源代码免费下载

商业开发
代码库系列

Delphi

案例开发集锦

（第二版）

陆　岚　黄显堂　康祥顺　编著
张　强　审校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

商业开发代码库系列

Delphi案例开发集锦
（第二版）

陆　岚　黄显堂　康祥顺　编著

张　强　审校

電子工業出版社

Publishing House of Electronics Industry

北京・BEIJING

内 容 简 介

本书主要通过具体的案例，介绍如何运用Delphi.NET开发工具开发实际的应用程序，涉及的范围从基本应用到高级处理，包括界面设计、图像处理、多媒体应用、系统文件处理、硬件控制、基本数据处理、系统控制、网络处理、ASP.NET Web程序设计、数据库应用编程、综合案例等。每个案例的编排都严格按照读者的阅读习惯进行组织，由具有丰富经验的项目开发人员亲手编写，大部分案例都已经在项目开发中经过了实践的检验，是指导读者进入实战型程序设计师领域的良师益友。

本书适用于大中专院校学生、程序设计人员、Borland Delphi产品爱好者以及.NET平台设计的追随者阅读参考。

图书在版编目（CIP）数据

Delphi案例开发集锦/陆岚，黄显堂，康祥顺编著.—2版.—北京：电子工业出版社，2008.1
（商业开发代码库系列）
ISBN 978-7-121-05065-7

Ⅰ. D… Ⅱ. ①陆… ②黄… ③康… Ⅲ. 软件工具—程序设计 Ⅳ. TP311.56

中国版本图书馆CIP数据核字（2007）第145804号

责任编辑：李 莹
印 刷：北京天竺颖华印刷厂
装 订：三河市金马印装有限公司
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编：100036
北京市海淀区翠微东里甲2号 邮编：100036
开 本：787×1092 1/16 印张：23.125 字数：590千字
印 次：2008年1月第1次印刷
定 价：35.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

反侵权盗版声明

前　言

你需要阅读本书吗

Delphi是Borland公司推出的优秀的前端开发工具。自Delphi问世以来，其友好的集成开发界面、可视化的双向开发模式、良好的数据库应用支持以及高效的程序开发和程序运行，备受广大程序设计师的好评。

自从Delphi 1.0发布以来，我就被这款工具所吸引，迄今为止，已经十年有余了。伴随着每次新版本的发布，我总会在第一时间收到Borland公司的测试盘，随之而来的是欣喜若狂地安装和查看新的特性，久久舍不得放下，因为，它给我的工作带来了愉快，极大地简化了设计，提高了开发的效率，这样的工具为何不用呢？我不是真正的程序员，因为我从没有认真地去学习和钻研C++，也没有时间和空间允许我去研究这些真正程序员的技术，但是，我足够聪明，所以选择了Delphi。

伴随着微软的.NET Framework技术的升温，人们对.NET技术的学习越来越重视。同样，Delphi也发现了软件技术的革命性潮流和趋势，率先发布了支持.NET开发的Delphi 8.0——纯粹的支持.NET开发的平台，随后又发布了融合Delphi Win32、Delphi for .NET、Visual Basic.NET的Delphi 2005，但是这些版本的推出都显得很仓促，不是十分的稳定和成熟，对于Delphi的销售没有什么显著的推动作用。痛定思痛，Borland公司经过努力，开发了集所有支持.NET开发的系统平台Delphi 2006，该版本经过四次测试才最终公诸于众，可以说是用心良苦，Borland公司的真正目的是想将这一版本做成当前最为完善的、最为便利的.NET开发平台，让用户不离开Delphi开发环境就能够实现所有的.NET开发，这在当前来说，是任何其他软件平台都没有实现的。

随着Delphi技术的进步和发展，它必将成为今后几年中最佳的编程语言。

学习一门编程语言，最为关键的不是去死记函数或者方法，而是掌握如何应用这些函数、方法去解决实际的问题。在学习的过程中，我们很可能都有过这样的经历：这门编程语言的函数、方法我也知道不少，利用函数、方法也能够编写几个简单的小程序，但是，当我再翻看别人的书或者使用别人的小软件时才会发现，原来这个函数还能这样用。这就是我们自己钻研所不能取得的成果，而且会事倍功半；然而本书汇集了Delphi.NET编程语言解决方案的所有经典实例，无论是界面设计，还是网络处理，无论是简单的函数应用，还是高级的底层API函数的应用，都包括在本书之中。阅读本书后，你能够在最短的时间内，迅速提升自己的编程水平以及解决问题的方式、方法，避免闭门造车所造成的低水平徘徊，解除因失去信心而带来的心理负担。因此，如果你已经开始学习Delphi并具有了一定的Delphi语法基础，那么本书将是引领你进入真正的专业级程序员行列的指南性图书，它将会使你在.NET平台下的开发水平发生质的飞跃！

本书涵盖的内容以及编排格式

本书共9章，每一章都由既相互独立、又相互间有一定逻辑关系的案例组成。每一个案例都分为4个部分编写，包括：

- 案例运行效果与操作——让读者对本案例有一个直观的印象，以便于读者进一步了解如何解决这个问题；
- 制作要点——让读者明白，要解决这个问题，应该采取什么方法，使用哪些函数、组件和技巧；
- 步骤详解——告诉读者如何一步一步地实现这个案例，以便于读者自己动手实现案例效果，达到加深理解、锻炼自我的目的；
- 程序源代码与解释——附带了所有的该案例的源代码，读者只要将这些源代码复制到自己的程序中就能够实现程序的功能，以方便读者进行验证；同时对于程序中应用到的难以理解的函数或者对象都添加了详细的解释，以便于读者弄清楚问题的根源。

这些案例的源代码，既可以用来试验某个编程的概念，也可以剪贴到自己的程序中，实现某种特定的功能，达到迅速解决问题的效果。通过阅读本书，将学习到以下几个方面的知识：

- 学会如何设计专业级的用户界面；
- 学习如何在程序中处理多媒体编程的问题；
- 学会在程序中处理系统文件；
- 学会如何在程序中控制系统硬件和外围设备；
- 学会如何在程序中调用Windows的底层API函数；
- 学会如何在系统中处理ADO.NET数据库；
- 学会如何开发和设计专业级的ASP.NET应用系统；
- 学会如何应用网络组件和API函数处理网络操作；
- 学会如何在程序中处理复杂的跨平台的应用问题；
- 学会如何应用Delphi.NET提供的组件、函数、对象实际开发功能复杂的应用系统，包括单机应用系统、B/S结构的应用系统、C/S结构的应用系统，让读者了解实际开发一个应用系统的过程；
- 学会.NET在支持图形图像和多媒体开发方面的GDI+类库的特性，让读者真正认识到利用GDI+开发多媒体程序的工作是多么的简单和快捷；
- 一个完整的应用Delphi 2006开发的ASP.NET门户网站，全面展现了应用.NET技术开发多层结构体系的应用程序的技巧，可以让读者在系统学习前面的案例后迅速成为一名专业的.NET设计人员；
- 其他更多的内容。

本书配套资源的使用

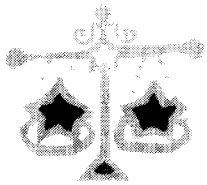
为了方便读者阅读、测试和使用本书提供的案例，本书每个案例的源代码和相关素材都打包后上传到华信教育资源网。源代码严格按照章节编排，例如第1章第一个案例的源代码存放位置为：程序源代码\第1章\福利彩票机选模拟器，每个案例目录下附带了所有用到的资源文件，如图片、数据表等。每个案例都是作者在Windows XP+Borland Delphi Studio 2006 for .NET下进行严格测试后再组织起来的，请尽管放心使用。值得注意的是，有些程序需要搭建特殊的运行环境，如数据库服务器、ASP.NET应用程序，这些案例的运行需要特殊的设置，请参照相关案例下的Readme.txt说明文档。

但是，鱼和熊掌不能兼得。虽然方便了读者，但同时很可能也使读者对代码的熟悉程度、技术要点和编程思路等问题的领会程度降低。因此，强烈建议读者先看看书中案例的运行效果，然后再看源代码，如果心中已经有数了，可以自己动手举一反三，设计类似的程序来检验自己的想法，或者对程序进行修改和改进。

本书每一章内容都是由具有丰富经验的实际从事项目开发的人员编写的，大部分的案例都已经在项目开发中得以运用，因此其操作性、实践性和指导性都相当强，这也是当今市面上难得的近乎实战的专业级案例图书，也是迄今为止内容全面、复杂、技巧性很强的Delphi.NET专业案例图书。参加本书编写工作的有：刘东鸿、张庆、崔竞、董军、保春艳、李欣、梅光耀、张小丽、康祥琴。这些同志虽然从事着繁重的项目开发工作，但是仍然在工作之余，兢兢业业、严谨认真地编写、测试了所有案例，保证了本书的质量和进度。

另外，在本书的编写过程中，电子工业出版社和北京美迪亚电子信息有限公司的全体员工也付出了辛勤的劳动，在此一并表示最诚挚的谢意！本书中的所有案例，编者都在Delphi 2006的开发环境中进行过测试，由于水平有限，尽管做了严格的审核和测试，书中难免仍有一些错误，敬请广大读者不吝赐教，编者在此表示感谢！

目　录



第1章

基本数据处理

本章内容

- ☆ 福利彩票机选模拟器
- ☆ 冒泡排序法
- ☆ 线程同步——存款机
- ☆ 汉诺塔数学智力游戏
- ☆ 字符串查找、替换
- ☆ 中英文字符数统计
- ☆ 将小写金额转换为大写金额
- ☆ 数值与字符串的相互转换

案例1：福利彩票机选模拟器

案例运行效果与操作

随机数是指产生不规则结果的机制，在程序中用于产生不确定的数值，它的原理就是从随机数表中随机抓取一个随机数，在Delphi for .NET中，可以使用System.Random类产生随机数，也可以使用Random()函数产生随机数。在使用Random()函数产生随机数时，在使用随机数前必须使用Randomize类重新设置随机数表的种子，这样才能够产生不同的随机数，否则会产生相同的随机数。随机数在制作游戏等项目中用得比较多。本案例设计了一个随机产生福利彩票中奖号码的程序，主要演示了怎么使用随机数，另外还显示了数组的操作技巧。程序运行效果如图1.1所示。

制作要点

1. Randomize类的应用。

该类用于初始化随机数表，以系统计时器为种子，下一个随机数以上一个随机数为种子，因此不会重复，如果使用相同的种子，则将会产生重复的随机数，这可能是不安全的。

2. Random()函数的应用。

用于产生一个0～1之间的随机数（大于等于0但是小于1），如果需要产生一个大于等于M小于等于N的随机数，可以这样写：Random()*(N-M+1)+M。

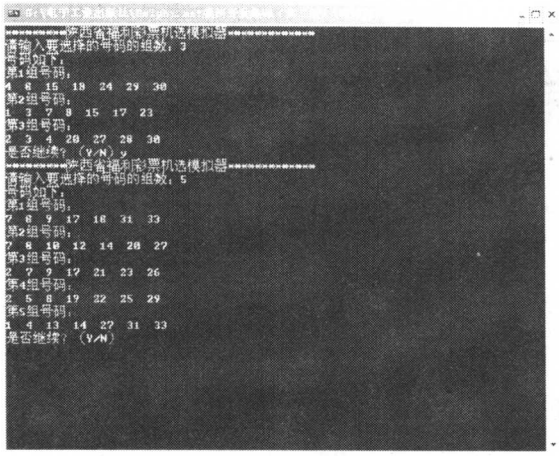


图1.1 机选号码结果

3. System.Random类的应用。

在Delphi for .NET中，随机数使用System.Random对象产生，该对象为了不产生重复的随机数，采用与时间对象绑定的策略，在该对象的构造函数中，可以指定一个产生随机数的范围值，例如，需要产生一个介于1～33的随机数，就可以在产生随机数的函数Next()中这样指定：System.Random.Next(1,34)，因为随机数产生一个大于等于最小值但是小于最大值的随机数。

4. Array数组对象的应用。

该对象用于对数组执行操作，如排序、查找、删除等。在Delphi for .NET中，如果需要将Delphi中的对象声明为.NET对象，必须在对象前添加“&”关键标识符，该类可以方便地执行数组的操作。

步骤详解

在Borland Developer Studio 2006 for .NET开发环境中，选择Delphi for .NET节点，新建一个“Console Application”控制台应用程序。

程序源代码与解释

```
program Project2;
{$APPTYPE CONSOLE}
uses
  sysutils;
 procedure GetNum(n:integer);
var
    TempNum:integer;
    Result:array [0..6] of integer;
    i,j,x:integer;
     MyRDM:System.Random;
begin
MyRDM:=System.Random.Create();
for i:=1 to n do      //产生指定组数的随机数序列
begin
```

```
    console.WriteLine('第'+i.ToString+'组号码：');
    j:=0;
    while j<7 do      //指定产生的随机数个数
    begin
          TempNum:=MyRDM.Next(1,34);  //产生一个介于1~34之间的随机数
          //不允许重复
          if system.&Array.IndexOf(Result,TempNum as system.&Object )>-1 then
          j:=j
          else
          begin
          result[j]:=tempnum;
          j:=j+1;
          end;
end;
system.&Array.Sort(Result);  //对随机数元素排序
//在控制台中输出经过排序的随机数
for x:=low(Result) to high(Result) do
console.Write(Result[x].ToString +'  ');
console.WriteLine('');
end;
//console.ReadLine;
end;

//控制台程序的主程序
var
     Num:integer;
     Answer:string;

begin
Answer:='Y';
while (Answer='Y') or (Answer='y') do
begin
 { TODO -oUser -cConsole Main : Insert code here }

 console.WriteLine('*********陕西省福利彩票机选模拟器*************');
 console.Write('请输入要选择的号码的组数：');
 Num:=strtoint(console.Readline); //必须使用ReadLine,否则将返回当前键的ASC值
 console.WriteLine('号码如下：');
 GetNum(Num);
 console.Write('是否继续？（Y/N）');
 Answer:=Console.ReadLine;
end;

end.
```

案例2：冒泡排序法

案例运行效果与操作

排序是依据数据的特性，由大到小或者由小到大排列数据的次序。知名的排序方法有冒泡排序法（Bobble Sort）、选择排序法（Selection Sort）、插入排序法（Insertion Sort）、Shell排序法等。当数据量比较大时，合适的排序方法会影响程序的执行效率。本案例采用冒泡排序法实现了将随机产生的数据进行排序，然后在控制台中输出。程序运行效果如图1.2所示。

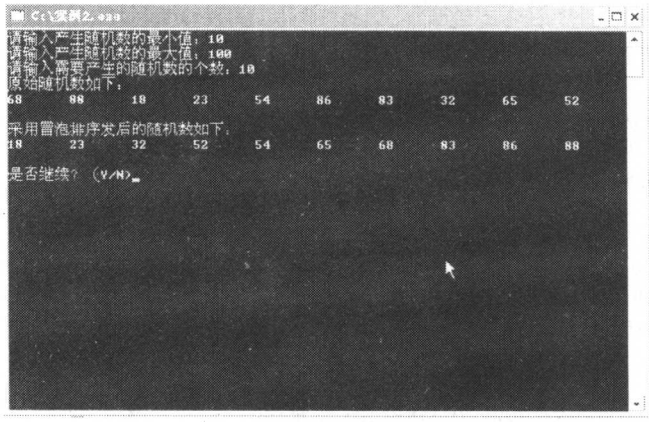


图1.2 冒泡排序法

程序运行以后，首先要求输入随机数的最大值和最小值，如果最小值大于最大值，或者，最小值、最大值不是数值，则要求重新输入；然后要求输入产生随机数的个数，接着会产生原始随机数和排序后的随机数。执行完后，程序会提示是否继续，回答Y则继续，回答N则退出。

制作要点

1. System.Random类的应用。

该类用于产生随机数，它提供一个方法Next(Range:Integer)；该方法将产生指定范围内的整数，例如，需要产生一个介于30～40之间的整数，就可以这样指定范围：

Next(30,40+1);

因为产生的整数将是大于等于最小范围而小于最大范围的值。

2. 冒泡排序法的应用。

冒泡排序法是拿前一个元素与它后面的所有元素进行比较，如果大于后面的元素，则交换位置，所以最小的数始终在最前面的位置。实际上，.NET专门提供了数组排序方法：

Array.Sort(数组名)

该方法将一维数组按照从小到大的顺序排列，如果是二维以上的数组，必须先转换为一维数组再进行排序操作，这可以采用下列方法：

Array.Sort(数组名1，数组名2)

步骤详解

在Borland Developer Studio 2006 for .NET开发环境中，选择Delphi for .NET节点，新建一个“Console Application”控制台应用程序。

程序源代码与解释

```
program Project1;
{$APPTYPE CONSOLE}
uses
  SysUtils;
```

```
//冒泡排序函数，注意一定要将函数参数设置为VAR输入输出类型
Procedure BobbleSort(var List:array of Integer);  //冒泡排序
var
   temp:integer;
   i,j:integer;
begin
        //开始冒泡排序
        For i :=low(list) To high(list) do
        begin
             For j := i + 1 To high(list)  do//比较剩下的元素
             begin
                If List[i] >List[j] Then //依次与第一个元素比较
                begin
                     temp := List[j];
                     //如果后面的元素比第一个元素小，则交换位置
                     List[j] := List[i]; //在第j个元素中存放第i个元素
                     List[i] := temp //在第i个元素中存放第j个元素
                end;
            end;
        end;
end;
//产生随机数的函数
    Procedure RandomNum( out list:array of integer;min:Integer; max:Integer; num:Integer);
    var
       i,temp:integer;
       Myrdm:System.Random;
       Mylist:array of integer;
    begin
        //min为随机数的最小值
        //max为随机数的最大值
        //number为随机数的个数
        setlength(mylist,num);
        Myrdm:=System.Random.Create;
       // n:=number;

        For i := 0 To num-1 do
        begin
            temp :=Myrdm.Next(min,max+1); //产生随机数
            mylist[i]:= temp; //存放随机数
        end;
        list:=mylist;
    end;

 var
    i,j:integer;
    min:integer;
    max:integer;
    number:integer;
    NewList:array of integer;//保存临时产生的数组元素
    answer:string;
begin
  { TODO -oUser -cConsole Main : Insert code here }
    answer:='y';
```

```
    while (answer='y') or (answer='Y') do
    begin
        Console.Write('请输入产生随机数的最小值：');
        min :=Convert.ToInt16(Console.ReadLine);
        Console.Write('请输入产生随机数的最大值：');
        max := Convert.ToInt16(Console.ReadLine);
        If min > max Then
            Console.WriteLine('最小值一定是小于最大值，请重新输入！');
        Console.Write('请输入需要产生的随机数的个数：');
        number :=Convert.ToInt16 ( Console.ReadLine);
        SetLength(NewList,number);//重新分配数组空间
        RandomNum(newlist,min, max, number);
        Console.WriteLine('原始随机数如下：');
        For i := 0 To high(newlist) do
            Console.Write(inttostr(newlist[i])+'  ');
        Console.WriteLine('');
        Console.WriteLine('采用冒泡排序法后的随机数如下：');
        BobbleSort(newlist);
        For j := 0 To high(newlist)  do
            Console.Write(inttostr(newlist[j])+'  ');
        Console.WriteLine('');
        Console.Write('是否继续？（Y/N)');
        answer := Console.ReadLine;
    end;
        Console.WriteLine('单击Enter键退出...');
        Console.Read()
end.
```

案例3：线程同步——存款机

案例运行效果与操作

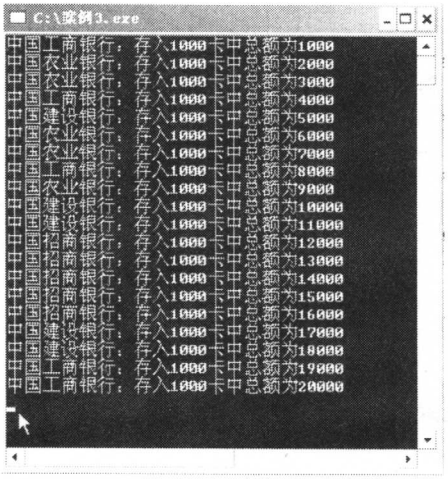


图1.3　随机存款的结果

本案例模拟了一个存款机的现实情况：一个用户随机地在4台不同的银行存款机上分5次存钱，每次存入1000元，在存储的过程中，存款机上需要显示此时卡中的存款总额。通过这个案例，可以学习到如何让多个线程保持同步。程序运行效果如图1.3所示。

为什么需要使用同步机制呢？你可以试一试，如果不使用同步，可能最后的存款总额为5000元，因为它们使用了各自的存款总额变量。

制作要点

1. Mutex类的应用。

该类可以让多个线程之间保持同步，它可以限制同一时间只能有一个线程对象运行它的方

法，其他的线程如果需要运行它的方法，就必须等到此方法空闲时才能够运行。该类的两个方法可以控制线程什么时间处于闲置状态，什么时间处于运行状态：WaitOne和ReleaseMutex，前者可以让线程进入工作状态，后者可以让线程释放资源给别的线程工作。

2. 静态变量的声明。

在类声明中，可以通过Class Var关键字声明类的静态变量，这个变量相当于全局变量。

步骤详解

在Borland Developer Studio 2006 for .NET开发环境中，选择Delphi for .NET节点，新建一个“Console Application”控制台应用程序。

程序源代码与解释

```
unit AutoMoney;

interface
 uses
    system.Threading;
type
  TMoney = class
  private
    { Private Declarations }
    name:string;//存款机名称
  public
    mut1:Mutex;    //同步基元对象
    total:integer; //卡中总额
    constructor Create(money_name:string);
    procedure start_save;
    procedure save_money(n:integer);
  end;

implementation

constructor TMoney.Create(money_name:string);
begin
  inherited Create;
  // TODO: Add any constructor code here
  name:=money_name;
  mut1:=Mutex.Create();
end;

procedure TMoney.save_money(n: integer);
var
  sum:integer;
begin
    mut1.WaitOne(); //等待唤醒
    Randomize;
    sum:= total + n;
    total := sum;
    Console.WriteLine(name + ': 存入' + n.ToString +'卡中总额为'+ total.ToString);
    System.Threading.Thread.Sleep(Random(500));
    mut1.ReleaseMutex(); //完成后释放给别的线程工作
end;
```