



高等学校电子信息类专业规划教材

VB.NET 程序设计

田 原 李素若 编 著
陈志宏 胡玉荣



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类专业规划教材

VB.NET 程序设计

田 原 李素若 陈志宏 胡玉荣 编著

清华大学出版社
北京交通大学出版社
· 北京 ·

内 容 简 介

本书面向本科院校的学生和 Visual Basic .NET 的初学者,注重从初学者的认识规律出发,强调实用性、可操作性。本书不仅对 Visual Basic .NET 的基本概念和基本设计方法的讲解浅显易懂、深入浅出,而且安排了大量典型实用的例题,使学习者结合实例学习、掌握设计的方法和技巧。书中还对 Visual Basic .NET 应用程序的设计做了比较全面的介绍。全书共 13 章,主要内容包括: Visual Basic .NET 基础、Visual Basic .NET 语言基础、Visual Basic .NET 常用控件、用户界面设计、面向对象程序设计、图形与多媒体应用程序设计、文件访问、调试与异常处理、ADO .NET 数据访问技术、定制 Visual Basic .NET 控件、部署 Visual Basic .NET 应用程序、Web 应用程序设计、建立 Web 服务。

本书结构清晰、内容丰富,可作为本科、大专层次院校的教材,也可作为工程师和程序设计爱好者的学习参考用书,并可作为各类培训班的培训用书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

VB .NET 程序设计 / 田原等编著. —北京:清华大学出版社;北京交通大学出版社,2006.6
(21 世纪高等学校电子信息类专业规划教材)

ISBN 7-81082-805-3

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

中国版本图书馆 CIP 数据字(2006)第 061198 号

责任编辑:杨祯 特邀编辑:吴炳林

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:18.25 字数:443 千字

版 次:2006 年 6 月第 1 版 2006 年 6 月第 1 次印刷

书 号:ISBN 7-81082-805-3/TP·286

印 数:1~4 000 册 定价:26.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043,51686008;传真:010-62225406;E-mail:press@center.bjtu.edu.cn。

前 言

在微软的软件开发工具中, Visual Studio 一直是 Windows 平台上集成能力最强、功能最齐的程序开发上具。新版的 Visual Studio .NET 针对多项软件开发技术加以强化。其中, Visual Basic .NET 不但继承了以往易学易用的特性, 让初学者得以轻松进入程序设计的殿堂, 而且支持 Visual Basic 以前版本所欠缺的面向对象机制。这样, 使用 Visual Basic .NET 来开发大型的软件系统时, 就可以降低软件设计与维护的复杂度。

在软件开发架构上, .NET 框架将成为未来软件开发的通用平台。.NET 框架很容易移植到各个不同的平台上(Windows、UNIX、PDA、手机等)。Visual Basic .NET 程序员可以直接运用 .NET 框架中完整且丰富的类库, 设计出跨平台的软件系统。另外通过公共语言运行库(Common Language Runtime)的接口, 在 .NET 框架上所开发的软件, 不论使用哪一种程序语言, 都可以直接互相引用。在多人共同开发软件系统时, 就可以使用程序员自己所专长的程序语言来参与软件开发, 直接解决了跨语言集成的问题。

微软提出的 .NET 战略所要建立的信息应用环境, 就是让用户可以在任何时刻、任何地点、使用任何设备来存取所需要的信息。而 .NET 框架与 Visual Studio .NET 正是用来构建 .NET 战略的基础平台与开发工具。通过开发 XML Web 服务与相关的机制, 不但能够集成各种平台上的开发成果, 降低企业 e 化成本与缩短导入时间, 而且提高了分布式系统的安全性 with 多功能性。

下面介绍本书的主要内容。

第 1 章, 简单介绍了 .NET Framework、Visual Basic .NET 的一些新的特性。通过本章学习, 读者应该对 .NET Framework、Visual Basic .NET 的集成开发环境有一个初步的了解。

第 2 章, 讲解 Visual Basic .NET 的语言基础知识。主要内容包括 Visual Basic .NET 的数据类型、运算、常量和变量、流程和控制结构, 以及过程和函数。

第 3 章, 介绍了 Visual Basic .NET 常用控件。使用控件可使应用程序的设计免除大量重复性工作, 简化设计过程, 有效地提高设计效率。要编写出具有实用价值的应用程序, 必须掌握每类控件的功能、用途, 并掌握其常用的属性、事件和方法。

第 4 章, 讲解用户界面设计知识。主要讲解控件布局、菜单设计、对话框设计、多文档界面设计和控制台应用程序设计。

第 5 章, 介绍了 Visual Basic .NET 中的面向对象的程序设计方法。Visual Basic .NET 不仅提供了继承, 而且还提供了其他许多重要的新特性。Visual Basic .NET 改进了创建和处理多接口的方法, 使得可以更容易地使用它们。另外, Visual Basic .NET 支持事件作为接口的一个部分, 使得现在可以在接口上表达所有的元素: 方法、属性和事件。

第 6 章, 对 Visual Basic .NET 的图形、图像功能做了全面的介绍, 并通过一些简单的实例说明了它们的功能和使用方法。通过音频与视频播放器、Flash 播放器的设计简单介绍了多媒体程序设计知识。

第 7 章, 介绍文件对象的属性和常用方法, 帮助读者掌握对文件进行读写操作的一般步

骤以及对文件进行读写的多种方法。

第8章,说明如何使用 Visual Basic. NET 的调试工具处理程序运行时的错误,并介绍如何开发用于处理运行时错误的代码。

第9章,介绍采用 ADO. NET 访问数据库的知识。

第10章,介绍 Visual Basic. NET 控件的基本概念和定制 Visual Basic. NET 控件的基本方法,如何根据自己的需求定制自定义 Visual Basic. NET 控件。

第11章,介绍如何通过向解决方案中添加部署项目来部署 Visual Basic. NET 应用程序,并通过运行安装向导来创建所需的安装文件。

第12章,介绍 Web 应用程序设计。Microsoft 公司在加强 . NET 框架的传统应用程序设计能力的同时,重点提升了 Web 应用程序开发能力和效率。

第13章,建立 Web 服务。在 Visual Basic. NET 中可以快速地建立起 Web 服务,可以很容易地开发出客户端的应用程序。在本章中,介绍如何在客户端使用这些 Web 服务,如何利用 Web 服务来满足我们对信息的需求。

在编写本书时,作者充分考虑到初学者学习程序语言的特点,在对基本知识的讲解上,力求做到深入细致,并结合大量示例,用操作步骤的形式,一步步向读者揭开 Visual Basic. NET 的神秘面纱。

本书结构清晰、内容丰富、重点突出、语言流畅,既有理论性的阐述,也有具体的开发实例,可作为本科、大专层次院校的教材,也适用于使用 Visual Basic. NET 进行软件开发的广大技术人员。

本书由田原、李素若、陈志宏、胡玉荣编著。其中第1、2、8、11章由田原编写,第3、4章由李素若编写,第5、6、7章由陈志宏编写,第9、10、12、13由胡玉荣编写。田原负责审定全书内容。

由于作者水平有限,书中不妥或错误之处在所难免,敬请读者指正。

作 者

2006 年 4 月

目 录

第 1 章 Visual Basic. NET 基础	(1)
1. 1 .NET Framework 概述	(1)
1. 1. 1 公共语言运行库	(1)
1. 1. 2 类库	(1)
1. 1. 3 编译成中间语言	(2)
1. 1. 4 程序集	(2)
1. 1. 5 引用集合	(3)
1. 1. 6 ASP. NET	(3)
1. 2 Visual Basic. NET 概述	(3)
1. 3 Visual Basic. NET 的新特点	(3)
1. 4 Visual Basic. NET 的集成开发环境	(6)
1. 5 Visual Basic. NET 的第一个应用程序	(8)
1. 5. 1 创建 Visual Basic. NET 工程	(8)
1. 5. 2 创建应用程序的用户界面	(9)
1. 5. 3 设置用户界面中各对象的属性	(10)
1. 5. 4 编写程序代码	(11)
1. 5. 5 保存和运行程序	(12)
1. 5. 6 创建可执行文件	(12)
1. 6 小结	(12)
第 2 章 Visual Basic. NET 语言基础	(13)
2. 1 Visual Basic. NET 的数据类型	(13)
2. 1. 1 Numeric 数据类型	(13)
2. 1. 2 Byte 数据类型	(13)
2. 1. 3 String 数据类型	(14)
2. 1. 4 Boolean 数据类型	(15)
2. 1. 5 Date 数据类型	(15)
2. 1. 6 Object 数据类型	(16)
2. 1. 7 用户自定义类型	(16)
2. 1. 8 数组	(17)
2. 2 Visual Basic. NET 的运算	(20)
2. 2. 1 算术运算	(20)
2. 2. 2 赋值运算符	(21)

2.2.3	二进制运算符	(21)
2.2.4	比较运算符	(22)
2.2.5	连接运算符	(23)
2.2.6	逻辑运算符	(23)
2.2.7	运算符的优先级	(24)
2.3	Visual Basic. NET 的常量和变量	(25)
2.3.1	常量	(25)
2.3.2	变量	(26)
2.4	Visual Basic. NET 的流程和控制结构	(29)
2.4.1	条件分支结构	(29)
2.4.2	循环结构	(31)
2.5	Visual Basic. NET 的过程和函数	(34)
2.5.1	Sub 过程	(34)
2.5.2	Function 过程	(35)
2.5.3	调用过程	(36)
2.5.4	向过程传递参数	(37)
第3章	Visual Basic. NET 常用控件	(40)
3.1	控件概述	(40)
3.1.1	控件的基本属性	(40)
3.1.2	焦点和设置 Tab 键次序	(41)
3.2	Button 控件	(42)
3.3	文本控件	(43)
3.3.1	Label 控件	(43)
3.3.2	TextBox 控件	(45)
3.4	图形控件	(48)
3.4.1	PictureBox 控件	(48)
3.4.2	ImageList 控件	(49)
3.5	选择控件——复选框和单选按钮	(51)
3.5.1	CheckBox 控件	(51)
3.5.2	RadioButton 控件	(52)
3.5.3	GroupBox 控件	(52)
3.6	选择控件——列表框和组合框	(54)
3.6.1	ListBox 控件	(54)
3.6.2	ComboBox 控件	(55)
3.7	滚动条控件	(57)
3.8	体现 Windows 风格控件	(58)
3.8.1	MainMenu 控件	(58)
3.8.2	ListView 控件	(59)

3.8.3	TreeView 控件	(62)
3.8.4	ToolBar 控件	(64)
3.9	辅助控件	(66)
3.9.1	ProgressBar 控件	(66)
3.9.2	Timer 控件	(66)
第4章	用户界面设计	(69)
4.1	界面设计概述	(69)
4.1.1	.NET 应用程序用户界面形式	(69)
4.1.2	界面设计的原则	(69)
4.2	控件的布局	(71)
4.2.1	控件的缩放和移动	(71)
4.2.2	控件的复制和删除	(71)
4.2.3	控件的布局	(71)
4.3	菜单设计	(72)
4.3.1	菜单的基本结构	(72)
4.3.2	创建下拉式菜单	(73)
4.3.3	弹出式菜单设计	(76)
4.4	工具栏设计	(77)
4.5	状态栏设计	(80)
4.6	对话框设计	(83)
4.6.1	消息框和输入框	(83)
4.6.2	通用对话框	(86)
4.6.3	自定义对话框	(92)
4.7	多文档界面设计	(95)
4.7.1	创建 MDI 父窗体与子窗体	(95)
4.7.2	确定活动子窗体	(96)
4.7.3	子窗体排列	(97)
4.8	控制台应用程序	(100)
第5章	面向对象程序设计	(102)
5.1	面向对象编程基础	(102)
5.1.1	对象和类	(102)
5.1.2	封装、继承和多态性	(102)
5.2	创建一个自己的类和对象	(103)
5.2.1	类关键字	(104)
5.2.2	类成员	(104)
5.2.3	类的实例	(105)
5.2.4	自定义命名空间	(106)
5.2.5	访问类型	(107)

5.2.6	数据成员	(108)
5.2.7	创建与重载方法	(110)
5.2.8	使用 Property 语句定义属性	(112)
5.2.9	构造器与析构器	(113)
5.3	继承与重载的实现	(114)
5.3.1	实现基本的继承	(115)
5.3.2	禁止继承	(117)
5.3.3	继承和作用域	(117)
5.3.4	Protected(保护)方法	(118)
5.3.5	重载的实现	(119)
5.3.6	窗体的继承和应用	(121)
5.4	共享类成员的实现	(123)
5.4.1	共享方法	(124)
5.4.2	共享变量	(124)
5.4.3	全局变量	(125)
5.5	事件	(125)
5.5.1	创建一个自定义事件	(126)
5.5.2	共享事件	(127)
5.5.3	在不同项目之间触发事件	(127)
5.6	接口	(130)
5.6.1	接口的声明	(130)
5.6.2	Implements 关键字	(131)
5.6.3	实现接口	(131)
5.6.4	实现多个接口	(132)
5.7	小结	(134)
第6章	图形与多媒体应用程序设计	(135)
6.1	坐标系统和颜色	(135)
6.1.1	坐标系统	(135)
6.1.2	使用 Visual Basic. NET 颜色	(135)
6.2	PictureBox 控件	(137)
6.3	ImageList 控件	(137)
6.3.1	ImageList 的属性和方法	(137)
6.3.2	Item 对象的属性	(138)
6.3.3	Images 集合的属性和方法	(138)
6.3.4	设置 ImageList 控件的属性	(139)
6.4	Graphics 对象	(140)
6.4.1	绘制文本	(141)
6.4.2	绘制直线	(142)

6.4.3	绘制长方形	(143)
6.4.4	绘制椭圆和圆形	(144)
6.4.5	绘制曲线	(146)
6.5	多媒体程序设计	(149)
6.5.1	Media Player 控件	(149)
6.5.2	Flash 播放器	(154)
6.6	小结	(156)
第7章	文件访问	(157)
7.1	文件系统对象	(157)
7.1.1	File 对象	(157)
7.1.2	Directory 对象	(162)
7.2	文件的读写	(165)
7.2.1	概述	(166)
7.2.2	Stream 对象	(166)
7.2.3	FileStream 对象	(167)
7.2.4	读取文件数据	(167)
7.2.5	向文件写入数据	(172)
7.3	小结	(175)
第8章	调试与异常处理	(176)
8.1	程序错误	(176)
8.2	程序调试	(178)
8.2.1	Visual Basic .NET 的调试工具	(178)
8.2.2	调试示例	(179)
8.3	异常捕获及处理	(185)
8.3.1	Try...Catch 语句	(185)
8.3.2	On Error 语句	(186)
8.3.3	On Error Goto...Resume 结构	(188)
8.3.4	On Error Goto...Resume Next 语句	(189)
8.3.5	Resume 与 Resume Next 的区别	(190)
8.3.6	Error 语句	(191)
8.3.7	使用 Err 对象	(192)
8.3.8	结构化的异常处理语句	(193)
8.4	小结	(194)
第9章	ADO.NET 数据访问技术	(195)
9.1	数据访问基础知识	(195)
9.1.1	数据库基础知识	(195)
9.1.2	ADO.NET 简介	(195)
9.1.3	结构化查询语言 SQL	(196)

9.1.4	ADO.NET 数据存取的基本概念	(198)
9.2	应用 ADO.NET 访问数据库	(200)
9.2.1	创建和使用 Connection 对象	(200)
9.2.2	创建和使用 Command 对象	(201)
9.2.3	创建和使用 DataAdapter 对象	(202)
9.2.4	创建和使用 DataSet 对象	(203)
9.3	数据绑定	(204)
9.3.1	简单的数据绑定	(204)
9.3.2	复杂的数据绑定	(205)
9.4	使用数据控件访问数据库	(206)
第 10 章	定制 Visual Basic.NET 控件	(216)
10.1	用控件创建向导创建 Visual Basic.NET 控件	(261)
10.1.1	创建 Windows 控件库项目	(216)
10.1.2	添加测试项目	(216)
10.1.3	设计 FirstControl 控件	(217)
10.1.4	测试生成的 FirstControl 控件	(219)
10.1.5	增强 FirstControl 控件	(223)
10.2	小结	(225)
第 11 章	部署 Visual Basic.NET 应用程序	(226)
11.1	设计部署项目	(226)
11.1.1	部署项目概述	(226)
11.1.2	部署应用程序的不同方法	(227)
11.2	创建部署项目	(228)
11.2.1	使用安装向导创建部署项目	(228)
11.2.2	运行安装向导	(229)
11.2.3	使用安装项目模板创建部署项目	(233)
11.3	自定义部署项目	(235)
11.3.1	配置编译方式	(235)
11.3.2	创建应用程序快捷方式	(236)
11.3.3	设置公司名称和版本信息	(237)
11.3.4	设置部署属性页	(237)
11.4	编译部署项目和测试安装程序	(239)
11.4.1	生成项目	(239)
11.4.2	运行安装程序	(240)
11.4.3	运行 HelloWorld 应用程序	(241)
11.5	检查安装文件和卸载应用程序	(242)
11.5.1	检查最终安装文件	(242)
11.5.2	卸载测试应用程序	(242)

第 12 章 Web 应用程序设计	(244)
12.1 ASP.NET 简介	(244)
12.1.1 ASP.NET 的功能	(245)
12.1.2 ASP.NET 与 ASP 比较	(247)
12.1.3 ASP.NET 中文件类型	(248)
12.2 基于 Web 表单的开发	(249)
12.2.1 Web Form 介绍	(249)
12.2.2 第一个 Web Form	(250)
12.2.3 页面处理过程	(255)
12.2.4 View State	(255)
12.3 使用 ASP.NET 开发应用程序	(256)
12.3.1 Global.asax 文件	(256)
12.3.2 Web.config	(258)
12.4 服务器控件	(261)
第 13 章 建立 Web 服务	(265)
13.1 创建 Web 服务	(265)
13.1.1 创建一个 XML Web 服务项目	(266)
13.1.2 建立 XML Web 服务客户端项目	(270)
13.2 一个关于数据库 XML Web 服务的实例	(273)
13.2.1 建立服务器端程序	(274)
13.2.2 对客户端进行身份验证	(276)
13.2.3 创建 Windows 应用客户界面	(277)
参考文献	(280)

第 1 章 Visual Basic. NET 基础

Visual Basic. NET 是 Visual Basic 的最新版本, 它对原版本进行了重大的升级和增强。其最大优势在于它能够提高日常开发效率。除此之外, 一旦适应了 Visual Studio. NET 的开发环境, 就能使用同样的工具为 Visual C ++ . NET、Visual C#. NET 以及其他第三方工具和编译器编写程序。

1.1 .NET Framework 概述

随着 .NET Framework 和 Visual Studio. NET 的引入, Microsoft 使得 Windows 应用程序和 Web 应用程序的编写发生了重大的变革。这些新产品使 Visual Basic 程序开发产生了很大的变化。.NET 不但为 Visual Basic 语言带来了真正的面向对象的编程, 并且使跨平台兼容开发项目变得更容易。

.NET 的两个主要组成部分是 Microsoft .NET Framework SDK (软件开发工具包) 和 Visual Studio IDE (集成开发环境)。IDE 用来开发程序, 而 Framework 则用于运行程序。

.NET Framework 为开发和运行在不同平台、不同语言编写的应用程序以及 XML Web 服务提供了一个平台。.NET Framework 由公共语言运行库、类库以及 ASP. NET 组成, ASP. NET 是基于组件的动态服务器页面。

1.1.1 公共语言运行库

公共语言运行库 (CLR) 是管理代码执行的环境。公共语言运行库可以为一些任务提供服务, 如集成由不同语言开发的组件, 跨语言错误处理, 安全处理, 管理对象的存储和释放等。

任何被编译到公共语言运行库中运行的代码称为托管代码 (managed code)。托管代码自动包含元数据。元数据是指描述数据的数据。每个公共语言运行库的可移植、可执行 (PE, Portable executable) 文件都同时包含了元数据和代码。元数据包括数据类型、成员、引用以及加载类和调用类中方法时所需要的信息。

公共语言运行库也管理数据存储。当不再使用某个对象时, 我们可以用公共语言运行库的无用单元收集器 (garbage collection) 组件来从内存中自动释放该对象。如果用运行库来处理对象的无用单元收集, 相关的数据就称为托管数据。虽然程序员可以自己管理数据的存储, 但最好还是让运行库来管理。

程序员编写的代码可以跟用其他编程语言编写的托管代码中的类及方法进行集成。公共语言运行库定义了数据类型标准, 从而可将类的实例传递给用不同语言编写的方法。

1.1.2 类库

作为 .NET 语言的一部分, 所有类和接口都保存在 .NET Framework 类库中。.NET Framework 类库由许多命名空间组成。应该熟悉一些常用的命名空间, 如 System 及 System. Drawing

等。每个命名空间都包含类、结构体、枚举、委托和/或接口，可以在程序中使用这些组成部分。

表 1-1 给出了 .NET Framework 类库中的部分命名空间。

表 1-1 .NET 类库中的部分命名空间

命名空间	说明
System	数据类型、事件以及事件驱动的基类和基本类
System.Collections	定义对象集合，如表、队列以及字典
System.DataADO.NET	体系结构
System.Drawing	GDI + 图形
System.IO	读写数据流和数据文件的类型
System.Security	权限的基类
System.Threading	多线程编程类
System.Web.Service	创建和使用 Web 服务的类
System.Windows.Forms	为在 Windows 操作系统中执行的程序创建图形组件的类
System.XML	支持 XML 处理，XML 是数据传输标准

.NET Framework 类库中的类遵循已发布的通用语言规范 (CLS) 标准。通用语言规范规定了与公共语言运行库进行交互的语言行为。如果想让程序可以同用其他语言编写的程序以及组件进行交互，那么就应该确保程序遵循通用语言规范。通用语言规范遵循规则可以在 .NET Framework Developer's Guide 下的文章《What is the Common Language Specification》中找到。

1.1.3 编译成中间语言

程序员编写的程序代码一般称为源代码。编译器会将源代码翻译成 Microsoft 中间语言 (MSIL)，有时也称为中间语言 (IL)。MSIL 是一组独立于平台的指令，它和元数据一起形成可移植可执行 (PE) 文件，该文件的扩展名为 .exe 或 .dll。当程序运行时，实时 (JIT) 编译器将 MSIL 转换为特定机器的本机代码，实时编译器是公共语言运行库的一部分，如图 1-1 所示。

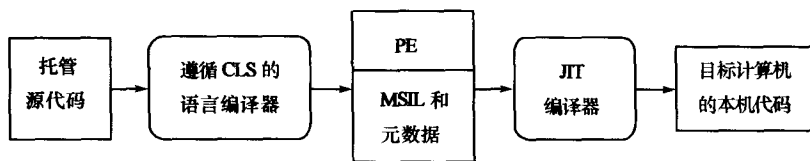


图 1-1 用 .NET 公共语言运行库编译和执行程序的步骤

1.1.4 程序集

程序集 (assembly) 是单个或多个可移植可执行文件代码的基本单位。每一个应用程序都包含在单个程序集中，程序集是可部署的最小代码段。每个程序集都有程序集清单，

程序集清单中包含了与版本有关的元数据、描述程序集需要的所有文件的表格以及程序集引用列表（assembly reference list），该列表指定了程序集需要的所有外部文件，如其他人创建的动态链接库等。程序集类似于 Visual Basic 早期版本编写的 exe 或 dll 文件，其中包含了运行应用程序或者组件的所有必要信息。

.NET Framework 类库不是一个大的文件，而是多个文件的集合。类被存储在许多扩展名为 .dll 的 dll 文件（动态链接库）中。类库中的每一个动态链接库都是一个程序集。当我们的程序开始执行时，只有需要用到的程序集才被加载到内存中。当想使用一个还没有被引用的类型时，必须添加一个对动态链接库（程序集）的引用。

特性（attribute）是包含与程序中某些部分（如类型和方法）相关信息的标记。系统定义了程序集的许多特性，例如名称、版本、语言以及安全等。特性是 .NET 程序集元数据的一部分。

1.1.5 引用集合

引用对象用来将 Visual Basic 项目连接到外部组件。引用对象的两种类型是程序集和组件对象模型（COM）对象。对另一个项目的引用称为程序集引用，也称为项目到项目引用（Project-to-project reference）。COM 对象是用以前版本的 VB 或其他不遵循通用语言规范的语言编写的组件。

1.1.6 ASP.NET

.NET 中很大一部分改进是针对 Web 开发的。ASP.NET 是活动服务器页面的最新版本。ASP.NET 是一种 Web 开发环境，它可以编译用包括 Visual Basic 在内的任一种 .NET 语言编写的应用程序。这意味着在开发 Web 应用程序时，我们可以利用公共语言运行库和托管代码的优点。ASP.NET 使得 Web 开发变得更为容易，因为它可以为 Web 窗体和 Web 服务提供同 Windows 应用程序一样的调试支持。

1.2 Visual Basic. NET 概述

Visual Basic. NET（又称 Visual Basic 7.0 或 VB 7.0）是 Microsoft 公司在 Visual Basic 6.0 之后推出的最新版本，集成在 Visual Studio. NET 中，与 Visual C++ .NET 以及 C# 组成了 .NET 构架。

Visual Basic. NET 是 Windows 环境下简单、易学、高效的一种编程语言，其快速开发的特性深受程序员的喜爱，但是 .NET 以前版本的 Visual Basic 面向对象的能力远远不能满足程序员的需要，这也是项目越大用到 Visual Basic 越少的原因。Visual Basic. NET 新增了许多面向对象的特性，比如继承、重载等。语言的新特点也包括了对进程的控制和底层结构的操作，这些新特性使得 Visual Basic. NET 再次成为程序员关注的焦点。

1.3 Visual Basic. NET 的新特点

一直以来，大家总是针对 Visual Basic 究竟是“面向对象”还是“基于对象”，甚至

“面向组件”的语言而争论不休。但是，这些争论在 Visual Basic.NET 面前，马上就会偃旗息鼓。因为不管依照怎样的标准来定义“面向对象”，Visual Basic.NET 绝对是符合面向对象的每一个标准的。下面看看它具有怎样的特性。

1. 构造函数

当一个对象被创建的时候，它是否能够被正确地初始化，这是我们比较关心的问题。利用构造函数我们就可以同时给这个对象中的成员赋初值，这样有助于正确地初始化一个对象。Visual Basic 以前的版本中，在创建对象的时候，必须要等到对象创建完毕之后，另外调用一个独立的方法来对它进行初始化。在 Visual Basic.NET 中，我们可以利用构造函数给对象赋初值，这样就不需要再去进行烦琐的调用赋初值了。构造函数简化了编码的过程，也减少了出错的机会。

2. 封装性

从字面上理解，封装就是将某事物包围起来，使外界不知道其实际内容。在程序设计中，封装的意思就是程序提供一个包含了一系列过程和函数的接口，其他的程序可以直接利用对象的这个接口，而不需要去了解接口里面的具体代码。接口中的代码一般来说总是实现一些相对完整的功能。

3. 自由线程

线程是进程中的一个实体，是被系统独立调度和分派的基本单位。线程是一个很复杂的概念，无法用简单的几句话来解释。基本上来讲，线程的意思就是让我们的程序可以在同一时间内做两件以上的的事情。比如，我们也许需要让程序在后台打印一个文档，同时还要让它响应当前用户不断发出的命令。线程在创建具有高度扩展性服务端程序的时候显得比较有用，并且，利用线程来编写具有极强交互性用户界面的时候也十分容易，例如我们可以为一个需要长时间运行的程序加一个 Cancel 按钮，以使用户有更多的选择。以前 Visual Basic 编程的时候对线程想尽了办法，调试线程的复杂和不稳定性使线程编写相当复杂（因为只能使用 API 来写，还是伪线程的，与其说是多线程，还不如说是多进程恰当）。现在，Visual Basic.NET 提供了编写线程的能力，线程编写在调试上是相当复杂的，Visual Basic.NET 终于以一种相对简单的形式来实现了。

4. 继承

继承是面向对象系统中的另一个重要的概念，一个语言是否具有继承特性常被人们用作判断是否是面向对象语言的关键标准。继承所表达的就是一种类的相关关系，它使得某类可以继承另外一种类的特征和能力。类间具有继承关系应有三种特征：类间具有共享特征；类间具有细微的差别或新增部分；类间具有层次结构。

从对象的角度来看，继承是这样的一种概念：如果一个对象能够获得另外一个对象的接口和方法，并且可以扩展这些接口和方法，我们就称这个对象继承了另一个对象。举个实际的例子，我们在产品存储程序中，可以创建一个通用的处理所有产品的产品对象 Product。从这个对象中，根据是否需要上税，又可以派生出一个免税产品对象 NonTaxableProduct 和一个需税产品对象 TaxableProduct。两个对象都继承原始产品对象 Product 的接口和所有方法，但是将根据各自不同的实际情况在需要的地方修改或者扩展原始对象的方法。

5. 初始化函数

以前在 Visual Basic 中还需要首先声明一个变量，然后才能对它赋初值，这样用户需

要写两行代码。初始化函数可以让用户将这两个步骤合并在一行代码中完成，虽然这是一个微小的变化，但是从代码的角度来看，这个改进提供了更少、更简单、更容易维护的代码。

6. 基于对象

对于一种计算机语言来讲，做到基于对象要比面向对象相对容易些。不过，什么是基于对象，到现在也只有一个很模糊的定义，大概是指一种语言具有直接和对象进行交互的能力。VB 3.0 就已经是基于对象的语言了，当时它能够和 DAO 对象以及控件交互。后来随着版本的提升，这些可交互的对象已经扩大到了 ActiveX 控件、RDO、ADO 对象等。

7. 面向对象

一种面向对象的语言至少需要满足三个条件：封装性（Visual Basic 4.0 中就已经实现了）、继承性（Visual Basic .NET 中才有）和多态性（多态性指的是多形态，就是对多个对象的相同接口，这在 Visual Basic 3.0 中就有了）。所以在 Visual Basic .NET 中，实现了完全的面向对象，因为它已经完全满足了这三个条件。

8. 重载

重载的含义是指通过为函数或运算创建附加定义而使它们的名字可以重载，也就是说相同名字的函数或运算符在不同场合可以表现出不同的行为。对象们经常在类似的数据上开展同样的功能，比如，我们需要能够加服务和产品项到一个订购对象上，现在，我们必须分别对订购对象执行一个方法（添加服务）和另一个方法（添加产品项）。利用函数重载，我们仅需要执行一个同名的方法，然后让 Visual Basic 语言自己根据情况去选择执行正确的函数。

9. 覆盖

当使用继承功能的时候，新类从父类那里得到了一切方法。但是，也许我们需要让其中的某一种或者几种函数执行另外的动作，可以通过覆盖原始的方法来实现这样的功能。我们的新代码也可以调用原始的父亲对象的原始方法。

10. 多态性

简单地说，这就是让两个不同的类型的对象，执行同一种方法的能力。另一种解释是：描述同一个消息可以根据发送消息对象的不同采用多种不同的行为方式。Visual Basic 支持后来的捆绑，而在 Visual Basic 5.0 中通过执行关键字和多个 COM 接口来实现多态。现在，在 Visual Basic .NET 版本中，我们将看到继承接口同样也可以具有多态性。最后，Visual Basic .NET 在执行多态性的时候给程序相当大的灵活性。

11. 共享成员

共享成员又叫做静态成员和类级成员。共享成员就是对类的所有的实例来说，相同的方法或变量，每一个对象创建的时候，基于一个给定的类。分享这些相同的变量和函数，比如，我们可以计算出一个类有多少个实例被创建了，只要我们声明一个共享的计数器变量就行。

12. 结构化错误处理

替代了不稳定的也极不灵活的 On Error GoTo 语句，我们有一个新的结构来处理错误。这个结构包含了 TRY、CATCH 和 END TRY 等关键字。在 TRY 和 CATCH 之间的代码块被保护，如果有个错误出现了，这个 CATCH 代码块就会运行，在 CATCH 运行完之后，模