

125

TP312BA
H179

Visual Basic 高级编程实例精解

Visual Basic 高级编程实例精解

黄志峰 编著
雨 人 策划

国防工业出版社

· 北京 ·



A0950372

图书在版编目 (CIP) 数据

Visual Basic 高级编程实例精解/黄志峰编著. —北京: 国防工业出版社, 2001.6

(软件高级编程实例精解丛书)

ISBN 7-118-02470-8

I. V... II. 黄... III. BASIC 语言-程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 01980 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码: 100044)

三河市腾飞胶印厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 28¹/₄ 654 千字

2001 年 6 月第 1 版 2001 年 6 月北京第 1 次印刷

印数: 1—3000 册 定价: 40.00 元

(本书如有印装错误, 我社负责调换)

前 言

本书介绍的是 Microsoft Visual Basic 的高级开发技术。

Visual Basic 是 Microsoft 公司的 Visual Studio 开发产品系列的一员。2000 年末, Microsoft 公司推出了 Visual Studio 的最新版本 Visual Studio.NET 的 Beta1, 今年即将推出包括 Visual Basic.NET 在内 Visual Studio.NET 的正式版。Visual Basic.NET 在 Visual Basic 6.0 的基础上实现了飞跃。

Visual Basic 的基础是 Basic 语言, Basic 语言简单易学, 再加上 Visual Basic 的可视化编程更是如虎添翼, 所以掌握 Visual Basic 编程技术的程序员非常之多。

然而, Visual Basic 入门容易, 精通则难。正是 Visual Basic 的简单易用以及可视化编程手段把 Windows 编程的精髓“掩盖”了, 所以要真正掌握 Visual Basic 编程技术的精华非一朝一夕能做到的。

本书将以一种新的思路来学习 Visual Basic 的高级编程技术。本书分 6 章, 从 Visual Basic 的对象编程、使用固有控件编程开始, 由简单到复杂, 一步步将初学者领进 Visual Basic 高级开发的“殿堂”。

本书的最大特点就是对实例程序分析得十分透彻, 由“编程的准备”、“程序的分析”、“程序的编写”、“程序的完善”和“程序的扩展”五部分组成。实例是最好的“老师”, 结合实例来学习 Visual Basic 将会获得事半功倍的效果。

融入精彩纷呈的实例

本书的每一章都以一个实例程序“压轴”。

第 1 个实例是“打印预览程序”, 它主要讲述了如何使用 Visual Basic 固有控件编程。固有控件, 如 TextBox 控件、PictureBox 控件等等, 这些都是初学者比较熟悉的控件。然而, 要真正发挥这些控件的功能也不是简单的事情。“打印预览程序”是一个深入探讨 PictureBox 控件的使用的程序。

第 2 个实例是“多媒体播放器”, 它主要讲述了如何使用 Media Player 控件来编写一个多媒体播放器。ActiveX 编程技术是一个非常有用的技术, 其中 ActiveX 控件的使用非常广泛, 而且它提供的功能往往比 Visual Basic 的固有控件要强许多。在此后的章节里使用的控件几乎都是 ActiveX 控件。

第 3 个实例是“字体编辑浏览器”, 它主要讲述了如何在 Visual Basic 中使用 Windows API 函数来编写应用程序。Windows API 函数是 Windows 编程的重要基础, 能使用 Visual Basic 实现的功能, API 函数都能够实现, 而且 API 函数还能实现 Visual Basic 不能实现的功能。

能。掌握 Windows API 编程技术是加深对 Windows 编程理解的重要过程。

第 4 个实例是“文件管理浏览器”，它主要讲述了使用 Visual Basic 进行文件系统的编程。这个文件管理浏览器是模仿 Windows 的资源管理器来编写的，其中融合了 Visual Basic 的 FSO 对象模型、TreeView 控件、ListView 控件等的编程技术。

第 5 个实例是“图书馆管理系统”，它主要讲述了如何在 Visual Basic 环境下编写数据库应用程序。数据库编程是当今最热门的高级技术之一，Visual Basic 的 ADO 是最新的数据访问技术。“图书馆管理系统”这个实例完整地展现了一个数据库应用程序的编写过程。

第 6 个实例是“QICQ 程序”，它主要讲述了如何在 Visual Basic 环境下编写网络应用程序。QICQ 程序是基于数据库、客户/服务器端及网络的应用程序，它融合了数据库编程和网络编程的技术。QICQ 程序实现了 ICQ 的部分基本功能，充分地利用了 Winsock 控件的所有功能，实现以 TCP 和 UDP 协议来发送和接收数据。

请读者注意的是，由于 Visual Basic.NET 的正式版本还没有推出，所以本书的实例均在 Visual Basic 6.0 环境中构建，而且都在 Windows 98/Me 中运行无误。Visual Basic.NET 对 Visual Basic 6.0 完全兼容，所以本书的实例将很容易地升级到 Visual Basic.NET 中。

体验高级开发的乐趣

从这 6 个实例所涉及的内容来看，使用 Visual Basic 编程技术能够最大限度地实现我们的设想。

其实高级开发并不神秘！当你一步一步地接近成功时，一种前所未有的喜悦涌上心头。本书试图给读者营造一个充满动力和生机的气氛，让读者感受到编程乐趣。

“嘿！别再忧郁，开心些！其实编程很简单，你会有很大的成就感！”

记住，您是未来的高级开发者

本书面向那些初步掌握了 Visual Basic 编程语言的用户，因此本书没有介绍 Visual Basic 这个开发工具的基本使用方法，而只是对 Visual Basic 的编程思想以及编程语言进行了高度的概括，着重讲述 Visual Basic 的高级技术。如果你是刚入门或者还没有入门的用户，请在使用本书前先阅读有关 Visual Basic 入门的书籍。如果你已经是一个 Visual Basic 的编程高手了，或许这本书的技术对你来说已经非常熟悉，但本书每一章的例程的编写思路或许会对你有所启发。

最好地发挥本书的作用

本书共有 6 章。

第 1 章是有关固有控件的编程技术，它首先高度概括了 Visual Basic 独特的编程思想，然后介绍 Visual Basic 的对象编程和使用固有控件编程。这是每一个初学者成为编程高手必须掌握的基础知识。

第 2 章是有关 ActiveX 编程的技术。ActiveX 编程涉及的范围很广，包括 ActiveX DLL、ActiveX EXE、ActiveX 文档和 ActiveX 控件四部分。本章还对 ActiveX 控件进行了分类，希

望给读者一个整体的印象。

第 3 章是有关 Windows API 编程的技术。Windows API 是一个庞大的函数体系，要全部掌握所有的 API 函数不是一件容易的事情。本章“授人以渔”，领读者进门。

第 4 章是有关文件系统编程的技术，通过旧的文件 I/O 函数和最新的 FSO 对象模型比较，深入地介绍了文件系统及其操作技术。

第 5 章是有关数据库编程的技术。数据库编程并不难，Visual Basic 提供了 ADO、RDO 和 DAO 三种数据访问技术以及相当多的相关的 ActiveX 控件。本章详细地介绍了这些技术，并以“图书馆管理系统”为实例，透彻地讲述了数据库编程技术的方方面面。

第 6 章是有关网络编程的技术。网络编程也是当今最热门的技术之一，它和数据库编程结合在一起将“如虎添翼”。本章详细地介绍了如何利用 Visual Basic 来编制 DHTML 应用程序，使用包括 Winsock 等控件来编写网络应用程序。

附带光盘价值多多

本书附带的光盘上有两个文件夹。

名为 BOOK 文件夹下面是本书的电子文档。

名为 SOURCE 文件夹下面是本书的所有实例程序。

单击光盘根目录下面的 Install.exe 文件可以把它们安装到本地机器上。

为节省读者开支，本书所附光盘不随书发售，有需要者可来信 zfxin@ndip.com.cn 邮购，或在 <http://www.ndip.com.cn/computer/> 中下载。

使用本书的几个约定

本书将使用到下面的几个图标，它们所代表的意义分别是：



“注意”，它提醒读者要注意到某些方面。



“提示和技巧”，它给读者讲述某些技巧或者给读者某个提示。



“说明”，它提供一些说明文字。



“代码”，提示接下来的是程序的源代码。

第 1 章

使用固有控件编程



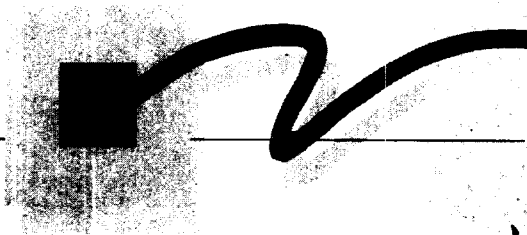
Visual Basic 的编程思想

Visual Basic 的固有控件编程

编程实例精解：打印预览程序

总结及启迪





本

章

导

读

本章是本书的起始章节，将给对 Visual Basic 已经有初步了解的读者架一座通往中高级编程的“桥梁”。

1.1 节介绍 Visual Basic.NET 的新特性，为读者展现 Visual Basic 特有的编程思想，从编程环境、语言风格以及独特的对象编程方面进行精心和系统的总结，开阅读者的思维。

1.2 节讲述 Visual Basic 编程的基本方法：使用固有控件编程。系统地介绍了各个控件最重要的属性、方法和事件，并展现了控件数组的巧妙用法。

1.3 节以打印预览程序作为例子，深入浅出地介绍如何使用固有控件编程，从而体现了 Visual Basic 的基本编程思想。

1.4 节则是总结及启迪。

1.1 Visual Basic 的编程思想

Microsoft Visual Basic 是在原有 BASIC 语言的基础上进一步发展起来的一整套的编程语言，或者称为编程系统。自 1991 年微软推出其 1.0 版本至今，已经发行至 6.0 版本。2000 年末微软推出了包括 Visual Basic.NET 在内的 Visual Studio.NET 的 Beta1 版本，可以预见，Visual Basic.NET 将会在今年正式推出。

对比起 Microsoft Visual C++ 编程，Visual Basic 的编程思路是截然不同的。Visual Basic 简单易学，它用十分巧妙的方法把 Windows 编程的复杂性“封装”起来，让用户使用它进行图形化编程，有如坐春风的感觉。由于 Visual Basic 的简单易学，不少人认为使用 Basic 编程的程序员不是真正的程序员。然而，事实并非如此！Visual Basic 入门容易，但精通它就不是一件容易的事情了。更何况，由于 Visual Basic 掩盖了 Windows 编程的复杂性，导致不少基于 Visual Basic 开发的程序员编程多年，仍未深知 Windows 系统的奥秘和神妙。其实，基于 Visual Basic 我们仍可以对 Windows 的底层进行操作，这将会在后面的章节中陆续介绍。

本书是针对对 Visual Basic 编程已经初步入门的用户而写的，所以本书将不会重点介绍 Visual Basic 的基本使用。本节是本书的起始章节，将对 Visual Basic 的基本的编程思想进行高度的概括，让读者温故而知新，并为下面的章节打下坚实的基础。

1.1.1 Visual Basic.NET 的新特性

2000 年初在旧金山举行的 Visual Basic 业界高峰会上，微软的史蒂夫巴尔默描述了 Visual Basic 的第 7 个版本，也就是 Visual Basic.NET 的新特性。从发布的 Visual Basic.NET Beta1 版本来看，Visual Basic 的革新是根本的。

从开发环境来看，Visual Studio.NET 将 Visual Basic、Visual C++、Visual C# 集成在同一个 IDE 开发环境里面，用户将在创建新的工程（Project）的时候选择开发语言。如图 1-1 所示。

从语言特性来看，Visual Basic.NET 真正实现了“面向对象编程”的所有特性，包括类的继承性，这将对以前 Visual Basic 版本包括 6.0 版本在代码层上实现突破，所以 Visual Basic.NET 将对 4.0 版本以前的 Visual Basic 不再具有兼容性。

面向对象编程要具备三个条件：封装性、继承性和多态性。Visual Basic 3.0 实现了多态性，而 Visual Basic 4.0 实现了封装性，Visual Basic.NET 将实现继承性。

另外，Visual Basic.NET 还实现了一些前所未有的新特性，包括自由线程、构造函数和析构函数等等。所有这一切的变革将会给 Visual Basic 的发展带来新的飞跃。

Visual Basic.NET 对 Visual Basic 6.0 将 100% 兼容，因此 Visual Basic 6.0 所编写的代码将可以升级到 Visual Basic.NET 版本。在 Visual Basic.NET 正式版本推出之前，使用 Visual Basic 6.0 仍是快速开发 Windows 应用程序的最好的编程工具之一。

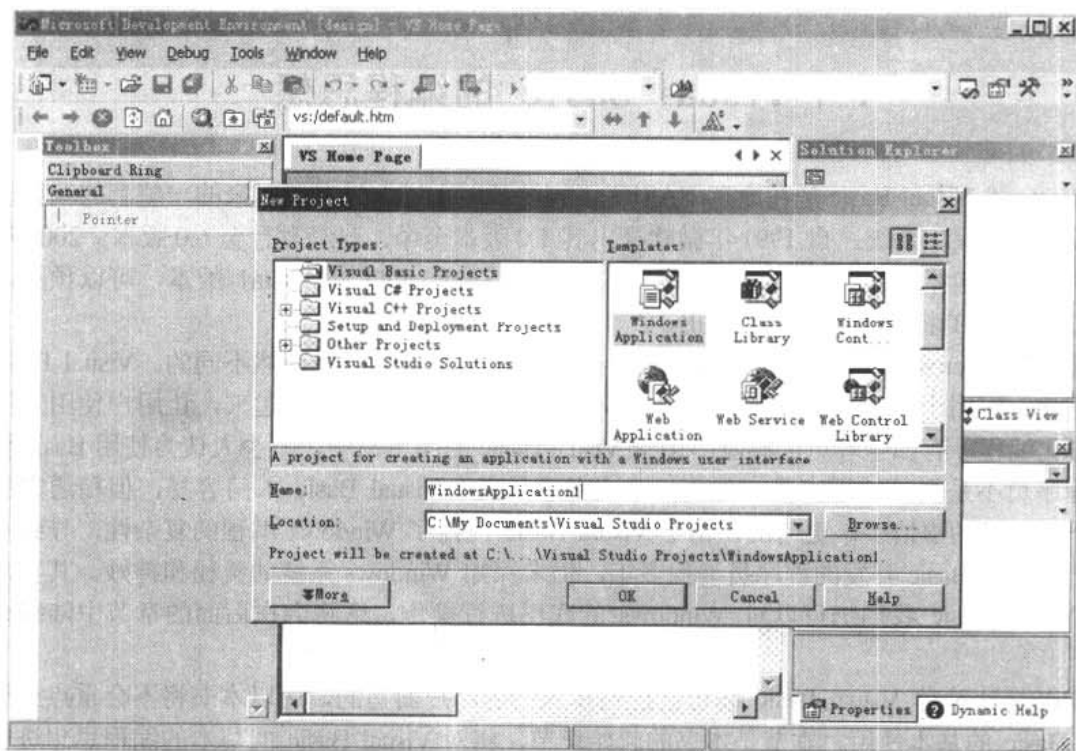


图 1-1 Visual Studio.NET 的开发环境

1.1.2 Visual Basic 的编程环境及语言风格

下面简单介绍 Visual Basic 6.0 的编程开发环境和 Basic 独特的语言风格。

1. 编程环境

Visual Basic 的编程开发环境是指集成开发环境 (IDE)，它在一个公共环境里集成了设计、编辑、编译和调试等功能。而每个功能都是独立运行的程序，有自己的界面。

Visual Basic 的集成开发环境见图 1-2。

此开发环境包括菜单条、上下文菜单、工具栏、工具箱、工程管理窗口、属性窗口、对象浏览器、窗体设计器、代码编辑器窗口、窗体布局窗口、立即窗口、本地窗口和监视窗口等。

在这里值得详细介绍的是对象浏览器和工具箱。

对象浏览器列出了你的 Visual Basic 工程中有效的对象，我们可以使用对象浏览器浏览或者查找 Visual Basic 中的对象及其应用程序，以及对象的方法和属性。

使用对象浏览器可以加快我们的开发速度。例如，我们建立一个新的工程，工程名称为“工程 1”。使用菜单【视图/对象浏览器】或者使用快捷键【F2】，就弹出对象浏览器。此时对象浏览器中罗列了<所有库>、stdole、VB、VBA、VBRUN、工程 1 包含的所有对象的信息：方法和属性。

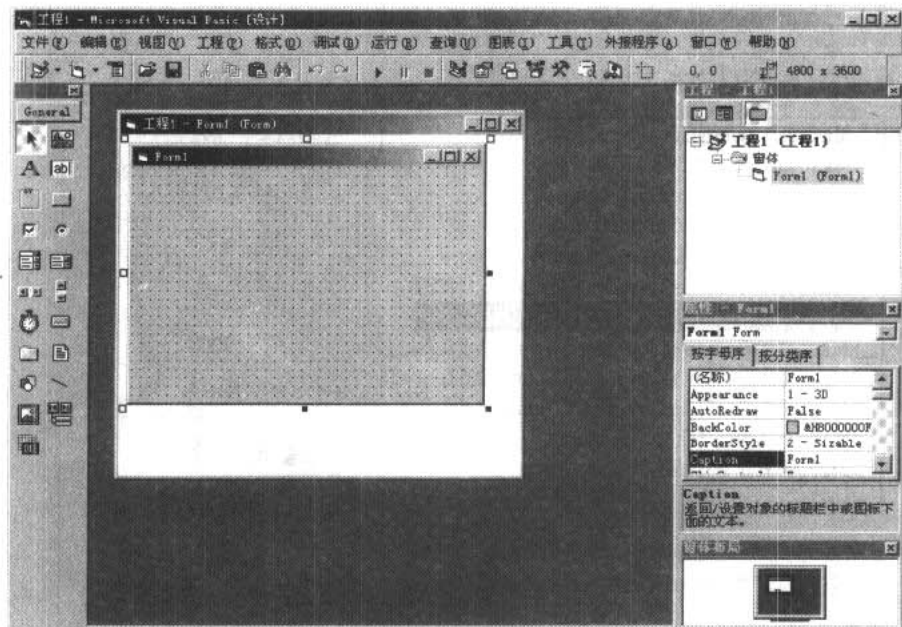


图 1-2 Visual Basic 6.0 集成开发环境



在对象浏览器中我们可以查找到 VB 所有自定义的常量和常数。

选择“工程 1”后，类窗口中有“<全局>”和“Form1”两项。选择“Form1”，则右边窗口显示出 Form1 的窗口类的属性和方法。例如，单击右边窗口中的 circle 方法后，窗口下面显示出相关的信息：

```
Sub Circle(Step As Integer, X As Single, Y As Single, Radius As Single,
Color As Long, Start As Single, End As Single, Aspect As Single)
```

这样，我们就可以迅速地找到某对象所需的相关简要信息。



使用对象浏览窗口可以把相关的代码过程粘贴进自己的应用程序里面，加快开发速度。

大家对工具箱都比较熟悉了。它默认提供了一组 Visual Basic 基本固有控件，如标签、文本框、单选框等。除此以外，我们还可以通过“添加选项卡”来添加控件以创建自定义布局。举个例子来作示范：

右键单击工具箱，选择“添加选项卡”，命名此选项卡为“自定义选项卡”。于是工具箱中增加了“自定义选项卡”一项。

工具箱中一般只放置了几个基本的固有控件，当我们编写复杂的 Windows 程序时，这些控件是远远不够的，我们需要其他功能更加强大的控件或者是 ActiveX 控件。右键单击“自定义选项卡”，选择【部件】，弹出如图 1-3 所示的“部件”窗口。

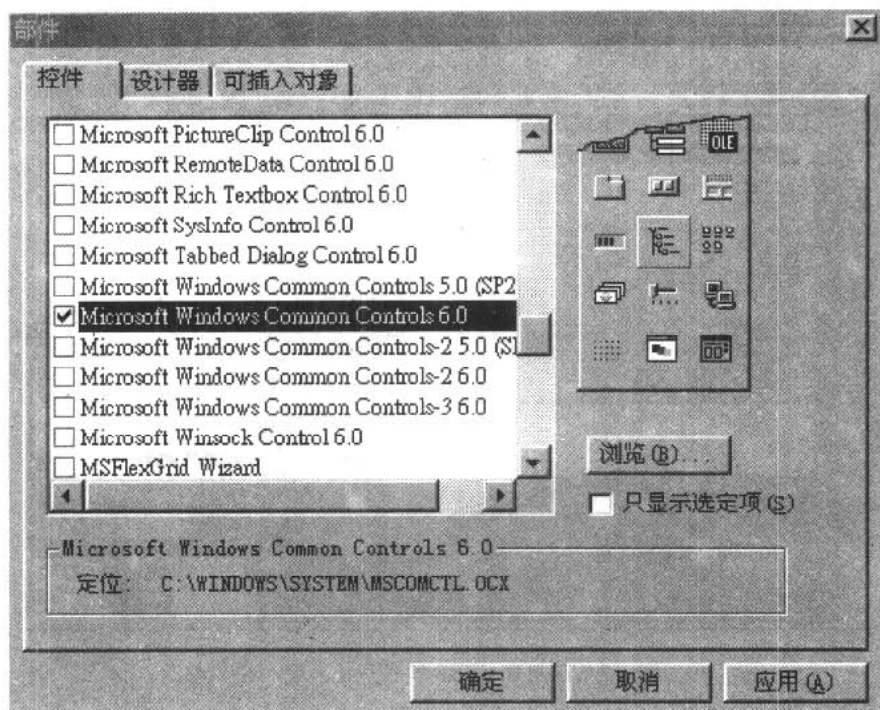


图 1-3 “部件”窗口

在这里我们可以选择“控件”、“设计器”、“可插入对象”。在“控件”中，我们选择我们需要的控件相应的.ocx 或.dll。例如，选择“Microsoft Windows Common Controls 6.0”一项，我们可以发现它“定位”于“c:/windows/system/mscomctl.ocx”中。

如果你的机器上没有 mscomctl.ocx，你就不能够使用微软提供的一系列通用工具条，如工具条控件 ToolBar，状态条控件 StatusBar，图像列表控件 ImageList 等等。单击“确定”后，则在“自定义选项卡”中显示出了这一系列的控件，具体如图 1-4 所示。你便可以在编程中随时使用它们了。

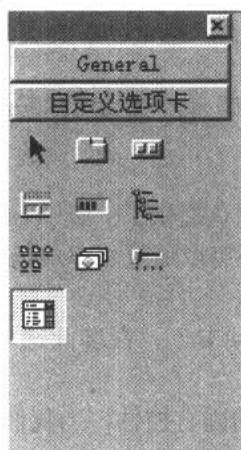


图 1-4 添加部件到工具箱

2. 语言风格

Visual Basic 的语言基础是 Basic，只是在 Basic 的基础上加上“可视化”编程的概念。Visual Basic 的语言代码存储在“模块”中。

我们可以把“模块”看做 Visual Basic 语言的集合单位，它是为实现某一功能的语言代码所形成的集合。在 Visual Basic 中，模块有三种类型：窗体模块、标准模块和类模块。

每个模块，无论是窗体模块、标准模块还是类模块，均包括“声明”和“过程”。

“声明”是对变量、常量或数据类型，甚至是动态链接库 DLL 的说明。

“过程”则是 Sub、Function 或者 Property 过程，它们包含可以作为单元来执行的代码片段。

3. 模块

◇ 窗体模块

窗体模块的文件扩展名为 .FRM，它是 Visual Basic 应用程序的基础。在窗体模块中声明的变量和常量均为窗体级声明，其生命力和作用范围只能在窗口模块内。在窗体模块中，包含对窗口事件响应的处理事件过程；还可以包括用户自定义的通过程。当然，我们可以在其他窗口模块中引用本窗口模块中的属性和方法。

◇ 标准模块

标准模块的文件扩展名为 .BAS，我们可以把它看做是公共的模块。它里面有变量、常数、类型、外部过程或全局过程的全局（在整个应用程序范围内有效的）声明或模块级声明。使用标准模块可以节省我们编程的时间和降低代码的繁琐程度。



在许多不同的应用程序中可以重用标准模块。

◇ 类模块

类模块的文件扩展名为 .CLS。类是面向对象编程的概念。我们可以在类模块中编写代码建立新的类对象，而这些新对象包含自己的属性和方法，供其他模块来调用。



类的概念在面向对象编程中尤为重要，实际上，在 Visual Basic 中可视化的窗体就是一个“类”的概念，只不过 Visual Basic 把它“掩盖”了。

4. Visual Basic 语言基础

◇ 变量、常量和数据类型

变量、常量和数据类型对于已经初步掌握 Visual Basic 编程的读者来说并不生疏，在这里不作详细介绍。但 Visual Basic 提供的一些特殊的语法对改善算法，简化编程有很大的帮

助，现逐一介绍，也算是抛砖引玉。

- 常量

前面已经提到，Visual Basic 提供了许多内部或系统定义的常量，供用户调用，我们可以在“对象浏览器”中查找相应的常量。用户也可以在应用程序中自定义常量，其格式是：

常量的定义

```
[Public|Private] Const constantname[As type] = expression
```

其中 Const 是关键字，constantname 为有效的符号名，type 为 Visual Basic 的数据类型，expression 由数值常数或字符串常数以及运算符组成，但不能是函数的调用形式。

例如：

常量的声明

```
Const conGoodDay=#7/9/00#
```

使用常量可以改进程序代码的可读性和可维护性。

- 动态数组

Visual Basic 在数组方面的功能比其它编程语言的强大就表现在动态数组上面。用户使用动态数组无需担心数组的上限，并且数组的上限可以用变量来限定！

5. 使用动态数组的步骤

① 声明数组，给数组附以一个空维数表，则将此数组声明为动态数组；

动态数组的声明

```
Dim Array()
```

② 在过程中使用 ReDim 来分配数组的实际元素的个数；

动态数组维数的分配

```
ReDim Array(1 to 100)
```

③ 每次执行 ReDim 的时候，先前存储在数组中的数据将全部丢失，Visual Basic 将重新给数组分配空间；如果要在数组原来的基础上扩展数组的上限，则要使用 Preserve 关键字，来扩大数组。

保留动态数组的内容

ReDim Preserve Array (UBound (Array) + 1)

'UBound() 函数得到数组的上界

◇ 数据类型之间的转换

Visual Basic 提供了强大的数据类型之间的转换函数集。使用这些函数可以把任一数据类型转换为指定的数据类型（当然要在有意义的转换间进行了）。

具体如表 1-1 所示。

表 1-1 数据类型的转换函数

函数名称	返回类型	参数范围
Cbool(expression)	Boolean	任何有效的字符串或数值表达式
Cbyte(expression)	Byte	0 ~ 255
Ccur(expression)	Currency	-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807
Cdate(expression)	Date	任何有效的日期表达式
Cdbl(expression)	Double	负数从 -1.79769313486232E308 ~ -4.94065645841247E-324; 正数从 4.94065645841247E-324 至 1.79769313486232E308
Cdec(expression)	Decimal	零变比数值, 即无小数位数值, 为 +/-79,228,162,514,264, 337,593,543,950,335。对于 28 位小数的数值, 范围则为 +/-7.9228162514264337593543950335; 最小的可能非零值 是 0.00000000000000000000000000000001
Cint(expression)	Integer	-32,768 ~ 32,767, 小数部分四舍五入
CLng(expression)	Long	-2,147,483,648 ~ 2,147,483,647, 小数部分四舍五入
CSng(expression)	Single	负数为 -3.402823E38 ~ -1.401298E-45; 正数为 1.401298E-45 ~ 3.402823E38
CStr(expression)	String	依据 expression 参数返回 Cstr
Cvar(expression)	Variant	若为数值, 则范围与 Double 相同; 若不为数值, 则范围与 String 相同

◇ 过程

我们可以认为, Visual Basic 的某一项工程为一大建筑物群, 则模块就是建筑物群里面的楼房, 而过程是楼房里面的单个的房间, 过程中的代码便是建造房间的砖瓦了。

Visual Basic 过程有三种类型: Sub 过程、Function 过程、Property 过程。

• Sub 过程

Sub 过程没有返回值。它的语法是:

Sub 过程

[Private|Public][Static]Sub 过程名称(参数)

语句

End Sub

我们可以把 Sub 过程放在窗体模块、标准模块、类模块中, 缺省声明为 Public。



当有不同的事件执行相同的动作时，我们可以把公共的代码放在同一个过程里面，并由响应事件的工程来调用它。

- Function 过程

Function 过程有返回值。它的语法是：

Function 过程

```
Private|Public|[Static]Function 过程名称(参数) [As type]
语句
End Function
```

Visual Basic 有内置的函数，例如 Sin()、Int()等，而用户也可以编写自己的函数。在编写函数的过程中，只要给“过程名称”赋上一个值，就可以返回这个值了。

- Property 过程

Property 过程返回或指定值，并设置对象的引用。在 Visual Basic 中有三种 Property 过程。

Property Get	得到返回值。
Property Let	设置返回值。
Property Set	设置对象的引用。

这个过程用于对对象编程中。当我们使用 Set 语法来给一变量赋对象变量时，Visual Basic 调用 Property Set 语句，如果没有则调用 Property Let 语句。

语言控制结构

Visual Basic 和其他编程语言一样，同样有两种语言控制结构：判断结构、循环结构。

- 判断结构

Visual Basic 在判断结构上有多种选择。
其语法为：

判断结构

```
IF 判断条件 Then 语句
End IF
或
IF 判断条件 Then
    语句
End IF
或
```

```
IF 判断条件 Then
    语句
Else[ElseIf 判断条件 1 Then]
    语句
End IF
```

或

```
Select Case 判断对象
Case 判断条件 1
    语句 1
Case 判断条件 2
    语句 2
.....
Case Else
    语句 N
End Select
```



当发觉使用 ElseIf 多的时候，把它转换为 Select Case 编写比较好。

- 循环结构

Visual Basic 的循环结构有三种。

其语法为：

循环结构

```
Do While 循环条件
    语句
Loop
或
Do
    语句
Loop While 循环条件
或
For 计数器=初始值 To 结束值[ Step 步长]
    语句
Next
```



在循环次数已知的情况下，使用 For 语句较好。

1.1.3 编程思想的独到之处

Visual Basic 之所以简单易用，是与它对面向对象编程的独到“理解”分不开的。

面向对象编程的难点在于对“对象”一词的理解。Visual Basic 使用图形化的编辑界面，以及对对象的“封装”，把编程变成一件简单的事情。然而，要真正掌握 Visual Basic 编程的奥妙，则要深刻理解对象编程，并且会使用对象编程。

1. 对象的概念

在 Visual Basic 中，对象是编程代码和数据的集合，可以是控件或是窗体。它拥有自己的属性和方法，以及事件。

对象是用类来定义的，我们通过声明建立类的相关实例。

对象的属性用来保存数据和设置，改变属性便可以改变属性的特性。例如，对于窗体 Form 来说，其属性 Caption 标志这个窗体的标题。

对象的方法是指对对象的操作过程。同样对窗体 Form 来说，Show 方法是使窗体显示出来这个过程。

对象的事件是要响应对象所能识别的事件消息。对窗体 Form 来说，Click 事件是指鼠标在窗体单击时所引发的响应过程。

值得大家注意的是，对象集合的概念和对象数组的概念的区别。

对象数组是指以 Name 相同的同一类对象为元素的数组，它们具有相同的属性、方法和事件。

对象集合是指对象集合在一集合对象中，作为集合对象的成员。集合对象有自己的属性和方法。集合对象的成员以“索引号”为标志。例如在窗体对象里面的所有控件组成一控件集合。



如果对集合中的所有对象都必须执行相同操作，则可用集合简化代码。

2. 对象的创建

Visual Basic 对对象的创建也是唾手可得的。新建一个工程时，系统将默认地新建一个窗体对象。而你只要选中工具箱里的控件对象，就可以在窗体中建立控件对象实例了。

其实，我们也可以用编程的方法来创建对象。注意，我们使用的都是对象的实例。

首先我们先声明对象变量，然后再把对象赋予此变量。

对象的创建

' 声明对象变量