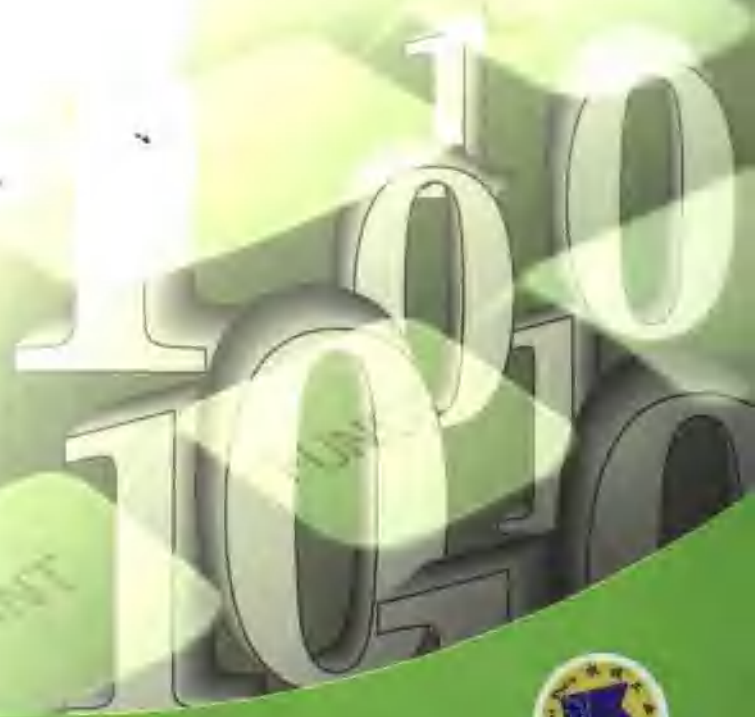
国家技能型紧缺人才培养培训工程
高职高专软件技术专业规划教材

Visual Basic.NET 面向对象程序设计

——基础、设计、实现与应用程序开发

邵鹏鸣　编著



机械工业出版社
CHINA MACHINE PRESS

国家技能型紧缺人才培养培训工程

高职高专软件技术专业规划教材

Visual Basic.NET 面向对象程序设计

——基础、设计、实现与应用程序开发

邵鹏鸣　编著

机 械 工 业 出 版 社

微软的.NET 战略是一场软件革命，它改变了开发人员开发应用程序的方式及思维方式，使得开发人员能创建出全新的各种应用程序。Visual Basic .NET 是微软公司推出的新一代面向对象的编程语言，它功能强大、编程简洁、明快，吸收了 Java 语言很多的特点和精华，是一种简便快捷地创建 .NET 应用程序（包括 XML Web services 和 Web 应用程序）的方法，也是微软的.NET 战略的重要组成部分。全书共分 11 章，通过大量的与实际应用程序设计有关的案例深入浅出地全面而详尽地讲解了 VB.NET 程序设计的基本方法与技巧及注意事项，注重培养编写实际应用程序的能力。帮助读者关注编写程序的重要环节及过程，养成良好的编程习惯。全书贯穿了面向对象编程的程序设计思想和设计方法，并用两章的篇幅讨论使用 ADO.NET 与 SQL 访问数据库的编程技术。通过本书的学习，读者应达到五个目标：面向对象的程序设计、Windows 应用程序设计、ADO.NET 及数据库应用程序设计、文件的输入输出以及它们的综合应用。

本书内容丰富、可操纵性强、语言生动流畅、没有晦涩的术语，采用面向实际的技术和面向实际的应用程序驱动的教学方式，使学生能够在轻松愉快的环境下掌握 Visual Basic .NET 的基本编程方法与技巧。

本书可作为高职高专院校计算机专业学生和应用型高等院校计算机专业学生的教材和教学参考书，也可作为初中级读者和培训班学员学习的教材。本书配有电子教案供教师使用，可发电子邮件至 wangyx@mail.machineinfo.gov.cn 邮箱索取。

图书在版编目（CIP）数据

Visual Basic.NET 面向对象程序设计：基础、设计、实现与应用程序开发 / 邵鹏鸣编著. —北京：机械工业出版社，2006.2

高职高专软件技术专业规划教材

ISBN 7-111-18334-7

Ⅰ. V… Ⅱ. 邵… Ⅲ. BASIC 语言—程序设计—高等学校：技术学校—教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2006）第 001145 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：王玉鑫 版式设计：霍永明 责任校对：刘志文

封面设计：鞠 杨 责任印制：洪汉军

三河市宏达印刷有限公司印刷

2006 年 3 月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 38.75 印张 · 961 千字

0001—4000 册

定价：48.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68326294

编辑热线（010）88379739

前　言

随着 21 世纪的到来，计算机技术的发展更加迅猛，在各行各业的应用更加广泛，面对日新月异的新技术、新方法，我们必须对现有计算机课程的设置和教学内容进行调整，以适应技术进步与市场变化的需要，使教授的内容是市场上最需要的。

今天，应用程序已由驻留在用户硬盘上的独立可执行文件发展为由 Web 服务器在 Internet 上传送的分布式应用程序，相应地，任何一种开发平台及程序设计语言都必须适应这种变化。微软的.NET 是一种开发平台，Visual Basic .NET 是在.NET 平台上编程的一种高级语言，.NET 有着广阔的应用前景，.NET 的应用必将对整个计算机产业产生重要而深刻的影响。.NET 不但改变了开发人员开发应用程序的方式及思维方式，而且使开发人员能创建出全新的各种应用程序，大幅度提高软件生产率。未来.NET 将无处不在。

微软以.NET 平台和 C#、VB.NET 编程语言的形式推出了与 Java 全面竞争的对手。这是微软在编程语言和编程环境这块竞技场上所做出的最出色的成果。当然，他们有相当有利条件：他们可以看到 Java 在什么方面做得好，在什么方面做得还不够好，然后，基于此去构建，并要具备 Java 不具备的优点。

目前，与 Java 相比，对.NET 平台人们所关心的最重要的问题是微软是否会将它完全移植到其他平台上。微软宣称这没有问题，而且 Mono 项目（www.go-mono.com）已经有了一个在 Linux 上运行的.NET 程序。

我们知道 C 语言是一种代码效率很高但不易于进行快速开发的程序设计语言。OOP 出现后，OOP 与 C 语言融和产生了 C++，继而有了集成开发环境。但 C++的出现，并没有使 C 语言家族在应用开发方面获得突飞猛进的发展。基本上还是占据着 UNIX、Linux 以及底层应用开发的天地，而在 Windows 大型应用软件特别是数据库和 Web 开发上，由于固有的复杂性和缺乏针对性，就不如 VB 等具有很强针对性的开发工具。如果使用过包括 C 和 C++在内的多种程序设计语言的人，相信会深深地体会到它们之间的区别。比如与 Visual Basic 相比，Visual C++程序员为实现同样的功能就要花费更长的开发周期。

Visual Basic .NET 程序结构十分清晰，较易学习和使用，同时又不失灵活性和强大的功能，它吸收了 Java 语言很多的特点和精华，它在开发能力和效率之间取得较好的平衡。它不仅具有快速开发应用程序的能力，而且已成为功能强大的面向对象的编程语言。

Visual Basic .NET 是 .NET 框架的支柱。它已不再是过去的 Visual Basic，它可以充分利用.NET 框架类库和公共语言运行库编写出功能强大的各种应用程序。Visual Basic .NET 改变了它传统的开发应用程序的方式及思维方式，使得开发人员能创建出全新的各种应用程序。Visual Basic .NET 为 .NET 提供了最方便的入口点。

学习一种先进的语言和一种先进的编程方法将激起学生更大的兴趣。这些知识在他们离开学校，进入一个由信息占据重要地位的世界时可以立即发挥作用。正是这一点激发了他们对知识的学习热情。当今的学生必须同时掌握基础语言、面向对象编程和类库，而 VB.NET 课程适合这类知识的学习。运用 VB.NET 能够出色地完成任务，所以他们更愿意投入更多的

精力和时间。

很多院校都有软件专业的学生，有的学校的学生在几年的学习中，不仅学了 C、C++而且也学了 Java、VB 或 VB.NET 甚至 C#，同时还学习了 Web 应用程序设计及其他有关软件开发的课程。但据作者调查的院校，有些院校软件专业的学生在毕业找工作时，很难找到一份对口的工作，他们的工作与软件无关，甚至与计算机无关。不是企业不需要软件专业的人才，而是没有所需要的人才。我认识一位学生，学习成绩不错，学习也很勤奋，在毕业前夕还通过了全国计算机技术与软件专业技术资格（水平）考试，获得软件设计师资格。但当我交给他一份实际任务时，他根本无从下手，尽管该任务很简单。这些都说明，我们的教学体系、教学内容、教学模式和方法都存在很多需要我们去改进的地方。

作者极力反对用“玩具例程”说明概念、原理的教学方法。这些例子完全脱离实际，看起来很幼稚，尽管它能够说明一个概念，学生也容易看懂，上课也容易听懂。但是检查学习好坏的标准，不是“知道还是不知道，听懂还是没有听懂，看懂还是没有看懂”，而是“会不会应用”。真实例程同样能够说明概念、原理和知识点。只不过是面对复杂的技术实践体系，需要花大量的时间、精力精心地去设计这样的例程，只要设计得好，真实例程同样能够说明概念、原理和知识点，学生同样容易看懂、听懂，并且懂得在实际应用中如何运用这些概念、原理和知识点，收到事半功倍的效果。这就是作者著写该书的目标之一。

作者的目标很明确，写一本实用，有一定的深度且易读的程序设计语言教材，无论读者有无编程经验，要做到开卷有益，重要的是能够激发他们的学习兴趣，能够将他们所学的知识应用于实际。

本书将基本技能培养和主流技术相结合，以市场对人才的需求为依据，从应用与工程实践的角度出发进行组织撰写，其主要特点如下：

1. 新体系、新内容、新手段、新思路。无论是内容体系、编写的思路、教学的模式及理念等都具有一定的新意。

2. 先进性及实用性。本书的内容反映最新的实用的程序设计方法及技术，顺应并符合教学改革发展的规律，书中讲授的程序设计方法与现代编程方法步调一致，具有很强的实用性。

3. 面向实际的应用程序驱动的教学方式。本书不是采用传统的提出概念—解释概念—举例说明的方法，而是以实践为主线，以应用为目标，通过完成任务的方式学习程序设计知识。本书采用提出任务—介绍完成任务的方法及步骤—最后归纳出一般的规律、概念及知识点的模式。每一个新概念、知识点的提出都伴随着一个完整的、可实际运行的实用程序及其输入、输出。学生通过完成实例，掌握概念、知识点。

4. 使读者从一开始就编写有用的程序，这有助于保持读者的学习兴趣及动力。书中的实例不是只有几行代码的小程序，这些实例来源于实际应用程序或其中的一部分，它们与我们的生活相关或者是大家感兴趣的内容。本书包含大量这样的实例，这些实例及其代码分析与讨论是本书的精髓。

5. 逐步展现主题鲜明的各章。本书将集中重要的主题并进行充分论述，而不是肤浅地涉及许多问题，罗列许多概念、术语、语法。本书的内容及体系结构使学生感到它既具有易读性又增长知识，具有很强的实用性。

6. 面向实际的技术。学习程序设计语言的目的是为了开发程序，本书不只是讲授 VB.NET 的语言要素、语法，而是教会读者如何用 VB.NET 开发程序，书中的每一个面向实际的应用

程序实例都说明了开发过程及包含的知识点。通过学习如何开发程序来掌握语言要素、语法。在第 8 章，通过一个实际应用程序实例，讲授了三层架构的应用程序设计方法和开发过程，三层应用程序设计是企业应用程序常用的程序设计方法。后续的许多实例（包括第 10 章和第 11 章的实例）都应用了三层应用程序设计，作者通过这种方式，使读者能够熟练应用这种企业常用的程序设计方法和开发过程。

7．面向对象的程序设计。本书从始至终贯穿面向对象编程的思想。我们的目标之一是使学生习惯于现实中的程序设计，仅仅知道面向对象的概念是不够的，学生们必须能够运用这些知识开发现实中的程序。教学概念的核心之一是在成为一名对象设计者之前，必须首先成为一名对象用户。换句话说，在有效的设计自己的类之前，必须首先学会使用预定义的类，我们从第一个程序开始就接触如何使用类。学习使用类库的类和编写自己的类相辅相成，互相促进，学习使用类库中的类有助于学习编写自己定义的类，学习自定义类又有助于学会使用类库中的一些类，并加深对类库中的一些类的理解。当今学生必须同时掌握基础语言，面向对象编程和类库，我们将这三者接合在一起。

8．循序渐进，由浅入深，综合应用。本书在设计时贯穿了一个思想：面向实际，同时使读者容易理解、明白所包含的所有细节。因此本书在安排时，循序渐进，由浅入深，后序章节在讲授新的知识点的同时综合应用了前面所学章节的知识。这样做既温故了前面所学的知识，同时也懂得了如何综合应用前面所学的知识解决问题。书中很多实际应用程序实例，不只是只讲授一个新的知识点，而是在讲授新知识点的同时，应用了前面所学过的知识。因为实际应用程序不可能只包含一个知识点。学习知识的目的，是为了应用知识。

9．数据库。数据库应用程序如今对所有企业来说是至关重要的，因此本书用两章的篇幅通过一个具体的实际应用程序实例讨论使用 ADO.NET 与 SQL 访问数据库的编程技术。并应用了三层架构的应用程序设计方法和开发过程。

我们要求读者遵循书中介绍的方法步骤实际建立实例程序，然后将例程序进行修改或扩展，并通过对实例的代码进行分析与讨论掌握实例背后包含的概念、原理、知识点和方法等，我们认为这是学习程序设计最稳妥、最有效、最快捷的途径。

通过本书的学习，读者应达到如下目标：

- 面向对象的程序设计。
- Windows 应用程序设计。
- ADO.NET 及数据库应用程序设计。
- 文件的输入、输出。
- 以上内容的综合应用。

由于水平和时间的关系，书中的错误在所难免。如果发现不当之处，欢迎提出宝贵意见，并将信息反馈给我们（pmshao@163.com），将不胜感激。

邵鹏鸣

目　录

第 1 部分　VB.NET 基本知识

第 1 章　认识 VB.NET

本章主要内容：

1. 什么是.NET。
2. 第一个简单的控制台应用程序。
3. 创建简单的 Windows 应用程序。
4. 对象、属性和方法。给对象的属性赋值。
5. Click 事件及其处理程序。

目标：

- 熟悉 Visual Studio.NET 集成开发环境（IDE）。
- 学会创建、编译和执行简单的.NET 应用程序。
- 使用输入和输出。
- 初步认识和了解窗体、控件、事件和方法。

1.1　什么是.NET

.NET 是微软公司的新战略，它包含微软公司对未来的核心战略、规划和洞察力。所有微软公司的产品都将围绕这个战略开发。此战略的核心就是.NET Framework，该框架提供了全面支持.NET 的核心技术。.NET Framework 是一种新的计算平台，它简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 具有两个主要组件：公共语言运行库和 .NET Framework 类库。

.NET 用来解决编程人员面临的许多问题：

1）它负责处理在创建大型、可靠的应用程序时的大量艰辛工作。

2）它允许程序员统一两种架构——在本地机器上运行的应用程序和通过 Web 访问的应用程序。

3）它减少了与编程框架相关的传统开销——不再需要高性能编程语言来编写复杂的代码以获取高速的.NET 程序。

4）它允许不同语言的程序员在一个应用程序中协同工作。

5）它开始兼容各种最终用户工具，包括桌面、PDA 和手机。最终实现使开发人员能够创建出摆脱设备硬件束缚的，能够在各种操作系统上运行的应用程序，能够轻松实现互联网的连接。

总之，.NET 提供了一种更简单、更快捷、更廉价的方式，来获得高效的程序。

在某些方法上，.NET 很像 Java。实际上，Java 的口号是“一旦编写出来，就能在任何地方运行”很符合.NET 的原则。不过.NET 并不是 Java 的克隆，微软的方法与之并不相同。

Java 是“一种语言，多个平台”，而.NET 是“多种语言，一个平台……（就现在而言）”，微软想扫除进入.NET 的障碍，为此就要让使用任何语言的人都可对它进行访问。.NET 有两种语言，即 C#和 Visual Basic.NET，而 Visual Studio.NET 提供了这两种语言。

微软的策略更像是一场军事战役。它运用对 Windows 平台的理解，首先攻占 Windows 平台。当它从 Windows 平台中脱颖而出后，就可以“侵袭”别的平台。例如 Linux，证明.NET 应用程序可以从一个平台移植到另一个平台上。“侵袭并征服了 Linux”后，它又可转向另外的平台，例如 Solaris。

简而言之，.NET 将操作系统平台割裂开来。无论是 Windows、Linux 等哪种平台，都可以分成两个层次：程序设计层和执行层。.NET 开发人员是为程序设计层而不是执行层编写软件，将来不管占统治地位的是 Windows 平台还是 Linux 平台，或者是其他出乎意料的系统，就无关紧要了。

可以将.NET Framework 的讨论分解成以下几部分：

（1）MS 中间语言（MS Intermediate Language） 在执行用户编写的所有程序代码前，应将其编译成更抽象、精简的形式。程序员可以使用任何.NET 语言编写代码，包括 VB、C#、Jscript 和其他大约 20 种语言。这些代码都会编译为 MSIL 这种.NET 的通用语言。.NET 在这个层次的操作无需用户干涉，因此本书不讨论它。

（2）公共语言运行时（Common Language Runtime——CLR） 这是一个复杂的系统，用于在计算机上执行 MSIL 代码。公共语言运行时负责执行与 Windows 和 IIS 交互时所涉及的全部实质性任务。这个层次也是在后台进行的，本书也不涉及。

（3）.NET Framework 类库（.NET Framework Class Libraries） 实现大量重要功能的代码库。用户可以非常方便地在应用程序中调用库函数，使复杂任务的程序代码更为简洁。本书将介绍这些功能。

（4）.NET 语言（.NET Language） .NET 语言是符合一些特殊结构要求（由公共语言规范定义）的编程语言，能够编译成 MSIL。可以用任一种语言进行开发，例如 C#或 VB.NET，没有任何限制，也可以用多种语言来开发程序。本书将用几章的篇幅讨论 VB 在 ASP.NET 中的应用，并在所有的代码示例中使用 VB。

（5）ASP.NET 这个代码模块扩展了 Internet Information Server(IIS),使之能为 Web 页面执行.NET Framework。

（6）Web 服务 虽然 Web 服务不是严格意义上的.NET 的一部分，但.NET 明确地支持它。它们是可以通过 Web 访问的组件，可以是任何主题，例如新闻标题、天气预报、服票走势、病毒预防和操作系统更新等。

1.1.1 从用户代码到机器代码

你也许知道，计算机识别以二进制形式表示的任何事情（二进制位是表示指令和数据的 1 与 0 的序列）。因此，人们热衷于用“数字”一词来描述任何事情，即使它们与计算机的关系很小。通常把这种二进制位指令称为机器码。很明显，对大多数人来说，记住用于打印“Good

Morning”时的 0.1 代码序列是不可能的，更不用说要记住定义复杂 Web 应用程序的程序代码了。因此，用户级编程语言，通过类似英语的单词来编写代码。

一旦用高级语言编写了程序代码，就需要将其转换为机器码，这一转换过程称为编译，编译器软件就可以将人类可读指令编译成机器可读指令。编译过程包括将本机环境信息写入经编译的程序代码这一过程，因此编译后的程序代码可以最高效地利用计算机的所有可用资源。

多年来，有以下两种编译类型，它们的编译过程完全不同。

（1）预编译型代码　在编码完成后并在运行前进行编译的代码。对这一过程代码，由于编译器会花时间分析全部的代码和要运行它的机器，因此代码能以很快的速度执行。不过，由于预编译的代码是针对某一计算机的，因此，除非仍在该计算机上使用编译过的代码。否则需要将另一台计算机设成与第一台计算机相同的系统与资源。

（2）解释型代码　边执行（在用户请求页面时）边编译的代码。这种代码执行较慢，因为需要为每个请求编译代码，且系统没有机会全面优化所编写的代码。不过，其优点是解释过程可以调整，以适应运行代码的机器。

所以，开发人员在选择语言时要做出取舍，既可以选择较慢的解释性代码，以适应运行代码的机器，也可以选择较快速的预编译代码，但不能充分利用机器的优势。

1.1.2　两种中间语言介绍

.NET 在编译时采用两步来解决这个问题。编写在.NET Framework 上运行的程序时（通常用 VB.NET 或 C#编写），需要在使用这些程序之前编译这些可读代码。.NET 编译器的设计方式意味着它并没有把我们带入会引起可移植性问题的二进制码。事实上，.NET 编译器将程序代码编译成称为 MS 中间语言（MSIL）的特殊格式。由于 MSIL 结构不需要像源代码那样易读，因此编译过程包括了一些优化操作。但是，并没有针对某台机器进行优化。因此，MSIL 具有一般的优化性能，并可以移植到任何.NET 服务器上。

当用户执行 MSIL 代码时（即请求 ASP.NET 页面），将其传给 CLR，CLR 是.NET Framework 的另一个核心。CLR 使用 JIT(just-In-Time)编译器将代码编译成真正的机器码，并对程序进行最后的且与机器相匹配的优化，以使程序能在其所在的计算机上以尽可能快的速度运行。

MSIL 和 CLR 组合使用，具有两种编译代码的优点，即预编译码的结构优化和解释码的可移植性。

更为重要的是，MSIL 与本身机器无关。因此，可以在装有 CLR 的任一计算机上运行。实际上，一旦编写出.NET 程序代码并将其编译，就可以将它复制到装有 CLR 的计算机，并在该计算机上执行。虽然目前的 CLR 只有与 Windows（9x、NT、2000、XP 版）兼容的版本，但现在已着手建立基于其他操作系统的版本。在 Web 上搜索与 Mono Project 的信息，就可以找到更多的 CLR 版本。

MSIL 也可以由任一遵循 CLS 的可读语言生成。VB.NET、C#和 Jscript.NET 是“与.NET 兼容的”三大语言。另外 MSIL 编译器还支持其他 20 多种语言。因此，可以在应用程序内部交替地使用这些兼容语言。一旦将一套文件编译成 MSIL，它们都将统一为一种语言。这种灵活性允许不同的小组在同一个 Web 站点上用不同的语言协同工作。

1.2 第一个简单的控制台应用程序

实例：打印一行文字

打印一行文字如图 1-1 所示。

主要知识点：VB.NET 程序的基本经过结构。控制台的输入和输出。

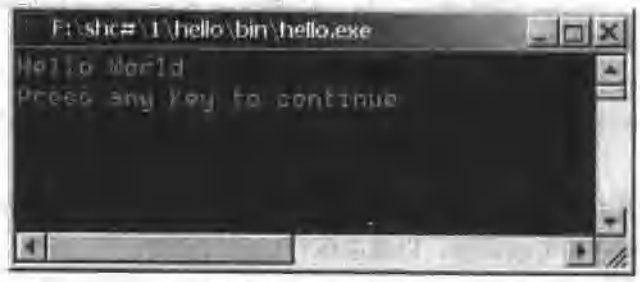


图 1-1 打印一行文字

实现步骤：

1）启动 Visual Studio。

2）从“文件”菜单上指向“新建”，然后选择“项目”。

3）在“项目类型”窗格中选择“Visual Basic”，然后在“模板”窗格中选择“空项目”。

4）在“名称”框中，键入 Hello 命名该项目。在“位置”框中，输入要将项目保存到的目录，或单击“浏览”按钮以定位目录。

5）单击“确定”按钮。Visual Studio 创建一个 Hello 新项目，并显示解决方案资源管理器。若没有显示解决方案资源管理器，请按 Ctrl+Alt+L，显示解决方案资源管理器。

6）在解决方案资源管理器中，选择解决方案下的 Hello 项目，单击鼠标右键，再将鼠标指向“添加”，然后单击“添加新项”。

7）在“类别”窗格中选择“本地项目”，然后在“模板”窗格中选择“代码文件”。在“名称”框中，键入 TESTHello。

8）单击“打开”。然后在空的 TESTHello 代码文件中输入如下代码：

```
'第一个简单的 VB.NET 控制台应用程序
Public Class HELLO
    Shared Sub Main ()
        System.Console.WriteLine("Hello，World ") ' 输出 Hello World
    End Sub
End Class
```

9）在解决方案资源管理器中，选择解决方案下的 Hello 项目，单击鼠标右键，然后单击“属性”。

10）在输出类型下，单击向下箭头，选择“控制台应用程序”，单击“确定”。

11）按 F5 键运行该应用程序，并验证在命令行窗口中已显示“Hello, World!”（见图 1-1）。

12）单击任意键结束程序运行。

代码分析与讨论：

（1）代码注释　第一行包含注释语句：

```
'第一个简单的 VB.NET 控制台应用程序
```

其中“'”字符称为注释符号，它将这行的其余内容转换为注释内容。可以将整行作为注释，或者可以在其他语句的结尾追加一个注释，如下所示：

```
System.Console WriteLine("Hello World ")   '   输出 Hello World
```

程序员在程序中加入注释，用于提高程序的可读性，使程序易于阅读和理解。计算机在执行程序时不会执行注释行。以“'”开始的注释只对当前行有效，如果注释需要多行，请在每行的前面使用注释符号。

（2）定义类　VB.NET 的每一个程序包括至少一个自定义类。这些类称作程序员自定义类或用户自定义类。在 VB.NET 中用关键字 class 引导一个类的定义，其后接着类的名称（本例中是 Hello），关键字是 VB.NET 的保留用字。End Class 用来结束类的定义。例如：

```
Public Class HELLO
...
End Class
```

（3）Main 方法　VB.NET 程序必须包含一个 Main 方法，而且必须按第二行那样定义，Main 方法是程序的入口点，程序控制在该方法中开始和结束。方法用来执行任务及在任务完成后返回信息。Sub 关键字表明该方法将执行一个任务，但完成任务后不返回信息。

Main 方法在类的内部声明，它必须具有 Shared 关键字，是静态方法，在第 5 章“类与对象”将讨论静态方法。在"Hello World!" 示例中，Main 方法是 Hello 类的成员。

“End Sub”用来结束 Sub 方法的定义。

（4）输入和输出　程序通常使用 .NET 框架的运行时库提供的输入/输出服务。在 Main 方法中，语句：

```
System Console WriteLine("Hello World!"),
```

使计算机打印双引号之间的字符串"Hello World!"，我们通常将双引号之间的字符或字符序列称为字符串。以上语句使用 了 WriteLine 方法。WriteLine 方法是类库中 Console 类的输出方法之一，WriteLine 方法在命令窗口中显示一行文字后，自动将光标移动到下一行。

如下代码段使用了 ReadLine 方法:

```
Dim   Str As string
Str=System Console. ReadLine ( )
```

ReadLine 方法是运行时库中 Console 类的输入方法之一，它用来输入一字符串，按回车键结束输入。其他 Console 方法用于不同的输入和输出操作。

如果在程序开头包含以下 Imports 语句：

```
Imports System
```

则可直接使用 Console 类和方法，无需使用完全限定名。例如：

```
Console WriteLine("Hello World!");
```

Imports System 语句引用一个由 Microsoft .NET 框架类库提供的、名为 System 的命名空间。此命名空间包含 Main 方法中引用的 Console 类。命名空间提供了一种分层方法来组织一个或多个程序的元素。Imports 语句可以非限定地使用属于命名空间的类。“hello, world”程序使用 Console.WriteLine 作为 System.Console.WriteLine 的简写形式。

（5）编译并运行程序　从 IDE 编译并运行程序：按 F5 键来生成并运行（对应于“调试”菜单中的“启动”）。

1.3 创建简单的 Windows 应用程序

前面一个程序在命令窗口中显示输出，大多数 VB.NET 程序使用窗口或对话框显示输出。

实例：在对话框中显示一行文字

在对话框中显示一行文字如图 1-2 所示。

主要知识点：Imports 语句。在对话框中显示信息。

图 1-2 在对话框中显示一行文字

实现步骤：

1）启动 Visual Studio。

2）从“文件”菜单上指向“新建”，然后选择“项目”。

3）在“项目类型”窗格中选择“Visual Basic”，然后在“模板”窗格中选择“空项目”。

4）在“名称”框中，键入 Hello 命名该项目。在“位置”框中，输入要将项目保存到的目录，或单击“浏览”按钮以定位目录。

5）单击“确定”按钮。

Visual Studio 创建一个 Hello 新项目，并显示解决方案资源管理器。若没有显示解决方案资源管理器，请按 Ctrl+Alt+L，显示解决方案资源管理器。

6）在解决方案资源管理器中，选择解决方案下的 Hello 项目，单击鼠标右键，再将鼠标指向“添加”，然后单击“添加新项”。

7）在“类别”窗格中选择“本地项目”，然后在“模板”窗格中选择“代码文件”。在“名称”框中，键入 TESTHello。

8）单击“打开”按钮。然后在空的 TESTHello 代码文件中输入如下代码：

```
Imports  System.Windows.Forms
Class TestHello
    Shared Sub Main()
        MessageBox.Show("Hello,World!")
    End Sub
End Class
```

9）在解决方案资源管理器中，选择 Hello 项目下的引用，单击鼠标右键，然后单击“添加引用”。

10）选择.NET 选项卡，拖动滚动条，选择 System.Windows.Forms.dll，然后双击它。

11）单击“确定”按钮。

12）按 F5 键运行该应用程序，可得到如图 1-2 所示的输出。

13）关闭对话框并返回 Visual Studio。

代码分析与讨论：

1）VB.NET 程序员既要考虑自定义类，也要考虑重用框架类库（FLC）中的类。VB.NET