通过一些窗口和对话框实现的。在 Delphi 中, 这些窗口和对话框就是程序设计阶段的窗体, Delphi 的可视化设计工作就是在窗体中进行的。

通常, 窗体中会有一些控件, 通过这些控件可以实现多种多样的功能。在 Delphi 中, 把这些运行期间出现在窗口和对话框中的控件称为可视组件。在窗体中, 不仅可以放置控件, 还可以放置一些运行期间不可视的组件, 这些不可视组件集中地实现了一些特殊的功能。

可视组件(控件)可分为两类: 窗口控件和图形控件。窗口控件是基于系统窗口的可视组件, 具有窗口句柄, 可作为输入焦点并含有其他控件, 例如 Edit 组件; 图形控件没有窗口句柄, 不可作为输入焦点, 也不能含有其他控件, 例如 Label 组件。图形控件比窗口控件使用的资源少, 尤其在控件数量比较多的情况下最好使用图形控件。

窗体和组件的关系非常密切。从视觉的角度来看, 窗体就像一个容器, 其中包容了可视组件和不可视组件。从程序代码的角度来看, 一个窗体对应一个源程序文件, 其中定义了窗体中所有组件的对象。

窗体和组件之间除了具有上述的不同点, 还具有一些相同点。例如, 从面向对象编程的角度来看, 所有组件都是 TComponent 类派生类的一个对象。窗体类(TForm)也是从 TComponent 类派生的, 所以可以将窗体看成一个特殊的组件。

有些组件也具有窗体的“容器”属性, 也就是在这些组件中可以放置其他组件, 通常这些组件称为容器组件。例如在 Label 组件中可以放置 Edit 组件, Label 组件就是一个容器组件。

可视组件库是 Delphi 中非常重要的一部分, 其不仅封装了所有组件对应的类, 还封装了一些通用的类。所有的这些类都是由一个基类派生出来的, 也就是 TObject 类。了解可视组件库对于学习组件、使用 Delphi 很有帮助。

1.2 Delphi 的集成开发环境

Delphi 是 Inprise 公司推出的基于对象 Pascal 语言的可视化集成开发工具。利用 Delphi 编程, 可以快速、高效地开发出基于 Windows 环境的各类程序, 尤其在数据库和网络方面, Delphi 更是一个十分理想的软件开发平台。

通常, Delphi 集成开发环境的主界面如图 1-1 所示。

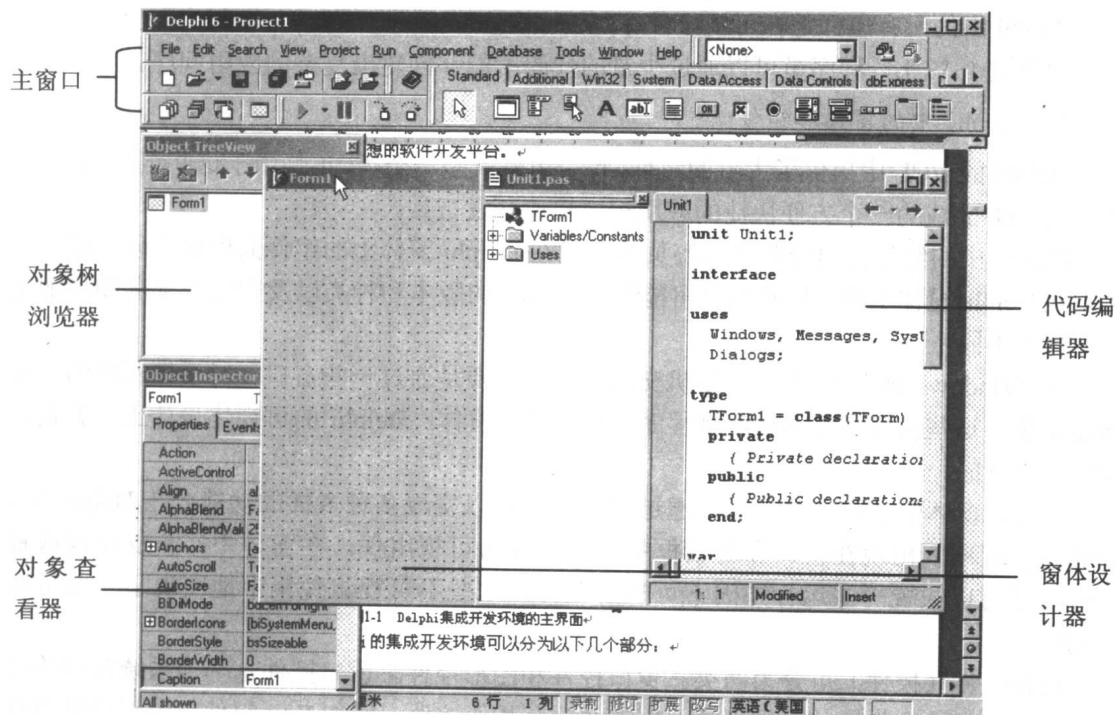


图 1-1 Delphi 集成开发环境的主界面

从图 1-1 中我们可以看到 Delphi 的集成开发环境可以分为以下几个部分:

- 主窗口
- 组件面板
- 工具栏
- 窗体设计器
- 代码编辑器
- 对象观察器(Object Inspector)
- 代码浏览器

1.2.1 主窗口

主窗口(如图 1-2 所示)可以认为是 Delphi IDE 的控制核心,它具有其他 Windows 应用程序的主窗口所具有的一切功能。主窗口主要包括三部分:主菜单、工具栏和组件面板。

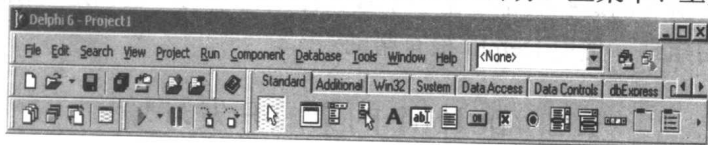


图 1-2 主窗口

1. 主菜单

与其他 Windows 应用程序一样,在 Delphi 中也可以通过主菜单创建、打开或保存文件、调用向导、查看其他窗口、修改选项等。主菜单的每一项都可以通过工具栏上的一个按钮

来实现。

Delphi 的绝大部分操作命令都可以通过菜单栏上的相应菜单项实现, Delphi 的菜单栏在形式上与 Windows 应用程序菜单栏相同。但与其他语言的开发环境(如 VisualC++等)相比, Delphi 在主菜单栏中将 Component(组件)和 Database(数据库)单独作为菜单项列出,可见在 Delphi 中使用组件和数据库程序设计是两个重要的部分,这可在后面的介绍中看到。

下面我们对 Delphi 菜单栏中的一些常用菜单项作一个简单介绍。

(1) File 菜单

1) New 选项

单击 New 选项可以弹出如图 1-3 所示对话框。

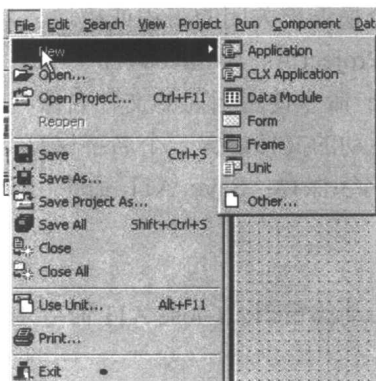


图 1-3 New 子菜单所包含的内容

Delphi 6 的 New 菜单项与 Delphi 5 相比改变比较大,它将以前 File 菜单中的 New Application, New Form 等菜单项作为其子菜单。这些子菜单常用的有

- Application 新建一个包含有单独应用程序的新工程组。
- Form 在工程中新建一个空白窗体和一个新单元。
- Other... 该子菜单项的功能与以前版本中 File 菜单的 New 菜单项相同,单击它会弹出 New Items 对话框,如图 1-4 所示。该对话框中有 12 个标签,不同的标签页包含不同的对象。

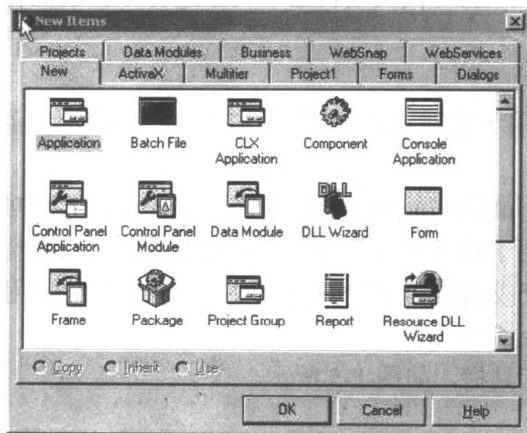


图 1-4 New Items 对话框

其中的 New 栏目当中的 Application 选项和我们刚才讲到的 File→New 中的 Application 功能是一样的。其他内容我们在以后的章节中会作介绍。

2) Open

Open 命令用于打开各种类型的文件，包括工程文件、FM 文件、单元文件或者其他文本文件。我们可以打开多个窗体、单元或文本文件，但在同一个时刻只能打开一个工程文件，当试图打开另一个工程文件时，Delphi 将提示保存当前工程，然后退出当前工程。

3) Open Project

Open Project 命令用于打开一个工程文件。

4) Reopen

Reopen 命令用于打开一个最近被关闭的工程或模块。

5) Save, Save As, Save Project As 与 Save All

用于保存文件。其中 Save 命令使用当前的文件名保存包含在打开的工程中的所有文件；Save As 命令用于将当前活动的文件以另一个名字保存；Save Project As 命令用于将工程文件 (.dpr 文件) 以另一个名字保存；Save All 命令用于保存所有打开的文件，包括当前的工程和模块。

6) Close 与 Close All

Close 命令用于关闭当前活动的窗口。Close All 命令用于关闭整个工程。

7) Use Unit

此命令用于把某个单元文件加入到工程中的当前窗体或单元的uses子句当中。

(2) Edit 菜单

Edit 菜单中是一些编辑命令，如 Redo, Cut, paste 等。这些操作的使用与 Windows 其他软件的相应操作相同，因而不作介绍。但它们不仅对代码编辑器中的文本有效，对窗体上的组件也同样有效。

在窗体设计器上操作组件时，最方便的是单击鼠标右键，在弹出的快捷菜单中选择相应的命令，所有与窗体设计器有关的快捷菜单在 Edit 菜单中都能找到。

(3) Search 菜单

Search 菜单中包含了一些对代码编辑器中文本进行定位的操作，例如：查找、替换和转移到某一行等操作。

(4) View 菜单

View 菜单中的命令用于管理 IDE 中的窗口的显示，决定显示哪些窗口和不显示哪些窗口。

1) Project Manager, object Inspector 和 Object TreeView 菜单项

这 3 个菜单项分别用于打开工程管理器、对象查看器和对象树浏览器。如果分别打开这 3 个菜单项，它们就变成当前活动的窗口。这 3 个管理器的使用见后面的介绍。

2) Alignment Palette 命令

此命令将打开排列工具栏，如图 1-5 所示。



图 1-5 排列工具栏

3) Browser 命令

此命令将打开类观察器。如果这个命令变灰，可能是程序尚未被编译。

4) Component List 命令

该命令用于打开 Components 窗口，以列表的形式显示组件面板上的所有组件，如图 1-6 所示。

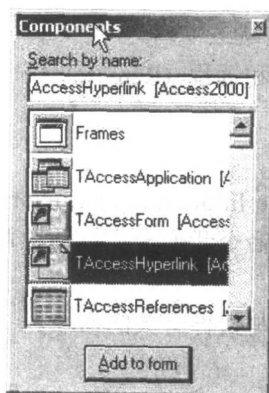


图 1-6 Components 窗口

在列表中选择一个组件，单击 Add to form 按钮将把所选的组件添加到窗体中，并且是在窗体的中心位置。如果事先在窗体中选择一个容器类组件 (如 TPanel 等)，则 Delphi 将把在 Components 窗口中选择的组件添加到这个容器类组件的中心位置。

5) Window List 命令

该命令用于显示 Window List 窗口，列出当前打开的所有窗口，如图 1-7 所示。利用它，用户可以方便地浏览整个 IDE，显示每一个窗口。

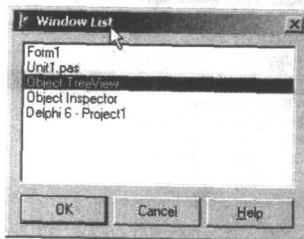


图 1-7 Window List 窗口

6) Toggle Form/Unit 命令

此命令用于在代码编辑器和窗体设计器之间切换。

7) Units 和 Forms 命令

这两个命令分别打开 View Unit 和 View Form 对话框，用于显示和选择单元或窗体。

8) Toolbars 菜单项

该菜单项中的子菜单项用于选择在工具栏中显示哪些类工具按钮。