

第 1 章 基础知识

本章主要介绍在 Delphi 数据库应用程序开发中,用户需要了解的有关面向对象程序设计的基本知识、Delphi 系统概况和主要功能。

1.1 Delphi 系统简介

1.1.1 Delphi 的启动和关闭

Delphi 系统的运行是基于 Windows 95 操作系统环境的,它的运行方法是:

1. 单击“开始”按钮,选择“程序(P)”菜单项,在弹出的“程序”子菜单中选择“Borland Delphi 2.0”,可以看到它的下一级子菜单中,包含两个子菜单项:“Borland Delphi 2.0”和“Delphi 2.0”。

2. 启动 Delphi 有两种方式。

第一种方式:单击第一个子菜单项“Borland Delphi 2.0”,打开 Delphi 2.0 程序组(见图 1.1),双击其中的“Delphi 2.0”图标,启动 Delphi。

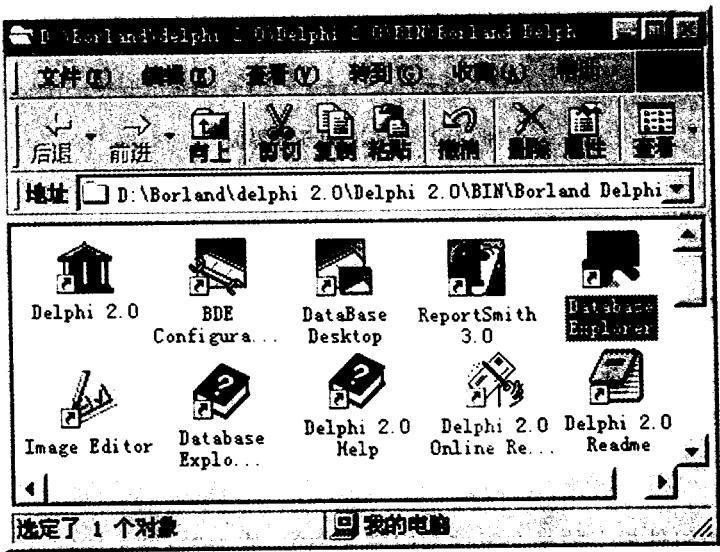


图 1.1 Delphi 的程序组

第二种方式:单击“Delphi 2.0”子菜单项,直接启动 Delphi。

为了简化叙述,上面的这些步骤(以第二种方式为例)简写为:单击 Win 95 的“开始 | 程序 | Borland Delphi 2.0 | Delphi 2.0”命令。在以后各章节中,凡需要多层次选择菜单项时,我们都采用这种简写方式。

为了更简单、更直接地启动 Delphi,用户可以在 Windows 95 的桌面上新建一个 Delphi 快捷方式图标,双击此图标就可以直接启动 Delphi。

启动 Delphi 之后,屏幕上显示出图 1.2 所示的界面,这就是 Delphi 的集成开发环境。顶部为标题,第二行为 Delphi 系统的菜单标题栏,下一行的左部为快速按钮组,右部为组件板,窗口的左下部为对象监视器(Object Inspector),右下部是程序代码编辑器和开发窗体。

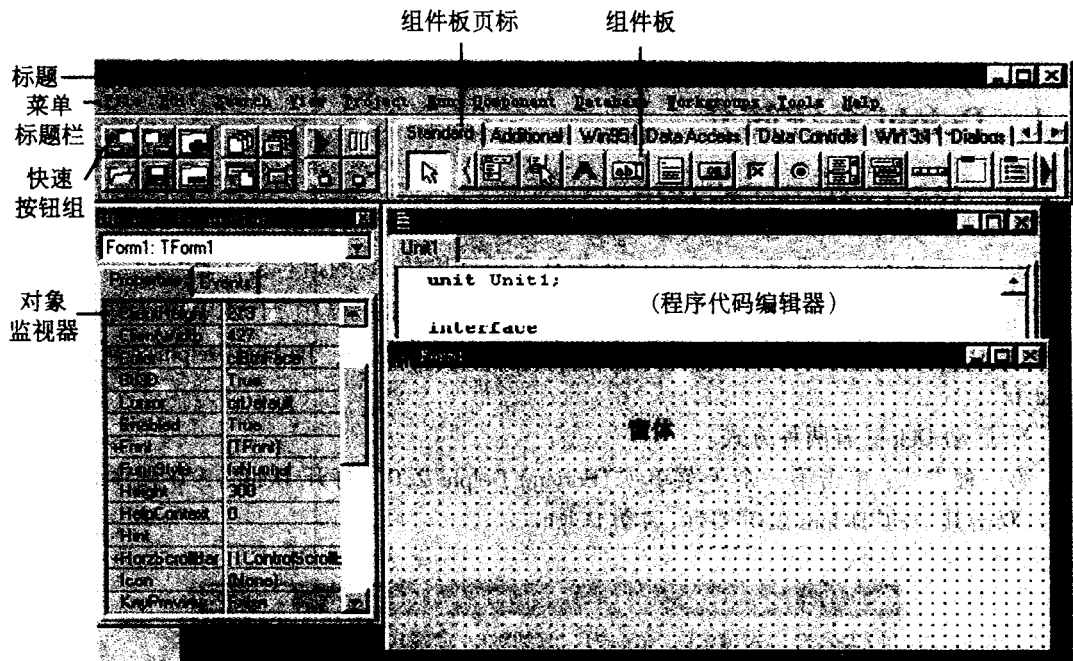


图 1.2 Delphi 的开发界面

3. 退出 Delphi。

单击“File | Exit”命令,或者单击窗口右上角的关闭按钮,都可退出 Delphi。

1.1.2 类和对象

类和对象是两个最基本、最重要的概念。我们可以认为,在面向对象程序设计中的每个实体都可看作是对象,每个对象都属于某一类,因而类和对象是既相关却又不相同的。

1.1.2.1 类

在 Delphi 中,组件板上的每个组件都构成一个类,称为“组件类”,除此之外还有其他的类。在组件类中,一部分组件是基于窗体的称为可视组件,即它有大小、位置、颜色等特征,设计时在窗体上显示的状态与运行时相同;另一部分为非可视组件,设计时它以图标形式显示在窗体上,用来管理其他可视对象,运行时并不出现。在可视组件中,某些类创建的对象还可包含其他对象,如窗体、面板等,在其上面可放置按钮、列表框等对象,这种类称为容器类。

1.1.2.2 对 象

对象是类的一个实例。一个类可以生成多个对象,每个对象都具有基本相同的性质,但它们却可以表现不同的状态。

- 对象属性: 每个对象都具有属性,属性反映了对象的状态,它是类中数据的表现形式。例如,按钮组件是一个类,它可以在窗体上生成按钮对象,那么,按钮对象的状态(如位置、尺寸、文字或图象等)就称为它的属性。
- 对象事件: 每个对象都可对某些动作做出识别和反应,这些动作称为事件。例如:单击按钮对象就可触发它的 Onclick 事件。
- 对象方法: 方法是与对象相关联的程序过程。它反映了对象可以执行的操作。我们仍以按钮为例,它可以在窗体上显示或隐藏,那么“Show”和“Hide”就是按钮具有的方法。

1.1.2.3 进一步理解类

类是对象的模板和框架。在类的定义中包含了对象数据的定义和处理这些数据的方法,类将两者结合在一起,具有紧密的代码结构。因而,类具有如下特征:

1. 封装性 使用类创建对象时,不必关心类本身的内部代码结构,就可随意设置对象属性,它会自动反映到定义类的代码中,从而使类创建对象极为简便。
2. 继承性 利用已有的类可以派生出新类,称为子类或派生类。它继承父类的全部功能,也可对其增加新的功能,父类的任何改变都反映到它的子类中,因此,类是分层次的。
3. 多态性 在同一层次的类中,对于同一种操作,不同的类做出各自不同的反映。例如:在组件板中选择不同的组件,在窗体上都使用同一个单击鼠标的操作,却创建出不同的对象。

1.1.3 组件板

Delphi 中具有丰富的组件,按照特性,它们被分成几组,放在组件板不同的页中。图 1.2 中显示的是标准组件板中的各种组件,单击组件板上的其他页标就会显示出相应页中的组件。本书常用到标准、辅助、数据库控制和数据库存取组件板中的组件,列于表 1.1 至表 1.4 中。其他组件板参见附录 A。

表 1.1 标准组件板

图 标	组件名称	可视性	说 明	所属程序单元
	MainMenu	否	下拉式主菜单	Menus
	PopupMenu	否	弹出式菜单	Menus
	Label	是	标签	StdCtrls
	Edit	是	编辑框	StdCtrls
	Memo	是	备注框	StdCtrls

续表

图 标	组件名称	可视性	说 明	所属程序单元
	Button	是	按钮	StdCtrls
	CheckBox	是	复选框	StdCtrls
	RadioButton	是	单选钮	StdCtrls
	ListBox	是	列表框	StdCtrls
	ComboBox	是	组合框	StdCtrls
	ScrollBar	是	滚动条	StdCtrls
	GroupBox	是	成组框	StdCtrls
	RadioGroup	是	单选钮成组框	ExtCtrls
	Panel	是	面板	ExtCtrls

表 1.2 辅助组件板

图 标	组件名称	可视性	说 明	所属程序单元
	BitBtn	是	图标按钮	Buttons
	SpeedButton	是	快速按钮	Buttons
	MaskEdit	是	屏蔽编辑框	Masks
	StringGrid	是	文本网格	Grids
	DrawGrid	是	绘图网格	Grids
	Image	是	图象框	StdCtrls
	Shape	是	图形	ExtCtrls
	Bevel	是	立体框	ExtCtrls
	ScrollBar	是	滚动框	Forms

表 1.3 数据库控制组件板

图 标	组件名称	可视性	说 明	所属程序单元
	DBGrid	是	数据库网格	DBGrids
	DBNavigator	是	数据库定位器	DBCtrls
	DBText	是	数据库文本	DBCtrls
	DBEdit	是	数据库编辑框	DBCtrls
	DBMemo	是	数据库备注框	DBCtrls
	DBImage	是	数据库图象框	DBCtrls
	DBListBox	是	数据库列表框	DBCtrls
	DBComboBox	是	数据库组合框	DBCtrls
	DBCheckBox	是	数据库复选框	DBCtrls
	DBRadioGroup	是	数据库单选钮成组框	DBCtrls
	DBLookupListBox	是	数据库查看列表框	DBLookup
	DBLookupComboBox	是	数据库查看组合框	DBLookup
	DBCtrlGrid	是	数据库控制网格	DBCtrls

表 1.4 数据库存取组件板

图 标	组件名称	可视性	说 明	所属程序单元
	DataSource	否	数据源	DB
	Table	否	表	DBTables
	Query	否	查询	DBTables
	StoredProc	否	存储过程	DBTables
	Database	否	数据库	DB
	Session	否	对话	DB
	BatchMove	否	批量移动	DBTables
	UpdateSQL	否	修改 SQL	DB
	Report	否	报表	Report

各个组件的含义无需记忆,当鼠标箭头移至各组件时,就会显示出组件的名称。

1.1.4 Delphi 的系统菜单

Windows 环境下应用程序的菜单有其相同的形式和操作方法,系统的环境设置和开发手段都要通过操作菜单命令来完成。因此,全面地了解菜单可以使开发人员对 Delphi 所提供的丰富功能做到心中有数,有助于简单、快速地执行所需操作,完成预期的开发工作。

菜单标题栏在 Delphi 主窗体的上部(参见图 1.2),显示各个菜单的标题,它们包括: File(文件)、Edit(编辑)、Search(查找)、View(视图)、Project(项目)、Run(运行)、Component(组件)、Database(数据库)、Workgroup(工作组)、Tools(工具)和 Help(帮助)。

以下各表列出 Delphi 系统各菜单标题包含的菜单项和主要功能。

表 1.5 File 菜单

命 令	快捷键	说 明
New...		打开“新项”对话框
New Application		创建新项目
New Form		创建新窗体
New Data Module		创建新数据模块
Open...		打开单元或项目
ReOpen		重新打开已打开的文件
Save	Ctrl + S	保存当前窗体和单元
Save As...		用新名称保存当前窗体和单元
Save Project As...		用新名称保存当前项目
Save All		保存当前项目和所有的当前窗体及单元
Close		关闭当前窗体和单元
Close All		关闭项目、窗体和单元等所有文件
Use Unit...		把 uses 子句加入到当前单元文件中
Add to Project...		把当前单元名称添加到当前项目文件的 uses 子句中
Remove from Project		将当前单元从当前项目中移去
Print...		打开“打印选项”对话框
Exit		退出 Delphi

表 1.6 Edit 菜单

命 令	快捷键	说 明
<u>U</u> ndelete/Undo	Ctrl + Z	恢复刚刚删去的项/取消刚刚进行的操作
<u>R</u> edo	Shift + Ctrl + Z	重复执行刚执行的操作
<u>C</u> ut	Ctrl + X	把窗体或单元中选取的对象、文本或项等剪切下来,放到剪贴板中
<u>C</u> opy	Ctrl + C	把窗体或单元中选取的对象、文本或项等复制到剪贴板中
<u>P</u> aste	Ctrl + V	把剪贴板中的内容粘贴过来
<u>D</u> elete	Ctrl + Del	删去选取的对象、文本或项等
S <u>e</u> lect All		选取当前窗体上所有的对象
A <u>l</u> ign to <u>G</u> rid		把选取的对象对齐到邻近的网格点上
B <u>r</u> ing to <u>F</u> ront		把选取的对象移到前面
S <u>e</u> nd to <u>B</u> ack		把选取的对象放到后面
<u>A</u> lign...		打开“对齐”对话框
<u>S</u> ize...		打开“尺寸”对话框
S <u>c</u> ale...		按比例缩放窗体上的所有对象
T <u>a</u> b O <u>r</u> der...		设置对象取得焦点的次序
C <u>r</u> ea <u>t</u> ion <u>O</u> rder...		改变对象的创建次序
L <u>o</u> ck <u>C</u> ontrols		锁住对象,防止偶然的移动或剪切
<u>O</u> bject		编辑或更改 OLE 对象

表 1.7 Search 菜单

命 令	快捷键	说 明
<u>F</u> ind...	Ctrl + F	显示“查找文本”对话框
<u>R</u> eplace...	Ctrl + R	显示“替换”对话框
S <u>e</u> arch A <u>g</u> ain	F3	重复前一次的查找操作
<u>I</u> ncremental Search	Ctrl + E	用户输入每个字符时进行查找
<u>G</u> o To Line Number...		转到代码的指定行号处
F <u>i</u> nd <u>E</u> rror...		显示“查找错误”对话框
B <u>r</u> owse <u>S</u> ymbol...		打开“对象浏览器”,查看对象树

表 1.8 View 菜单

命 令	快捷键	说 明
<u>P</u> roject Manager		打开“项目管理器”
P <u>r</u> oject Source		显示项目文件(.dpr)
<u>O</u> bject Inspector	F11	显示对象监视器
<u>A</u> lignment Palette		显示“对齐板”
<u>B</u> rowser		显示对象的层次结构
<u>B</u> reakpoints		显示“断点列表”对话框
<u>C</u> all Stack		显示函数调用的栈结构
<u>W</u> atches		显示“监视值”列表
<u>T</u> hreads		显示各个程序的状态和位置
Component <u>L</u> ist		显示组件列表(按字母顺序排列)
Window List...	Alt + O	显示窗口列表
T <u>o</u> ggle Form/Unit	F12	在窗体和代码窗口之间进行切换
<u>U</u> nits...	Ctrl + F12	打开“单元”对话框,可从列表中选择一个单元
<u>F</u> orms...	Shift + F12	打开“窗体”对话框,可从列表中选择一个窗体
New <u>E</u> dit Windows		打开新的代码编辑器
<u>S</u> peedbar		设置是否显示“快速按钮组”
Component <u>P</u> alette		设置是否显示“组件板”

表 1.9 Project 菜单

命 令	快捷键	说 明
<u>A</u> dd To Project...		打开“加入项目”对话框,将窗体和单元加到项目中
<u>R</u> emove from Project...		打开“从项目中移去”对话框,将窗体和单元从项目中移去
Add <u>T</u> o Repository...		打开“保存为模板”对话框,将当前窗体加到陈列窗中
<u>C</u> ompile	Ctrl + F9	编译当前项目中所有修改过的单元文件
<u>B</u> uild All		重新编译项目中的所有单元文件
<u>S</u> yntax Check		编译项目中的模块并进行语法检查,但不进行链接
<u>I</u> nformation...		打开“信息”对话框
<u>O</u> ptions...		显示“项目选项”对话框

表 1.10 Run 菜单

命 令	快捷键	说 明
<u>R</u> un	F9	编译并运行当前应用程序
<u>P</u> arameters		为运行程序设置命令行参数
<u>S</u> tep Over	F8	单步执行程序代码(不跟踪到过程和函数内部)
<u>T</u> race Into	F7	单步执行程序代码(跟踪到过程和函数内部)
Trace to Next Source Line	Shift + F7	停止在下一行源代码上
Run to <u>C</u> ursor	F4	运行到光标所在源代码位置处
Show Execution Point		返回编辑窗口,显示程序的当前执行点
Program Pause		暂停程序执行
Program <u>R</u> eset	Ctrl + F2	当前程序返回到启动状态
Add <u>W</u> atch...	Ctrl + F5	加入一个调试监视值
Add <u>B</u> reakpoint...		在代码中插入一个断点
<u>E</u> valuate Modify...	Ctrl + F7	检查和修改表达式的值

表 1.11 Component 菜单

命 令	说 明
<u>N</u> ew...	打开“组件”对话框,准备创建组件
<u>I</u> nstall...	加入新组件
Open <u>L</u> ibrary...	打开组件库
<u>R</u> ebuild Library	重建组件库
Configure <u>P</u> alette...	重新配置组件板

表 1.12 Database 菜单

命 令	说 明
<u>E</u> xplore	启动数据库资源管理器
<u>S</u> QL Monitor	启动 SQL 监视程序,查看通过 SQL 链接或 ODBC 数据源得到的 SQL 语句调用
<u>F</u> orm Expert...	打开数据库窗体向导

表 1.13 Workgroup 菜单

命 令	说 明
Browse <u>P</u> VCS Projects	显示 PVCS(版本管理程序)项目窗口,在其中执行所有版本管理任务
Manager Archive <u>D</u> irectories	显示“档案文件目录”对话框,可加入、删除或建立档案文件目录
<u>A</u> dd Project to Version Control	把当前项目存放到版本管理目录中
<u>S</u> et Data Directories	显示“数据目录”对话框,可设置公有或私有根目录的位置

表 1.14 Tools 菜单

命 令	说 明
Options...	显示“环境选项”对话框
Repository...	打开“对象陈列窗”窗口
Tools...	显示“工具选项”对话框
Image Editor	显示图象编辑器
Database Desktop	打开数据库桌面

表 1.15 Help 菜单

命 令	说 明
Help Topics	显示“帮助主题”对话框
About...	显示 Delphi 图标、版权和版本信息

要想全面了解 Delphi 的功能或解决开发中所遇到的疑难问题,使用联机帮助是一条捷径。获取联机帮助的方式主要有两种:一种是选上述“Help|Help Topics”命令,显示出帮助对话框,在“Index”页中输入要检索的主题词(例如:对象、属性、事件或方法的名称),单击“Display”按钮,可查阅到该主题的详细信息,单击“Example”,还可以显示出应用范例;另一种是用单击对象或在对象监视器的对象选单中选取对象的办法来指定对象,然后在对象监视器的属性或事件页中选有关属性或事件,再按“F1”键,在帮助窗口显示出其功能和用法。对象的方法是系统提供给开发人员编程时运用的一些过程和函数,在对象监视器中看不到它们,要想了解其含义,可单击对象,再按“F1”键,在显示出的帮助窗口中,单击“methods”,列出该对象可使用的各种方法的名称,单击某一方法名称,则会提供该方法的用途和使用方式等信息。

1.1.5 快速菜单

当鼠标指向某一对象单击右键时,就会弹出一个菜单,称为快速菜单。快速菜单是与特定对象相联系的,它列出适用于该对象的一系列命令,如图 1.3。

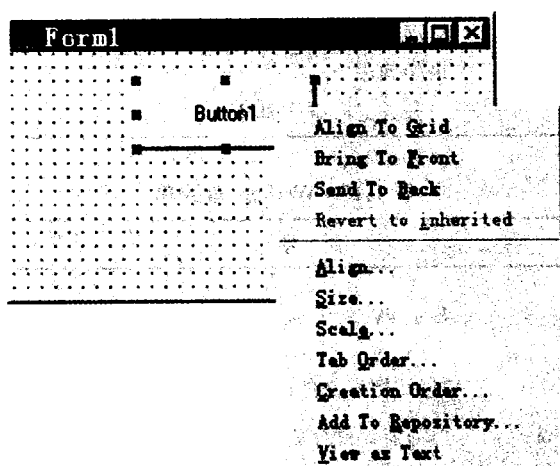


图 1.3 快速菜单

1.1.6 快速按钮组

Delphi 将菜单中使用频率较高的功能选出,用单击按钮的快捷方式启动,这样形成了快速按钮组。快速按钮与某些常用菜单项的作用相同,单击它就可以执行相应的菜单项命令。

Delphi 提供的快速按钮组见表 1.16。

表 1.16 快速按钮组

图标	名称	对应菜单项
	Open Project	File Open
	Save All	File Save All
	Add File to Project	File Add to Project
	Select Unit from List	View Units
	Select Form from List	View Forms
	Run	Run Run
	Pause	Run Program Pause
	Open File	File Open
	Save File	File Save
	Remove File from Project	File Remove from Project
	Toggle Form/Unit	View Toggle Form/Unit
	New Form	File New Form
	Trace Into	Run Trace Into
	Step Over	Run Step Over

1.1.7 对象监视器

对象监视器用于查看和管理对象属性和事件,选“View | Object Inspector”命令可将它显示在窗体界面上,见图 1.4。

对象监视器上方是对象选单,例如当前对象是 Form1,则在对象选单中为“Form1; TForm1”,左边是该对象的名称,与该对象的 Name 属性相同,右边为其类的名称,Delphi 中类使用固定的前缀“T”。单击对象选单的组合框可以弹出该窗体上所有对象名称的列表。

对象监视器具有两个页标: Properties(属性)和 Events(事件)。在窗体设计阶段,点选属性页可以显示和设置该对象的各种属性;点选事件页(见图 1.5)列出该对象具有的事件,双击某一事件,可弹出事件代码编辑器窗口,用于编写或修改事件处理程序代码。

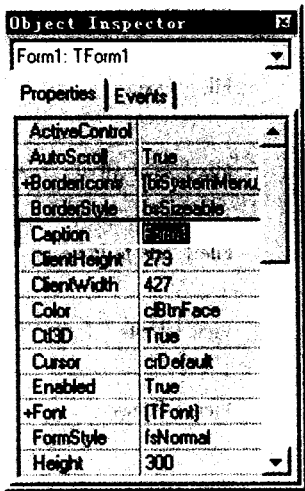


图 1.4 对象监视器的属性页

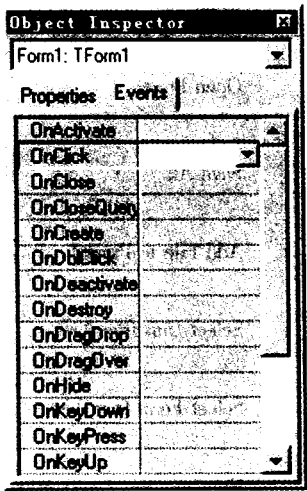


图 1.5 对象监视器的事件页

1.1.8 窗 体

启动 Delphi 时,系统自动建立一个项目和一个空窗体,参见图 1.2。
运用 Delphi 开发应用程序的过程实际上就是在窗体上放置各种对象、进行属性设置和编写事件处理代码的过程,因此,开发人员要熟练掌握窗体上的基本操作。

1. 在窗体上建立对象的几种方式
- 在组件板上选择所需组件后,可以有以下几种方式给窗体加入相应的对象:
- (1) 双击所需组件,则在窗体中央位置以默认大小加入相应对象。
 - (2) 点选所需组件,在窗体的适当位置上单击一下,即可在此位置上以默认大小加入相应对象。
 - (3) 点选所需组件,在窗体的所需位置上点拉出一个适当大小的方框,建立相应对象。
 - (4) 若要在窗体上“连续加入”多个同类对象,可以按住 Shift 键,用鼠标点选所需组件,然后在窗体上按照上述方式之一连续加入几个对象;若要结束此种“连续加入对象”状态,只需单击组件板左侧的鼠标图标即可。

无论用上述哪一种方式加入对象,当加入的对象为被选中状态,可拖拉该对象放到适当位置,也可将其放大或缩小。

2. 选取多个对象、调整对齐方式和大小
- 在设计窗体时,有时需要对一组对象进行统一处理(例如共同移动到适当位置),这时就

需要同时选取多个对象,选取的方式有:

- (1) 点选第一个对象(称为“参照”对象),然后按住 Shift 键,再相继点选其他各个对象。
- (2) 在窗体上点拉一个范围把相邻的几个对象框在其中。
- (3) 若要选取窗体上的所有对象,可直接选“Edit | Select All”命令。

选取一组对象之后,就可对它们进行共同处理,常用的操作是把它们对齐或调整大小。

例如在窗体上先后建立了两个编辑框对象,位置不齐、大小也不相同,很不协调。首先要按上述办法同时选取这两个对象,见图 1.6。

先进行对齐操作,有两种方式:

- (1) 使用“对齐板”:选“View | Alignment Palette”命令,出现“对齐板”工具,如图 1.7 所示,选取所需对齐模式,就可执行对齐操作。

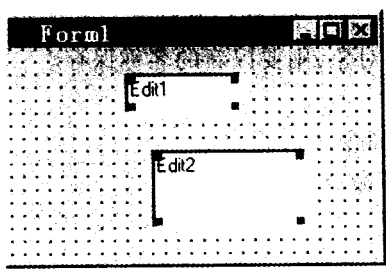


图 1.6 选取两个对象

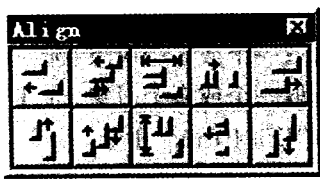


图 1.7 “对齐板”

- (2) 使用“对齐”对话框:选“Edit | Align”命令,或者单击鼠标右键,从快速菜单中选取该命令,弹出“对齐”对话框,见图 1.8,选择横向、纵向对齐模式,单击“OK”按钮即可。

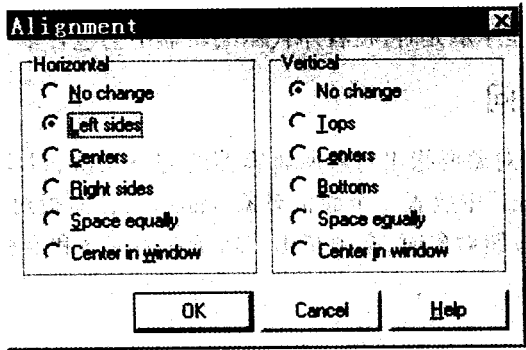


图 1.8 “对齐”对话框

两个编辑框对齐后,若其大小还不相同,此时这两个对象仍为被选中状态,选“Edit | Size”命令,或者在快速菜单中选取该命令,弹出“Size”对话框,选取相应的宽度和高度,见图 1.9,单击“OK”按钮关闭对话框。调整对齐方式和改变大小之后的编辑框见图 1.10。

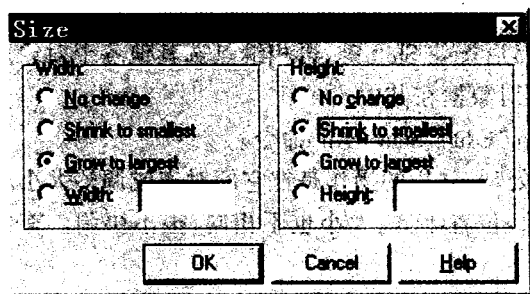


图 1.9 “Size”对话框

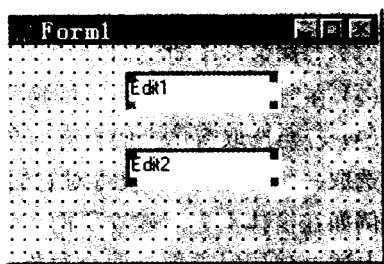


图 1.10 调整对齐方式和大小之后的对象

3. 剪切、复制和粘贴

程序开发人员可以任意剪切或复制一个或多个对象,然后粘贴到其他的位置,这时,对象属性和事件就会随之移动或复制,若出现同名对象,则要以新的名称保存。

操作方式如下:

剪切: 选取一个或多个对象,选“Edit | Cut”命令。

复制: 选取一个或多个对象,选“Edit | Copy”命令。

粘贴: 选“Edit | Paste”命令。

1.1.9 程序代码编辑器

在启动 Delphi 后,程序代码编辑器通常被自动建立的窗体覆盖,若需要查看程序代码时,可将窗体缩小或移动,以显露出程序代码编辑器;或者选“View|Toggle Form/Unit”命令,在窗体和代码窗口之间进行切换;或者选“View| Units”命令,再从中选择需要的单元文件进行查看。

若项目中包含多个程序单元时,在程序代码编辑器中会包含多个页标。

1.1.10 对象的访问

在程序代码中,访问对象要指出对象所属容器的完整层次,以实现访问对象的唯一性。例如,在一个窗体“Myform1”中放置一个面板对象“Panel1”,在面板对象上又放置了按钮对象“Button1”,那么,要访问按钮对象时,必须指出按钮所在的窗体和面板对象,即写为:

Myform1.Panel1.Button1

1.1.10.1 设置属性值

对象的属性表示对象的状态,可以在设计阶段设置,在运行过程中可以引用或修改。由于属性总是针对某一对象的,因此,设置属性值时必须写出完整的对象名。例:

Var: = Myform1 . Panel1 . Button1 . Name 将按钮 Button1 的名称赋给变量 Var
Myform1 . Panel1 . Button1 . Name: = "Exmpo1" 将按钮 Button1 的名称命名为 Exmpo1

1.1.10.2 方法的调用

在程序中调用对象的方法可实现某种操作。由于方法也是属于某一对象的,所以,调用

对象的方法也必须写出完整的对象层次,例:

Myform1.Panel1.Button1.Hide

隐藏按钮 Button1

Myform1.Panel1.Button1.Show

显示按钮 Button1

1.2 Delphi 的文件类型

运用 Delphi 进行数据库应用程序开发时,经常使用三种最基本的文件类型:项目文件,后缀为“. dpr”;单元文件,后缀为“. pas”;窗体文件,后缀为“. dfm”。

1.2.1 项目文件

一个 Delphi 应用程序一定有一个项目文件,它用于记录和管理该项目中所有的文件,其中最主要的是窗体文件和单元文件。项目文件是一个可运行的文件,也称为应用程序文件。

查看项目文件有三种办法:

1. 选“View | Project Manager”命令,出现图 1.11 的窗口,它显示出项目文件中包含的窗体和单元文件名称。
2. 选“View | Project Source”命令;或者先选“View | Units”命令,再选项目文件名称(见图 1.12),单击“OK”按钮。

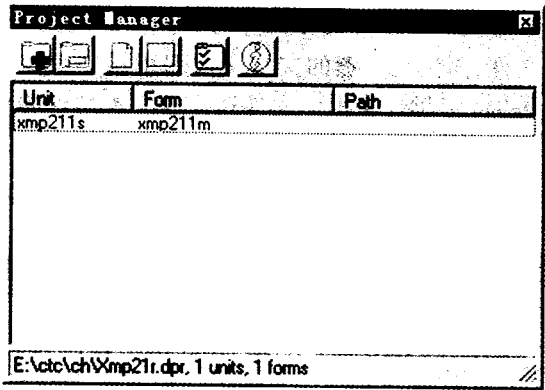


图 1.11 “View | Project Manager”窗口

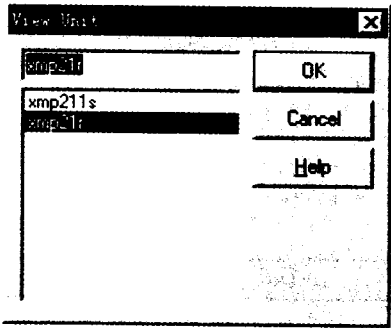
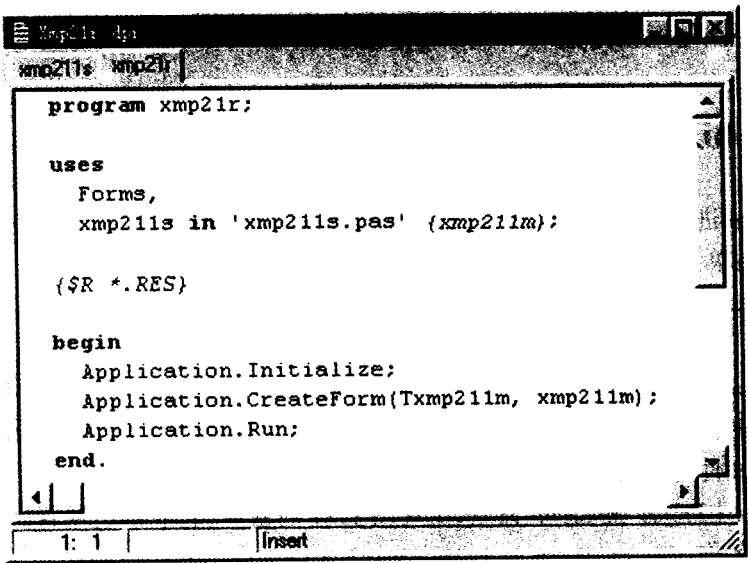


图 1.12 选择项目文件

用以上两种方式都可显示项目文件,见图 1.13。



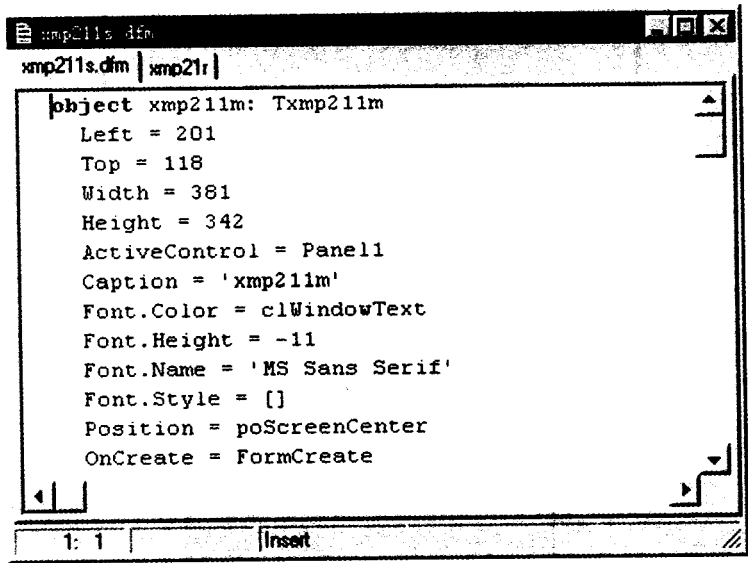
```
program xmp211r;  
  
uses  
    Forms,  
    xmp211s in 'xmp211s.pas' {xmp211m};  
  
{$R *.RES}  
  
begin  
    Application.Initialize;  
    Application.CreateForm(Txmp211m, xmp211m);  
    Application.Run;  
end.
```

图 1.13 显示项目文件

1.2.2 窗体文件

窗体文件是二进制文件,无法在一般的文本文件编辑器中看到。它保存了设计时窗体所包含的对象的属性。Delphi 提供了将窗体文件转变成文本文件的手段。在窗体的快速菜单中选“View as Text”命令,可显示出文本形式的窗体文件,见图 1.14。

Delphi 产生的各种文件的后缀参见附录 C。



```
object xmp211m: Txmp211m  
    Left = 201  
    Top = 118  
    Width = 381  
    Height = 342  
    ActiveControl = Panel1  
    Caption = 'xmp211m'  
    Font.Color = clWindowText  
    Font.Height = -11  
    Font.Name = 'MS Sans Serif'  
    Font.Style = []  
    Position = poScreenCenter  
    OnCreate = FormCreate
```

图 1.14 文本形式的窗体文件

1.2.3 单元文件

单元文件是一个 Object Pascal 源程序代码文件,在程序代码编辑器中进行编辑和修改。

1.3 Delphi 3 和 Delphi 4 的新特性和增强功能

1.3.1 Delphi 3 的新特性和增强功能

Borland 公司于 1997 年 5 月隆重推出的 Delphi 3 立即成为万众瞩目的焦点,因为它具有更强大的功能和更优越的性能,它优于 Delphi 2.0 的主要方面是:

1. 执行代码更短,速度更快

Delphi 3 的执行代码比 Visual Basic 5.0 快 3~6 倍,比 C++ 快 70%。

2. 引入“程序包”的概念

程序包(Package)是特殊的动态链接库,可缩减执行代码,支持应用程序间的代码共享,便于在 Internet 上发布和执行。

3. 代码解释器(Insight)

用于简化代码编写工作。它包括四项功能:代码补充、代码模板向导、代码参数向导和表达式值提示。

4. 新的文档设计

包括:用户指南、开发者指南、对象 Pascal 语言指南和可视化组件库参考。

5. 加强了 RTL

RTL(运行阶段库)增加了数学单元和系统单元的数量和功能。

6. 改进编译

支持程序包、接口和断言。

7. 支持多字节字符

增加和修改运行阶段库中的函数,以支持多字节字符。

8. 新的 BDE 和 SQL 链接

BDE(Borland 数据库引擎)和 SQL 链接都升至相应的 4.0 版。

9. 分布式数据集

提供新的数据集组件 TClientDataSet,可建立不依赖数据库的数据集对象。提供强大的服务器纠错程序,易于完成客户/服务器程序的调试。

10. 支持 COM 和建立 ActiveX

支持 COM(组件对象模型 Component Object Model)跨平台对象通讯标准;具有 ActiveX 组件,支持 OLE 自动化和可视化编辑的复合文档对象。

11. 全面支持 Internet 和 Intranet

提供一系列新组件,支持各种通信协议。

12. 加强了 OLE

OLEEnterprise 提供 OLE/DCOM 与 RPC 或 COM 服务器的接口。

13. 新的样板应用程序