



编程实例五十讲丛书

突破

Delphi 6.0

编程实例

五十讲

陈志华 等编著



附光盘



中国水利水电出版社
www.waterpub.com.cn

万水编程实例五十讲丛书

突破 Delphi 6.0

编程实例五十讲

陈志华 等编著

 中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书通过实例来说明 Delphi 编程通常的编程思路以及具体方法。对每个实例从编程思路和代码实现进行详细分析和讲解,力求带领读者快速地提高自己的程序设计水平,并达到举一反三的目的。很多实例的重要部分都做了特别注意和技巧提示,目的是强调技巧的使用或使读者避免一些常见错误,以便更高效地利用 Delphi 这个强大的程序开发工具。

本书精选了五十多个经典的实例从易到难、由浅入深、由简单到综合进行讲解,内容涉及面也很广泛,基本上涵盖了用 Delphi 进行应用程序设计的方方面面,凝聚了作者多年的 Delphi 编程经验,相信对启发读者的思想并提高读者的编程水平会有很大帮助。

本书主要面向中高级读者,但对那些熟悉 Object Pascal 语法并有一定编程经验的初级读者也很有帮助。

图书在版编目(CIP)数据

突破 Delphi 6.0 编程实例五十讲 / 陈志华等编著. —北京:中国水利水电出版社, 2002

(万水编程实例五十讲丛书)

ISBN 7-5084-1120-X

I.突… II.陈… III.DELPHI 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2002)第 037585 号

书 名	突破 Delphi 6.0 编程实例五十讲
作 者	陈志华 等编著
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn
经 售	电话: (010) 68359286 (万水) 63202266 (总机) 68331835 (发行部) 全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 23.75 印张 524 千字
版 次	2002 年 6 月第一版 2002 年 6 月北京第一次印刷
印 数	0001—5000 册
定 价	42.00 元(含 1CD)

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

《万水编程实例五十讲丛书》

编 委 会

主 编 万 博 王 龙 韩中领

编 委 苏 瑞 索双有 袁 博 陈海亮 黎 骅 康通博

陈 飞 郭燕婷 关 宁 张美生 文 锋 娄俊杰

杨现青 胡晓冰 商 铮 特 铮 赵庆元 姜仁武

王 静 陈心怡 陈江龙 谢小卓 陈志华 顾卫华

吴志华 王志伟 邹振亚 刘 杨

丛书序

一个优秀的程序员和一个普通程序员的区别在什么地方？有一个答案是这样的：普通程序员花了一个星期，苦思冥想如何实现程序功能和排除 Bug。而优秀程序员只用了一天就做完了，因为他给已经做过类似东西的朋友打了个电话。

这段话多半是玩笑，但是也说明了交流的重要。很多读者在学习编程的过程中，早已摆脱了教条式的入门讲解。他们需要的是前人的经验和总结，这就需要交流，看更多的实例，学更多的实用技巧。

如何把前人摸索出来的经验和套路传授给急需它们的读者，如何使计算机的入门教育变成更具有艺术性的教育，是我们一直在寻求的突破口。为此，我们策划并编写了“编程实例五十讲”丛书，从写作到排版风格，从整体结构到每个小例子，我们都做了精心的安排。

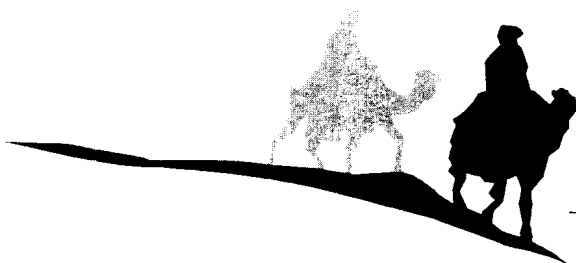
在本丛书的编写过程中贯穿了如下的原则：

- (1) 由浅入深、由易到难。整本书结构清晰，适用于初级和中级读者。
- (2) 内容实用、可操作性强。书中对于实例程序行和操作过程的描述要求清楚透彻、简明扼要，编写风格则要求“功能性强”、“示范性强”。
- (3) 知识讲解与动手操作并行。每个实例都突出讲解了某些知识点和使用技巧。知识点与实例的结合是本丛书的一大特点。
- (4) 简单常用的基础实例与综合性较强的应用实例相结合。每本书中都讲解了一些经常会碰到的应用实例，给读者提供思路上的参考。
- (5) 不做简单的功能介绍。本丛书是侧重于应用的图书，我们假定丛书的读者对软件已有一定的了解、或是计算机基础较好，需要快速地在实践中掌握软件的应用。

愿本丛书能带领读者进入程序设计的广阔天地。

编者

2002 年 4 月



我们寻求突破

前 言

2001年5月, Borland的年度巨献 Delphi 6.0 终于在世人的面前揭开神秘的面纱。Delphi 6.0 不但对于 Borland 来说是一个重要的产品, 事实上对于现在的 Windows 平台也是一个意义重大的产品。因为当 Microsoft 全力发展 .NET 虚拟环境, 而 Borland 本身也往 Linux 平台上开发 RAD 工具 Kylix 的时候, 由 Borland 窗口开发小组所研发的原生视窗开发工具正好填补了仍占多数 Windows 程序员的需要。

由于信息技术不断快速进步, 许多 Windows 程序员也不断需要在应用系统中加入最新的技术, 或是使用这些新的技术开发下一代的应用系统。例如 XML/XSL 的盛行, 让程序员对于如何处理和发挥 XML 的技术有强烈的需求, 例如 SOAP 的逐渐成为远程和分布式计算的标准, 因此程序员也需要使用 SOAP 来开发或是整合应用系统。更别说是由 SOAP 和 WSDL/WSML, UDDI 等技术所衍生出的最新 Web 应用系统标准 Web Service 等。对于 Windows 平台上的程序员而言, 即使是使用现在的开发工具, 例如 Delphi 4/5 或是 VB 等也仍然可以利用这些新的信息技术开发应用程序, 但是如此做不但有一定的难度, 而且可能旷废时日。

Delphi 6.0 刚好可以帮助企业和程序员解决这些信息技术上的困扰, Delphi 6.0 不但在传统的开发能力方面有持续的进步, 让程序员能够享有更具生产力的开发环境, 对于最新的信息技术, 例如 XML/XSL, SOAP, Web Service 等都有非常良好的支持, 让程序员能够快速地使用这些新的技术。此外, Delphi 6.0 在核心函数库以及编译器方面也有了幅度不小的改善, 它不但可以使程序员开发功能更强的基于 Windows 平台的应用程序, 也可以使所开发的程序能平稳地移植到 Linux 平台; 通过使用跨平台的最新可视化组件 Framework CLX, 即使是 Windows 平台的程序员也能够快速地开发基于 Linux 平台上的应用程序。

本书是一本通过讲解 Delphi 实例, 来说明使用 Delphi 编程时通常采用的编程思路以及具体方法的计算机书籍。通过对每个实例从编程思路和代码实现的详细分析和讲解, 力求带领读者快速地提高自己的程序设计水平, 并达到举一反三的目的。

本书的内容主要是面向中高级读者, 所以读者最好具有一定的 Delphi 程序设计基础, 如熟悉 Object Pascal 的语法, 有一定的程序设计经验, 最好有一定的项目经验。但初级读者肯定也能从中学到很多新颖的东西, 包括设计思想和技巧等, 不过这需要读者在遇到不懂的地方多多参考 Delphi 提供的帮助以及 Microsoft 的 MSDN。

本书精选 50 多个经典或有独特的设计思想或需要一些编写技巧的实例来进行分析讲解。实例共分七篇, 前六篇是基础实例, 主要讲解界面、多媒体、网络、数据库、系统等方面的基础性技巧; 最后一篇综合应用主要讲解了综合使用以上各篇技术所编写的比较大的实例, 提高读者对前面讲解的技术的综合应用能力。各篇例子分布较为均匀, 内容有浅有深,

既适合初学者入门学习，又可以给高级读者提供参考。

随书光盘中附带所有实例的源程序，读者可随时参看程序运行结果以加深对代码的理解。

很多实例的重要部分都有注意、技巧等特殊说明，目的是加深读者的印象，避免不必要的错误，以使读者可以更高效地利用 Delphi 这个强大的程序开发工具。

本书的实例内容涉及面比较广泛，基本上囊括了通常程序设计中的各个方面，凝聚了作者多年的 Delphi 编程经验，相信对启发读者的思想并提高读者的编程水平会有很大帮助。本书立足经典但更突出新颖，而且绝不是简单的控件的使用。

在本书的编写过程中，得到很多同行人士和朋友的热情帮助，为本书的编写提供了部分宝贵资源，再次表示由衷的感谢。

由于时间仓促、作者水平有限，书中未免会有不足和疏漏之处，敬请广大读者提出宝贵意见。

作者

2002年4月于北京

第一篇

基础入门

本篇导读

作为本书的开头,本篇主要向读者介绍使用 Delphi 编程的入门知识,主要着眼于熟悉 Object Pascal 语言的基本语法以及 Delphi 简单控件的使用。

本篇向读者演示了三个很基础的实例:一个简单的计算器、一个 21 点的小游戏和一个猜数字游戏。通过三个实例的学习和讲解,我们希望读者了解 Delphi 编程的一些基本思想。

如果你已经有一定的编程基础,那么本篇可以跳过。对于新手,还是建议认真地学习一下,熟悉最基本的 Delphi 内容。

第一讲 一个简单的计算器

【实例说明】

本例的任务是创建一个简单的计算器，实现两个有理数之间加、减、乘、除的基本运算。有理数的输入是通过鼠标单击数字按钮来实现，并使用一个文本框（edit1）来显示输入的有理数和运算结果。

【编程思路】

该程序是一个常见的示例程序。Windows 操作系统一般都自带有一个简单的计算器，如图 1.1 所示。



图 1.1 Win98 自带的计算器

本例仅实现其中加、减、乘、除的功能，读者可在学习完本节后尝试自己实现其他的一些简单运算功能。本讲为一个介绍性的例子。读者可在本讲中再温习一下 Pascal 语言。

Delphi 的编程语言是以 Pascal 为基础的。Pascal 语言具有可读性好、编写容易的特点，这使得它很适合作为基础的开发语言。同时，使用编译器创建的应用程序只生成单个可执行文件（.EXE），正是这种结合，使得 Pascal 成为 Delphi 这种先进开发环境的编程语言。

Object Pascal 的跳转语句有 if 和 case。

本例步骤 2 中用到的就是 if 语句。if 语句会计算一个表达式，并根据计算结果决定程序流程，if 保留字后跟随一个生成 Boolean 值 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 if 的语句。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。

步骤 6 中用到的是 case 语句。case 语句适用于被判断的变量或属性是整型、字符型、枚举型或子界型时 (LongInt 除外)。用 case 语句进行逻辑跳转比编写复杂的 if 语句容易阅读, 而且程序代码执行较快。

在编写自己的 Object Pascal 程序时, 要注意程序的可读性。Pascal 语言是英式结构语言, 在程序中选择合适的缩排、大小写风格, 并在需要时将程序代码分行, 会使得程序代码易懂。一般的程序员都有这样的体验: 如果不给程序加上适当的注解, 一段时间后, 自己也难以理清程序的流程。给程序及时地加上注释是良好的编程习惯。Delphi 的注释需要加在“{}”之间, 编辑器会把它们处理成为空白。Delphi 保留了 Borland Pascal 编辑器的风格, 关键字采用黑体字, 被注释的部分会变暗, 这使得编程风格良好、易读易写。

【步骤】

运行 Delphi 6.0, 选择 File→New→Application 新建一个工程。

1. 在 Form1 上建立一个 Edit 控件和 18 个 Button 控件, 分别设置属性如下:

```
Edit1.text:='';
Button2.text:='2';
Button4.text:='4';
Button6.text:='6';
Button8.text:='8';
Button10.text:='0';
Button12.text:='清除';
Button14.text:='-';
Button16.text:='/';
Button18.text:='退出';
Button1.text:='1';
Button3.text:='3';
Button5.text:='5';
Button7.text:='7';
Button9.text:='9';
Button11.text:='.';
Button13.text:='+';
Button15.text:='*';
Button17.text:='=';
```

控件排布如图 1.2 所示, 同时定义如下变量:

```
var
  x,y: real;
  flag: char;
```

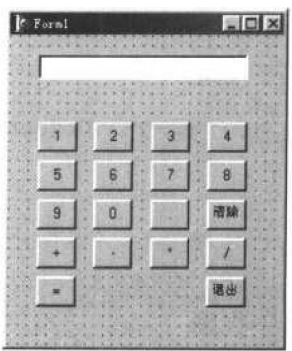


图 1.2 控件排布

2. 双击 Button1 按钮，输入如下过程：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if edit1.Text='0' then edit1.Text:='1'
  else
    edit1.Text:=edit1.Text+'1';
end;
```

过程与函数是程序中执行特定工作的模块化部分。Delphi 的运行库包含许多过程与函数以供应用程序调用。您不必了解过程与函数的逻辑，但要知道过程与函数的用途。在对象中说明的过程和函数称为方法（Method）。所有的事件处理过程都是过程，以保留字 `procedure` 开头。每一个事件处理过程只包含了当这一事件发生时需要执行的程序代码。在事件处理过程中使用 Delphi 已经存在的过程与函数，只需在程序代码中调用它们即可。

本段代码执行的过程为：程序运行时，单击 Button1 按钮，如果文本框中显示为 0，则以 1 代替。如果文本框中非 0，则在原数字后加 1。用同样的方法编写 Button2 到 Button10 的代码。添加的数字分别为 2, 3, 4, 5, 6, 7, 8, 9, 0。

3. 双击 Button11 按钮，编写如下代码：

```
procedure TForm1.Button11Click(Sender: TObject);
begin
  if edit1.Text='' then edit1.Text:=''
  else
    begin
      if pos('.',edit1.Text)<>0
      then edit1.Text:=edit1.Text
      else edit1.Text:=edit1.Text+'.';
    end;
end;
```

该程序首先判断文本框中是否有数字，如没有，则仍然保持为空。若有数字，则用 `pos` 函数判断该字符串中是否包含小数点，没有小数点的话则在原数字后添加小数点，已有小数点的话则什么也不做。

4. 双击 Button12 按钮，编写如下代码：

```
procedure TForm1.Button12Click(Sender: TObject);
begin
  edit1.Clear;
end;
```

用作清除文本框的内容。

5. 双击 Button13 按钮，编写如下代码：

```
procedure TForm1.Button13Click(Sender: TObject);
begin
```

```
if edit1.Text='' then edit1.Text:=''  
else begin  
flag:='+';  
x:=strtofloat(edit1.Text);  
edit1.Clear;  
end;  
end;  
end;
```

该程序判断文本框是否已有非空的字符串，如果文本框为空，则什么也不做，如果文本框中已经有非空字符串，则给字符变量 flag 赋值为“+”，同时将文本框中的字符串转化为数字，赋给变量 x，然后清空文本框内容。同样，编写 Button14, Button15, Button16 三个按钮的 click 事件，分别给字符变量 flag 赋值为“-”、“*”、“/”。

6. 双击 Button17 按钮，编写如下代码：

```
procedure TForm1.Button17Click(Sender: TObject);  
begin  
if edit1.Text='' then edit1.Text:=''  
else  
case flag of  
'+' : edit1.text:=floattostr(x+strtofloat(edit1.Text));  
'-' : edit1.text:=floattostr(x-strtofloat(edit1.Text));  
'*' : edit1.text:=floattostr(x*strtofloat(edit1.Text));  
'/' : begin  
y:=strtofloat(edit1.Text);  
if y=0 then edit1.Text:='999999999999999999999999999999'  
else edit1.text:=floattostr(x/strtofloat(edit1.Text));  
end;  
end;  
end;  
end;
```

单击该按钮后，程序判断文本框内是否有非空字符串，若文本框内字符串为空，则什么也不做，若文本框中有非空字符串，则查询字符变量 flag 的值，然后判断进行什么样的操作。在进行除法运算时，还要考虑到除 0 的问题，本例中，除 0 则结果定义为 999999999999999999999999999999，表示无穷大。

7. 双击 Button18 按钮，编写如下代码：

```
procedure TForm1.Button18Click(Sender: TObject);  
begin  
close;  
end;
```

单击该按钮后退出程序。

【实例效果】

1. 按 F9 键，运行后出现如图 1.3 所示界面。

2. 用鼠标单击 1234, 文本框中显示为 1234。如图 1.4 所示。

3. 单击“*”按钮后, 再分别单击 9876, 输入数字 9876, 如图 1.5 所示。按“=”按钮得到结果如图 1.6 所示。



图 1.3 运行界面 1



图 1.4 运行界面 2



图 1.5 运行界面 3



图 1.6 运行结果

【小结】

本例我们学习了计算器的制作。制作计算器一般要注意几个问题:

1. 没有任何数字输入时如何处理。
2. 几个数字的连续运算问题如何处理, 如 $1+2+4=?$, 这里要始终将前一个运算得出的结果作为下一个运算的因子。
3. 几个数字的连续运算时, 中间有数字被修改, 如 $1+2$ (在结果后添加 1) $+4=?$, 即计算 $31+4=?$, 这里要考虑的是如何使得下一个运算因子是 31, 而不是 3 或是 1。

读者试分析一下本例中哪些步骤是对应解决以上问题的。



说明：本例中用到了一个 `pos` 函数，它的格式为 `pos('字符串 a' , '字符串 b')`，用于判断字符串 `b` 是否包含字符串 `a`。还可返回 `a` 在 `b` 中出现的位置。这是一个很有用的函数，读者也可尝试自己用 Pascal 编一个具有查询功能的函数，以加深对 Pascal 语言的了解。

第二讲 一个 21 点的小游戏

【实例说明】

本例通过编写一个小游戏——21 点，全面介绍 Delphi 在解决具体问题时的常见技巧、分析设计思路。

21 点是一个常见的游戏，游戏人数可以是两人或更多的人。在游戏开始之前，首先准备一副或若干副牌，均匀洗牌。游戏者按照一定的顺序抓牌，每个游戏者都可以在任何时候停止抓牌，当全部游戏者停止抓牌后，开始判定胜负情况。所有超过 21 点的玩家都为负，在所有不超过 21 点的玩家中，点数较大者取胜。在本例中，将模拟一个有两个游戏者的 21 点游戏，其中电脑模拟成一个游戏者，和玩家进行对抗，在游戏结束时，判定双方的胜负情况。

【编程思路】

1. 首先分析程序可能的活动有开始游戏、电脑叫牌、人叫牌、电脑判断、输出结果。人的操作有电脑开始、人开始、人叫牌、人停止叫牌、清除游戏。把这些活动和操作画成一个图，如图 2.1 所示。

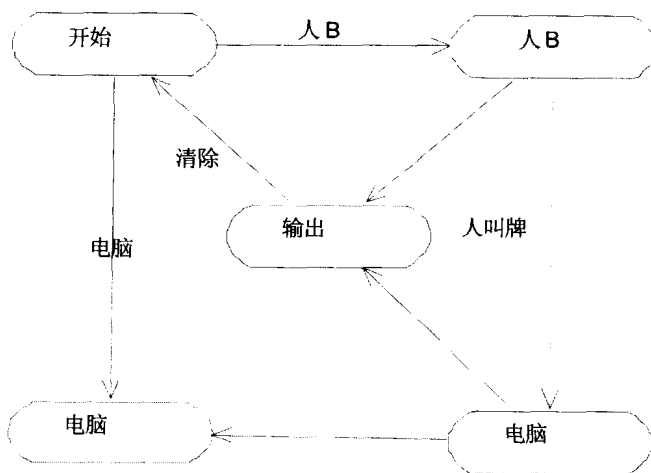


图 2.1 思路示意图

2. 根据这样的图，逐个编写每个活动和操作，从而大大地降低了程序的难度和复杂度。

在图 2.1 中可以看出，本程序包括开始游戏、人叫牌、电脑判断、电脑叫牌、输出结果五个状态，在这些状态之间有些是自动转换的，例如从电脑判断到电脑叫牌、从电脑判断到

输出结果，这样的设计方式可以使得程序的结构变得清晰。

【步骤】

创建一个新工程。具体的步骤如下：

1. 在 Form 上建立两个 Image 控件、一个 Label 控件和五个 Button 控件，更改名字为 FormMain, ImageStack, ImageComputer, LabelStatus, ButtonComputerBegin, ButtonHumanBegin, ButtonCall, ButtonHumanBegin, ButtonClear, ButtonStop。

属性设置为：

FormMain.Caption = '21 点游戏';

ImageStack.Picture 选取一张扑克牌背面的图案;

ImageComputer.Picture 选取一张机器人的图案。

控件排布如图 2.2 所示。

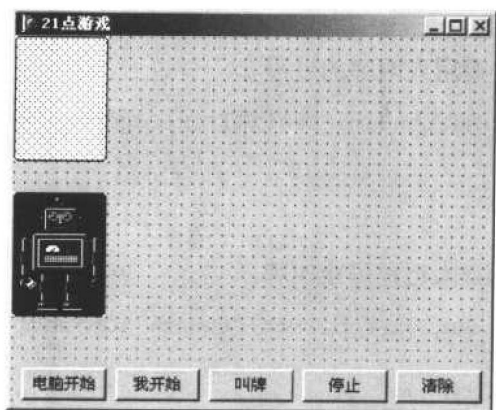


图 2.2 控件排布

2. 在程序的开头进行如下声明：

```
private
    iHumanSum : Integer;           //人的总点数
    iHumanCardCount : Integer;     //人的牌数
    bHumanStop : Boolean;         //人停止叫牌
    iComputerSum : Integer;       //计算机的总点数
    iComputerCardCount : Integer; //计算机的牌数
    bComputerStop : Boolean;      //计算机停止叫牌
    procedure HumanCall();        //人叫牌
    procedure GameBegin();       //游戏开始
    procedure ComputerCall();    //计算机叫牌
    procedure Result();          //输出结果
    procedure ComputerJudge();   //计算机判断
```


3. 当窗体加载时应该自动开始游戏，所以直接调用游戏开始过程。

```
procedure TFormMain.FormCreate(Sender: TObject);
begin
    GameBegin();
end;
```

4. 将操作对应到具体的活动上，当计算机开始的时候，调用计算机叫牌的过程。

```
procedure TFormMain.ButtonComputerBeginClick(Sender: TObject);
begin
    LabelStatus.Caption := '';
    ComputerCall();
end;
procedure TFormMain.ButtonCallClick(Sender: TObject);
begin
    LabelStatus.Caption := '';
    HumanCall();
end;
```

当人开始游戏的时候，调用人叫牌的过程。

```
procedure TFormMain.ButtonHumanBeginClick(Sender: TObject);
begin
    HumanCall();
end;
```

当停止叫牌的时候，标记人已经停止叫牌。

```
procedure TFormMain.ButtonStopClick(Sender: TObject);
begin
    LabelStatus.Caption := '';
    bHumanStop := True;
    if bComputerStop Then
    begin
        Result();
    end
    else
    begin
        ComputerJudge();
    end;
end;
```

5. 考虑游戏的开始，进行如下编码：

```
procedure TFormMain.GameBegin();
begin
    iHumanSum := 0;
    iHumanCardCount := 0;
    bHumanStop := False;
    iComputerSum := 0;
```