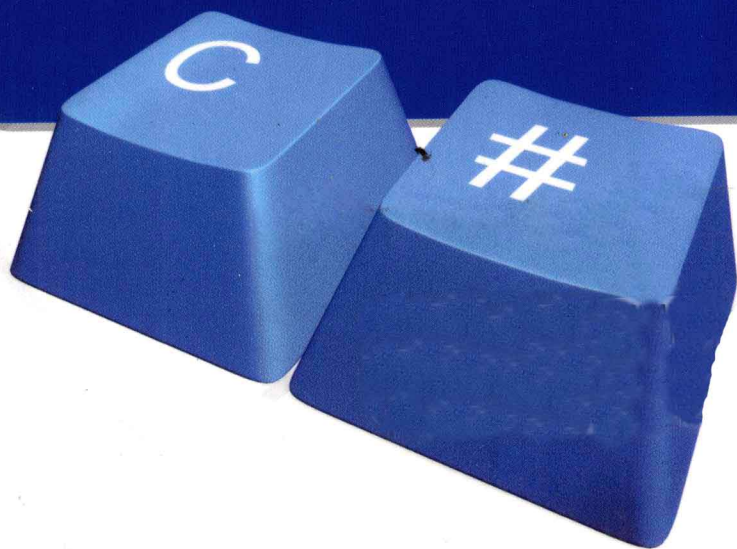


21 世纪面向工程应用型
计算机人才培养规划教材

罗福强 杨 剑 张敏辉 主 编
胡杰华 熊永福 温 林 副主编

C# 程序设计经典教程



清华大学出版社

21 世纪面向工程应用型计算机人才培养规划教材

C# 程序设计经典教程

罗福强 杨 剑 张敏辉 主 编
胡杰华 熊永福 温 林 副主编

清华大学出版社
北 京

内 容 简 介

C# 是微软公司推出的新一代编程语言。它在 C++ 的基础之上重新打造, 成为一门全新的完全面向对象的程序设计语言, 能够提供更高的可靠性和安全性, 不仅能用于开发传统的控制台应用程序和 Windows 应用程序, 还可用于开发 Web 应用程序、Silverlight 动画和 XNA 游戏。

全书共 14 章, 分为两部分, 第 1 部分为 C# 基础, 共 8 章, 重点介绍了 C# 的基本语法、面向对象的程序设计方法、基于事件驱动的程序设计方法以及程序的调试和异常处理方法。第 2 部分共 6 章, 在 C# 基础之上, 比较全面地展示了 C# 的各种应用技术, 包括 Windows 程序设计技术、数据库编程技术、文件操作与编程技术、XML 与 LINQ 高级数据访问技术、网络应用编程技术和多媒体编程技术等。本书内容丰富, 可操作性强, 叙述简洁流畅, 没有晦涩的术语, 所有实例精心设计, 能够帮助读者轻松、愉快地掌握 C# 的基本语法、编程方法和应用技巧。

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。

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。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C# 程序设计经典教程/罗福强等主编. —北京: 清华大学出版社, 2012. 2

(21 世纪面向工程应用型计算机人才培养规划教材)

ISBN 978-7-302-27505-3

I. ①C… II. ①罗… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 260582 号

责任编辑: 高买花 薛 阳

封面设计: 杨 兮

责任校对: 梁 毅

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京季蜂印刷有限公司

装 订 者: 三河市李旗庄少明印装厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24.75 字 数: 623 千字

版 次: 2012 年 2 月第 1 版 印 次: 2012 年 2 月第 1 次印刷

印 数: 1~3000

定 价: 39.50 元

产品编号: 041564-01

前言

C#是基于.NET Framework之上的微软公司推出的新一代编程语言。在保持了C++强大功能的同时,它又整合了Java语言的所有优点,成为一种全新的完全面向对象的开发语言。它简单、安全、灵活、功能强大,能够快速地开发各种应用软件。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,C#更容易理解和接受;相对于Java来说,C#更容易使用,开发软件的效率更高。C#既能用于开发传统的DOS应用程序、Windows应用程序和组件程序,还能用于开发Web应用程序、XML Web服务等。特别是C#将Web技术、Silverlight动画编程技术和XNA游戏编程技术紧密结合,使得它成为如今Web 2.0时代最有效的程序设计语言。

本书以Visual Studio .NET 2010和C# 4.0为蓝本。全书共分14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,重点阐述了Windows程序设计技术、数据库访问与编程技术、文件操作与编程技术、XML和LINQ高级数据访问技术、网络应用编程技术、多媒体编程技术等实用技术。

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#的语法讲透彻、讲清楚,文字叙述要简练;第二,紧紧围绕面向对象程序设计思想和可视化的Windows程序设计方法展开教学内容;第三,书中所有案例均精心设计,不仅代码完整,还贴近学生实际生活;第四,坚持零起点原则,学生可以在没有C/C++基础的情况下使用本书;第五,坚持应用为纲,全书分为基础篇和应用实战篇,特别在应用部分,全面展示了C#在各应用领域的编程技巧。

本书具有以下三个特点: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 2010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C#程序设计课程的教材或参考资料,也可供软件开发人员参考使用。

本书获电子科技大学成都学院教材建设项目资助。参与本书编写工作的有罗福强、杨剑、张敏辉、胡杰华、熊永福和温林等,前三位担任主编工作,后三位担任副主编工作。其中,

罗福强编写了第 8、11、12 和 14 章,杨剑编写了第 9、10 和 13 章,张敏辉编写了第 4、5、6 章,胡杰华编写第 2、3 章,熊永福编写第 1 章和所有上机实验,温林编写了第 7 章和所有习题。本书最后由罗福强统稿。

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

2011 年 10 月

目 录

第 1 章 C# 概述	1
1.1 C# 简介	1
1.1.1 C# 语言的发展	1
1.1.2 C# 语言的特点	2
1.2 C# 程序的开发环境	3
1.2.1 Visual Studio. NET 的基本操作	4
1.2.2 Visual Studio. NET 解决方案和项目文件的组织结构	7
1.2.3 Visual Studio. NET 的帮助	8
1.3 C# 程序入门	10
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 Win 32 应用程序	12
1.3.3 一个具有输入功能的 Win 32 应用程序	15
1.3.4 一个简单的 Web 应用程序	17
1.3.5 C# 程序的特点	19
习题	20
上机实验 1	21
第 2 章 C# 程序设计基础	22
2.1 常量与变量	22
2.1.1 常量	22
2.1.2 变量	24
2.2 C# 的数据类型	26
2.2.1 简单类型	26
2.2.2 枚举型 enum	27
2.2.3 结构型 struct	28
2.2.4 数据类型转换	30
2.3 运算符与表达式	31
2.3.1 算术运算符与表达式	32
2.3.2 赋值运算符与表达式	33
2.3.3 关系运算符与表达式	34
2.3.4 逻辑运算符与表达式	35
2.4 数组和字符串	36

2.4.1 一维数组	36
2.4.2 多维数组	39
2.4.3 数组型的数组	40
2.4.4 字符串 string	42
习题	44
上机实验 2	45
第 3 章 C# 程序的流程控制	47
3.1 C# 程序的分支语句	47
3.1.1 if 语句	47
3.1.2 switch 语句	49
3.1.3 分支语句的嵌套	52
3.2 C# 程序的循环语句	54
3.2.1 while 语句	54
3.2.2 do/while 语句	55
3.2.3 for 语句	56
3.2.4 foreach 语句	58
3.2.5 循环语句的嵌套	60
3.3 跳转语句	62
3.3.1 goto 语句	62
3.3.2 break 语句	63
3.3.3 continue 语句	64
习题	66
上机实验 3	67
第 4 章 面向对象程序设计入门	69
4.1 面向对象的基本概念	69
4.1.1 对象	70
4.1.2 事件与方法	70
4.1.3 类与对象	71
4.1.4 抽象、封装、继承与多态	71
4.2 类的定义	73
4.2.1 类的声明和实例化	73
4.2.2 类的数据成员和属性	76
4.2.3 类的可访问性	79
4.2.4 值类型与引用类型	79
4.3 类的方法	81
4.3.1 方法的声明与调用	81
4.3.2 方法的参数传递	83

4.3.3 方法的重载	91
4.4 构造函数	93
4.4.1 构造函数的声明和调用	93
4.4.2 构造函数的重载	94
4.4.3 构造函数与只读字段	96
4.4.4 对象的生命周期和析构函数	97
习题	98
上机实验 4	99
第 5 章 面向对象的高级程序设计	102
5.1 静态成员与静态类	102
5.1.1 静态成员	102
5.1.2 静态构造函数	105
5.1.3 静态类	106
5.2 类的继承性与多态性	106
5.2.1 类的继承性	106
5.2.2 类的多态性	111
5.3 抽象类与接口	115
5.3.1 抽象类	115
5.3.2 接口	119
5.3.3 抽象类与接口的比较	125
5.4 嵌套类、分部类与命名空间	125
5.4.1 嵌套类	125
5.4.2 分部类	127
5.4.3 命名空间	129
习题	130
上机实验 5	131
第 6 章 集合、索引器与泛型	135
6.1 集合	135
6.1.1 集合概述	135
6.1.2 ArrayList	136
6.1.3 哈希表 Hashtable	141
6.1.4 栈和队列	143
6.2 索引器	144
6.2.1 索引器的定义与使用	145
6.2.2 索引器与属性的比较	149
6.3 泛型	149
6.3.1 泛型概述	149

6.3.2 泛型集合	150
6.3.3 泛型类、泛型方法和泛型接口	152
习题	158
上机实验 6	158
第 7 章 程序调试与异常处理	160
7.1 程序错误	160
7.1.1 程序错误分类	160
7.1.2 调试程序错误	162
7.2 程序的异常处理	165
7.2.1 异常的概念	165
7.2.2 try-catch 语句与异常处理	166
7.2.3 finally 语句	168
7.2.4 throw 语句与抛出异常	169
习题	170
上机实验 7	170
第 8 章 基于事件驱动的程序设计技术	173
8.1 基于事件的编程思想	173
8.2 委托	175
8.2.1 委托概述	175
8.2.2 委托的声明、实例化与使用	175
8.2.3 多路广播与委托的组合	178
8.3 事件	179
8.3.1 事件的声明	179
8.3.2 事件的预订	181
8.3.3 事件的引用	182
8.4 基于事件的 Windows 编程	186
8.4.1 Windows 应用程序概述	186
8.4.2 Windows 窗体与事件驱动编程	188
习题	193
上机实验 8	194
第 9 章 Windows 程序的界面设计	199
9.1 窗体与控件概述	200
9.1.1 Windows 窗体	200
9.1.2 窗体的控件	201
9.2 按钮与文本显示、编辑控件	203
9.2.1 按钮控件	203

9.2.2	文本显示控件	205
9.2.3	文本编辑控件	206
9.2.4	应用实例	208
9.3	列表与选择控件	210
9.3.1	RadioButton 控件	210
9.3.2	CheckBox 控件	211
9.3.3	ListBox 控件	212
9.3.4	ComboBox 控件	213
9.3.5	其他常用控件	213
9.3.6	应用实例	213
9.4	图形显示控件	217
9.4.1	PictureBox 控件	217
9.4.2	ImageList 控件	217
9.4.3	应用实例	218
9.5	容器控件	219
9.5.1	GroupBox 控件	219
9.5.2	Panel 控件	219
9.5.3	TabControl 控件	219
9.5.4	应用实例	220
9.6	对话框	223
9.6.1	对话框概述	223
9.6.2	消息框	226
9.6.3	通用对话框	228
9.6.4	应用实例	230
9.7	菜单、工具栏和状态栏	231
9.7.1	菜单	231
9.7.2	工具栏	233
9.7.3	状态栏	234
9.7.4	应用实例	235
9.8	SDI 和 MDI 应用程序	237
9.8.1	创建 SDI 应用程序	237
9.8.2	创建 MDI 应用程序	237
9.8.3	应用实例	238
	习题	239
	上机实验 9	239
第 10 章	C# 数据库编程技术	244
10.1	数据库与 ADO.NET 概述	244
10.1.1	数据库概述	244

10.1.2	SQL 概述	247
10.1.3	ADO.NET 概述	249
10.2	使用 ADO.NET 访问数据库	250
10.2.1	使用 Connection 对象访问数据库	251
10.2.2	使用 Command 对象访问数据库	252
10.2.3	使用 DataReader 对象访问数据库	256
10.2.4	使用 DataAdapter 与 DataSet 对象操作数据库	262
习题		267
上机实验 10		267
第 11 章	文件操作与编程技术	270
11.1	文件的输入/输出	270
11.1.1	文件 I/O 与流	270
11.1.2	读写文本文件	271
11.1.3	读写二进制文件	273
11.1.4	对象的序列化	275
11.2	文件操作控件	278
11.2.1	SaveFileDialog 控件	278
11.2.2	OpenFileDialog 控件	280
11.2.3	FolderBrowserDialog 控件	281
11.2.4	应用实例: 简易的写字板程序	282
习题		285
上机实验 11		286
第 12 章	高级数据访问与处理技术	290
12.1	XML 编程	290
12.1.1	XML 概述	290
12.1.2	XML 文档的创建	292
12.1.3	XML 文档的查询	298
12.1.4	XML 文档的编辑	300
12.2	LINQ 编程	304
12.2.1	LINQ 概述	304
12.2.2	LINQ 的查询子句	307
12.2.3	LINQ to XML 的应用	311
12.2.4	LINQ to SQL 的应用	313
习题		320
上机实验 12		321

第 13 章 网络应用编程技术	323
13.1 网络编程基础	324
13.1.1 计算机网络的概述	324
13.1.2 计算机网络的通信协议	324
13.1.3 System.Net 概述	325
13.2 Socket 编程	333
13.2.1 Socket 编程概述	333
13.2.2 TCP 应用编程	335
13.2.3 UDP 应用编程	341
13.3 电子邮件的发送和接收	344
13.3.1 发送邮件	344
13.3.2 接收邮件	346
习题	350
上机实验 13	351
第 14 章 多媒体编程技术	354
14.1 GDI+ 绘图	354
14.1.1 GDI+ 概述	354
14.1.2 System.Drawing 命名空间	356
14.1.3 创建 Graphics 对象	356
14.1.4 颜料、钢笔和画笔	357
14.1.5 点、线和图形	359
14.1.6 图像和文本	366
14.1.7 坐标系统及变换	369
14.2 Windows Media Player 组件	375
14.2.1 Windows Media Player 组件的介绍	375
14.2.2 Windows Media Player 组件的使用	377
习题	380
上机实验 14	380
参考文献	382

第1章

C#概述

总体要求

- 了解 C# 语言的特点及其发展。
- 了解简单的三种应用程序(控制台应用程序、Windows 窗体应用程序和 Web 应用程序)的结构及其特点。
- 初步掌握 Visual Studio .NET 2010 中的操作方法,包括创建项目、编辑程序、生成和调试应用程序的方法。

相关知识点

- 了解计算机软件、计算机语言及分类的知识。
- 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。

学习难点

C# 程序的结构和特点。

1.1 C# 简介

1.1.1 C# 语言的发展

在过去的 20 年里,C 和 C++ 已经成为在商业软件的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。与 Visual Basic 等语言相比,同等级别的 C/C++ 应用程序往往需要更长的时间来开发。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平衡点。

目前,有些语言以牺牲灵活性的代价来提高效率。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),其所提供的功能难以令人满意。这些语言无法方便地同原来的系统交互,也无法与当前的网络编程很好地结合。

对于 C/C++ 用户来说,最理想的解决方案无疑是在快速开发的同时又可以调用底层平台的所有功能。他们想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,使得程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序。

它是从 C 和 C++ 语言演化而来的。它在语句、表达式和运算符方面使用了许多 C++ 的功能。它在类型安全性、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。它提供对常用 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,通常同 .NET Framework 一起,随 Visual Studio .NET 一起发布。目前,比较流行的 C# 主要有 4 个版本,分别是 C# 1.0、C# 2.0、C# 3.0 和 C# 4.0。C# 1.0 同 .NET Framework 1.1 随 Visual Studio .NET 2003 一起发布。C# 2.0 同 .NET Framework 2.0 随 Visual Studio .NET 2005 一起发布。C# 3.0 同 .NET Framework 3.5 随 Visual Studio .NET 2008 一起发布。而 C# 4.0 同 .NET Framework 4.0 随 Visual Studio .NET 2010 一起发布。本书以 C# 4.0、.NET Framework 4.0 和 Visual Studio .NET 2010 为范本,所有案例均在 Visual Studio .NET 2010 中经过调试运行无误。

1.1.2 C# 语言的特点

C# 是一种简洁、类型安全的面向对象的语言,开发人员可以使用它来构建在 .NET Framework 上运行的各种安全、可靠的应用程序,包括控制台应用程序、Windows 窗体应用程序、WPF 应用程序、Web 应用程序、Silverlight 应用程序、WCF 服务、分布式组件、客户-服务器应用程序、数据库应用程序等。

作为一种面向对象的语言,C# 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。C# 程序的生成过程比 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。C# 源文件可以定义任意数量的类、结构、接口和事件。

1. C# 的优点

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# 语言可实现具有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、WCF 服务和 .NET Framework。使用 C# 语言,一方面实现组件之间的相互调用,也就实现了使用不同的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

现了组件之间的跨互联网的调用。

(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,例如编写一个小工具来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# 增强了开发者的效率,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。

2. C# 4.0 的新增功能

相对于 C# 之前的版本,C# 4.0 新增了以下功能。

1) 动态支持

C# 4.0 引入了一种新类型 `dynamic`,以便支持对动态类型的后期绑定。借助这个新类型可实现许多新方案,包括 COM API(例如 Office API)、动态 API(例如 IronPython 库)和文档对象模型 (DOM) 的简化访问。

2) Office 可编程性

通过增加命名参数和可选参数、`dynamic` 类型、索引属性以及可选的 `ref` 修饰符,极大地增强了对 COM 接口(包括 Office API)的访问。

3) 类型等效性支持

现在部署的应用程序可以包含嵌入的类型信息来替换从主互操作程序集 (PIA) 导入的类型信息。包含嵌入类型信息的应用程序可以使用运行时中的类型,而无须引用运行时程序集。如果发布了各种版本的运行时程序集,则包含嵌入类型信息的应用程序可以使用各种版本,而不必重新编译。

4) 协变和逆变

通过“协变”,可以使用派生程度更大的类型;而通过“逆变”,可以使用派生程度更小的类型。这样可以对实现了变体接口的类进行隐式转换,并提高将方法签名与变体委托类型进行匹配的灵活性。可以使用新的 `in` 和 `out` 语言关键字创建变体接口和委托。.NET Framework 还引入了针对若干现有泛型接口和委托的变体支持,包括 `IEnumerable(Of T)` 接口以及 `Func(Of T, TResult)` 和 `Action(Of T)` 委托。

5) 新命令行选项

`/langversion` 命令行选项导致编译器只接受在指定版本 C# 中有效的语法。

`/appconfig` 编译器选项使 C# 应用程序能够向编译器指明程序集的应用程序配置文件的位置。

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 4 个版本: Visual Studio .NET 2003、Visual Studio .NET 2005、Visual

Studio .NET 2008 和 Visual Studio .NET 2010。本节将主要介绍 Visual Studio .NET 2010 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 2010 后,用户只需选择“开始”→“所有程序”→Microsoft Visual Studio 2010→Microsoft Visual Studio 2010 系统菜单即可启动 Visual Studio.NET 2010。启动成功后,显示 Visual Studio.NET 2010 的操作窗口,如图 1-1 所示。

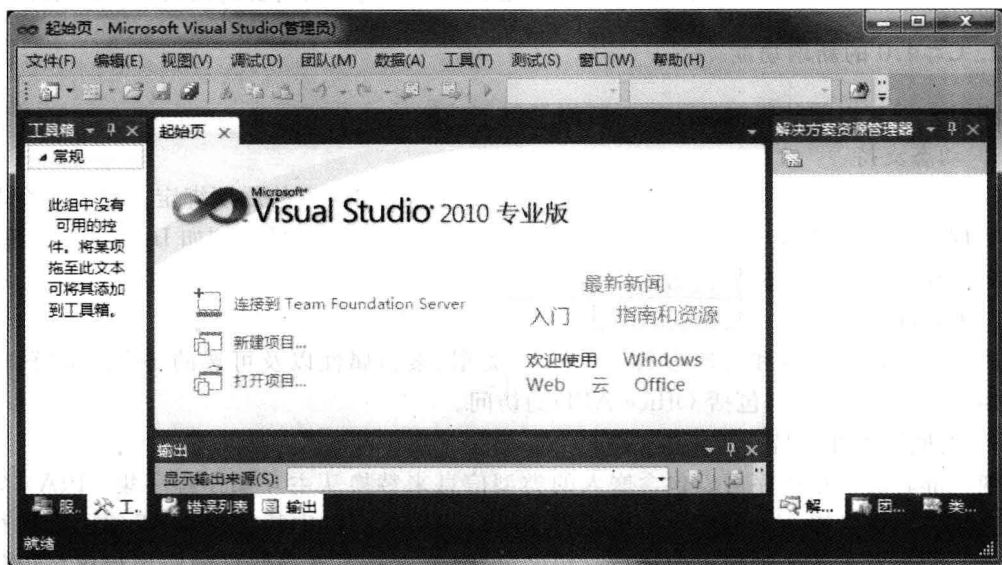


图 1-1 Visual Studio .NET 2010 窗口

刚启动的 Visual Studio.NET 2010 的窗口由菜单栏、工具栏、工具箱、起始页、解决方案资源管理器等组成。其中,菜单栏列出 Visual Studio .NET 2010 的所有操作命令,工具栏则列出常用的操作命令,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以及文件列表,工具箱用于显示在设计应用程序操作界面时所要使用的可视化控件。

1. 应用程序项目的创建与打开

1) 新建项目

Visual Studio .NET 2010 是一个高度集成的开发工具。它集 Visual Basic、C++、C# 和 F# 四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)、WCF 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,然后根据需要选择所有创建的项目即可,如图 1-2 所示。例如,要创建 Windows 窗体应用程序,首先在



图 1-2 “新建项目”对话框

“已安装的模板”列表框中选择 Visual C# 及其子节点 Windows, 然后在中间的模板列表框中选择“Windows 窗体应用程序”, 接着在“名称”文本框中输入应用程序项目的名字(如“MyFirstApp”), 在“位置”下拉列表框中可设置应用程序项目的创建位置(如“E:\我的文档\Visual Studio 2010\Projects”), 在“解决方案名称”文本框中输入解决方案的名称(如“MyFirstApp”), 最后单击“确定”按钮即可。

如果选择“文件”→“新建”→“网站”菜单命令, 则弹出“新建网站”对话框, 然后根据需要选择所有创建的项目即可, 如图 1-3 所示。在创建网站时, 应注意以下几点。

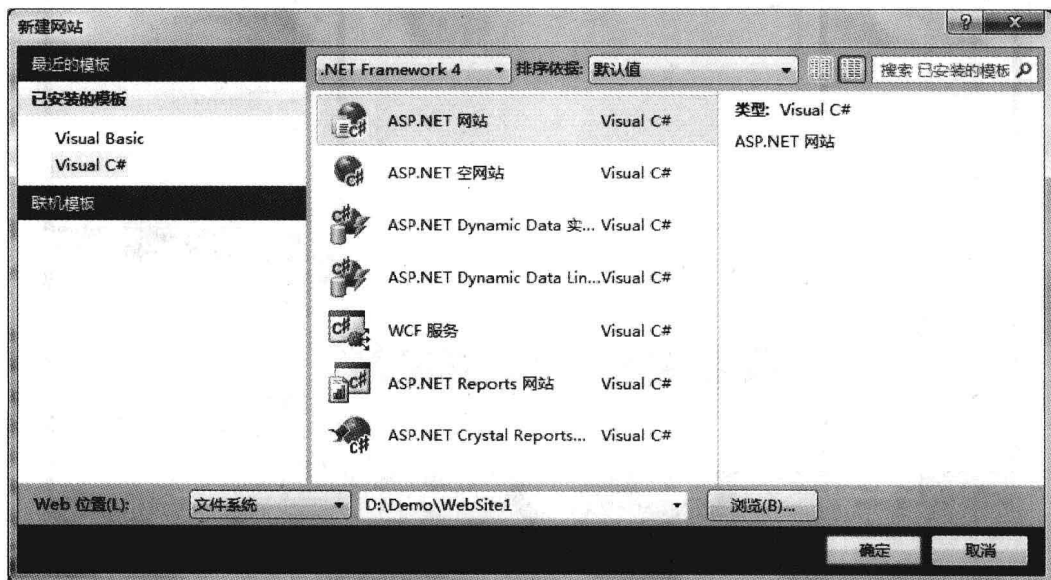


图 1-3 “新建网站”对话框