秦光源　徐瑗蔚　张　剑　著

C# 编程技术的研究及案例分析

C# BIANCHENG JISHUDE YANJIU
JI ANLI FENXI

原子能出版社
Atomic Energy Press

C# 编程技术的研究及案例分析

秦光源　徐瑗蔚　张　剑　著

中国原子能出版社
China Atomic Energy Press

图书在版编目（CIP）数据

C# 编程技术的研究及案例分析 / 秦光源，徐瑗蔚，张剑著. -- 北京 : 中国原子能出版社，2018.8
ISBN 978-7-5022-9353-6

Ⅰ. ①C… Ⅱ. ①秦… ②徐… ③张… Ⅲ. ①C 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 204623 号

内容简介

C# 是微软公司发布的一种面向对象的、运行于 .NET 平台上的高级程序设计语言，一种安全的、稳定的、简单的、优雅的，由 C 和 C++ 衍生出来的面向对象的编程语言，它在继承 C 和 C++ 强大功能的同时去掉了一些它们的复杂特性。《C# 编程技术的研究及案例分析》首先对 C# 的基础理论进行论述，将文字与代码结合，全面的阐述 C# 语言的各种特性。然后结合各种应用实例进行分析，从而使得读者轻松了解并学习 C# 编程程序的开发精髓，可以更好的领会 C# 的魅力。

C# 编程技术的研究及案例分析

出版发行	中国原子能出版社（北京市海淀区阜成路 43 号　100048）
责任编辑	王　丹　高树超
装帧设计	河北优盛文化传播有限公司
责任校对	冯莲凤
责任印制	潘玉玲
印　　刷	定州启航印刷有限公司
开　　本	787 mm×1092 mm　1/16
印　　张	14.75
字　　数	319 千字
版　　次	2019 年 1 月第 1 版　　2019 年 1 月第 1 次印刷
书　　号	ISBN 978-7-5022-9353-6
定　　价	56.00 元

发行电话：010-68452845

前言

我国的软件产业保持着高速增长趋势，该产业已成为我国重要的产业之一。在互联网背景下，优秀的IT人才需求量大，供不应求，培养合格、优秀的IT人才是当下研究的问题和重点。

计算机编程语言从诞生开始就在不断更新、不断发展、不断成熟化。.NET框架是微软在2000年专业开发者会议上提出的开发平台，这是一个革命性的应用程序开发平台，该平台建立在开放的Internet协议和标准之上，采用了许多新的工具用于计算和服务。创建XML Web服务（下一代软件）平台，将信息、设备和人以一种统一的、个性化的方式联系起来。在该平台中，C#作为微软面向下一代应用平台的核心语言，能够让开发人员在.NET平台上快速开发应用程序。可以说，C#是面向.NET框架的、可以兼顾开发能力和效率的、与当前的Web应用结合较好的、全新的开发工具和首选开发语言。C#为程序员提供了开发Web应用程序所需要的强大而灵活的功能，未来大量.NET平台的应用将由C#开发，C#将是未来开发企业级分布式应用程序的首选。

鉴于.NET框架以及核心编程语言C#在微软的大力推广下将成为未来影响人类生活的最先进技术平台之一，并且是今后IT人才培养的基础，因此，学习相关技术就显得十分必要和迫切。微软.NET平台包括工具、服务器、XML Web服务、客户端，其新的理念涉及计算机技术的方方面面，很多内容都可以作为一个专题进行研究。

作者具有十多年的项目开发经验和程序设计教学经验，经过大量的相关技术分析和参考众多国内外相关书籍编写本书，主要研究C#语言及其开发环境、.NET框架、ADO. NET和ASP. NET等内容。希望通过本书，帮助初学者快速掌握C#语言，提升开发能力。

目录

第一章　C# 概述

C# 是微软公司发布的一种面向对象的、运行于 .NET Framework 之上的高级程序设计语言。C# 具有一种安全、稳定、简单的特征，是由 C 和 C++ 衍生出来的面向对象的编程语言。C# 与 Java 相似，具有与 Java 几乎同样的语法和编译成中间代码再运行的过程。本章主要介绍 C# 的由来、特点以及运行环境，为后面 C# 编程做铺垫。

第一节　C# 的发展历史与特点

C# 语言诞生于 20 世纪末期，它也是一种高级编程语言。下面介绍 C# 语言的来源、特点，C# 与 Java、C++ 有什么关系。

一、C# 的产生

当微软以 MS-DOS 主宰着全球个人计算机操作系统市场的时候，微软在软件技术上的成就，并不是很突出。微软操作系统的能力，以 MS-DOS 的 16 位单机操作系统为例，尚比不上当时 IBM 的 OS/2 或是尚未登上操作系统主流舞台的 Linux。要论开发环境，微软除 Quick Basic 之外，大部分的开发环境仍是 Borland 的天下，Turbo C++ 仍是当时的主流开发平台。

这个态势在微软推出 Windows 3.0 视窗操作系统时更为明显。虽然是自家推出的视窗环境操作系统，但具有讽刺意味的是，微软并没有为这个操作系统提供任何完全视窗环境下的集成开发环境（IDE）。倒是开发软件供应商的老大哥——Borland 率先提供了 Borland C++ 这个完全在视窗环境下，就可以开发软件的开发工具。

1995—1996 年，Internet 以出人意料的速度发展，同时造就了许多新兴的事业与公司。其中给人印象最深的要数 Netscape。Netscape 以开发 www 浏览器——Netscape Navigator 起家，在两年的时间内迅速崛起，成为 Internet 上最知名的软件公司。在同一时期，Sun 也以 Java 拓展了自己的另一块事业版图。在 Sun 与 Netscape 的相互应合之下，微软这个软件帝国的霸主地位，正逐渐地受到怀疑和挑战。

1996 年，是微软反败为胜最为关键的一年。这个时候，微软终于开始正视 Internet 的威力了。在网络用户端方面，微软以免费的方式发行自家的浏览器（其实是并购来的）Internet Explorer 以打击 Netscape。在网络服务器方面，微软在 Windows NT 4.0 中，免费附加了 IIS（Internet Information Server）2.0，内附有 WWW、FTP 以及 Gopher 等服务器，同样是希望能够以免费的手段来取得市场占有率。但最重要的技术革新，还是在 ActiveX 之前微软在 Internet 应用上所做的努力，已获得若干成效。Internet Explorer 的占有率，也终于能够与 Netscape 并驾齐驱，并且有取而代之的气势。此时，微软欠缺的就是一个全面性的网络整合平台。

1999 年初，微软推出了在 Windows NT 平台上的 Windows NT 4.0 Option Pack，其中包含了 IIS 4.0、MTS（Microsoft Transaction Server）、ASP（Active Server Pages）、信息队列（Microsoft Message Queue，MSMQ）等功能。同时，微软的第一个完全整合 Windows 与 Internet 的平台——Windows DNA 也诞生了。

在 Windows DNA 之后，微软便开始将重心转到一个新的计划——新一代的 Windows 系统（Next Generation Windows System，NGWS）。NGWS 的目的是整合新的设备和技术，希望带给用户一个全新的使用体验。NGWS 包括：Windows DNA 的新一代——Windows DNA 2000、ASP 的新一代——ASP.NET（之前名为 ASP+）、Visual Studio 6.0 的新一代——Visual Studio.NET，以及将来的 Windows.NET。这个 NGWS 计划，就是现在所看到的 .NET。C# 即为其核心开发的计算机语言。

计算机编程语言（如 C++ 和 Java）和消费类电子设备（如移动电话）的进步，带来了新的问题和需求。即对各种语言编写的组件进行集成是一件很困难的事，而且由于新版本的共享组件和旧的软件不兼容，安装也常常出现问题。同时，开发人员还发现他们需要基于 Web 的应用程序，以便可以通过 Internet 进行访问和使用。移动电子设备的普及使得软件开发人员意识到客户不再局限于桌面计算机系统。开发人员意识到一种软件需求：让任何人从任何地方，在任何时间使用任何设备访问 Internet 上的服务。基于这些要求，微软发布了它的 C#（读作 C Sharp）和 C# 所在的集成开发环境 Visual Studio.NET（读作 dot-net，简记为 .NET）。

二、C# 的定义

（一）C# 的定义

C# 编程语言是由微软的 Anders Hejlsberg 和 Scott Wiltamuth 领导的一个小组开发的，是为 .NET 平台专门设计的语言，以便让程序员更容易转移到 .NET 平台上来。因为 C# 是在 C、C++ 和 Java 的基础上，吸取了每种语言的优点并增加了自己的特点，因此这种转移是易于实现的。

C# 是一种现代的、类型安全的、完全面向对象和可视化的编程语言。它的目标是将 Visual Basic 的高产和 C++ 底层高效的特性结合起来。它可以使程序员快速而容易地为微软 .NET 平台开发解决方案。

（二）C# 与 Java 及 C++ 的对比

作为一种高级语言，C# 语言与其他高级语言，特别是与 Java 语言比较，又有什么异同点呢？下面对 C# 和 Java 以及 C++ 进行比较。

C# 和 Java 的相同点是非常多的，软件技术的发展同样也是一个继承和发展的过程，因此，C# 和 java 有很多相似或相同的内容是很正常的。其相同点包括：可编译为机器独立、语言独立的代码，运行在托管运行环境中；采用垃圾收集机制，同时摒弃了指针；具有强有力的反射能力；没有头文件，所有代码都包装在程序集里，不存在类声明的循环依赖问题；支持多继承接口、单继承实现；所有的类都派生自 object，且必须用 new 关键字分配于堆上等。

既然 C# 和 Java 之间有这么多的相同之处，那么，C# 是否就是微软推出的 Java 的克隆产品？是否就是 C 与其他高级语言的简单组合呢？其实这个理解是错误的。

首先，C# 不是 Java 的克隆。在实现方式上，C# 与 Java 之间存在着一定的差异。Java 可以在任何有 Java 虚拟机器的平台上执行。C# 目前只能在 Windows 上执行。C# 先将代码编译为中间语言（Intermediate Language，IL），然后通过 just-in-time（JIT）的编译方式或原生码编译方式来执行；Java 可以在任何有 Java 虚拟机器（Java Virtual Machine，JVM）的平台上执行，也可以编译成原生码。而在 .NET 框架下，所有的语言都被编译为相同的 IL 代码，运行时由公共（通用）语言运行时（Common Language Runtime，CLR）负责管理，使用相同的组件，真正做到了多语言的集成。

其次，C# 也不是 C 与 Java 或其他高级语言的简单组合。在设计 C# 期间，考察了多种语言，如 C++、Java、Modula2、C、Smalltalk 等。很多语言都包含有与 C# 开发环境相同的核心思想，如深度面向对象、简化对象等。C# 和这些高级语言，尤其是 Java 的关键不同点是，它非常接近 C++。C# 从 C++ 直接借用了大多数的操作符、关键字和声明。同时，C# 还保留了许多被 Java 抛弃的语言特性，如枚举等特性。在 C++ 中，枚举显然是一个很有意义的概念，在 C# 中，保留了枚举并同样使其类型安全。C# 还保留了操作符重载和类型转换。C# 名字空间的整体结构也非常接近 C++。

另外，设计 C# 语言的一个关键的目标是使其面向组件。C# 加入了在编写组件时所需要的所有概念，如属性（Property）、方法、事件、特性（Attribute）和文档等。设计的特性（Attributes）是一种崭新的声明性信息，是全新的和创新的方法。它使得软件开发者不仅可以通过特性来定义设计层面的信息以及运行时信息，而且还可以利用特性建立自描述（Self describing）组件，可为任何对象加入有类型的、可扩展的元数据。这是目前其他程序语言所不具备的。C# 也是第一个合并 XML 注释标记的语言，编译器可以用其直接从源码中生成可读的文档。因此，可以说 C# 集成了许多原有技术的优点并加以创新和提高。

总之，C++ 高效但是不安全，Java（跨平台）安全但是较低效，C# 安全且较高效。表 1-1 列出了这三种面向对象语言的功能和特点的部分比较。

表 1-1　C# 与 C++ 和 Java 的比较

功　能	C++	Java	C#
跨平台	源代码（部分）	字节码	通用语言结构 CLI
执行方式	编译	编译 + 解释	编译 +JIT 转换
中间代码	无	字节码 Bytecode	微软中间语言 MSIL
运行环境	操作系统	JVM	CLR
内存管理	直接分配和删除	垃圾内存自动回收	垃圾内存自动回收
多重类继承	支持	不支持	不支持
操作符重载	支持	不支持	部分支持
对象访问	地址 / 指针	引用	引用
接口类型	无	有	有
属性成员	无	无	有
命名空间	支持	包机制	支持
指针	支持	不支持	部分非安全代码
函数指针	支持	适配器 + 监听程序	委托
全局函数与变量	有	无	无
无符号整数类型	有	无	有
强制类型转换	支持	不支持	支持
越界自动检查	无	有	有
多维数组	数组的数组	数组的数组	真正多维数组
索引	支持	不支持	支持
线程同步	调用函数	语言内部	语言内部
异常处理	可选	支持检查异常	只支持非检查异常
标准类库	贫乏	丰富	庞大
适用领域	面向对象的系统和界面编程	跨平台（服务器端）网络编程	基于 Windows 平台 .NET 和组件编程

表 1-1 中，公共语言基础结构（Common Language Infrastructure，CLI）是 CLR 的一个子集，为 IL 代码提供运行的环境。.NET 环境下的任何语言编写的代码，通过其特定的编译器转换为 MSIL 代码之后均可运行其上。

三、C# 的特点

由于 C# 几乎集中了所有关于软件开发和软件工程研究的最新成果，即面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理等，因此，C# 应该是目前最好的计算机编程语言之一。

C# 语言的特点可以归纳为以下五个方面。

（一）简单性

C# 语言的简单性具体体现在：

（1）C# 限制了指针操作，虽仍能够使用，但在默认情况下，不允许直接对内存进行操作。在 C++ 中所使用的操作符，如“：：”和“->”都不能在 C# 中使用，在 C# 中支持的相应操作符为“.”。如果一定要对内存地址进行操作，可在不安全模式下进行。

（2）因为 C# 是基于 .NET 平台的，因此，它继承了自动内存管理和垃圾回收的特点。

（3）在 C# 中，整形数值 0 和 1 不再作为布尔值出现。C# 中的布尔值是纯粹的 true 值和 false 值。“= =”被用于进行比较操作，而“=”被用作赋值操作。

（二）现代性

C# 语言是微软为了其推出的 .NET 平台而开发的一门高级语言，是目前最新的计算机语言。具体体现在：

（1）C# 支持组件编程，对于创建相互兼容的、可伸缩的、健壮的应用程序来说，是非常强大和简单的。

（2）C# 拥有内建的支持将任何组件转换成一个 Web Service 的功能，运行在任何平台上的任何应用程序，都可以通过互联网来使用这个服务。

（3）C# 是第一个合并 XML 的语言，编译器可以用其直接从源代码中生成可读的文档。

（三）面向对象性

C# 语言是一种真正的面向对象语言，具体体现在：

（1）与 C++ 相比，C# 没有全局变量和全局函数等，所有的代码都必须封装在类中（甚至包括入口函数 Main 方法），不能重写非虚拟的方法，增加了访问修饰符 internal，不支持多重类继承（似 Java，用多重接口实现来代替）。

（2）C# 中提供了装箱和拆箱机制。这样，在 C# 的类型系统中，每一个类型都可以看作一个对象。

（3）C# 只允许单继承，即一个类只能有一个基类，从而避免了类型定义的混乱。

（四）类型安全性

C# 语言具有很强的保护功能，确保使用的安全性，具体体现在：

（1）取消了不安全的类型转换，如在 C# 中不能将 double 转换成 boolean。

（2）不允许使用未初始化的变量。值类型（常量类型）被初始化为零值而引用类型初始化时默认为 null。

（3）数组类型下标从零开始，进行越界检查，不让数组越界。

（4）检查类型溢出。

（5）用委托取代函数指针，增强了类型安全。

（五）相互兼容性

C# 语言有很强的兼容性，在其集成开发环境中，各种语言可以交叉使用，体现在：

（1）C# 提供对 COM 和基于 Windows 的应用程序的支持。

（2）允许对原始指针的有限制地使用，C# 允许用户将指针作为不安全的代码段来操作老的代码。

（3）用户不再需要显示地实现 unknown 和其他 COM 界面，这些功能已经建成。

（4）VB.NET 和其他中间代码语言中的组件可以在 C# 中直接使用。

第二节　C# 语言开发环境的构建

C# 是 Visual Studio.NET 集成开发环境中的代表性语言，本书首先介绍 C# 所在的集成开发环境，然后介绍 C# 的程序结构，以及如何在 Visual Studio.NET 集成开发环境下编译调试 C# 程序。

一、Visual Studio.NET 集成开发环境简介

Visual Studio.NET 是一套完整的开发工具集，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，并提供了在设计、开发、调试和部署 Web 应用程序、XML Web Services 和传统的客户端应用程序时所需的工具。Visual Basic、Visual C++、Visual C# 和 Visual J# 都使用这个相同的集成开发环境（IDE），利用此 IDE 可以共享工具且有助于创建混合语言解决方案。另外，这些语言利用了 .NET Framework 的功能，通过此框架可使用简化 ASP Web 应用程序和 XML Web Services 开发的关键技术。

（一）Visual Studio 2015 的安装环境要求

安装 VisualStudio 2015 版本的计算机需满足下列主要系统要求（目前一般计算机均可满足该版本以及更高版本对硬件的要求）：

（1）处理器：1 GHz 以上。

（2）内存：1 G 以上。

（3）可用硬盘空间：不含 MSDN 时，系统驱动器上需要 1 GB 的可用空间，安装驱动器上需要 2 GB 的可用空间；包含 MSDN 时，系统驱动器上需要 1 GB 的可用空间，完整安装 MSDN 的安装驱动器上需要 3.8 GB 的可用空间，默认安装 MSDN 的安装驱动器上需要 2.8 GB 的可用空间。

（4）操作系统：Windows 7/8/10 版本。

从 https://msdn.microsoft.com 网址下载 Visual Studio 2015 安装文件，按照操作步骤安装并配置文件即可。一旦安装了 IIS，IIS 便会自动运行，而且会在每次开机之后自动运行。完成 IIS 的安装后，系统会自动建立两个文件夹。

（二）Visual Studio.NET 集成开发环境

Visual Studio.NET 是一套完整的、功能强大的集成开发工具，用于生成 ASP Web 应用程序、XML Web services、桌面应用程序和移动应用程序。Visual Studio.NET 为 Visual Basic.NET、Visual C++.NET、Visual C#.NET 和 Visual J#.NET 提供了相同的集成开发环境（IDE）。该环境允许它们共享工具，并且有助于创建混合语言解决方案。另外，这些语言利用了 .NET Framework 的功能。使用该框架，有助于简化 ASP.NET Web 应用程序的开发，以及 XML Web services 的构建。图 1-1 所示为 Visual Studio.NET 集成开发环境。这是设置成默认的 Visual C# 开发设置，并打开一个 Visual C# 项目时的情况。

集成开发环境是一个标准的 Windows 应用程序界面，由主窗口、解决方案资源管理器窗口、属性窗口、服务器资源管理器窗口、工具箱、错误列表窗口等各种工具窗口组成。

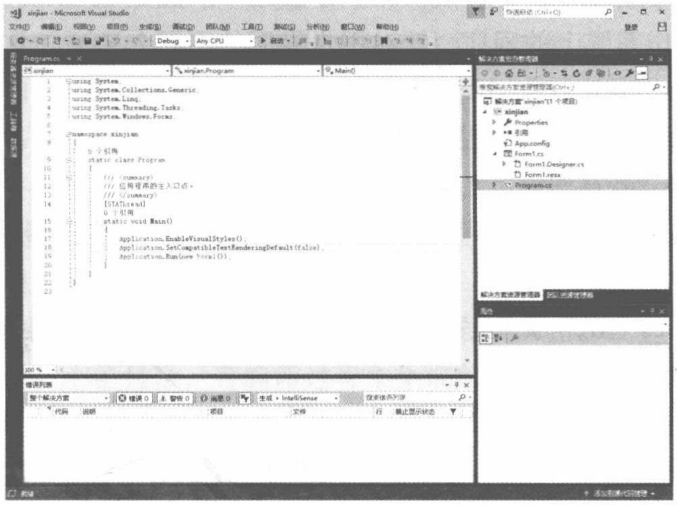

图 1-1 Visual Studio.NET 集成开发环境

（1）主窗口：位于整个用户界面的中央，其作用主要是用户代码编写区和图形用户界面的显示区。该窗口是 Visual Studio 最主要的部分，称之为“操作区”。

（2）解决方案资源管理器窗口：在默认布局设置的情况下，在集成开发环境的右边上侧，该窗口主要用来对各种解决方案进行管理。当有项目打开后，项目文件就会以树状在这里显示，解决方案资源管理器窗口，主要用于显示解决方案、解决方案的项目及这些项目中的项。可以在其中对项目文件进行添加、打开、复制、删除等一系列操作。只需要在各项上右击，在弹出的右键快捷菜单上，进行相应的选择即可。该窗口的上方，提供了相应的操作按钮。将鼠标移动至每个按钮上，都会有相应的提示。

（3）属性窗口：主要是用来对当前所编辑的对象进行属性的设置。默认情况会在右侧下

方显示。可以在视图菜单中，选择“属性窗口”打开，也可以直接按 F4 键打开，还可以右击所编辑的对象，选择“属性”菜单打开。

（4）服务器资源管理器窗口：作用主要是设置本地或远程服务器，对服务器端的数据库或数据建立连接。该窗口可以在工具菜单上单击“视图”菜单打开，也可以直接使用快捷键 Ctrl+Alt+S 组合键打开。

（5）工具箱：通常隐藏在画面的左侧，主要为相应的项目文件提供控件工具。工具箱中具体显示的工具控件，会自动选择与当前编辑文件匹配的所用工具控件。

（6）错误列表窗口：默认情况下会在下方显示。可以通过在“视图”菜单中选择“错误列表”来实现，也可以直接使用快捷键 Ctrl+E 组合键来打开。该窗口的作用是显示程序的错误、警告和消息的内容和具体位置，对程序的调试和编写有很大帮助。

二、C# 编程的设计流程

（一）C# 可以用来开发三种不同类型的程序

（1）Windows Applications（Windows 应用程序）：所谓的 Windows 应用程序是指运行于 Windows 操作系统上的程序。针对用户界面的不同，又可以分为控制台应用程序（Console Application）及窗口应用程序（Windows Application）两种。前者就是在 Windows 的命令提示符窗口执行的程序，而后者就像是在 Windows 操作系统下执行的功能软件，包括菜单、工具栏、各式各样的窗口之类的程序。

（2）Web Application（Web 应用程序）：简单来说就是，以浏览器为运行平台的网页应用程序，一般称为 ASP.NET。

（3）Web Services（Web 服务）：简单来说就是，以 HTTP 为通信协议的类，一般称为 ASP.NET Web Services。

本书绝大部分的情况下，使用控制台应用程序作为范例来说明 C# 的语法及面向对象程序设计，对于其他类型的程序仅做一般性的入门介绍。

（二）C# 的基本编码规则

（1）标志符和保留字：C# 语言中，标识符是以字母、下划线“_”或“@”开始的一个字符序列，后面可以跟字母、下划线、数字。C# 语言区分大小写。一般变量名首字母小写，后面各单词首字母大写；而常量、类名、方法、属性等首字母大写。有些标识符具有专门的意义和用途，不能当做一般的标识符使用，这些标识符称为关键字或保留字，如 namespace、static、using。

（2）书写规则：每行语句以“;”结尾。空行和缩进被忽略。多条语句可以处于同一行，用分号分隔。大括号 { } 要成对出现。

1. 创建项目

解决方案是 Visual Studio 提供的有效管理应用程序的容器。一个解决方案可以包含一个或多个项目，而每个项目能够解决一个独立的问题。下面来创建第一个项目。在安装了

Visual Studio.NET 2015 集成开发环境的计算机上，可以通过选择“文件”菜单，指向“新建”，然后单击“项目”菜单项，这时会打开“新建项目”对话框。在左侧“项目类型”栏中选择“Visual C#”下的“Windows”选项，然后在右侧“模板”栏中选择“控制台应用程序”，最后在“名称”栏中填入项目名称“Hello”，单击“确定”按钮。按刚才的创建路径依次打开，可以看见一个名为 Hello 的文件夹。打开该文件夹，可以看见两个解决方案文件（扩展名为 .sln 和 .suo 的文件）。

Visual Studio.NET 采用两种文件类型（.sln 和 .suo）来存储特定解决方案的设置。这些文件总称为解决方案文件，为解决方案资源管理器提供显示管理文件的图形接口所需的信息。从而使用户每次开发任务时，都能够全身心地投入到项目和最终目标中，不会因开发环境而分散精力。

.sln：它是解决方案文件，使用 Visual Studio 创建解决方案会自动创建该文件。该文件为解决方案资源管理器，提供显示管理文件的图形接口所需的信息。该文件将一个或多个项目的所有元素组织到单个的解决方案中。

.suo：它是 solution user option 的缩写，是非常重要的文件，这种文件是隐藏的，使用 Visual Studio 创建解决方案时就会自动创建该文件。该文件储存了用户界面的自定义配置，包括布局、断点和项目最后编译的而又没有关掉的文件（下次打开时用）等，以便于下一次用户打开 Visual Studio 时，可以恢复这些设置。因此不要随便删除，当然也无法删除。

在 C# 中，项目是一个独立的编程单位，其中包含一些相关的文件，若干个项目就组成了一个解决方案。解决方案资源管理器以树状的结构，显示整个解决方案中包含哪些项目以及每个项目的组成信息。除了 .sln 和 .suo 类型的文件外，还有以下类型的文件。

bin 文件夹：包含一个子目录，含文件，即生成的可执行文件。

obj 文件夹：包含一个子目录，含编译过程中生成的中间代码，文件包含完整的调试信息。

.ico 文件：应用程序图标文件。

AssemblyInfo 模块：包含部件属性设置，用来设定生成的 dll 程序集的一些常规信息，部分信息可以在引用 dll 时，从属性中直接看到。

.csproj 文件：项目文件，创建应用程序所需的引用、数据连接、文件夹和文件的信息。

.csproj.user 文件：解决方案用户选项文件。

其他多个 .cs 文件：用户自定义的项目文件，即 C# 源代码文件。

2. 编写代码

这样，就创建了一个名为“Hello”的解决方案，在该解决方案中包含着一个同名的项目，在右边的“解决方案资源管理器”窗口中可以看到该项目所包含的所有文件。双击“Program.cs”文件，使其在代码编辑器中打开，观察文件中的代码。

这些代码是自动生成的，其中有这样一段：

```
static void Main(string [ ] args)
```

```
{
}
```

这是 C# 程序的主函数，是 C# 程序的起点，现在向主函数内添加一段代码。代码的功能是在控制台输出窗口中输出打印“Hello，World！”，代码如下：

```
//Program.cs
/* 第一个 C# 程序，C# 源程序代码的文件后缀名为 .cs*/
using System ; /// 包含 .NET 应用程序使用的大多数基本类型
using System.Collections.Generic ; /// 包含用于处理集合的泛型类型
using System.Text ; /// 包含与字符串处理和编码相关的类型
name space Hello
{
    class Program
    {
    static void Main(string[ ] args)
    {
        Console.WriteLine(“Hello，World！”);
    }
  }
}
```

下面从外向内介绍程序 Program.cs 代码的各个组成部分：

（1）namespace 关键字：在编写规模比较大的程序时，随着代码的增多，由于有较多的名称、命名数据、已命名方法以及已命名类等，这就很可能发生两个或者两个以上的名称冲突，造成程序的混乱甚至错误。微软在 .NET 中引入了命名空间（namespace），就是用来解决这个问题的。它为各种标志符创建一个已命名的逻辑容器，同名的两个类如果不在同一个命名空间中，就不会发生冲突。例如，System 是一个命名空间，Console 是该命名空间中的类。

（2）using 关键字：使用命名空间内的成员时，要在它们的前面加上一长串的命名空间限定：

```
命名空间 . 命名空间 .……. 命名空间 . 类名称 . 方法名（参数，……）;
```

例如：

```
System.Console.WriteLine(“Hello World”);
```

这显然很不方便，为了使得代码简洁，C# 语言中使用 using 语句来导入 .NET Framework System 类库提供的命名空间。例如：

```
using System ;
```

然后，在类中就可以这样写：

Console.WriteLine(“Hello，World”)；

调用不同类，实现所需要的功能。这样就可以在以后的程序中直接使用此命名空间中的类。在后面的章节中读者会逐渐了解这三个命名空间的作用，而且还会认识和了解更多的命名空间。表 1-2 所列是系统定义的命名空间的一部分。

表 1-2 常用命名空间

命名空间	说 明
System	定义通常使用的数据类型和数据转换的基本 .NET 类
System.Drawing	处理图形和绘图，包括打印
System.Text	供 ASCII、Unicode、UFT-7 和 UTF-8 字符编码处理
System.Data	ADO.NET 中处理数据存取和管理
System.XML	提供对 XML 文档的处理
System.IO	管理对文件和流的同步和异步访问
System.Windows	处理基于窗体的窗口的创建
System.Reflection	包含从程序集读取元数据的类
System.Threading	包含用于多线程编程的类
System.Collections	包含定义各种对象集的接口和类

在后续的章节中还会介绍一些其他的命名空间。

（3）class 关键字：C# 是一种面向对象的语言，使用 class 关键字表示类，程序员编写的任何代码都应该包含在一个类中（即将所有属于这个类的代码都放在类名后面的一对大括号中），类要包含在一个命名空间中。在程序模板生成时，Visual Studio 会自动起一个类名 Program。如果不喜欢这个命名，可以进行更改。C# 与 Java 不同，不要求类名必须与源文件的名字一样。C# 语言区分字母的大小写，如“Program”和“program”，但在编写程序时尽量不要通过大小写来区分名称。

（4）Main 方法：C# 中的 Main 方法（与 Java 中的 Main 方法作用相同），是应用程序的入口点，应用程序从这里开始运行。虽然程序不是从上往下逐条执行的，但也要从一个固定的位置开始执行，这个固定位置在程序中叫做“入口”。而 Main 方法就是 C# 程序的入口，Main 方法是所有 C# 应用程序的起始点，没有 Main 方法，计算机就不知道该从哪里开始执行程序。在编写 Main 方法时，要求按照上面的格式和内容进行书写，Main 方法前面可使用 public、static、void 修饰，而且顺序不要改变，中间用空格分隔。一个程序只能有一个 Main 方法，而且 C# 中的 Main 方法首字母必须大写。Main 方法的返回值可以用 void 或者