

高等学校计算机应用规划教材

# C#语言程序设计教程

## (微课版)

王先水 彭玉华 刘 艳 编著

清华大学出版社

北 京

## 内 容 简 介

本书以学生成绩管理系统项目为主线,以 Windows 窗体应用程序开发为载体,讲解面向对象 C#语言的基础知识和项目设计的基本技能,定位应用型人才培养为目标。

本书共 13 章,内容涵盖 C#开发环境概述、C#语言程序设计基础、字符串和数组、类和方法、继承和多态、集合和泛型、调试和异常处理、委托和事件、Windows 窗体应用程序、文件和流、进程和线程、ADO.NET 技术、数据绑定技术。

本书理论与实践相结合,注重基础,案例丰富,步骤完整,Windows 窗体应用程序界面设计、事件驱动后台代码设计详细,实例均在 Visual Studio 2013 环境下通过。

本书可作为高等院校计算机及相关专业的教材,也可作为计算机编程爱好者的自学教材。

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#语言程序设计教程:微课版/王先水,彭玉华,刘艳 编著. —北京:清华大学出版社,2020.8

高等学校计算机应用规划教材

ISBN 978-7-302-55055-6

I. ①C… II. ①王… ②彭… ③刘… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2020)第 040723 号

责任编辑:刘金喜

封面设计:高娟妮

版式设计:思创景点

责任校对:马遥遥

责任印制:丛怀宇

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:23.75

字 数:623 千字

版 次:2020 年 8 月第 1 版

印 次:2020 年 8 月第 1 次印刷

定 价:68.00 元

---

产品编号:086354-01

# 前 言

C#是微软公司在 2000 年为 Visual Studio 开发平台推出的一款简洁、类型安全的面向对象的编程语言，开发人员通过它可以编写在 .NET Framework 上运行的各种安全可靠的应用程序。Visual Studio 开发平台则凭借其强大的可视化用户界面设计，让软件开发人员从复杂的界面设计中解脱出来，使软件开发成为一种享受。C#可用于开发数据库管理系统，也可用于开发集声音、动画、视频为一体的多媒体应用程序和 Web 应用程序，已逐渐成为软件开发市场使用的主流语言。

按三部委(教育部、发改委、财政部)联合印发本科高校向应用型转变指导意见，计算机领域的教学改革必须与计算机技术的发展相适应，需要用最新知识、教材、手段并结合学生的实际情况进行教学，用科学性强且简单易懂、生动活泼的形式组织教学内容，实现“以学生动手能力为基础，以运用知识解决问题为突破口，以基础知识+上机实训+项目实训模式组织教学”的应用型人才培养目标。

## 1. 主要特色

### 1) 组织合理

本书以学生成绩管理系统项目为主线，以 Windows 窗体应用程序开发为载体，以应用型人才培养为目标组织教材内容，设计教学实例，提供微视频，各个实例有具体详细的界面设计、代码设计、测试效果。

### 2) 资源丰富

本书采用 Visual Studio 2013 为开发平台，提供教材所有实例及上机实验项目的源代码，提供教学大纲、教学课件(PPT)、教学讲稿、教学实例微视频辅助教师的教学。

### 3) 易学易用

本书为面向对象编程语言，以窗体应用程序开发讲解 C#语言基础知识、类的设计、对象的实例化、面向对象程序设计的基本特征封装、继承、多态、抽象、接口、ADO 数据库技术及数据绑定技术。详细介绍窗体应用程序的界面设计、事件驱动后台代码设计，对代码进行大量注释说明，对教师的教和学生的学起到较大的指导作用。

## 2. 教学内容

全书共 13 章，各章内容如下。

第 1 章 C#开发环境概述，重点讲述 .NET Framework 体系结构，Visual Studio 2013 的安装及开发 Windows 窗体程序的具体步骤。

第2章 C#语言程序设计基础，重点讲述 C#语言基本数据类型、运算符、常量和变量、选择语句、循环语句。

第3章 字符串和数组，重点讲述 C#的常用字符串、数据类型的转换、正则表达式、一维数组、枚举和结构体。

第4章 类和方法，重点讲述类的设计、方法的设计、构造方法及重载、属性的作用、几个常用类的属性及方法。

第5章 继承和多态，重点讲述继承的应用、多态的实现、抽象类和抽象方法的实现、接口的实现。

第6章 集合和泛型，重点讲述 ArrayList 类的属性和方法的应用、Queue 类和 Stack 类的属性和方法的应用、Hashtable 类和 SortedList 类的属性和方法的应用、泛型类、泛型方法、泛型集合的高级应用。

第7章 调试和异常处理，重点讲述 try...catch...finally 形式语句的应用。

第8章 委托和事件，重点讲述命名方法委托、多播委托、事件。

第9章 Windows 窗体应用程序，重点讲述窗体属性、事件和方法、窗体中的基本控件、窗体中的对话框控件、窗体间的数据交互。

第10章 文件和流，重点讲述文件基本操作、流的基本应用。

第11章 进程和线程，重点讲述进程的基本操作、线程的基本操作。

第12章 ADO.NET 技术，重点讲述 ADO.NET 五大对象、ADO.NET 技术操作数据库实现的增删改查。

第13章 数据绑定技术，重点讲述数据视图控件使用代码法绑定数据的基本方法和基本应用。

本书在武汉工程科技学院信息工程学院及武昌理工学院人工智能学院的大力支持下，由武汉工程科技学院信息工程学院计算机科学与技术教研室王先水、刘艳，武昌理工学院人工智能学院计算机科学与技术教研室彭玉华三位老师共同编写完成。书中的实例全部来自教师上课的讲稿，同时引用了参考文献中列举的 C#语言程序设计相关书籍中的部分内容，吸取了同行的宝贵经验，在此表示谢意。

由于编者水平有限，书中难免有不妥和疏漏之处，敬请广大读者批评指正。

本书教学资源可通过扫描下方二维码获取。

服务邮箱：476371891@qq.com



教学资源下载

编者

2019年10月于武汉

# 目 录

<b>第 1 章 C#开发环境概述</b> ..... 1	
1.1 C#概述..... 1	
1.2 .NET Framework概述..... 1	
1.3 Visual Studio 2013的安装与卸载... 2	
1.3.1 安装的必备条件..... 2	
1.3.2 安装与卸载的步骤..... 3	
1.4 Visual Studio 2013的开发环境..... 4	
1.4.1 Visual Studio 2013中常用的菜单..... 4	
1.4.2 C#程序入门..... 6	
习题1..... 11	
<b>第 2 章 C#语言程序设计基础</b> ..... 12	
2.1 基本数据类型..... 12	
2.1.1 整型..... 12	
2.1.2 浮点型..... 13	
2.1.3 字符型和字符串型..... 13	
2.1.4 布尔类型..... 13	
2.2 运算符..... 14	
2.2.1 算术运算符..... 14	
2.2.2 逻辑运算符..... 16	
2.2.3 比较运算符..... 16	
2.2.4 赋值运算符..... 17	
2.2.5 三元运算符..... 17	
2.2.6 运算符的优先级..... 17	
2.3 常量和变量..... 18	
2.3.1 命名规范..... 18	
2.3.2 声明常量..... 19	
2.3.3 声明变量..... 21	
2.4 选择语句..... 23	
2.4.1 if语句..... 23	
2.4.2 switch语句..... 27	
2.5 循环语句..... 30	
2.5.1 for循环语句..... 30	
2.5.2 while循环语句..... 34	
2.5.3 do...while循环语句..... 36	
2.5.4 跳转语句..... 38	
习题2..... 42	
上机实验2..... 42	
<b>第 3 章 字符串和数组</b> ..... 44	
3.1 字符串..... 44	
3.1.1 常用字符串操作..... 44	
3.1.2 数据类型转换..... 47	
3.1.3 正则表达式..... 50	
3.2 数组..... 52	
3.2.1 一维数组..... 52	
3.2.2 多维数组..... 57	
3.3 枚举和结构体..... 59	
3.3.1 枚举..... 59	
3.3.2 结构体..... 62	
习题3..... 66	
上机实验3..... 67	
<b>第 4 章 类和方法</b> ..... 69	
4.1 面向对象程序设计思想..... 69	
4.2 类与类的成员..... 70	
4.2.1 类的定义..... 70	
4.2.2 字段..... 71	
4.2.3 定义方法..... 72	
4.2.4 定义属性..... 74	
4.2.5 访问类的成员..... 78	
4.3 构造方法及方法重载..... 83	
4.3.1 构造方法..... 83	

4.3.2	析构方法	86	6.2.1	泛型概述	159
4.3.3	方法的重载	86	6.2.2	可空类型	161
4.3.4	方法中的参数	88	6.2.3	泛型方法	163
4.4	嵌套类与部分类	93	6.2.4	泛型类	165
4.4.1	嵌套类	93	6.2.5	泛型集合	167
4.4.2	部分类	95	6.2.6	泛型高级应用	173
4.5	常用类介绍	97	习题6		174
4.5.1	Console类	97	上机实验6		174
4.5.2	Random类	98	第7章 调试和异常处理		182
4.5.3	DateTime类	100	7.1	异常类	182
4.5.4	string类	102	7.2	异常处理语句	182
习题4		106	7.2.1	try...catch形式的应用	182
上机实验4		107	7.2.2	try...finally形式的应用	187
第5章 继承和多态		111	7.2.3	try...catch...finally形式的应用	190
5.1	继承	111	7.3	自定义异常	192
5.1.1	继承的概念	111	7.4	调试	194
5.1.2	使用类图表示继承关系	115	7.4.1	常用的调试语句	194
5.1.3	Object类	115	7.4.2	调试程序	196
5.2	多态	116	习题7		197
5.2.1	多态的概念	116	上机实验7		197
5.2.2	继承实现多态	116	第8章 委托和事件		200
5.3	抽象	120	8.1	委托	200
5.3.1	抽象类	120	8.1.1	命名方法委托	200
5.3.2	抽象方法	121	8.1.2	多播委托	205
5.3.3	继承实现抽象	121	8.1.3	匿名委托	207
5.4	接口	125	8.2	事件	211
5.4.1	接口的定义	125	习题8		216
5.4.2	接口的实现	126	上机实验8		216
5.4.3	接口与抽象的比较	130	第9章 Windows 窗体应用程序		219
5.4.4	使用接口实现多态	130	9.1	Windows窗体程序	219
习题5		132	9.1.1	窗体中的属性	219
上机实验5		132	9.1.2	窗体中的事件	220
第6章 集合和泛型		143	9.1.3	窗体中的方法	221
6.1	集合	143	9.1.4	创建窗体	222
6.1.1	集合的概述	143	9.1.5	消息框	222
6.1.2	ArrayList类	144	9.2	窗体中的基本控件	225
6.1.3	Queue类和Stack类	151	9.2.1	标签和文本框	226
6.1.4	Hashtable类和SortedList类	155	9.2.2	按钮和复选框	227
6.2	泛型	159	9.2.3	列表框和组合框	236

9.2.4	图片控件	240	11.2.1	操作线程的类	307
9.2.5	日期时间控件	242	11.2.2	简单线程	309
9.2.6	菜单栏和工具栏	247	11.2.3	多线程	312
9.2.7	MDI窗体	250	11.2.4	线程同步	314
9.2.8	TreeView控件	251	习题11		319
9.3	Windows窗体中的对话框		上机实验11		319
	控件	257	<b>第12章 ADO.NET 技术</b>		<b>322</b>
9.3.1	字体对话框	257	12.1	ADO.NET概述	322
9.3.2	文件对话框	258	12.1.1	ADO.NET相关概念	322
9.3.3	颜色选择对话框	262	12.1.2	ADO.NET结构	323
9.4	窗体之间的数据交互	263	12.2	ADO.NET五大对象	324
9.4.1	通过属性实现窗体之间的数据交互	264	12.2.1	Connection对象	324
9.4.2	窗体构造函数实现窗体之间的数据交互	266	12.2.2	Command对象	327
习题9		268	12.2.3	DataReader对象	330
上机实验9		269	12.2.4	DataAdapter对象	332
<b>第10章 文件和流</b>		<b>272</b>	12.2.5	DataSet对象	332
10.1	文件操作	272	12.2.6	DataRow类和DataColumn类	333
10.1.1	查看计算机硬盘驱动器信息	272	12.3	数据库访问模式	336
10.1.2	文件夹操作	274	12.3.1	连接模式	336
10.1.3	File类和FileInfo类	276	12.3.2	断开模式	337
10.1.4	Path类	279	12.4	ADO.NET技术操作数据库	340
10.2	流	280	12.4.1	数据的添加	340
10.2.1	文本读写流	280	12.4.2	数据的更新	343
10.2.2	文件读写流	283	12.4.3	数据的删除	345
10.2.3	以二进制形式读写流	287	习题12		348
10.2.4	对象的序列化	291	上机实验12		348
10.3	文件操作控件	294	<b>第13章 数据绑定技术</b>		<b>350</b>
10.3.1	SaveFileDialog	294	13.1	使用组合列表框控件绑定数据	350
10.3.2	OpenFileDialog	295	13.2	数据视图控件绑定数据	352
10.3.3	FolderBrowserDialog	299	13.3	数据视图控件的应用	355
习题10		299	13.3.1	创建课程信息表	355
上机实验10		300	13.3.2	课程管理模块课程信息添加	356
<b>第11章 进程和线程</b>		<b>305</b>	13.3.3	课程管理模块课程信息查询	359
11.1	进程的基本操作	305	13.3.4	课程管理模块课程信息修改	361
11.1.1	Process类	305	13.3.5	课程管理模块课程信息删除	364
11.1.2	进程使用	306	习题13		367
11.2	线程的基本操作	307	上机实验13		368
			参考文献		370





# C#开发环境概述

## 1.1 C#概述

C#(英文名为 C Sharp)是微软公司推出的一种面向对象的、运行于.NET Framework 的编程语言,具有现代的、类型安全的、面向对象语言的基本特征,即封装性、继承性、多态性、抽象及接口,并增加了事件和委托,增强了编程的灵活性。C#凭借通用的语法和便捷的使用方法受到很多企业 and 开发人员的青睐。Visual Studio 是专门的一套基于组件的开发工具,主要用于 .NET 平台的软件开发。

**C#与 C 和 C++的关系:** C#是从 C 和 C++语言演变而来,C#在语句、表达式和运算符方面使用了 C 和 C++的许多功能,但在类型安全性、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,提供了对常用的应用程序编程接口(Application Programming Interface, API)的访问。

**C#与 Java 的关系:** 两者都是面向对象的编程语言,它们都能实现面向对象的思想(封装、继承、多态、抽象、接口)。但两者也有所不同,主要体现在以下方面: C#使用 Visual Studio, Java 使用 MyEclipse; 跨平台 Java 采用虚拟机, C#采用 Framework; 所有平台都支持 Java, 只有 Windows 平台和 Linux 平台支持 C#。

使用 C#语言,程序员可开发 Windows 客户端应用程序、XML Web 服务应用程序、分布式组件应用程序、客户端/服务器应用程序、数据库应用程序等。

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 有以下几个版本: Visual Studio.NET 2010、Visual Studio.NET 2013、Visual Studio.NET 2015 及更高版本。

本书所用开发平台是 Visual Studio.NET 2013 版本。

## 1.2 .NET Framework 概述

.NET 平台是 Microsoft 公司在 2000 年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之上,集 Microsoft 在软件领域的主要技术成就于一身。.NET 框架也是 Windows 7、Windows 8、Windows 10 操作系统的核心部件。

微软对.NET 的定义是:.NET 拥有以新方式融合计算和通信的工具和服务,它是建立于开放互联网协议标准的革命性的新平台。于 2000 年提出的一种新的跨语言、跨平台、面向组件的操作系统

环境，适用于 Web 服务(Web Services)和因特网(Internet)分布式应用程序的生成、部署和运行。.NET 应用是一个运行于 .NET Framework 之上的应用程序。

.NET 平台主要由 .NET Framework(.NET 框架)、基于 .NET 的编程语言、开发工具 Visual Studio 构成。其中，.NET Framework(.NET 框架)是基础和核心。.NET Framework 的体系结构如图 1.1 所示。

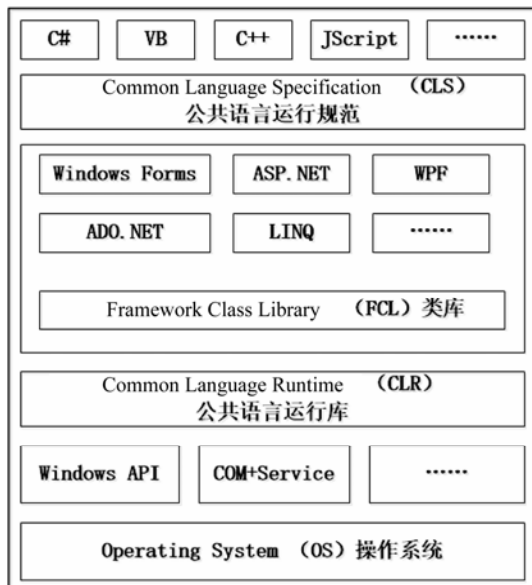


图 1.1 .NET Framework 的体系结构

.NET Framework(.NET 框架)由编程语言(如 C#、VB 等)、公共语言运行规范(CLS)、类库(开发 Windows Forms 应用程序、ASP.NET 应用程序等所用到的类库文件 FCL)、公共语言运行库(用户可以将 CLR 看作一个在执行管理代码的代码，以公共语言运行库为目标的代码称为托管代码，不以公共语言运行库为目标的代码称为非托管代码)、操作系统构成。

类库(Framework Class Library, Fcl)和公共语言运行库(Common Language Runtime, CLR)两部分为 .NET 框架的核心。

### 1) 框架类库

框架类库为开发人员提供统一的、面向对象的、分层的可扩展的类库集，其主要部分是 BCL(Base Class Library, 基类库)。通过创建跨所有编程语言的公共 API 集，公共语言运行库使得跨语言继承、错误处理和调试成为可能。框架类库采用命名空间来组织和使用。

### 2) 公共语言运行库

公共语言运行库负责内存分配和垃圾回收，保证应用和底层系统的分离。公共语言运行库所负责的应用程序在运行时是托管的，具有跨语言调用、内存管理、安全性处理等优点。

## 1.3 Visual Studio 2013 的安装与卸载

### 1.3.1 安装的必备条件

在 Windows 7 及以下的操作系统中要求必须具备管理员权限才能安装 .NET Framework 框架。此

外, 在 Windows 7 系统上安装 .NET Framework 需要操作系统有 SP1 补丁。

目前, .NET Framework 仅支持在 Windows 操作系统上安装, 其最高版本是 .NET Framework 4.7.2, 本书使用的是 .NET Framework 4.5.1, 它支持的操作系统有 Windows 7、Windows 8、Windows 10 的 32 位或 64 位。

### 1.3.2 安装与卸载的步骤

用户在安装 Visual Studio 2013 之前, 可以在微软的官网上了解 Visual Studio 2013 中各版本的具体功能和特点、安装的操作系统要求。

#### 1. 启动安装程序

从微软的官网下载 Visual Studio 2013 的安装程序, 下载后解压, 出现如图 1.2 所示的文件夹。

在图 1.2 中单击 vs\_ultimate 文件进入安装界面, 安装界面如图 1.3 所示。

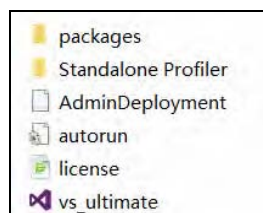


图 1.2 Visual Studio 2013 安装解压后的效果

#### 2. 选择安装位置并安装

在图 1.3 中, 单击“我同意许可条款和隐私策略”前的复选框并单击[下一步]按钮, 进入图 1.4 所示的安装位置选择界面。在图 1.4 所示界面中允许选择程序安装的位置(建议最好不安装到操作系统所在的驱动器下, 以免占用过多资源, 影响开机速度), 可以选择“典型”或“自定义”, 典型安装是将 Visual Studio 2013 的主要功能全部安装, 自定义安装是根据用户的需要有选择性地安装。选择后, 单击[安装]按钮, 进入 Visual Studio 2013 的安装界面。



图 1.3 Visual Studio 2013 安装界面



图 1.4 安装位置选择界面

### 3. 安装过程

Visual Studio 2013 的安装过程，如图 1.5 所示。

在 Visual Studio 2013 中可使用微软账号登录，并且在不同的设备上共享程序，也可不登录直接使用 Visual Studio 2013 工具。

卸载 Visual Studio 2013 和卸载 Windows 中的其他应用程序一样，直接在控制面板中选择[添加或删除程序]，选择 Visual Studio 2013 程序卸载即可。



图 1.5 Visual Studio 2013 安装过程

## 1.4 Visual Studio 2013 的开发环境

Visual Studio 2013 是一款便于学习和使用的开发工具，并提供了大量的帮助文档供用户参考。

### 1.4.1 Visual Studio 2013 中常用的菜单

启动 Visual Studio 2013 后的主界面如图 1.6 所示。

在 Visual Studio 2013 的主界面中看到的是起始页，用于显示最近打开的项目，可以进行新建项目、打开项目等操作。在该页面中还能了解 Visual Studio 2013 的一些新功能，提供了菜单和工具栏，下面介绍菜单栏中常用的功能。

- (1) 文件：主要用于新建项目、打开现有项目及保存项目等操作。
- (2) 编辑：主要用于文件内容的复制、剪切、粘贴、保存等操作。

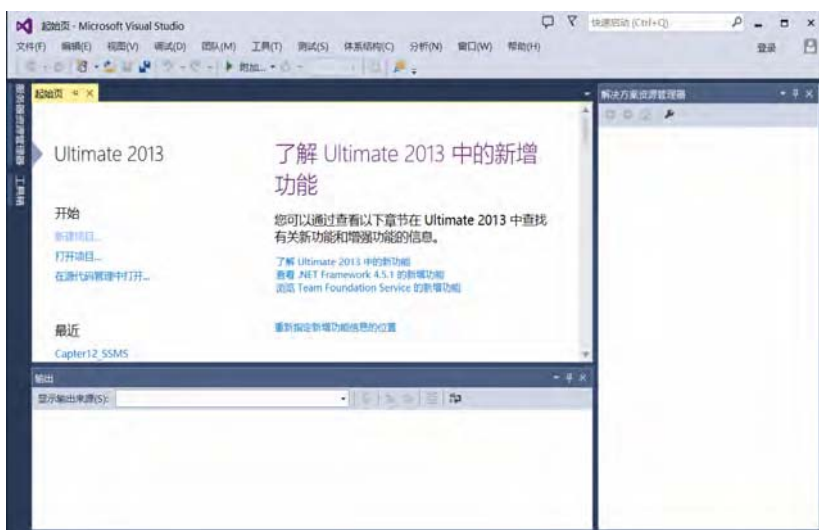


图 1.6 Visual Studio 2013 的主界面

(3) 视图: 主要用于显示 Visual Studio 2013 界面中的不同窗口, [视图]菜单项如图 1.7 所示。常用窗口主要有: 解决方案资源管理器、服务器资源管理器、SQL Server 对象资源管理器、错误列表窗口、输出窗口、工具箱窗口、属性窗口等。

- 解决方案资源管理器: 用于管理 Visual Studio 2013 中创建的项目。
- 服务器资源管理器: 用于管理数据库连接、移动服务、应用服务等。
- SQL Server 对象资源管理器: 用于管理 Visual Studio 2013 中自带的或其他的 SQL Server 数据库。
- 错误列表窗口: 用于显示程序在编译或运行后出现的错误信息。
- 输出窗口: 用于显示在程序中的输出信息。
- 工具箱窗口: 用于显示在 Windows 窗体应用程序或 WPF 应用程序、网络应用程序中可以使用控件。
- 属性窗口: 用于设置项目或程序中使用的的所有控件等内容的属性。

(4) 调试: 用于在程序运行时调试使用。

(5) 团队: 用于团队开发项目时使用。

(6) 工具: 用于连接到数据库、连接到服务器、选择工具箱中的工具等操作。

解决方案资源管理器(P)	Ctrl+Alt+L
团队资源管理器(M)	Ctrl+V, Ctrl+M
服务器资源管理器(V)	Ctrl+Alt+S
体系结构资源管理器(A)	Ctrl+V, Ctrl+R
SQL Server 对象资源管理器	Ctrl+V, Ctrl+S
书签窗口(B)	Ctrl+K, Ctrl+W
调用层次结构(H)	Ctrl+Alt+K
类视图(A)	Ctrl+Shift+C
代码定义窗口(D)	Ctrl+V, D
对象浏览器(J)	Ctrl+Alt+J
错误列表(I)	Ctrl+V, E
输出(O)	Ctrl+Alt+O
起始页(G)	
任务列表(K)	Ctrl+V, T
工具箱(X)	Ctrl+Alt+X
通知(N)	Ctrl+W, N
查找结果(N)	▶
其他窗口(E)	▶
工具栏(T)	▶
全屏显示(U)	Shift+Alt+Enter
所有窗口(I)	Shift+Alt+M
向后导航(B)	Ctrl+-
向前导航(F)	Ctrl+Shift+-
下一任务(X)	
上一任务(R)	
属性窗口(W)	F4
属性页(Y)	Shift+F4

图 1.7 [视图]菜单中的菜单项

- (7) 测试：用于对程序进行测试。
- (8) 体系结构：用于创建 UML 模型或关系图。
- (9) 分析：用于分析程序的性能。
- (10) 窗口：用于设置 Visual Studio 2013 界面中显示的窗口，并提供重置窗口的选项，方便用户重置 Visual Studio 2013 的操作界面。

## 1.4.2 C#程序入门

使用 C#语言，可以创建多种应用程序，包括控制台应用程序、Windows 窗体应用程序、WPF 应用程序、Web 应用程序和 Silverlight 应用程序。在 Visual Studio 2013 中，创建这些程序的模式基本上相同。

操作步骤：

- (1) 启动 Visual Studio 2013。
- (2) 新建项目。

执行[文件]->[新建]->[项目]菜单命令，弹出[新建项目]对话框，在左侧列表框中选择[已安装]->[模板]->[Visual C#]->[Windows]，同时在中间列表框中选择要开发的程序，例如，如果开发 Windows 窗体应用程序，则选择 Windows 窗体应用程序。

在[名称]文本框中，输入项目的名称(如 Project1)；在[位置]框中输入保存项目的文件夹(选择计算机保存的磁盘及路径)；在[解决方案名称]框中输入项目所在的解决方案的名称(如 Capter1)，单击[确定]按钮。

系统自动完成项目的配置，作为一个 Windows 应用程序，必不可少的配置包括对 .NET Framework 类库的引用及应用程序项目的属性设置等。

**【实例 1.1】** 设计一个 C#控制台程序，在控制台中输出信息：“欢迎来到 C#编程世界！”。

**【操作步骤】** 实现操作步骤如下。

- (1) 启动 Visual Studio 2013。
- (2) 新建项目。

在启动的 Visual Studio 2013 界面中，执行[文件]->[新建]->[项目]菜单命令，弹出[新建项目]对话框，在[已安装的模板]列表框中选择[Visual C#]选项，同时在对话框中间的模板列表框中选择[控制台应用程序]选项。

在[名称]文本框中，输入项目的名称(如 Project1)；在[位置]组合框中，输入保存项目的磁盘位置；在[解决方案名称]框中输入项目所在的解决方案的名称(如 Capter1)，单击[确定]按钮。

系统自动完成项目的配置，其中必不可少的配置是对 .NET Framework 类库的引用，控制台应用程序和源代码文件是 Program.cs。新建项目效果如图 1.8 所示。

从图 1.8 中可以看到，项目创建好后自动生成一段程序代码，其中 Main 所在行的代码表示一个方法，在该方法中编写程序代码，并且 Main()方法是程序的主入口，程序执行时会从 Main()方法开始执行。

图 1.8 中，第 1~5 行是 C#程序引用的命名空间。命名空间是一种组织 C#程序中出现的不同类型的方式，可使类具有唯一的完全限定名称。一个 C#程序至少包含一个或多个命名空间，命名空间可由程序员定义，或者用先前编程写好的类库的一部分定义，或者由系统生成项目自动生成。第 6 行是声明的命名空间，也就是项目的名称。第 9 行是声明的类。第 11 行是程序的入口或静态的 Main()



方法，其返回值是 void。

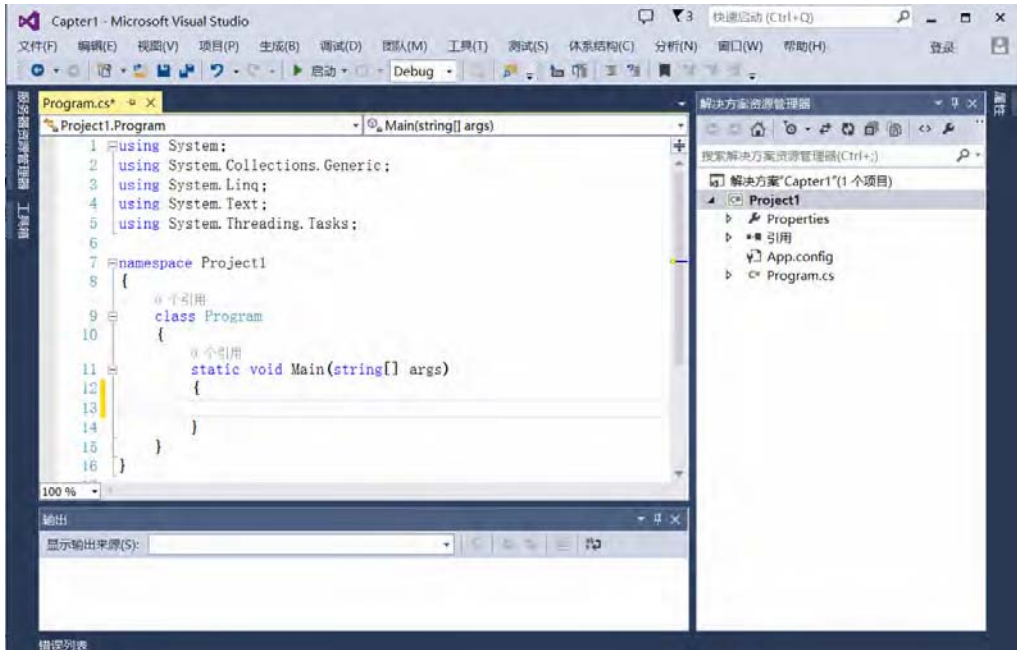


图 1.8 创建解决方案下的新项目效果

程序实现功能语句在 Main()方法的一对大括号内进行书写。该语句是调用系统中的 Console 类的 WriteLine()方法实现信息输出。

### (3) 编写 C#语言源程序代码。

在编写 C#程序时，程序中出现的空格、括号、逗号、分号等符号必须采用英文半角格式，否则程序会出现编译错误。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks; //系统自动导入的命名空间

namespace Project1 //用户自定义命名空间
{
    class Program //定义的类名称
    {
        static void Main(string[] args) //静态主方法
        {
            Console.WriteLine("欢迎来到 C#编程世界!"); //用户编写的代码
            Console.ReadKey();
        }
    }
}
```

#### (4) 编译并运行程序。

运行程序的方法：选择[调试]->[启动调试]菜单命令(也可以按功能键 F5)，或者选择[调试]->[开始执行(不调试)]菜单命令(也可以按功能组合键 Ctrl+F5)，或者单击工具栏中的[启动]按钮。

运行程序的原理：使用 C#语言进行程序开发时，必须了解 C#语言程序的运行原理。程序运行机制分两步完成：①编译期，CLR 对 C#代码进行第一次编译，将编写的代码编译成.dll 文件或.exe 文件，此时的代码被编译为中间语言；②运行期，CLR 针对目前特定的硬件环境使用即时编译(JIT)，将中间语言编译成为本机代码并执行，把编译后的代码放入一个缓冲区中，下次使用相同的代码时，就直接从缓冲区中调用，也就是说相同的代码只编译一次，从而提高了程序运行的速度。C#程序在.NET Framework 中编译和运行的过程如图 1.9 所示。

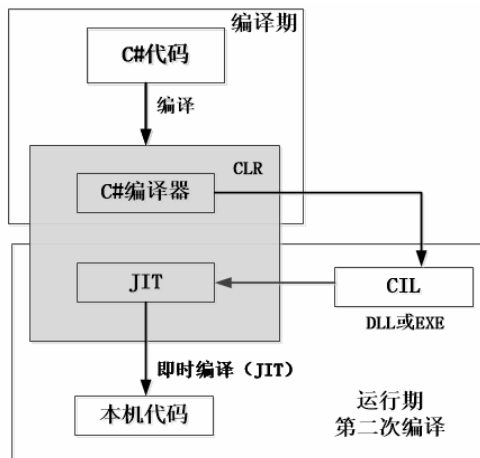


图 1.9 C#程序在.NET Framework 中编译和运行的过程

单击工具栏中的[启动]按钮后，Visual Studio 2013 将自动启动 C#语言编译器编译源程序并执行程序，最后将程序的运行结果显示在命令提示符窗口中。

程序运行的结果如图 1.10 所示。

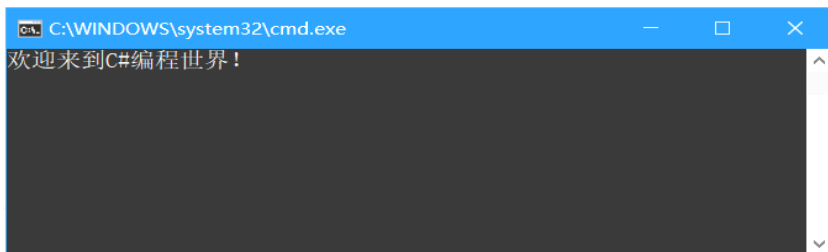


图 1.10 实例 1.1 程序运行结果

#### 【结果分析】

对于一个 C#控制台程序来说，虽然功能单一，但包含了 C#程序的基本结构，主要结构如下。

(1) 引用命名空间：告诉编译器程序集引用的命名空间。

(2) 自定义命名空间：C#使用命名空间来控制源程序代码的范围，以加强源程序代码的组织管理。采用 namespace 关键字来声明，Visual Studio 2013 在创建应用程序项目时，自动使用项目名称来设置命名空间的名字，用户编写的代码就放在自定义的命名空间的一对大花括号{ }中。

(3) 定义类：C#是面向对象的编程语言，C#语句必须封装在类中，一个程序至少包括一个自定义类。类的定义采用 class 关键字声明，类中成员语句放在类的一对大花括号{ }中。

(4) 定义方法：C#控制台程序必须有一个 Main()方法，程序运行时，首先从 Main()方法开始执行，当最后一条语句被执行之后，程序结束运行。方法有规范的格式要求，在 Main()方法中有一个可带若干个字符串参数的数组 args。

(5) 编写语句：一个 C#程序由若干条语句组成，每个语句必须符合 C#语法规则，每个语句的结



束标志为英文字符“;”。

(6) 添加代码注释：C#语言的注释有单行注释//、块注释/\* \*/、参数化注释///。

**【实例 1.2】**设计一个 C#的 Windows 程序，要求实现功能：在姓名文本框中输入姓名“王先水”字符串，单击[显示]按钮，在 Windows 的指定位置显示信息“王先水：你好！欢迎学习 C#语言程序设计！”。

#### 【操作步骤】

(1) 启动 Visual Studio 2013。

(2) 新建项目。



在启动的 Visual Studio 2013 界面中，执行[文件]->[新建]->[项目]菜单命令，弹出[新建项目]对话框，在[已安装的模板]列表框中选择[Visual C#]选项，同时在对话框中间的模板列表框中选择[Windows 应用程序]选项。

在[名称]文本框中，输入项目的名称(如 Project2)；在[位置]组合框中，输入保存项目的位置；在[解决方案名称]框中输入项目所在的解决方案的名称(如 Capter1)，单击[确定]按钮。

单击[确定]按钮后，系统自动完成项目的配置，包括对 .NET Framework 类库的引用，生成包括 Main()方法的 Program.cs 文件，生成 Windows 窗体文件 Form1.cs。

(3) 窗体界面设计。

在窗体界面上需要设计 2 个 Label 标签对象，一个表示静态提示信息标签，一个表示动态显示信息标签；1 个输入信息的 TextBox 文本框对象；1 个 Button 按钮对象。各控件对象的属性、属性值如表 1.1 所示。

表 1.1 各控件对象的属性、属性值

控件对象	属性	属性值	控件对象	属性	属性值
Label2	Name	lblShow	Label1	Text	姓名
	AutoSize	False	TextBox1	Name	txtName
	BorderStyle	Fixed3D	Button1	Name	btnShow
	Text	Null		Text	显示

根据表 1.1 设计的窗体界面如图 1.11 所示。



图 1.11 设计的窗体界面

#### 注意：

各控件对象中的 Name 属性的属性值是程序后台要引用的变量名。

(4) 后台代码设计。

C#的 Windows 应用程序开发是基于事件驱动编程，在该程序开发中是[显示]按钮单击事件，因

此编写[显示]按钮单击事件程序，设计代码如下。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Project2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnShow_Click(object sender, EventArgs e)
        {
            lblShow.Text = txtName.Text + "： 你好！ 欢迎学习 C#语言程序设计！ ";
        }
    }
}
```

(5) 编译并运行程序。

运行程序方法：选择[调试]->[启动调试]菜单命令(也可以按功能键 F5)，或者选择[调试]->[开始执行(不调试)]菜单命令(也可以按功能组合键 Ctrl+F5)，或者单击工具栏中的[启动]按钮。

程序运行的结果如图 1.12 所示。

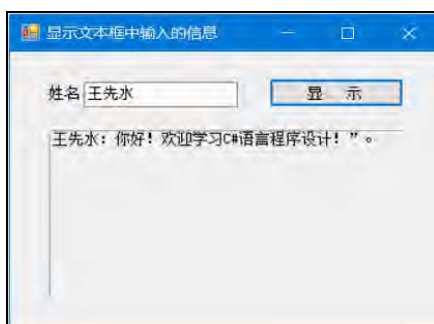


图 1.12 实例 1.2 程序运行结果

### 【结果分析】

在自命名空间 Project2 的 From1 类的 btnShow\_Click()事件方法中编写程序代码。将文本框中输入的字符串“王先水”与字符串“： 你好！ 欢迎学习 C#语言程序设计！”通过连接运算符“+”连接的结果在窗体的 lblShow 标签外显示。

C# Windows 程序的 From1 类继承了基类 From 类基本属性, 如窗体的最大化、最小化、还原属性。在 From1 类中, 有一个 From1 类的构造函数(在后续章节中介绍), 在构造函数中调用初始化窗体的方法 InitializeComponent()进行窗体界面设计, 有一个显示按钮的单击事件方法 btnShow\_Click()实现程序功能代码设计。

## 习题 1

### 1. 填空题

(1) C#是一种\_\_\_\_\_类型的编程语言。

(2) .NET Framework 中的两个主要组件分别是\_\_\_\_\_和\_\_\_\_\_。

2. 编写一个 Windows 程序, 在文本框中分别输入自己的“学号”“姓名”“专业”, 单击[显示]按钮在窗体的指定标签处分行显示相关的信息。