中国计算机软件专业技术水平考试教学辅导用书

程序设计
重点综述与试题分析

（初级程序员）

补充资料

- 综述实用
- 要点突出
- 切题准确
- 分析详尽

中国民航出版社

中国计算机软件专业技术水平考试教学辅导用书

程序设计
重点综述与试题分析
（初级程序员）

第二版

补　充　资　料

策　划：何学仪
编　者：钟　珞　夏红霞　吕品

中国民航出版社

图书在版编目(CIP)数据

中国计算机软件专业技术水平考试　程序设计——重点综述与试题分析(初级程序员)补充资料/钟珞等编.
——北京:中国民航出版社,2001.6
ISBN 7-80110-243-6
Ⅰ.程…　Ⅱ.钟…　Ⅲ.电子计算机-软件资格水平考试-学习参考资料
Ⅳ.TP3
中国版本图书馆 CIP 数据核字(2000)第 08279 号

程序设计
重点综述与试题分析
(初级程序员)第二版
补充资料
钟珞等编
*
中国民航出版社出版发行
(北京市朝阳区光熙门北里甲 31 号楼 5 层)
华东政法学院印刷厂
开本:787×1092　1/16　印张:4.5　字数:108 千字
2001 年 6 月第 1 版　2001 年 6 月第 1 次印刷
ISBN 7-80110-243-6/G·086　定价:6.00 元

(发行电话:(021)63052990　本书如有印装错误,印刷厂负责调换)

前 言

根据《2001 年专业技术人员资格考试工作计划》(人办发[2000]75 号)一文要求,全国计算机软件专业技术资格和水平考试定于 2001 年 10 月 14 日举行。2001 年度的计算机软件专业技术水平考试增加网络程序员、网络设计师两个级别的考试。

依据中国计算机软件专业技术资格和水平考试大纲的最新要求,我们新增了计算机专业英语和 VB 的有关知识等考试要点及试题分析,并作为《程序设计重点综述与试题分析》一书的补充资料,给应试人员以提纲指导并快速掌握新大纲的新增知识点的相关内容。

本书由何学仪策划,武汉理工大学钟珞、夏红霞、吕品等同志参加了本书的编写工作。

因水平有限,书中难免存在错漏和不妥之处,望请读者指正,以利于改进提高。

目 录

新增知识点

一、一种可视化的编程工具
(Visual Basic 6.0 中文版)

1.重点综述

□创建应用程序界面(窗口、常用控件、菜单等)

Visual Basic 是集成开发环境 IDE,下面分别介绍如下:

1.1 标题栏

标题栏是屏幕顶部的水平条。在标题栏里,它显示了你所操作的项目的名称,由于新建立项目时,Visual Basic 默认项目名为 Project1,所以在标题栏里,显示的是 Project1。而 design(设计)则表明现在整个项目正处于编辑状态,而不是下文将提到的运行(Run)状态或断点(Break)状态。(随着工作状态的不同,方括号中的信息也随之改变,可能是"run"模式或"break"模式)。

1.2 菜单条

标题栏的下面是集成环境的主菜单,在菜单条里,包含了运行 Visual Basic 进行编程开发项目所需要的命令。这些菜单命令提供了开发、调试、保存应用程序所需要的工具。基本的菜单命令有:

·File(文件):包含打开(Open Project)、保存(Save Project)和删除(Remove)项目、保存(Save Form)和加入窗体(Add Form)以及生成执行文件(Make Project1.exe),打印窗体代码等与文件操作有关的一系列命令和最多到四个的最近用到的项目。

·Edit(编辑):包含了一系列与编辑有关的命令,如撤消上次操作(Undo),剪切(Cut)、粘贴(Paste)以及查找(Find)、替换(Replace)等常用编辑命令和缩进(Indent)、凸出(Outdent)等排版命令。

·View(视图):包含了显示和隐藏集成开发环境(IDE)元素的命令。如显示或隐藏代码窗口(Code)、显示或隐藏属性窗口(Properties Window)、显示和隐藏工具栏等命令。

·Project(工程):包含将窗体、模块加入当前项目、引用 Windows 对象和工具框新工具的命令。

·Format(格式):包含对齐窗体控件(Align)和锁定窗体控件使之不能移动(Lock Controls)等命令。

·Debug(调试):包含对程序进行调试的各个命令,如逐语句(Step Into)、逐过程(Step Over)运行程序,设置监视(Add Watch)等命令。

·Run(运行):包含了运行程序(Start)和停止运行(End),重新运行程序(Restart)等在 Visual Basic IDE 环境下运行应用程序的命令。

·Tools(工具):包含了启动菜单编辑器(Menu Editor)的命令和对 IDE 环境进行配置(Op-

tions)的选项,还包含了建立 ActiveX 构件和 ActiveX 控件时所要的工具。

·Add - Ins(外接程序):缺省包含了可视化数据管理器(VisData)外接程序(Visual Data Manager),编程人员可以通过外接程序管理器加入或删除其它的外接程序。

·Help(帮助):包含了 Visual Basic 的帮助信息。

1.3 工具条

工具条位于菜单条的下面,即主窗口的底部,它以图标的形式提供了部分常用菜单命令的功能,利用工具条,可以快速访问这些常用的菜单命令,而不必下拉菜单条来找菜单命令。除了缺省显示在菜单条下面的主工具条处,通过选择菜单 View 下的 toolbars 子菜单中的工具条子菜单的名字,来打开或关闭工具条。可供选择的其它专用工具条有编辑(Edit)工具条,窗体设计(Form Editor)工具条,Debug 调试工具条等。

·Standard 标准工具条:在菜单的正下方。

·Debug 调试工具条:是 Debug 菜单命令的快捷方式。

·Edit 编辑工具条:是 Edit 菜单命令的快捷方式,包含了 Edit 菜单的命令。

·Form Editor 窗体设计工具条:是 Format 菜单命令的快捷方式。

1.4 常用控件及其功能

Toolbox 工具框中包含了控件的图标,可以将这些控件拖到窗体中,生成应用程序的人机对话接口。

常用控件及其功能如表 1。

表 1　常用控件及其功能

控件名	功能
PictureBox(图形框控件)	该控件用于显示图形,用控件的 Picture 属性设置所要显示的图形。
Label(标记控件)	该控件显示窗体上用户不能编辑的文本。用该控件的 Caption 属性来设置要显示在窗体上的文本。在程序中通过更改 Caption 属性达到更改文本的要求。标记控件通常用来标识其它控件。
TextBox(文本框控件)	该控件用于显示用户可以编辑的文本。与 Label 控件不同的是,该控件是个微型文字编辑器,利用该控件的 Text 属性,可以设置控件上的文本或读取用户输入的文本。
Frame(桢控件)	该构件用于组合其它元素,如复选框。
Command Button(命令按钮控件)	该控件用于在人机交互中提供确定的命令要求。改变该控件的 Caption 属性将改变显示在命令按钮上的文本。
CheckBox(复选框控件)	复选框控件提供给用户若干个可以选择的选项,用户一次可以选择一个或若干个选项。程序通过判断控件的 Value 属性来判断用户是否选中该选项。选中该控件时,控件的 Value 值为 1,未选中时,控件的 Value 值为 0。
Option Button(选项钮控件)	选项钮控件一般是成组出现的,用户只能选择一个选项,这一点与 CheckBox 控件不同。选定控件时,控件的 Checked 属性值为 True,否则,控件的 Checked 属性值为 False。

控件名	功能
ListBox(列表框控件)	该控件包含了一系列选项,用户可以选择其中的一个或几个选项。列表框控件中所选项目由控件的 Text 属性给定。而列表框控件的 Sorted 属性确定了表中项目是否排序。
ComboBox(组合框控件)	组合框控件与 ListBox 控件类似,但不同的是组合框控件包含了一个文本编辑字段,用户既可以选择表中的项目,亦可以在编辑字段中输入新字符串。利用该控件的 Text 属性就可以得到用户所选择的项目或用户输入的字符串。
Horizontal ScrollbBox(水平滚动条控件)	用于定性输入一个数据,由在控件的最小值和最大值之间的位置确定用户的输入值的大小。
Vertical ScrollBar(垂直滚动条控件)	其作用与水平滚动条控件的作用是一致的,唯一的区域是滚动的形状不一样。
Timer(定时器控件)	用于完成确定时间间隔必须进行的任务。用 Interval 属性确定事件发生的时间间隔。
Drive ListBox(盘列表框控件)	显示系统上的驱动器,用户在其中可以选择某个驱动器,如 A 盘、C 盘等。
Directory ListBox(目录列表框控件)	显示当前盘上的一系列文件目录。
File ListBox(文件列表框控件)	显示当前目录下的所有文件清单。与上面两个控件一起可以组成一个文件系统,为用户出现提供文件处理能力。提供给用户访问和处理磁盘、目录、文件的接口。
Shape(形状控件)	用于在窗体上绘制方框和圆形等图形的功能。
Line(直线控件)	其属性与 Shape 控件基本一致,用于在窗体上绘制直线。
Image(图像控件)	类似于图形框控件,用于显示图形,但它只支持图形框控件的几个特性,需要更少的资源。
Data(数据控件)	该控件提供了存放在数据库中的数据的一系列访问命令,它有很多属性,是数据库编程中很重要的一个数据访问控件,在数据库操作一章中将重点介绍。
OLE Container(OLE 容器控件)	OLE 容器控件是一个窗口,可以将其放在窗口中,用于放置其它应用出现的中文档,通过该控件,用户可以访问支持 OLE 的其它应用程序,如 Microsoft Excel 文档。

□设置对象的属性

窗体和控件是 Visual Basic 应用程序用户接口的基本元素。在 Visual Basic 中,这些元素被称为对象,它们具有属性和方法,并响应外部事件。属性是对象的表观显现,如同我们周围的物体一样,如文本框控件有字体属性、背景颜色、前景颜色等属性,可以在设计时通过属性窗口或程序运行时通过代码程序进行修改,而我们周围的物体,如电脑,它也有它的属性,如机箱形式,CPU 型号等,在购买电脑时,我们可以根据我们的要求购买立式机箱或是卧式机箱,可以购买 Pentium166 或购买 Pentium266 的 CPU。同样,Visual Basic 根据用户要求,通

过属性窗口或代码修改窗体或控件的属性，使之符合用户的要求。

对于每个放置在窗体上的控件，包括窗体本身，Visual Basic 都提供了缺少属性，如上述第一个例子，文本框控件的名字是 Text1，命令按钮控件的名字是 Command1 和 Command2。即每个控件的名字的缺省值是控件名加上一个序号。

有些属性只能在窗体设计时给定，有些属性则必须在程序运行时由程序代码设定，而有些属性则既可在设计时在属性窗口中定义，也可在程序运行时由程序代码设定，如控件名 Name 属性只能在窗体设计时给定，文本框控件的 Text 属性则既可设计时给定，也可在程序运行中设置，而下拉列表框控件的各个下拉项则必须在程序运行时由程序给定。

有些属性可读可写，有些属性则只能读不能写，如文本框控件的 Text 属性既可读又可写，而列表框控件的下拉列表项目数、选中的项目在列表中的位置 Index 属性只能读不能写。它们的改变随下拉项目数的改变和用户所选项目所处的位置的改变而改变。

Visual Basic 中常用的属性如下：

Name（名字）属性设置控件的名字，用于区分该控件是该控件而不是别的控件的身份证，利用该属性，可以访问控件的其它属性和方法。

Appearance（外观）设置控件在窗体上的外观形式，该属性值为 0 表示是平面型，1 表示是三维外观。

BackColor（背景色）该属性用于设置显示文字或绘制图形的背景颜色。

ForColor（前景色）该属性设置文本等的显示颜色。

Font（字体）该属性用于设置控件标题（Caption）或文本（Text）属性的属性值时所用字体的字体、字号等。

Caption（标题）用于设置不能接受文本输入的控件上的文本，如命令按钮上显示的“OK”和“退出”就是设置了两个命令按钮的 Caption 属性。

Text（文本）用于设置接受输入的控件上显示的文本，如文本框控件。

Enabled（使能）设置控件是否能够获得焦点，该属性的缺省值是 True，默认控件可以获得焦点，如果设置为 False 时，则控件不能获得焦点，也就无法对其进行操作（指程序运行时）。

Visible（可见性）用于设置控件在程序运行时是否可见，其缺省值为 True，默认控件是可见的，如果设置为 False 时，则程序运行时，该控件是不可见的，用户同样不能对其进行操作。

Left、Top（控件左、上方坐标）控件在窗体上的位置，在设置窗体时，程序员放置控件时的位置为其缺省值，程序员调整控件位置时，这两个属性值随之改变，其缺省单位是 Twips（1/20point，1/440inch）。

Width、Height（控件长度、宽度）设置控件的尺寸。同 Left、Top 属性，在程序员放置控件时，控件的长度和宽度就已经获得了其缺省值，程序员在改变控件的长度、宽度时，该属性值随之改变。

□编写程序代码

1.1 常量，变量和数据类型

Visual Basic 语言包括以下几个组成部分

·常量

·变量

·过程

·函数

·变元

·语句

变量存放数据，而过程是操作变量的代码。Visual Basic 应用程序中的大多数代码都涉及控件属性的操作。

在程序执行过程中，数值不会改变的量，这就是常量。它代表内存中指定的存储单元。如程序要进行数学计算，可能多次出现数值 Pi(3.1415926...)，这些值最好用常量表示，每次需要输入 3.1415926...的地方，都可以用常量名 Pi 代替，这样，更容易理解，输入不易出错。例如：

Area = Pi * Radius^2

语句就比下列语句

Area = 3.1415926 * Radius^2

更容易理解，输入更简洁，不易出差错。

声明常量有两个好处：

①安全特性。常量不能改变数值，一旦声明为常量，就不能在此后的语句中改变它的数值，也不能重新赋值。这就可以确保常量声明中指定的数值在程序的其余部分有效。

②常量处理速度快。程序运行时，常量值不需要查找，编译器只要把常量名换成常量值即可。因此程序的运行速度更快。

常量区分为不同的类型，如 10、3 为整型常量，3.14 为实型常量，“Adam34”为字符常量。声明常量的方式是：

Const constantname[As type] = value

声明的类型(As type)部分是可选的。省略时，常量的类型由赋予的值确定。常量的使有范围，可以是 Public 或 Private，例如，在模块中声明常量 Pi 为 Public，则每个过程都可以引用它：

Public Const Pi As Double = 3.1415926

constantname 是有效的常量名。

value 是常量值，是直接数，可是由数字和字符串常量运算符组合成的简单表达式。常量声明中不能使用函数。

Visual Basic 大量使用常量定义各个方法变元和设置各个控件属性。例如，复选框控件值选项有三个：0(取消)、1(选定)或 2(变灰)。Visual Basic 中定义了自己的内部常量，相对于上述三个值，分别用 Visual BasicUnchecked，Visual BasicChecked，Visual BasicGrayed 表示取消、选定或变灰。如：

Check1.Value = 0

Check1.value = 1

可以用下列语句来代替：

Check1.Value = Visual BasicUnchecked

Check1.Value = Visual BasicChecked

Visual Basic 常量均用前缀 Visual Basic 表示，在声明自己的常量时，请最好不要用这个前缀。

常量声明语句中可以包括其他常量。如：

Public Const Pi As Double = 3.1415926

Public Const Pi Double As Double = 2 * Pi

上列中的第二句定义的常量 PiDouble 就引用了第一句里的常量 Pi 的定义。

对常量的定义最好是在一个模块里定义，这样较容易维护和修改。如果常量在不同的模块里定义，一定要避免循环定义。如下例所示，

Const a1 = a2 * 4

Const a2 = a1/4

这个循环定义没有意义，应当避免。

在 Visual Basic 中，变量是用于程序执行期间保存数值时，在程序的执行中，其值是可以改变的。一个变量应该有一个名字，在内存中占用一定的存储单元，在该存储单元中存放变量的值，请注意区分变量名和变量值这两个不同的概念。Visual Basic 通过变量名来访问内存中的数据。

和其他高级语言一样，用来标识变量名、符号常量名、数组名、类型名、文件名的有效字符序列称为标识符(identifier)。简单地说，标识符就是一个名字。

对于 Visual Basic，标识符的命名规则是：

·只能由字母、数字和下划线组成；

·必须以字母开头，最后一个字符可以是类型说明符；

·不能包含嵌入小数点和其他类型声明字符；

·不能超过 255 个字符，而且名字的有效字符为 40 个；

·在变量范围中必须唯一；

·不能用 Visual Basic 的保留字(关键字)作变量，但可以把保留字嵌入变量名中；变量名也不能是末尾带有类型说明符的保留字。例如，变量 Print 和 Print $ 都是非法的变量名，但变量 Pint_Page 是合法的变量名。

如下面是合法的标识符，也是合法的变量名：

dateoftoday, present, student_name, student_id

而下面是不合法的标识符和变量名：

3d, a * b, D.Lin

Visual Basic 不区分变量名和其他名字中字母的大小写，dataoftoday、DataOfToday、DATAOFTODAY 指的都是同一个变量。也就是说，在定义一个变量后，只要字符相同则不管其大小写是否相同，Visual Basic 都认为它们代表的同一个变量。为了便于阅读和理解，每个单词开头的字母一般用大写，即大小写混合使用组成变量名(或其他名字)，例如 DataOfToday。此外，符号常量一般用大写字母定义。

在多数编程语言中，变量都要声明。也就是说，要事先告诉编译器，准备使用那些变量。如果编译器知道变量及其类型，则可以产生优化的代码。因此可以产生最紧凑和最有效的代码，提高程序的速度和效率。而 Basic 语言的一个特点是，你用不着声明所有变量。这也是 Basic 语言的一个显著的特点，同时也是遭受非议的特点。

a.显式声明

要声明变量，用 Dim 语句，后跟变量名和类型。举例如下：

Dim Stud_Name As String

Dim Stud_Birthday As Date

第一个变量用来存放字符串变量，如“Zhangsan”，第二个变量用来存放日期型数据。有

关变量类型，稍后将详细介绍。

当然，你也可以在同一行中同时声明多个类型相同的变量，如下例所示：

Dim Stud_Name, Stud_Id As String

Visual Basic 在翻译程序时，发现 Dim 语句时，则自动根据语句中的指定生成一个或几个新变量，即在内存中保留一些空间并为其取名，生成占位符。后面程序中每次使用该变量名时，Visual Basic 即用这个内存区来读取或设置变量的值。如：

Stud_Name = "ZhangSan"

则 Visual Basic 自动将字符串"ZhangSan"赋给变量 Stud_Name，放到变量 Stud_Name 所占用的内存中。当程序要用到变量 Stud_Name 的值时，Visual Basic 从同一内存区中取出变量的值。如：

Messagebox Stud_Name

Visual Basic 自动从变量 Stud_Name 所处的内存中取出变量值"ZhangSan"，显示在屏幕上。

同时，与常量不一样，在程序中还可以重新设置变量的值，如：

Stud_Name = Stud_Name&"andLiSi"

则变量 Stud_Name 的值，即变量所处的内存中的值，现在被改为"ZhangSan and LiSi"。

在声明变量时，可以不定义变量的类型，Visual Basic 默认为变体型(Variant)，可以放置任何类型的一般变量。如

Dim temp

则变量 temp 的类型是变体型，它可以用来存放各种类型的数值。

声明变量的一个重要的原因是，让 Visual Basic 知道变量要存放的信息类型，并可以验证变量的数值是否和所定义的类型相符。如果赋给变量的值与数据型不相符时，就会产生类型不匹配(Type Mismatch)的运行错误。如在程序中给 Stud_Name 赋值如下(假定 Stud_Name 如上文被定义为 String 变量类型)：

Stud_Name = 123

即将一个整型数值赋给字符串类型的变量 Stud_Name，破坏了变量的声明类型，Visual Basic 不能执行这个语句，并产生一个类型错误。

b.隐式声明

在 Visual Basic 的程序中，还可以不声明变量。Visual Basic 翻译程序时，遇到没有声明的变量名时，它就临时生成新的变量，分配内存地址。新变量的类型为变体型，可以放置所有其他类型的数据。即等价于声明了一个变体型的变量，Visual Basic 根据变量被赋予的数值来调整变量类型。如：

temp1 = "ZhangSan"

temp2 = 234

Visual Basic 自动将"ZhangSan"赋给 temp1，并认为 temp1 是一个字符串变量，将 234 赋给 temp2，并认为 temp2 是一个数字变量。可以用 TypeName()函数返回变量的类型以验证变量的类型。

表 2　数据类型定义字符

符号	数据类型	举例
%	Integer	Count
&	Long	ColorValue

符号	数据类型	举例
!	Single	distance!
$	String	Text $
#	Double	Distance #

Print"temp1:"&Type Name(temp1)

Print"temp2:"&Type Name(temp2)

另外,可以省略声明语句而用变量声明字符生成确定类型的变量。在程序执行时,生成有确定类型的变量。如:

temp1 $ = "ZhangSan"

temp2% = 234

则 temp1 是字符串变量,而 temp2 是整型变量。表 2 列出了数据类型定义字符后缀。还有一种变量定义方法是,用 DefXXX 语句声明变量,如:

DefInt a - z

则所有以 a,b,…,z 打头的变量都是整型变量。其他的 DefXXX 语句有:

DefBool	DefByte	DefInt
DefLng	DefCur	DefSng
DefDbl	DefDate	DefStr
DefOb	DefVar	

以这种方式定义的变量只对它所在的模块起作用。类型说明符(%、&、#、!、@、$)总是比 DefType 语句优先起作用

这种定义方式已经过时,Visual Basic5.0 与旧版本 Basic 语言兼容,保留了这种定义方式。

Visual Basic 识别下列变量类型:

·Numeric 数字型

·String 字符串型

·Date 日期型

·Bool 布尔型

·Object 对象型

·Variant 变体型

数字型变量存放数字,字符串型变量存放文本,日期型变量存放日期数据,对象型变量存放各种类型的对象,而变体型变量存放各种类型的数据。从表面上看,好象变体型变量最好,一种类型就可以存放各种类型的变量,其实变体型的变量也存在各种不足,如程序运行速度慢。

下面分别介绍各种变量类型。

数字型变量

数字型变量的声明方式是:

Dim Count As Integer

Dim LngCount As Long

Dim a As Single

Dim b As Double

如变量声明中所举例子，数字型变量包含了以下几种数据类型：

·整型变量 Integer、Long

·精度较低的浮点数 Single

·精度较高的浮点数 Double

之所以有各种数字型的变量类型，主要是要提高程序的运行速度和效率，能用整型变量的地方，就不要用浮点数来表示，因为用整数可以大大提高程序的计算速度和运行速度。

Visual Basic 数字型数据类型可见表 3：

表 3　Visual Basic 数字型数据类型

数据类型	作　用
Integer(整型)	存放整数(－32768 到 32767)
Long(长整型)	存放整数(－2147483648 到 2147483647)
Single(单精度数)	存放单精度浮点数
Double(双精度数)	存放双精度浮点数
Currency(货币型)	存放四个小数位的定点数

Integer(整型数)在内存中占有两个字节，可以存放从 -2^{15} 到 $2^{15}-1$ 即(－32768 至 32767)之间的任何一个整数(可以用两个字节来表示)。Long(长整型数)在内存中占有四个字节，可以存放从 -2^{31} 到 $2^{31}-1$(即－2147483648 到 2147483647)之间的任何一个整数。如果一个变量的值不可能大于 32767，同时不可能小于－32768，则建议采用 Integer 型变量，因为 Integer 型变量的运算速度最快。对于浮点数，则需采用 Single、Double、Currency 型的变量。Single 型变量占有四个字节的内存空间，Double 型变量占有八个字节的内存空间，Single 型变量和 Double 型变量的区别主要是两者所表示数值的精度不同，从下列可以看出这一点：

```
Dim a As Single
Dime b As Double
a = 5/3
b = 5/3
Msgbox a
Msgbox b
```

程序运行结果分别是 a = 1.666667，而 b = 1.6666666666667。

从结果可以看出，Double 型的变量精度较 Single 型的变量要高，如果程序用于数学计算，则选用 Double 型变量。对于一般的没有较高精度要求的计算，可以用 Single 型变量，因为较之 Double 型，Single 型变量的运算速度较快。Currency 型用于存放定点数，正如其各称所代表的，该数据类型适用于财务计算。

字符串型变量

字符串型变量用来存放文本，其变量声明方式是：

```
Dim Stud_Name As String
Dim Number_String As String
Dime String_LengthIsTen As String * 10
```

字符串型变量可以赋予任何文件，如：长度 10 字节

```
Stud_Name = ""
Stud_Name = "ZhangSan"
```

Stud_Name = "Zhang San"

Number_String = "1234"

String_LengthIsTen = "abcdefghigklmn"

上面的字符串变量赋值语句都是正确的，其中第一个语句将一个字符串赋给变量 Stud_Name，字符串可以包含空格，最后一句语句将一个只包含数字的字符串赋给变量 Number_String。

在字符串变量定义例子里的最后一个例子定义了一个字长字符串 String_LengthIsTen，该字符串长度是确定的，最多只能是 10 位，当赋给该字符串变量的字符串长度大于 10 位时，Visual Basic 自动将超出 10 位的后面的字符给截掉，即将前 10 位字符赋给该字符串，如上例，String_LengthIsTen 字符串变量的值是"abcdefghig"，而当赋予一个字符串变量的字符串长度小于 10 位时，则 Visual Basic 自动用空格将变量填满。

日期型变量

日期型变量的声明方式是：

Dim DateOfBirthday As Data

日期型变量可以存放日期或时间值。如下例所示：

DataOfBirthday = "10/20/96"

DateOfBirthday = # 10/20/96 #

DataOfBirthday = "15:05:54PM"

DataOfBirthday = # 15:05:54PM #

DataOfBirthday = "10/20/96 15:05:54PM"

DataOfBirthday = # 10/20/96 15:05:54PM #

Visual Basic 能够自动处理日期型变量值，用 Data()函数可以获得系统日期，Time()函数可以获得系统时间，用 Year()、Month()、Day()可以从日期变量中取出年、月、日。日期型变量可以进行加、减计算，但不能进行乘除运算。下面举例说明：

```
Dim DateOfToday As Date
Dim DateOfYestoday As Date
Dim DateOfToday = Data()
DateOfYestoday = DateOfToday - 1
Msgbox DateOfYestoday
Msgbox Year(DateOfToday)
Msgbox Month(DateOfToday)
Msgbox Day(DateOfToday)
```

假设今天的日期是 03/23/98，则上例中的结果是：

03/22/98

1998

3

23

Bool 布尔型

布尔型变量的声明如下：

```
Dim bEnabled As Boolean
```

布尔型变量的缺省值是 False。

布尔型变量可以由 NOT、AND、OR、XOR 等逻辑运算符进行组合。

布尔型变量用来存放数值 True 和 False，在 Visual Basic 中，布尔型变量用两个字节存放变量值，而在内存中，布尔值实际上是 -1(True)和 0(False)

Object 对象型

Object 变量是对 Visual Basic 的多种对象之一的引用，可以对 Object 变量访问实际对象，下面是 Object 变量的一个简单的例子。一个窗体上有两个命令按钮 Command1 和 Command2，可以声明两个 Object 变量如下：

```
Dim a As CommandButton, b As CommandButton
```

每个 Object 变量都可用下列语句设置两个命令按钮之一：

```
Set a = Command1
Set b = Command2
```

从现在起，可以通过变量 a 和变量 b 操作这两个命令按钮的属性，如要改变第一个命令按钮的 Caption 属性，可以用下列语句：

```
a.Caption = "OK!"
```

要设置第二个命令按钮的 Enabled 属性，使其按钮变灰，不能操作，则可以使用下列语句：

```
b.Enabled = false
```

以后各章将会有更多的对象变量的例子，还会介绍如何生成自己的对象及其属性和方法。

Variant 变体型

变体型变量(Variant)是最灵活的数据类型，可以适用于所有其他的数值类型。Visual Basic 根据变量的当前内容，处理声明为变体型变量和没有声明的变量，如果向一个 Variant 变量赋予整数值，则 Visual Basic 把它当作整型变量；如果向一个 Variant 变量赋予字符串，则 Visual Basic 把它当作字符串变量。Variant 还可以在同一程序运行期间放置不同类型的数据，Visual Basic 会进行必要的转换。

要声明变体型变量，在 Dim 语句中不指定数据类型：

```
Dim myVar
```

还可以指定 Variant 类型，使程序更清晰：

```
Dim myVar As Variant
```

或者根本不声明此变量的类型，在程序中引用新变量时，Visual Basic 为其建立一个 Variant。如果变量 Another Var 没有声明，则 Visual Basic 遇到下列代码时：

```
AnotherVar = "Basketball"
```

会生成一个新的变体型变量，并向其赋值"Basketball"。

变体型变量也可以进行数字和字符串计算，下面有一个例子：

```
A = "10"
B = "11"
Debug.Print A + B
Debug.Print A & B
```

两个 Print 语句都打印字符串"1011"。你的原意是两个字符串相加，Visual Basic 也是这

样理解的。对于字符串，+和&运算符的执行结果是一致的，如果将第二个变量的定义变成为：

B = 11

则第一个 Print 语句打印数字值 21，而第二个 Print 语句打印字符串“1011”。

因此，在使用变体型变量进行计算时，应仔细思考。

有时需要将一种变量类型转换成另一种变量类型，表 4 列出了进行变量类型转换的 Visual Basic 函数。

表 4　Visual Basic 数据变量类型函数

函　数	说　明	函　数	说　明
Cbool	将变量转换成布尔值	CLng	将变量转换成长整型
Cbyte	将变量转换成字节型	CSng	将变量转换成单精度型
Ccur	将变量转换成货币型	CStr	将变量转换成字符串型
Cdate	将变量转换成日期型	CVar	将变量转换成变体型
CDbl	将变量转换成双精度型	CVErr	将变量转换成错误值
Cint	将变量转换成整型		

变量的作用域

变量的作用域指变量的有效作用范围。有定义了一个变量后，为了正确地使用变量，应当明确在程序的什么地方可以访问该变量。变量作用范围根据变量的声明来确定，有局部变量和全局变量两种类型。

变量是在应用程序中使用的。为了说明变量的作用域，我们先来看一看 Visual Basic 应用程序的结构。

Visual Basic 应用程序由窗体(Form)和模块(Module)组成。窗体由事件过程(Event Procedure)、通用过程(General Procedure)和声明部分组成；而模块由通用过程和声明部分组成。根据定义位置和所使用的变量定义语句的不同，Visual Basic 中的变量可以分为三类，即局部变量(Local)、窗体和模块变量(Form and Module)及全局变量(Public 或 Global)。各种变量位于不同的层次。它们的作用域如表 5 所示。

表 5　变量的作用域

名　称	作用域	声明位置	使用语句
局部变量	过程	过程中	Dim、Static
模块级变量	窗体及模块	模块或窗体的声明部分	Dim
全局变量	整个应用程序	模块的声明部分	Public、Global

局部变量

在过程内定义的变量叫局部变量，其作用域是它所在的过程。局部变量通常用来存放程序的中间结果或用来用作临时变量。应用程序的某一过程的执行只对该过程内的变量产生作用，对其他过程中的相同名字的局部变量没有任何影响。因此，在不同的过程中可以定义相同名字的局部变量，它们之间没有任何关系，即在一个过程中变量与另一个过程中的变量之间是两个不同的变量。

局部变量的声明是在过程中用 Dim、Static 定义，如：

```
Sub Form_Load(Cancel As Integer)
    Dim Stud_Name As String
    Static Counter As Integer
```