



新世纪高职高专计算机软件技术专业规划教材

Visual Basic. NET

邵鹏鸣
编著

程序设计



机械工业出版社
CHINA MACHINE PRESS



新世纪高职高专计算机软件技术专业规划教材

Visual Basic.NET 程序设计

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的 Visual Basic 语言。它功能强大、编程简洁、明快,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 12 章,通过大量的与现实程序设计有关的实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,注重培养学生编写实际应用程序的能力,帮助学生关注编写程序的重要环节及过程,养成良好的编程习惯,避免犯某些常见的错误。全书贯穿了面向对象编程的程序设计思想和设计方法,并用一整章篇幅讨论使用 ADO.NET 与 SQL 访问数据库的编程技术。

本书内容丰富、可操作性强、语言生动流畅、没有晦涩的术语,擅长实例说明,能够使学生在轻松、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本院校非计算机专业学生的教材,也可作为初中级读者和培训班学员学习的教材。

图书在版编目(CIP)数据

Visual Basic.NET 程序设计 / 邵鹏鸣编著. —北京:

机械工业出版社, 2004. 9

新世纪高职高专计算机专业规划教材

ISBN 7-111-15263-8

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(2004)第 094167 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:王世刚等

封面设计:解辰 责任印制:石冉

保定市印刷厂印刷·新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

1000mm×1400mm B5·16.5 印张·639 千字

定价:42.00 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) 68993821、88379646

封面无防伪标均为盗版

新世纪高职高专

计算机软件技术专业规划教材编审委员会

主任委员：何友义 番禺职业技术学院

副主任委员：（以姓氏笔划为序）

	王世刚	机械工业出版社
	贡克勤	机械工业出版社
	贺 平	番禺职业技术学院
	陈周钦	广东交通职业技术学院
	蔡昌荣	广州民航职业技术学院
	梁炳钊	广东白云职业技术学院
	刘跃南	深圳职业技术学院
	姚和芳	湖南铁道职业技术学院
委	员：	于 斌 广州民航职业技术学院
	古凌兰	广东轻工职业技术学院
	卢 奕	广西柳州市交通学校
	张 杰	湖南铁道职业技术学院
	李新燕	广州航海高等专科学校
	刘秋菊	河南济源职业技术学院
	邵鹏鸣	番禺职业技术学院
	杨小元	广州金融高等专科学校
	杨得新	广东白云职业技术学院
	杨丽娟	深圳职业技术学院
	赵从军	广东白云职业技术学院
	徐人凤	深圳职业技术学院
	柳 青	广州航海高等专科学校
	郭庚麒	广东交通职业技术学院
	翁建红	湖南铁道职业技术学院
	谢 川	杭州职业技术学院
秘	书：	王玉鑫 机械工业出版社

编写说明

党的十六大提出要走新型工业化道路,坚持以信息化带动工业化,以工业化促进信息化,加快发展现代服务业,全面建设小康社会。在推进国民经济信息化中,计算机应用、网络、软件专业人才的需求每年将在几十万人,为此教育部等六部门联合启动“制造业和现代服务业技能型紧缺人才培养培训工程”,同时教育部制定下发了“关于批准高等学校试办示范性软件职业技术学院的通知”,将计算机应用、网络及软件技术人才培养列入优先、快速和重点发展的地位。探索新的软件人才培养培训模式,把提高培养者的职业能力放在突出的位置,以应用为目的,构建就业导向的课程体系,坚持能力本位的课程设计原则,组织开发和编写具有鲜明特色的教材,是实施“工程”的目标任务之一。

根据上述高等职业教育的方针和软件人才培养的目标,本套教材编审委员会与机械工业出版社积极组织IT办学水平较高、教学改革成效显著的高职院校的计算机应用、网络及软件技术专业方面的学科带头人和教学骨干,开展产业人才需求调研、研讨人才培养模式、构建课程体系与教材开发等一系列工作。

在课程体系的构建中,注重对当前产业应用的主流IT技术清楚的认识,对IT企业人才需求全面的了解和对IT技术发展的透彻的理解及预见性的把握。同时在教材中突出以实践为主的原则,通过理论讲授、上机练习、案例教学、实际项目演练及企业实习等多种形式的教学内容介绍,强化技能训练,达到职业及专业能力培养。

本套教材体现了教学内容紧密结合专业核心能力对理论知识的要求,形成了有技术应用特点的理论知识体系,构成技术运用理论基础,满足了培养对象的需求。同时,注重融入信息技术的最新发展,更新内容,介绍新知识、新技术、新流程和新方法,把握主流技术和成熟技术的运用,实现专业教学基础性与先进性的统一。

本套教材还具有连贯性和递进性的特点,在实验、实训、实习、项目训练、工程训练的内容安排上力求具有新的特色,能反映专业岗位的工作需求,并成为软件人才成长的一套科学性、系统性、实用性较好的软件技术教育培训教材。

新世纪高职高专计算机软件技术专业规划教材编审委员会

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, 使教授的知识是市场上最需要的知识。

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 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, .NET 有着广阔的应用前景。

我们知道 C 语言是一种代码效率很高但不易进行快速开发的程序设计语言。自从 OOP 出现后, OOP 与 C 语言融合产生了 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。基本上还只是占据着 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, 相信会深深的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。

另一方面, 虽然 C 语言和 C++ 为我们带来高度的灵活性, 但我们必须要忍受学习的艰苦和开发的长期性。特别对 VC++ 来说, 大部分的程序结构都被封装在 MFC 中。所以, 对于初学者来说, 程序结构显得十分混乱, 学习将变得十分艰苦。而且自从出现了 VC++2.0 以后, 为了适应不断更新的技术 (例如 COM、ATL 等), 又要与前一个版本兼容, VC++ 在此之后的每一次升级都给 VC++ 程序员带来一份痛苦——程序结构变得越来越复杂, 而且出现了越来越多的变量类型, 从而带来了更多的问题。

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, 而且具有 C++ 的基本特征——面向对象, Visual Basic .NET 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(面向对象的编程方法)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 Internet 和 WWW 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。

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,无论是没有任何编程经验的读者,还是具有编程经验的读者,做到开卷有益,重要的是能够激发读者的学习兴趣。

本书从应用与工程实践的角度出发进行组织编写,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、编写教材的思路、教学的模式及理念等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,顺应并符合新世纪教学发展的规律,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) 本书不是采用传统的提出概念—解释概念—举例说明的方法,而是以实例为主线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) 使学生从一开始就编写有用的程序,这有助于保持学生的学习兴趣及动力。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是大家感兴趣的内容。实例包含的知识点可能跨越若干章节。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进知识,具有很强的实用性。

(6) 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,而是教会学生如何用 VB.NET 开发程序,书中的每一个实例都说明了开发过程,通过学习如何开发程序来掌握语言要素、语法。

(7) 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世界的程序设计。学生仅仅知道面向对象的概念是不够的,必须能够运用

这些知识开发现实世界的程序。我们教学法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，在有效地设计自己的类之前，必须首先学会使用预定义的类，学生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) 我们要求学生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，我们认为这是学生学习程序设计最稳妥、最有效、最快捷的途径。我们不鼓励学生只是简单遵循书中介绍的方法步骤，也不鼓励学生只阅读源代码。

我们已努力使本书没有错误，但是由于水平和时间的关系，错误仍然在所难免。如果发现错误（如拼写错误等），欢迎大家批评指正，也欢迎大家提出宝贵意见，并将信息反馈给我们，我们将不胜感激。

作者

目 录

编写说明

前言

第 1 章 认识 VB.NET.....1

1.1 第一个简单的控制台应用程序.....1

1.2 创建简单的 Windows 应用程序.....3

习题.....11

第 2 章 变量与数据类型.....12

2.1 变量与常数.....12

2.2 基本数据类型.....19

习题.....37

第 3 章 程序流控制.....39

3.1 选择语句.....39

3.2 循环语句.....58

3.3 跳转语句.....69

习题.....75

第 4 章 方法与数组.....77

4.1 方法.....77

4.2 数组.....84

4.3 传递参数——传值方式和 传引用方式.....101

4.4 多维数组.....106

习题.....115

第 5 章 类与对象.....118

5.1 类、对象、字段、属性和 方法.....119

5.2 实例构造函数.....136

5.3 实例构造函数重载.....144

5.4 对象成员与类的成员.....149

5.5 对象参数与返回值为对象.....153

5.6 方法的重载.....166

习题.....173

第 6 章 继承.....175

6.1 直接基类与派生类.....175

6.2 派生类实例构造函数声明.....181

6.3 隐藏从基类继承的成员.....187

6.4 含直接基类构造函数的 构造函数声明.....189

6.5 Overridable 方法与重写方法.....191

习题.....203

第 7 章 多态性.....204

7.1 抽象方法与抽象类.....204

7.2 接口.....220

7.3 委托.....233

7.4 事件.....245

习题.....253

第 8 章 常用控件.....255

8.1 滚动条控件.....255

8.2 PictureBox 图片框控件.....261

8.3 文本框控件与按钮控件.....270

8.4 Windows 窗体事件及 事件处理程序.....280

8.5 复选框和单选按钮.....287

8.6 列表框.....291

8.7 带复选框的列表框.....300

习题.....304

第 9 章 GDI+图形.....307

9.1 第一个绘图程序.....308

9.2 创建 Graphics 对象.....309

9.3 笔、画笔和颜色.....314

9.4 绘制线条和形状.....322

9.5 用 GDI+呈现图像.....343

9.6 画点.....347

9.7 坐标系统	348	11.3 序列化对象	422
9.8 全局变形可与局部变形合并	358	11.4 顺序访问文件	426
习题	360	11.5 随机访问文件	442
第 10 章 与用户交互	363	习题	466
10.1 菜单	363	第 12 章 使用 ADO.NET 进行数据	
10.2 鼠标事件	371	库编程	467
10.3 键盘事件处理	377	12.1 Connection 和 Command	
10.4 通用对话框	385	对象	467
10.5 编写多文档界面应用程序	393	12.2 使用 ADO.NET DataReader	474
习题	406	12.3 使用 DataAdapter、DataTables	
第 11 章 用流进行文件输入和		和 DataSets	493
输出	408	12.4 修改数据	504
11.1 文件与流	408	习题	513
11.2 二进制读取器和写出器	418	参考文献	515

第1章 认识 VB.NET

本章实例:

- 打印一行文字。
- 在对话框中显示一行文字。
- 在文本框中显示一行文字。

通过完成实例及对实例的代码进行分析与讨论, 可达到如下目标:

- 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 第一个简单的控制台应用程序

实例: 打印一行文字 (图 1-1)。

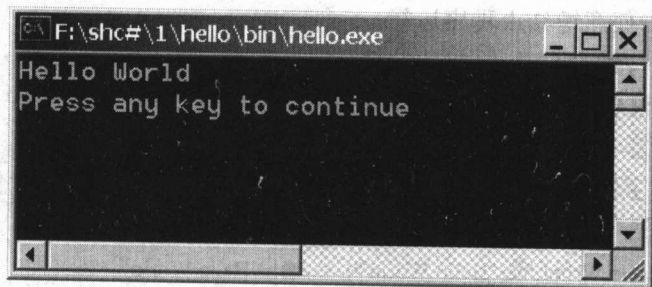


图 1-1 打印一行文字

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”。

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，用鼠标选择解决方案下的 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 ")
```

```
    End Sub
```

```
End Class
```

(9) 在解决方案资源管理器，用鼠标选择解决方案下的 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!”(如图 1-1 所示)。

(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。

(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。

用“End Sub”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, 我们通常将双引号之间的字符称为字符串。使用了 WriteLine 方法, 它是类库中 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) 从 IDE 编译并运行程序。按 F5 来生成并运行 (对应于“调试”菜单中的“启动”)。

1.2 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 (图 1-2)。



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”。

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，用鼠标选择解决方案下的 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：

```
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，用鼠标选择 TestMessageBox 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 程序中要使用类库中的类,必须添加包含要使用的类库的程序集的引用。类库中的每个类都从属于特定的命名空间,我们使用 Imports 语句定位所使用类库中的类的命名空间,以便我们在程序中直接使用该类。例如,要使用类库中的 MessageBox 类,首先必须添加包含 MessageBox 类的类库的程序集(动态链接库)“System.Windows.Forms.dll”的引用,由于 MessageBox 类在命名空间“System.Windows.Forms”中,我们在程序中使用 Imports 语句:

```
Imports System.Windows.Forms
```

定位 MessageBox 类的命名空间,这样在我们的程序中可直接使用 MessageBox 类,如:

```
MessageBox.Show("Hello,World!");
```

而不需要使用完全限定名:

```
System.Windows.Forms.MessageBox.Show("Hello,World!")
```

否则会发生编译错误: MessageBox 未定义。

(2) 在 Main 方法中,语句:

```
MessageBox.Show("Hello,World!")
```

表明一个对类 MessageBox 中的方法 Show 的调用,它的作用是在对话框中显示一字符串。方法 Show 是 MessageBox 类中的一个特殊的方法,称为静态方法。一般通过在类名称后加上点操作符 (.) 和方法名称来调用它们。在第 5 章“类与对象”将讨论静态方法。

实例: 在文本框中显示一行文字。

创建一 Windows 应用程序 (图 1-3), 该程序的功能为: 当单击窗体上的“显示”按钮时, 在窗体上文本框中显示“Hello, World”; 当单击窗体上的“清除”按钮时, 清除窗体上文本框中的内容。



图 1-3 在文本框中显示一行文字

实现步骤

1. 创建项目和窗体

(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 项目”，然后在“模板”窗格中选择“Windows 应用程序”。

注意：如果已经打开了解决方案，请选择“关闭解决方案”。在生产项目中，经常会在同一个解决方案中有多个项目。但是在本示例中，您将关闭任何已打开的解决方案，并与项目一起创建一个新的解决方案，使在此执行的操作与任何现有窗体、数据集等之间不存在任何冲突。

(4) 在“名称”框中，键入 `Hellocs` 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”。

Visual Studio 创建一个新项目，并在 Windows 窗体设计器中显示新窗体。

(6) 从“工具箱”的“Windows 窗体”选项卡中，首先单击 **Button** 控件并将其拖到窗体上，再单击 **TextBox** 控件并将其拖到窗体上。按此方法再向窗体上添加一个 **Button** 控件，并用鼠标将它们拖到适当的位置并调整其大小（单击要调整大小的控件并拖动八个尺寸柄中的一个可调整其控件大小），如图 1-4 所示。

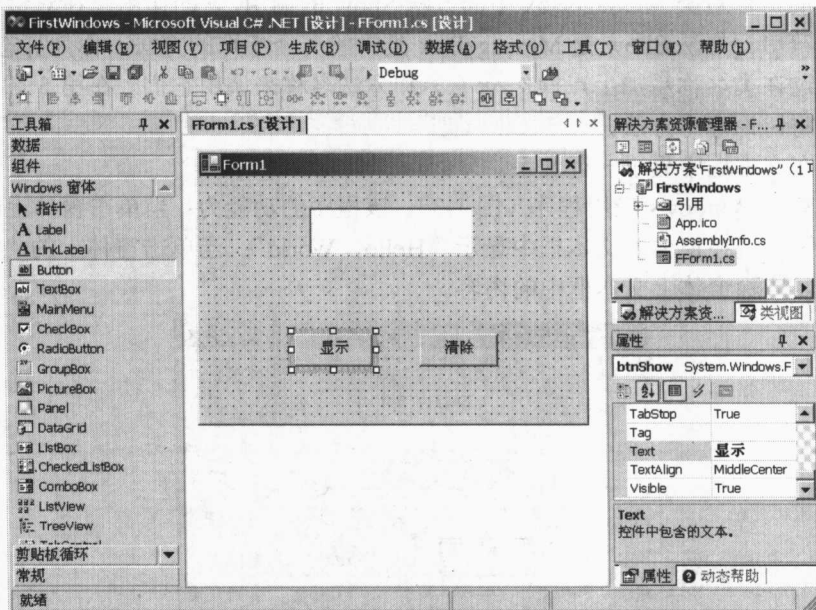


图 1-4 VB.NET 集成开发环境

(7) 按 **F4** 键以显示“属性”窗口。

(8) 在 Windows 窗体设计器中, 单击 **Button1** 按钮将其选定。在“属性”窗口中, 将该按钮 **Name** 属性设置为 **btnShow**, 将 **Text** 属性设置为“显示”。单击 **Button2** 按钮将其选定, 将该按钮 **Name** 属性设置为 **btnClear**, 将 **Text** 属性设置为“清除”。

(9) 在 Windows 窗体设计器中, 单击 **textBox** 控件将其选定。在“属性”窗口中, 将其 **Name** 属性设置为 **textB**, 将 **Text** 属性设置为空。选择 **Multiline** 属性, 这时 **Multiline** 属性条目的右边出现一个下箭头图标按钮, 单击该按钮, 出现下拉列表框, 在出现下拉列表框中选择 **True**, 在“属性”窗口中, 选择 **TextAlign** 属性, 单击其右边的按钮, 选择 **Center**。

2. 编写应用程序的代码

(1) 双击“显示”按钮, 为“显示”按钮的 **Click** 事件添加事件处理程序。此时将打开代码编辑器, 鼠标指针已位于事件处理程序中。

(2) 插入下列代码

```
Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnShow.Click
        textB.Text="Hello, World"
End Sub
```

(3) 双击“清除”按钮, 为“清除”按钮 **Click** 事件添加事件处理程序。此时将打开代码编辑器, 鼠标指针已位于事件处理程序中。

(4) 插入下列代码

```
Private Sub btnClear_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnClear.Click
        textB.Text=""
End Sub
```

3. 测试应用程序

(1) 按 **F5** 键运行该应用程序。

(2) 当运行应用程序时, 单击“显示”按钮并验证已显示“Hello, World!”。

(3) 单击“清除”按钮验证已显示的“Hello, World!”从文本框中清除。

(4) 关闭此 Windows 窗体并返回 Visual Studio。

代码分析与讨论

(1) 当单击“显示”按钮, “显示”按钮发生 **Click** 事件, 执行“显示”按钮的 **Click** 事件的事件处理程序 **btnShow_Click**:

```
Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnShow.Click
```