



高职高专“十一五”计算机类专业规划教材

Visual Basic.NET 程序设计

白德淳 主编

Visual Basic.NET
CHENGXU SHEJI



机械工业出版社
CHINA MACHINE PRESS

赠电子课件



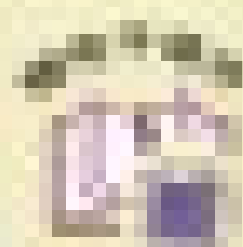


Microsoft “4+X” 认证系列教材

Visual Basic.NET 程序设计

第2版 王爽 著

Visual Basic .NET 程序设计



清华大学出版社

高职高专“十一五”计算机类专业规划教材

Visual Basic. NET 程序设计

主 编 白德淳

副主编 齐彦力

参 编 罗大伟 温春友 桑莉君



机械工业出版社

本书结合高职教育的特点,围绕 VB. NET 的各项应用技术,系统地介绍了 VB. NET 的基本概念、技术内容和实践方法。各内容单元均以验证为目的,以相对完整的技术组合为背景,给出应用实例,详细介绍实践步骤和方法,并注有关键性说明。各实践内容除与讲述内容吻合外还具有承上启下的特点,具有一致的技术风格。全书从问题实例入手,通过一般性方法讲解相关的技术内容 and 应用意义,强调技术构成和应用标准;精讲相关的概念和理论方法,通过总结的方式完成必要的理论升华。

本书实践性强,各章配有习题,主要章节附有实践与指导内容,为实践教学打下良好的基础和提供充足的可选内容空间。同时也考虑到在实际教学中的内容重组需求,故此在内容编排上特别注意保持各单元的相对独立性和总体的完整性以及对基本内容的覆盖方式。

本书可作为高职院校软件技术类专业的教材,也可作为相应的培训教材。

本教材配有电子教案,可登录机械工业出版社教材服务网 www.cmpedu.com 下载,或发送电子邮件至 cmpgaozhi@sina.com 索取。咨询电话:010-88379375。

图书在版编目(CIP)数据

Visual Basic. NET 程序设计/白德淳主编. —北京:机械工业出版社, 2007. 10

高职高专“十一五”计算机类专业规划教材

ISBN 978 - 7 - 111 - 22538 - 6

I. V… II. 白… III. BASIC 语言 - 程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 157797 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 王玉鑫

责任编辑: 张 芳 责任校对: 魏俊云

封面设计: 马精明 责任印制: 邓 博

北京京丰印刷厂印刷

2008 年 1 月第 1 版·第 1 次印刷

184mm × 260mm · 19 印张 · 468 千字

0 001—4 000 册

标准书号: ISBN 978 - 7 - 111 - 22538 - 6

定价: 28.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 68354423

封面无防伪标均为盗版

前言

VB.NET 是 Microsoft 公司集成在 Visual Studio.NET 开发平台中的新一代程序设计语言,也是当今较为流行的软件开发工具之一。本书以应用能力培养为出发点,围绕 VB.NET 的各项应用技术,系统地介绍其基本概念、技术内容和实践方法。通过详尽的实现过程描述,使学生能够由浅入深地逐步掌握基本的程序设计技能。

全书分为 9 章,第 1 章介绍 Visual Studio.NET 的家族产品及 VB.NET 的基本概念、技术背景、特征以及相应开发环境的应用;第 2 章介绍 VB.NET 的基本语法格式、流程控制结构、编程技术以及在开发环境中的工程方法;第 3、4 章介绍界面控件的应用技术;第 5 章介绍面向对象的编程技术以及应用程序结构的组织;第 6 章介绍窗体与对话框的应用、界面布局方法;第 7 章介绍数据流与文件的概念及典型实用技术;第 8 章介绍基于 ADO.NET 的数据库访问技术与程序设计方法;第 9 章介绍基本的 Web 应用程序开发方法。各章均列出了学习重点,配有习题,主要章节附有实践与指导内容。

本书充分考虑到高职学生能力系统化培养的基本要求,可作为软件技术类专业的相应教材。本书主要针对 VB.NET 的程序设计方法,全过程注重典型方法与技巧的介绍,追求工程背景。课程目标旨在为后续软件项目教学提供技术支持。

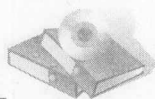
本书由白德淳任主编,齐彦力任副主编,参编有罗大伟、温春友、桑莉君。在本书编写过程中,曾得到诸多同行的支持,从中吸取了许多有益借鉴。在此谨表谢意。

由于水平和时间的关系,书中难免有错误及不足之处,欢迎读者提出宝贵意见(jl_bdc@126.com)。

编 者

目 录

前言	2.3.2 选择结构	34
第1章 Visual Studio. NET 与 VB. NET	2.3.3 循环结构	39
1.1 Visual Basic. NET 简介	2.3.4 工程实例	43
1.1.1 Visual Basic. NET	2.4 数组	44
1.1.2 Microsoft. NET	2.4.1 一维数组	44
1.2 Visual Basic. NET 的安装	2.4.2 二维数组	47
1.3 Visual Studio. NET IDE	2.4.3 动态数组	48
1.3.1 Visual Studio. NET IDE 概述	2.4.4 工程实例	49
1.3.2 主菜单与工具栏	2.5 过程	52
1.3.3 Visual Studio. NET IDE 窗口	2.5.1 过程的基本概念	52
1.3.4 使用 Visual Studio. NET 的帮助	2.5.2 Sub 过程	53
1.4 应用 Visual Studio. NET IDE 编写 VB. NET 程序	2.5.3 函数过程	55
1.4.1 编写第一个 VB. NET 应用程序	2.5.4 参数传递	56
1.4.2 编写第二个 VB. NET 应用程序	2.5.5 变量的作用域	60
1.4.3 Visual Studio. NET 的解决方案和项目	2.5.6 过程重载	61
本章小结	2.5.7 工程实例	63
习题	2.6 实践与指导	68
第2章 VB. NET 的编程基础	本章小结	70
2.1 数据类型、变量、常量	习题	70
2.1.1 值类型	第3章 简单界面控件与应用	72
2.1.2 变量	3.1 界面控件的应用概述	72
2.2 系统函数	3.2 简单界面控件	74
2.2.1 数学处理函数	3.2.1 Label 控件	74
2.2.2 字符处理函数	3.2.2 TextBox 控件	77
2.2.3 日期时间函数	3.2.3 Button 控件	78
2.2.4 随机数函数	3.2.4 CheckBox、RadioButton 和 GroupBox 控件	79
2.3 程序控制结构	3.2.5 ComboBox 控件	81
2.3.1 程序控制结构的基本概念	3.2.6 ListBox 与 CheckedListBox 控件	83
	3.2.7 PictureBox 控件	87
	3.2.8 Timer 控件	89
	3.3 简单界面控件的工程实例	90
	3.3.1 实例1	90
	3.3.2 实例2	93



3.4 实践与指导	94	6.2 对话框	169
本章小结	96	6.2.1 简单对话框	169
习题	96	6.2.2 通用对话框	170
第4章 高级界面控件与应用	97	6.2.3 对话框的应用实例	175
4.1 高级界面控件	97	6.3 菜单	176
4.1.1 ImageList 控件	97	6.3.1 菜单、菜单项对象	177
4.1.2 ListView 控件	98	6.3.2 菜单应用实例	178
4.1.3 TreeView 控件	105	6.3.3 快捷菜单	179
4.1.4 TabControl 控件	111	6.4 工具栏	180
4.1.5 DateTimePicker 控件	113	6.5 状态栏	184
4.1.6 Panel 控件	115	6.6 实践与指导	186
4.1.7 Splitter 控件	116	本章小结	187
4.2 高级界面控件的工程实例	117	习题	187
4.3 实践与指导	121	第7章 流与文件	189
本章小结	122	7.1 流与文件概述	189
习题	122	7.1.1 流的基本概念	189
第5章 面向对象编程技术	123	7.1.2 System.IO 命名空间	190
5.1 面向对象的基本概念	123	7.2 应用流操作文件	191
5.1.1 类与对象	123	7.2.1 FileStream 流与 File 类	191
5.1.2 字段与属性	125	7.2.2 应用 StreamReader 与 StreamWriter 操作文件	195
5.1.3 类的方法	127	7.2.3 应用 BinaryReader 与 BinaryWriter 操作文件	198
5.1.4 构造函数与析构函数	128	7.3 文件管理操作	201
5.2 面向对象的应用技术	130	7.3.1 文件夹管理	201
5.2.1 创建类与应用对象	130	7.3.2 文件操作	205
5.2.2 继承	135	7.4 工程实例	209
5.2.3 多态	138	7.5 实践与指导	213
5.2.4 共享成员	139	本章小结	214
5.2.5 应用实例	141	习题	214
5.3 接口	146	第8章 ADO.NET 数据库访问 技术	216
5.3.1 接口的基本概念	146	8.1 ADO.NET 概述	216
5.3.2 接口的应用实例	148	8.2 .NET 数据提供者	218
5.4 结构	151	8.2.1 SQL Server.NET 与 OLEDB.NET 数据提供者	218
5.5 异常和异常处理	153	8.2.2 应用 .NET 数据提供者访问 数据库的实例	222
5.6 面向对象技术的工程实例	155	8.3 数据集	226
5.7 实践与指导	160	8.3.1 DataSet 层次结构	226
本章小结	162	8.3.2 应用 DataSet 访问数据库	229
习题	162		
第6章 应用程序的界面设计	164		
6.1 窗体类与窗体设计	164		
6.1.1 窗体的属性、事件与方法	164		
6.1.2 窗体的界面控件管理	167		



8.4 应用 DataSet 与 DataGrid	237
8.4.1 DataGrid 显示风格	237
8.4.2 应用 DataView	241
8.4.3 应用 DataGrid 与 DataTable 的 事件	244
8.4.4 简单数据绑定	246
8.5 应用 Command 对象	248
8.5.1 应用 SqlCommand 执行存储 过程	248
8.5.2 应用 Command 的“执行” 方法	250
8.5.3 为 DataAdapter 配置 Command 型属性	251
8.6 工程实例	254
8.7 实践与指导	258
本章小结	260
习题	260
第9章 Web 应用程序开发基础	263

9.1 ASP.NET 应用程序	263
9.1.1 ASP.NET 概述	263
9.1.2 ASP.NET 的运行环境	264
9.1.3 创建 Web 应用程序项目	267
9.2 Web 窗体编程	271
9.2.1 Web 窗体的生存期	271
9.2.2 Web 应用程序中的常用对象	273
9.2.3 Web 窗体的控件	274
9.2.4 Web 窗体的数据保持	278
9.2.5 Web 窗体的应用实例	280
9.3 开发 Web 环境下的数据库 应用程序	283
9.3.1 应用 ADO.NET 读取数据库	283
9.3.2 应用 ADO.NET 更新数据库	288
本章小结	295
习题	295
参考文献	296

第 1 章 Visual Studio. NET 与 VB. NET

Visual Basic. NET (简称 VB. NET) 是微软推出的基于 .NET 开发平台的新一代编程语言, 可用于生成 ASP Web、XML Web Services、Windows 应用程序和移动应用程序。

学习重点:

- 1) Microsoft. NET 的基本构成, .NET 兼容语言程序在 .NET 环境下的编译、执行特点。
- 2) Visual Studio. NET 安装步骤与方法。
- 3) Visual Studio. NET 的帮助机制与获取帮助信息的基本方法。
- 4) IDE 的基本功能、创建应用程序的基本方法。
- 5) 解决方案、项目以及工程文件的基本概念和应用方法。

1.1 Visual Basic. NET 简介

Visual Basic. NET 真正具备了面向对象 (OOP) 编程语言的所有特征, 并可与 Visual C++ .NET、Visual C# .NET 和 Visual J# .NET 共同基于 Visual Studio. NET 集成开发环境, 构成兼容 .NET 平台的开发语言体系。

1.1.1 Visual Basic. NET

Visual Basic 曾经是世界上应用最为广泛的快速开发语言之一。在 20 世纪 90 年代后期, 随着可编程设备, 如个人数字助理 (PDA)、手机以及其他移动电子设备的流行, 这些组件与其他设备通过 Internet 进行信息交互的需求日益迫切, 使得软件开发人员必须面临 Internet 环境, 开发供任何人从任何类型设备上都能实现相互访问或进行信息交换的软件。为适应这种需要, 微软公司于 2000 年推出 Visual Studio. NET 集成开发平台, 这是一种全新的编程模式。它包括一个强大的组件库, 可使程序员更快地开发应用程序。Visual Basic. NET 完全集成在其他 Visual Studio. NET 语言中, 不仅可以通过不同的语言开发组件, 还可以通过交叉语言继承, 在一种语言编写的类中派生另一种语言编写的类。

应用 Visual Basic. NET 能够很容易地开发分布式应用, 如基于 Web 环境的应用程序。Visual Basic. NET 程序可以通过互联网交互使用行业标准, 如简单对象访问协议 (Simple Object Access Protocol, SOAP) 和可扩展标记语言 (Extensible Markup Language, XML) 等。

1.1.2 Microsoft. NET

Microsoft. NET 能够帮助程序员创建 Web 服务和 Web 应用程序, 允许应用程序通过互联网进行通信和共享数据, 并且与所采用的操作系统、设备或编程语言无关。.NET 策略的主要内容之一就是构建独立于任何语言的平台, 使开发人员可以应用 .NET 兼容语言体系中的任何一种语言来集体开发应用程序。



.NET Framework 有两个主要成员，一是公共语言运行库或称通用语言运行时环境 (Common Language Runtime, CLR)，二是 .NET Framework 类库 (Framework Class Library, FCL)。

1. 公共语言运行库

公共语言运行库是 .NET Framework 的基础，它负责提供代码管理，包括处理加载程序、运行程序的代码以及提供所有支持服务的代码。同时还强制性地实施类型安全检查，事实上，CLR 在应用程序的开发阶段与运行阶段都起到了重要作用。

.NET 兼容语言程序的编译、运行是一个全新的过程，需要经过两次编译方可变成可执行的机器指令代码。

(1) 编译 源程序首先被编译成微软中间语言 (Microsoft Intermediate Language, MSIL 或 IL) 代码。它是一种不依赖于任何 CPU 指令体系结构的汇编语言，可以快速地被转换成内部机器代码 (由 JIT 编译)。编译好的 MSIL 形式的软件包由多个叫做“装配件”的代码单元组成，它包含有中间语言代码，其中的一个装配件被指定为可执行程序，具有主程序的启动入口，其他装配件被组织成“库”，全部过程由 CLR 管理完成。

(2) 执行 在执行应用程序时，CLR 首先载入中间语言代码，并将其编译成适合当前运行环境的机器代码。第一个被加载的是含有程序入口的主程序装配件，经过编译后运行它，当需要调用其他方法时，再载入其他的装配件并编译、运行。当装配件或方法第一次被调用执行后，就被缓存起来，以便再次调用。每一个方法只被编译一次，方法一经编译，它的入口地址就被编译好的可执行代码的地址代替。如果一个方法从来没有被调用过，则与之对应的中间语言代码就不会被编译器转换成机器代码。由于中间语言代码只在需要的时候才被编译，使得应用程序获得了较高的执行效率。这一过程就是通常所说的即时编译 (Just In Time, JIT)。在这期间，CLR 会检查代码的类型和安全性。例如，运行代码时，CLR 会监视内存的使用，确保编译器不会访问无效的代码内存地址。

代码管理是 CLR 的基本原则。在 .NET 环境下运行或被 CLR 管理的任何代码都被称为托管代码，外部的其他代码也可运行在 Windows 环境上，这些代码被称为非托管代码。

使用中间语言和 JIT 编译有许多好处，首先它能够生成独立于任何操作系统平台的代码。这种代码形式只包含对 .NET 核心类型和基本类库中函数的调用，即使是在 Windows 平台下开发的应用程序，也不包含对 Windows API 和 Intel CPU 基本函数的调用。另一方面，CLR 的 JIT 编译器可将中间语言代码编译成符合运行环境的执行代码。这就说明，任何操作系统平台只要嵌入了合适的 .NET Framework，它就能支持中间语言运行。

CLR 还有两个重要的成员，一个是通用语言规范 (CLS)，任何 .NET 的兼容语言都遵守这个规范，这就决定了它们之间可以互操作、继承和扩展其他语言。另一个是通用类型系统 (CTS)，它规范了 .NET Framework 的基本数据类型，并且规定类型必须继承自 .NET Framework 基类库中的一个成员类型。CTS 支持直接包含数据的值类型和包含数据实际内存地址的引用类型。

2. .NET Framework 类库

.NET Framework 类库是一个综合性的、面向对象的类型集合，封装了大量的系统对象和功能，任何 .NET 语言都可以使用它。这为开发人员提供了统一的、面向对象的、分层且可扩展的类库集 (API)，也为程序员提供了对类库的访问机制，程序员可以使用它开发各种



应用程序。例如，可以应用类库中的窗体和其他界面控件（文本框、命令按钮、下拉列表框等）编写 Windows 应用程序，可以应用类库中基本或扩展的 Web 控件编写 ASP. NET 应用程序，也可以应用 ADO. NET 编写数据库应用程序。

3. .NET Framework 的应用结构

图 1-1 简单描述了 .NET Framework 的基本应用结构。.NET Framework 基于 Windows 操作系统，它将 .NET 应用程序与操作系统隔离开。.NET Framework 类库提供了适用于各种平台的界面技术、组件技术和数据技术，可用于开发各种应用程序，也可以向基于任何操作系统或平台的任何类型的用户端提供 Web 服务。.NET Framework 不仅提供了通往外部世界的接口，而且凭借 CLR 功能，使得 .NET 应用程序的开发和运行可以不依赖于任何类型的操作系统和 CPU 设备。

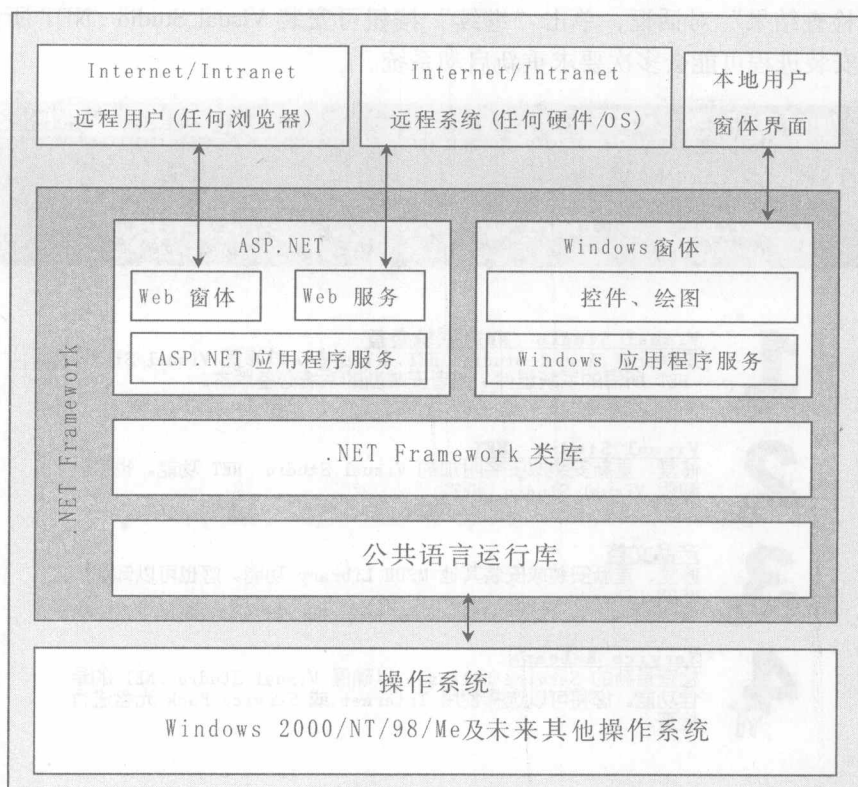


图 1-1 .NET Framework 的应用结构

公共语言运行库完成源程序的编译与执行过程的代码管理，监视内存的使用，并在适当的时候调用“无用存储单元收集器”，检查程序中的变量，释放不再使用的内存单元。

1.2 Visual Basic. NET 的安装

Visual Studio. NET 产品通常由 Windows 系统必备组件、Visual Studio. NET 和产品帮助文档 (MSDN) 3 部分组成。其中 Visual Studio. NET 包含 Visual Basic. NET 及其他 .NET 家族产



品。

1. Visual Studio .NET 产品安装

安装过程分为三步进行,启动安装程序后,系统首先默认的安装位置是启动系统的驱动器。用户可以更改这个位置,但不管安装在什么位置,安装进程都会在系统驱动器上放置一些必要的文件。所以,应确保系统驱动器有一定的存储空间。

(1) 安装 Visual Studio .NET 的系统必备盘 将标有“Visual Studio .NET”的 1 号盘插入光驱,执行根目录下的“Setup.exe”可打开如图 1-2 所示的“Visual Studio .NET 安装程序”对话框。用鼠标单击“Visual Studio .NET 系统必备”链接,安装程序会提示将系统必备盘放入驱动器,单击“确定”按钮,系统会打开“安装系统必备”对话框,在这个界面上,单击“同意”选项后,再单击“继续”按钮,这时系统检查需更新的组件内容,随后打开一个“检查结果”对话框,单击“继续”按钮可安装 Visual Studio .NET 所需的组件。在此期间,安装进程可能会多次要求重新启动系统。

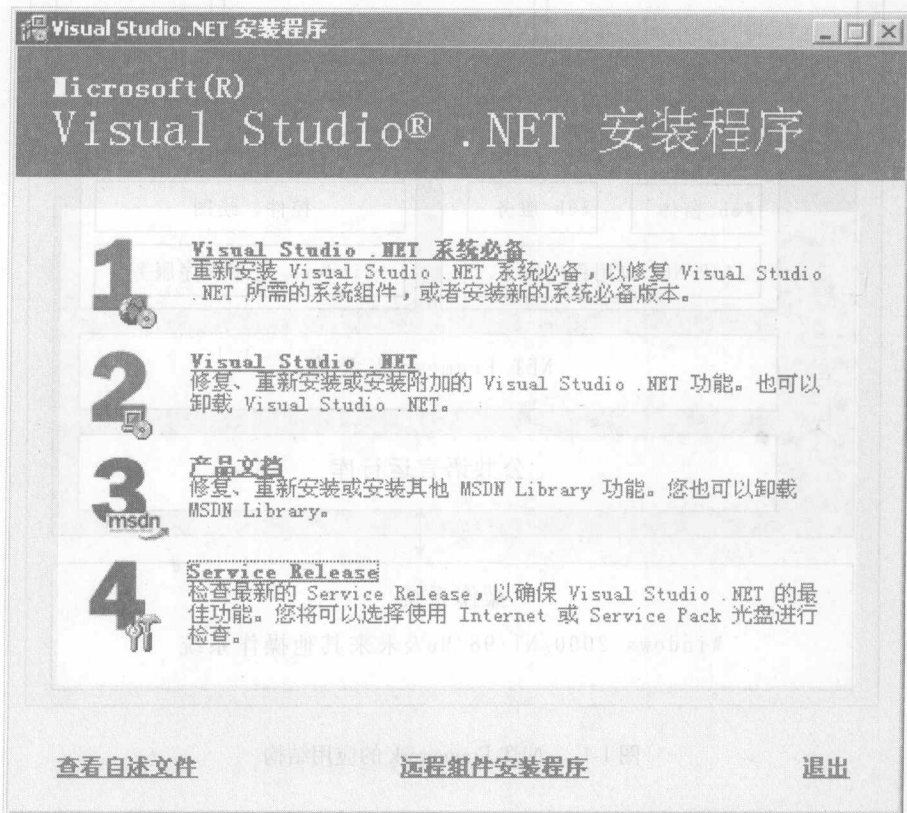


图 1-2 “Visual Studio .NET 安装程序”对话框

(2) 安装 Visual Studio .NET 组件更新结束之后,用鼠标单击“Visual Studio .NET”链接,安装进程提示重新放置 1 号盘。放入 1 号盘后会打开如图 1-3 所示的“Visual Studio .NET 安装程序-起始页”对话框,按提示内容操作后,安装进程打开如图 1-4 所示的“Visual Studio .NET 安装程序-选项页”对话框。页面左侧是一个树视图,可在此选择需要安装的项目,然后单击“立即安装”按钮。这期间系统会提示放置 2 号盘。

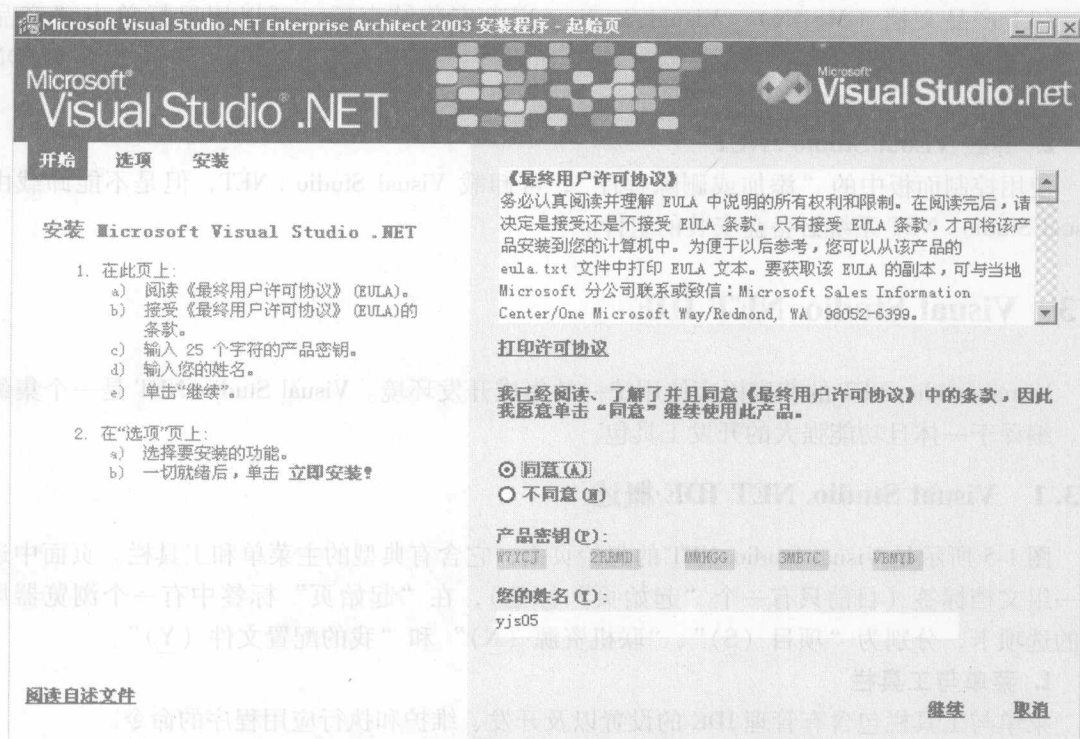
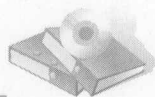


图 1-3 “Visual Studio .NET 安装程序-起始页”对话框

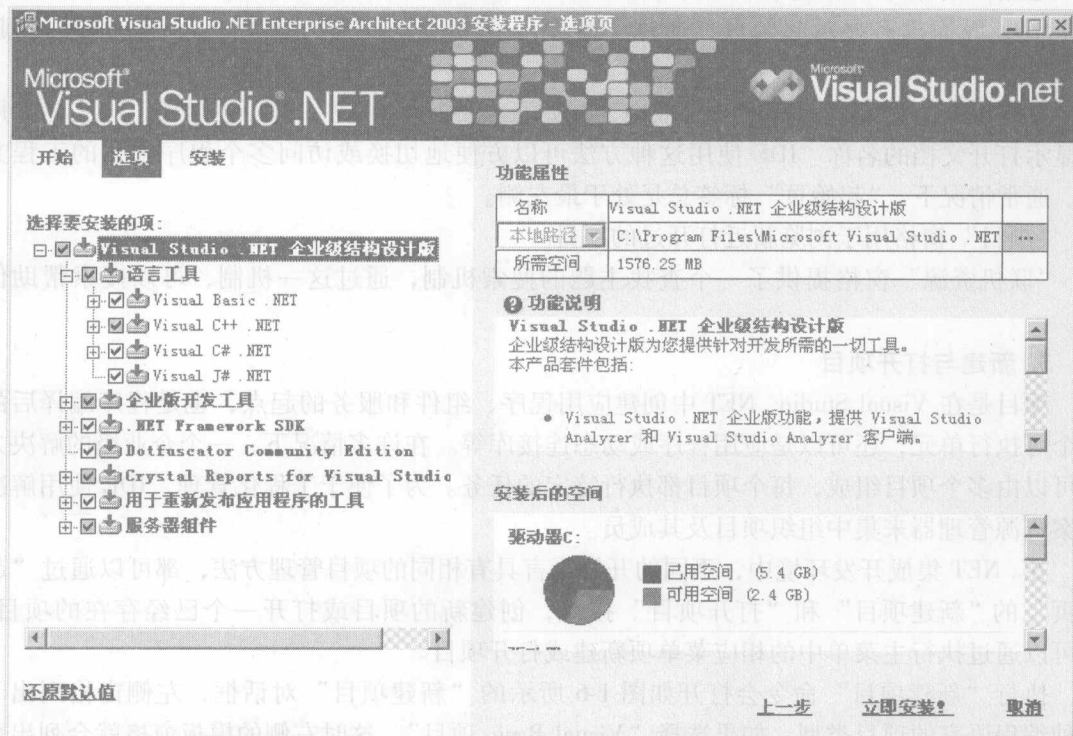
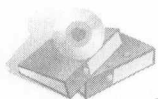


图 1-4 “Visual Studio .NET 安装程序-选项页”对话框



(3) 产品文档 (MSDN) Visual Studio .NET 安装结束后, 可以用鼠标单击“产品文档”链接, 打开“产品文档安装向导”对话框, 按提示内容操作可连续安装 MSDN 的 3 张盘。

2. 卸载 Visual Studio .NET

使用控制面板中的“添加或删除程序”, 可卸载 Visual Studio .NET, 但是不能卸载由 Visual Studio .NET 系统必备盘安装的组件。

1.3 Visual Studio. NET IDE

Visual Studio. NET 的兼容语言使用统一的集成开发环境。Visual Studio. NET 是一个集编辑、编译于一体且功能强大的开发工具包。

1.3.1 Visual Studio. NET IDE 概述

图 1-5 所示是 Visual Studio. NET 的起始页面, 它含有典型的主菜单和工具栏。页面中还有一组文档标签 (目前只有一个“起始页”标签), 在“起始页”标签中有一个浏览器风格的选项卡, 分别为“项目 (S)”、“联机资源 (N)”和“我的配置文件 (Y)”。

1. 菜单与工具栏

菜单与工具栏包含有管理 IDE 的设置以及开发、维护和执行应用程序的命令。

2. 起始页与标签

起始页有 3 个窗格选项, 用于设置首选项、阅读产品新闻、联系开发伙伴等操作。如果 IDE 界面没有显示起始页, 通过“帮助”菜单上的“显示起始页”选项可打开起始页。

在应用 IDE 开发应用程序的过程中, 起始页右侧还会出现一组其他标签, 这些标签用于显示打开文档的名称。IDE 使用这种方法可以方便地切换或访问多个程序项目的工程文件。通常情况下, “起始页”标签总是处于最左侧。

“项目”窗格用于浏览新近打开过的项目。

“联机资源”窗格提供了一个查找主题的搜索机制, 通过这一机制, 可以搜索帮助信息。

3. 新建与打开项目

项目是在 Visual Studio .NET 中创建应用程序、组件和服务的起点, 也是程序编译后的一个可执行单元, 还可以是应用程序或动态连接库等。在许多情况下, 一个企业级的解决方案可以由多个项目组成, 每个项目都执行特定的任务。为了便于工程化管理, IDE 使用解决方案资源管理器来集中组织项目及其成员。

在 .NET 集成开发环境中, 不同的开发语言具有相同的项目管理方法, 都可以通过“起始页”的“新建项目”和“打开项目”按钮, 创建新的项目或打开一个已经存在的项目, 也可以通过执行主菜单中的相应菜单项新建或打开项目。

执行“新建项目”命令会打开如图 1-6 所示的“新建项目”对话框, 左侧窗格列出了各种编程语言的项目类型。如果选择“Visual Basic 项目”, 这时右侧的模板窗格就会列出可选的模板对象。

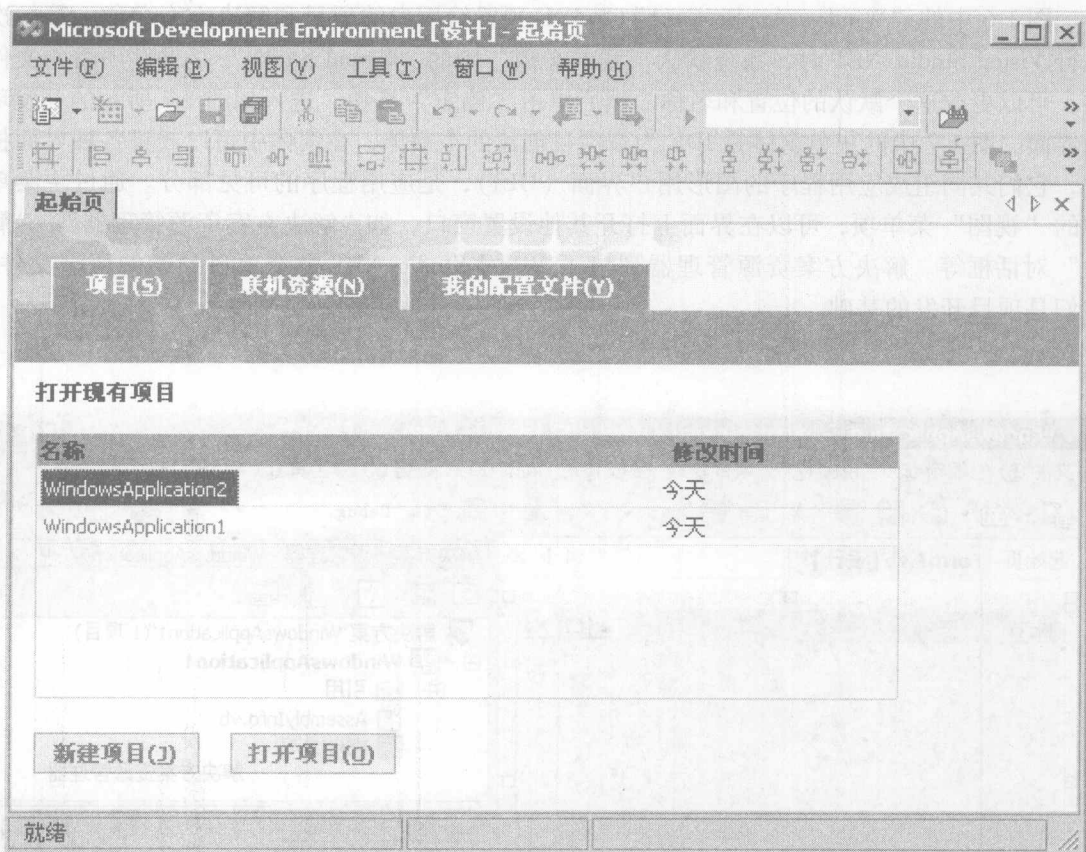
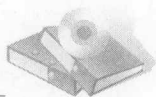


图 1-5 Visual Studio. NET 起始页

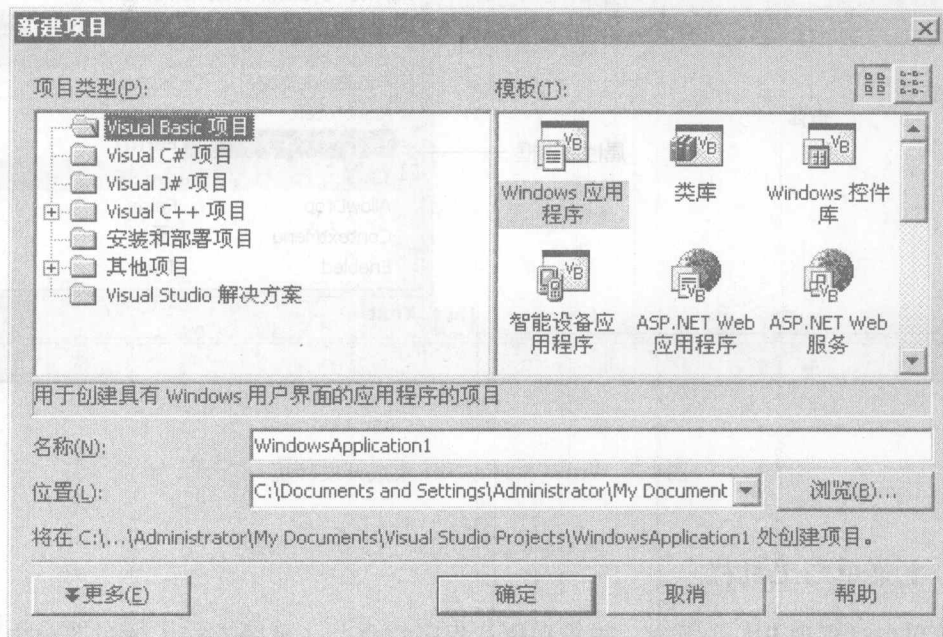


图 1-6 “新建项目”对话框



图 1-6 中的“WindowsApplication1”是 IDE 预置给用户的项目和解决方案名称。第一次执行 Visual Studio .NET 时，系统默认“我的文档”中的“Visual Studio”是保存项目的文件夹，可以更改这个默认的位置和名称，然后单击“确定”按钮，进入如图 1-7 所示的开发环境界面。图 1-7 中的灰色窗体是 IDE 为新项目创建的主窗体，在窗体中可以放置各种界面控件，它们共同组成应用程序的图形用户界面（GUI），是应用程序的可见部分。通过主菜单中的“视图”菜单项，可以在界面上打开其他设置窗口，如“解决方案资源管理器”、“属性”对话框等。解决方案资源管理器列出了 Visual Studio .NET 为新项目预置的所需文件，它们是项目开发的基础。

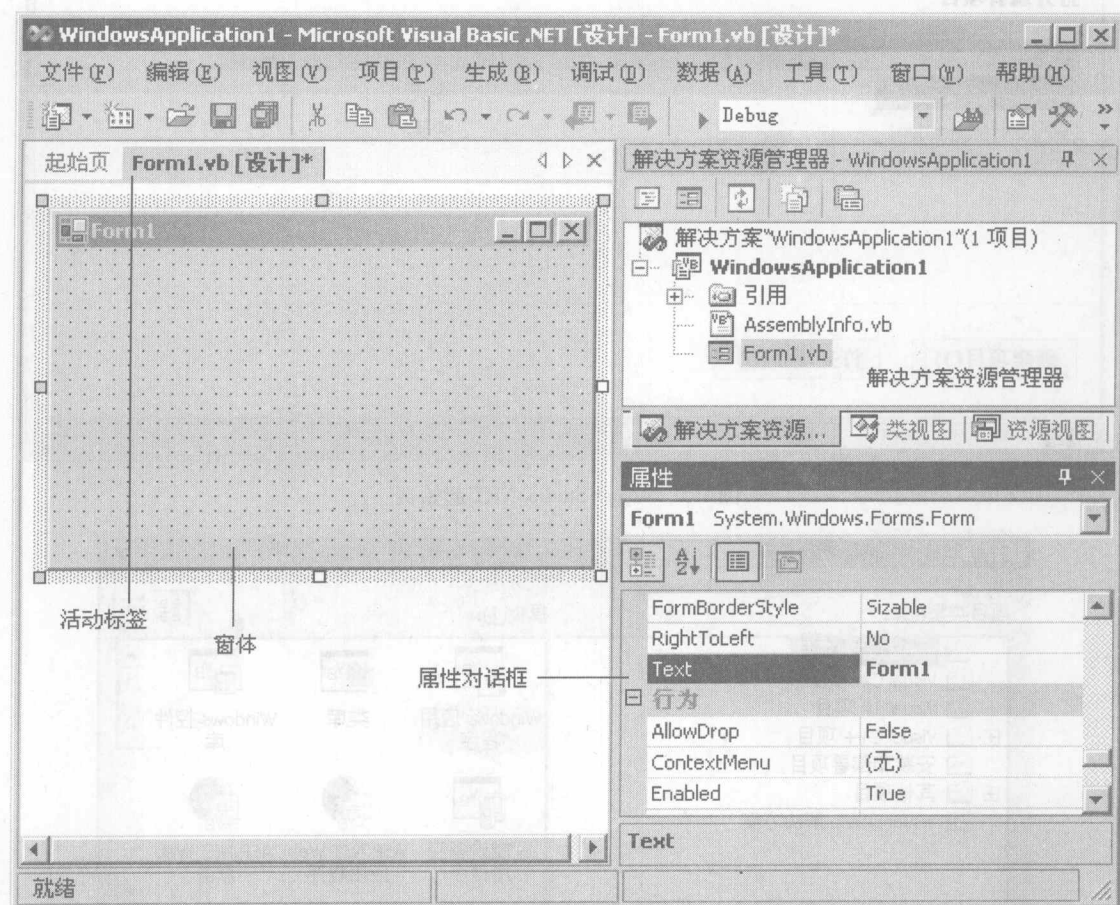


图 1-7 Visual Studio .NET 集成开发环境

1.3.2 主菜单与工具栏

主菜单包括管理 IDE、开发、维护和执行程序的命令（见图 1-8），通过菜单中的菜单项可以执行 IDE 的各种特定操作，表 1-1 列出了 IDE 的菜单项功能。

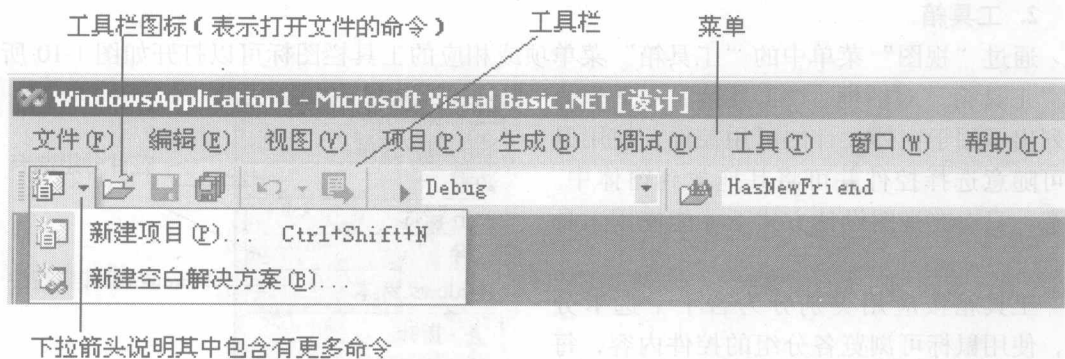


图 1-8 Visual Studio. NET 菜单与工具栏

表 1-1 Visual Studio. NET IDE 的菜单项

菜单	执行意义	菜单	执行意义
文件	包含打开、关闭项目，打印项目数据等命令	数据	包含与数据库交互的命令
编辑	包含剪切、粘贴、查找、撤销操作等命令	格式	包含排列窗体控件的命令
视图	包含显示 IDE 窗口和工具栏的命令	工具	包含访问更多定制工具和 IDE 选项的命令
项目	包含管理项目及其文件的命令	窗口	包含排列和显示窗口的命令
创建	包含编译程序的命令	帮助	包含访问 IDE 帮助特性的命令
调试	包含调试和运行程序的命令		

1.3.3 Visual Studio. NET IDE 窗口

Visual Studio. NET 提供了访问项目文件和定制控件的窗口，通过“视图”中的菜单或主窗体的工具栏可访问这些设置窗口。

1. 解决方案资源管理器

解决方案资源管理器用于访问解决方案中的所有文件，如果窗体中没有出现解决方案资源管理器窗口，可通过“视图”菜单查看代码 视图设计 显示所有文件 属性对话框中的“解决方案资源管理器”菜单项或相应的工具栏图标打开这个窗口，如图 1-9 所示。

当解决方案只有一个项目时，它就是解决方案的启动项目。向解决方案添加新项目的方法是，选中解决方案标题后，使用右键菜单执行“添加”→“新建项目”→“现有项目”命令可添加新项。添加项目后，还可以重新设置启动项目。选中需要作为启动



图 1-9 解决方案资源管理器