

德望

IT学院



附CD-ROM 1张

Visual Basic 程序开发 完整实例教程

赵学慧 赵 玮 编著

全书完整
技巧放送
深入浅出
举一反三
深入探讨
传授真经
拓展思路
触类旁通



海洋出版社

Visual Basic 程序开发 完整实例教程



Visual Basic Chengxu Kaifa Wanzheng Shili Jiaocheng

B

Visual Basic

程序开发完整实例教程

赵学慧 赵 玮 编著

2003年·北京



海洋出版社

内 容 提 要

本书以具体实例为基础,循序渐进地讲解了 Visual Basic 程序开发的过程和方法,使读者在实例开发过程中逐步掌握应用 Visual Basic 开发程序。

为了使广大读者能够快速、深入地应用 Visual Basic 进行开发程序,在编写本书时,精选了一些具有代表性的完整开发实例,并将 Visual Basic 的新特点、新功能融合到实例中,对利用 Visual Basic 开发程序的过程和步骤进行了详细讲解,并且还以技术专题、举一反三的形式,延伸思考,加深读者对 Visual Basic 程序开发方法的理解。

本书面向中高级用户,适合作为利用 Visual Basic 程序开发人员的自学指导书。本书配套光盘中包含书中的所有实例及源文件,以供读者参考学习。

图书在版编目(CIP)数据

Visual Basic 程序开发完整实例教程/赵学慧,赵玮编著. - 北京:海洋出版社,2003. 3
ISBN 7-5027-5859-3

I. V … II. 赵… III. BASIC 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 021442 号

责任编辑 王 勇

责任印制 严国晋

海洋出版社 出版发行

<http://www.oceanpress.com.cn>

(100081 北京市海淀区大慧寺路 8 号)

北京海洋印刷厂印刷 新华书店发行所经销

2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

开本: 787mm×1092mm 1/16 印张: 18.75

字数: 460 千字 印数: 1~6000 册

定价: 32.00 元

海洋版图书印、装错误可随时退换

前言

Visual Basic是一种简单易学的编程语言,是目前Windows平台上可视化程度最高的开发工具,它提供了丰富的功能组件,适合开发Windows操作系统下的各种应用程序。已成为专业人员和非专业人员得心应手的开发工具。Visual Basic以其功能强大、简单易学等特点而倍受人们的青睐。

版本的更新和功能的增强,使得Visual Basic的应用范围越来越广。它不但能满足一般应用程序的需要,而且在ActiveX、多媒体、数据库以及Internet等多个领域都有着十分明显的优势。

大多数学过Visual Basic的人员,都希望能深入了解Visual Basic的奥秘,以便能开发出功能更强大的应用程序。本书正是为了满足这些读者的需求而编写的。本书第1章简单地介绍了Visual Basic语言的基本语法规则。其后几章从不同的技术点出发,以具体实例的方式介绍Visual Basic的各种功能。第2章和第4章模拟了Windows附件中的“画图”程序和“记事本”程序;为了增加趣味性,本书还编写了两个游戏程序,即第3章的“趣味打字”和第10章的“俄罗斯方块”;第5章介绍了Windows中各种风格的工具栏的编写;第6章用一个简单的时钟程序介绍用Visual Basic如何生成ActiveX控件;第7章介绍了如何使用Visual Basic编写的应用程序成为屏幕保护程序;第8章通过设计MP3播放器、标准音频文件播放器、VCD播放器来介绍Visual Basic的多媒体技术;第9章编写了一个类似ICQ的聊天程序,读者通过这个程序可以对Visual Basic的网络编程有一个大概的了解;第11章通过一个具体的工资管理程序,介绍了Visual Basic中的数据库编程。

本书在每个实例的后面都有一个技术专题,对本实例所涉及到的技术点有一个更详细的介绍。每个实例的最后,都有举一反三内容和上机练习,使读者对所学的知识有更多的了解和提高。

本书配套光盘中包含了书中的实例效果及源文件。

为读者服务、对读者负责是深源图书创作室的工作宗旨;严谨、创新、求实是深源图书创作室的工作态度;高品位、高质量是深源图书创作室孜孜以求的目标。尽管倾心相注、精心而为,但书中的错误和不足在所难免,恳请读者批评指正。

编著者

2003年3月



目次

第1章 Visual Basic 程序设计概述

1.1	VB 特点	2
1.2	VB 中的基本概念	2
1.3	VB 语言基础	5
1.3.1	代码书写规则	5
1.3.2	变量	7
1.3.3	常量	9
1.3.4	数据类型	10
1.3.5	数组	14
1.3.6	动态数组	16
1.3.7	过程	17
1.3.8	程序流程的控制	24
1.4	集成开发环境(IDE)	31

第2章 模拟 Windows 中的画图程序

2.1	实例简介	34
2.2	预习要求	34
2.3	设计思路	35
2.4	设计步骤	35
2.4.1	建立应用程序界面	36
2.4.2	菜单设计	36
2.4.3	设置对象的属性	36
2.4.4	编写程序代码	37
2.5	技术专题	43
2.5.1	VB 中的坐标系统	43
2.5.2	VB 中的鼠标事件	44
2.5.3	DrawMode 属性	45
2.6	举一反三	46
2.6.1	简单的计算器	46
2.6.2	沿正弦曲线滚动的小球	46
2.7	上机练习	47

第3章 实用的打字小游戏

3.1	实例简介	50
3.2	预习要求	50
3.3	设计思路	51
3.4	制作过程	51



3.4.1	窗体设计	51
3.4.2	功能实现	53
3.5	技术专题	58
3.5.1	控件数组的应用	58
3.5.2	KeyPress、KeyUp 事件	59
3.6	举一反三——为程序添加声音效果	59
3.7	上机练习	60

第4章 模拟 Windows 中的记事本程序

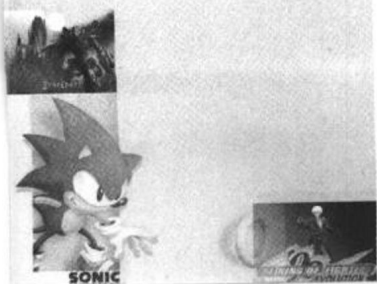
4.1	实例简介	62
4.2	预习要求	63
4.3	设计思路	63
4.4	设计步骤	64
4.4.1	建立应用程序的界面	64
4.4.2	编写程序代码	68
4.5	技术专题	80
4.5.1	剪贴板的使用	80
4.5.2	公共对话框控件	81
4.5.3	文件处理技术	85
4.6	举一反三——多文档记事本的制作	91
4.6.1	多文档界面	91
4.6.2	创建 MDI 窗体	92
4.6.3	加载 MDI 窗体及其子窗体	93
4.6.4	设置子窗体的大小和位置	94
4.6.5	设置子窗体的排列方式	94
4.6.6	显示所有打开子窗体的标题	95
4.6.7	卸载 MDI 窗体及其子窗体	95
4.7	上机练习	96

第5章 用标准控件制作工具栏

5.1	实例简介	98
5.2	预习要求	99
5.3	设计思路	99
5.4	设计步骤	100
5.4.1	第一种风格的工具栏的设计	100
5.4.2	第二种风格的工具栏的设计	110
5.5	技术专题——容器	118
5.6	举一反三	119
5.7	上机练习	121

第6章 设计时钟 ActiveX 控件

6.1	实例简介	124
6.2	预习要求	124



6.3	设计思路	125
6.4	设计步骤	125
6.4.1	创建 ActiveX 控件工程	125
6.4.2	在 ActiveX 控件工程中添加需要的对象	126
6.4.3	建立时钟 ActiveX 控件使用的资源	127
6.4.4	设计时钟 ActiveX 控件的代码	128
6.4.5	测试及调用时钟 ActiveX 控件	130
6.4.6	为时钟 ActiveX 控件添加属性	131
6.4.7	为时钟 ActiveX 控件添加事件	132
6.4.8	为时钟 ActiveX 控件添加事件	133
6.4.9	编译成 .OCX 格式	134
6.5	技术专题——设计 ActiveX 控件的基础知识	134
6.5.1	UserControl 对象的基本知识	135
6.5.2	VB 中创建控件的方法	135
6.6	举一反三	137
6.7	上机练习	139
第 7 章 屏幕保护程序		
7.1	实例简介	142
7.2	预习要求	144
7.3	设计思路	144
7.4	设计步骤	145
7.4.1	建立屏幕保护程序的界面	145
7.4.2	设置对象的属性	146
7.4.3	设计代码	147
7.5	技术专题——管理应用程序的设置值	159
7.5.1	SaveSetting 语句	160
7.5.2	GetSetting 函数	161
7.5.3	GetAllSettings 函数	161
7.5.4	DeleteSetting 语句	161
7.6	举一反三	163
7.7	上机练习	165
第 8 章 多媒体播放程序		
8.1	实例简介	168
8.2	预习要求	168
8.3	设计思路	169
8.4	设计步骤	170
8.4.1	应用程序主界面的设计	170
8.4.2	MP3 播放器的设计	171
8.4.3	标准音频文件播放器的设计	175
8.4.4	VCD 播放器的使用	179
8.5	技术专题——利用多媒体信息制作课件	182



8.5.1	使用 VB 制作的多媒体课件的类型	182
8.5.2	使用 VB 制作多媒体课件的流程	183
8.5.3	素材的具体使用和设计	184
8.6	举一反三	185
8.7	上机练习	186

第 9 章 自己的聊天工具

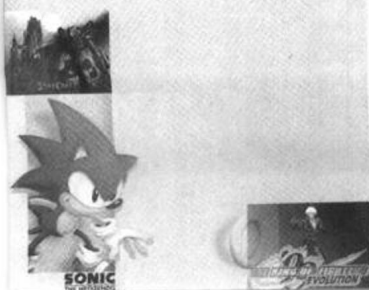
9.1	实例简介	188
9.2	预习要求	189
9.3	设计思路	190
9.4	设计步骤	190
9.4.1	建立启动窗体	190
9.4.2	建立客户端窗体	191
9.4.3	建立服务器的窗体界面	192
9.4.4	编写程序代码	193
9.5	技术专题	203
9.5.1	Internet 基础知识	203
9.5.2	Winsock 控件	203
9.5.3	WebBrowser 控件和 Internet Explorer 对象	205
9.6	举一反三——制作专用 Web 浏览器	210
9.7	上机练习	211

第 10 章 “俄罗斯方块” 游戏

10.1	实例简介	214
10.2	预习要求	216
10.3	设计思路	216
10.4	设计步骤	217
10.4.1	建立应用程序的界面	217
10.4.2	添加资源文件	219
10.4.3	排行榜窗体的设置	220
10.4.4	编写程序代码	221
10.5	技术专题——Windows API	244
10.5.1	声明 API 函数	245
10.5.2	使用“API 浏览器”应用程序	245
10.5.3	API 函数参数	246
10.5.4	介绍两个常用的 API 函数	247
10.6	举一反三	249
10.7	上机练习	250

第 11 章 工资管理系统

11.1	实例简介	252
11.2	预习要求	252
11.3	设计思路	252



11.4	设计步骤	254
11.4.1	主控模块的设计	254
11.4.2	登录对话框的设计	256
11.4.3	建立数据库文件	257
11.4.4	在工程中添加“数据环境”	258
11.4.5	“增加”模块的设计	259
11.4.6	“修改”模块的设计	262
11.4.7	“删除”模块的设计	266
11.4.8	“浏览”模块的设计	267
11.4.9	“批处理”模块的设计	269
11.4.10	“打印”模块的设计	271
11.4.11	打印工资报表——DataReport1.DSR 的设计	273
11.4.12	打印工资条——DataReport2.DSR 的设计	274
11.4.13	工程的编译	275
11.4.14	创建数据库应用程序的安装包	276
11.5	技术专题——数据库应用程序开发	278
11.5.1	基本的数据库组织	279
11.5.2	数据库应用系统开发的一般步骤	281
11.5.3	SQL 语言简介	283
11.5.4	数据访问技术概述	285
11.6	举一反三	287
11.6.1	Connection 对象	288
11.6.2	Command 对象	288
11.6.3	Parameter 对象	288
11.6.4	Recordset 对象	288
11.6.5	Field 对象	289
11.7	上机练习	290



Visual Basic 程序开发完整实例教程



第1章 Visual Basic 程序设计概述

本章将简单介绍Visual Basic(以下简称VB)语言的特点、程序设计中常用的基本概念、语法基础和集成开发环境(IDE)，这可以更好地帮助我们理解与学习后面介绍的实例。

Visual Basic

1.1 VB 的特点

VB 是 Microsoft 公司开发的一个强有力的应用程序开发工具，使用它可以开发出 Windows 环境下的任何应用程序。

Visual 的英文原意是“可视的”，在这里是指开发图形用户界面（GUI）的方法，即“可视化程序设计”。BASIC 是 Beginners All-purpose Symbolic Instruction Code 的缩写，意思是“初学者通用符号指令代码”。具有面向普通使用者、易学易用的优点。VB 6.0 在原有的 BASIC 语言的基础上有了很大的发展，至今包含了数百条语句、函数及关键词，其中很多和 Windows GUI 有直接的关系。专业人员可以用 VB 实现其他任何 Windows 编程语言的功能，而初学者只要掌握几个关键词就可以编写实用的应用程序。

VB 已由小巧玲珑的编程工具发展成为强大的开发工具。从数值计算、数据库管理、网络、多媒体软件，到 Internet 软件都可以用 VB 开发完成。它之所以有这么广泛的用途是因为具有以下特点：

- 真正的面向对象编程，提高了系统的开发速度，减轻了程序员开发系统的工作量。
- 提供了可视化的编程工具。利用 VB 的工具箱，可以在窗体上像画图一样“画”出各种对象，而不用编写任何代码。这其实是 VB 最显著的特点。
- 程序采取“事件驱动”方式，即程序的执行是由事件来驱动的，执行的顺序由用户决定。这样，对于程序员来说，只需要分别编写出在某个对象上发生的某一事件时要执行的代码即可。
- 数据访问特性允许对包括 Microsoft SQL Server 和其他企业数据库在内的大部分数据库格式建立数据库和前端应用程序。
- 通过 ActiveX 技术可以使用其他应用程序提供的功能。

另外，VB 还支持很多功能，如 Internet 功能、多窗口平台、OLE 编程能力、快速应用开发以及可生成真正的 .exe 文件。这些特性使 VB 成为快速创建全特性应用程序的一种优秀的编程语言。

1.2 VB 中的基本概念

真正理解了 VB 中的基本概念和常用术语，可以帮助我们更好地学习这门语言。

(1) 设计时

指开发人员在 VB 开发环境中开发应用程序时的时间。

(2) 运行时

指应用程序正在运行时的时间。在运行时，编程者就像最终用户那样与应用程序进行交互。



(3) 窗体

也就是未来应用程序的窗口。指 VB 为程序设计人员提供的可供设计应用程序的界面, 或各种对话框的窗口。

(4) 控件

可放置在窗体上的对象, 也就是应用程序运行时出现在窗口中的内容。

(5) 对象

对象是 VB 程序设计中最基本、最重要的概念, 是程序设计的核心。一般情况下, 对象是指程序设计人员在编程时可以访问的元素。在 VB 中, 对象是用来描述组成一个应用程序的所有窗体和控件的一个通用术语。

所有对象都有两个共同点: 一是自己的状态或外观, 如果用数据来表示, 这些数据就称为对象的“属性”; 二是自己所具有的行为, 可以用代码来实现, 这些代码称为对象的“方法”。所以对象是数据和代码的集合。即对象是既包含数据又包含处理该数据的代码的独立个体, 是类的实体化。

(6) 类

类是为描述具有相同特征的对象而引入的。类是用来创建对象的样板, 它包含所创建对象的状态描述和方法定义, 而对象只是类的一个实例。

(7) 属性

属性是对象的一个基本要素, 用来描述对象的外观特征和所具有的状态; 不同的对象具有不同的属性, 当然有时也会出现一些相同的情况。属性一般用数据来表示。通过修改对象的属性值, 就可以控制对象的外观和状态。

有些属性只在设计时提供, 有些属性只在运行时提供。如文本框控件的 `MutiLine` 属性只能在设计时设置, 不能在运行时改变; 而列表框的 `ListCount` 属性只能在运行时提供, 属于只读属性。

几个常用的属性:

- `Name`: 设置控件的名称, 在程序代码中用来访问控件的属性和方法。
- `Appearance`: 用来表示对象的外观, 0 表示平面外观, 1 表示三维外观。
- `BackColor`: 设置显示文字或绘制图形时的背景颜色。
- `ForeColor`: 设置前景色 (画笔的颜色或文本的颜色)。
- `Font`: 设置控件上文本的字体、字形和字号。
- `Caption`: 在许多不接受输入的控件上设置所显示的文本, 如命令按钮控件的标题和复选框与单选钮控件旁边显示的字符串。
- `Text`: 在接受输入的控件上设置显示的文本, 如文本框控件。
- `Width`、`Height`: 设置控件的尺寸。
- `Left`、`Top`: 设置控件左上角的坐标。



- Enabled: 缺省情况下, 这个属性值为 True, 表示控件可以变为焦点; 设置为 False 则关闭控件。关闭的控件显示为灰色, 无法接受用户的输入。

- Visible: 将这个属性设置为 False 时, 控件不可见。有时用不可见的控件保存应用程序内部使用的信息, 使用户无法见到该控件或不能对该控件进行操作。

(8) 事件

事件就是对象上所发生的事情。在 VB 中, 事件是指由系统预先定义好的、能够被对象(窗体或控件)识别和响应的动作。不同类型的对象所能识别的事件是不同的。每个对象的事件是固定的, 不能建立新的事件。

事件可分为“用户事件”和“系统事件”两种类型。所谓用户事件是指由用户通过键盘上的键或鼠标操作而产生的事件; 而系统事件是指由系统本身产生的事件。如定时器控件的 Timer () 事件就是典型的系统事件。

两组最常见的事件是鼠标事件和键盘事件。

常用的鼠标事件有以下几种:

- Click、DbClick: Click 事件在用户单击鼠标左键时发生; DbClick 事件在用户双击鼠标左键时发生。

- MouseDown、MouseUp: MouseDown 事件在鼠标键按下时发生, 而 MouseUp 事件在释放鼠标键时发生。

- MouseMove: 这个事件在鼠标经过控件时连续发生。

鼠标事件发生的顺序如下:

- ① 鼠标移动时, 连续触发 MouseMove 事件。
- ② 用户按下鼠标时, 触发 MouseDown 事件。
- ③ 用户按下鼠标键连续移动时, 程序不断收到 MouseMove 事件。
- ④ 用户放开鼠标键时, 触发 MouseUp 事件。
- ⑤ 如果按住鼠标左键, 则 Click 事件在 MouseUp 事件之后立即触发。
- ⑥ 双击鼠标时, 发生下列事件: MouseDown、MouseUp、Click、DbClick、和 MouseUp。

常用的键盘事件有:

- KeyDown、KeyUp: KeyDown 事件在键按下时触发, KeyUp 事件在键弹起时触发。

- KeyPress: 实际中, 程序并不关心 KeyDown 和 KeyUp 事件, 而要用 KeyPress 事件确定按下的是哪个键。KeyPress 事件常用于编写文本框的事件处理器, 因为这个事件发生在字符键按下和显示在文本框中之前。用这个事件可以抑制某些字母或数字的输入。

- Change: Change 事件在各种控件的内容改变时触发。Change 事件当在文本框控件中输入或删除新字符时产生, 当在复选框控件中用户单击改变其状态时也可产生。

(9) 事件过程

对象可以接受事件, 并对接受的事件做出反应。当事件由用户触发(如单击鼠标)或由系统触发(如装载窗体)时, 对象就会对该事件做出响应, 响应某个事件后所执行的程序代码就是“事件过程”。



在“事件过程”中必须编程控制这些事件的响应方法。因为应用程序并不决定程序的流程，而是由用户确定的事件决定应用程序的流程。事件过程只有在事件发生时，系统才会访问它，并且执行事件过程中的程序代码。没有事件时，整个事件过程处于“停滞”状态，所以事件过程又可称为“事件驱动程序”。

(10) 方法

方法就是针对每个控件事先设计好的、不需要编程人员帮助即可以完成的操作。方法隐藏了控件特性的实现细节，是 VB 提供的专门子程序，用户能做的工作就是按照约定直接调用它们。如在程序中调用窗体的 Cls 方法就可以清除窗体，调用它的 Hide 方法就可以隐藏窗体。

几个常用的方法如下：

- Clear：告诉控件要放弃它的内容。如果对象是列表框，则 Clear 方法删除控件中的所有项目。Clear 方法也可用于清除剪贴板中的内容
- Move：所有运行时可见的控件都有 Move 方法，用于在应用程序代码中调整控件的大小和位置。
- AddItem、RemoveItem：这些方法用于操作列表框和组合框控件中的项目。应用程序不必知道项目在控件中如何存放，只要发出 AddItem 方法，控件就会负责向表中添加或插入新项目，这就是使用方法的原因。

1.3 VB 语言基础

本节对编写代码应该掌握的基本语法作了初步介绍。只有熟练地掌握了这些内容，才能在编程时开发出功能强大的 VB 应用程序。

1.3.1 代码书写规则

(1) 续行

当一条语句太长时，就要用续行符“_”（空格后跟一个下划线）将其分成多行。例如：

```
Picture2.Print Tab (3); i; Tab (10); Top _ Data (i) .TopName; _  
Tab (25); Top _ Data (i) .TopScore
```

(2) 注释

在程序中使用注释是一个良好的习惯。可以用注释来说明某段代码或声明某个变量的用途，这样做既方便自己，又方便以后阅读代码的其他编程人员。

VB 使用注释符“'”作为注释文字的开始。注释符告诉 VB 忽略该符号后面的内容。注释可以和语句在同一行，也可以单独占一行。例如：

```
Form1.Caption = "记事本" '修改窗体的标题文字  
' 在窗体的中心画一个红色的圆
```



```
Form1.Circle (Form1.ScaleWidth/2, Form1.ScaleHeight/2), _  
Form1.ScaleHeight/2.5, vbRed
```



一点通：在同一语句中，续行符后面不能加注释。

(3) 将多行语句写到一行

VB 代码通常是一行一条语句。如果想将两行或多行语句放在同一行中，那么必须用冒号“:”将其分开。例如：

```
X = 13 : Y = 40
```

但从代码易读性角度考虑，最好还是一行放一条语句。



一点通：VB 语句不用结束符，这一点与有的语言不同，例如 C 语言，其语句最后就要加一个分号“;”作为结束符。

(4) 使用不同进制的数字

VB 中的数字默认是十进制的，但在某些场合下使用其他进制（如八进制或十六进制）的数字可能会更方便一些。VB 用前缀 &H 表示十六进制数，而用 &O 表示八进制数。例如，十进制数 255，用八进制表示，则为 &O377；而用十六进制表示，则为 &HFF。

(5) 命名约定

编程序时经常需要声明和命名许多过程、变量和常量。声明时，过程、变量和常量的名字必须遵守以下命名约定：

- 名字必须以字母开头，而不能用数字或其他字符开头。
- 名字由字母、数字或下划线（_）组成。
- 名字中不能含有句点（.）或类型声明字符。
- 名字最长不超过 255 个字符。控件、窗体、类和模块的名字不能超过 40 个字符。
- 名字不能是关键字。关键字是语言的组成部分，如预定义语句（If、Loop 等）、函数（如 Len 和 Abs 等）和运算符（如 Or 和 Mod 等）等。

窗体和控件的名字可以与关键字同名。例如，可以将某个控件命名为 Len。但不能用通常的方法来引用与关键字同名的窗体和控件，因为 VB 默认将 Len 看成关键字。例如，以下语句就会产生错误：

```
Len.Visible = True
```

为了引用与关键字同名的窗体或控件，就必须进行限定或将其用方括号（[]）括起来。例如，以下代码就不会产生错误：

```
Form1.Len.Visible = True
```

```
[Len].Visible = True
```

虽然可用方括号对窗体和控件的引用进行限制，但是建议读者尽量少用关键字作为窗体和控件的名字。



1.3.2 变量

程序中经常需要临时存储数据，这些数据一般在设计时是未知的。想临时存储数据，就要用到变量。变量是一个被命名的内存位置，变量中存储的数据在程序运行期间可以发生改变。变量有名字和数据类型。变量的名字表明变量所存储的信息；而数据类型确定变量所能存储信息的种类，如文本、日期、数字和对象等。

(1) 声明变量

使用变量之前，最好先声明变量。声明变量就是事先将变量的有关信息通知程序。变量的声明有两种方式，即显式声明和隐式声明。

1) 显式声明

Dim 语句用于显式声明变量，语法格式为：

```
Dim 变量名 [As 数据类型]
```

Dim 是关键字，变量名用于在程序中引用变量，As 子句指定变量的数据类型。省略 As 子句时，变量的数据类型被默认为 Variant 型。Variant 是变体数据类型，用于在不同的场合表示不同的数据类型。

以下是显式声明的例子：

```
Dim meters As Integer
```

```
Dim greetings As String
```

```
Dim Age
```

以上 3 个语句声明了 3 个变量 meters、greetings 和 Age，数据类型分别为 Integer、String 和 Variant。如果想在同一行中声明多个变量，那么必须用逗号“,” 隔开。例如：

```
Dim meters As Integer, greetings As String
```

如果声明变量时用关键字 Private、Static 和 Public 代替 Dim，那么可以声明模块级变量、静态变量和全局变量，这些变量的具体含义将在后面介绍。例如：

```
Private meters As Integer
```

```
Static greetings As String
```

```
Public fail as Boolean
```

变量声明后，就可以用赋值语句来存储和检索变量中的数据。例如：

```
meters = 23
```

```
greetings = "Thank you for your help"
```

以上语句中的等号“=” 是赋值号，而不是运算符“等于”。

2) 隐式声明

也可以不声明变量。VB 遇到没有声明的变量名时，它临时生成新的变量并使用这个变量。新变量的类型为 Variant。在程序中使用新变量等于不带类型的声明。如：

```
Dim Var1, Var2
```

```
Var1 = "Thank you for your help"
```

```
Var2 = 49.98
```

Var1 变量是个字符串变量，而 Var2 是个 Double 变量。此后如果在程序中倒过来赋值：