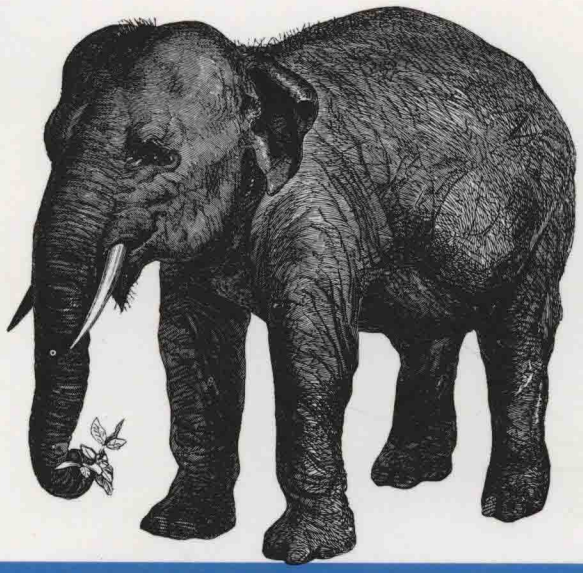
工业和信息化“十三五”
人才培养规划教材



C#
程序设计教程

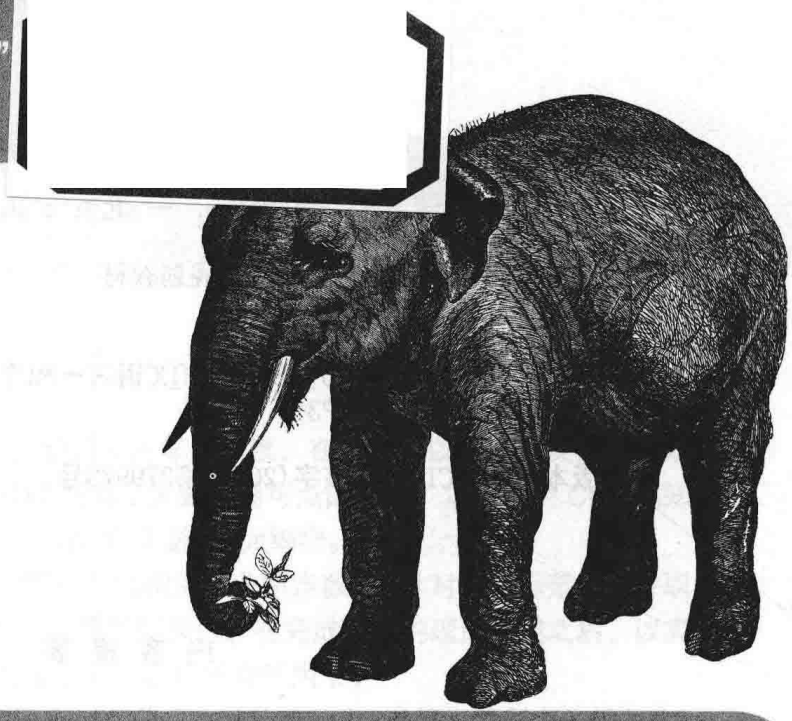
C# Programming

陈娜 付沛 ◎ 主编
罗炜 谢日星 鄢军霞 ◎ 副主编
王路群 ◎ 主审

循序渐进地介绍了 **C#** 入门所需的各方面知识，
对**基本概念**和**技术**进行了清楚准确的解释并**结合实例**加以说明。
以**实践应用**为目标，理论阐述主要围绕**实际应用技术**组织和展开。

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化“十三五”
人才培养规划教材



C# 程序设计教程

C# Programming

陈娜 付沛 ◎ 主编
罗炜 谢日星 鄢军霞 ◎ 副主编
王路群 ◎ 主审

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

C#程序设计教程 / 陈娜，付沛主编. -- 北京 : 人民邮电出版社，2019.1
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-39847-5

Ⅰ. ①C… Ⅱ. ①陈… ②付… 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(2018)第279973号

内 容 提 要

Visual C#融C++的灵活性和强大功能与Java的简单性于一身，已成为在.NET平台上进行程序开发的首选语言。

本书介绍了使用Visual C# 2017开发应用程序的基本知识。全书分为9章，分别介绍了C#与Visual Studio集成开发环境，C#的基本语法，流程控制语句，数组、集合和泛型，面向对象，面向对象的高级应用，程序的生成、异常处理和调试，流与文件，基于Windows的应用程序。

全书通过简洁的语言和详细的步骤，帮助读者迅速掌握使用Visual C# 2017开发应用程序所需的基本知识。

本书适合没有任何编程经验的读者和Visual C#新手阅读，也可供大中专院校的学生学习Visual C#编程时参考。通过本书，读者可循序渐进地掌握C#编程技术，从而开发出优秀的应用程序。

◆ 主　　编　陈　娜　付　沛
副 主 编　罗　炜　谢日星　鄢军霞
主　　审　王路群
责任编辑　祝智敏
责任印制　马振武

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
固安县铭成印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：15.75　　2019年1月第1版
字数：371千字　　2019年1月河北第1次印刷

定价：46.80元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字20170147号

前言 FOREWORD

Visual C#作为微软的旗舰编程语言，经过十几年的发展，在全球得以迅速普及，成为众多程序开发人员的首选语言。Visual C# 2017 新增了大量可圈可点的特性，本书围绕 C#基础知识和这些新特性全面介绍如何利用 Visual Studio 2017 进行 C#编程。

本书作为全国示范性软件职业学院计算机及其相关专业指定教材，针对全国示范性软件职业学院特点，淡化理论，够用为度，强化技能，重实际操作，在完成必要的理论阐述之后，以实际的代码案例来解释理论知识，适合作为计算机基础教材或自学用书。

本书是作者在多年的教学实践、科学研究以及项目实践的基础上，参阅大量国内外相关教材，几经修改而成。本书的主要特点如下。

1. 语言严谨、精练。

对 C#的基本概念和技术进行了清楚准确的解释并结合实例说明，读者可以轻松地掌握每一个知识点。

2. 合理、有效的组织。

按照由浅入深的顺序，循序渐进地系统介绍了 C#程序设计的相关知识和技能。各个章节的内容编写以实践应用为目标，理论阐述主要围绕实际应用技术组织和展开，使练习的重要性得以体现，练习不再只是附属于相关理论知识。

3. 本书配有全部的程序源文件和教学 PPT。

为方便读者使用，书中全部实例的源代码及 PPT 均免费提供给读者。

本书以 Visual Studio 2017 为基础循序渐进地介绍了 C#入门所需的各方面知识，包括开发环境的配置、C#语法、Windows 应用程序开发、文件处理等。同时还介绍了大量 Visual Studio 2017 的开发经验，对使用中的重点、难点进行了专门的讲解。

本书由陈娜、付沛担任主编，罗炜、谢日星、鄢军霞担任副主编，董宁、陈丹、张松慧、赵丙秀、张新华参加编写，王路群统审全稿。

由于时间仓促，加之编者水平有限，书中不妥之处在所难免，殷切希望广大读者批评指正，以便尽快更正，编者将不胜感激。作者 E-mail：231292594@163.com。

编　者

2018 年 9 月

目 录 CONTENTS

1 Chapter

第1章

C#与Visual Studio集成开发环境

本章学习目标

本章主要内容包括.NET 基础知识，Visual Studio 集成开发环境的使用，创建一个控制台应用程序，创建一个 Windows 应用程序和控制台应用程序结构简介。通过本章，读者应该掌握以下内容：

1. Visual Studio 集成开发环境的使用
2. 创建控制台应用程序
3. 创建简单的 Windows 应用程序
4. 控制台应用程序的结构

1.1 .NET 简介

.NET 是 Microsoft 的 XML Web 服务平台。Microsoft .NET 平台包含广泛的产品系列，都是基于 XML 和 Internet 行业标准构建的，不论操作系统或编程语言有何差别，XML Web 服务都能使应用程序在 Internet 上传输和共享数据。

.NET Framework 是构成 Microsoft .NET 平台核心部分的一组技术，为开发 Web 应用程序和 XML Web Service 提供了基本的构建模块，也为创建和运行.NET 应用程序提供了必要的编译和运行基础。

.NET Framework 是 Windows Server System 构建、部署与运行 Web 服务与应用程序的编程模型，其托管了大部分底层结构，让开发人员只需专注于撰写应用程序的业务逻辑代码。

.NET Framework 是支持生成和运行下一代 Web 应用程序和 XML Web Services 的内部 Windows 组件，旨在实现下列目标。

- 提供一个一致的面向对象的编程环境，无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于.NET Framework 的代码可与任何其他代码集成。

.NET Framework 有两个主要组件：公共语言运行库和.NET Framework 类库。公共语言运行库是.NET Framework 的基础，可以看作是一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且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保障。事实上，代码管理是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。.NET Framework 类库是一个综合性的面向对象的可重用类型集合，可以使用它开发多种应用程序，包括传统的命令行或图形用户界面（GUI）应用程序，也包括基于 ASP.NET 的应用程序（如 Web 窗体和 XML Web Services）。

1. 公共语言运行库

通用语言框架（Common Language Infrastructure，CLI）是一种为虚拟机环境而制订的规范，使得由各种高级语言所编制的程序可以在不同的系统环境中执行，而不必更改或重新编译源代码。

.NET 的基础公共语言运行库（Common Language Runtime，CLR）就是 CLI 的一个实例，只不过是 CLI 规范在个人计算机和 Windows 操作系统中的一个执行而已。毫无疑问，在其他操作系统环境和硬件平台上，CLI 也同样可行。CLI 和 CLR 有时会交换使用，但很明显它们不是一回事。CLI 是一种标准规范，而 CLR 是微软对 CLI 的实现。

公共语言运行库也叫公共语言运行时，是.NET Framework 的基础。公共语言运行库简化了应用程序的开发，提供了一个强大的、安全的执行环境，支持多语言，并简化了应用程序的部署和管理。公共语言运行库也称为“托管环境”，在这个托管环境中自动提供诸如垃圾回收和安全性等通用服务。

例如，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，通常具有扩展名.exe 或.dll。程序集包含的清单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 代码转换为本地机器指令。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 执行的代码有时也称为“托管代码”，它与编译为面向特定系统的本地机器语言的“非托管代码”相对应。图 1-1 展示了 C#源文件、.NET Framework 类库、托管程序集和 CLR 的编译时与运行时的关系。

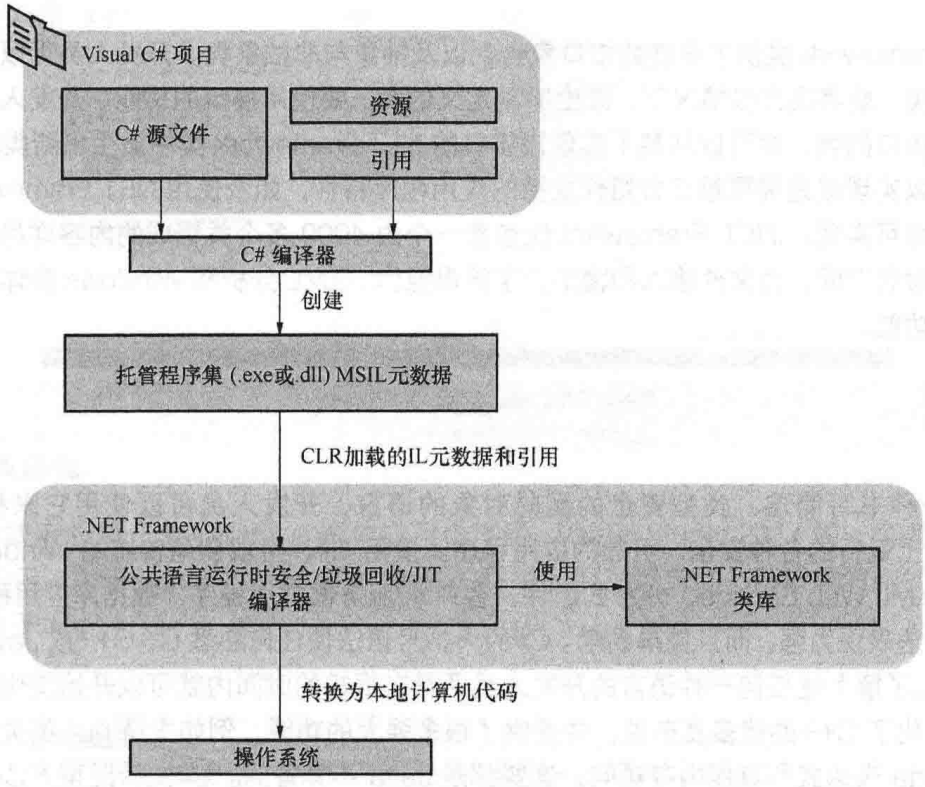


图 1-1 关系图

语言互操作性是.NET Framework 的一项主要功能。由 C#编译器生成的 IL 代码符合通用类型系统（Common Type System，CTS）规范，因此由 C#生成的 IL 代码可以与 Visual Basic、Visual C++、Visual J#的.NET 版本或者其他 20 多种符合 CTS 规范的语言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，并且类型之间可以相互引用，就像它们是用同一种语言编写的一样。

公共语言运行库还提高了开发人员的工作效率。例如，开发人员可以用他们选择的开发语言编写应用程序，仍能充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以公共语言运行库为目标的编译器供应商都可以这样做。以.NET Framework 为目标的语言编译器

使得用该语言编写的现有代码可以使用.NET Framework 的功能，这大大减轻了迁移现有应用程序的工作负担。

公共语言运行库负责运行时服务，如语言集成、强制安全，以及内存、进程和线程管理。除此之外，它还在开发时期承担如生命周期管理、强类型命名、跨语言异常处理以及动态绑定之类的角色，以减少开发人员将事务逻辑转换成可重用组件必须编写的代码数量。

公共语言运行库为开发人员构建不同类型的应用程序提供了可靠的基础，让设计含有跨语言对象的组件与应用程序变得更加容易。不同语言编写的对象可以互相通信，它们的行为可以被紧密集成。

2. .NET Framework 类库

在早期的开发中，各种应用的开发人员使用各自平台提供的工具类库，开发适用于不同平台的应用，开发人员要掌握多种类库的使用方法，因而造成大量的资源浪费，也降低了开发人员的工作效率。

.NET Framework 提供了丰富的接口集合，以及抽象与非抽象类。开发人员可以原封不动地使用非抽象类，或者在许多情况下，派生出自定义的类。要使用接口的功能，开发人员既可以创建一个实现接口的类，也可以从某个实现该接口的.NET Framework 类中派生出新类。

曾经难以实现或是需要第三方组件支持的应用程序特性，如今使用.NET Framework 后，通过少量代码即可实现。.NET Framework 还包含一个由 4000 多个类组成的内容详尽的库，这些类被组织为命名空间，为文件输入和输出、字符串操作、XML 分析和 Windows 窗体控件提供了各种有用的功能。

1.2 C#

C#是一种书写简洁、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。使用 C#，可以创建传统的 Windows 客户端应用程序、XML Web Service、分布式组件、客户端/服务器应用程序、数据库应用程序等。

C#的语法表现力强，而且简单易学。C#的大括号语法使任何熟悉 C、C++或 Java 的人都可以立即上手。了解上述任何一种语言的开发人员通常在很短的时间内就可以开始使用 C# 高效地工作。C#简化了 C++的诸多复杂性，并提供了很多强大的功能，例如支持 null 值类型、枚举、委托、lambda 表达式和直接内存访问，这些都是 Java 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。C#还提供了迭代器，允许集合类的实施者定义自定义的迭代行为，以便被客户端代码使用。

作为一种面向对象的语言，C#支持封装、继承和多态。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。类只能直接从一个父类继承，但可以实现任意数量的接口。重写父类中的虚方法要求使用 override 关键字来避免意外重定义。在 C#中，结构类似于一个轻量类，是一种使用堆栈的类型，可以实现接口，但不支持继承。

C#的生成过程比 C 和 C++简单，比 Java 灵活。C#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。

1.3 Visual Studio 集成开发环境

Visual Studio 2017 是微软于 2017 年 3 月 8 日正式推出的新版本，也是迄今为止最具生产力的 Visual Studio 版本。其内建工具整合了.NET Core、Azure 应用程序、微服务(microservice)、Docker 容器等。

1. 起始页

单击“开始”→“所有程序”→“Visual Studio 2017”，启动 VS2017 (Visual Studio 2017 的缩写)，在默认情况下会显示图 1-2 所示的起始页。

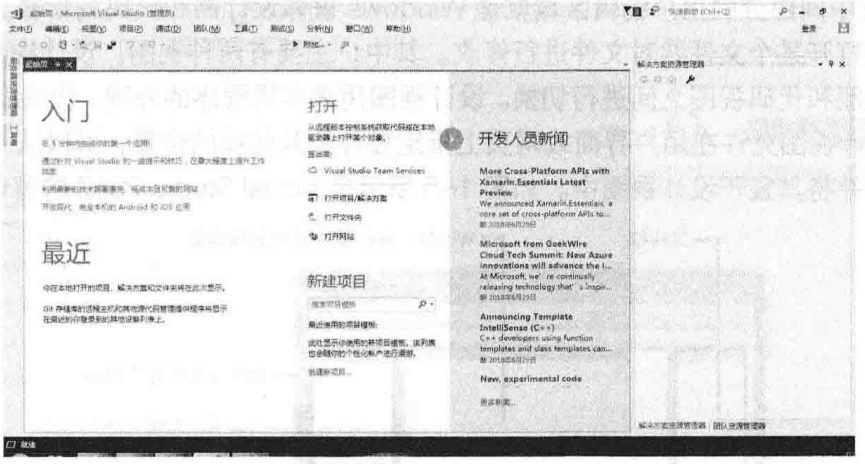


图 1-2 Visual Studio 2017 起始页

2. 开发环境

当打开或者新建一个 Windows 窗体应用程序后，Visual Studio 2017 的一个典型开发环境如图 1-3 所示。由于 Visual Studio 2017 的开发环境布局可以定制，所以你看到的界面有可能会与图 1-3 不同。

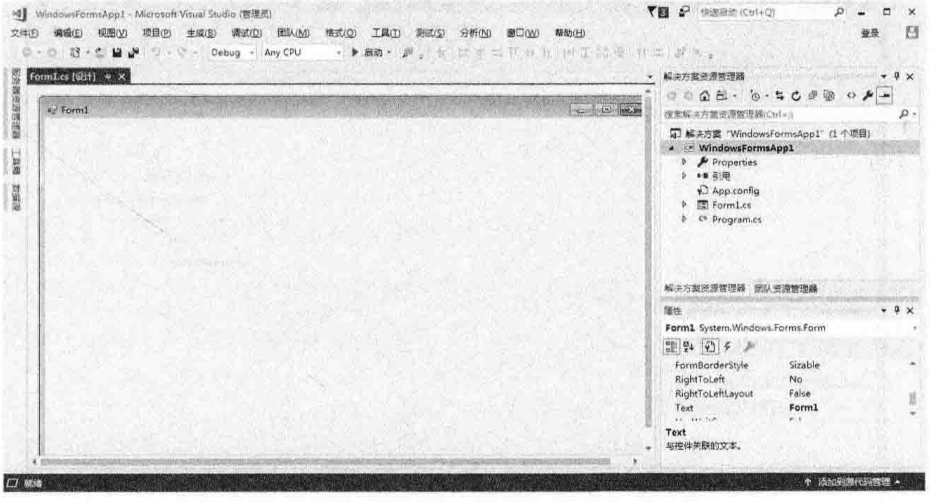


图 1-3 Visual Studio 2017 开发环境

Visual C#集成开发环境（IDE）是通过常用用户界面公开的开发工具的集合。有些工具是与其他 Visual Studio 语言共享的，还有一些工具（如 C#编译器）是 Visual C#特有的。

以下是 Visual C#中最重要的工具和窗口。大多数工具和窗口可通过“视图”菜单打开，这里仅介绍初学者需要掌握的 5 个窗口。

- 代码编辑器：用于编写源代码。
- 工具箱：用于使用鼠标快速开发用户界面。
- 解决方案资源管理器：用于查看和管理项目文件和设置。
- 属性窗口：用于配置用户界面中控件的属性和事件。
- 任务列表：常用来显示错误列表。

（1）Windows 窗体设计器/代码编辑器

图 1-4 正中间部分的用户编辑区域就是 Windows 窗体设计器和代码编辑器。在用户编辑区域，用户可以打开某个文件并对文件进行修改。其中，主要有两种视图：设计视图和代码视图，可以在设计视图和代码视图之间进行切换。设计视图用来实现程序的外观，代码视图用来实现程序的功能。设计视图允许在用户界面或网页上指定控件和其他项的位置，可以从“工具箱”中轻松拖动控件，并将其置于设计视图中。如图 1-5 所示是 Visual Studio 2017 的窗体设计视图。

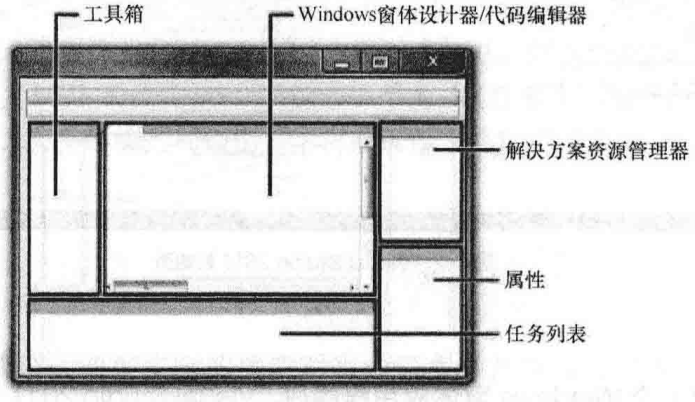


图 1-4 Visual C# 集成开发环境 (IDE)示意图

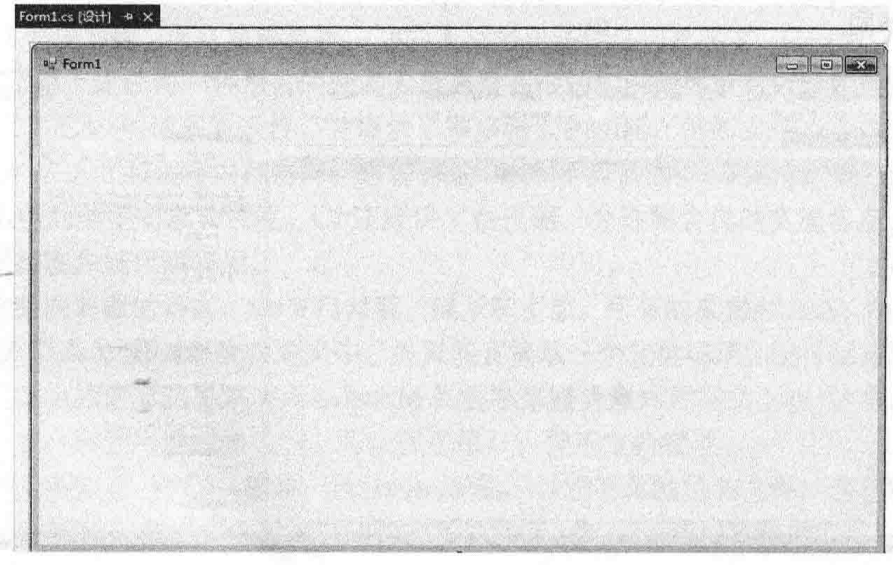


图 1-5 Visual Studio 2017 窗体设计视图

在窗体设计视图里，以可视化的方式显示组件（如 Windows 窗体、Web 页面、用户控件和数据集等）。Visual Studio 2017 最重要的特点就是所见即所得（What You See Is What You Get），看到的界面就是程序运行的最终效果。开发人员可以修改窗体的布局和设置，用户可以通过单击选中一个窗体或者控件，也可以通过鼠标的拖放来改变控件或窗体的位置和大小。

在设计视图下单击菜单“视图”→“代码”，可以切换到代码视图，如图 1-6 所示，用于显示文件或文档的源代码。代码视图支持编码帮助功能，如 IntelliSense（智能感知）、可折叠代码节、重构和代码段插入等，还有一些其他功能，如自动换行、书签和显示行号等。在代码视图中，用户可以编写代码，实现想要完成的功能。单击菜单“视图”→“设计器”，可以切换到设计视图。如果开发人员打开了多个文件，这些文件将以标签的方式显示在用户编辑区域的顶部，标签名即为文件名。如果标签名后面带一个“*”号，如图 1-7 所示，则表明这个文件已经进行了修改但尚未保存。单击工具栏上的按钮，即可保存全部修改，“*”号消失。

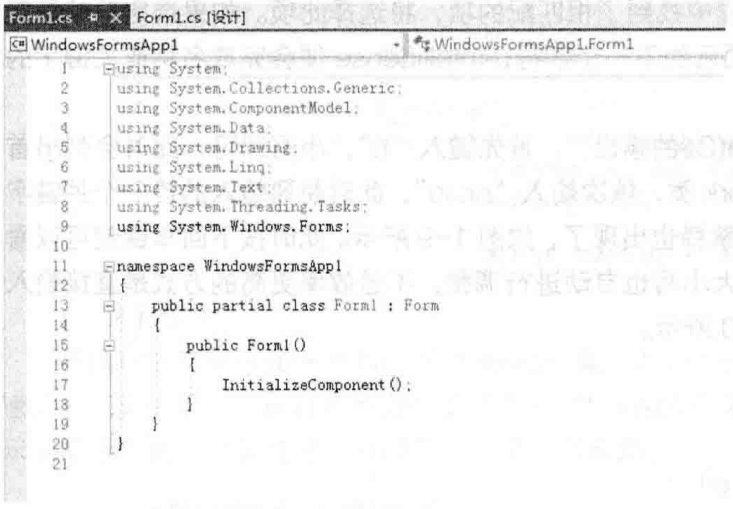


图 1-6 Visual Studio 2017 代码视图

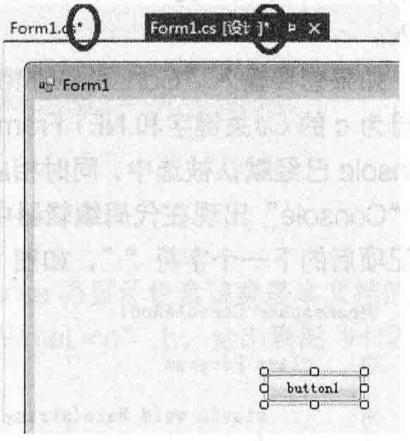


图 1-7 带“*”号的文件名（设计视图和代码视图都带）

注意

左边的“工具箱”和右下方的“属性”窗口仅在设计视图中才可用。切换到代码视图后，“工具箱”和“属性”窗口均不可用，如图 1-8 所示。

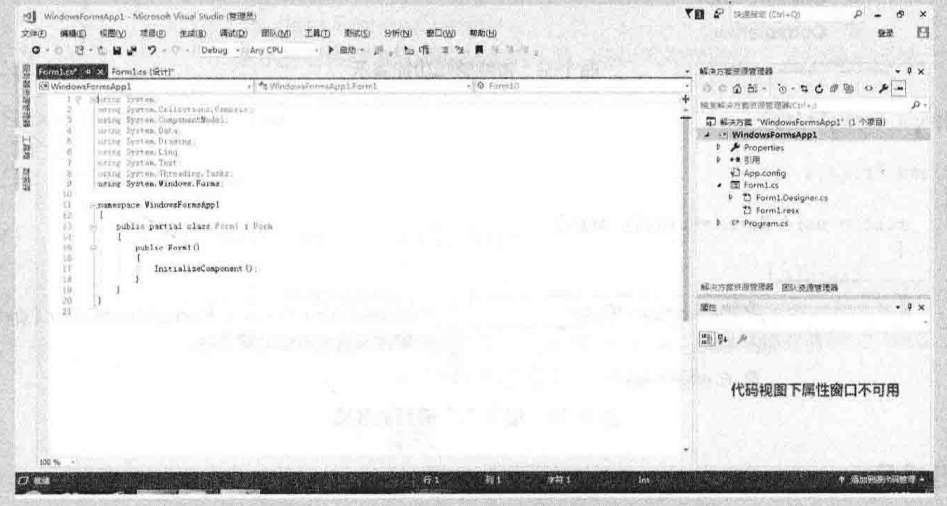


图 1-8 代码视图下的“工具箱”和“属性”窗口

Visual C#代码编辑器是编写源代码的字处理程序。就像 Microsoft Word 对句子、段落和语法提供广泛支持一样，C#代码编辑器也为 C#语法和.NET Framework 提供广泛支持。这些支持主要包括以下三个主要的类别。

① IntelliSense

在代码编辑器中键入.NET Framework 类和方法时，会不断地对其基本文档进行更新，同时具有自动代码生成功能。

IntelliSense 是一组相关功能的总称，旨在减少查找帮助所需的时间，有助于开发人员更加准确高效地输入代码。IntelliSense 提供了在代码编辑器中键入的关键字、.NET Framework 类型和方法签名的基本信息，这些信息会显示在工具提示、列表框和智能标记中。

- 完成列表

在代码编辑器中输入源代码时，IntelliSense 将显示一个包含所有 C#关键字和.NET Framework 类的列表框。如果在列表框中找到了相匹配的项，将选择此项。如果选定项就是需要的，只需按 Tab 键、回车键或匹配项后的下一个字符，IntelliSense 便会完成名称或关键字的输入。

如果想要输入“Console.WriteLine(C#的输出)”，首先键入“c”，小写即可，此时会列出首字母为 c 的 C#关键字和.NET Framework 类，依次输入“onso”，此时想要输入的第一个关键字 Console 已经默认被选中，同时相应的解释也出现了，如图 1-9 所示。此时按下回车键就可以看见“Console”出现在代码编辑器中，大小写也自动进行调整，不过效率更高的方式是直接输入匹配项后的下一个字符“.”，如图 1-10 所示。

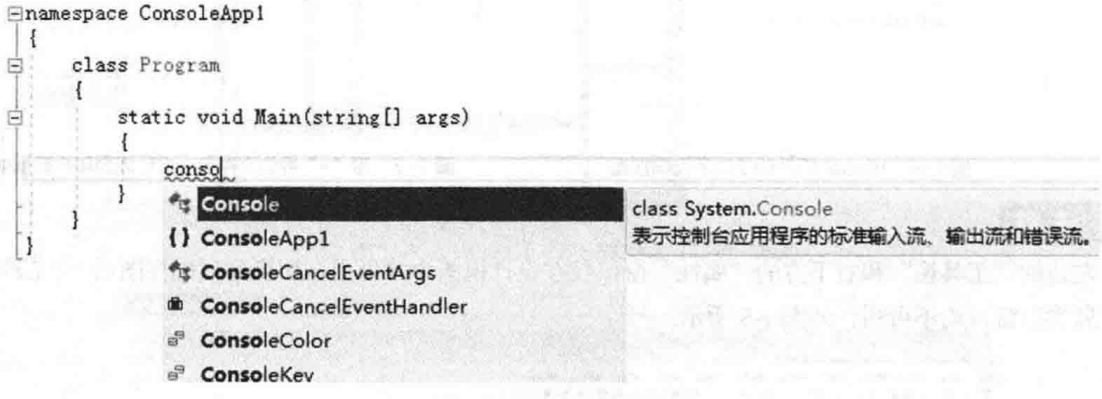


图 1-9 智能感知功能演示

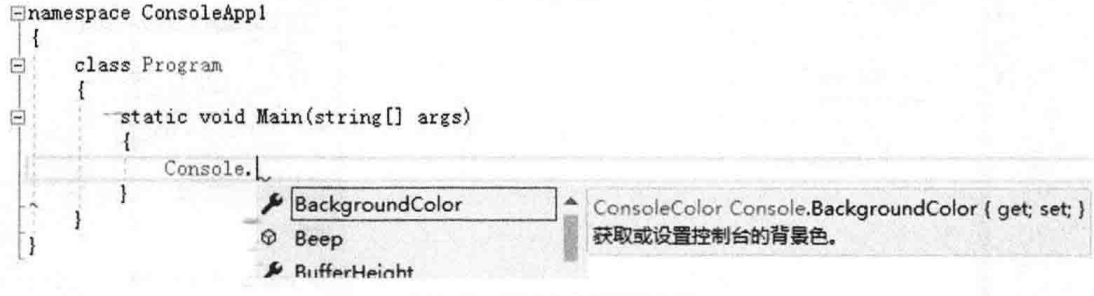


图 1-10 按下“.”键后的效果

- 列出成员

将一个.NET Framework 类输入代码编辑器，再键入点运算符（.），IntelliSense 将显示包含

该类成员的列表框。如果键入的内容有一个以上可能的匹配或根本没有匹配（例如输入“w”），将显示成员列表框。使用“↑”键或者“↓”键可以选择列表中的某个成员，当选中 WriteLine 后，在按回车键插入之前，将获得有关该项的快速信息和该项的所有代码注释。列表项左边的图标表示成员的类型，如命名空间、类、函数或变量。可以按 Tab 键或回车键将该成员插入到代码中，当然最好的办法是输入匹配项后的下一个字符，例如“(”。

- 参数信息

在代码编辑器中输入方法名称，再键入左括号后，会出现参数信息提示工具，其中显示了参数的顺序和类型，如图 1-11 所示。如果已重载此方法，可以在所有已重载的方法中上下滚动进行查找。

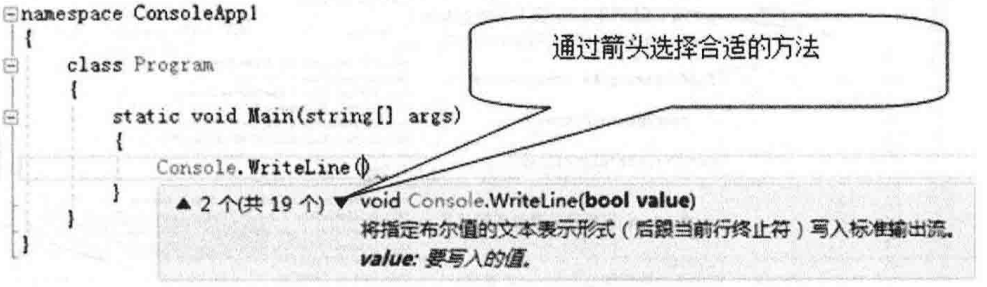


图 1-11 参数信息提示工具

- 快速信息

将鼠标指针悬停在一个.NET Framework 类上时，IntelliSense 将显示包含该类基本文档的快速信息工具提示。将鼠标指针分别放在单词“Console”和“WriteLine”上，会出现图 1-12 所示的提示信息，这些信息对程序开发人员很有帮助。

图 1-12 出现快速提示信息