



Visual C# 2008

程序开发入门与提高

赵增敏 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Visual C# 2008

程序开发入门与提高

——第2版——

陈皓明 著

Visual C# 2008

程序开发入门与提高

赵增敏 编著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Visual C# 2008 是 Microsoft 公司为生成在 .NET Framework 上运行的各种应用程序而设计的一种编程语言,该语言可以借助于 Visual Studio 2008 这套世界级的可视化开发工具来实现应用程序的快速开发。本书通过大量实例循序渐进地介绍了 Visual C# 2008 编程语言及其在控制台应用程序、Windows 窗体应用程序、WPF 应用程序,以及具有 AJAX 功能的 ASP.NET Web 应用程序开发中的应用。本书内容全面,涉及使用 Visual C# 2008 开发多种应用程序的方方面面。书中共提供了 280 个开发实例,具有较高的参考价值。

本书结构合理,内容翔实、新颖,概念清晰,实例丰富,操作步骤完整。为了方便读者的学习,在随附光盘中包含了书中的所有实例。

本书不仅适合于初学 Visual C# 2008 的读者使用,对于那些具有一定编程经验的 C#开发人员同样很有参考价值。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C# 2008 程序开发入门与提高 / 赵增敏编著. —北京: 电子工业出版社, 2009.8
ISBN 978-7-121-08619-9

I. V… II. 赵… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 049338 号

责任编辑: 孙学瑛

印 刷: 北京天宇星印刷厂

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 860×1092 1/16 印张: 50.25 字数: 1260 千字

印 次: 2009 年 8 月第 1 次印刷

印 数: 3500 册 定价: 89.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

Visual C# 2008 是 Microsoft 公司为生成在 .NET Framework 上运行的各种应用程序而设计的一种编程语言，具有语法简单、功能强大和类型安全等优点，而且是完全面向对象的。Visual C# 2008 凭借在许多方面的创新，在保持 C 语言的表现力强和风格雅致特征的同时，实现了各种应用程序的快速开发。它不仅支持传统的控制台应用程序和 Windows 窗体应用程序的开发，也支持 WPF 及 ASP.NET Web 应用程序的开发，而且可以为 ASP.NET Web 应用程序添加 AJAX 功能。

Visual C# 2008 应用程序开发可以借助 Visual Studio 2008 这套世界级的可视化开发工具来实现，Visual Studio 2008 通过功能齐全的代码编辑器、编译器、项目模板、设计器、代码向导、功能强大且易用的调试器及其他工具实现了对 Visual C# 的支持。在 Visual Studio 中，可以通过 .NET Framework 类库方便地访问许多操作系统服务和其他有用的精心设计的类，这些类可以显著地缩短开发周期。在 Visual Studio 2008 集成开发环境中使用 Visual C# 2008 语言来开发各种应用程序，是当今众多开发人员的首选。

作者结合自己长期从事 Visual C# 应用开发和教学工作的经验精心地编写了本书，旨在帮助读者快速掌握使用 Visual C# 2008 开发各种应用程序的方法、步骤和编程技巧。作者从实践中认识到，对于任何一种编程语言，不但要精通语言本身，而且更要注重其应用，精通的目的全在于应用。基于这种认识，本书首先结合控制台应用程序设计深入细致地讲述了 Visual C# 语言基础和面向对象程序设计的有关内容，然后详细地讨论了 Visual C# 语言在 Windows 窗体应用程序、WPF 应用程序设计，以及 ASP.NET Web 应用程序设计等方面的应用，而且还介绍了 ASP.NET 中的 AJAX 功能等最新知识。

本书特色

1. 结构合理

本书坚持以语言为基础和应用为主导的编写原则，用一半的篇幅来讲述 Visual C# 语言本身的内容，用另一半篇幅介绍该语言在各种应用程序开发中的应用。为了便于读者学习和掌握 Visual C# 语言的特征，本书通过控制台应用程序设计来讲述这种编程语言。介绍 Visual C# 语言本身的内容之后，书中依次介绍了该语言在 Windows 窗体应用程序、WPF 应用程序，以及 ASP.NET Web 应用程序中的应用。通过学习开发各种应用程序的方法，读者可理解这些应用程序的共性和个性。

2. 内容新颖

本书中除了讲述如何使用 Visual C# 语言开发 Windows 窗体应用程序和 ASP.NET 应用程序外，也介绍了 WPF 应用程序设计的最新知识，包括 XAML 可扩展应用程序标记语言。此外还介绍了 ASP.NET 中的 AJAX 功能，包括 ASP.NET AJAX 服务器控件及 AJAX Control Toolkit 控件，这些新知识对于学习 Visual C# 的开发人员来说是十分新颖的。

3. 实例丰富

为了帮助读者在短时间内掌握使用 Visual C# 语言开发各种应用程序中的知识点和编程技巧, 本书提供了 280 多个通过测试可运行的完整例子。本书在讲述这些例子时, 首先给出最终的运行效果图, 然后介绍详细的设计步骤。对于容易出现问题的地方, 则以“提示”或“注意”的方式介绍常用的技巧。

本书内容

第 1 篇(第 1 章至第 3 章): 介绍 C# 语言程序设计的基础知识, 包括 Visual C# 2008 概述、C# 编程基础, 以及 C# 中的流程控制语句。

第 2 篇(第 4 章至第 12 章): 介绍 C# 面向对象程序设计的内容, 包括类与对象、属性与索引器、委托与事件、接口与泛型、LINQ 表达式、属性、迭代器和命名空间、字符串与正则表达式、数组与集合, 以及文件输入/输出。

第 3 篇(第 13 章至第 16 章): 介绍 C# 在 Windows 窗体编程方面的应用, 包括 Windows 窗体与控件、菜单、工具栏与对话框、GDI+ 图形编程, 以及 Windows 窗体数据访问。

第 4 篇(第 17 章至第 20 章): 介绍 C# 在 WPF 程序设计方面的应用, 包括 WPF 应用程序与窗体、XAML 语言、WPF 控件, 以及 WPF 图形、动画和媒体。

第 5 篇(第 21 章至第 23 章): 介绍 C# 在 ASP.NET Web 应用程序设计方面的应用, 包括 ASP.NET Web 编程基础、ASP.NET 数据访问以及 ASP.NET AJAX 应用开发。

致谢

本书由赵增敏编著, 参加本书编写、文字录入, 以及代码调试的还有朱粹丹、赵朱曦及郭宏等, 在此一并致谢。非常感谢博文视点郭立总经理和孙学瑛编辑, 她们为本书的选题策划、编辑加工和出版发行付出了辛勤的劳动。

作 者

2009 年 3 月于郑州

目录

第 1 篇 C#语言编程基础

第 1 章 Visual C# 2008 概述2

1.1 C#语言简介	2
1.1.1 C#语言与 .NET Framework	2
1.1.2 C#语言与其他语言的比较	4
1.1.3 C# 3.0 的新功能	5
1.2 Visual Studio 2008 简介	6
1.2.1 Visual Studio 2008 的新功能	6
1.2.2 安装 Visual Studio 2008	7
1.2.3 选择默认环境设置	10
1.3 认识 Visual Studio 2008 IDE	10
1.3.1 起始页	10
1.3.2 菜单与工具栏	11
1.3.3 解决方案资源管理器	12
1.3.4 设计器窗口	13
1.3.5 工具箱	14
1.3.6 属性窗口	14
1.3.7 代码编辑器	15
1.3.8 对象浏览器	17
1.4 创建第 1 个 Visual C#应用程序	18
1.4.1 创建 C#控制台应用程序	18
1.4.2 创建 C# Windows 窗体应用程序	21
1.4.3 创建 C# WPF 应用程序	26
1.4.4 创建 ASP.NET Web 应用程序	29

第 2 章 C# 语言编程基础32

2.1 基本语法	32
2.1.1 标识符	32
2.1.2 关键字	33
2.1.3 基本语法规则	34

2.1.4 控制台应用程序基本结构	34
2.2 数据类型	38
2.2.1 值类型	38
2.2.2 引用类型	43
2.2.3 数据类型转换	49
2.2.4 装箱与拆箱	52
2.3 变量和常量	54
2.3.1 声明和使用变量	54
2.3.2 声明和使用常量	55
2.4 运算符	56
2.4.1 算术运算符	57
2.4.2 关系运算符	58
2.4.3 逻辑运算符	59
2.4.4 位运算符	61
2.4.5 赋值运算符	62
2.4.6 其他运算符	63
2.4.7 运算符优先级	67
2.5 控制台输入/输出	68
2.5.1 屏幕缓冲区与控制台窗口	68
2.5.2 Console 类的成员	68
2.5.3 控制台输入	69
2.5.4 控制台输出	73

第 3 章 流程控制语句 77

3.1 选择语句	77
3.1.1 if 语句	77
3.1.2 switch 语句	79
3.2 迭代语句	81
3.2.1 while 语句	81
3.2.2 do 语句	83

3.2.3 for 语句	84	3.4.3 throw 语句	95
3.2.4 foreach 语句	85	3.5 程序调试	97
3.3 跳转语句	87	3.5.1 程序错误类型	97
3.3.1 break 语句	87	3.5.2 程序调试方法	97
3.3.2 continue 语句	88	3.6 预处理器指令	100
3.3.3 goto 语句	89	3.6.1 定义符号	100
3.3.4 return 语句	90	3.6.2 条件编译	100
3.4 异常处理语句	91	3.6.3 报告编译错误和警告	101
3.4.1 异常处理概述	91	3.6.4 定义代码块	102
3.4.2 try-catch-finally 语句	92		

第 2 篇 C#面向对象编程

第 4 章 类与对象	105	4.5.5 可视化 OOP 工具	136
4.1 面向对象编程基本概念	105	4.6 方法	140
4.1.1 类	105	4.6.1 声明方法	140
4.1.2 对象	106	4.6.2 方法形参	141
4.2 声明类	106	4.6.3 方法重载	146
4.2.1 类声明语法	106	4.6.4 静态方法和实例方法	147
4.2.2 类修饰符	108	4.6.5 虚方法和重写方法	149
4.2.3 类成员	111	4.6.7 密封方法	151
4.2.4 结构与类的区别	113	4.6.8 抽象方法	152
4.3 构造函数与析构函数	115	4.6.9 外部方法	153
4.3.1 实例构造函数	115	4.6.10 扩展方法	154
4.3.2 私有构造函数	118	4.7 嵌套类	155
4.3.3 静态构造函数	119	4.7.1 声明嵌套类	156
4.3.4 析构函数	121	4.7.2 隐藏嵌套类	157
4.4 常量与字段	122	4.8 分部类与分部方法	159
4.4.1 声明常量	122	4.8.1 分部类	159
4.4.2 声明字段	124	4.8.2 分部方法	162
4.4.3 静态字段和实例字段	125	4.9 静态类与静态成员	164
4.4.4 只读字段	126	4.9.1 静态类	165
4.4.5 可变字段	128	4.9.2 静态成员	166
4.5 继承	129	第 5 章 属性与索引器	169
4.5.1 类的继承	129	5.1 属性	169
4.5.2 抽象类	131	5.1.1 声明属性	169
4.5.3 密封类	134	5.1.2 属性访问器	170
4.5.4 System.Object 类	135	5.1.3 静态属性与实例属性	174

5.1.4 属性与继承.....	176	7.1.9 接口重新实现.....	247
5.1.5 非对称访问器.....	179	7.2 泛型.....	250
5.1.6 自动实现的属性.....	183	7.2.1 泛型概述.....	250
5.1.7 匿名类型.....	186	7.2.2 泛型类型参数.....	252
5.2 索引器.....	187	7.2.3 类型参数约束.....	253
5.2.1 声明索引器.....	187	7.2.4 泛型类.....	254
5.2.2 索引器与属性的比较.....	189	7.2.5 泛型接口.....	257
5.2.3 使用索引器.....	189	7.2.6 泛型方法.....	260
5.2.4 索引器重载.....	191	7.2.7 泛型和数组.....	262
5.2.5 多维索引器.....	193	7.2.8 泛型委托.....	263
第6章 委托与事件.....	196	7.2.9 default 关键字.....	265
6.1 委托.....	196	第8章 LINQ 表达式.....	266
6.1.1 委托概述.....	196	8.1 LINQ 查询概述.....	266
6.1.2 声明委托.....	197	8.1.1 基本概念.....	266
6.1.3 实例化委托.....	198	8.1.2 LINQ 与泛型类型.....	269
6.1.4 调用委托.....	199	8.1.3 Lambda 表达式.....	271
6.1.5 多路广播委托.....	200	8.1.4 在 C# 中编写 LINQ 查询.....	273
6.1.6 通过委托实现回调.....	203	8.2 使用 LINQ 查询子句.....	275
6.1.7 静态委托.....	205	8.2.1 from 子句.....	275
6.1.8 委托中的协变和逆变.....	206	8.2.2 where 子句.....	277
6.1.9 委托的异常处理.....	208	8.2.3 select 子句.....	278
6.2 事件.....	210	8.2.4 group 子句.....	280
6.2.1 事件概述.....	210	8.2.5 orderby 子句.....	283
6.2.2 使用预定义事件.....	211	8.2.6 join 子句.....	284
6.2.3 实现自定义事件.....	215	8.2.7 let 子句.....	287
6.2.4 事件访问器.....	220	第9章 属性、迭代器和命名空间.....	289
第7章 接口与泛型.....	225	9.1 属性.....	289
7.1 接口.....	225	9.1.1 属性概述.....	289
7.1.1 接口概述.....	225	9.1.2 自定义属性.....	290
7.1.2 声明接口.....	226	9.1.3 应用属性.....	293
7.1.3 声明接口成员.....	227	9.1.4 访问属性.....	295
7.1.4 访问接口成员.....	229	9.1.5 通用属性.....	297
7.1.5 接口实现.....	231	9.1.6 交互操作的属性.....	300
7.1.6 显式接口实现.....	238	9.2 迭代器.....	302
7.1.7 接口映射.....	242	9.2.1 迭代器概述.....	302
7.1.8 接口实现继承.....	244	9.2.2 创建和使用迭代器.....	303

9.2.3 为泛型类创建迭代器	305	11.1.2 数组概述	349
9.3 命名空间	307	11.1.3 动态创建数组	351
9.3.1 编译单元	307	11.1.4 数组排序	353
9.3.2 声明命名空间	308	11.1.5 搜索数组	355
9.3.3 使用命名空间	310	11.1.6 反转数组	357
9.3.4 使用命名空间别名限定符	314	11.1.7 复制数组	359
9.3.5 使用 My 命名空间	315	11.2 非泛型集合	360
第 10 章 字符串与正则表达式	317	11.2.1 System.Collections 概述	360
10.1 不变字符串	317	11.2.2 ArrayList 类	361
10.1.1 String 类概述	317	11.2.3 Queue 类	364
10.1.2 比较字符串	319	11.2.4 Stack 类	365
10.1.3 检索子字符串	321	11.2.5 Hashtable 类	367
10.1.4 复制字符串	323	11.2.6 SortedList 类	369
10.1.5 拆分字符串	324	11.3 泛型集合	372
10.1.6 连接字符串	325	11.3.1 System.Collections.Generic 概述	372
10.1.7 修改字符串	326	11.3.2 List<T> 类	373
10.1.8 转换大小写	328	11.3.3 Dictionary<TKey, TValue> 类	376
10.1.9 格式化字符串	329	11.3.4 SortedDictionary<TKey, TValue> 类	378
10.2 可变字符串	331	第 12 章 文件输入/输出	381
10.2.1 StringBuilder 类概述	332	12.1 文件管理	381
10.2.2 设置 StringBuilder 容量	332	12.1.1 System.IO 概述	381
10.2.3 追加操作	333	12.1.2 文件类 File	382
10.2.4 插入操作	336	12.1.3 文件信息类 FileInfo	385
10.2.5 删除操作	338	12.2 文件读/写	388
10.2.6 替换操作	339	12.2.1 文件流类 FileStream	388
10.3 字符串编码	340	12.2.2 流写入类 StreamWriter	392
10.3.1 Encoding 类概述	340	12.2.3 流读取类 StreamReader	393
10.3.2 字符串编码示例	341	12.2.4 二进制写入类 BinaryWriter	395
10.4 正则表达式	342	12.2.5 二进制读取类 BinaryReader	397
10.4.1 正则表达式概述	342	12.3 目录与驱动器	399
10.4.2 编写正则表达式	343	12.3.1 目录类 Directory	399
10.4.3 使用 Regex 类	344	12.3.2 目录信息类 DirectoryInfo	401
第 11 章 数组与集合	347	12.3.3 路径类 Path	403
11.1 数组	347	12.3.4 驱动器信息类 DriveInfo	404
11.1.1 Array 类	347		

第3篇 Windows 窗体编程

第13章 Windows 窗体与控件.....407

- 13.1 Windows 窗体.....407
 - 13.1.1 创建 Windows 窗体.....407
 - 13.1.2 创建 Windows 应用程序项目.....409
 - 13.1.3 Form 类.....412
 - 13.1.4 在项目中添加窗体.....414
 - 13.1.5 窗体设计技巧.....416
 - 13.1.6 在窗体中创建事件处理程序.....418
 - 13.1.7 创建 MDI 窗体.....421
- 13.2 Windows 窗体控件.....422
 - 13.2.1 分类.....422
 - 13.2.2 控件的基本操作.....424
 - 13.2.3 执行操作的控件.....426
 - 13.2.4 显示信息的控件.....430
 - 13.2.5 编辑文本控件.....433
 - 13.2.6 选项设置控件.....443
 - 13.2.7 列表选择控件.....447
 - 13.2.8 日期选择控件.....460
 - 13.2.9 帮助组件.....462
 - 13.2.10 显示和存储图像的控件.....466
 - 13.2.11 容器控件.....467
 - 13.2.12 杂项控件.....469

第14章 菜单、工具栏和对话框.....473

- 14.1 主菜单.....473
 - 14.1.1 MenuStrip 控件概述.....473
 - 14.1.2 创建主菜单.....474
 - 14.1.3 创建菜单项事件处理程序.....477
 - 14.1.4 创建 MDI 子窗口列表.....481
- 14.2 快捷菜单.....483
 - 14.2.1 ContextMenuStrip 控件概述.....483
 - 14.2.2 创建和显示快捷菜单.....484
- 14.3 工具栏.....486
 - 14.3.1 ToolStrip 控件概述.....486
 - 14.3.2 创建工具栏.....488

- 14.4 对话框.....492
 - 14.4.1 打开和保存文件.....492
 - 14.4.2 设置字体和颜色.....496
 - 14.4.3 打印文档.....498

第15章 GDI+图形编程.....504

- 15.1 GDI+概述.....504
 - 15.1.1 GDI+的功能.....504
 - 15.1.2 System.Drawing.....505
 - 15.1.3 Graphics 对象.....506
- 15.2 使用钢笔绘图.....509
 - 15.2.1 Pen 类概述.....509
 - 15.2.2 使用钢笔绘制线条和形状.....510
- 15.3 使用画笔填充形状.....515
 - 15.3.1 使用纯色填充形状.....515
 - 15.3.2 使用阴影图案填充形状.....517
 - 15.3.3 使用图像纹理填充形状.....519
 - 15.3.4 使用渐变画笔填充形状.....521
- 15.4 图像处理.....526
 - 15.4.1 加载和显示图像.....526
 - 15.4.2 常见图像操作.....527

第16章 访问 Windows 窗体数据.....530

- 16.1 使用 ADO.NET 对象.....530
 - 16.1.1 ADO.NET 组件概述.....530
 - 16.1.2 数据库连接.....531
 - 16.1.3 数据命令.....534
 - 16.1.4 数据读取器.....536
 - 16.1.5 数据适配器.....540
 - 16.1.6 数据集.....543
- 16.2 使用数据访问控件.....549
 - 16.2.1 数据访问控件介绍.....549
 - 16.2.2 设计基本数据窗体.....552
 - 16.2.3 设计主-详细窗体.....557

第4篇 创建 WPF 应用程序

第17章 WPF 应用程序与窗体561

17.1 WPF 编程入门	561
17.1.1 使用 csc 编译器生成 WPF 应用程序	561
17.1.2 用 MSBuild 生成 WPF 应用程序 ...	563
17.1.3 使用 VS 创建 WPF 独立 应用程序	567
17.1.4 使用 VS 创建 WPF 浏览器 应用程序	570
17.2 管理 WPF 应用程序	573
17.2.1 Application 类	573
17.2.2 应用程序生存期	575
17.2.3 管理属性、资源和窗口	578
17.2.4 应用程序导航	582
17.3 WPF 窗口	585
17.3.1 Window 类	585
17.3.2 窗口生存期	587
17.3.3 窗口的外观和行为	590
17.3.4 窗口所属权	591

第18章 XAML 语言594

18.1 XAML 概述	594
18.1.1 对象元素	594
18.1.2 命名空间	595
18.1.3 x:Class 属性	596
18.1.4 子元素	597
18.2 设置属性	599
18.2.1 属性语法	599
18.2.2 属性元素	600
18.2.3 附加属性	601
18.3 标记扩展	603
18.3.1 标记扩展语法	603
18.3.2 XAML 标记扩展	603
18.3.3 WPF 标记扩展	607
18.4 XAML 子元素	613

18.4.1 子元素处理规则	613
18.4.2 内容属性	613
18.4.3 集合项	614
18.4.4 普通文本	615
18.5 加载和解析 XAML	615
18.5.1 从字符串中加载 XAML	615
18.5.2 从资源文件中加载 XAML	617

第19章 WPF 控件620

19.1 WPF 控件概述	620
19.1.1 WPF 控件分类	620
19.1.2 使用 WPF 控件	622
19.2 布局控件	625
19.2.1 Canvas 控件	625
19.2.2 DockPanel 控件	626
19.2.3 Grid 控件	627
19.2.4 StackPanel 控件	630
19.2.5 WrapPanel 控件	631
19.3 常用控件	633
19.3.1 按钮控件	633
19.3.2 用户信息控件	635
19.3.3 输入控件	638
19.3.4 选择控件	640
19.3.5 菜单控件	649
19.3.6 工具栏和状态栏控件	652
19.3.7 导航控件	654
19.3.8 数字墨迹控件	657

第20章 WPF 图形、动画和媒体660

20.1 图形	660
20.1.1 绘制基本形状	660
20.1.2 形状的拉伸和变换	666
20.1.3 使用画笔	668
20.2 动画	672
20.2.1 演示图板动画技术	672
20.2.2 From/To/By 动画	673

20.2.3 关键帧动画	675	20.3.1 媒体播放模式	680
20.2.4 路径动画	677	20.3.2 播放音频	680
20.3 媒体播放	680	20.3.3 播放视频	684

第5篇 创建 ASP.NET Web 应用程序

第21章 ASP.NET Web 编程基础.....687

21.1 ASP.NET 基础	687
21.1.1 ASP.NET 概述	687
21.1.2 创建 ASP.NET Web 应用程序	688
21.1.3 ASP.NET 网页语法	691
21.2 ASP.NET 服务器控件	693
21.2.1 HTML 服务器控件	694
21.2.2 Web 服务器控件	698
21.2.3 验证控件	702
21.2.4 用户控件	706
21.3 ASP.NET 状态管理	708
21.3.1 基于客户端的状态管理	708
21.3.2 基于服务器的状态管理	711

第22章 ASP.NET 数据访问.....718

22.1 数据源控件	718
22.1.1 SqlDataSource 控件	718
22.1.2 AccessDataSource 控件	721
22.1.3 其他数据源控件	722
22.2 GridView 控件	723
22.2.1 GridView 控件概述	723
22.2.2 分页显示数据	725
22.2.3 排序数据	727
22.2.4 实现主/详细页	729
22.2.5 更新数据	731
22.2.6 删除数据	735
22.3 DetailsView 控件	736
22.3.1 DetailsView 控件概述	737
22.3.2 分页显示数据	738

22.3.3 更新数据	740
22.4 FormView 控件	742
22.4.1 FormView 控件概述	743
22.4.2 创建模板	743
22.4.3 分页显示数据	744
22.4.4 更新数据	746
22.5 其他数据控件	750
22.5.1 Repeater 控件	750
22.5.2 DataList 控件	751
22.5.3 ListView 控件	752

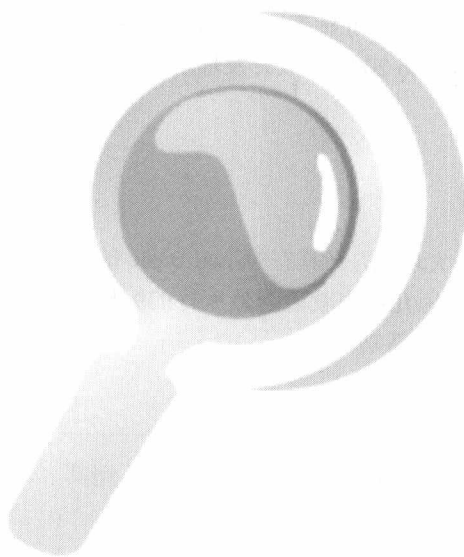
第23章 ASP.NET AJAX 应用开发.....753

23.1 ASP.NET AJAX 概述	753
23.1.1 ASP.NET AJAX 体系结构	753
23.1.2 用 JavaScript 实现 AJAX 功能	756
23.1.3 ASP.NET AJAX 功能应用示例	759
23.2 ASP.NET AJAX 服务器控件	762
23.2.1 ScriptManager 控件	762
23.2.2 UpdatePanel 控件	766
23.2.3 UpdateProgress 控件	769
23.2.4 Timer 控件	771
23.3 ASP.NET AJAX 控件工具箱	774
23.3.1 安装	774
23.3.2 工具箱中的控件	777
23.3.3 RoundedCorners 控件	779
23.3.4 AutoComplete 控件	780
23.3.5 CascadingDropDown 控件	782
23.3.6 Accordion 控件	785

第 1 篇

C#语言编程基础

Visual C# 2008 程序开发入门与提高



第 1 章

Visual C# 2008 概述

Visual C# 2008 (读做“C sharp”)是 Microsoft 为生成在 .NET Framework 上运行的多种应用程序而设计的一种编程语言,该语言简单、功能强大、类型安全,而且是面向对象的。C#语言保持了 C 样式语言的表示形式和优美,同时也进行了许多创新,实现了应用程序的快速开发。

Visual Studio 是一套完整的开发工具,可用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Studio 支持包括 Visual C#在内的多种编程语言,这是通过功能齐全的代码编辑器、项目模板、设计器、代码向导、功能强大且易于使用的调试器,以及其他工具来实现的。通过 .NET Framework 类库可以访问多种操作系统服务和其他有用的精心设计的类,这些类可显著地提高开发效率。

本章首先简要介绍 C#语言,然后说明如何安装、配置和使用 Visual Studio 2008 集成开发环境 (IDE),最后讲述如何在这个集成开发环境中创建 4 种不同类型的 C#应用程序。

1.1 C#语言简介

可以把 C#看做是 C++与 Visual Basic 的完美结合,它既具有 C++的强大功能,又具有 Visual Basic 的快速应用开发特性 (RAD),因此受到了众多开发人员的青睐,在推出后的短短几年内就成为最流行的编程语言。本节介绍 C#语言与 .NET Framework 及其他语言的比较,以及 C# 3.0 的新功能等。

1.1.1 C#语言与 .NET Framework

.NET Framework 是 Windows 的一个组成部分,C#是一种基于 .NET Framework 平台的编程语言,用其可以创建在 .NET Framework 上运行的多种应用程序。

1. C#语言

C#是一种简洁且类型安全的面向对象的语言,开发人员可以用其来构建在 .NET Framework 上运行的各种安全且可靠的应用程序。使用 C#可以创建传统的 Windows 客户端应用程序、XML Web services、分布式组件、客户端/服务器应用程序,以及数据库应用程序等。Visual C# 2008 提供了高级代码编辑器、方便的用户界面设计器、集成调试器和许多其他工具,使开发人员可以更容易地在 C#语言 3.0 版和 .NET Framework 3.5 版的基础上开发应用程序。

C#语言具有以下特点。

(1) 语法表现力强,简单易学。C#中的花括号语法 ({}) 使任何熟悉 C、C++或 Java 的开发人员都可以迅速上手,只要了解上述任何一种语言,通常在很短的时间内就可以开始使用 C#高效地工作。C#语法简



化了 C++ 的诸多复杂性，并提供了很多强大的功能。例如，null 的值类型、枚举、委托、lambda 表达式和直接内存访问，这些都是 Java 所不具备的。C# 支持泛型方法和类型，从而提供了更出色的类型安全和性能。并且提供了迭代器，允许集合类的实施者创建自定义的迭代行为，以便容易被客户端代码使用。在 C# 3.0 中，LINQ（语言集成查询）表达式使强类型查询成为一流的语言构造。

(2) 面向对象。C# 语言支持封装、继承和多态性的概念，所有的变量和方法，包括 Main 方法（应用程序的入口点）都封装在类定义中。类可能直接从一个父类继承，可以实现任意数量的接口。重写父类中的虚方法时，要求 override 关键字作为一种避免意外重定义的方式。在 C# 中，结构类似于一个轻量类。它是一种堆栈分配的类型，可以实现接口，但不支持继承。除了这些基本的面向对象的原理之外，C# 还通过几种创新的语言构造简化了软件组件的开发，这些构造如下。

- 用委托封装方法签名，实现了类型安全的事件通知。
- 属性（Property）充当私有成员变量的访问器。
- 属性（Attribute）提供关于运行时类型的声明性元数据。
- 内联 XML 文档注释。
- LINQ 提供了跨各种数据源的内置查询功能。

(3) 在 C# 中，如果必须与其他 Windows 软件（如 COM 对象或本机 Win32 DLL）交互，则可以通过一个称为“互操作”的过程来实现。互操作使 C# 程序能够完成本机 C++ 应用程序可以完成的几乎所有任务。在直接内存访问必不可少的情况下，C# 甚至支持指针和“不安全”代码的概念。

(4) C# 的生成过程比 C 和 C++ 简单，比 Java 更为灵活。没有单独的头文件，也不要求按照特定顺序声明方法和类型，在 C# 源文件可以定义任意数量的类、结构、接口和事件。

2. .NET Framework 平台体系结构

C# 程序在 .NET Framework 平台上运行，该平台是 Windows 的一个不可或缺的组件，包括一个称为“公共语言运行库”（CLR）的虚拟执行系统和一组统一的类库。CLR 是 Microsoft 的公共语言基础结构（CLI）的商业实现，它提供内存管理、线程管理和远程处理等核心服务，并且还强制实施严格的类型安全，以及可提高安全性和可靠性的其他形式的代码准确性。CLI 是一种国际标准，是用于创建语言和库在其中无缝协同工作的执行和开发环境的基础。

用 C# 编写的源代码被编译为一种符合 CLI 规范的中间语言（IL），IL 代码与资源（例如位图和字符串）一起作为一种称为“程序集”的可执行文件存储在磁盘上，通常具有的扩展名为 .exe 或 .dll。程序集包含清单，它提供有关程序集的类型、版本、区域性和安全要求等信息。

执行 C# 程序时，程序集将加载到 CLR 中，并可能根据清单中的信息执行不同的操作。如果符合安全要求，CLR 就会执行实时（JIT）编译以将 IL 代码转换为本机机器指令。CLR 还提供与自动垃圾回收、异常处理和资源管理有关的其他服务。由 CLR 执行的代码有时称为“托管代码”，它与编译为面向特定系统的本机机器语言的“非托管代码”相对应。如图 1-1 所示为 C# 应用程序的编译与运行流程。

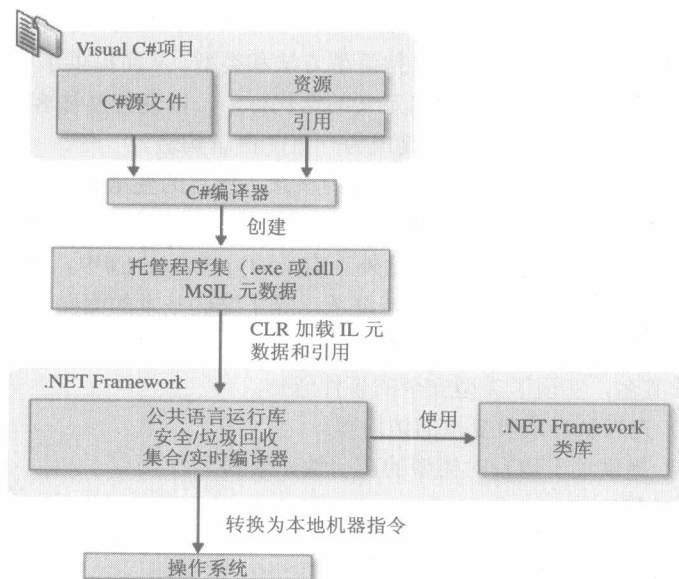


图 1-1 C#应用程序的编译与运行流程

语言互操作性是 .NET Framework 的一项主要功能，因为由 C#编译器生成的 IL 代码符合公共类型规范 (CTS)，因此从 C#生成的 IL 代码可以与从 Visual Basic、Visual C++、Visual J#的 .NET 版本或者其他 20 多种符合 CTS 语言中的任何一种生成的代码进行交互。单一程序集可能包含用不同 .NET 语言编写的多个模块，并且类型可以相互引用，就像它们是用同一种语言编写的。

除了运行时服务之外，.NET Framework 还包含一个由 4 000 多个类组成的内容详尽的库。这些类被组织为命名空间，为从文件输入和输出、字符串操作、XML 分析到 Windows 窗体控件的所有内容提供了各种有用的功能，典型的 C#应用程序使用 .NET Framework 类库广泛地处理各种常见任务。

1.1.2 C#语言与其他语言的比较

C#是一种面向组件的新型语言，它具有许多其他 .NET Framework 编程语言所共有的功能。C#的关键字大约只有 80 个，其中大多数关键字对于使用过 C、C++、Java 或 Visual Basic 的开发人员来说都很熟悉。不同语言的关键字之间存在语法差别，但这种差别通常很小。对于熟悉其他语言的程序员而言，了解 C#与其他语言的差异很有必要的。

1. 与 C 和 C++比较

C#与 C 和 C++的主要差异如下。

- (1) 内存用垃圾收集系统来管理，没有用来撤销 new 调用的 Delete 方法。
- (2) 不使用 (.h) 头文件，此类文件也不是必需的，现在将存储于头文件中的信息作为程序集的一部分存储。
- (3) 类定义中右花括号 {} 后面不必使用分号。
- (4) Main 方法的首字母大写，而且是静态类的成员，其返回类型为 int 或 void。