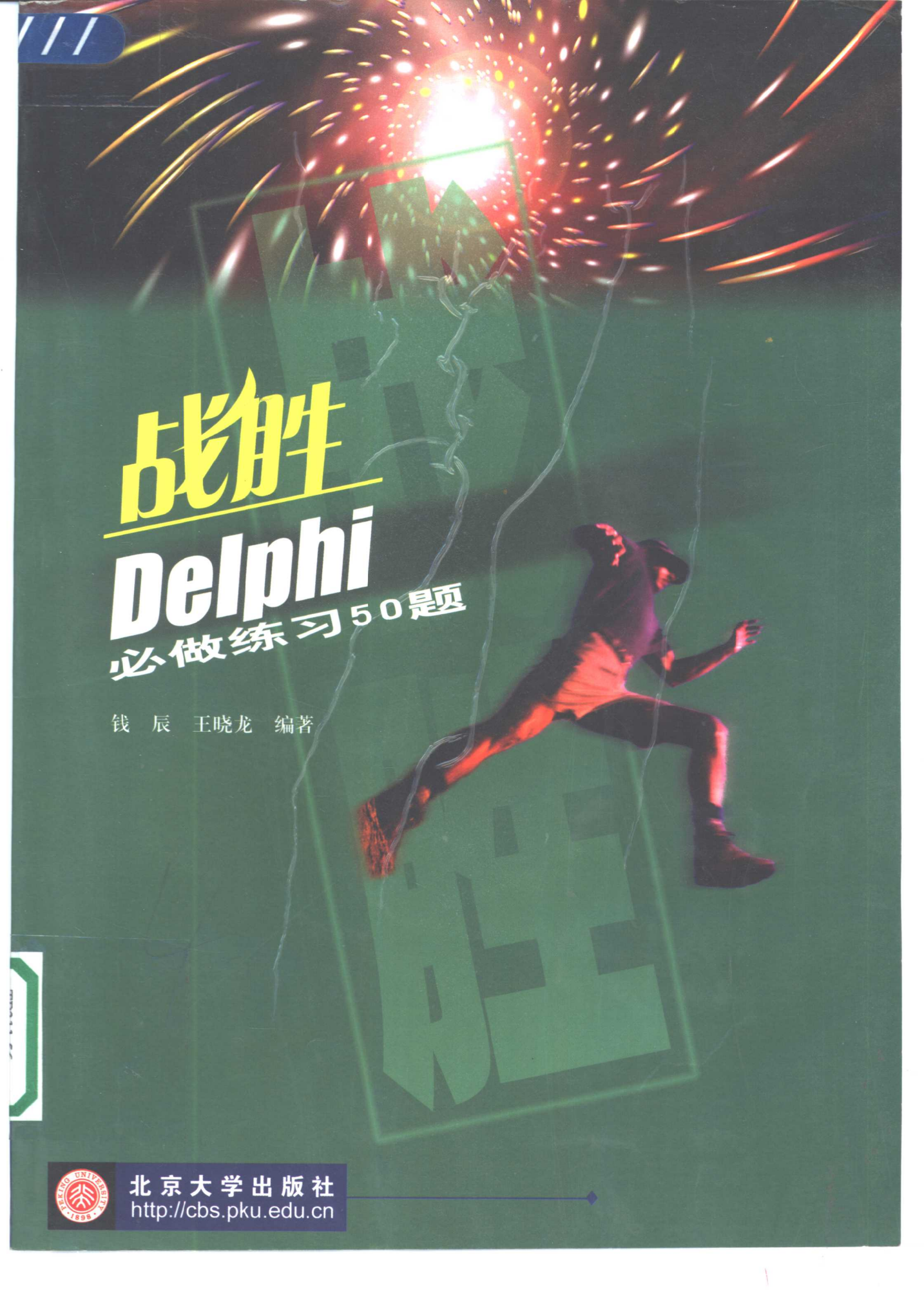




战胜 Delphi

必做练习50题

钱 辰 王晓龙 编著



北京大学出版社
<http://cbs.pku.edu.cn>

TP311.56

96D

战胜 Delphi 必做练习 50 题

钱 辰 王晓龙 编著

北 京 大 学 出 版 社

北 京

内 容 简 介

Delphi 5.5 是 Inprise 公司继 Delphi 5 之后推出的又一个 Windows 可视化编程工具。

本书力求全面介绍用 Delphi 5.5 编程开发的基本知识, 并重点介绍 Delphi 5.5 的新特点和使用技术。全书共包含 50 个实用性和代表性很强的实例, 通过这些例子, 读者能够了解如何使用 Delphi 来编写 Windows 应用程序。本书实例涉及控件的使用、程序语法、数据结构、多线程使用、DLL 编写、本地数据库使用和服务器端数据库设计等。

本书可作为 Delphi 编程开发人员和计算机爱好者的自学用书和编程指导书。

本书含 1 张教学光盘。

本图书及配套光盘的版权由北京大学出版社所有。未经北京大学出版社书面许可, 任何人或任何单位不得以任何形式、任何手段复制或传播其中的任何部分。

图 书 名: 战胜 Delphi 必做练习 50 题

图书写作者: 钱 辰 王晓龙

图 书 责 编: 段志刚 黄庆生

光 盘 责 编: 王 原

本 版 号: ISBN 7-900636-87-0/TP.64

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 62752015 编辑部 62765013 发行部 62750672

网 址: <http://cbs.pku.edu.cn> E-mail: xxjs@pup.pku.edu.cn

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 13.875 印张 352 千字

2001 年 11 月第 1 版 2001 年 11 月第 1 次印刷

定 价: 26 元 (含光盘 1 张)

前 言

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。

Delphi 有众多优越特性。它提供了新颖的可视化设计工具，采用面向对象的方法将 Windows 编程的复杂性封装起来，实现了将可视化设计与 Object Pascal 语言的有机集成，配有 Database Engine，可通过 SQL Links、ODBC 访问多种数据库，并且提供了强大的开发基于客户/服务器模式的数据库应用的能力。Delphi 5.5 是 Inprise 公司继 Delphi 5 之后推出的又一个 Windows 可视化编程工具。Delphi 5.5 简化了 Windows 应用程序和浏览器、Web 服务器、中间件及后端大型数据库系统的集成，并支持 XML 和 HTML4，加快了 Internet 和分布式计算应用系统的设计开发速度。Delphi 5.5 在开发 Internet 应用、数据库系统等方面的性能有了很大提高，同时在提高开发人员的效率、方便开发人员的使用方面也做了很多改进。

在本书的编著过程中，我们尽量注意减少冗长无味的说明，代之以具体实用的例题演示。通过例题，引导读者把握 Delphi 的精髓所在。本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于各个层次的 Delphi 用户。在本书的难度上，我们遵循循序渐进、由浅入深的原则，便于读者理解和学习。对初学者来说，通过本书的阅读可迅速加入 Delphi 高级用户的行列；对有一定使用经验的读者，也可通过本书掌握 Delphi 深层次的开发方法，学会用更巧妙的办法开发出高水平的 Delphi 应用。

本书由钱辰、王晓龙主编，另外，余晓鹏、张伟华、何广、张石勇、战祥森、张松伟、吴绍伟、孙科峰、渠继永、覃文圣、牟南、贾鹏、李衡、钟光辉、邓瑞峰、满晓宇、罗捷、肖健、解灵运等也参加了本书编写工作。

限于作者水平，难免在内容选材和叙述上有不当之处。竭诚欢迎广大读者对本书提出批评和建议。

编 者

2001 年 11 月

目 录

练习 1 初识 Delphi.....	1
练习 2 基本语句.....	5
练习 3 静态时钟.....	10
练习 4 动态时钟.....	14
练习 5 测试可变类型	18
练习 6 简单计算器（一）	22
练习 7 简单计算器（二）	25
练习 8 简单密码校验	29
练习 9 形状容器（一）	33
练习 10 形状容器（二）	36
练习 11 撞球（一）	39
练习 12 撞球（二）	41
练习 13 文本编辑器（一）	43
练习 14 文本编辑器（二）	47
练习 15 菜单的创建	51
练习 16 带菜单的文本编辑器.....	54
练习 17 MDI 应用程序（一）	58
练习 18 MDI 应用程序（二）	62
练习 19 日历制作.....	68

练习 20	日历制作 (二)	71
练习 21	碰不着的按钮	75
练习 22	列表拖拽	79
练习 23	格式化输入输出	84
练习 24	文件列表	88
练习 25	磁盘容量	92
练习 26	通用口令 DLL	95
练习 27	TeeChart 图表	100
练习 28	多线程处理	105
练习 29	AVI 播放器 (一)	109
练习 30	AVI 播放器 (二)	114
练习 31	网络浏览器	118
练习 32	学生档案管理 (一)	122
练习 33	学生档案管理 (二)	128
练习 34	创建数据库	135
练习 35	编辑数据库	138
练习 36	SQL 语言 (一)	142
练习 37	SQL 语言 (二)	148
练习 38	SQL 语言 (三)	153
练习 39	SQL 语言 (四)	157
练习 40	简单数据库窗体	161
练习 41	数据库编辑程序	165

练习 42 基于 Table 的查询	170
练习 43 查询范围	175
练习 44 查询范围	179
练习 45 查询范围	184
练习 46 建立服务器端数据库	190
练习 47 C/S 模式中客户端设计	195
练习 48 联机订票系统	202
练习 49 制作数据库报表	208
练习 50 简单的 Web 数据库	212

练习 1 初 识 Delphi

练习目的

本练习是一个简单的 Delphi 程序，通过它读者可以对 Delphi 有初步的认识，同时在本练习中还将介绍一些基本的 Delphi 知识，使大家对这个编程软件有所了解。本练习的效果如图 1-1 所示。

练习原理



图 1-1 Hello World 效果图

Delphi 是 INPRISE 公司最新推出的一种优秀的软件产品，一经推出，就风靡全球。Delphi 是一种先进的开发工具，它具有以下特点：

- 将真正的编译器与真正的可视化开发环境结合在一起。
- 拥有双向代码生成器。
- 完整的、一步到位的 ActiveX 支持。
- 拥有高性能的数据库访问结构。
- 支持跨平台的 Internet。

Delphi 将可视化技术和面向对象技术完美地结合在一起，突破了应用程序开发越来越复杂的障碍，让开发人员将注意力较多地集中在业务需求的开发上，而较少注意底层 API 调用、工程管理配置、代码导航等。

在 Delphi 的新版本 Delphi 5.5 中，INPRISE 公司仍让其使用者保持在技术的最前沿。Delphi 5 提供了众多的新特性来满足软件开发过程中各个阶段的需求：

- 最新改进的用户界面更加友好、易用。
- 先进的工程管理和代码浏览功能，大大减少了管理复杂工程及快速定位代码的麻烦。
- 功能强大的全新调试工具可以更快地发现更多的 bugs。
- 新增的程序设计语言特性，包括 Action Lists。
- 全面支持 Windows 98、Windows NT 和 Oracle 8。
- 完整地、一步到位地开发 COM（组件对象模型）和 CORBA 组件，一个组件支持 COM 和 CORBA 两种标准。
- 与 Java 的良好互操作性。

Delphi 使用 Object Pascal——一种真正面向对象的程序开发语言，可以方便地实现面向对象的开发，既不会使初学者感到困难，也不会使专业人员感到受限制。任何 C++ 可以完成的任务 Delphi 同样能够胜任，而且往往采取一种更简单更直接的方式。Delphi 5 提供了多种新的语言特性，使 Object Pascal 在处理各种程序开发时更加灵活。

Delphi 可以使开发人员创建一流的应用程序。最新的用户界面技术使用起来更简便、更高效。Delphi 完全支持 Microsoft BackOffice 技术，以使这些应用程序可以建立在 NT、MTS、SQL Server 和 Internet Information Server(IIS)上。Delphi 5 全面支持 Windows 98 的新用户界面元素。所有的 Delphi 代码生成工具均加入了 Windows 98 的新特性，使开发人员可以方便地开发基于 Windows 98 的应用程序。同时，Delphi 5 还包括一些新的组件，并支持 Windows 98 新的多监视器的功能。Delphi 5 提供对 Windows NT Services 的最佳支持：Service 是随着操作系统的启动而自动在后台运行的一种应用程序。到目前为止，使用 NT 这种功能强大的操作系统的特性意味着需要了解复杂的 API。现在，Delphi 5 让所有的开发人员都可以涉足 Service 的开发，它提供了组件、两种新向导以及惟一可视化的 Service 设计器。

最新版本的 Delphi 对数据访问和决策支持应用程序的开发具有许多优势，主要体现在以下几个方面：高性能的本地 SQL 可以访问 Oracle、Sybase、Informix、DB2、SQL Server 和 Interbase，同时直接支持本地和网络的 dBase、Paradox、Access、Foxpro 和 ASCII 格式。丰富的数据库组件和控件，包括多维表格和坐标系统、图形与报表以及可视化的数据库向导和可视化的设计工具，用于数据库访问、分析和显示。通过应用服务器和分布式组件模型，完全支持 client/server（客户机 / 服务器）和多层分布式数据库应用程序。

Delphi 的易用性可以从 Delphi 的基本形式以及面向对象编程这两个方面来理解。

从 Delphi 的基本形式和 Delphi 的面向对象编程的概念两个方面来看就知道了：

Delphi 的基本形式：Delphi 实际上是 Pascal 语言的一种版本，但它与传统的 Pascal 语言有天壤之别。一个 Delphi 程序首先是一个应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，仍可以严格地按照设计运行。用户的工作只是在“骨架”中加入自己的程序。缺省的应用程序是一个空白的窗体 (Form)，用户可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以被放大、缩小、移动、最大、最小化等，但用户却不用编写任何程序。因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。Delphi 已经为用户做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。用户所需要做的，只是在程序中加入完成自己所需功能的代码而已。

面向对象编程的概念：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。用户不必自己建立对象，只要在所提供的程序框架中加入完成功能的代码，其余的工作都交给 Delphi 去做。

练习过程

(1) Delphi 的安装与其他应用软件并无不同。2.0 版必须在 Windows 95 以上的操作系统中使用。启动 Windows 95 或 Windows NT 后，将 Delphi 的光盘放入光驱（CD-ROM）中，运行光盘上的 \INSTALL\SETUP.EXE 文件，它的安装程序会提示用户正确地装入 Delphi。

(2) 双击“开始”菜单中的 Delphi 程序项以启动 Delphi 应用程序，加载后会出现如图 1-2 所示的窗口。

(3) 首次加载 Delphi，屏幕上会出现四个窗口：

- 标题为“Delphi 5-Project1”的 Delphi 主窗口。
- Object Inspector 窗口。
- 标题为“Form1”的窗体（Form）窗口。
- 标题为“Unit1.PAS”的代码编辑窗口。刚启动时这一窗口的大部分被“Form1”窗体所掩盖。将“Form1”窗体移开，或单击 Form1 窗体下方的状态行，可以使其全部可见。在“Form1”窗体的任意可见位置上单击鼠标，可以恢复主窗口，使其可见。

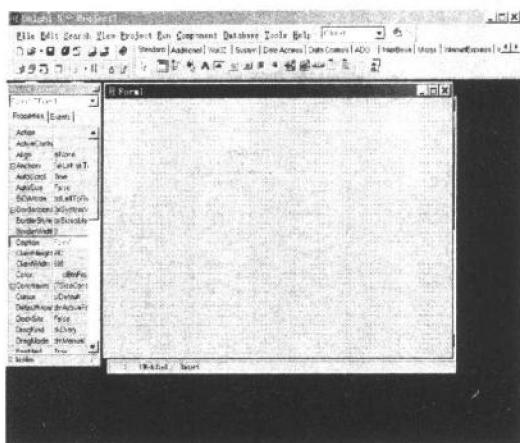


图 1-2 加载 Delphi 后出现的窗口

(4) Delphi 的主窗口位于屏幕的上端，包括 Menu（菜单）、Speed Bar（加速条）和 Component Panel（元件选项板）。Menu 是下拉式主菜单。Speed Bar 位于主窗口的左下端，由两排共 14 个加速按钮组成。这些按钮是菜单功能的快捷方式，各种图标直观地表示了它能执行的动作。Component Panel 由一行、若干页对象按钮所组成，利用它来选择需要的元件并将它放到窗体中去。

Object Inspector 窗口含有两标签页：Properties 标签页显示窗体中当前被选择部件的属性信息，并允许改变对象的属性；Events 标签页列出了当前部件可以响应的事件。按动 Object Inspector 下端的 Events 标签页，使得 Events 标签页可见，这一定的事件后边的空白处，可以定义对象接受到相应事件时执行的动作。首次启动时，Object Inspector 窗口显示的是当前窗体 Form1 的属性。Object Inspector 上还有 Object Selector（对象选择器），位于 Object Inspector 上方的下拉式菜单中，它显示了窗体上所有部件的名称和类型，也包含窗体本身。

Forms 窗口是开展大部分设计的工作区域。首次启动 Delphi 2.0 时显示的是窗体 Form1。可以把部件放在窗体中，通过移动位置、改变尺寸等操作随心所欲地安排它们，以此来开发

应用程序的用户界面。用户可以把窗体想象成一个可以放置其他部件的容器。窗体上有栅格 (Grids)，供放置部件时对齐位置用，在程序运行时 Grids 是不可见的。

代码窗口一开始处于窗体窗口之下。因为在 Delphi 中，设计用户界面直接在窗体中进行，运行结果和设计样板完全一致。当部件被放到窗体上时，Delphi 会自动生成大部分的用户界面代码。用户所应做的只是在生成的框架中加入完成所需功能的程序段而已。

(5) 下面来制作一个简单的程序，首先改变 Form 窗口的大小，可以直接拖动它的边缘，将其设置为所需的大小，也可以在 Object Inspector 窗口的属性标签页中设置 Form1 元件的 Width 和 Height 属性的数值。



图 1-3 放置好元件后的窗体

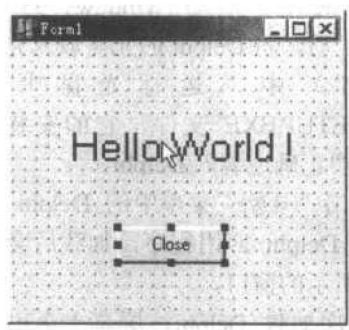


图 1-4 修改后的窗体

(6) 然后双击 Component Panel 上的 Label 元件和 Button 元件的快捷按钮，在 Form1 中加入这两个元件，调整好它们的位置。此时的窗体如图 1-3 所示。

(7) 选中 Label1，在 Object Inspector 窗口的属性标签页中设置 Font 属性，加大文字大小，将它的 Caption 属性改为 Hello World!。然后选中 Button1，在 Object Inspector 窗口的属性标签页中设置它的 Caption 属性为 Close。修改后的窗体如图 1-4 所示。

(8) 下面为 Close 按钮加入代码，双击 Close 按钮，在光标出现处加入代码：Close；

(9) 这样这个窗体就设计完成了。按下【F9】或快捷工具栏上的运行按钮，编译程序，得到的效果如本练习最前面的效果图 1-1 所示。

【练习小结】

本练习介绍了 Delphi 的基本特性以及 Delphi 5 的强大功能，无论是编程的初学者还是编程的熟练者，无论是编简单的应用程序还是分布式应用程序或 Internet 应用程序，Delphi 5 都是读者最佳的选择。可以说选择了 Delphi 5，读者的编程工作就成功了一半。

【思考题】

1. 熟悉 Delphi 编程界面。
2. 修改 Button 和 Lable 的 Font 属性和 Caption 属性，看看效果。
3. 查看本练习的代码。

练习2 基本语句

练习目的

在本练习中，将制作一个控制窗口（图 2-1）大小的小程序。这个小程序将使用直接赋值、判断和循环三种方式设定窗口的大小。

通过这个练习，读者将初步认识 Delphi，将对 Delphi 的基本语句有所了解。

练习原理

在事件处理过程中，最常用到的工作就是把一个新值赋给一个属性或变量。在设计用户界面时，可以使用 Object Inspector 来改变其属性；但有时需要在程序执行时改变属性的值，而且有些属性只能在执行时改变，这些属性在 Delphi 的在线帮助的 Property 主题中被标为执行期属性。进行这种改变，就必须使用赋值语句。下文的赋值语句表征一个 OnClick 事件。当按钮按动后，将编辑框部件 Edit1 的 Color 属性设置为 clRed:

```
procedure TForm1. Button1Click(Sender:
TObject);
begin
Edit1. Color : = clRed;
end;
```

当按动按钮后赋值语句执行，编辑框变成红色。在语句中，部件的名称在属性前，中间用“.”表示属性的所属关系。这样就准确地指定了要将 clRed 值赋给哪一部件的哪一属性。赋值号为“: =”，不论给属性还是给变量赋值，都是将右边的值赋给左边的属性或变量。当将一个属性值、变量、常量或文本数据赋给属性或变量时，所赋值的类型和接受此值的属性或变量的类型应相同或兼容。一个属性或变量的类型定义了此属性或变量的可能值集合，也定义了程序代码可以执行的运算。在前边的例程中，编辑框部件的 Color 属性和 clRed 的类型都是 TColor。

判断语句以 if 开头，它将计算一个表达式，并根据计算结果决定程序流程。if 保留字

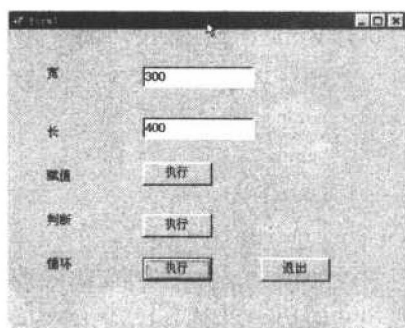


图 2-1 窗口

后跟随一个生成 Boolean 值为 True 或 False 的表达式。一般用“=”作为关系运算符，比较产生一个布尔型值。当表达式为 True 时，执行 then 后的语句。否则执行 else 后的代码，if 语句也可以不含 else 部分，表达式为 False 时自动跳到下一行程序。if 语句可以嵌套，当使用复合语句表达时，复合语句前后需加上“begin...end”。else 保留字前不能加“;”，而且，编译器会将 else 语句视为属于最靠近的 if 语句。必要时，须使用“begin...end”保留字来强迫 else 部分属于某一级的 if 语句。

Object Pascal 的循环语句有三种：repeat、while 和 for 语句。

首先看看 repeat 语句，它会重复执行一行或一段语句直到某一状态为真。它以 repeat 开始，until 结束，until 后跟随被判断的布尔表达式。如：

```
i := 0;
repeat
  i := i+1;
  Writeln(i);
until i=10;
```

当此语句被执行时，窗体的下方会出现 1 到 10 的数字。布尔表达式 i=10（与其他语言不同的是，“=”是关系运算符，而不能进行赋值操作）直到“repeat..until”程序段的结尾才会被计算，这意味着 repeat 语句至少会被执行一次。

while 语句和 repeat 语句的不同之处在于，它的布尔表达式在循环的开头进行判断。while 保留字后面必须跟一个布尔表达式。如果该表达式的结果为真，循环被执行，否则会退出循环，执行 while 语句后面的程序。下面的例程和上面的 repeat 例程达到同样的效果：

```
i := 0;
while i<10 do
begin
  i := i+1;
  writeln(i);
end;
```

for 语句的程序代码会执行一定的次数。它需要一个循环变量来控制循环次数。用户需要对变量进行说明，它的类型可以是整形、布尔型、字符型、枚举型或子界型。下面的程序段运行后会显示 1 到 5 的数字，i 为控制变量：

```
var
  i : integer;
for i := 1 to 5 do
  writeln(i);
```

以上介绍了三种循环语句。如果用户知道循环要执行多少次的话，可以使用 for 语句。for 循环执行速度快、效率比较高。如果用户不知道循环要执行多少次，但知道至少会执行一次的话，选用“repeat..until”语句比较合适。当用户认为程序可能一次都不执行的话，最好选用“while..do”语句。

练习过程

(1) 打开 Delphi 的执行程序，系统将自动新建一个空白窗体 Form1，它的大小是一定的。在旁边的对象监视器窗口中选择属性标签页，找到 Height（高度）属性，在右边的输入框中输入数值 400；找到 Width（宽度）属性，在右边的输入框中输入数值 450。

(2) 双击 Delphi 程序窗口上方的控件面板“Standard”标签页上的 Label 快捷按钮，在窗体中加入五个标签控件，用类似方法在页面中加入两个 Edit 控件和四个 Button 元件，将它们按照图 2-2 所示排列好。

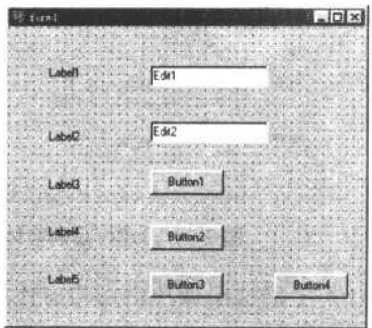


图 2-2 插入各个控件后的页面

(3) 下面设定各个元件的属性，将 Label1、Label2、Label3、Label4、Label5 的 Caption 属性分别改为宽、长、赋值、判断、循环。设置 Edit1 和 Edit2 的 Text 属性为“空”。设置 Button1、Button2、Button3 的 Caption 属性为“执行”，而 Button4 的 Caption 属性为“退出”，最后效果如图 2-3 所示。

(4) 下面使用赋值语句设定窗体的大小，双击赋值标签右边的“执行”按钮，在其中加入如下语句：

```
form1.Width:=strtoint(edit1.text);
form1.height:=strtoint(edit2.text);
```

这样在执行时，只要在长和宽输入框中输入所需的数值后点击“执行”按钮，就可以将窗体设定为所需大小。

(5) 下面使用判断语句为窗体设置大小，双击判断标签右边的“执行”按钮，在其中加入如下语句：

```
if strtoint(edit1.text)<400 then
    form1.Width:=400
else
    form1.Width:=600;
if strtoint(edit2.text)<400 then
    form1.height:=400
else
    form1.height:=600;
```

这样如果长或宽的输入数值小于 400，则将它们设定为 400；如果大于 400，则将它们设定为 600，如图 2-4 所示。如果设定宽为 300，则窗体实际的宽为 450；设定长为 800，而实际执行的结果窗体的长为 600。

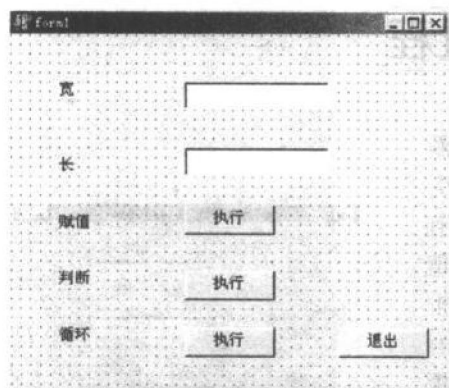


图 2-3 设定好后的窗体

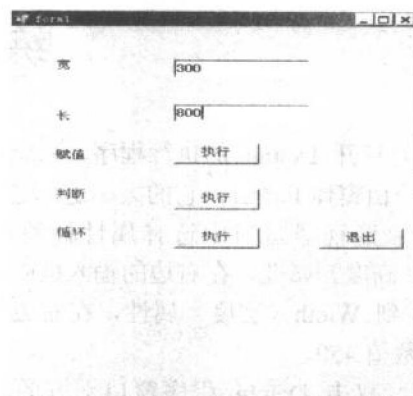


图 2-4 判断语句的执行结果

(6) 下面使用循环语句为窗体设置大小，双击循环标签右边的“执行”按钮，在其中加入如下语句：

```
procedure TForm1.Button3Click(Sender: TObject);
var i,m:integer;
begin
form1.Width:=1;
form1.height:=1;
if (strtoint(edit1.text)>=strtoint(edit2.text))then
    m:= strtoint(edit1.text)
else
    m:=strtoint(edit2.text);
for i:=1 to m do
begin
    form1.Width:=form1.Width+1;
    form1.height:=form1.height+1;
end;
end;
```

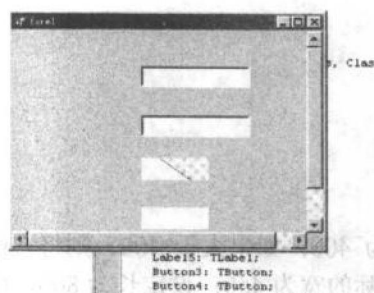


图 2-5 窗体增大的过程

在这里，先使用 Var 语句定义两个整型变量：i 和 m。通过两个对窗体 Width 和 Height 的属性设置将窗体的长宽初始化为 1，然后利用一个判断语句找出输入数值中大的一个，最后利用循环语句长和宽都设定为这个值。每次循环使长宽同时增加一个像素，所以当程序运行的时候就会出现很有趣的动态效果，图 2-5 显示了窗体增大过程中的一个瞬间。这样这个程序就完成了。

【练习小结】

本练习制作了一个简单的 Delphi 程序，通过它，读者应对面向对象的编程方法有了初步的了解。在本练习中，还学习了 Delphi 中赋值、判断、循环语句的语法。

【思考题】

1. 编写一个程序，实现对三个数进行排序的功能。
2. 编写一个程序，实现从 1 到 100 连加求值的功能。
3. 从头到尾浏览一遍本练习的程序清单。

练习3 静态时钟

练习目的

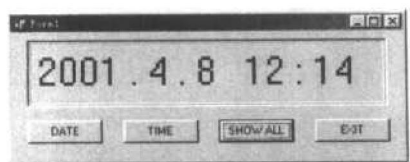


图 3-1 静态时钟

在这个练习中，将制作一个显示当前时间的小程序。在这个小程序中按下 **DATE**（日期）按钮或者 **TIME**（时间）按钮将会显示出该时刻的日期或者时间。由于它只能显示按下时刻的时间，所以称之为静态时钟（图 3-1），更复杂的时间处理将在后面的练习中谈到。

通过这个练习；读者将对面向对象的编程方法有所了解。

练习原理

Delphi 是全新的可视化编程环境，它是一种方便、快捷、强大的 Windows 应用程序开发工具。它使用了 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编程语言（Object-Oriented Language）。Delphi 是当今世界上最快的语言编辑器、最为领先的数据库技术。

所谓对象指的是编程中的一些通用元件，例如窗体、按钮元件等，用户无需了解如何生成这些元素，只需要知道这些元件的属性和方法并加以调用就可以了。面向对象的编程语言使得复杂的工作条理清晰、编写容易。结合 Delphi 可视化的编程界面，想要生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁，它将帮助用户轻松地完成。

在这个练习中，读者将了解到 Delphi 编程的一般顺序：首先建立一个窗体，在其中加入所需的元件对象，然后设定这些对象的属性，使其满足使用的需要；最后为各个控制元件加入程序代码，使其对触发的事件做出反应，以实现程序的功能。

练习过程

(1) 打开 Delphi 的执行程序，系统将自动新建一个空白窗体 Form1，它的大小是一定的。在旁边的对象监视器窗口中选择属性标签页，找到 Height（高度）属性，在右边的输入