卓越文化 UNIQUE ZHUOYUE WENHUA

主编 熊松明

Visual Basic.NET 标准教程

本书内容

- ☆ VB.NET 初步 / 语法和编程规范
- ☆ VB.NET 面向对象的实现
- ☆ 用 VB.NET 打造NameSpace
- ☆ WinForm 的应用程序 / 数据绑定
- ☆ VB.NET 的数据库基础编程
- ☆ 类和封装性 / 继承
- ☆ 程序多重定义和母子对象关系
- ☆ Me 参考值 / 重载程序和共享成员
- ☆ Prototype 样式 / Whole-Part 关系
- ☆ 程序详解和聊天室范例

计算机教育图书研究室 Computer Education Books 总策划

航空工业出版社

Visual Basic.NET 标准教程

计算机教育图书研究室
Computer Education Books 总策划

主　编　熊松明

编　委　刘　春　王　宁

于晓利　魏　霞

航空工业出版社

内 容 简 介

程序开发人员要快速创建企业级的 Web 应用程序，必须依赖于可伸缩性、强壮性和可重用性等商业逻辑。在过去的几年中，面向对象的程序设计业已成为符合这些要求的典范，使用面向对象的编程语言有助于使大规模的系统更易于理解、更易于调试、更迅速地升级。

本书从 VB.NET 的变化，以及安装 VS.NET，VB.NET 的语法特点与其他平台的比较说起，通过典型实例向读者阐述了 VB.NET 的特性和功能。同时，对 VB 6.0 和 VB.NET 的编程思维和方法进行了比较。本书面向的是对 Visual Basic 有基本了解的用户，通过对本书的学习得以快速地提高。

图书在版编目（CIP）数据

Visual Basic.NET 标准教程 / 熊松明主编. －北京：航空工业出版社，2002.12
ISBN 7-80183-072-5

Ⅰ.V… Ⅱ.熊… Ⅲ.BASIC 语言－程序设计－教材
Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2002）第 082050 号

航空工业出版社出版发行
(北京市安定门外小关东里 14 号 100029)
北京云浩印刷厂印刷　　全国各地新华书店经售
2002 年 12 月第 1 版　　2002 年 12 月第 1 次印刷
开本：787×1092　1/16　　印张：17　　字数：306 千字
印数：1-6000　　定价：22.00 元

本社图书如有缺页、倒页、脱页、残页等情况，请与本社发行部联系调换。联系电话：010-65934239 或 64941995

前　言

在传统的结构化程序设计中存在着一些弊端：所有的代码被书写成结构化的形式，而不是模块。数据元素可以在任何代码中被访问，也可以在开发人员不知道的情况下被修改，这样就增加了在调试过程中发现运行错误的难度。而且，程序的维护也可能随之成为一项繁重的任务，因为理解结构化编程中修改一行代码所造成的全局影响是非常困难的。此外，依赖于开发人员控制代码和数据结果会导致较低的可重用性。

面向对象的程序设计（OOP）解决了这些问题，它将数据和在其上实施的方法包装成一个独立的单元，叫做对象。一个对象的数据可以隐藏，以防止未经授权被他人修改，并且公开一组可以在数据上进行操作的公共方法，这叫做封装。由于实现细节和接口相分离，底层的编程逻辑可以在后期改变，因而不必担心破坏调用对象的代码。

OOP 还允许开发人员通过继承的同时重用代码和数据。通过从先前确定的对象中继承，开发人员可以更迅速地构造复杂的应用程序。由于编写新的代码总是会有带入错误的潜在可能，重用经过测试的代码可以使产生额外错误的可能性降到最小。

为了适应更多的需要，Visual Basic.NET 新增了一些语言特性，这些特性可以使其拥有以上所描述的种种优点，使其成为一流的面向对象的编程语言。

为了使 Visual Basic 开发人员能够从面向对象设计中受益，并简化企业级 Web 应用程序的开发，Visual Basic 的新版本——Visual Basic.NET 支持包括实现继承在内的全部面向对象的语言特性。具备了这些语言特性，Visual Basic.NET 将具有快速开发企业级关键应用程序所需要的所有能力，同时保持了使其成为世界上最流行的开发工具的直接访问性。

Visual Basic.NET 提供了一流的面向对象的程序设计语言特性，诸如实现继承、重载和参数化的构造器。此外，开发人员还可以通过显式的自由线程编写具有高度可伸缩性的代码，同时通过其他现代化的语言特征（如结构化的异常处理等）编写具有较高可维护性的代码。Visual Basic.NET 提供了开发人员创建强壮、可伸缩的分布式 Web 应用程序所需要的所有语言特性。

本书由计算机教育图书研究室总策划，熊松明主编。由于作者水平有限，书中错误之处在所难免，希望广大读者批评指正。

http://www.china-ebooks.com

编　者

2002 年 10 月

目 录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　VB.NET 初步

Visual Basic 7.0 也被称为 Visual Basic.NET（VB.NET），它具备了面向对象（OOP）编程语言的所有特征。VB 程序员对面向对象的概念和面向对象编程方式都不陌生。如果问什么是面向对象程序设计，他可能会说出一大堆诸如类、接口、消息隐匿、封装、继承、多态性这样的名词。但面向对象编程并非是通过一两天的学习或听一次课就能掌握的，要真正地掌握面向对象程序设计，不但需要掌握一定的理论知识，同时还需要进行一些实际的编程练习。本章主要介绍 VB.NET 面向对象编程的优点和特性，以及它的集成开发环境。

1.1　VB.NET 的优点和特性

Visual Basic.NET 提供了一流的面向对象的程序设计语言特性，诸如实现继承、重载和参数的构造器等。此外，开发人员可以通过显式的自由线程编写具有高度可伸缩性的代码，同时通过其他现代化的语言特征（如结构化的异常处理等）编写具有较高可维护性的代码。VB.NET 将为开发人员提供创建强健、可伸缩的分布式 Web 应用程序所需要的所有语言特性。

1.1.1　VB.NET 面向对象编程的优点

不知道用户是否考虑过为什么现代程序设计语言会向面向对象编程靠拢？C++、JAVA 为什么这么普及？这是因为面向对象编程具备了许多优点，比如，代码维护方便、可扩展性好、支持代码重用技术等。这些优点是过程编程语言所不具备的。下面就来列举面向对象技术的这些优点。

1. 维护简单

模块化是面向对象编程中的一个特征。实体被表示为类和同一名称空间中具有相同功能的类，程序开发人员可以在名称空间中添加一个类而不会影响该名称空间的其他成员。

2. 可扩充性

面向对象编程从本质上支持扩充性。如果有一个具有某种功能的类，就可以很快地扩充这个类，创建一个具有扩充功能的类。

3. 代码重用

由于功能是被封装在类中的，并且类是作为一个独立实体而存在的，所以提供一个类库就非常简单了。事实上，任何一个.NET Framework 编程语言的程序员都可以使用.NET Framework 类库，.NET Framework 类库提供了很多的功能。更令人高兴的是，程序开发人员可以通过提供符合需求的类来扩充这些功能。

1.1.2　从 VB 6 到 VB.NET 的变化

.NET 平台为所支持的语言提供了公共类型系统，这就意味着所有语言都必须支持公共

语言运行时间环境所强制的相同的数据类型，如此就消除了不同语言之间数据类型的不兼容性。例如，同在一个 32 位的 Windows 平台上，在 C++语言中，integer 数据类型占据四个字节，而在 VB 中则占据两个字节。以下是 VB.NET 中与数据类型有关的变化：

- 在.NET 下，VB.NET 中的 integer 数据类型也是四个字节。
- VB.NET 没有 currency 数据类型，作为代替，它提供了 decimal。
- VB.NET 引入了一种新的数据类型，叫做 char。char 数据类型占据两个字节，可以存储 Unicode 字符。
- VB.NET 没有 variant 数据类型。为了得到与 variant 类型相同的结果，读者可以使用 object 数据类型。因为在.NET 中的一切，包括原始数据类型，都是一个对象，一个 object 类型的变量，它可以指向任何数据类型。
- 在 VB.NET 中没有固定长度字符串的概念。
- 在 VB 6 中，使用 Type 关键字来声明用户自定义结构。VB.NET 引入了与此目的相同的结构关键字，语法的其他部分是相同的。

```
Structure MyStruct1
  ...
End Structure
```

1. 变量声明

下面看一个简单的 VB 6 例子。

```
Dim x,y as integer
```

在这个例子中，VB 6 将 x 看成 variant，把 y 看成 integer。VB.NET 改进了这个解释，它将 x 和 y 都创建成 integer。另外，VB.NET 还允许读者在声明语句中为变量赋初始值，如下：

```
Dim str1 as string="hello"
```

VB.NET 还引入了 Read-Only（只读）变量。只读变量不同于常数，声明它的时候不用初始化，但是一旦给它赋值，就不能再修改。请看下面的例子：

```
'未初始化
Dim readonly x as integer
'附值
x=100
'不能再被修改，因此下面的语句将出错
x=200
```

2. 数组使用

在 VB 6 中，读者可以通过编程来定义数组的上限和下限。而在 VB.NET 中，数组的下限总是 0，定义数组时是这样的：

```
Dim a(0 To 50) as integer
```

这样实际创建的元素是 51 个，其中 0 是下限，50 是上限。

3. 变量范围

考虑以下 VB 6 代码：

```
If x=y then
  Dim z as integer
```

此为试读，需要完整PDF请访问：www.ertongbook.com

```
    ' 其他程序代码
End If
z=100
```

以上代码在 VB 6 中运行得非常好，因为它没有模块级变量作用范围。模块级变量应用在其他高级编程语言中，如 C++。在声明模块中定义的变量，如在 If ... Then 模块中定义的变量，当这个声明模块结束时就会落在范围之外。这样一来，如果在其定义 If ... Then 模块之外存取 z，在其他高级编程语言中就会导致一个错误。而 VB.NET 与 VB 6 相反，它使用了模块级的变量范围。

4. Set 和 Let 声明

在 VB 6 中，用户必须使用 Set 声明为变量分配一个对象示例。出于默认属性的原因，这在 VB 6 中是必须的。要想给对象本身指定一个变量（与对象的默认属性值相反），就必须要使用 Set 关键字。例如，在 VB 6 中，使用 Set 声明为变量分配对象的代码如下：

```
Dim x as Variant
Dim strName as string

StrName=Text1

Set x=Text1
```

但是，在 VB.NET 中是不允许默认属性的（除非是参数化了的属性），因此就不需要使用 Set 关键字。同样，Let 关键字也从 VB.NET 的语法中去掉了。

5. 错误处理

Visual Basic 最终结合了结构化错误处理。关键字 Try、Catch 和 Finally 使错误处理变得简单，也使 VB.NET 与 C++或 C#这些语言可以并列起来。Try ... Catch 的模式允许开发人员将可能导致例外的代码放置在一个 Try 模块中。如果那个代码确实造成了一个例外（与造成错误同义），就执行 Catch 模块中的代码；这个模块中的代码应该是用来处理例外的。

请注意，为了向后兼容，VB 6 中旧的错误处理技术(Error Resume Next 及类似的)还是受到支持的，但是当用 VB.NET 编写新的应用程序时应该争取不使用这些旧的技术。下面的这些代码片段说明了 VB.NET 的各种错误处理技术，代码如下：

```
Try
  ...
Catch
  ...
End Try
```

上面的代码只是捕捉相关的 Try 模块中冲突代码所导致的例外。VB.NET 允许用户使用多个 Catch 模块来处理特殊的例外，代码如下：

```
Try
  ...
Catch e1 as NullPointerException
  ...
Catch e2 as Exception
  ...
End Try
```

除了捕捉预先定义的例外，用户还可以创建自己定义的例外类，这个类是从System.Exception 基础类中继承而来的。读者还可以激活自定义例外，这与 VB 6 中 Err 对象的 Raise 方法相同，代码如下：

```
If myvar < 1000 then
  Throw new Exception("Business Logic Error")
End If
```

6. 静态方法

VB.NET 允许开发人员在类中创建静态方法。静态方法指的是那些不需要开发人员创建类的例示就能调用的方法。例如，如果有一个名为 Foo 的类，其中有一个非静态方法 NonStatic()和一个静态方法 Static()，就可以这样来调用 Static()方法，代码如下：

```
Foo.Static()
```

但是，非静态方法则需要创建类的一个例示，代码如下：

```
'Create an instance of the Foo class
Dim objFoo as New Foo()
'Execute the NonStatic() method
objFoo.NonStatic()
```

要想在一个 VB.NET 类中创建一个静态方法，只需要给方法定义加上前缀 Shared 关键字即可。

7. 程序和函数

在默认状态下，VB 6 中所有的程序参数都是通过引用方式（ByRef）来传递的，而在 VB.NET 的默认状态下，这些参数是用值方式（ByVal）来传递的。不管程序和函数是否接收参数，当调用它们的时候都需要圆括号。VB 6 中，函数返回值使用这样的语法：FunctionName = return_value。在 VB.NET 中，读者可以使用 Return 关键字以 Return return_value 方式来返回值，也可以继续使用旧的语法，它们仍然有效。

8. 属性语法

在 VB 6 中，使用 Property Get 和 Property Set/Let 在类中创建属性，它们分别出现在单独的程序中，代码如下：

```
Public Property Get PropertyName() as DataType
  ...
End PropertyPublic Property Let PropertyName(value as DataType)
  ...
End Property
```

在 VB.NET 中，Property Get 和 Property Let/Set 被合并成一个 Property 语句，而不再是两个单独的语句。另外，在 Property 语句的 Set 部分中，变量 Value 指的是用户给指定属性赋值时输入的那个值，代码如下：

```
Public (ReadOnly | WriteOnly) Property PropertyName as DataType
  Get
    Return m_var
  End Get

  Set
```

```
    M_var = Value
  End Set
End Property
```

VB.NET 的语义和语法都发生了一些变化，这里所列举的都是作为一个 ASP.NET 开发人员将要遇到的最重要的变化。当读者用 VB.NET 创建 ASP.NET Web 页面时，需要牢记的最重要的几点是：

- 变量可以被分类，不再都是 Variants。也就是说，如果读者需要一个 Integer 变量，就应该使用 Dim i as Integer，而不仅仅是用 Dim i。分类的变量比无分类的变量的性能要好得多。
- VB.NET 要求子程序调用时在调用参数周围加上圆括号。这就是说 Response. Write "Hello, World!" 会生成错误，正确的应该是 Response.Write("Hello, World!")。
- VB.NET 不再支持默认属性，用户必须对从一个对象中存取的属性进行明确地说明。
- 声明数组时要注意，VB.NET 中所有数组的下限都是 0，上限则是由读者规定的数字。

1.1.3　VB.NET 面向对象编程的新特性

在传统的结构化程序设计中存在着一些弊端：所有的代码被书写成结构化的格式，而不是模块。因为数据元素可以在任何代码中被访问，它可能在开发人员不知道的情况下被修改，这样会增加在调试过程中发现运行错误的难度。而且，程序维护也可能随之成为一项繁重的任务，因为理解结构化编程中修改一行代码所造成的全局影响是非常困难的。此外，依赖于开发人员控制代码和数据结果会导致较低的可重用性。

面向对象的程序设计（OOP）解决了这些问题，它将数据和在其上实施的方法包装成一个独立的单元，叫做对象。一个对象的数据可以隐藏，以防止未经授权被他人修改，并且公开一组可以在数据上进行操作的公共方法，这种概念叫做封装。由于实现细节和接口相分离，底层的编程逻辑可以在后期改变，而不必担心破坏调用对象的代码。

OOP 还允许开发人员通过继承的同时重用代码和数据。通过从先前确定的对象中继承，开发人员可以更迅速地构造复杂的应用程序。由于编写新的代码总存在带入错误的潜在可能，重用经过测试的代码可以使产生额外错误的可能性达到最小。

为了适应这些需要，Visual Basic.NET 将提供一些新增的语言特性，这些特性可以使其拥有以上所描述的种种优点，使其成为一流的面向对象的编程语言。

下面就来阐述 VB.NET 面向对象编程的一些新增特性。

1. 继承

一直以来，对 Visual Basic 特性的要求中呼声最高的就是对继承的支持。在 Internet 时代的程序开发，要求快速地组装和大量的重用。Visual Basic.NET 完全实现继承，包括可视化窗体的继承。开发人员可以使用新的关键字 Inherits 从一个已存在的类中派生，代码如下：

```
Class1
Function GetOrders()
...
End Function
```

```
Class2
Inherits Class1
Function GetDetails()
...
End Function
```

继承语句支持所有与继承相关的特性。派生类的实例支持所有基类所支持的方法和接口。当然，派生类可以扩展基类所支持的方法和接口的集合。派生类可以使用 Overrides 关键字来替代基类中定义的方法。为减少编程错误，Visual Basic 会防止意外地替代一个函数。只有声明为“可替代”的函数允许在派生类中被替代。

2. 重载

Visual Basic.NET 允许函数重载，这使开发人员具有建立一个拥有相同名称，但参数类型不同的过程或函数的能力。

当对象模型中规定了要使用名称相近但操作于不同类型的数据之上的过程时，继承尤其有用。

例如，一个可能表现为几种不同数据类型的类可以有这样一个 Display 过程：

```
Overloads Sub Display (theChar As Char)
...
Overloads Sub Display (theInteger As Integer)
...
Overloads Sub Display (theDouble As Double)
```

如果没有继承，程序员就需要为每个过程使用不同的名称或 Variant 参数。重载提供了一种更为清晰、有效的方法来处理多种数据类型。

3. 参数化的构造器

参数化的构造器（简称为构造器）允许创建一个类的新实例的同时，向这个实例传递参数。构造器对于面向对象的编程来说是必需的，因为它允许用户定义的构造代码通过实例的创建者传递参数，它通过允许一个新的对象实例在一个单独的表达式中创建并初始化，以简化客户机程序的代码。

1.1.4 VB.NET 其他的现代化语言特性

Visual Basic.NET 增加了一些新特征，用于简化具有强壮性和可伸缩性应用程序的开发。这些特性包括：自由线程、结构化的异常处理、严格的类型安全以及诸如初始化设置和共享成员等。

1. 自由线程

现在当开发人员在 Visual Basic 中创建应用程序时，他们所编写的代码是同步的，这意味着每行代码要在下一行代码之前执行。开发 Web 应用程序的时候，可伸缩性是关键。开发人员需要的是实现并行处理的工具。

通过自由线程，开发人员可以在生成一个线程来完成一些运行时间较长的任务、执行一个复杂的查询或运行一个多部分的计算的同时，使得应用程序的其他部分继续执行，从而实现异步处理。代码如下：

```
Sub CreateMyThread()
Dim b As BackGroundWork
Dim t As Thread
Set b = New BackGroundWork()
Set t = New Thread(New ThreadStart(AddressOf b.Doit))
t.Start
End Sub

Class BackGroundWork
Sub DoIt()
...
End Sub
End Class
```

2. 结构化的异常处理

开发企业级的应用程序要求创建可重用的、可维护的部件。在过去的 Visual Basic 版本中，Basic 语言的一个具有争议的方面是其对错误处理的支持。利用现存的 On Error Goto 语句的错误处理方法，有时会减缓大规模应用程序的开发和维护。其名称就反映出这样一些问题：就像 Goto 所意味的，当一个错误发生时，控制权转移到子程序中一个有标记的位置。一旦错误代码运行，它必须时常通过另外的清除位置来转向，而后者又要经过一个标准的 Goto，最后还是要通过其他的 Goto 或 Exit 来退出过程。使用 Resume 和 Next 的多种组合来处理几个不同的错误将会产生难以读懂的代码，并且在执行路径没有被完全考虑到的时候会导致频繁的错误。

而利用 Try...Catch...Finally，这些问题将不复存在，开发人员可以嵌套其异常处理，同时这是一种用于编写在正常条件和异常条件下执行清洁代码的控制结构。代码如下：

```
Sub SEH()
Try
Open "NEWFILE" For Output As #1
Write #1, CustomerInformation
Catch
Kill "NEWFILE"
Finally
Close #1
End try
End Sub
```

3. 严格的类型检查

当前的 Visual Basic 语言在其可能产生隐式的类型强制转换上是非常自由的。对于赋值和除了引用方式之外的参数传递，Visual Basic 编译器允许几乎任何一种数据类型通过运行时的强制类型转换向其他数据类型转换。如果要转换的值不能在没有数据损失的情况下被转换，那么运行时的强制转换可能会失败。通过增加一个新的编译选项，Visual Basic.NET 可以对任何可能在运行时发生错误的转换产生编译时错误。选项 Strict 通过要求一个可能在运行时失败的转换，或诸如在数字类型和字符串之间的在用户预期之外的自动转换产生

错误时来改善类型安全。

4. 共享成员

共享成员是指由类的所有实例所共享的数据和函数成员。在类的所有实例中，共享一个数据成员或函数的单个实例是使用继承的 Visual Basic 应用程序所需要的。一个共享数据成员独立地存在于类的每个实例中。共享方法不同于普通的方法，它并不是隐式的传递类的一个实例。由于这个原因，在共享方法中对非共享数据成员的无限制引用是不允许的。共有的共享成员可以被间接地访问，而且它们可以从类的实例后期绑定。

5. 初始化设置

Visual Basic.NET 支持在变量的声明行中对其初始化。初始化设置可以在包括控制结构的任何地方使用。含有初始化设置的过程级声明的语义与一个声明语句后紧跟一个赋值语句是相同的。即：Dim X As Integer=1 与 Dim X As Integer X=1 是相同的。

Visual Basic.NET 现在是一流的面向对象编程语言，使用 Visual Basic.NET，开发人员可以通过显式的自由线程创建高度可伸缩的代码。代码中增加了如结构化异常处理等现代化语言特性，具有很高的可维护性。Visual Basic 将为开发人员创建强壮的、可伸缩的分布式 Web 应用提供所需要的一切语言特性。

1.2 VB.NET 的集成开发环境

1.2.1 安装 VS.NET

下面以 Windows 2000 操作系统为例，介绍 VS.NET 的安装。首先准备好所需的安装光盘，将第一张光盘插入光驱，弹出如图 1.1 所示的窗口。

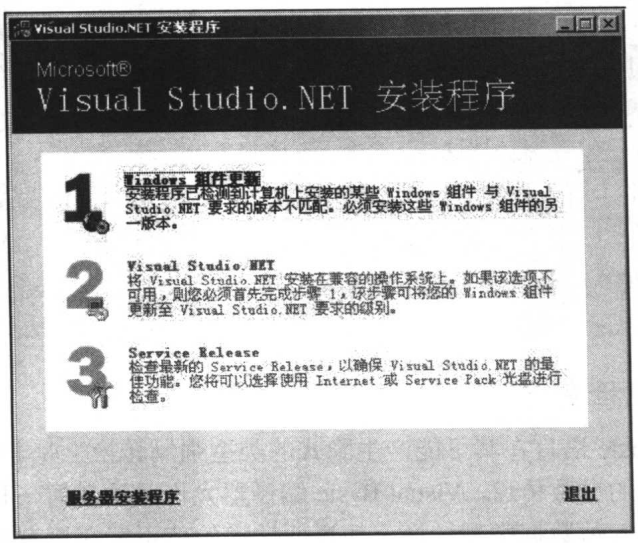

图 1.1 安装步骤（1）

如果没有安装 FrontPage 2000 服务扩展，将会出现如图 1.2 所示的对话框。也可以以后再添加，单击“继续”按钮。

此为试读，需要完整PDF请访问：www.ertongbook.com