

计算机高级编程技术



第一章 DELPHI 简介





第1章 Delphi 7.0简介

Delphi是Borland公司开发的可视化软件开发系统，基于Windows 95/98/NT/2000/XP，采用高度结构化的Object Pascal语言，具有结构清晰、高效优化等特点。

- 1.1 Delphi 7.0的特点
- 1.2 Delphi 7.0的集成开发环境
- 1.3 Delphi的可视化编程





1.1 Delphi 7.0的特点

Delphi 7.0是32位的应用开发环境，作为一种优秀的开发环境，它具有一系列新特性：

- 改进的集成开发环境
- 强劲的工程管理功能
- 先进的调试技术
- 扩展的语言特性
- 其强大的数据库应用程序开发能力





1.1.1 集成开发环境（IDE）的改进

□ 1. ADO数据集

ADO数据集提供了通过Microsoft的Active Data Objects（ADO）技术访问各种格式的数据。

□ 2. 数据模块设计窗口

数据模块设计窗口是一种可视化的数据模块设计工具。

□ 3. InterBase Express组件

InterBase Express（IBX）组件将InterBase融入Delphi，具有更强的数据访问能力。





1.1.1 集成开发环境（IDE）的改进

□ 4. MIDAS的增强

支持MIDAS应用程序的结构现在可以支持远程数据模块和新增的Internet Express组件，这样就使用户在浏览器与MIDAS应用程序服务器冲突时方便地创建网络应用程序。

□ 5. CORBA的更新

CORBA进行了更新以完成与VisiBroker的协同工作。





1.1.1 集成开发环境（IDE）的改进

□ 6. 改进的调试技术

Delphi 7.0的集成调试功能确保了应用程序顺利开发，针对特殊程序设置了调试选项。

□ 7. Frames结构

Frames是一种特殊形式的窗体，它可以与窗体或者其他的Frames嵌套。

□ 8. 用户定义的桌面设置

用户可以定义各种桌面布局，给它们命名或将它们保存。





1.1.1 集成开发环境（IDE）的改进

□ 9. To-Do Lists

To-Do Lists是用来显示一个工程将执行任务的列表，用户可以在源程序代码中直接加入各条任务，也可以在该列表中直接加入。

□ 10. 对象观察器的属性分类

对象观察器允许用户通过属性分类对属性和事件进行显示和过滤。

□ 11. 对象观察器的下拉列表

在对象观察器中加入的Owner draw支持使得用户可以从对象观察器的属性下拉列表里预览一些图形。





1.1.1 集成开发环境（IDE）的改进

□ 12. 工程管理器功能的增强

新的工程管理器引入工程组的概念，通过工程管理器对工程组进行管理。通过工程组可以很直观地找到各工程直接的联系，将相互联系的工程中共同使用的动态链接库文件（DLL）与可执行文件组织在一起。

□ 13. ActiveX的增强

COM服务器可以作为可视化应用程序中的组件。





1.1.1 集成开发环境（IDE）的改进

□ 14. 新的应用程序模板

一共新增了两个应用程序模板，一是Control Panel Application 数据库开发技术；另一个是选择Console Wizard来建立相应的应用程序。

□ 15. 代码编辑器功能的增强

Delphi 7.0的代码编辑器拥有类生成器，模块导航和代码浏览器，这些工具大大简化类的创建。





1.1.1 集成开发环境（IDE）的改进

□ 16. 窗体可以保存为文本

窗体文件现在可以作为文本文件保存而不再是传统的二进制文件。

□ 17. IDE命令行选项

可以通过设置命令行的选项启动IDE，这一特性在调试时显得尤为重要。

□ 18. 网络工具

Delphi 7.0提供了一套工具Integrated Translation Environment (ITE)，用于建立不同地点同时应用的应用程序。





1.1.2 新的语言扩充

将Object Pascal语言进行了扩充，增加了动态数组、方法重载、默认参数等，使应用程序的编程过程更加简单。

□ 1 动态数组

动态数组的概念是相对于静态数组而言的。所谓静态数组就是指声明数组的同时必须定义该数组的大小。而这样往往会造成资源的极大浪费，因为用户不得不将数组定义得足够大以避免出现数组的大小不够。而引入了动态数组，在用户不能确定数组大小的时候，可以不必事先声明其大小，系统会自动地分配内存以满足数组大小的需要。





1.1.2 新的语言扩充

□ 2 方法重载

在Delphi 7.0中引入了方法重载的概念，即一个方法的名称可以对应多个方法，这些方法可以有着不同的参数类型和参数数目，在编译过程中，系统会自动根据方法参数类型以及参数数目的不同来确定到底使用了哪一种方法。

□ 3 默认参数

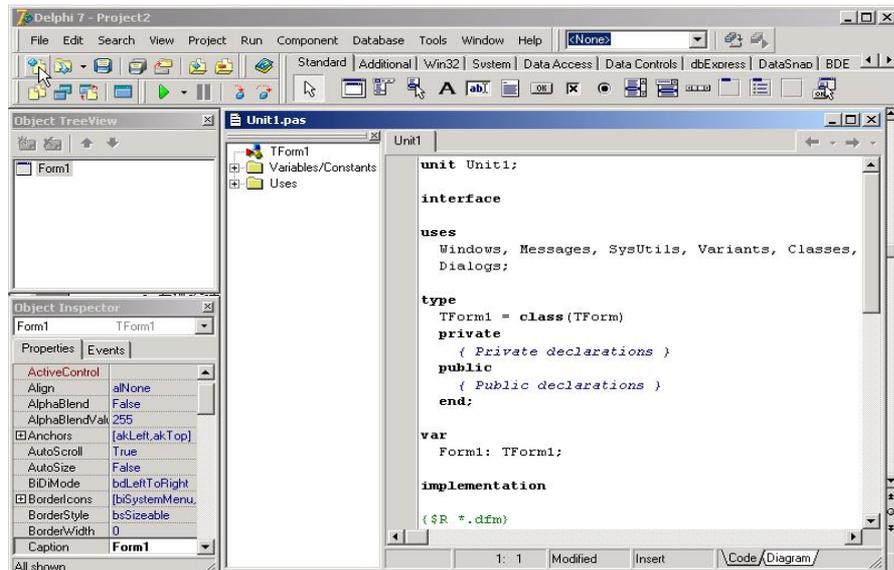
在函数中使用默认参数，将一些不是经常使用的参数设置为默认值，避免了以往出现的程序员编程时需要不停地查找函数手册或记忆大量但不是很重要的信息。





1.1.2 新的语言扩充

Delphi 7.0在数据库应用程序开发方面的能力增强主要表现在：
增加的数据库组件；可访问远程和本地数据库及网络；对Access和Oracle的完全支持；可更好地支持Client/Server数据库应用程序开发；客户数据集的增强。





1.1.2 新的语言扩充

□ 1 Business Insight

Business Insight功能为资料转换成信息做更佳的决策支援。

□ 2 Decision Cube

Decision Cube功能能够提供交差分析表用于多维资料分析。

□ 3 Quick Reports 2.0

Quick Reports 2.0可轻松简便地产生内嵌式的报表打印与预览，并可直接提供图片与统计图表打印。

□ 4 Active Insight

Active Insight作为企业元件开发中心，提供最高的可重用生产力。





1.2 Delphi 7.0的集成开发环境

- Delphi 7.0与Visual Basic一样，是用来创建Windows应用程序的一种快速应用开发工具，一种可视化编程环境。用户可以直观地使用图形化工具来创建Windows应用程序。
- Delphi 7.0的IDE与Delphi以前的版本类似，使用一组窗口、菜单和应用程序来设计可视化界面，将代码以事件的形式与界面的每一元素建立联系，并可以对整个应用程序进行调试。

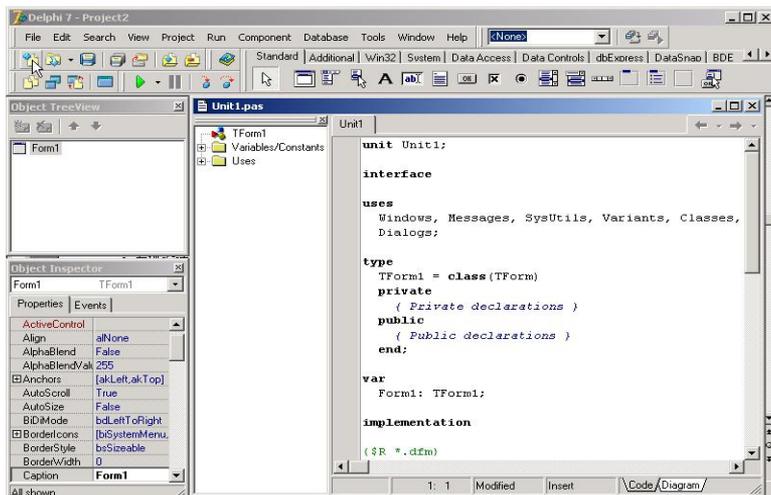




1.2 Delphi 7.0的集成开发环境

运行Delphi 7.0时的开发环境，如图所示。从图中用户可以清楚地看到Delphi 7.0的开发环境由以下几部分组成：

- ❑ 1.2.1 主窗口
- ❑ 1.2.2 代码编辑器
- ❑ 1.2.3 窗体设计器
- ❑ 1.2.4 对象观察器
- ❑ 1.2.5 工程管理器





1.2.1 主窗口

Delphi 7.0的主窗口位于整个开发环境的顶部，如图所示。它是开发过程中的核心，用户通过对主窗口进行操作从而实现软件的开发工作。



主窗口

主窗口分为以下三个部分：菜单栏（Menu）、快捷工具栏（Speed Bar）和组件板。





1.2.1 主窗口

1. 菜单栏



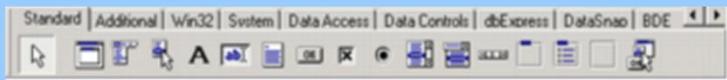
利用菜单可以完成Delphi 7.0的IDE提供的所有功能。比如选用组件、设置环境、选择编译窗口选项、查看帮助信息以及建立新窗体或单元。

2 快捷工具栏



快捷工具栏位于主窗口的左下端。利用快捷工具栏，用户可以完成工程的打开、文件的保存、向工程中添加文件、文件的打开、文件的删除、窗体设计器和代码编辑器之间的切换、创建新的窗体以及一些基本的调试功能。

3. 组件板



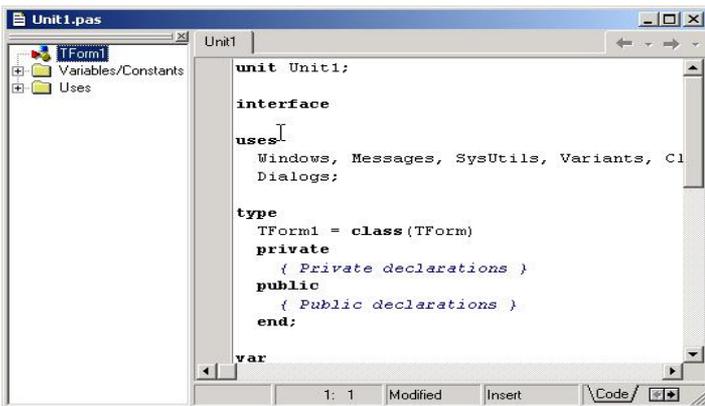
组件板是存放在一起的各种组件，按各自类型存放在不同的页面中。



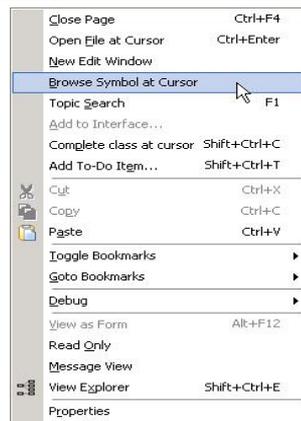


1.2.2 代码编辑器

代码编辑器又称作单元窗口，是编写程序代码的地方。它一般隐藏在窗体设计器的下面。



代码编辑器



代码编辑器的右键快捷菜单

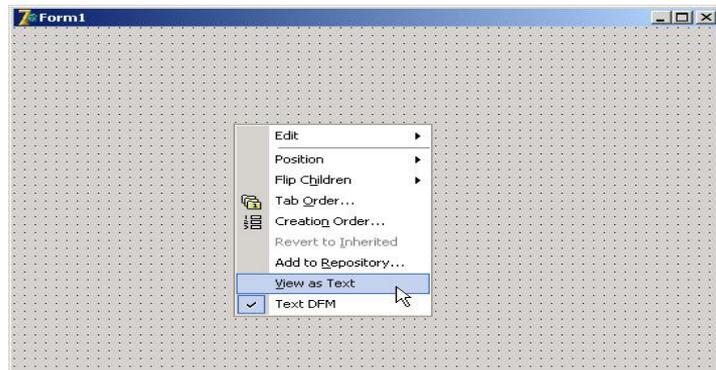




1.2.3 窗体设计器

窗体设计器作为**IDE**的一个部分，主要通过向窗体中添加组件以及更改窗体及组件的属性来完成窗体的设计。

窗体设计的主要步骤就是向窗体中添加可以完成应用程序任务的组件；更改组件属性使之满足程序设计要求；将组件的事件与程序代码相联系，用代码实现组件的活动。



窗体设计器及其右键菜单



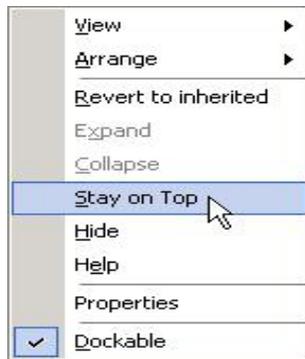


1.2.4 对象观察器

对象观察器（Object Inspector）是用来描述组件对象及窗体对象的属性特征和行为事件的。它是应用程序设计过程中最重要的一个工具。



对象观察器



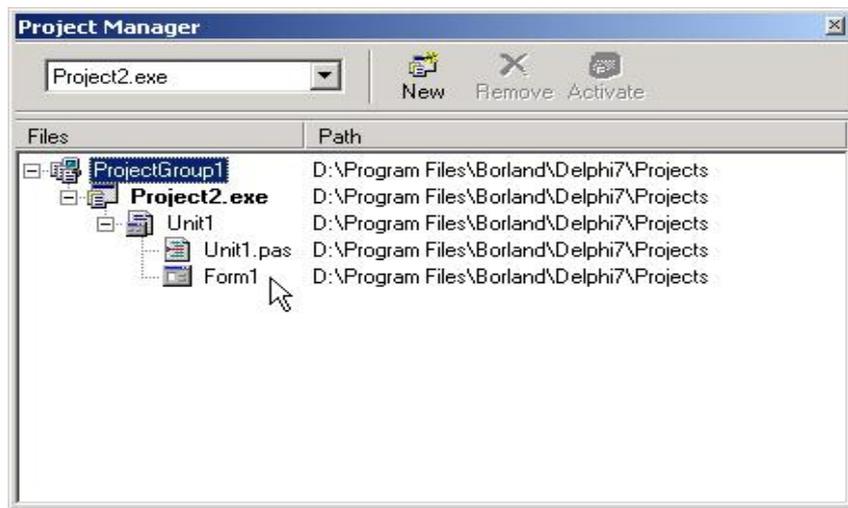
属性的弹出式菜单





1.2.5 工程管理器

工程管理器通过工程组进行管理，列出了反映各工程之间联系的树状表，使用户可以很直观地找到各工程直接的联系，方便地将相互联系的工程中共同使用的动态链接库文件与可执行文件组织在一起。



Project Manager窗口





1.3 Delphi的可视化编程

“可视化编程”与传统编程方法不同，不再需要编写大量代码去描述界面元素的外观与位置，而是采用面向对象、事件驱动的方法，利用Delphi所提供的可视“控件”，在系统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。

Delphi可视化编程的一般步骤为：

- ① 设计界面：利用控件在窗体上创建各种对象。
- ② 设置属性：设置窗体和控件等对象的属性。
- ③ 编写代码：在Delphi所提供的程序框架中加入完成功能的代码。





1.3 Delphi的可视化编程

Delphi可视化编程的具体过程:

□ 1. 创建一个工程

在Delphi中开发的每个应用程序都被称为工程，Delphi编程首先从建立一个工程开始。

新建一工程由以下两种方法:

- ① 启动Delphi后，系统自动生成一个缺省的工程Project1。
- ② 在“File”菜单中选择“New Application”选项，系统将创建一个新的工程。





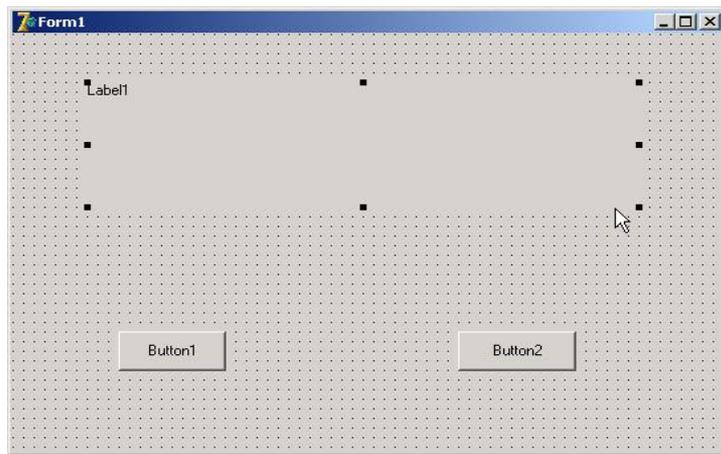
1.3 Delphi的可视化编程

□ 2. 添加控件

设计工程直接面对的是窗体，在窗体中可以添加各种对象，进行窗体的界面设计。

向窗体中添加对象的方法为：

- ① 单击组件板中的控制图标。
- ② 在窗体的适当位置按下鼠标左键并拖动鼠标，即可画出相应的对象。



设计窗体的界面





1.3 Delphi的可视化编程

3. 设置属性

对象属性设置是在对象观察器中属性选项卡中完成的，操作方法如下：

① 首先设置窗体Form1的属性：单击窗体的空白区域，可从“对象”下拉列表框中查看。

② 设置其他对象的属性：单击窗体上的对象，然后根据需要依次设置对象的各个属性。



属性设置后的窗体





1.3 Delphi的可视化编程

□ 4. 编写代码

在事件选项卡中选择OnClick事件，用鼠标双击右边一栏，为Button1添加OnClick事件过程。

Delphi自动生成该事件过程的代码框架，同时打开代码编辑器窗口，将光标定位在代码框架中过程的首行处，等待输过程代码。





1.3 Delphi的可视化编程

在begin与end之间输入过程代码:

```
procedure TForm1.Button1Click(Sender:TObject);  
begin  
    Label1.Caption:='当前时间是'  
    '+chr(13)+Format('%s',[timetostr(Time)]);  
end;
```





1.3 Delphi的可视化编程

用同样的方法，输入按钮Button2的单击（OnClick）事件过程代码如下：

```
procedure TForm1.Button2Click(Sender:TObject);  
begin  
    close;  
end;
```



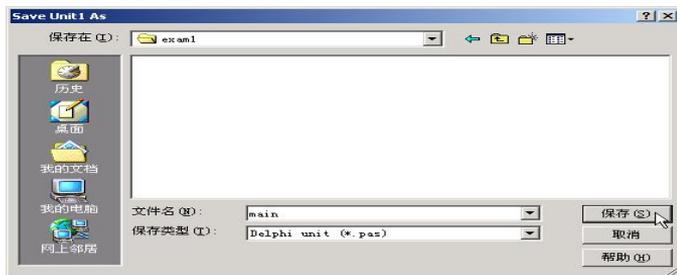


1.3 Delphi的可视化编程

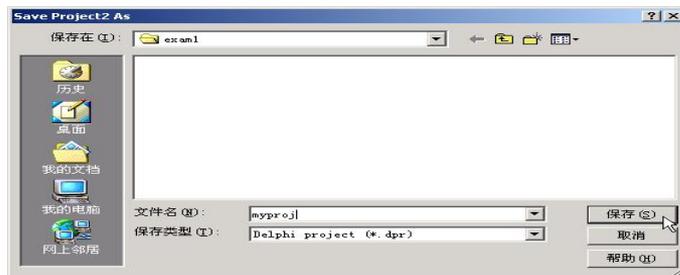
5. 保存工程

保存文件的方法有两种：

- ① 单击菜单“File”，选择“Save All”。
- ② 单击“Standard”工具条上的“Save All”按钮。



保存单元文件



保存工程文件





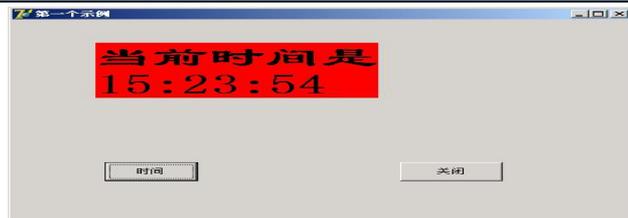
1.3 Delphi的可视化编程

□ 6. 运行工程

单击“Debug”工具条上的“Run”按钮或按〈F9〉键，系统将开始编译、连接、运行该工程。若发现错误，编译器将返回代码编辑器，并提示错误；若无错误，系统将声称可执行文件myproj.exe，并执行它。



运行工程的初始界面



单击“时间”按钮后的界面

