开发人员专业技术丛书

Visual Basic .NET 技术与技巧

Visual Basic .NET Tips & Techniques

(美) Kris Jamsa 著

陈维军 王慧英 等译

机械工业出版社
China Machine Press

开发人员专业技术丛书

Visual Basic .NET 技术与技巧

（美）Kris Jamsa 著

陈维军 王慧英 等译

机械工业出版社
China Machine Press

本书是一本指导你如何最大程度地使用Visual Basic.NET开发的权威性参考书。书中详细介绍了关键的编程概念和利用.NET环境来开发应用程序的基础知识，本书还提供了几百个如何在.NET环境下使用Visual Basic.NET功能的技巧、具有实践性的建议以及数百个可以立即运行的重要解决方案的详细源代码，内容主要涉及以下几个方面：ASP.NET页面、ADO.NET数据库应用、Web服务、Web和Windows窗体、事件和错误处理程序，以及使用.NET程序集和版本化进行应用程序部署等等。通过本书的学习，你可以脱离使用鼠标拖放来实现应用程序的阶段，能够理解关键操作的内部实现机理，有助于快速利用Visual Basic.NET来实现大量的编程任务。

本书适用于Visual Basic.NET程序员及爱好者。

本书版权登记号：图字：01-2002-4798

图书在版编目（CIP）数据

Visual Basic . NET技术与技巧 / (美) 贾马沙 (Jamsa, K.)著；陈维军等译. – 北京：机械工业出版社，2003.2

(开发人员专业技术丛书)

书名原文：Visual Basic. NET Tips & Techniques

ISBN 7-111-11525-2

Ⅰ. V… Ⅱ. ① 贾… ② 陈… Ⅲ. Basic语言－程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2003）第001651号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑：周 睿

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年3月第1版第1次印刷

787mm × 1092mm 1/16 · 33.25印张

印数：0 001- 4 000册

定价：55.00元

前　言

阅读本书的理由

近几年来，Visual Basic已成为使用最广泛的编程语言，而Visual Basic .NET通过引进结构化错误处理，支持多线程执行，快速建立和使用Web服务的功能，建立新的数据库交互模式（ADO .NET）等等，又大大扩展了原有的编程能力。在这之前，要完成这些操作，程序员们不得不使用诸如C++或Java这样的语言。

过去，许多程序员都会抱怨说，Visual Basic并不适用于开发专业级的应用程序，因为性能上不能满足要求。随着Visual Basic .NET的发布，这种性能方面的问题就迎刃而解了。在 .NET环境下，无论是用Visual Basic .NET、C#，还是Visual C++ .NET编写的程序，都可以利用公用语言运行时库（Common Language Runtime，CLR）和基类库（Base Class Library，BCL）提供的相同方法和类来实现特定的操作。由于每一种编程语言都使用相同的库（这是因为Visual Basic编译器得到了改进），因此，不管是用哪一种编程语言建立应用程序，.NET程序都能实现相似的功能。

随着Visual Basic .NET技术的进步和发展，它必将成为今后几年中最佳的编程语言。

本书分析了几百个如何在 .NET环境下使用各种Visual Basic .NET功能和特性的技巧。每一个技巧都提供了现成的源代码，你可以用来试验某个编程概念，也可以把它们剪贴到自己的程序中。此外，在介绍每个技巧时还提供了对代码执行过程中每个步骤的解释。通过阅读本书，你将学习到以下几个方面的知识：

- 学会使用Visual Basic .NET的面向对象编程功能，如类、继承、接口和反射等。
- 学习如何在程序中应用公用语言运行时库和基类库中几千种独立于编程语言的方法和类，以完成特定的任务。
- 学会在程序中使用“通用对话框”显示标准化的对话框，用于打开、保存、打印文件和选择字体、颜色等操作。
- 了解Visual Basic .NET是如何支持多线程功能的，这一功能可使用户以为你的程序能同时执行多项任务。
- 领略 .NET环境如何通过使用Windows窗体，使得基于窗体的用户界面适用于所有编程语言。
- 用Try-Catch语句检测异常，并对异常做出响应（这就是程序员们所指的结构化错误处理）。
- 用Visual Basic .NET或C#等编译语言生成ASP.NET Web页。
- 用Visual Studio创建和使用Web服务（程序可以远程调用这些函数或子例程，以完成特定的任务）。
- 了解 .NET环境是如何用ADO .NET取代ADO模式进行数据库访问的。

- 学习Visual Basic .NET中的许多操作，以便迅速提高程序的性能和功能。
- 其他更多内容！

本书包括哪些内容

本书共18章，每章都详细讨论了Visual Basic .NET中某个方面的知识，或者 .NET环境提供的某个重要功能。在每章中，首先介绍主要的基本概念，然后介绍其应用，最后介绍如何运用这些后台操作对程序进行性能优化和功能的扩充。各章都提供了大量现成的程序，你可以从Osborne公司的网站（www.Osborne.com）下载所有程序的源代码。

第1章“Visual Basic .NET基础知识” 过去的十年中，Visual Basic逐渐成为了绝大多数程序员所选择的编程语言。如果你没有Visual Basic编程经验，第1章将为你介绍主要的数据类型、运算符和程序结构（如If和While语句）等，这些都是创建Visual Basic程序必备的知识。此外还将介绍如何创建基于控制台的程序，把输出显示到类似于MS-DOS的窗口上，以及如何创建基于Windows的程序，用窗体来提供用户界面。如果你有Visual Basic编程经验，那么，第1章中提供的技巧涵盖了老的Visual Basic和Visual Basic .NET之间的主要区别。在学完第1章的基础上，你就具有构建Visual Basic .NET程序的专业知识了。

第2章“利用 .NET环境” 如果你跟许多程序员一样，刚刚接触Visual Basic .NET，那么，你不仅要学习一门新的编程语言，而且还需要学习 .NET环境的许多新特性，以及如何在你的程序中运用这些新特性。第2章就是 .NET技术的“速成教材”，你将从中学习公用语言运行时库和通用类型系统，学习用Visual Basic .NET和C#创建ASP.NET Web页，用Web窗体为ASP .NET页构造用户界面，了解 .NET环境是如何用元数据（关于数据的数据）来实现程序对对象的查询。然后，你将学习 .NET环境是如何使用可执行程序和类库的中间语言（IL）代码的。最后，介绍如何对Visual Basic .NET程序采用结构化错误处理，用以处理由自己的源代码产生的异常和类库中的类方法产生的异常。如果准备好好利用 .NET环境，那么，你必须掌握第2章中提供的所有信息。

第3章“Visual Basic .NET类编程” 要建立面向对象的程序，必须广泛地使用类来对对象的数据（变量）和方法（对变量进行操作的函数和子例程）进行分组。在第3章中，我们首先介绍Visual Basic .NET类的基础知识，例如，如何进行类定义，并且对这个类创建一个或多个实例（对象）；如何使用点运算符访问类变量和类方法，如何用Public、Private等访问修饰符限制某个程序能够直接访问的类组件；如何使用专门的New方法（程序员们称之为构造器方法）初始化对象的字段；.NET环境如何用“可管理内存”来保存你所建立的对象，.NET垃圾收集器和基于类的析构器方法的作用等。读完第3章，你就能够了解Visual Basic .NET类的所有细节，以及所有对你所建立和使用的类对象产生影响的后台操作。

第4章“Visual Basic .NET中的面向对象编程” Visual Basic .NET程序广泛地使用类来表达对象。你往往可以使用 .NET环境内置的类去执行一些主要的系统操作，如检索系统日期和时间、操纵文件等。除了创建自己的类来代表某个对象，你还可以根据现有的类定义某个新的对象。例如，你可以用现成的Person类中已定义的变量和方法去定义新的Employee类；或者根据现有的Book类去定义新的ComputerBook类。根据现有的类去定义新类的时候，需要用到继承的

概念。第4章详细介绍了继承，介绍了如何利用现有类（也叫基类）中的构造器和析构器方法，如何定义与新类（也叫派生类）名称相同的方法来覆盖基类方法的执行过程。此外，该章还介绍了如何创建多态对象，多态对象的形式会根据程序执行情况而发生改变。例如，某个多态的Phone对象，根据程序执行情况的不同，既可以是便携式电话，也可以是投币式电话。最后，该章还将介绍如何在Visual Basic .NET程序中创建和使用界面。

第5章“使用公用语言运行时库和基类库” .NET环境提供了公用语言运行时库（CLR）和基类库（BCL），其中包含数千个独立于编程语言的类和程序，你可以把它们用到自己的代码中去执行特定的操作。 在第5章中，你将学习几种主要的贯穿本书所有技巧的重要类。首先是DateTime类，你将学习如何用它的类方法去定义当前的系统日期和时间，进行日期比较，确定过去和将来的日期等等。然后是String类，它能实现许多老的Visual Basic字符串操作功能。你将了解String类“永恒不变”的性质将如何降低你的程序性能，以及如何用StringBuilder对象代替它去操纵字符串内容，以便减少系统开销。接着介绍Math类及其方法。最后介绍如何在Visual Basic .NET程序中或ASP .NET页内发送电子邮件消息。

第6章“Visual Basic .NET文件和目录操作编程” Visual Basic .NET程序中大量地使用把信息从一个用户会话保存到另一个用户会话。在第6章中，我们将详细介绍 .NET公用语言运行时库中主要的内置类，你可以用它们执行主要的文件和目录操作。首先，介绍如何创建、选择、移动和删除目录；然后，介绍如何检索包含了某个目录中的文件或子目录的集合。然后介绍如何创建、复制、移动和删除文件，如何使用文件属性，如文件被创建和最后被访问的日期与时间等。接着介绍如何把数据（如表单内容）保存到文件中去，以及如何用文件流操作检索文件内容。该章最后还将介绍如何监视目录中文件发生的改变。

第7章“使用 .NET 通用对话框” 多年以来，Visual Basic程序员一直都采用表单来构造用户界面。为了让程序能以一种统一的模式执行诸如打开、保存、打印文件这样的基本操作，.NET环境提供了一系列用来显示对话框的类，你可以在程序中使用它们来与用户进行交互，这一系列对话框称为通用对话框。第7章详细介绍了其中每个类的用法。另外，该章还介绍了如何预览并打印文档内容。

第8章“使用多线程” 现在，用户通常可以在某个多任务操作系统（如Windows或Linux）下同时运行两个或多个程序。虽然多任务操作系统看起来像是同时运行两个或多个程序，但实际上，CPU在任何给定的时间里，都只能执行一个程序的指令。多任务操作系统只是能够在每个活动程序之间非常迅速地切换CPU控制，所以制造了同时运行多个程序的假象。由于操作系统切换CPU如此迅速，所有程序看起来像是在同时运行。Visual Basic .NET就是使用了类似于这种CPU切换的技术，通过使用多线程执行，使程序看起来像是在同时执行两个或多个任务。一般来说，一个线程的执行对应于一套指令，比如一个子例程。而在Visual Basic .NET程序中，你可以建立多线程对象，指示它们执行特定的程序指令。在程序执行过程中，CPU控制将在线程对象之间进行切换，使人误以为所有线程都在同时执行。第8章详细介绍了线程的执行情况。此外，还介绍了如何同步线程的操作，以防止线程之间的相互干扰。

第9章“结构化错误处理” 多年以来，Visual Basic程序和基于VBScript的动态服务器主页都广泛采用ON ERROR语句对错误做出响应。采用ON ERROR语句，当某个操作产生错误时，

程序能够确定将要执行的语句。在 .NET环境下，Visual Basic .NET程序采用的是公用语言运行时库中内置的大量方法和类。这些方法大多都能响应发生异常时产生的错误。如果程序没有检测并对异常做出响应，程序运行就将终止，同时显示出致命的错误消息，该消息中描述了所出现的异常。一直以来，C++程序员都使用结构化错误处理去响应异常，现在，Visual Basic .NET也提供了同样的功能。在第9章中，我们将学习如何在Visual Basic .NET程序中检测异常并对异常做出响应。

第10章“响应和处理事件” 为了构造用户界面，Visual Basic程序员需要编写代码来响应基于窗体的事件，如鼠标点击事件等。Visual Basic .NET改变了用来定义处理事件的子例程的格式，而且，Visual Basic .NET还提供了Delegate对象，你可以在程序中向这个对象分配一个或多个子例程的地址，当特定事件发生时，程序就去执行该子例程。第10章详细讨论了事件处理和Delegate对象。

第11章“Windows窗体编程” 为了快速构建用户界面，Visual Basic程序员会使用大量的窗体。用Visual Basic窗体构造用户界面的时候，你只需向窗体拖放控件。与Visual Basic相比较，用其他编程语言（如C++）构造用户界面显得相当困难。事实上，构造用户界面也确实常常是程序开发过程中最耗时的一个环节。然而，.NET环境为编程语言（如C#和Visual Basic .NET）提供了用Windows窗体创建用户界面的功能。总体来说，Windows窗体是一种独立于编程语言来实现的一种基于窗体的功能，已被Visual Basic程序员们使用多年了。用Windows窗体，你只需要向窗体拖放控件即可，Visual Studio将会在后台向你的程序中自动添加创建、显示各个控件以及与每个控件进行交互的代码。第11章详细讨论了Windows窗体及其主要控件。

第12章“深入研究 .NET程序集和版本化” 在 .NET环境下创建应用程序的时候，驻留在某个专门文件中的应用程序叫做程序集。该程序集用.exe扩展名表示程序，用.dll扩展名表示类库。程序集中除了保存可执行代码，还包含元数据（关于数据的数据），元数据用来描述该程序以及程序要求的组件（即保存在.dll文件中的类库）。此外，元数据还提供程序版本信息和程序要求的每个组件的版本号信息。程序集提供的这种版本信息对于 .NET功能来说至关重要，因为当新的程序版本或新的.dll文件发布时，用上述版本信息可以避免引起冲突。在第12章，你可以用Visual Studio提供的几种工具仔细研究并更新程序集中包含的信息。此外，你还将学习如何创建类库，以及要在Visual Basic .NET、C#和ASP .NET应用程序之间实现库共享所必须执行的步骤。

第13章“ASP .NET解决方案编程” 目前，程序员们都广泛地使用动态服务器主页来驱动Web站点。通过在脚本（通常写成VBScript）中结合HTML标记，开发者可以在动态服务器主页中建立可根据每个用户使用情况而改变的动态内容、能够与用户进行交互的页面以及使用数据库内容的页面。.NET环境还引入了ASP .NET，从而扩展了Web开发者用来构造动态页面的功能。ASP .NET页跟动态服务器主页一样，也在程序代码中结合了HTML标记。你可以用可编译编程语言如Visual Basic .NET和C#创建ASP .NET页。当使用可编译语言时，ASP .NET页还为你提供了编程语言的所有特性，以及.NET公用语言运行时库的所有特性。在一个*ASP .NET*页内，你可以创建类、使用继承、利用多线程执行、通过异常处理错误等等。由于ASP .NET使用的是可编译语言，其脚本运行比不可编译语言的脚本要快得多。而且，你将在第15章学习到，ASP .NET

页还支持Web窗体，这样，通过向页面拖放Web控件，就能在程序中构造页面。许多Web控件都是基于服务器的，也就是说，服务器所显示的页面是对控件做出的响应，这样可以大大提高每个控件能够执行的处理能力。该章将详细讨论ASP .NET页。

第14章“Windows服务编程” 在Windows环境下，服务就是后台运行的用来执行特定任务的一种专门程序。例如，打印脱机程序能够监视你发送到打印机的输出。同样，Internet Information Services（IIS）服务器也是在后台运行，并向用户发送它们的浏览器要求的Web页和文件。用Visual Studio创建Windows服务非常简单。第14章将向你介绍每个服务器为了与Windows进行通信而必须提供的主要子例程，以及每次Windows启动时，安装和启动某个服务所必须执行的步骤。

第15章“Web窗口编程” .NET环境自带了ASP .NET模式，用于构造动态Web页。你可以用可编译编程语言（如C#或Visual Basic .NET）创建ASP .NET页。用可编译编程语言构造Web页的一个主要好处，就是可以在程序中使用公用语言运行时库内置的程序。而且，为了简化构造Web页用户界面的过程，.NET环境还引进了Web窗体。通常，使用Web窗体，你可以向Web页拖放控件，这跟在传统的Visual Basic程序中构造表单很相似。然后，再向每个控件分配代码。Web控件是你惟一能够直接指示它执行服务器处理的控件，这一点使得控件执行的处理变得非常复杂。第15章详细讨论了Web窗体和各个基于Web的控件。

第16章“Web服务编程” 一直以来，应用程序始终都是朝着广泛使用网络和支持远程操作的方向发展。通常，在分布式环境下，运行在客户机上的应用程序能够访问存储在远程服务器上的数据，比方说，访问某个公司的数据库。过去，许多网络操作系统都支持远程过程调用（称为RPC），就是允许程序调用驻留在远程系统中的函数或过程。在这种程序的源代码中，对远程过程的调用看起来就像通常的过程调用，其中同样包括子例程名或函数名及其参数。然而，在后台，远程过程调用却要求程序与服务器之间进行信息交换：程序向服务器发送消息，指定它所需要调用的子例程和函数及其对应的参数；远程过程工作完毕以后，服务器向主程序发回该过程运行的结果。Web服务为程序员提供了通过Internet执行远程过程调用的能力，而 .NET环境又使得创建Web服务非常之简便。第16章将向你详细介绍创建和调用Web服务所必须执行的步骤。

第17章“ADO .NET入门指南” 以前，程序员们都使用动态数据对象（Active Data Object，ADO）简化数据库应用程序。.NET环境中含有一种新的数据库访问模式，叫做ADO .NET，它在ADO模式的基础上有了明显的提高。ADO .NET模式引进了DataSet对象，用来代替过去ADO中的RecordSet对象。第17章将向你介绍如何用ADO .NET连接、查询和更新数据库；ADO .NET模式如何用XML为DateSet对象中包含的数据提供数据结构；如何在ASP .NET页内执行ADO .NET操作，以及如何将保存在表格中的数据映射到DataGrid控件中并显示出来。

第18章“.NET反射和程序属性编程” .NET环境广泛地使用元数据（关于数据的数据）来让对象实现自我描述。也就是说，在程序执行时，该程序还能查询某个对象，以便了解该对象提供的功能。程序员们把这种对对象进行查询的功能叫做反射。通过反射，程序可以了解类中所有方法的细节、类成员变量等等。第18章介绍了查询对象功能必须执行的步骤。你将学习如何查询一个程序集，该程序集中含有某个应用程序，或者它提供的所有类的类库；然后，你将

对那些类进行信息检索，比方说检索每个方法的类型（子例程或函数），以及该方法使用的参数数目和类型。有了这些信息，程序就可以调用类方法，并向方法传递正确的参数值。为了向程序提供更多关于某个实体的信息，比如类、方法，甚至程序集本身，Visual Basic .NET还支持属性。例如，你可能用一套属性去影响编译器的操作，而用另一套属性去控制调试器执行其他的操作。第18章将详细介绍Visual Studio插入到你程序中的属性，以及创建和使用自己定制的属性的方法。

如何使用本书

本书各章都是根据前面章节的内容进行编写的，不过，我们还对本书结构进行了一些处理，你可以翻到任何一条技巧，并找到你所需要的内容。翻阅本书的时候，请注意[redacted]图标，它醒目地列出了各个使用步骤，你可以立即执行这些步骤完成一项任务。例如，假设你需要保存数据到文件中，并从文件中检索数据，你可以直接翻到第6章中标题为“启动文件流”的技巧，它将告诉你如何进行文件读写；如果需要检测并处理某个具体的异常，可以翻到第9章标题为“捕获特定异常”的技巧。

为了帮助你迅速找到所需信息，我们在每一章的开头都列出了该章介绍的所有技巧。如果需要某个主题下的更多信息，可以阅读各章中的介绍性文字，它们将为你灵活使用这些技巧打下坚实的基础。

如果你没有Visual Basic编程经验，应当首先阅读第1章，其中包含了必须要了解的介绍性信息。如果你有Visual Basic编程经验，也请从第1章开始，因为其中提供的技巧详细分析了老版本的Visual Basic与Visual Basic .NET之间的区别。如果还不了解 .NET环境，请你阅读第2章，其中详细分析了许多重要的 .NET功能。

参加本书翻译的人员除封面署名外还有：龚克、田蕴哲、牛志奇、龚舒宇、丁天、李红玲、白红莉、金荣学、薛彪、叶哲、田韫、李林、张巧莉等。

目　录

第1章 Visual Basic .NET基础知识

本章技巧

- 创建第一个控制台应用程序。
- 构造基于Windows的应用程序。
- 选择正确的Visual Basic类型。
- 在Visual Basic .NET程序中声明变量。
- 用Console.Write和Console.WriteLine显示屏幕输出。
- 用Console.WriteLine格式化程序输出。
- 向字符串末尾追加字符。
- 强制程序指定变量的类型。
- 小心变量溢出和变量的精度。
- 执行数字操作。
- 不同变量类型之间值的转换。
- 用条件运算符做判断。
- Visual Basic .NET的关系运算符和逻辑运算符。
- 用Select处理多个条件。
- 重复执行系列指令。
- 避免无限循环。
- 提前结束循环。
- Visual Basic .NET支持滞后求值以提高性能。
- 长语句换行。
- 使用Visual Basic赋值运算符。
- 对程序代码进行注释。
- 用Console.Read和Console.ReadLine读取键盘输入。
- 在消息框中显示消息。
- 用输入框提示用户输入。
- 把程序分成便于管理的程序段。
- 向函数或子例程传递参数。
- 在函数或子例程中声明局部变量。
- 在子例程中改变参数值。
- 用变量的作用域表示程序中变量有意义的区域。
- 在同一变量中保存相同类型的多个值。
- 使用结构进行值分组。

• 用常量提高代码的可读性。

• Visual Basic与Visual Basic .NET区别小结。

在过去的十年中，Visual Basic逐渐成为了绝大多数程序员首选的编程语言。许多程序员认为Visual Basic易于使用，这是它主要的成功之道。也有人认为，在Visual Basic中通过向窗体拖放控件能够迅速构造用户界面，这一功能是Visual Basic得到广泛使用的主要原因。

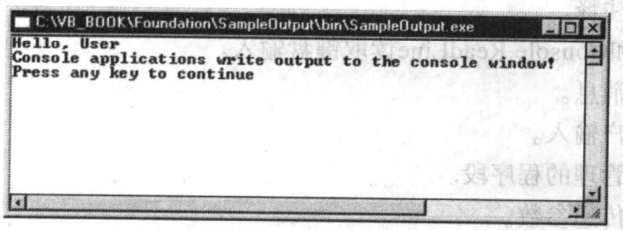
尽管许多程序员已经用Visual Basic实现了广大范围内的编程任务的解决方案，但是，也有大量的程序开发者，其中许多人已经使用了多年的C或C++语言，他们拒绝承认Visual Basic语言适用于专业级的编程。他们认为，虽然Visual Basic提供了一种构造原型的方便渠道，但最终还必须用C++等语言重写代码，才能获得更好的性能。

Microsoft公司在 .NET环境下主要集成了两种崭新的编程语言： Visual Basic .NET和C#（Microsoft也把Visual C++ .NET作为了 .NET环境的组成部分）。稍后你会了解到， .NET环境提供了独立于编程语言的类和程序，C#和Visual Basic .NET同样可以使用它们。也就是说，无论程序员编程时使用的是C#还是Visual Basic .NET，他都能够使用相同的 .NET功能。从性能的角度来讲，Visual Basic .NET应用程序将会与用C#编写的相同程序并驾齐驱。虽然 .NET环境为C#程序员提供了拖放控件到窗体中以迅速构造用户界面的功能，但绝大多数原先的Visual Basic用户转而使用Visual Basic .NET使它作为了一种 .NET编程语言。

本书中的18个章节将详细讨论Visual Basic .NET和 .NET环境。本章旨在向不熟悉Visual Basic的用户介绍一些基本知识，以帮助他们理解Visual Basic .NET和 .NET环境提供的功能。如果有Visual Basic编程经验，可以简单地浏览一下本章提供的这些技巧，找出你不熟悉的部分阅读，然后，请翻到本章的最后一条技巧，其中总结了Visual Basic和Visual Basic .NET之间的主要区别。不管你是否熟悉Visual Basic，现在就让我们开始吧！

1.1　创建第一个控制台应用程序

用Visual Basic .NET可以创建各种应用程序类型，比方说，基于控制台的程序是在类似于MS-DOS的窗口中显示输出，如下图所示；而基于窗口的程序则常常显示基于窗体的界面，或者ASP .NET页，等等。



由于基于控制台的应用程序便于使用（你可以快速地创建程序并显示简单的输出，而不需要向窗体放置控件），本书中的许多技巧都采用基于控制台的应用程序。

实践　用Visual Studio创建控制台应用程序，应当遵循如下的步骤：

1）在Visual Studio中，选择File | New | Project，Visual Studio将显示New Project对话框。

2）在New Project对话框中，点击Console Application图标。在Name栏输入项目名称，用来

描述你构建的程序，比方说DemoProgram（不需要输入扩展名）。然后，在Location栏输入文件夹名，也就是你希望Visual Studio存放你的项目文件夹和文件的位置。点击OK，Visual Studio将显示代码窗口，如图1-1所示，你可以在其中输入你的程序语句。

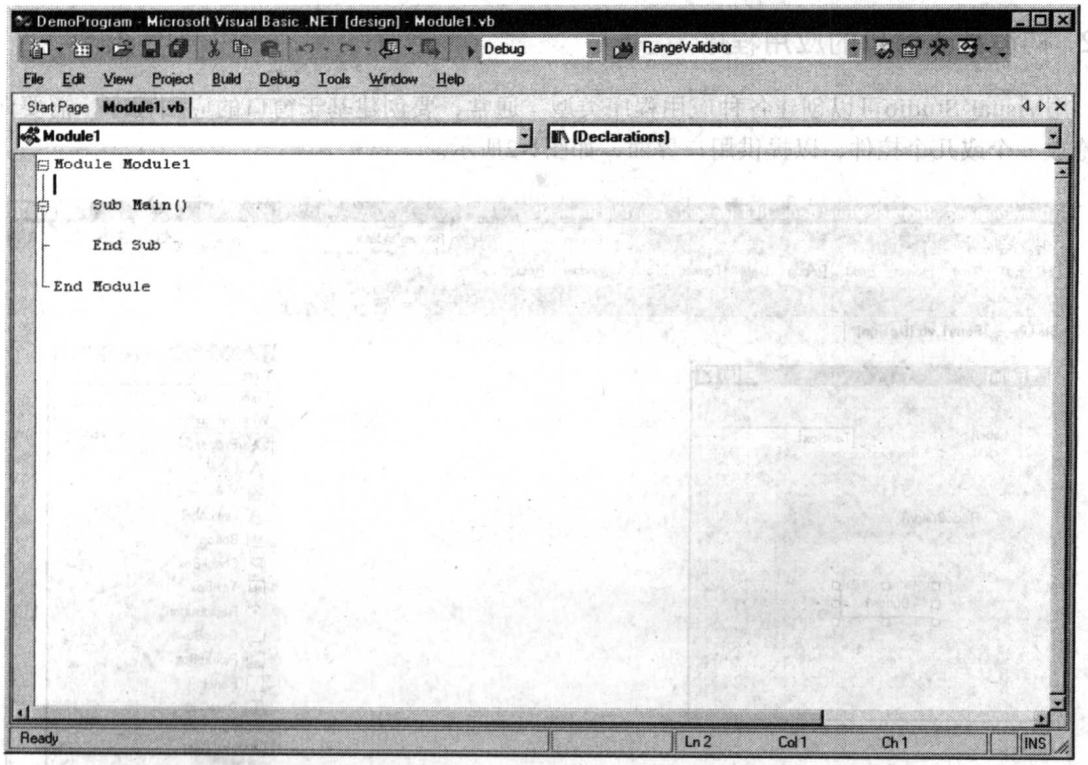


图1-1 在Visual Studio中显示代码窗口

请你在代码窗口的Main子例程中输入下列语句：

```
Sub Main()
Console.WriteLine("My first VB.NET Program!")
Console.ReadLine()
End Sub
```

然后，选择Debug | Start运行这个程序。Visual Studio将在控制台窗口中显示你的程序的输出。按Enter键则中止这个程序，即指示Visual Studio关闭该窗口。

当输入Visual Basic .NET程序语句的时候，请务必按照本书中出现的格式原样输入，就连引号、逗号、句号等也不能漏掉。否则，你的程序就可能违反一条或几条Visual Basic .NET语法规则（也就是定义语言结构和程序格式的规则，创建程序时必须遵循这些规则）。发生语法错误时，Visual Studio将显示错误消息，该消息描述了所发生的错误以及代码中出现错误的行号。你必须在Visual Studio编译你的程序之前找到并改正这个语法错误。

每次对程序代码做过修改以后，都应当让Visual Studio重新生成程序，这样你的修改才有效。选择Build | Build Solution重新生成程序。例如，在前面的程序语句中，需要修改代码使之显示