



Visual C#.NET

程序设计实用教程

李康乐	主 编
张玉芬	赵立波 副主编
李冰冰	杨 萌 编 著
齐景嘉	主 审

21 世纪高等学校计算机应用技术规划教材

Visual C#.NET 程序设计实用教程

李康乐 主编
张玉芬 赵立波 副主编
李冰冰 杨萌 编著

清华大学出版社
北 京

内 容 简 介

本书以 Visual Studio 2008 为开发平台,全面系统地介绍了 C# 面向对象的编程思想,将面向对象的程序设计方法贯穿始终。全书共 11 章,介绍了 C# 语言基础、程序流程控制、Windows 窗体程序、数组和集合、类和对象、面向对象技术、异常处理、数据库编程和文件操作等,每章都配有本章小结和习题,以方便读者巩固所学知识。特别地,在应用性较强的章中,多加一节具有实际应用的案例,通过案例将各知识点结合起来,达到学以致用目的。

本书案例典型,相关知识讲解系统,内容丰富。适合作为普通高等院校计算机及相关专业的教材,也可作为初学编程人员的自学用书。

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

Visual C#.NET 程序设计实用教程/李康乐主编.--北京:清华大学出版社,2014

21 世纪高等学校计算机应用技术规划教材

ISBN 978-7-302-35156-6

I. ①V… II. ①李… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 013717 号

责任编辑:付弘宇 薛 阳

封面设计:杨 兮

责任校对:李建庄

责任印制:李红英

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.5

字 数:534 千字

版 次:2014 年 6 月第 1 版

印 次:2014 年 6 月第 1 次印刷

印 数:1~2000

定 价:36.00 元

产品编号:057526-01

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机应用技术规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

本书是黑龙江省高等教育教学改革项目(项目名称:计算机科学与技术专业金融应用人才培养模式的研究与实践;项目编号:JG2012010240)的部分研究成果。

C#是微软公司专门为.NET应用开发的一种全新且简单、安全、面向对象的程序设计语言。它吸取了C/C++、Delphi、Java等语言的优点,体现了当今最新的程序设计技术的功能和精华,因此,赢得了越来越多的程序开发人员的喜爱。C#不仅适合开发传统的Windows客户端应用程序,还特别适合于开发数据库应用程序和企业级Web应用程序,近年来已成为一门主流语言。

本书基于Visual Studio 2008开发环境,通过通俗易懂的语言和大量生动典型的实例,由浅入深、循序渐进地介绍使用C#进行程序开发的常用技术和方法。书中的实例全部出自编者实际教学和工作过程中所采用的实例,都在C#平台上编译调试通过,以方便读者自学理解。书中源程序注释清晰明了,可以直接使用和更改,方便自行修改和升级。

全书共11章,分别介绍了C#语言基础、程序流程控制、Windows窗体程序、数组和集合、面向对象编程技术、异常处理、界面设计、数据库编程和文件操作等内容,每章都配有本章小结和习题,以方便读者巩固所学知识。

与市场上其他的C#教程相比,本书具有以下特点:

1. 结构合理,详略得当

本书结构安排合理、由浅入深,将面向对象的程序设计方法贯穿始终,让读者能够逐步体会并掌握面向对象技术的精髓。既避开了晦涩难懂的理论知识,又覆盖了编程所需的各方面技术。

2. 循序渐进,轻松上手

本书内容叙述从零起步,循序渐进,全面提高学、练、用的能力。通过实例,可以使读者轻松上手,快速掌握所学内容。

3. 实例丰富,贴近实际

本书每部分内容都配有示例,简单易懂,帮助读者理解相关知识内容。特别地,在应用性较强的章中,多加一节具有实际应用的案例,通过案例将各知识点结合起来,达到学以致用的目的。

4. 图文并茂,步骤详细

本书在讲解技术和例题时,图文并茂,步骤详细,读者只需要按照步骤操作,就可以体会到编程带来的乐趣和成就感。

本书可作为普通高等院校计算机及其相关专业的教材,也可作为初学编程人员的自学用书。在清华大学出版社的网站(<http://www.tup.tsinghua.edu.cn>)上提供了本书的多媒体课件和所有例题源代码。

本书由李康乐任主编、由张玉芬和赵立波任副主编,其中第2章和第4章由李康乐编写,第1章、第3章和第5章由张玉芬编写,第6章和第7章由赵立波编写,第8章和第11章由李冰冰编写,第9章由李康乐和杨萌共同编写,第10章由张玉芬和杨萌共同编写,全书由李康乐统稿。

由于时间仓促、作者水平有限,书中难免存在疏漏和不足,恳请读者批评指正,使本书得以改进和完善。编者联系邮箱: lk1e_216@163.com 或 fuhy@tup.tsinghua.edu.cn。

编 者

2014年4月

目 录

第 1 章 概述	1
1.1 .NET Framework 简介	1
1.2 C# 简介	3
1.3 Visual Studio 2008 集成开发环境	3
1.4 C# 程序的建立与执行	11
1.4.1 控制台应用程序	11
1.4.2 Windows 窗体应用程序	14
本章小结	22
习题	23
第 2 章 C# 语言基础	24
2.1 C# 程序结构	24
2.1.1 程序的组成要素	24
2.1.2 语法格式中的符号约定	27
2.2 数据类型	27
2.2.1 值类型	27
2.2.2 引用类型	31
2.3 变量和常量	33
2.3.1 变量	33
2.3.2 常量	34
2.3.3 类型转换	35
2.4 运算符和表达式	38
2.4.1 运算符与表达式类型	38
2.4.2 运算符的优先级	41
2.5 常用 .NET 框架类	41
2.5.1 Random 类	41
2.5.2 Math 类	42
2.5.3 DateTime 类	43
本章小结	45
习题	45
第 3 章 程序流程控制	47
3.1 顺序结构	47

3.1.1 赋值语句	47
3.1.2 输入语句与输出语句	48
3.2 选择结构	54
3.2.1 if 语句	54
3.2.2 switch 语句	60
3.3 循环结构	62
3.3.1 while 语句	62
3.3.2 do-while 语句	63
3.3.3 for 语句	64
3.3.4 foreach 语句	66
3.3.5 循环嵌套	67
3.3.6 跳转语句	68
3.4 程序流程控制的应用	72
本章小结	81
习题	81

第4章 Windows 窗体程序

4.1 窗体	84
4.1.1 窗体的组成	84
4.1.2 窗体的属性	85
4.1.3 窗体的方法	86
4.1.4 窗体的事件	86
4.1.5 窗体的布局	87
4.2 常用控件	89
4.2.1 基本控件	90
4.2.2 选择类控件	92
4.2.3 PictureBox 控件和 ImageList 组件	101
4.2.4 Timer 组件和 ProgressBar 控件	105
4.3 容器控件	109
4.3.1 GroupBox 控件	109
4.3.2 Panel 控件	109
4.3.3 TabControl 控件	109
本章小结	112
习题	112

第5章 数组和集合

5.1 一维数组	114
5.1.1 一维数组的声明	114
5.1.2 一维数组的初始化	115

5.1.3 访问一维数组中的元素	116
5.2 二维数组	119
5.2.1 二维数组的声明	120
5.2.2 二维数组的初始化	121
5.2.3 访问二维数组中的元素	121
5.3 集合	124
5.3.1 ArrayList 集合类	125
5.3.2 HashTable 集合	131
5.4 数组的应用	133
本章小结	138
习题	139
第 6 章 面向对象程序设计基础	141
6.1 面向对象编程	141
6.1.1 面向对象编程简介	141
6.1.2 面向对象编程语言的特点	142
6.2 类和对象	143
6.2.1 定义一个类	144
6.2.2 对象的创建	145
6.2.3 类的成员简介	146
6.3 字段	147
6.4 方法	148
6.4.1 定义方法	149
6.4.2 方法的参数和返回值	150
6.4.3 方法的重载	157
6.4.4 变量的作用域	160
6.5 this 关键字	161
6.6 构造函数和析构函数	163
6.6.1 构造函数	163
6.6.2 析构函数	168
6.7 属性	169
6.8 类的静态成员	174
本章小结	182
习题	182
第 7 章 面向对象技术	186
7.1 继承	186
7.1.1 继承的实现	186
7.1.2 基类成员的隐藏	189

7.1.3	派生类的构造函数	191
7.2	多态	194
7.2.1	虚方法	194
7.2.2	抽象类和抽象方法	197
7.2.3	密封类和密封方法	198
7.3	接口	199
7.3.1	接口的声明	200
7.3.2	接口的实现	201
7.3.3	接口和抽象类	206
7.4	委托	206
7.4.1	委托的声明	207
7.4.2	委托的使用	208
7.5	事件	210
7.5.1	使用事件	211
7.5.2	定义事件	213
7.6	综合应用	215
	本章小结	222
	习题	222
第8章	异常处理	225
8.1	异常处理	225
8.1.1	异常类	225
8.1.2	引发异常	227
8.1.3	异常处理机制	228
8.2	程序调试	230
8.2.1	程序错误	231
8.2.2	程序调试	231
	本章小结	234
	习题	234
第9章	界面设计	236
9.1	菜单、工具栏与状态栏	236
9.1.1	菜单	236
9.1.2	工具栏	241
9.1.3	状态栏	244
9.2	对话框	246
9.2.1	通用对话框	246
9.2.2	自定义对话框	250
9.3	多文档操作	250

本章小结	253
习题	253
第 10 章 数据库编程	255
10.1 SQL 基础知识	255
10.1.1 查询语句	256
10.1.2 插入语句	258
10.1.3 修改语句	258
10.1.4 删除语句	259
10.2 ADO.NET 概述	259
10.2.1 ADO.NET 概念	259
10.2.2 ADO.NET 对象模型	259
10.2.3 ADO.NET 访问数据库的两种模式	262
10.3 利用 ADO.NET 访问数据库	264
10.3.1 Connection 对象	264
10.3.2 Command 对象	268
10.3.3 DataReader 对象	271
10.3.4 DataAdapter 对象	277
10.3.5 DataSet 对象	279
10.3.6 ADO.NET 相关组件	283
10.3.7 数据绑定	291
10.4 数据库技术的应用	297
本章小结	306
习题	306
第 11 章 文件和流	308
11.1 文件和流的概念	308
11.2 文件的存储管理	309
11.2.1 DriveInfo 类	309
11.2.2 Directory 类和 DirectoryInfo 类	310
11.2.3 Path 类	312
11.2.4 File 类和 FileInfo 类	313
11.3 文件的操作	315
11.3.1 Stream 类	316
11.3.2 FileStream 类	319
11.3.3 StreamReader 类和 StreamWriter 类	321
11.3.4 BinaryReader 类和 BinaryWriter 类	326
本章小结	328
习题	328
参考文献	329

1.1 .NET Framework 简介

Microsoft .NET(简称.NET)是微软公司推出的面向网络的一套完整的开发平台,从程序员的角度来看,.NET 是一组用于生成 Web 服务器应用程序、Web 应用程序、Windows 应用程序和移动应用程序的软件组件,用该平台建立的应用程序在公共语言运行库的控制下运行。

1. .NET 的特点

(1) 统一应用层接口

.NET 框架将 Windows 操作系统底层的 API(Application Programming Interface,应用程序接口)进行封装,为各种 Windows 操作系统提供统一的应用层接口,从而消除了不同 Windows 操作系统带来的不一致性,用户只需直接调用 API 进行开发,无需考虑平台。

(2) 面向对象开发

.NET 框架使用面向对象的设计思想,更加强调代码和组件的重用性,提供了大量的类库,每个类库都是一个独立的模块,供用户调用。同时,开发者也可自行开发类库给其他开发者使用。

(3) 支持多种语言

.NET 框架支持多种开发语言,允许用户使用符合公共语言运行库 CLR(Common Language Runtime)规范的多种编程语言开发程序,包括 C#、VB.NET、J#、C++等,然后再将代码转换为中间语言存储到可执行程序中。在执行程序时,通过.NET 组件对中间语言进行编译执行。

2. .NET 体系结构

.NET 体系结构的核心是.NET 框架(.NET Framework),.NET 框架在操作系统之上为程序员提供了一个编写各种应用程序的高效的工具和环境,如图 1-1 所示。.NET 体系结构的顶层是用各种语言所编写的应用程序,这些应用程序由公共语言运行库控制执行。

.NET 能支持多种应用程序的开发,其中控制台程序是一种传统而简单的程序形式,一般是字符界面,可以编译为独立的可执行文件,通过命令行运行,在字符界面上输入和输出。

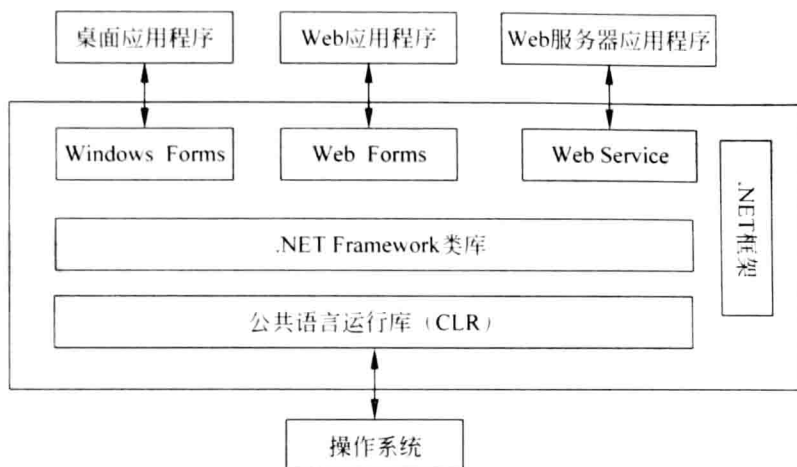


图 1-1 .NET 框架

Windows 应用程序是基于 Windows Form (Windows 窗体) 的应用程序, 是一种基于图形用户界面的应用程序, 一般是在用户计算机本地运行。本书案例采用的是控制台应用程序或 Windows 应用程序。

3. .NET 框架的两个实体

.NET 框架具有两个主要组件, 即公共语言运行库和 .NET Framework 类库。

(1) 公共语言运行库

公共语言运行库又称公共语言运行时 (Common Language Runtime, CLR) 或公共语言运行环境, 是 .NET 框架的底层。其基本功能是管理用 .NET 框架类库开发的所有应用程序的运行并且提供各种服务。

使用 CLR 的一大好处是支持跨语言编程, 即 .NET 将开发语言与运行环境分开, 凡是符合公共语言规范 (Common Language Specification, CLS) 的语言所编写的对象都可以在 CLR 上互相通信、互相调用。这是因为基于 .NET 平台的所有语言的共同特性 (如数据类型、异常处理等) 都是在 CLR 层面实现的。例如, 用 C# 语言编写的应用程序, 能够使用 VB.NET 编写的类库和组件, 反之亦然, 这大大提高了开发人员的工作效率。

(2) .NET Framework 类库

.NET Framework 类库是一个面向对象的可重用类型集合, 该类型集合可以理解成预先编写好的程序代码库, 这些代码包括一组丰富的类与接口, 程序员可以用这些现成的类和接口来生成 .NET 应用程序、控件和组件。例如, Windows 窗体类是一组综合性的可重用的类型, 使用这些类型可以轻松灵活地创建窗体、菜单、工具栏、按钮和其他屏幕元素, 从而大大简化了 Windows 应用程序的开发。

.NET 支持的所有语言都能使用类库, 任何语言使用类库的方式都是一样的。程序员可以直接使用类库中的具体类, 或者从这些类派生出自己的类。 .NET 框架类库是程序员必须掌握的工具, 熟练使用类库是每个程序员的基本功。

4. Microsoft 中间语言和即时编译器

.NET 框架上可以集成几十种编程语言, 这些编程语言共享 .NET 框架的庞大资源, 还可以创建由不同语言混合编写的应用程序, 因此可以说 .NET 是跨语言的集成开发平台。

.NET 框架上的各种语言分别有各自不同的编译器,编译器向 CLR 提供原始信息,各种编程语言编译器负责完成编译工作的第一步,即把源代码转换为用微软中间语言(Microsoft Intermediate Language, MSIL)表示的中间代码,如图 1-2 所示。MSIL 是一种非常接近机器语言的语言,但还不能直接在计算机上运行。第二步编译工作就是将中间代码转换为可执行的本地机器指令(本地代码),在 CLR 中执行,这个工作由 CLR 中包含的即时编译器(Just In Time, JIT)完成。

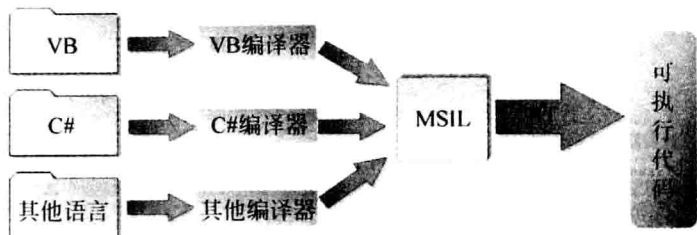


图 1-2 .NET 程序编译过程

1.2 C# 简介

C# 语言是微软公司专门为 .NET 平台量身打造的程序设计语言,是一种强大的、基于现代面向对象设计方法的程序设计语言,它是为生成运行在 .NET 框架上的企业级应用程序而设计的。

微软对 C# 的定义: C# 是一种安全的、现代的、简单的,由 C 和 C++ 衍生而来的面向对象的编程语言。设计 C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力, C# 已经成为 Windows 平台上软件开发的绝对主流语言。

作为 .NET 的核心语言, C# 有很多的优点,例如完全面向对象的设计、组件技术、跨平台异常处理、强大的类型安全、自动的垃圾回收功能和版本处理技术等,读者将在后续的 C# 学习与使用中深入体会这些优点。

1.3 Visual Studio 2008 集成开发环境

Visual Studio 2008 的安装步骤较简单,这里不再介绍。在 Visual Studio 2008 中, Visual C# 2008、Visual Basic 2008 和 Visual C++ 2008 的绝大多数的界面和功能是相同的,下面将介绍 Visual Studio 2008 编程环境下的 Visual C# 2008 开发界面。

1. Visual C# 2008 的主界面

Visual C# 2008 不仅是一个程序编辑器,它还是一个集成开发工具,集程序的设计、编译、调试和运行等多种功能于一体。但每个功能都相对独立,具有自己的界面,下面介绍一下 Visual C# 2008 集成开发环境。

打开 Visual Studio 2008 的方法为: 单击“开始”菜单,选择“程序”,然后找到并单击 Microsoft Visual Studio 2008 子菜单项,则会启动 Visual Studio 2008 集成开发环境。如果

是第一次运行 Visual Studio 2008, 会出现一个如图 1-3 所示的对话框, 提示用户选择默认的开发环境。

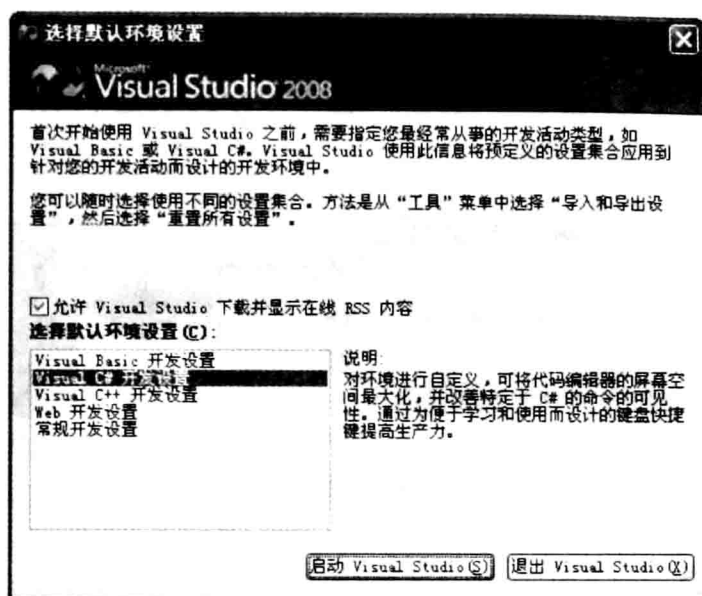


图 1-3 选择默认环境设置

Visual Studio 2008 可以根据用户的首选开发语言自己进行调整, 在集成开发环境中, 各个对话框工具针对用户选择的语言建立它们的默认设置。从列表中选择“Visual C# 开发设置”, 再单击“启动 Visual Studio”按钮, 就会出现 Visual Studio 2008 集成开发环境, 如图 1-4 所示。

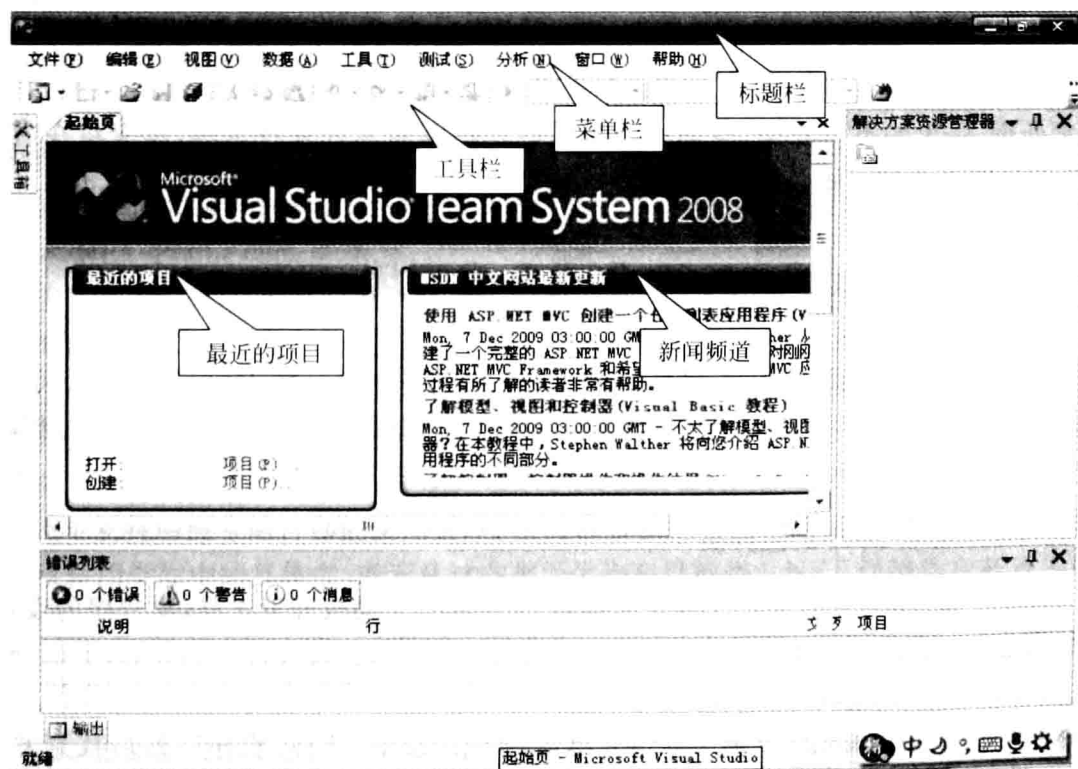


图 1-4 Visual Studio 2008 起始页

选择“文件”→“新建”→“项目”(或在起始页中选择“创建项目”)命令,会弹出如图 1-5 所示的对话框。

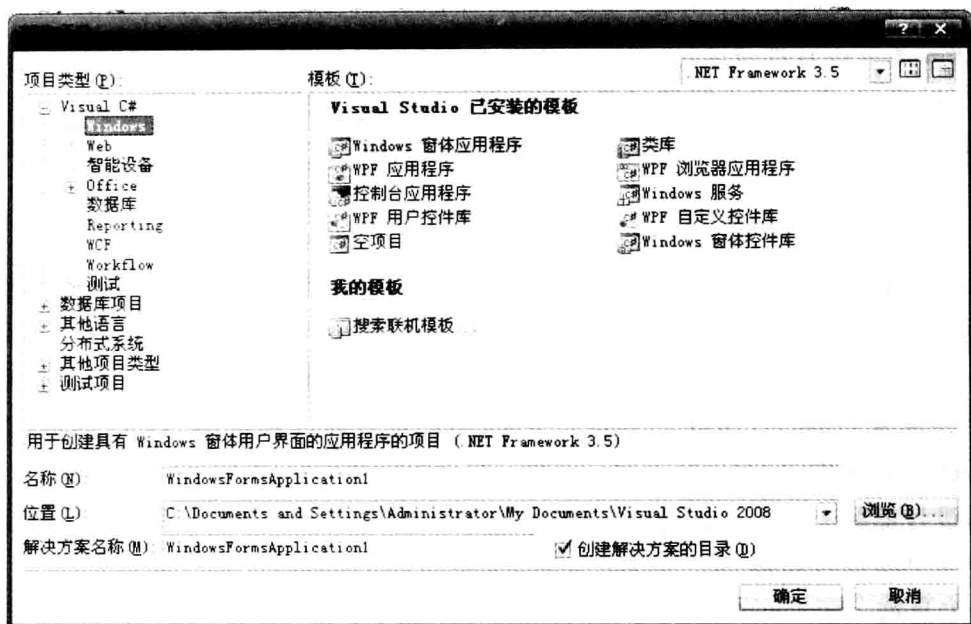


图 1-5 “新建项目”对话框

选择 Visual C# 项目的 Windows, 打开 Visual Studio 已安装的模板, 选择“Windows 窗体应用程序”, 输入项目的名称, 并选择项目的存放位置(在这里保留默认的设置), 然后单击“确定”按钮, 出现 Visual C# 2008 的主界面, 如图 1-6 所示。

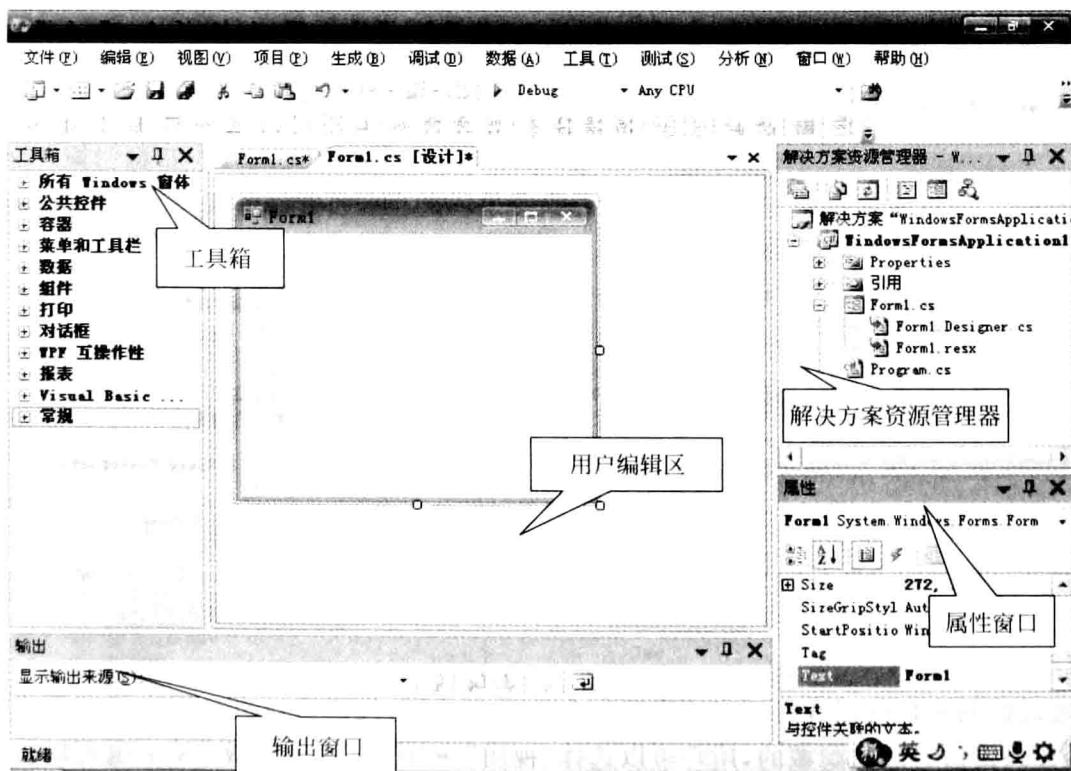


图 1-6 Visual C# 2008 主界面