

21 世纪高等院校
计算机系列课程教材

Visual Basic 程序设计 习题与上机指导

主 编 余 华
副主编 赵 平 尹红征

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

21 世纪高等院校计算机系列课程教材

Visual Basic 程序设计 习题与上机指导

主 编	余 华	
副主编	赵 平	尹红征
编 者	何玉香	魏秀然
	阎素红	王 栋

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

在我国, Visual Basic 正在受到越来越多的计算机专业和非专业人士的重视, 许多高校相继开设了“Visual Basic 程序设计”课程。为了提高本课程的教学质量, 我们编写了《Visual Basic 程序设计》一书。

本书是《Visual Basic 程序设计》的配套用书, 共分为三部分, 第一部分是上机实验指导, 循序渐进地安排了 12 个实验, 旨在使读者初步掌握程序设计思想, 并熟练使用 Visual Basic 集成环境; 第二部分是练习题及解答, 可以帮助读者复习和巩固所学知识; 第三部分是计算机等级考试题解, 对读者参加计算机等级考试有所启迪和帮助。

本书内容丰富, 实用性强, 对读者提高开发 Visual Basic 应用程序的能力十分有益。本书不仅可以作为其他各类 Visual Basic 教材的辅助教材, 而且可以作为自学者的参考资料, 也可供各类报考 Visual Basic 考试的读者参考。

版权专有 侵权必究

图书在版编目(CIP)数据

Visual Basic 程序设计习题与上机指导/余华主编. —北京: 北京理工大学出版社, 2005.2

(21 世纪高等院校计算机系列课程教材)

ISBN 7-5640-0390-1

I. V... II. 余... III. BASIC 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 010486 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(发行部)

网 址 / <http://www.bitpress.com.cn>

电子邮箱 / chiefedit@bitpress.com.cn

经 销 / 全国各地新华书店

印 刷 / 北京地质印刷厂

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 10.75

字 数 / 256 千字

版 次 / 2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

印 数 / 1~6000 册

定 价 / 15.00 元

责任校对 / 陈玉梅

责任印制 / 刘京凤

图书出现印装质量问题, 本社负责调换

前 言

Visual Basic 以功能强大、简单易学的特点,受到越来越多的计算机专业和非专业人士的重视。我国许多高校相继开设了“Visual Basic 程序设计”课程。为了提高本课程的教学质量,我们编写了本书,与《Visual Basic 程序设计》一书相配套。

本书共分为三部分:第一部分是上机实验指导,循序渐进地安排了 12 个实验,详细指明了学习中应该达到的具体目标,并结合具体实例详细阐述程序调试手段,旨在为从课堂教学过渡到独立编程提供更佳的模拟环境和知识拓展,使读者初步掌握程序设计思想,并熟练使用 Visual Basic 集成环境;第二部分是练习题及解答,可以帮助读者复习和巩固所学知识;第三部分是计算机等级考试题解,相信会对读者参加计算机等级考试有所启迪和帮助。

程序设计是创造性的劳动,它需要编程人员全身心地投入,并充分发挥主观能动性。本书中的程序都是编者精心选取和编写的,并给出了运行结果,以印证编程的思路,体现程序的意图,而且由结果反观程序也能更好地理解程序中各语句的功能,于细微之处体会编程的技巧。如能对这些程序细加揣摩,那么读者的编程能力定会有所提高。

本书不仅与《Visual Basic 程序设计》一书相配套,互为补充,相辅相成,而且内容丰富,实用性强,对读者提高开发 Visual Basic 应用程序的能力十分有益。本书亦可作为其他各类 Visual Basic 教材的辅助教材、自学者的参考资料和参加 Visual Basic 考试的辅导用书。

我们衷心希望本书的出版,能给 Visual Basic 爱好者带来一些便利。感谢读者选择和使用本书。但限于时间、水平和能力,书中难免存在不妥之处,恳请广大读者和同仁批评指正。

编者

2004 年 12 月

目 录

第一部分 上机实验指导	(1)
实验一 VB 程序开发环境	(1)
实验二 数据类型与表达式	(5)
实验三 Visual Basic 程序设计	(8)
实验四 数组	(19)
实验五 过程	(23)
实验六 文件操作	(28)
实验七 窗体设计	(36)
实验八 菜单和工具栏设计	(43)
实验九 对话框设计	(50)
实验十 图形操作	(58)
实验十一 数据库功能	(64)
实验十二 综合实验	(69)
第二部分 练习题与答案	(95)
第 1 章 概述	(95)
第 2 章 数据类型与表达式	(95)
第 3 章 VB 语言基础	(97)
第 4 章 数组	(101)
第 5 章 过程	(101)
第 6 章 文件操作	(102)
第 7 章 窗体设计	(102)
第 8 章 标准控件的使用	(104)
第 9 章 菜单与工具栏设计	(108)
第 10 章 对话框设计	(108)
第 11 章 图形操作	(108)
第 12 章 数据库应用	(108)
练习题答案	(109)
第三部分 计算机等级考试试题及分析解答	(136)
上机模拟试题一	(136)
上机模拟试题二	(143)
上机模拟试题三	(148)
全国计算机等级考试二级笔试试卷及解答	(154)

第一部分 上机实验指导

实验一 VB 程序开发环境

一、知识要点

1. VB 程序开发环境

(1) VB 的特点、版本、安装和启动

- 了解 VB 面向对象、可视化、事件驱动的特点。
- Visual Basic 有三种版本：专业版、企业版和学习版。
- VB 帮助系统盘 MSDN 的安装和使用。

(2) VB 集成开发环境

主窗口：VB 的界面上有 5 个窗口：①菜单栏；②工具箱(Tools Box)，里面是 VB 所需的控件；③VB 程序要设计的窗体(Form)；④控件及窗体的属性(Properties)；⑤VB 的文件控制窗口。

其他窗口：

① 窗体设计器和工程资源管理器：窗体为控件的容器，多文档接口(MDI)窗体可作为子窗体和一些控件的容器。工程资源管理器显示当前打开工程及其内容的层次结构列表。

② 属性窗口和工具箱窗口：属性窗口包含被选择的窗体、控件在设计时拥有的属性的列表，这些属性能够在设计时被改变，当用户选择多个控件时，属性窗口包含的属性列表被所选择的控件共享。工具箱窗口包含应用程序在当前可用的控件和可插入的对象。

2. 对象及其操作性

(1) 对象

① VB 对象的概念，对象的属性、方法和事件。

② 对象属性设置：每个对象都有它的属性，对于属性，可以在窗体的布局操作中完成，也可以在程序运行中改变，但有些属性是只读的，它只能在窗体的布局操作中改变。

(2) 窗体

窗体是应用程序的编程窗口和对话框。窗体可包含多个控件。

① 窗体的结构与属性：窗体的常用属性有：Name 属性、Caption 属性、ControlBox 属性、BorderStyle 属性、AutoRedraw 属性、BackColor 属性、ForeColor 属性、Icon 属性、MaxButton 属性、MinButton 属性、WindowState 属性、Enabled 属性、FontBold 属性、FontItalic 属性、FontStrikethru 属性、FontUnderline 属性、FontTransparent 属性、FontSize 属性、Picture 属性、

Height 和 Width 属性、Left 和 Top 属性、Visible 属性等。

② 窗体事件: Paint 事件、Resize 事件、Load 事件和 Unload 事件等。

(3) 控件

常用的控件包括: 标签、文本框、命令按钮、框架、单选按钮、复选框、列表框、组合框、水平和垂直滚动条等。每一个控件建立的时候, 系统都会给它设一个默认的名字(Name)。

(4) 事件驱动

“面向对象的编程”是由事件驱动的, 对于不同的对象, 事件也会不同。

① 窗体和图像框类事件: Paint 事件、Resize 事件、Load 事件、Unload 事件。

② 当前光标(Focus)事件: GotFocus 事件、LostFocus 事件。

③ 鼠标操作事件: Click 事件、DbClick 事件、MouseDown 事件、MouseUp 事件、MouseMove 事件、DragDrop 事件和 DragOver 事件。

④ 键盘操作属性: KeyDown 事件、KeyUp 事件、KeyPress 事件。

⑤ 改变控制项事件: Change 事件、DropDown 事件和 PathChange 事件。

⑥ 其他事件: Timer 事件。

3. VB 可视化编程的步骤及应用程序的构成和管理

VB 可视化编程的步骤包括新建一个工程、添加控件、设置属性、编写代码、运行工程、修改工程、保存工程、工程的编译和制作安装盘等。

VB 应用程序的组成: 一个.vbp 工程文件、一个(或多个).frm 窗体文件、自动产生的.frx 二进制文件, 还可以有.bas 及.cls 文件。

要能够区分窗体名称和窗体文件名。

二、实验目的

① 了解 Visual Basic 6.0 的特点。

② 熟悉 Visual Basic 6.0 的开发环境。

三、实验任务

了解一个 Visual Basic 6.0 程序的建立、编辑、调试、运行和保存方法。

实验内容 1:

启动 Visual Basic 6.0, 创建一个“标准 EXE”类型的应用程序。要求: 在窗体上显示“HELLO! WORLD”。以 char1.frm 和 char1.vbp 分别为窗体名和工程名保存在自己的软盘上。

【分析】

该题不需要编程实现, 只需在窗体上增加一个标签控件, 然后修改该标签的 Caption 属性的值即可。

【实验步骤】

(1) 启动 Visual Basic 6.0 开发环境。

(2) 创建一个工程。

① 启动 Visual Basic 6.0 开发环境后出现“新建工程”对话框(如图 1-1 所示)。



图1-1 “新建工程”对话框

② 单击“标准 EXE”图标，单击“打开”按钮，就创建了一个 Visual Basic 6.0 的工程(如图 1-2 所示)。

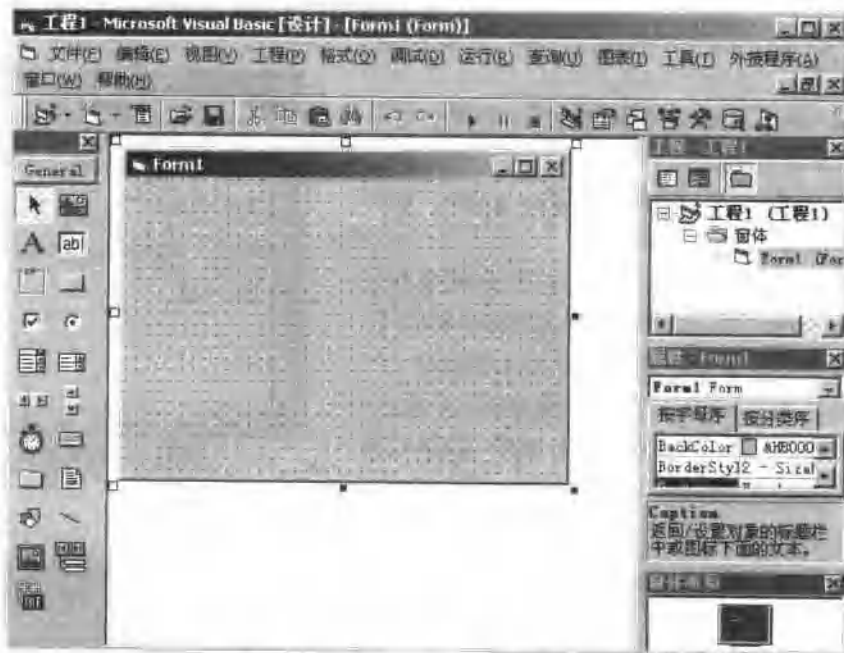


图1-2 创建 Visual Basic 6.0 的工程

(3) 在窗体上添加一个标签 label1，选中 label1，修改其属性中的 Caption 值为：“HELLO! WORLD” (如图 1-3 所示)。

(4) 在指定的位置保存窗体文件和工程文件。

(5) 单击运行菜单，选择“启动”；或者直接单击标准工具栏上的启动按钮，就可以看到运行结果(如图 1-4 所示)。



图1-3 标签的Caption属性



图1-4 运行结果

实验内容 2:

Print 方法练习, 在窗体上输出“我开始学习 Visual Basic 6.0 了。”以 char2.frm 和 char2.vbp 分别为窗体名和工程名保存在自己的软盘上。

【分析】

由于该题是 Print 方法的练习, 所以需要很简单的编程, 通过本例, 对面向对象编程中对象、事件和方法的实际操作意义有更深一步的认识。

【实验步骤】

(1) 启动 Visual Basic 6.0 开发环境, 创建一个工程(方法同实验内容 1)。

(2) 选择代码窗口。在通用处选择 Form, 在声明处选择 Click(如图 1-5 所示)。

(3) 在 Private Sub Form_Click() 和 End Sub 之间输入代码: print “我开始学习 Visual Basic 6.0 了”

(4) 在指定的位置保存窗体文件和工程文件。

(5) 单击“启动”按钮, 运行界面如图 1-6 所示。

(6) 单击窗体, 激活 Form_Click 事件, 就可以看到如图 1-7 所示的运行结果。

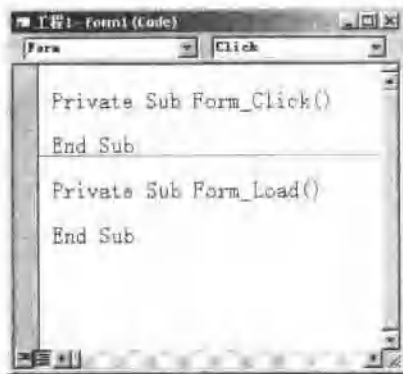


图1-5 代码窗口



图1-6 运行界面

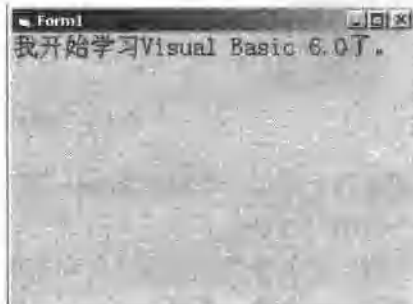


图1-7 运行结果

实验二 数据类型与表达式

一、知识要点

1. 数据类型及其运算

(1) 数据类型

①基本数据类型: 整型(Integer)、长整型(Long)、单精度实型(Single)、双精度实型(Double)、字符型(String)、布尔型(Boolean)。

② 用户自定义类型: 可以用 `Type...End Type` 语句创建用户定义的类型。

③ 枚举类型: 可以用 `Enum...End Enum` 语句来定义一个相关的命名常数集合。

(2) 变量和常量

① 局部变量与全局变量: 用 `Public` 声明的变量就是全局变量。在模块和窗体中用 `Dim` 或 `Static` 等声明的变量叫局部变量。

② 变体类型变量: `Variant` 数值类型是对所有的未定义的变量的默认数据类型。

③ 默认声明: 在变量名后面加上如下符号即确定了其类型, 如单精度浮点数`!`(Single)、双精度浮点数`#`(Double)、整型变量`%`(Integer)、长整形数据`&` (Long)、货币型数据`@` (Currency)、字符串型数据`$`(String)。

④ 在 VB 中, 常量有三类, 分别为:

直接常量——字符串常量、数值常量、布尔常量、日期常量。例如: “A”、“123”、1、1.13、0.123E1、True、False、#06/20/2004#。

用户自定义常量——声明常量的语法为:

`[Public/Private]Const<常量名>[As <数据类型>]=<表达式>`

`<表达式>`由数值常量、字符串等常量及运算符组成, 可以包含前面定义过的常量, 但不能使用函数调用。

VB 系统提供的常量——例如: `VbOkonly`、`VbYesno`。

(3) 常用内部函数

数学函数: `Abs`、`Sqr`、`Sgn`、`Exp`、`Log`、`Sin`、`Cos`、`Fix`、`Int` 和 `Rnd` 等。

字符函数: `Mid`、`Len`、`Left`、`Chr`、`Asc`、`Lcase`、`Ucase`、`LTrim()`和 `RTrim()`等。

日期和时间函数: `Date`、`Now`、`Day`、`Month`、`Weekday` 和 `Year` 等。

转换函数: `Val` 和 `Str` 等。

(4) 运算符与表达式

① 算术运算符: `+`、`-`、`*`、`/`(除法)、`\`(整除)、`Mod`(求余)和`^`(幂运算)。

② 关系运算符: `=`、`>`、`<`、`>=`、`<=`和`<>`。

③ 逻辑运算符有: `Not`、`And`、`Or`、`Xor`、`Eqv`(相等)和`Imp`(蕴含)。

④ 表达式的书写规则: 从左到右在一行上书写, 无高低区分; 表达式中的运算符可以用圆括号括起来, 以改变运算的优先级或增加表达式的清晰度。不能使用方括号或花括号, 圆括号可以出现多个, 但要配对。

2. 程序的书写规则

(1) 语句

VB 程序中的一行代码称为一条程序语句，简称为语句。每个语句行以回车键结束。一个语句行的长度最多不能超过 1023 个字符。

(2) 书写规范

几个语句放在一个语句行中，语句之间用冒号“:”隔开。

用续行符“_”将一个较长的语句分为多个程序行。在使用续行符时，在它前面至少要加一个空格，并且续行符只能出现在行尾。

(3) 程序注释

注释是软件可读性的具体体现。程序注释不能用抽象的语言，要精确表达出程序的处理说明。避免每行程序都使用注释，可以在一段程序的前面加一段注释，具有明确的处理逻辑。注释必不可少，但也不应过多，不要被动地为写注释而写注释。

以下是几种必要的注释：

标题、附加说明；函数、类等的说明；在代码不明晰或不可移植处必须有一定的说明；少量的其他注释，如自定义变量的注释、代码书写时间等。

二、实验目的

- ① 了解 VB 各种数据类型，掌握常量与变量的各种数据类型。
- ② 掌握变量名的命名规则。
- ③ 掌握表达式的类型及其正确书写规则。
- ④ 掌握常用内部函数(常用字符串函数、日期和时间函数、格式输出函数、随机数语句和函数、数据类型转换函数)的使用。
- ⑤ 掌握语句的书写规则。

三、实验任务

实验内容 1:

数值变量和表达式的简单应用。

【实验步骤】

- (1) 分析下面的代码，写出运行结果。
- (2) 建立一个工程，输入下面的代码并运行。校验自己给出的运行结果是否正确。

```
Private Sub Form_Click()
```

```
    Dim inta As Integer, intb As Integer, sinc As Single, doud As Double  
    inta = 20  
    intb = 20 \ 7  
    Print intb  
    intb = 20 / 7  
    Print intb  
    sinc = 20 \ 7  
    Print sinc
```

```
sinc = 20 / 7
Print sinc
doud = 20 \ 7
Print doud
doud = 20 / 7
Print doud
Print inta Mod 7
Print inta - 31
Print -inta
Print inta ^ 2
```

End Sub

实验内容 2:

字符变量和表达式的简单应用。

【实验步骤】

- (1) 分析下面的代码，写出运行结果。
- (2) 建立一个工程，输入下面的代码并运行。校验自己给出的运行结果是否正确。

```
Private Sub Form_Click()
    Dim str1 As String, str2 As String
    str1 = "中华人民共和国"
    str2 = "China"
    Print Left(str1, 1) + Right(str1, 1) + str2
    Print Len(str1)
    Print Len(str2)
    Print Left(str1, 1) + Right(str1, 1) + Mid(str1, 3, 2)
End Sub
```

实验内容 3:

关系运算符和逻辑运算符的使用。

【实验步骤】

- (1) 分析下面的代码，写出运行结果。
- (2) 建立一个工程，输入下面的代码并运行。校验自己给出的运行结果是否正确。

```
Private Sub Form_Click()
    Dim s1 As Boolean, s2 As Boolean, s3 As Boolean
    Print "abcde" = "abcd"
    Print "abcde" > "aba"
    Print "abcdfgr" > "az"
    Print 31 < 3
    s1 = False
    s2 = True
```

```
s3 = True
Print Not s1
Print s1 And s2
Print s1 Or s2
End Sub
```

实验内容 4:

算术函数的使用。

【实验步骤】

(1) 分析下面的代码，写出运行结果。

(2) 建立一个工程，输入下面的代码并运行。校验自己给出的运行结果是否正确。

```
Private Sub Form_Click()
    Print Abs(-90.8)
    Print Log(10)
    Print Exp(1)
    Print Log(Exp(3))
    Print Int(-9.9) - Fix(-9.9)
    Print Rnd
    Print Rnd(0)
    Print Rnd(1)
    Print Rnd(-1)
    Print Rnd(0)
End Sub
```

实验三 Visual Basic 程序设计

一、知识要点

1. 基本语句

(1) 赋值语句 Let

[Let]<名称>=<表达式>

“Let”是可选项，完成复制赋值功能只需“=”(赋值号)；

“名称”是变量或属性的名称；

“表达式”可以是算术表达式、字符串表达式、关系型表达式或逻辑表达式。赋值号两边的数据类型必须一致。

(2) 卸载对象与结束程序

Unload 语句的语法格式为：Unload 对象名

结束语句：End

(3) 注释与暂停

注释语句: Rem<注释内容> 或'<注释内容>

2. 选择结构语句

(1) 单行结构条件语句

If <条件> Then <语句>

或 If <条件> Then
 <语句块>

End If

其中, 条件可以是任何表达式, 即算术、字符、关系和逻辑表达式, 按表达式的值非 0 为 True, 0 为 False 来判断。

多行的 If 必须与 End If 配对; 单行格式没有 End If 关键字。

(2) 块结构条件语句

① 一般格式如下:

```
If <条件 1> Then
    <语句块 1>
[ElseIf<条件 2>Then
    <语句块 2>]
[ElseIf<条件 3>Then
    <语句块 3>]
...
[Else
    <语句块 n>]
End If
```

这里的“语句块”可以是单个语句, 也可以是多个语句。当有多个语句时, 可以分别写在多行里; 如果写在同一行中, 则各语句之间用冒号隔开。

② 当省略 ElseIf 子句和 Else 子句时, 块结构条件语句简化为:

```
If <条件> Then <语句 1>
    <语句块>
End If
```

(3) If 语句的嵌套和使用

```
If<条件 1>Then
    ...
    If<条件 11>Then
        ...
    End If
    ...
End If
```

区分嵌套的层次方法: 每个 End If 与它上面最近的 If 配对。书写为锯齿形, 便于区分和配对。

(4) 情况语句 Select Case 的格式和使用

格式:

```
Select Case<测试表达式>
    [Case<表达式列表 1>
        [<语句块 1>]]
    [Case<表达式列表 2>
        [<语句块 2>]]
    ...
    [Case Else
        <语句块 n>]

End Select
```

“测试表达式”可以是数值表达式或字符串表达式,通常为变量;每个语句块由一行或多行 VB 语句组成;“表达式列表”中的表达式必须与“测试表达式”的类型相同;当使用多个表达式列表时,表达式与表达式之间要用“,”隔开。

(5) 使用 If 函数

If 函数的语法结构为:

```
If(<条件表达式>,<真部分>,<假部分>)
```

“条件表达式”可以是关系表达式、布尔表达式、数值表达式。如果用数值表达式,则非 0 为真,0 为假。“真部分”是当条件表达式为真时函数返回的值,可以是任何表达式。“假部分”是当条件表达式为假时函数返回的值,可以是任何表达式。

3. 循环语句

(1) For 循环控制结构

```
For<控制变量>=<初值> To <终值> [Step <步长>]
    <循环体>
Next<控制变量>
```

常用于已知循环次数的场合。步长可正可负,可为整数,也可为实数。步长不能为 0,否则作“死循环”。掌握循环次数的计算:

$$n = \frac{\text{终值} - \text{初值}}{\text{步长}} + 1$$

当控制变量(步长为正)大于终值,循环终止。当步长为负值时,判断控制变量是否小于终值。

(2) 当循环控制结构

While...Wend 语句格式如下:

```
While<条件>
    <语句块>
Wend
```

(3) Do...Loop 循环控制结构

① 先判断条件形式的 Do...Loop 语句。

语句格式如下:

Do[While/Until<条件>]

[<语句块>]

Loop

当指定的循环条件为 True 或直到指定的循环结束条件变为 True 之前重复执行语句块组成的循环体。其中保留字 While 用于指定循环条件, 保留字 Until 用于指定循环结束条件。

② 后判断条件形式的 Do...Loop 语句。

语句格式如下:

Do

[<语句块>]

Loop[While/Until<条件>]

后判断条件形式的 Do...Loop 语句与先判断条件形式的区别是首先执行循环体, 然后测试循环条件或循环终止条件, 决定是否继续循环。所以, 这种结构的语句至少执行一次循环体。

(4) 循环的嵌套及注意事项

在一个循环结构的循环体内含有另一个循环结构称为嵌套或多重循环。各种形式的循环语句都能够互相嵌套, 嵌套的原则是, 外层循环与内层循环必须是层层相套, 循环体之间不能交叉。

(5) 循环的退出

4. 数据的输入和输出

(1) 数据输入

① 使用“文本框”控件的输入: 文本框(TextBox)是一种通用控件, 可以由用户输入或显示文本。缺省时, 文本框只能输入单行文本, 并且输入的字符最多为 2048 个。若控件的 MultiLine 属性设置为 True, 则可以输入多行文本, 并且文本的内容可多达 32KB。

② 使用焦点: 焦点(Focus)就是光标, 当对象具有“焦点”时才能响应用户的输入, 具有焦点的对象通常会以突出显示标题或标题栏来表示。当文本框具有焦点时, 用户输入的数据才会出现在文本框中。仅当控件的 Visible 和 Enabled 属性被设置为(True)时, 控件才能接收焦点。当控件接收焦点时, 会引发 GotFocus 事件, 当控件丢失焦点时, 将会引发 LostFocus 事件。可以用 SetFocus 方法在代码中设置焦点。

③ InputBox 函数显示一个能接受用户输入的对话框, 并返回用户在对话框中输入的信息。其语法格式为:

变量=InputBox(<信息内容>[, <对话框标题>][, <默认内容>])

④ MsgBox 函数在对话框中显示信息, 等待用户单击按钮, 并返回一个整数以标明用户单击了哪个按钮。其语法格式为:

变量=MsgBox(<信息内容>[, <对话框类型>[, <对话框标题>]])

(2) 数据输出

可以用 Print 方法, 格式输出(FORMAT\$)函数。

使用 Print 方法可以在窗体上输出文本字符串或表达式的值, 其语法格式为:

[<对象名称>].Print[<表达式列表>][[, 1;]]

Print 方法具有计算和输出的双重功能,对于表达式,总是先计算后输出。

与 Print 方法有关的函数有 Tab 函数和 Spc 函数。

清除方法 Cls 的语法为:

[<对象名称>].Cls

二、实验目的

- ① 了解顺序结构的特点。
- ② 掌握赋值语句的正确书写规则。
- ③ 掌握数据输入与输出的各种方法与函数(InputBox、Msgbox)的使用。
- ④ 掌握逻辑表达式的正确书写规则。
- ⑤ 掌握单行结构条件语句与块结构条件语句的使用。
- ⑥ 掌握 If 语句的嵌套和使用。
- ⑦ 掌握情况语句的使用及多分支条件语句的使用。
- ⑧ 掌握 For 循环控制结构的使用。
- ⑨ 掌握 Do 语句的各种形式的使用。
- ⑩ 掌握循环的嵌套、循环的退出。
- ⑪ 掌握如何控制循环条件,防止死循环或不循环。

三、实验任务

实验内容 1:

将 a、b 两个变量的值互相交换,并在窗体上输出变量互换前后的值,如图 3-1 所示。以 char3_1.frm 和 char3_1.vbp 分别为窗体名和工程名保存在自己的软盘上。

【程序代码】

```
Private Sub Form_Click()  
    Dim a As Single, b As Single, n As Single  
    a = InputBox("请输入变量 a 的值")  
    b = InputBox("请输入变量 b 的值")  
    Print "变量交换前的值分别为: a="; a; "b="; b  
    n = a  
    a = b  
    b = n  
    Print "变量交换后的值分别为: a="; a; "b="; b  
End Sub
```

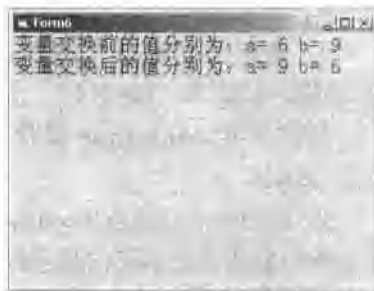


图3-1 值的互换

实验内容 2:

窗体设计如图 3-2 所示,圆半径的值由文本框输入,分别单击圆周长、圆面积按钮,则在其对应的文本框输出相应的值。运行结果如图 3-3 所示。以 char3_2.frm 和 char3_2.vbp 分别为窗体名和工程名保存在自己的软盘上。