微软指定MCAD/MCSD教材

Microsoft Press

从入门到精通自学用书

考试70-306和70-316

- 单本全球销量超过200万册
- 应用程序开发人员最具价值的手册
- 通过MCAD/MCSD考试的可靠保障

MCAD/MCSD制胜宝典

用Visual Basic .NET和Visual C# .NET开发Windows应用程序

微软公司 著
郑宇红 徐泓 译

清华大学出版社

微软指定 MCAD/MCSD 教材

MCAD/MCSD 制胜宝典
用 Visual Basic .NET 和 Visual C# .NET 开发 Windows 应用程序

[美] 微软公司 著
郑宇红 徐 泓 译

清 华 大 学 出 版 社
北 京

内 容 简 介

本书是 MCAD/MCSD 制胜宝典系列丛书之一，全面介绍了使用 Visual Basic .NET 或 Visual C# .NET 开发基于 Windows 的解决方案的知识和技能。主要内容包括：创建用户界面，添加控件和验证用户输入，使用包括封装和方法重载的 OOP 技术，生成自定义控件和.NET 程序集，使用 XML 和 ADO.NET 访问和修改数据，实现打印支持、在线帮助、可访问性与全局化特性，测试和调试代码错误，配置和确保应用程序的安全，借助可移动媒体、Web 或网络部署应用程序，维护和优化应用程序性能等。

本书适用于需要设计、规划、实现和支持基于 Windows 的应用程序的软件开发人员，也适用于准备参加 MCP 70-306 和 70-316 考试的人员使用。

MCAD/MCSD 制胜宝典——用 Visual Basic .NET 和 Visual C# .NET 开发 Windows 应用程序
MCAD/MCSD Self-Paced Training Kit:Developing Windows-based Applications with Microsoft Visual Basic .NET and Microsoft Visual C# .NET(ISBN 0-7356-1533-0)
Microsoft Corporation

北京市版权局著作权合同登记号：图字 01-2002-0247 号

书　　名：MCAD/MCSD 制胜宝典
用 Visual Basic . NET 和 Visual C# . NET 开发 Windows 应用程序
作　　者：[美]微软公司
出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
译　　者：郑宇红　徐　泓
文稿编辑：冯　涛
封面设计：陈刘源
印 刷 者：北京四季青印刷厂
发 行 者：新华书店总店北京发行所
开　　本：850×1168　1/16　**印张**：25.25　**插页**：1　**字数**：670 千字
版　　次：2003 年 8 月第 1 版　2003 年 8 月第 1 次印刷
书　　号：ISBN 7-89494-114-X
印　　数：0001～4000
定　　价：50.00 元(附光盘 1 张)

前　言

欢迎使用 MCAD/MCSD 制胜宝典系列丛书之一《用 Visual Basic .NET 和 Visual C# .NET 开发 Windows 应用程序》一书。通过本书的学习，您可以掌握使用 Visual Basic .NET 或 Visual C# .NET 开发基于 Windows 的解决方案所必备的知识和技能。

另外，本书也顾及 MCP 考试目的(微软认证专家考试 Exam 70-306 和 Exam 70-316)。这里的可自定进度的课程所提供的内容，能够支持这种考试所要评估的技能。

注意　若要了解成为微软认证应用程序开发员(MCAD)或微软认证解决方案开发员(MCSD)的更多信息，请参阅附录 C 的“微软认证专家计划”。

前言中“课前准备”部分提供了重要的设置指导，说明了完成本书学习所需的硬件和软件要求。此外，还提供了完成部分操作步骤所必需的网络配置信息。开始各章节学习之前，请详细阅读这部分内容。

读 者 对 象

本书适用于需要设计、规划、实现和支持基于 Windows 的应用程序的软件开发人员，也适用于准备参加如下相关 MCP 考试的人员：

- Developing and Implementing Windows-Based Applications with Microsoft Visual Basic .NET and Microsoft Visual Studio .NET(Exam 70-306)
- Developing and Implementing Windows-Based Applications with Microsoft Visual C# .NET and Visual Studio .NET(Exam 70-316)

先 决 条 件

学习本书的人员应具备以下先决条件：

- 能使用 Visual Basic .NET 或 Visual C# .NET 创建简单的应用程序。
- 能描述 Visual Basic .NET 或 Visual C# .NET 应用程序中的基本控件与菜单的用途和用法。
- 能描述控件和事件之间的关系。
- 适当了解基本的结构化查询语言(Structured Query Language，SQL)的语法。

本书配套光盘

配套光盘中包含可用于全书学习的各种信息，其中包括：

- **完整的实验** 本书各章都包含一个实验。其中包含一系列强化所学技能的练习。同时还包括这些应用程序的完全版，以便对比练习结果。如果在完成练习时遇到困难，也可以参考这些完整的应用程序。
- **必需的文件** 执行动手程序所需的练习文件。当在练习中有所说明时，应使用这些文件。
- **考试问题示例** 若要进行认证考试的练习，可使用光盘中提供的考试问题示例。这些示例问题有助于测评对本书的理解。

关于本书和光盘的附加支持信息(包括安装和使用的常见问题的答案)，请访问微软出版社技术支持 Web 站点 *www.microsoft.com/mspress/support/*。也可以给 TKINPUT@MICROSOFT.COM 发电子邮件，或者发信至 Microsoft Press, Attn：Microsoft Press Technical Support, One Microsoft Way, Redmond, WA 98502-6399。

本 书 特 色

每章包含若干小节，均经过精心设计，帮助您从各章之中最大限度地获益：

- 每章都以“先决条件”开始，提示学习该章内容要做的准备。
- 每章都划分为若干小节。每一小节包含用于特定技能的参考信息和操作步骤信息。这些小节和练习提供了详细分步的操作过程，并使用 ➢ 符号予以标识。
- 在每一节的末尾是“本节小结”部分，指出了本节中的关键概念。
- “实验”部分提供了动手练习，以强化每章各节讲授的技能。这些练习使读者有机会使用刚刚学到的技能、或者研究所描述的应用程序部分。实验中的练习尽可能建立于彼此之上，以在实验结束时创建一个完整的应用程序。
- 每章的结尾是“本章复习”部分，用于测试所学的内容。
- 附录 A“问题与答案”，包含每章的所有问题及其答案。

注意

本书有以下两类特殊提示：

- “注意”提供补充信息。
- “警告”包含对可能丢失数据的警告。

约定

本书将使用如下约定：

- 输入的字符和命令以**粗体**表示。
- 语法语句中的*斜体*表示变量信息的占位符。本书的标题和程序元素以及网址也均用斜体。
- 除非是直接输入，否则文件的名称和文件夹的名称均以词首大写字母出现。如果未做特别

要求，当在对话框中或命令提示符下输入文件名时，可使用小写字母。

- 当不出现文件名时，文件扩展名采用小写字母。
- 缩写词全部以大写字母出现。
- 等宽字体表示代码示例、屏幕文本示例或是可以在命令提示符下或初始化文件中输入的项。
- 尖括号“〈〉”用于在语法语句中包容可选项。例如，命令语法中的*<filename>*表示可以使用该命令输入文件名。只键入尖括号内部的信息，而不是方括号本身。
- 当在文本中同时提到 Visual Basic 和 Visual C#术语时，Visual Basic 术语首先显示，后面跟着是在圆括号中的 Visual C# 术语。

键盘约定

- 两个键名称之间用加号(+)表示必须同时按下这两个键。例如，按 Alt+Tab 表示在按 Alt 键的同时按下 Tab 键。
- 两个或多个键名称之间的逗号(,)号表示必须依次而不是同时按这些键。例如，按 Alt，F，X 表示要依次按下和依次释放这些键。按 Alt+W，L 表示要先同时按 Alt 和 W 键，然后释放再按 L 键。
- 可通过使用键盘来选择菜单命令。按 Alt 键可以激活菜单栏，然后顺序按下与高亮显示或有下划线的菜单名和命令名对应的键。对于有些命令，也可以按下菜单中列出的组合键。
- 可使用键盘选择或清除对话框中的复选框或选项按钮。按 Alt 键，然后再按与选项名的有下划线的字母相对应的键。或可以按 Tab 键直到选项被高亮显示，然后按空格键来选择或清除复选框或选项按钮。
- 通过按 Esc 键可取消对话框的显示。

各 章 综 述

本书结合提示、操作步骤和复习题，介绍如何用 Visual Basic .NET 和 Visual C# .NET 创建基于 Windows 的应用程序。一般来说，最好按本书设计的先后顺序学习，但是您也可以根据自定的学习计划，仅学习自己感兴趣的部分，相关详细信息请参阅附录 B“考试重点索引”。如果您按照自己制定的学习计划来学习，需要参阅每章中的“先决条件”。任何需要前面章节知识的操作步骤均请参考适当的章节。

本书划分为以下章节：

- 前言部分包含自学概述，并介绍了本书的构成。仔细阅读此部分，以便充分利用本书，并确定自己需要学习的章节。
- 第 1 章介绍了.NET 框架、公共语言运行库和.NET 基类库。描述了如何使用垃圾回收进行内存管理。还说明了如何创建类和结构以及如何实现及限制方法等。
- 第 2 章描述了用户界面的元素以及创建用户界面的步骤。解释了控件、菜单、窗体和控件事件的使用以及用户输入验证等内容。
- 第 3 章详细说明了如何实现和使用自定义类型。还介绍了如何创建数组和集合，以及如何实现属性和事件等内容。

- 第 4 章介绍了面向对象编程的概念。描述了如何创建重载成员以及如何通过继承和接口实现来实施多态。
- 第 5 章介绍了如何使用 Visual Studio .NET 提供的工具来调试应用程序。说明了如何使用 Trace 类和 Debug 类从应用程序中获得实时反馈，以及如何抛出和处理异常。此外，还讲述了如何为应用程序开发单元测试方案。
- 第 6 章详细说明了如何使用 ADO.NET 从不同的源中访问数据。还介绍了连接的和断开的数据访问，以及将数据绑定到用户界面等内容。此外，还提供了 XML 数据主题的采样。
- 第 7 章介绍了如何使用.NET 框架来创建用户控件、自定义控件和继承控件。还介绍了如何利用 GDI+技术来呈现可视元素，以及如何利用控件完成一些常见任务。
- 第 8 章介绍了一些高级的开发主题。说明了如何创建本地化应用程序，以及如何在应用程序中实现打印。还讲述了如何访问 COM 组件、Windows API 和 Web Services 等。此外，还说明了如何为应用程序实现帮助和设置可访问性属性。
- 第 9 章说明了如何创建程序集和使用资源。讲述了如何从配置文件检索数值并且在应用程序中使用它们。还说明了如何在应用程序中实现基于角色的安全和代码访问安全。
- 第 10 章说明了如何为应用程序创建和配置一个安装项目，以及如何为程序规划部署方案。
- 附录 A 列出了本书中所有的复习题。
- 术语表提供了本书中很多术语和概念的定义。

课 前 准 备

本自学的制胜宝典包含了帮助您学习使用 Visual Basic .NET 和 Visual C# .NET 开发基于 Windows 的应用程序的动手练习。

硬件要求

每台计算机都必须具有以下的最低配置。所有硬件都必须是 Microsoft Windows XP 或 Microsoft Windows 2000 硬件兼容性列表中的硬件。

- Pentium II 级处理器，450 MHz。
- 160 MB 物理内存，推荐 256 MB。
- 光驱或 DVD 光驱，推荐 12 倍以上光驱。

注意 安装 Visual Studio .NET Professional Evaluation Edition 软件需要 DVD 光驱。

- 安装硬盘上要有 3.5 GB 可用磁盘空间，包括系统驱动器的 500 MB 可用磁盘空间。
- Super VGA(800×600)或 256 色的更高分辨率的显视器。
- 微软鼠标或可兼容的指示设备。

软件要求

完成本书中的练习需要安装以下软件：

- Microsoft Windows 2000 或 Microsoft Windows XP Professional Edition

- Microsoft Visual Studio .NET，Professional Edition、Enterprise Developer Edition、或 Enterprise Architect Edition
- Microsoft Access 2000 或具有 Jet 4.0 数据引擎的更高版本

安装指导

请根据制造商的指导安装您的计算机。

练习文件

配书光盘包括一套练习文件和解决方案文件，您需要将其中的一些文件复制到您的硬盘以完成本书中的练习。

将练习文件安装到硬盘。

1. 将配书光盘插入到光驱中。

注意 如果您机器中的 Autorun 被禁用了，请参阅光盘中的 Readme.txt 文件。

2. 在用户接口菜单中选择 Labs Folder，然后选择要查看的练习文件。如果需要这个练习，将这些文件复制到您硬盘上的工作文件夹中。

考试样题

将考试样题安装到您的硬盘中。

1. 将配书光盘插入光驱中。

注意 如果您机器中的 Autorun 被禁用了，请参阅光盘上的 Readme.txt 文件。

2. 单击用户接口菜单上的 Sample Exam Questions，并且按照提示进行。

支 持 信 息

如果您对本书或配书文件有任何建议、意见或想法，请通过以下电子邮件与清华大学出版社计算机应用编辑二室客户服务部取得联系：

service@wenyuan.com.cn

或致函：

北京 100084-157 信箱

读者服务部

邮编：100084

亦可致电：010-62792098-220。

请注意，上述地址并不提供软件产品的支持。

目　录

前言 VII

第 1 章　.NET 框架简介 1

1.1　.NET 框架和公共语言运行库 1

1.1.1　.NET 框架概述 1

1.1.2　语言和.NET 框架 2

1.1.3　.NET 应用程序的结构 2

1.1.4　.NET 应用程序的编译和执行 3

1.1.5　本节小结 3

1.2　.NET 基类库 4

1.2.1　引用类型和值类型 5

1.2.2　在应用程序中使用.NET 框架类型 6

1.2.3　Imports 语句和 using 语句 8

1.2.4　引用外部库 9

1.2.5　本节小结 9

1.3　使用类和结构 10

1.3.1　成员 10

1.3.2　创建类 10

1.3.3　创建结构 11

1.3.4　添加成员 11

1.3.5　嵌套类型 12

1.3.6　实例化用户定义类型 12

1.3.7　类与结构的对比 13

1.3.8　本节小结 14

1.4　使用方法 14

1.4.1　添加方法 14

1.4.2　调用方法 15

1.4.3　方法变量 15

1.4.4　参数 16

1.4.5　构造函数和析构函数 18

1.4.6　本节小结 19

1.5　作用域和访问级别 20

1.5.1　成员访问修饰符 20

1.5.2　类型访问修饰符 22

1.5.3　嵌套类型的访问修饰符 22

1.5.4　Shared(static)成员 22

1.5.5　本节小结 24

1.6　垃圾回收 24

1.6.1　循环引用 25

1.6.2　本节小结 26

1.7　实验 1：类和垃圾回收 26

1.7.1　练习 1.1：制作 Demo 类 27

1.7.2　练习 1.2：演示垃圾回收 28

1.8　本章复习 29

第 2 章　创建用户界面 30

2.1　用户界面设计原则 30

2.1.1　窗体、控件和菜单 31

2.1.2　布局 31

2.1.3　本节小结 33

2.2　使用窗体 33

2.2.1　给项目添加窗体 34

2.2.2　可视化继承 34

2.2.3　设置起始窗体 35

2.2.4　设置起始位置 37

2.2.5　改变窗体外观 37

2.2.6　BackColor、ForeColor 和 Test 属性 38

2.2.7　Font、Cursor 和 BackGroundImage 38

2.2.8　Opacity 38

2.2.9　使用窗体方法 39

2.2.10　Show 和 ShowDialog 39

2.2.11　Activate 40

2.2.12　Hide 40

2.2.13 Close 40
2.2.14 使用窗体事件 41
2.2.15 窗体生命周期事件 42
2.2.16 本节小结 44
2.3 使用控件和组件 44
2.3.1 使用控件 45
2.3.2 设置控件 Tab 键顺序 46
2.3.3 可包含其他控件的控件 46
2.3.4 停靠和锚定控件 48
2.3.5 使用控件集合 50
2.3.6 将控件添加到 Toolbox 51
2.3.7 为控件创建事件处理程序 51
2.3.8 使用扩展程序提供程序组件 52
2.3.9 本节小结 53
2.4 使用菜单 54
2.4.1 在设计时创建菜单 54
2.4.2 使用 MainMenu 组件 54
2.4.3 分隔菜单项 55
2.4.4 菜单访问键和快捷键 56
2.4.5 使用菜单项事件 57
2.4.6 创建上下文菜单 57
2.4.7 在运行时修改菜单 57
2.4.8 启用和禁用菜单命令 58
2.4.9 显示菜单项上的复选标记 58
2.4.10 显示菜单项上的单选按钮 58
2.4.11 使菜单项不可视 58
2.4.12 复制菜单 59
2.4.13 在运行时合并菜单 59
2.4.14 在运行时添加菜单项 59
2.4.15 本节小结 60
2.5 验证用户输入 61
2.5.1 字段级验证 61
2.5.2 在字段级验证中使用事件 62
2.5.3 处理焦点 64
2.5.4 窗体级验证 66
2.5.5 提供用户反馈 67
2.5.6 本节小结 69
2.6 实验 2：虚拟油炸圈饼坊 69
2.6.1 练习 2.1：创建用户界面 70
2.6.2 练习 2.2：添加菜单 73
2.6.3 练习 2.3：创建验证处理程序 74
2.7 本章复习 76
第 3 章 类型和成员 77
3.1 使用数据类型 77
3.1.1 .NET 数据类型 78
3.1.2 类型转换 80
3.1.3 使用数据类型功能 82
3.1.4 本节小结 84
3.2 常量、枚举、数组和集合 85
3.2.1 常量和枚举 85
3.2.2 数组 88
3.2.3 集合 91
3.2.4 枚举一个数组或集合中的成员 94
3.2.5 本节小结 95
3.3 实现属性 96
3.3.1 实现属性 96
3.3.2 只读或只写属性 98
3.3.3 参数化属性 99
3.3.4 默认的属性和索引程序 100
3.3.5 集合属性 101
3.3.6 本节小结 103
3.4 实现委托和事件 103
3.4.1 委托 104
3.4.2 声明和引发事件 105
3.4.3 实现事件处理程序 106
3.4.4 处理多个事件的事件处理程序 107
3.4.5 有多个处理程序的事件 108
3.4.6 在运行时删除处理程序 108
3.4.7 本节小结 109
3.5 实验 3：添加组件并实现成员 109
3.5.1 练习 3.1：创建 Doughnut Machine 组件 110
3.5.2 练习 3.2：将 DoughnutMachine 添加到用户界面 116
3.6 本章复习 119

第 4 章 面向对象编程与多态 121
4.1 面向对象编程简介 121
4.1.1 对象、成员和抽象 121
4.1.2 封装 123
4.1.3 多态 123
4.1.4 本节小结 124
4.2 重载成员 124
4.2.1 创建重载方法 125
4.2.2 用 Visual C#重载运算符 126
4.2.3 本节小结 128
4.3 接口多态 128
4.3.1 定义接口 128
4.3.2 使用接口的多态 130
4.3.3 实现接口 131
4.3.4 本节小结 133
4.4 继承多态 133
4.4.1 继承 134
4.4.2 继承的成员 135
4.4.3 抽象类和成员 140
4.4.4 本节小结 143
4.5 实验 4： 使用继承类 143
4.5.1 练习 4.1：通过扩展 CollectionBase 创建一个强类型集合类 143
4.5.2 练习 4.2：实现附加的油炸圈饼类型 146
4.5.3 练习 4.3：实现 RemoveStale 方法 148
4.6 本章复习 152
第 5 章 测试和调试应用程序 153
5.1 使用调试工具 153
5.1.1 错误的类型 153
5.1.2 Break 模式 155
5.1.3 设置断点 157
5.1.4 使用调试窗口 158
5.1.5 本节小结 161
5.2 使用 Debug 和 Trace 类 161
5.2.1 跟踪的工作方式 162
5.2.2 写 Trace 和 Debug 输出 162
5.2.3 Listeners 集合 164
5.2.4 使用 Trace 开关 166
5.2.5 配置 Trace 开关 167
5.2.6 本节小结 169
5.3 创建单元测试计划 169
5.3.1 单元测试计划 169
5.3.2 本节小结 172
5.4 处理和抛出异常 173
5.4.1 如何处理异常 173
5.4.2 创建一个异常处理程序 173
5.4.3 抛出异常 177
5.4.4 本节小结 180
5.5 实验 5：调试应用程序 180
5.6 本章复习 183
第 6 章 使用 ADO. NET 的数据访问 184
6.1 ADO.NET 概述 184
6.1.1 非相连数据库访问 184
6.1.2 ADO.NET 数据体系结构 185
6.1.3 本节小结 187
6.2 访问数据 187
6.2.1 连接到数据库 188
6.2.2 使用数据命令 190
6.2.3 使用 DataReaders 194
6.2.4 创建和配置 DataAdapter 199
6.2.5 用 DataAdapter 获取数据 201
6.2.6 本节小结 205
6.3 使用 DataSet 对象和更新数据 205
6.3.1 不用 DataAdapter 创建和填充 DataSet 对象 206
6.3.2 DataRelation 对象 209
6.3.3 约束 211
6.3.4 编辑和更新数据 212
6.3.5 本节小结 218
6.4 绑定、查看和筛选数据 218
6.4.1 数据绑定 218
6.4.2 筛选数据和排序数据 224
6.4.3 本节小结 228
6.5 在 ADO.NET 中使用 XML 229

6.5.1 从 SQL Server 2000 数据库获取 XML 229
6.5.2 对 DataSet 使用 XML 230
6.5.3 使用 XmlDataDocument 类 231
6.5.4 本节小结 234
6.6 实验 6：与数据库连接 234
6.6.1 练习 6.1：添加数据访问及使用 DataReader 235
6.6.2 练习 6.2：使用 DataAdapter 对象和 DataSet 获取和更新数据 236
6.6.3 练习 6.3：用 XML Designer 创建一个类型化 DataSet 238
6.6.4 练习 6.4：使用 Data Form 向导 239
6.7 本章复习 242
第 7 章 使用.NET 框架创建控件 243
7.1 使用 GDI+ 243
7.1.1 System.Drawing 命名空间 244
7.1.2 Graphics 对象 244
7.1.3 颜色、画笔和笔 247
7.1.4 呈现简单的形状 249
7.1.5 呈现文本 250
7.1.6 呈现复杂的形状 251
7.1.7 本节小结 253
7.2 控件创作 253
7.2.1 控件创作概述 254
7.2.2 创建继承控件 256
7.2.3 创建用户控件 258
7.2.4 创建自定义控件 259
7.2.5 本节小结 261
7.3 使用控件的公共任务 261
7.3.1 将控件添加到 Toolbox 261
7.3.2 为控件提供 Toolbox 位图 262
7.3.3 调试控件 263
7.3.4 管理控件授权 264
7.3.5 将控件宿主在 Internet Explorer 中 265
7.3.6 本节小结 266
7.4 实验 7：创建自定义控件 266
7.4.1 练习 7.1：创建控件 266
7.4.2 练习 7.2：测试控件 270
7.5 本章复习 271
第 8 章 高级.NET 框架主题 272
8.1 实现打印功能 272
8.1.1 PrintDocument 组件 273
8.1.2 打印内容 274
8.1.3 使用 PrintPreviewControl 279
8.1.4 配置打印 280
8.1.5 本节小结 282
8.2 访问和调用组件 283
8.2.1 访问.NET 和 COM 类型库 283
8.2.2 实例化 ActiveX 控件 284
8.2.3 访问 Web Service 284
8.2.4 访问 Windows API 288
8.2.5 本节小结 288
8.3 实现可访问性 289
8.3.1 可访问性设计 289
8.3.2 可访问性和 Windows 程序 Certified 290
8.3.3 Windows Forms 控件的可访问性属性 291
8.3.4 本节小结 291
8.4 在应用程序中实现帮助 292
8.4.1 Help 类 292
8.4.2 HelpProvider 组件 293
8.4.3 本节小结 294
8.5 全局化和本地化 294
8.5.1 全局化和本地化 294
8.5.2 特定区域性格式设置 298
8.5.3 本节小结 302
8.6 实验 8：使用打印支持创建本地化窗体 302
8.6.1 练习 8.1：创建窗体 302
8.6.2 练习 8.2：本地化窗体 305
8.6.3 练习 8.3：添加打印支持 308
8.7 本章复习 310

第 9 章 程序集、配置和安全机制 311
9.1 程序集与资源 311
9.1.1 程序集 312
9.1.2 资源和资源程序集 313
9.1.3 共享程序集 318
9.1.4 本节小结 320
9.2 配置和优化应用程序 320
9.2.1 创建配置文件 321
9.2.2 使用动态属性配置应用程序 322
9.2.3 优化应用程序性能 325
9.2.4 本节小结 326
9.3 保护应用程序 327
9.3.1 权限 327
9.3.2 配置基于角色的授权 328
9.3.3 配置代码访问安全 332
9.3.4 使用具有强制安全的异常处理 337
9.3.5 本节小结 337
9.4 实验 9：配置和设置应用程序安全 338
9.4.1 练习 9.1：添加配置文件 338
9.4.2 练习 9.2：保护应用程序 339
9.5 本章复习 341
第 10 章 部署应用程序 343
10.1 规划项目的部署方案 343
10.1.1 XCOPY 部署 344
10.1.2 创建安装项目 344
10.1.3 配置安装项目的生成属性 346
10.1.4 生成、分布和部署安装项目 349
10.1.5 本节小结 351
10.2 配置安装项目 351
10.2.1 设置 Setup Project 属性 352
10.2.2 安装编辑器 353
10.2.3 安装本机程序集图像 361
10.2.4 验证已安装程序集的安全策略 361
10.2.5 本节小结 362
10.3 实验 10：创建一个安装程序 362
10.3.1 练习 10.1：创建安装项目 362
10.3.2 练习 10.2：配置应用程序 363
10.3.3 练习 10.3：安装应用程序 364
10.4 本章复习 364
附录 A 问题与答案 365
附录 B 考试重点索引 373
附录 C 微软认证专家计划 375
术语表 379

第 1 章　.NET 框架简介

1.1　.NET 框架和公共语言运行库 1
1.2　.NET 基类库 4
1.3　使用类和结构 10
1.4　使用方法 14
1.5　作用域和访问级别 20
1.6　垃圾回收 24
1.7　实验 1：类和垃圾回收 26
1.8　本章复习 29

本章概要

本章讨论了 Microsoft .NET 框架和公共语言运行库，并介绍了类、结构和方法声明的语法。

先决条件

完成本章各节的学习，无先决条件。

1.1　.NET 框架和公共语言运行库

.NET 框架是一个用于代码开发和执行的集成和托管的环境。本节介绍了.NET 框架、其基本原理以及工作方式等。

本节学习目标

- 了解.NET 框架的元素
- 了解程序集的组成部分以及识别每部分中包含的内容
- 了解如何编译和执行.NET 应用程序

估计学习时间：20 分钟

1.1.1　.NET 框架概述

对应用程序开发和执行来说，.NET 框架是一个托管的、类型安全的环境。它管理程序执行的各个方面：为数据和指令的存储分配内存；授予或拒绝应用程序的适当权限；初始化并管理应用程序执行；对不再需要的资源管理内存的重新分配。.NET 框架由两个主要组件构成：公共语言运行库和.NET 框架类库。

公共语言运行库可被认为是管理代码执行的环境。它提供了多种核心服务，例如代码编译、内

存分配、线程管理以及垃圾回收等。通过公共类型系统(CTS)，公共语言运行库加强了严格的类型安全，并通过加强代码访问安全来确保在安全的环境中执行代码。

.NET 框架类库提供了一些有用的和可重用的类型的集合。设计这些类型的目的是与公共语言运行库相集成。.NET 框架提供的类型是面向对象的并可以完全扩展的，并允许应用程序与.NET 框架无缝集成。.NET 基类库将在 1.2 节中进一步讨论。

1.1.2 语言和.NET 框架

设计.NET 框架的目的，是为了实现跨语言的兼容性。简单地说，也就是无论.NET 组件最初是用哪种语言编写的，它们都可以彼此交互。这样，一个用 Visual Basic .NET 编写的应用程序可以引用一个用 C#编写的 DLL 文件，而这个 DLL 文件又可以访问用托管 C++或者任何其他.NET 语言编写的资源。这种语言的互操作性扩展到全部的面向对象的继承。例如，一个 Visual Basic .NET 类可从 C#类继承而来，反之亦然。

由于有了公共语言运行库，实现这样好的跨语言兼容性才成为可能。当一个.NET 应用程序被编译时，它被从编程时所使用的语言(Visual Basic .NET、C#或其他.NET 兼容语言)转化成 **Microsoft 中间语言**(MSIL 或 IL)。这是一种可被公共语言运行库读取和理解的低级语言。由于所有.NET 可执行文件和 DLL 文件都作为中间语言而存在，因而它们可以自由地相互操作。**公共语言规范**定义了.NET 语言编译器必须遵守的最低标准，这样就可确保由.NET 编译器编译的任何源代码都能与.NET 框架互操作。

CTS 确保了.NET 组件之间的类型兼容性。由于在部署和执行以前，.NET 应用程序被转化为 IL，因此所有的基元数据类型都表示为.NET 类型。这样，Visual Basic 中的 Integer 和 C#中的 int 在 IL 代码中都将表示为 System.Int32。因为这两种语言都使用了一个公共的并且可以相互转化的类型系统，就有可能在组件之间传递数据，并且避免了耗时的转化或出现难以查找的错误。

Visual Studio .NET 配备了如 Visual Basic .NET、Visual C#、具有托管扩展的 Visual C++以及 JScript 脚本语言。也可用其他语言为.NET 框架编写托管代码。针对 Fortran .NET、Cobol .NET、Perl .NET 和一些其他语言，存在第 3 方编绎器。所有这些语言共享相同的跨语言兼容性和继承性。这样，可以选择自己喜爱的语言为.NET 框架编写代码，该代码将能够与用其他任何语言为.NET 框架编写的代码交互。

1.1.3 .NET 应用程序的结构

要理解公共语言运行库如何管理代码的执行，必须先分析.NET 应用程序的结构。.NET 应用程序的主要单元是**程序集**。程序集是代码、资源和元数据的自描述集合。**程序集清单**包含程序集内部构成的信息。程序集清单提供了：

- 标识信息，例如程序集的名称和版本号。
- 程序集公开的所有类型的列表。
- 程序集需要的其他程序集的列表。
- 程序集的代码访问安全指令列表。它包括程序集需要的权限列表和程序集拒绝的权限列表。

每个程序集有且仅有一个程序集清单，它包含程序集所有的描述信息。程序集清单可包含于它

自己的独立文件中，它也可以包含于一个程序集**模块**之内。

程序集也包含一个或多个模块。模块包含组成应用程序或库的代码，以及描述这种代码的元数据。当将项目编译为程序集时，代码就从高级代码转化成 IL。由于所有托管代码首先被转化为 IL 代码，用不同语言编写的应用程序就可以轻松地交互。例如，开发人员可以用 Visual C#编写一个应用程序来访问用 Visual Basic .NET 编写的 DLL。这两种资源在被执行以前都将被转化为 IL 模块，这样可以避免任何语言不兼容性问题。

每个模块也包含很多类型。类型是描述一组数据封装和功能的模板。类型有两种：**引用类型**(类)和**值类型**(结构)。这些类型将在 1.3 节中更加详细地讨论。程序集清单向公共语言运行库描述每一种类型。一个类型可包含字段、属性和方法，其中每一个都应与一个公共功能相关联。例如，可能有一个代表银行账户的类。它将包含与实现银行账户所需功能相关联的字段、属性和方法。字段表示一种特定类型数据的存储。字段可能存储一个账户持有者的姓名。属性与字段相似，但在数据被设置或检索时，属性通常提供某种验证。属性可能表示账户可用的余额。当试图修改这个值时，属性能够检验此修改是否大于预定的限制，如果确实如此，就不允许修改此值。方法代表行为，例如对类中存储的数据采取动作，或者对用户界面作出更改。继续以银行账户为例，您可能有 Transfer 方法要将余额从经常账户传输到储蓄账户，或者 Alert 方法在余额低于预定水平时向用户发出警告等。

1.1.4 .NET 应用程序的编译和执行

当编译.NET 应用程序时，应用程序不是被编译成二进制机器代码，而是被转换成 IL。IL 是一个能被公共语言运行库理解的低级别的指令集。这是部署应用程序所采用的形式——一个或多个由 IL 形式的可执行文件和 DLL 文件组成的程序集。这些程序集中至少有一个将包含设计为应用程序的入口点的可执行文件。

当程序开始执行时，第一个程序集被载入内存。在这一点上，公共语言运行库检查程序集清单并且确定运行该程序的前提要求。公共语言运行库检查程序集请求的安全权限，并且将它们与系统的安全策略相比较。如果系统的安全策略不允许被请求的权限，应用程序将不会运行。如果应用程序通过了系统的安全策略，公共语言运行库就执行代码。它为应用程序创建了一个在其中运行的进程，并启动应用程序执行。当执行开始时，需要被执行的代码的第 1 位被载入内存，并被公共语言运行库的 Just-In-Time (JIT，实时)编译器从 IL 编译为本地二进制代码。代码一旦被编译，就开始执行并且作为本地代码存储于内存中，因此，代码的每一部分在应用程序执行的过程中仅编译一次。程序执行无论何时扩展到未被执行的代码，JIT 编译器将在执行前编译它并且将它作为二进制代码存储在内存中。这样，由于只有被执行的程序部分被编译，应用程序的性能就得到了最大化。

1.1.5 本节小结

- .NET 框架是软件开发的基础。.NET 框架由公共语言运行库和.NET 基类库组成。公共语言运行库提供了很多程序执行所需的核心服务，.NET 基类库则公开了一套预先开发的类以促进程序开发。公共语言规范(Common Language Specification)定义了所有使用.NET 框架的语言必须支持的最小的标准集，CTS 确保了用不同语言开发的组件之间的类型兼容性。
- .NET 应用程序的主要单元是程序集，它由一个程序集清单组成。这个程序集清单描述了程序集和一个或多个模块，它们包含了应用程序的源代码。
- .NET 可执行文件作为 IL 文件存储。在加载时，会根据本地系统的安全策略检查程序集。

如果它被允许运行，第一个程序集就被载入内存，并被实时(JIT)编译为本地二进制代码，并就地存储，等待随后的程序执行。

1.2 .NET 基类库

.NET 基类库是一个面向对象的类型和接口的集合，它为您将面临的很多复杂的编程任务提供了对象模型和服务。.NET 基类库提供的大多数类型是完全可扩展的，使您可以生成类型，将自己的功能合并到您的托管代码之中。在本节中，将介绍一些.NET 基类库命名空间，以及如何引用库并使用它的类型和方法。

本节学习目标

- 描述.NET 基类库
- 了解值类型和引用类型之间的区别
- 创建对命名空间的引用
- 创建.NET 框架的类和值类型的实例

估计学习时间：30 分钟

.NET 框架基类库包含许多基类。这些基类提供了编写应用程序时需要的很多服务和对象。类库被组织到**命名空间**中。命名空间是执行相关功能的类型的逻辑分组。例如，System.Windows.Forms 命名空间包含所有组成 Windows 窗体和在窗体中使用的控件的类型。

命名空间是相关类的逻辑分组。.NET 基类库中的命名空间是按照层次关系组织的。.NET 框架的根(Root)是 System 命名空间。其他命名空间可用.(句点)运算符访问。典型的命名空间的结构如下所示：

```
System
System.Data
System.Data.SQLClient
```

第 1 个例子指 System 命名空间。第 2 个例子指 System.Data 命名空间。第 3 个例子指 System.Data.SQLClient 命名空间。表 1.1 介绍了一些更通用的.NET 基类命名空间。

表 1.1 一些有代表性的.NET 命名空间

命名空间	说 明
System	该命名空间是.NET 框架需要的很多低级别类型的根，而且也是基元数据类型的根。它还是.NET 基类库中所有其他命名空间的根
System.Collections	该命名空间包含表示很多不同容器类型的类，例如 ArrayList、SortedList、Queue 和 Stack 等。也可找到对实现自己的集合功能非常有用的抽象类(例如 CollectionBase)
System.ComponentModel	该命名空间包含涉及组件创建和包容的类，例如特征、类型转换器和许可提供程序等
System.Data	该命名空间包含执行数据库访问和操作所需要的类以及用于数据访问的附加命名空间
System.Data.Common	该命名空间包含一组由.NET 托管数据提供程序共享的类
System.Data.OleDb	该命名空间包含的类，用于为 OLE DB 数据访问编写托管数据提供程序