

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual C#.NET 程序设计教程

Visual C#.NET Programming

罗福强 白忠建 杨剑 编著

- 知识结构完整，根据认识规律编写内容
- 实例完整丰富，帮助读者理解程序算法
- 全书配备实训，真正培养学生编程能力



高校系列



人民邮电出版社
POSTS & TELECOM PRESS

Visual C#.NET Programming

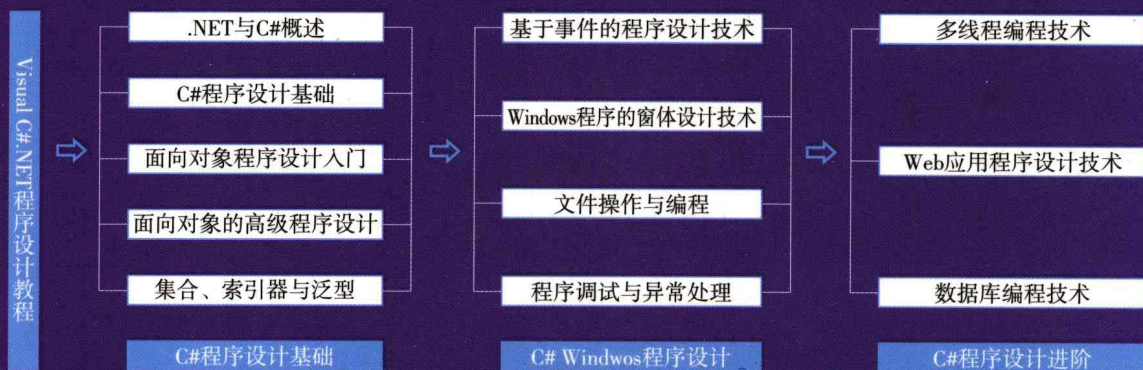
Visual C#.NET 程序设计教程

本书融入了全体编者的教学和项目开发经验,以及微软认证的心得体会。在本书编写过程中,编写思路如下:第一,面向应用型本科院校学生,立足于把C#的语法讲清楚,文字叙述要简练;第二,以基础为主,主要介绍Visual C# .NET的语法基础及面向对象程序设计思想和方法;第三,辅以适当的实例,以提高学习兴趣和编程技能,实例以Windows程序为主,贴近学生的实际生活;第四,学生可以在没有C/C++基础的情况下使用本书;第五,坚持以应用为纲,最后以完整的实例展现了Visual C# .NET应用程序的开发过程。

本书以Visual Studio .NET 2005和Visual C# .NET 2.0为蓝本。全书共分为12章,基本上覆盖了C#的主要领域,在讲解Visual C# .NET语法的基础上,以面向对象的程序设计方法和基于事件的程序设计方法为核心,重点阐述Windows程序设计、Web应用程序设计、数据库编程、文件操作与编程等实用技术。

本书可作为高等院校Visual C# .NET课程的教材或参考资料,也可供软件开发人员参考使用。

本教材的结构框图



免费提供

PPT等教学相关资料



人民邮电出版社
教学服务与资源网
www.ptpedu.com.cn

教材服务热线: 010-67170985

反馈/投稿/推荐信箱: 315@ptpress.com.cn

人民邮电出版社教学服务与资源网: www.ptpedu.com.cn



ISBN 978-7-115-19394-0



9 787115 193940 >

ISBN 978-7-115-19394-0/TP

定价: 32.00 元

封面设计: 董志桢

人民邮电出版社网址: www.ptpress.com.cn

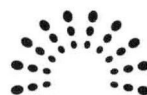
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual C#.NET 程序设计教程

Visual C#.NET Programming

罗福强 白忠建 杨剑 编著



高校系列

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Visual C#.NET程序设计教程 / 罗福强, 白忠建, 杨剑
编著. —北京: 人民邮电出版社, 2009. 3
21世纪高等学校计算机规划教材
ISBN 978-7-115-19394-0

I. V… II. ①罗…②白…③杨… III. C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 (2009) 第003056号

内 容 提 要

Visual C#.NET 是微软公司推出的新一代编程语言。本书共分为 12 章, 在讲解 Visual C#.NET 语法的基础上, 以面向对象的程序设计方法和基于事件的程序设计方法为核心, 重点阐述 Windows 程序设计、Web 应用程序设计、数据库编程、文件操作与编程等实用技术。本书内容丰富, 可操作性强, 叙述简洁流畅, 没有晦涩的术语, 大量的实例, 能够使学生轻松、愉快地掌握 Visual C#.NET 程序设计的基本思想、方法和技巧。

本书可作为高等院校计算机相关专业学生的教材, 也可作为初、中级读者和培训班学员学习的参考用书。

21 世纪高等学校计算机规划教材

Visual C#.NET 程序设计教程

-
- ◆ 编 著 罗福强 白忠建 杨 剑
责任编辑 蒋 亮
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京楠萍印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 18.75
字数: 487 千字
印数: 1—3 000 册
- 2009 年 3 月第 1 版
2009 年 3 月北京第 1 次印刷

ISBN 978-7-115-19394-0/TP

定价: 32.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

出版者的话

现今社会对人才的基本要求之一就是应用计算机的能力。在高等学校，培养学生应用计算机的能力，主要是通过计算机课程的体制改革，即计算机教学分层、分类规划与实施；密切联系实际，恰当体现与各专业其他课程配合；教学必须以市场需求为导向，目的是培养高素质创新型人才。

人民邮电出版社经过对教学改革新形势充分的调查研究，依据目前比较成熟的教学大纲，组织国内优秀的有丰富教学经验的教师编写一套体现教学改革最新形势的“高校系列计算机教材”。在本套教材的出版过程中，我社多次召开教材研讨会，广泛听取了一线教师的意见，也邀请众多专家对大纲和书稿做了认真的审读与研讨。本套教材具有以下特点。

1. 覆盖面广，突出教改特色

本套教材主要面向普通高等学校（包括计算机专业和非计算机专业），是在经过大量充分的调研基础上开发的计算机系列教材，涉及计算机教育领域中的所有课程（包括专业核心骨干课程与选修课程），适应了目前经济、社会对计算机教育的新要求、新动向，尤其适合于各专业计算机教学改革的特点特色。

2. 注重整体性、系统性

针对各专业的特点，同一门课程规划了组织结构与内容不同的几本教材，以适应不同教学需求，即分别满足不同层次计算机专业与非计算机专业（如工、理、管、文等）的课程安排。同时本套教材注重整体性的策划，在教材内容的选择上避免重叠与交叉，内容系统完善。学校可根据教学计划从中选择教材的各种组合，使其适合本校的教学特点。

3. 掌握基础知识，侧重培养应用能力

目前社会对人才的需要更侧重于其应用能力。培养应用能力，须具备计算机基础理论、良好的综合素质和实践能力。理论知识作为基础必须掌握，本套教材通过实践教学与实例教学培养解决实际问题的能力和知识综合运用能力。

4. 教学经验丰富的作者队伍

高等学校在计算机教学和教材改革上已经做了大量的工作，很多教师在计算机教育与科研方面积累了相当多的宝贵经验。本套教材均由有丰富教学经验的教师编写，并将这些宝贵经验渗透到教材中，使教材独具特色。

5. 配套资源完善

所有教材均配有 PPT 电子教案，部分教材配有实践教程、题库、教师手册、学习指南、习题解答、程序源代码、演示软件、素材、图书出版后要更新的内容等，以方便教与学。

我社致力于优秀教材的出版，恳请大家在使用的过程中，将发现的问题与提出的意见反馈给我们，以便再版时修改。

人民邮电出版社

前 言

Visual C# .NET 是微软公司推出的新一代编程语言。它在保持了 C++强大功能的同时，整合了 Java 语言的所有优点，是一种全新的完全面向对象的开发语言，是简单、功能强大、安全而灵活的程序设计语言，是一种能够快速开发应用软件的程序设计语言。它解决了存在于许多程序设计语言中的问题，如安全问题、可靠性问题、与其他语言协调的能力、跨平台的兼容性等。相对于 C++来说，Visual C# .NET 更容易理解和接受；相对于 Java 来说，Visual C# .NET 更容易使用，开发软件的效率更高。Visual C# .NET 既能用于开发传统的 DOS 应用程序、Windows 应用程序和组件程序，又能开发 Web 应用程序和 XML Web 服务等。正是因为 Visual C# .NET 与 Web 紧密结合，使得它成为目前最好的 Web 应用程序设计语言。

本书以 Visual Studio .NET 2005 和 Visual C# .NET 2.0 为蓝本。全书共分为 12 章，基本上覆盖了 C#的主要领域，在讲解 Visual C# .NET 语法的基础上，以面向对象的程序设计方法和基于事件的程序设计方法为核心，重点阐述 Windows 程序设计、Web 应用程序设计、数据库编程、文件操作与编程等实用技术。

本书在编写时遵循 5 点基本编写思路：第一，面向应用型本科院校学生，立足于把 C#的语法讲清楚，文字叙述要简练；第二，以基础为主，主要介绍 Visual C# .NET 的语法基础及面向对象程序设计思想和方法；第三，辅以适当的实例，以提高学习兴趣和编程技能，实例以 Windows 程序为主，实例要贴近学生的实际生活；第四，学生可以在没有 C/C++基础的情况下使用本书；第五，坚持以应用为纲，最后以完整的实例展现了 Visual C# .NET 应用程序的开发过程。

本书具有以下 3 个特点：第一，知识结构完整，根据循序渐进的认识规律设计编写内容及顺序；第二，提供了大量的实例，所有实例程序都是完整的，都是通过 Visual Studio .NET 2005 调试的，并给出了运行效果，其中部分复杂的实例还有详细的分析，以帮助读者理解程序算法并学会程序设计；第三，全书配备了丰富的、符合教学实际的、能真正培养学生编程能力的实训任务。

本书可作为高等院校 Visual C# .NET 课程的教材或参考资料，也可供软件开发人员参考使用。

本书由电子科技大学成都学院罗福强、白忠建、杨剑、罗佳和范畅等编写。本书融入了全体编者的教学和项目开发经验，以及微软认证教学的心得体会。罗福强编写第 2、3、4、5、11 章，白忠建编写第 1、6 章，杨剑编写第 7、12 章，罗佳编写第 9、10 章，范畅编写第 8 章，最后由罗福强统稿。在本书的编写过程中，电子科技大学成都学院计算机系提供了强有力的技术支持和组织保障，在此特别表示感谢。

由于时间仓促，书中难免有不妥之处，我们殷切地期望读者提出中肯的意见。

编 者

2009 年 2 月

目 录

第 1 章 .NET 与 C#概述..... 1

1.1 .NET 与 C#简介1

1.1.1 .NET 平台简介 1

1.1.2 .NET Framework 2

1.1.3 .NET Framework 的优点 3

1.1.4 C#概述 3

1.2 C#程序的开发环境.....5

1.2.1 Visual Studio.NET 的基本操作..... 5

1.2.2 Visual Studio .NET 解决方案和 项目文件的组织结构 8

1.2.3 Visual Studio.NET 的帮助..... 8

1.3 3 种简单的 C#程序9

1.3.1 一个简单的 C#控制台应用程序..... 10

1.3.2 一个简单的 Win32 应用程序 12

1.3.3 一个简单的 Web 应用程序..... 14

1.3.4 一个具有输入功能的 Win32 应用程序 15

1.3.5 C#程序的特点 18

本章小结.....19

习题.....19

实验 1.....19

第 2 章 C#程序设计基础 21

2.1 C#的值类型21

2.1.1 简单类型 22

2.1.2 枚举型 (enum) 23

2.1.3 结构型 (struct) 24

2.2 常量与变量.....26

2.2.1 常量 26

2.2.2 变量 28

2.2.3 数据类型转换 29

2.3 运算符与表达式.....30

2.3.1 算术运算符与算术表达式 30

2.3.2 赋值运算符与赋值表达式 31

2.3.3 关系运算符与关系表达式 32

2.3.4 逻辑运算符与逻辑表达式32

2.4 C#的引用类型.....33

2.4.1 类 (class)33

2.4.2 接口 (interface)34

2.4.3 委托 (delegate)36

2.4.4 数组 (array)37

2.4.5 字符串 (string)39

2.4.6 装箱和拆箱40

2.5 程序的分支语句41

2.5.1 if 语句41

2.5.2 switch 语句43

2.6 程序的循环语句45

2.6.1 while 语句45

2.6.2 do-while 语句46

2.6.3 for 语句47

2.6.4 foreach 语句49

2.6.5 循环语句的嵌套50

2.6.6 break 与 continue 语句52

本章小结53

习题53

实验 2-154

实验 2-255

第 3 章 面向对象程序设计入门57

3.1 面向对象的基本概念57

3.1.1 对象57

3.1.2 事件与方法58

3.1.3 类与实例58

3.1.4 封装、继承与多态59

3.2 类的定义60

3.2.1 类与类的实例60

3.2.2 类的数据成员63

3.2.3 类的可访问性66

3.2.4 类的嵌套67

3.3 类的方法68

3.3.1 方法的声明与调用68

3.3.2 方法的参数传递	70	5.1.3 集合的创建与操作	117
3.3.3 方法的重载	73	5.2 索引器	121
3.4 构造函数与析构函数	75	5.2.1 索引器的定义与使用	121
3.4.1 对象的生命周期	76	5.2.2 接口中的索引器	123
3.4.2 构造函数	76	5.2.3 索引器与属性的比较	125
3.4.3 析构函数	78	5.3 泛型	126
3.5 静态类与静态成员	79	5.3.1 泛型概述	126
3.5.1 静态类	79	5.3.2 泛型接口、泛型类及泛型方法	127
3.5.2 静态成员	79	本章小结	132
3.5.3 静态构造函数	82	习题	132
本章小结	83	实验 5	132
习题	83		
实验 3-1	84	第 6 章 基于事件的程序设计技术	134
实验 3-2	85	6.1 基于事件的编程思想	134
第 4 章 面向对象的高级程序设计	86	6.2 事件	135
4.1 类的继承性与多态性	86	6.2.1 事件的声明	135
4.1.1 类的继承性	86	6.2.2 事件的订阅	137
4.1.2 类的多态性	89	6.2.3 事件的引用	138
4.2 抽象类与接口	93	6.3 基于事件的 Windows 编程	141
4.2.1 抽象类	93	6.3.1 Windows 应用程序概述	141
4.2.2 接口	96	6.3.2 Windows 窗体与事件驱动编程	143
4.2.3 抽象类与接口的比较	101	本章小结	147
4.3 分部类与命名空间	101	习题	147
4.3.1 分部类的定义	102	实验 6	148
4.3.2 使用命名空间	103		
4.4 委托	105	第 7 章 Windows 程序的窗体设计技术	150
4.4.1 委托的概述	105	7.1 窗体与控件概述	150
4.4.2 委托的声明、实例化与使用	106	7.1.1 Windows 窗体	150
4.4.3 委托中的协变与逆变	108	7.1.2 窗体的控件	151
4.4.4 多路广播与委托的组合	111	7.2 按钮、标签与文件框控件	154
本章小结	112	7.2.1 按钮控件	154
习题	113	7.2.2 标签控件	156
实验 4-1	113	7.2.3 文本框控件	156
实验 4-2	114	7.2.4 用户登录实例	158
第 5 章 集合、索引器与泛型	115	7.3 选择控件	160
5.1 集合	115	7.3.1 RadioButton 控件	160
5.1.1 集合概述	115	7.3.2 CheckBox 控件	161
5.1.2 选择集合类	116	7.3.3 ListBox 控件	162

7.3.4	ComboBox 控件	163	8.2.3	读写二进制文件	201
7.3.5	应用实例: 添加学生信息	163	8.2.4	对象的序列化	203
7.4	容器控件	166	8.3	文件操作控件	205
7.4.1	GroupBox 控件	166	8.3.1	OpenFileDialog 控件	205
7.4.2	Panel 控件	166	8.3.2	SaveFileDialog 控件	207
7.4.3	TabControl 控件	167	8.3.3	FolderBrowseDialog 控件	209
7.4.4	应用实例: 添加课程信息	167	本章小结		210
7.5	PictureBox 控件	170	习题		211
7.5.1	PictureBox 控件	170	实验 8		211
7.5.2	应用实例: 添加“关于”窗体	170			
7.6	菜单、工具栏和状态栏	171	第 9 章 程序调试与异常处理		216
7.6.1	菜单	172	9.1	程序错误	216
7.6.2	工具栏	173	9.1.1	程序错误分类	216
7.6.3	状态栏	174	9.1.2	调试程序错误	218
7.6.4	应用实例: 添加主窗体	176	9.2	程序的异常处理	220
7.7	对话框	178	9.2.1	异常处理的概念	220
7.7.1	模态对话框	178	9.2.2	使用 try-catch 语句处理异常	221
7.7.2	非模态对话框	180	9.2.3	使用 finally 语句	223
7.7.3	消息框	181	9.2.4	使用 throw 语句抛出异常	224
7.7.4	通用对话框	182	本章小结		225
7.7.5	应用实例: 一个简单的文本编辑器	184	习题		225
7.8	SDI 和 MDI 应用程序	186	第 10 章 多线程编程技术		226
7.8.1	创建 MDI 应用程序	186	10.1	多线程的概念	226
7.8.2	应用实例: 创建学生成绩管理系统界面	187	10.1.1	什么是线程	226
本章小结		188	10.1.2	单线程和多线程	227
习题		188	10.2	线程创建与控制	228
实验 7-1		188	10.2.1	创建和启动线程	228
实验 7-2		191	10.2.2	控制线程	228
第 8 章 文件操作与编程		196	10.3	多线程的同步	232
8.1	文件操作概述	196	10.3.1	线程安全	232
8.1.1	驱动器	196	10.3.2	线程同步策略	232
8.1.2	目录	197	10.4	线程池	235
8.1.3	文件	198	10.4.1	线程池管理	235
8.1.4	路径	198	10.4.2	ThreadPool 类	236
8.2	流、文件操作	199	本章小结		238
8.2.1	文件 I/O 与流	199	习题		239
8.2.2	读/写文本文件	200	第 11 章 Web 应用程序设计技术		240
			11.1	Web 应用程序概述	240

11.1.1 Web 应用技术的发展	240	12.2.2 使用 Command 对象访问 数据库	264
11.1.2 ASP.NET 概述	242	12.2.3 使用 DataReader 对象访问 数据库	266
11.1.3 ASP.NET Web 应用程序	244	12.2.4 使用 DataAdapter 与 DataSet 对象 操作数据库	267
11.2 Web 应用程序的设计	245	12.3 数据库应用程序实例	271
11.2.1 设计 Web 应用程序的一般 步骤	245	12.3.1 添加数据库	272
11.2.2 ASP.NET Web 应用程序的 发布与测试	249	12.3.2 实现登录功能	274
本章小结	252	12.3.3 实现添加学生信息功能	275
习题	252	12.3.4 实现添加课程信息功能	276
实验 11	252	12.3.5 实现学生信息管理功能	277
第 12 章 数据库编程技术	255	12.3.6 实现课程信息管理功能	280
12.1 数据库与 ADO.NET 概述	255	12.3.7 实现学生成绩管理功能	281
12.1.1 数据库概述	255	本章小结	285
12.1.2 常用的 SQL 语句	257	习题	285
12.1.3 ADO.NET 概述	260	实验 12	286
12.2 使用 ADO.NET 访问数据库	261	参考文献	289
12.2.1 使用 Connection 对象访问 数据库	262		

第 1 章

.NET 与 C#概述

建议学时：4 学时

总体要求

- 了解.NET 平台、.NET Framework 的结构及其优点。
- 了解 C#语言的特点及其发展。
- 了解控制台应用程序、Win32 应用程序和 Web 应用程序的结构及其特点。
- 初步掌握 C#程序在 Visual Studio .NET 2005 中的建立、编辑、生成和运行过程。

相关知识点

- 了解计算机软件、计算机语言及分类知识。
- 熟悉 Windows 系统基础知识及操作。

学习重点

- C#程序的结构、特点。
- C#程序的建立、编辑、生成和运行等基本上机操作过程。

Microsoft .NET Framework 代表了一种构建和运行 Web 应用程序的主流技术，C#语言是.NET 平台的主要程序设计语言，Visual Studio .NET 是 C#程序的主要开发环境。因此，本章将介绍与 C#语言和 C#程序有关的基本概念和上机操作方法。

1.1 .NET 与 C#简介

1.1.1 .NET 平台简介

.NET 平台是由 Microsoft 推出的全新的应用程序开发平台，可用来构建和运行新一代 Microsoft Windows 和 Web 应用程序。它建立在开放体系结构基础之上，集 Microsoft 在软件领域的主要技术成就于一身。.NET 平台包括以下 4 种核心技术：.NET Framework、.NET 企业服务器、构建模块服务、Visual Studio.NET。

其中，.NET Framework 是.NET 平台核心中的核心，它为.NET 平台下应用程序的运行提供基本框架，如果把 Windows 操作系统比作一幢摩天大楼的地基，那么.NET Framework 就是摩天大楼中由钢筋和混凝土搭成的框架。

.NET 企业服务器是 Microsoft 为企事业单位提供的技术服务中的一部分，它包括 Microsoft

BizTalk Server、Microsoft Commerce Server、Host Integration Server、Exchange Server、SQL Server 等技术的开发工具包。

构建模块服务主要是 Microsoft 提供的 COM+ 组件服务和 XML Web 服务技术, 利用构建模块服务开发应用程序模块, 可迅速地组装一个功能完备的软件。

Visual Studio.NET 是 .NET 平台的主要开发工具, 由于 .NET 平台是建立在开放体系结构基础之上的, 因此应用程序开发人员也可以使用其他开发工具。

1.1.2 .NET Framework

为了创建和运行基于 .NET 平台的应用程序, .NET Framework 提供了必要的编译程序和运行程序的基础。

1. .NET Framework 的结构

.NET Framework 是支持生成和运行下一代应用程序和 XML Web 服务的内部 Windows 组件, 它也由不同的组件组成, 如图 1-1 所示, 这些组件有助于创建和运行基于 .NET 平台的应用程序。

.NET Framework 只能在 Win32 系列 (包含 Windows 98、Windows 2000、Windows XP 以及 Windows 2003 等) 基础上运行, 它可以与 Windows 操作系统的各种应用程序服务组件 (如消息队列服务、COM+ 组件服务、Internet 信息服务 (IIS)、Windows 管理工具 (WMI) 等) 整合, 以开发各种应用程序。

在 .NET Framework 框架的最顶层是程序设计语言, .NET Framework 支持诸如 Visual Basic、C++、C#、Perl、J# 等 20 多种计算机程序设计语言。在 Visual Studio.NET 开发环境中, 可直接使用 Visual Basic、C#、C++、J#、JScript 这 5 种语言开发任何应用程序。

.NET Framework 具有两个主要组件: 公共语言运行时 (Common Language Runtime, CLR)

和 .NET Framework 类库, 除此之外还包括 ADO.NET、ASP.NET、XML Web 服务等。

.NET Framework 支持 3 种类型的用户界面: 一种是命令控制台, 可用来设计纯字符界面的应用程序; 另一种是 Windows 窗体, 可用来设计 Windows 界面的应用程序; 第 3 种是 Web 窗体, 可用来设计 Web 页界面的应用程序。

2. 公共语言运行时 (CLR)

公共语言运行时是 .NET Framework 的基础, 是应用程序与操作系统之间的“中间人”, 它为应用程序提供内存管理、线程管理和远程处理等核心服务。在 .NET 平台上, 用户无论使用哪一种编程语言编写的程序, 在编译时语言编译器都会把它翻译成 MSIL (微软中间语言代码), 在运行应用程序时, 公共语言运行时自动把中间语言代码再次翻译成计算机操作系统所能识别的机器语言代码, 之后运行并返回运行结果。因此, 公共语言运行时是应用程序的托管环境, 对于这种在公共语言运行时基础之上运行的应用程序, 我们称之为托管应用程序, 而传统的直接在操作系统基础之上运行的应用程序则称为非托管应用程序。

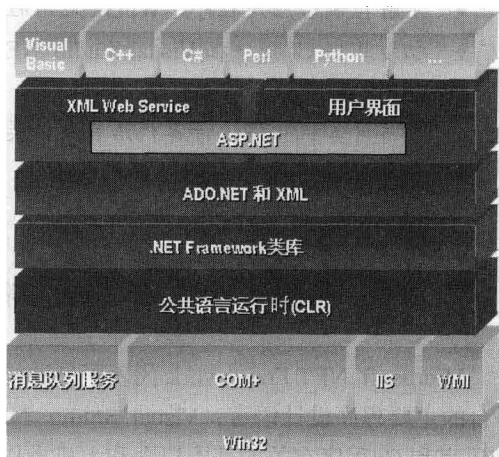


图 1-1 .NET Framework 的结构

3. .NET Framework 类库

.NET Framework 类库是一个综合性的面向对象的可重用类型集合,利用它可以开发多种应用程序,包括传统的命令行或图形用户界面 (GUI) 应用程序,也包括 Web 应用程序。

4. ADO.NET

ADO.NET 是 .NET Framework 提供的微软新一代的面向对象的数据处理技术,利用它可以简便、快捷地开发数据库应用程序。

5. ASP.NET

ASP.NET 是 .NET Framework 提供的全新的 Web 应用程序开发技术,利用它开发 Web 应用程序如同开发 Windows 应用程序一样简单。

6. XML Web Service

XML Web Service 是一种能够在 Internet 环境中直接调用的组件,.NET Framework 提供了创建、测试和部署 XML Web Service 的工具和类。

1.1.3 .NET Framework 的优点

在 .NET 平台诞生之前,虽然 Internet 已经出现,但很少有应用程序能运行在各种不同类型的客户端上,也不能和其他应用程序进行无缝集成。这种局限性导致开发人员花费大量的时间去改写应用程序,以保证它们能在各种客户端和平台上运行,而不是利用这些时间去设计新的应用程序。.NET Framework 的最大特色就在于它为应用程序开发人员提供了一个真正与平台无关的开发环境。使用 .NET Framework 开发应用程序有以下优点。

1. 基于 Web 的标准

.NET Framework 完全支持现有的 Internet 技术,包括 HTML (超文本标记语言)、HTTP (超文本传输协议)、XML (可扩展标记语言)、SOAP (简单对象访问协议)、XSLT (可扩展样式表语言转换)、XPath (XML 路径语言) 和其他 Web 标准。

2. 使用统一的应用程序模型

任何与 .NET 兼容的语言都可以使用 .NET Framework 类库。.NET Framework 为 Windows 应用程序、Web 应用程序和 XML Web 服务提供了统一的应用程序模型,因此同一段代码可被这些应用程序无障碍地使用。

3. 便于开发人员使用

在 .NET Framework 中,代码被组织在不同的命名空间和类中,而命名空间采用树形结构,以便开发人员引用。当开发人员企图调用 .NET Framework 类库的类时,只需将该类属性命名空间添加到引用解决方案中即可。

4. 可扩展类

.NET Framework 提供了通用类型系统,它根据面向对象的思想把一个命名空间或类中代码的实现细节隐藏,开发人员可以通过继承来访问类库中的类,也可以扩展类库中的类,甚至于构建自己的类库。

1.1.4 C#概述

1. C#的发展

C 和 C++ 曾经是商业软件开发领域中使用最广泛的语言,它们为程序员提供了十分灵活的操作,不过同时也牺牲了一定的效率。由于 C/C++ 语言的复杂性,许多程序员都试图寻找一种新的

语言，希望能在功能与效率之间找到一个更为理想的权衡点。

目前有些语言以牺牲灵活性为代价来提高效率，可是这些灵活性正是 C/C++ 程序员所需要的，这些解决方案对编程人员的限制过多（如屏蔽一些底层代码控制的机制），所提供的功能难以令人满意，并且这些语言无法方便地同早先的系统交互，也无法很好地和当前的网络编程相结合。

对于 C/C++ 用户来说，最理想的解决方案是在快速开发的同时又可以调用底层平台的所有功能。他们需要一种和最新的网络标准保持同步，并且能和已有的应用程序良好整合的环境。另外，一些 C/C++ 开发人员还需要在必要的时候进行一些底层的编程。

C#（读作 C Sharp）是微软对这一问题的解决方案。C# 是一种最新的、面向对象的编程语言，它是一种简单但功能强大的编程语言，程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序。

C# 是从 C 和 C++ 语言演化而来的，它在语句、表达式和运算符方面沿用了许多 C++ 的功能，在类型安全性、版本转换、事件和垃圾回收等方面进行了相当大的改进和创新，并提供对常用 API（如 .NET Framework、COM+ 等）的访问。

C# 自推出以来，已得到不断的改进和优化，它通常同 .NET Framework 一起，随 Visual Studio.NET 一同发布。目前，比较流行的 C# 主要有 3 个版本，分别是 C# 1.0、C# 2.0 和 C# 3.0。C# 1.0 同 .NET Framework 1.1 一起，随 Visual Studio .NET 2003 一同发布；C# 2.0 同 .NET Framework 2.0 一起，随 Visual Studio .NET 2005 一同发布；C# 3.0 同 .NET Framework 3.5 一起，随 Visual Studio .NET 2008 一同发布。本书以 C# 2.0、.NET Framework 2.0 和 Visual Studio .NET 2005 为范本，所有案例均在 Visual Studio .NET 2005 中经过调试运行。

2. C# 的优势

相对于其他计算机程序设计语言来说，C# 具有如下优点。

（1）C# 是一种精确、简单、类型安全、面向对象的语言。正是由于 C# 面向对象的卓越设计，使它成为构建各种应用程序组件的理想之选——无论是高级的商业对象还是系统级的应用程序。

（2）C# 具有生成持久系统级组件的能力，提供 COM+ 或其他技术平台支持，以集成现有代码；提供垃圾回收和类型安全，以实现应用程序可靠性；提供内部代码信任机制，以保证应用程序的安全性。

（3）C# 利用 .NET Framework 的通用类型系统能够与其他程序设计语言交互操作。C# 应用程序能跨语言、跨平台互相调用，使用 C# 语言可实现不同专业技术背景的人员协同工作，完成软件系统的设计和开发。

（4）C# 支持 MSMQ（微软消息队列）服务、COM+ 组件服务、XML Web 服务和 .NET Framework。使用 C#，一方面实现组件之间的相互调用，即实现不同软件技术开发的组件之间的集成应用；另一方面能够把传统的组件转化为 XML Web 服务，实现组件之间跨互联网的调用。

（5）C# 语言允许自定义数据类型，以扩展元数据，这些元数据可以应用于任何对象。项目构建者可以定义领域特有的属性，并把它们应用于任何语言元素——类、接口等，然后开发人员可以编程检查每个元素的属性。这样，很多工作都变得方便多了，比如编写一个小工具，自动检查每个类或接口是否被正确定义为某个抽象商业对象的一部分，或者只是创建一份基于对象的领域特有属性的报表。定制的元数据和程序代码之间的紧密对应，有助于加强程序的预期行为和实际实现之间的对应关系。

（6）C# 增强了开发效率，同时也致力于消除编程中可能导致严重结果的错误。C# 使 C/C++ 程序员可以快速进行网络开发，同时也保持了开发者所需要的强大性和灵活性。

1.2 C#程序的开发环境

Visual Studio .NET 是 C#程序的主要集成开发环境。目前使用较多的 Visual Studio .NET 有 3 个版本, 分别是 Visual Studio .NET 2003、Visual Studio .NET 2005 和 Visual Studio .NET 2008。本节将主要介绍 Visual Studio.NET 2005 的基本操作方法。

1.2.1 Visual Studio.NET 的基本操作

当计算机安装了 Visual Studio.NET 2005 后, 选择“开始”→“所有程序”→“Microsoft Visual Studio 2005”→“Microsoft Visual Studio 2005”菜单命令, 即可启动 Visual Studio.NET 2005。启动成功后, 显示 Visual Studio.NET 2005 的操作窗口, 如图 1-2 所示。



图 1-2 Visual Studio .NET 2005 窗口

刚启动的 Visual Studio.NET 2005 窗口由菜单栏、工具栏、起始页、解决方案资源管理器以及属性窗口组成。其中, 菜单栏列出了 Visual Studio .NET 2005 的所有操作命令; 工具栏则列出了常用的操作命令; 解决方案资源管理器用于显示将要创建的应用程序项目的文件夹结构; 属性窗口列出了当前操作对象的所有属性。

1. 应用程序项目的创建与打开

(1) 新建项目

Visual Studio.NET 2005 是一个高度集成的开发工具, 它集 Visual Basic、C++、C#和 J# 4 种程序设计语言为一体, 可以创建这 4 种语言的应用程序。它具有创建控制台应用程序、Windows 应用程序、类库、设备应用程序、Windows 控件库、安装项目、Web 应用程序 (ASP .NET 网站) 和 ASP .NET Web 服务等功能。因此, 在创建新项目之前应该先做好选择。

如果选择“文件”→“新建”→“项目”菜单命令, 则弹出“新建项目”对话框, 根据需要选择所有创建的项目即可, 如图 1-3 所示。例如, 在“项目类型”列表框中选择“Visual C#”选项, 在“模板”列表框中选择“Windows 应用程序”选项, 在“名称”文本框中输入应用程序项目的名字 (如 MyFirstWinApp), 在“位置”下拉列表框中设置应用程序项目的创建位置 (如 e:\

我的文档\Visual Studio 2005\projects), 在“解决方案名称”文本框中输入解决方案的名称(如 MyFirstWinApp), 然后单击“确定”按钮即可。

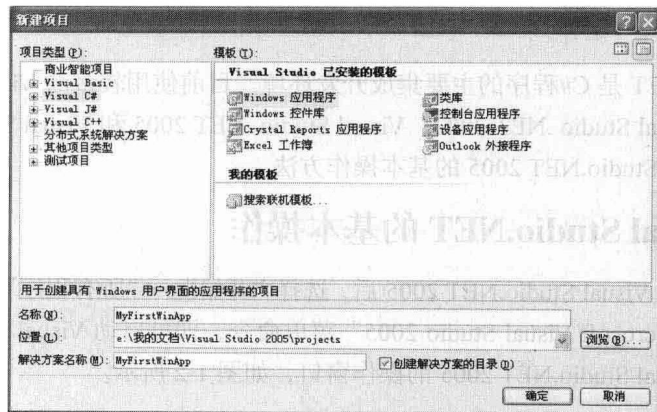


图 1-3 “新建项目”对话框

如果选择“文件”→“新建”→“网站”菜单命令, 则弹出“新建网站”对话框, 根据需要进行选择所有创建的项目即可, 如图 1-4 所示。在创建网站时, 应注意以下几点:

- ① “模板”列表框中的“ASP.NET 网站”表示 Web 应用程序, 而“ASP.NET Web 服务”表示能通过互联网远程调用的组件;
- ② “位置”下拉列表框中有 3 种选择, 其中,“文件系统”表示将网站创建到磁盘文件夹中,“HTTP”表示把网站直接创建到某一个 Web 服务器中,“FTP”表示把网站创建到某一个 FTP 服务器, 以便于上传文件到 Web 服务器;
- ③ “语言”下拉列表框中列举了可用的程序设计语言, 如“Visual C#”。

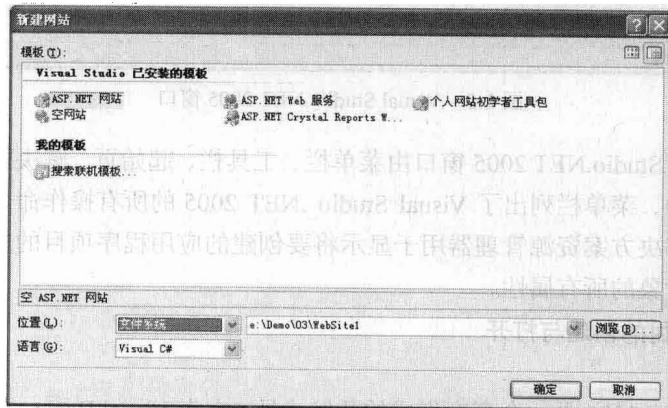


图 1-4 “新建网站”对话框

(2) 打开项目

要打开一个已经存在的应用程序项目, 可按如下方法操作:

选择“文件”→“打开”→“项目/解决方案”菜单命令, 打开控制台应用程序、Windows 应用程序等; 选择“文件”→“打开”→“网站”菜单命令, 打开 Web 应用程序项目; 也可以通过选择“文件”→“最近的项目”菜单命令打开已存在的项目, 还可以通过“起始页”选择最近的项目来打开。

2. 应用程序项目的设计与编辑

在创建或打开项目之后,即可在 Visual Studio .NET 2005 窗口的工作区中对应用程序项目进行设计和编辑。Visual Studio .NET 2005 窗口的工作区一般分为 3 部分,左边是工具箱,中间是设计区,右边是解决方案资源管理器和属性窗,如图 1-5 所示。



图 1-5 设计或编辑应用程序

如果要修改项目文件名或添加、删除项目文件,则只需在解决方案资源管理器中进行操作即可。最简单的操作方法是:先在解决方案资源管理器中选择操作对象,然后单击鼠标右键,再选择相应的快捷菜单命令。

如果要设计或修改应用程序的用户界面(Windows Form 或 Web Form),则只需在工具箱和设计区中进行操作即可。向 Windows Form 或 Web Form 中添加内容的最简单操作方法是:拖动工具箱中的工具按钮(控件图标)并放到设计区。

在设计用户界面时,如果要修改 Form 中某个控件或对象的属性,在“属性”窗口中即可完成操作。

如果要编辑或修改程序源代码,可在设计区单击鼠标右键,在快捷菜单中选择“查看源代码”命令,将 Visual Studio .NET 2005 的设计视图切换到源代码编辑视图即可;反之,在源代码编辑视图中,在文本编辑窗口中单击鼠标右键,在快捷菜单中选择“查看设计器”命令,即可切换到设计视图。

当应用程序的设计和编辑操作完成之后,如果需要保存,只需单击工具栏上的“全部保存”按钮,即可把所有编辑或修改操作保存到磁盘中。

3. 应用程序项目的调试与生成

在应用程序开发过程中或开发结束后,可能需要查看设计的某个部分的运行情况或结果,此时只需选择“调试”→“启动调试”菜单命令,或直接按<F5>键进行调试。当程序中有错误时,Visual Studio .NET 2005 会在“错误列表”窗口中显示语言编译器的错误提示信息,程序员可根据错误提示信息修改程序。值得注意的是,语言编译器只能发现源程序中的语法错误,并不能发现程序逻辑错误。

当设计已经全部完成后,可选择“生成”→“生成×××”菜单命令,生成解决方案(其中,“×××”为项目名)。此时,语言编译器会在应用程序项目所在目录下的 bin\debug 文件夹中生成一个可执行文件(.exe)。