

普通高等教育“十三五”规划教材 (软件工程专业)

# C# 程序设计教程

主 编 李祥琴  
副主编 罗传军 张 牧 杨 利 周东来

Software

- ◎ **内容全面、结构完整**：知识点介绍由浅入深、循序渐进。
- ◎ **案例典型实用**：代码详细、步骤清晰。
- ◎ **巩固提高**：习题题型丰富、题量适中。
- ◎ **资源立体配套**：提供全部源码和配套实验指导。



普通高等教育“十三五”规划教材（软件工程专业）

# C#程序设计教程

主 编 李祥琴

副主编 罗传军 张 牧 杨 利 周东来



中国水利水电出版社  
www.waterpub.com.cn

·北京·

## 内 容 提 要

本书以 Visual Studio 2017 为操作平台,介绍了 C#程序设计的方法,全书共 13 章: C#语言概述、C#程序设计基础、流程控制、面向对象程序设计基础、继承与接口、数组与集合、泛型、委托与事件、Windows 窗体应用程序设计、界面设计、线程编程、文件操作、ADO.NET 数据访问技术。

本书内容丰富、由浅入深,特别注重实用性和引导性,书中列举大量编程实例,突出对应用能力的培养,便于初学者学习。

本书可作为高等院校计算机及相关专业教材,也可作为软件开发人员和计算机编程爱好者的参考书。

## 图书在版编目 (C I P) 数据

C#程序设计教程 / 李祥琴主编. — 北京: 中国水利水电出版社, 2019. 5  
普通高等教育“十三五”规划教材. 软件工程专业  
ISBN 978-7-5170-7650-6

I. ①C… II. ①李… 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(2019)第080351号

策划编辑: 杨庆川

责任编辑: 张玉玲

封面设计: 李 佳

书 名	普通高等教育“十三五”规划教材(软件工程专业) C#程序设计教程 C# CHENGXU SHEJI JIAOCHENG
作 者	主 编 李祥琴 副主编 罗传军 张 牧 杨 利 周东来
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市铭浩彩色印装有限公司
规 格	184mm×260mm 16开本 20.75印张 508千字
版 次	2019年5月第1版 2019年5月第1次印刷
印 数	0001—3000册
定 价	48.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换  
版权所有·侵权必究

# 前 言

C#是微软公司推出的一种面向对象的、运行于.NET Framework 平台上的编程语言。它继承了 C 和 C++的强大功能,又吸收了 Java 等其他语言的优点,是一种非常有竞争力的程序设计语言。本书开发环境使用 Visual Studio 2017,数据库管理系统使用 SQL Server 2012。本书全面细致地介绍了 C#程序设计语言的基础知识,结合大量典型实例和实验,帮助读者掌握 C#的编程方法和技巧,为今后学习相关的程序设计课程或从事 C#编程工作打下坚实的基础。

本书共分为 13 章:第 1 章介绍 C#语言的特点、.NET Framework 的体系结构、Visual Studio 2017 集成开发环境和 C#应用程序的开发步骤;第 2 章介绍 C#的数据类型、常量和变量、运算符与表达式;第 3 章介绍条件选择控制语句、循环控制语句、跳转语句和异常处理;第 4 章介绍面向对象的基本概念、类、对象、方法、构造函数与析构函数;第 5 章介绍继承的定义、派生类的声明、派生类的初始化顺序、覆写基类方法、虚方法、抽象类和抽象方法、接口的成员与实现;第 6 章介绍一维数组、二维数组、交错数组、集合的定义和使用;第 7 章介绍泛型类、泛型方法、泛型约束和泛型集合;第 8 章介绍委托的声明、实例化和调用,以及事件的定义和使用;第 9 章介绍 Windows 窗体的属性、窗体的事件和方法、常用控件、多文档界面、继承窗体的创建;第 10 章介绍菜单的结构、下拉式菜单和弹出式菜单、工具栏和状态栏的设计方法、通用对话框;第 11 章介绍线程的创建、线程的挂起与恢复、线程休眠、阻塞线程、终止线程,以及实现线程同步的方法;第 12 章介绍文件操作、文件夹操作、使用流对文件进行读写;第 13 章介绍 ADO.NET 对象模型、Connection 对象、Command 对象、DataReader 对象、DataSet 对象、DataAdapter 对象,以及常用的数据绑定控件。为了帮助读者巩固知识点,每章最后提供了一定数量的练习题供选用。同时,为了方便上机练习,提高程序设计的综合能力,本书提供了配套的实验指导,涵盖了 C#的主要内容,读者可从中国水利水电出版社网站或万水书苑下载,网址:<http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

本书具有如下特点:

- (1) 内容全面、结构完整,由浅入深、循序渐进地介绍各个知识点。
- (2) 精心选择典型案例,代码详细,步骤清晰。
- (3) 各章都提供了大量习题,帮助读者巩固所学知识。
- (4) 为便于教学,本书提供全部源码以及配套的实验指导,可读性强。

本书由李祥琴任主编,罗传军、张牧、杨利、周东来任副主编。其中,荆楚理工学院的李祥琴编写第 1 章、第 6 章至第 9 章、第 13 章,荆门市电子政务信息中心的罗传军编写第 4 章和第 10 章,荆楚理工学院的张牧编写第 11 章和第 12 章,池州学院的杨利编写第 2 章和第 3 章,荆州职业技术学院的周东来编写第 5 章,全书由李祥琴统稿。

在本书编写过程中，编者得到了荆楚理工学院、池州学院和荆州职业技术学院的大力支持和帮助，游明坤、武永成、胡秀、李俊梅、刘珊燕等多位老师提出了宝贵的意见和建议，在此一并表示感谢。

由于时间仓促及编者水平有限，书中难免存在一些疏漏和不足，恳请广大读者批评指正。

编者

2019年3月

# 目 录

前言	
第1章 C#语言概述	1
1.1 什么是C#语言	1
1.1.1 C#语言的发展历史	1
1.1.2 C#语言的特点	2
1.2 .NET 概述	3
1.2.1 .NET 平台	3
1.2.2 什么是.NET Framework	4
1.2.3 .NET Framework 体系结构	4
1.2.4 .NET 程序执行过程	6
1.3 Visual Studio 2017 集成开发环境	6
1.3.1 Visual Studio 2017 的安装	7
1.3.2 Visual Studio 2017 界面介绍	10
1.3.3 Visual C#开发环境的配置	11
1.4 C#应用程序	12
1.4.1 控制台应用程序	13
1.4.2 Windows 应用程序	14
1.4.3 Web 应用程序	17
习题 1	18
第2章 C#程序设计基础	20
2.1 C#程序的相关元素	20
2.1.1 标识符	20
2.1.2 关键字	21
2.1.3 命名空间	21
2.2 数据类型	22
2.2.1 值类型	23
2.2.2 引用类型	28
2.2.3 数据类型转换	29
2.2.4 .NET 支持的方法转换	32
2.3 常量和变量	33
2.3.1 常量的定义和使用	33
2.3.2 变量的声明和赋值	34
2.4 运算符与表达式	35
2.4.1 算术运算符	35
2.4.2 关系运算符	36
2.4.3 赋值运算符	36
2.4.4 逻辑运算符	37
2.4.5 位运算符	37
2.4.6 条件运算符	39
2.4.7 其他运算符	39
2.4.8 运算符的优先级	40
习题 2	41
第3章 流程控制	44
3.1 条件选择语句	44
3.1.1 if 语句	44
3.1.2 switch 语句	49
3.2 循环语句	51
3.2.1 while 语句	51
3.2.2 do...while 语句	53
3.2.3 for 语句	54
3.2.4 foreach 语句	55
3.3 跳转语句	57
3.3.1 break 语句	57
3.3.2 continue 语句	58
3.3.3 goto 语句	59
3.3.4 return 语句	60
3.4 异常处理	61
3.4.1 异常处理语句	62
3.4.2 常用的异常类	66
3.4.3 用户自定义异常	66
习题 3	67
第4章 面向对象程序设计基础	69
4.1 面向对象概述	69
4.1.1 面向对象的基本概念	69
4.1.2 面向对象的特点	70
4.2 类	71
4.2.1 类的声明	72

4.2.2 类的成员	73	6.1.5 Array类	119
4.2.3 嵌套类	78	6.2 集合	121
4.2.4 分部类	78	6.2.1 集合接口	121
4.3 对象	79	6.2.2 ArrayList类	122
4.3.1 对象的声明和实例化	79	6.2.3 Queue类	124
4.3.2 访问类的对象	80	6.2.4 Stack类	126
4.4 方法	81	6.2.5 Hashtable类	128
4.4.1 方法的声明和调用	81	6.2.6 SortedList类	131
4.4.2 静态方法和实例方法	82	习题6	133
4.4.3 方法的参数	83	第7章 泛型	135
4.4.4 方法的重载	86	7.1 泛型概述	135
4.5 构造函数和析构函数	87	7.1.1 泛型的定义	135
4.5.1 构造函数	88	7.1.2 泛型类	135
4.5.2 析构函数	89	7.1.3 泛型结构	138
习题4	90	7.1.4 泛型接口	139
第5章 继承与接口	93	7.1.5 泛型方法	140
5.1 继承	93	7.1.6 泛型的继承	141
5.1.1 继承的定义	93	7.2 泛型约束	141
5.1.2 派生类的声明	94	7.3 泛型集合	146
5.1.3 派生类的初始化顺序	95	7.3.1 List<T>	146
5.1.4 密封类	98	7.3.2 Queue<T>	150
5.1.5 静态类	99	7.3.3 Stack<T>	152
5.2 多态	100	7.3.4 Dictionary<K,V>和 KeyValuePair<K,V>	154
5.2.1 虚方法	100	7.3.5 SortedList<K,V>	157
5.2.2 覆写基类方法	101	7.3.6 HashSet<T>	159
5.2.3 隐藏基类方法	102	习题7	161
5.2.4 抽象类和抽象方法	103	第8章 委托与事件	162
5.3 接口	104	8.1 委托的定义和使用	162
5.3.1 接口的定义	105	8.1.1 委托的声明	162
5.3.2 接口的成员	105	8.1.2 委托的实例化	162
5.3.3 接口的实现	106	8.1.3 委托的调用	164
5.3.4 接口的派生	108	8.2 多播委托	165
习题5	110	8.3 匿名方法	167
第6章 数组与集合	112	8.4 委托中的协变和逆变	169
6.1 数组	112	8.5 Lambda表达式	170
6.1.1 一维数组	112	8.6 委托的应用	172
6.1.2 二维数组	115	8.7 事件	174
6.1.3 多维数组	117	8.7.1 事件的概念	174
6.1.4 交错数组	118		

8.7.2 声明事件	175	10.3 对话框	237
8.7.3 订阅事件	175	10.3.1 打开文件对话框	238
8.7.4 触发事件	178	10.3.2 保存文件对话框	240
8.7.5 扩展 EventArgs 类	180	10.3.3 颜色对话框	240
8.7.6 事件访问器	182	10.3.4 字体对话框	241
习题 8	183	10.3.5 打印对话框	241
第 9 章 Windows 窗体应用程序设计	185	习题 10	244
9.1 窗体设计	185	第 11 章 线程编程	246
9.1.1 创建窗体	185	11.1 线程概述	246
9.1.2 窗体的类型	187	11.1.1 单线程简介	246
9.1.3 窗体的属性	187	11.1.2 多线程简介	247
9.1.4 窗体的方法	189	11.2 线程控制	248
9.1.5 窗体的事件	191	11.2.1 Thread 类	248
9.2 常用控件	193	11.2.2 线程的创建与启动	251
9.2.1 Label 控件	193	11.2.3 线程的挂起与恢复	252
9.2.2 TextBox 控件	195	11.2.4 线程休眠	252
9.2.3 RichTextBox 控件	199	11.2.5 阻塞线程	253
9.2.4 Button 控件	203	11.2.6 终止线程	254
9.2.5 ListBox 控件	204	11.3 线程优先级	255
9.2.6 CheckedListBox 控件	207	11.4 线程的同步	257
9.2.7 ComboBox 控件	208	11.4.1 使用 lock 关键字	257
9.2.8 RadioButton 控件	210	11.4.2 使用 Monitor 类	259
9.2.9 CheckBox 控件	212	11.4.3 使用 Mutex 类	260
9.2.10 GroupBox 控件	213	习题 11	261
9.2.11 TabControl 控件	215	第 12 章 文件操作	263
9.2.12 PictureBox 控件	217	12.1 文件和 System.IO 模型概述	263
9.2.13 Timer 控件	218	12.1.1 文件类型	263
9.3 多文档界面	219	12.1.2 文件的属性	264
9.4 继承窗体设计	223	12.1.3 文件访问方式与文件流	264
9.4.1 继承窗体概述	223	12.1.4 System.IO 模型	264
9.4.2 创建继承窗体	224	12.2 对文件进行操作	265
习题 9	225	12.2.1 File 类	265
第 10 章 界面设计	228	12.2.2 FileInfo 类	267
10.1 菜单	228	12.3 对文件夹进行操作	268
10.1.1 MenuStrip 控件	229	12.3.1 Directory 类	268
10.1.2 ContextMenuStrip 控件	231	12.3.2 DirectoryInfo 类	270
10.2 工具栏和状态栏	232	12.4 使用流对文件进行读写	271
10.2.1 ToolStrip 控件	232	12.4.1 FileStream 类	271
10.2.2 StatusStrip 控件	234	12.4.2 StreamWriter 类和 StreamReader 类	274

12.4.3 BinaryWriter 类和 BinaryReader 类	278	13.4.2 创建和使用 DataReader 对象	300
习题 12	281	13.5 DataSet 对象	301
<b>第 13 章 ADO.NET 数据访问技术</b>	<b>283</b>	13.5.1 DataSet 对象概述	301
13.1 ADO.NET 概述	283	13.5.2 DataSet 对象的属性和方法	302
13.1.1 ADO.NET 简介	283	13.5.3 DataTable 对象	303
13.1.2 ADO.NET 数据提供程序	284	13.5.4 DataColumn 对象	305
13.1.3 ADO.NET 对象模型	285	13.5.5 DataRow 对象	306
13.2 Connection 对象	287	13.5.6 DataView 对象	308
13.2.1 Connection 对象的连接字符串	287	13.6 DataAdapter 对象	310
13.2.2 Connection 对象的创建	288	13.6.1 DataAdapter 对象的属性和方法	310
13.2.3 Connection 对象的属性和方法	289	13.6.2 创建 DataAdapter 对象	311
13.2.4 连接池	291	13.6.3 使用 DataAdapter 对象	312
13.3 Command 对象	292	13.7 数据绑定控件	314
13.3.1 创建 Command 对象	292	13.7.1 BindingSource 控件	314
13.3.2 Command 对象的属性	293	13.7.2 BindingNavigator 控件	317
13.3.3 Command 对象的方法	294	13.7.3 DataGridView 控件	318
13.4 DataReader 对象	299	习题 13	321
13.4.1 DataReader 对象的属性和方法	299	参考文献	324

# 第 1 章 C#语言概述

## 【学习目标】

- 掌握 C#语言的特点。
- 了解 .NET Framework 的体系结构。
- 熟悉 Visual C#集成开发环境。
- 掌握 C#控制台应用程序的创建、打开和运行方法。

## 1.1 什么是 C#语言

C#（读作 C sharp）是微软公司开发的一种简单、安全、现代、面向对象的高级程序设计语言，它源自 C/C++ 家族，继承了 C 的语言风格和 C++ 的面向对象特性，继承它们强大的功能的同时去掉了一些复杂特性。C# 看似基于 C++ 写成，但又借鉴了其他语言如 Pascal、Java、Visual Basic、Delphi 等，它综合了 Visual Basic 简单的可视化操作。对于 Web 开发而言，C# 有点像 Java，同时具有 Delphi 的一些优点。

### 1.1.1 C#语言的发展历史

1998 年，Delphi 语言的设计者 Hejlsberg 带领着 Microsoft（微软）公司的开发团队开始了第一个版本 C#语言的设计。2000 年 9 月，国际信息和通信系统标准化组织为 C#语言定义了一个微软公司建议的标准。最终 C#语言在 2001 年得以正式发布。

2002 年 2 月，微软推出了 Visual Studio .NET 2002 和 C#语言的第一个正式版本 C# 1.0，随后推出 C# 1.1 版本。

2003 年 4 月，微软推出了 Visual Studio .NET 2003，发布了 C# 1.2 版本。

2005 年 11 月，微软推出了 Visual Studio 2005，发布了 C# 2.0 版本，该版本的两大特性是泛型、匿名方法。

2007 年 11 月，微软推出了 Visual Studio 2008，发布了 C# 3.0 版本，该版本改进了许多功能，查询表达式 LINQ 是其中最大的一个特性。

2010 年 4 月，微软推出了 Visual Studio 2010，发布了 C# 4.0 版本，该版本又增加和改进了许多功能：动态绑定、命名参数和可选参数、泛型的协变和逆变、开启嵌入类型信息及增加引用 COM 组件程序的中立。

2012 年 8 月，微软推出了 Visual Studio 2012，发布了 C# 5.0 版本，对比 C# 4.0 版本，增加了异步方法和调用方信息特性。

2015 年 6 月，微软推出了 Visual Studio 2015，发布了 C# 6.0 版本，该版本在之前的基础上增加了引用静态类、自动属性初始化等特性。

2017 年 3 月，微软推出了 Visual Studio 2017，发布了 C# 7.0 版本，引入了许多新特性：

变量直接声明、元组、模式匹配、引用返回值和局部变量、`async` 中使用泛型返回类、`Throw` 可以在表达式中使用、数字分隔符、局部函数等。

### 1.1.2 C#语言的特点

C#是微软公司专门为支持.NET框架的开发而设计的语言,它继承了C和C++的强大功能,吸收了Java等其他语言的优点,成为一款非常有竞争力的语言。C#语言主要具有以下特点:

#### (1) 语法简洁。

C#语言继承了C语言的简洁性,不允许直接存取内存,它的最大特色是没了指针,用引用操作代替。C++中使用各种不同的操作符,而在C#中只支持“.”,比如用“.”操作符代替了C++中的“::”“.”“->”操作符。

C#对C++中的语法冗余也进行了简化,只保留了常见的形式。

#### (2) 彻底的面向对象。

C#具有面向对象语言所应有的一切特性,例如封装、继承和多态性。与C++相比,C#没有全局变量和全局函数等,所有的成员都必须封装在类中,代码具有更好的可读性,减少了发生命名冲突的可能;在C#的类型系统中,每种类型都可以看作一个对象;C#也不支持多重继承,只允许单继承,从而避免了类型定义的混乱。

#### (3) 与Web紧密结合。

由于有了Web服务框架的帮助,对程序员来说,Web服务看起来更像是C#的本地对象。C#组件将能够方便地为Web服务,并允许它们通过Internet被运行在任何操作系统上的任何语言所调用。例如,XML已经成为网络中数据结构传递的标准,为了提高效率,C#允许直接将XML数据映射成为结构,这样就可以有效地处理各种数据。

#### (4) 版本处理技术。

C#提供内置的版本支持来减少开发费用,使用C#将会使开发人员更加轻易地开发和维护各种商业用户。

C#在语言中内置了版本控制功能,如函数重载必须被显式声明,不像C++可以随意被修改,从而防止了代码级错误和保留版本化的特性。

#### (5) 完整的安全性能与错误处理能力。

安全性与错误处理能力是衡量一种计算机语言是否优秀的重要依据。C#语言提供了完善的错误和异常处理机制,可以消除许多软件开发中的常见错误,并提供了包括类型安全在内的完整的安全性能。例如,C#中不能使用未初始化的变量;C#不支持不安全的指向,不能将整数指向引用类型;C#中提供了边界检查与溢出检查功能等。

为了减少开发中的错误,C#会帮助开发者通过更少的代码完成相同的功能,这不但减轻了编程人员的工作量,同时更有效地避免了错误的发生。

#### (6) 灵活性和兼容性。

在简化语法的同时,C#并没有失去灵活性。尽管它不是一种无限制的语言,比如它不能用来开发硬件驱动程序,在默认的状态下没有指针等,但如果需要,C#允许将某些类或者类的某些方法声明为非安全的。这样就能够使用指针、结构和静态数组,并且调用这些非安全代码不会带来任何其他的问题。此外,C#还提供了委托(delegate)来模拟指针的功能;C#不支持类的多继承,但是通过对接口的继承也可获得多继承功能。

C#遵守.NET 的公共语言规范 (Common Language Specification, CLS), 从而保证了 C# 组件与其他语言开发的组件兼容。

## 1.2 .NET 概述

.NET 是微软公司推出的软件开发平台, 以 Web services 为核心, 允许应用程序通过 Internet 进行通信和共享数据, 而不管所采用的是何种操作系统、设备或编程语言。

微软总裁兼首席执行官 Steve Ballmer 将 .NET 定义为: .NET 代表一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。也就是说, .NET=新平台+标准协议+统一开发工具。

我们通常所说的 .NET 通常指 .NET Framework、Visual Studio.NET 及开发出来的应用程序。

### 1.2.1 .NET 平台

.NET 是基于 Internet 的新一代开发平台。通过该平台, 可以创建和使用基于 XML 的应用程序、进程和 Web 站点以及服务, 使程序员在同一个开发环境下进行编码、编译和建立部署项目。

.NET 平台主要包括以下 5 个方面:

#### (1) 底层操作系统。

底层操作系统为 .NET 应用程序的开发提供软硬件支持, 如微软公司开发的几种操作系统 Windows XP、Windows 7、Windows 8、Windows 10、Windows Vista 等都可为 .NET 平台提供服务。

#### (2) .NET 企业服务器。

.NET 企业服务器可以为各类企业提供专业应用系统、知识管理、电子商务、业务电子化、无线互联网等多种解决方案, 满足了企业用户对系统的高可靠性、高扩展性、低成本、快速部署的要求。 .NET Enterprise Servers 是微软公司推出的进行企业集成和管理所有基于 Web 服务应用的系统产品, 包括 Mobile Information Server、SharePoint Portal Server 等, 为企业信息化和信息集成提供帮助。

#### (3) Microsoft XML Web 服务构件。

微软作为一个 Web 服务的底层技术提供商, 它主要提供一些公共性的 Web 服务, 包括身份认证、密码认证、个性化服务、软件传输等。Microsoft XML Web 服务构件提供了一系列高度分布、可编程的公用性网络服务, 它可以从任何支持 SOAP 的平台上进行访问, 也可以从内部局域网或以 Internet 的方式发布和访问。

#### (4) .NET Framework。

.NET Framework 是整个开发平台的基础, 它为运行于该平台上的应用程序提供环境, 允许不同的应用程序设计语言和库无缝结合共同创建基于 Windows 的应用程序, 管理、部署, 并与其他网络系统集成。

#### (5) .NET 开发工具。

.NET 开发工具包括 Visual Studio.NET 集成开发环境和 .NET 编程语言。Visual Studio.NET 是微软提供的一套完整的应用程序开发工具集, 在这套工具集中可以使用 Visual Basic、Visual

C++、Visual C#、Visual J#、Jscript.NET 等 .NET 编程语言进行开发。

### 1.2.2 什么是 .NET Framework

.NET Framework, 也称为 .NET 框架, 它是支持生成和运行下一代应用程序和 XML Web Services 的内部 Windows 组件, 很多基于此架构的程序需要它的支持才能够运行。

.NET Framework 主要为了实现以下几个目标:

(1) 提供一个一致的面向对象的编程环境, 而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, 或者是在远程执行的。

(2) 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。

(3) 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。

(4) 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。

(5) 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。

(6) 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。

### 1.2.3 .NET Framework 体系结构

.NET Framework 主要由公共语言运行库和 Microsoft .NET Framework 类库组成, 其体系结构如图 1-1 所示。

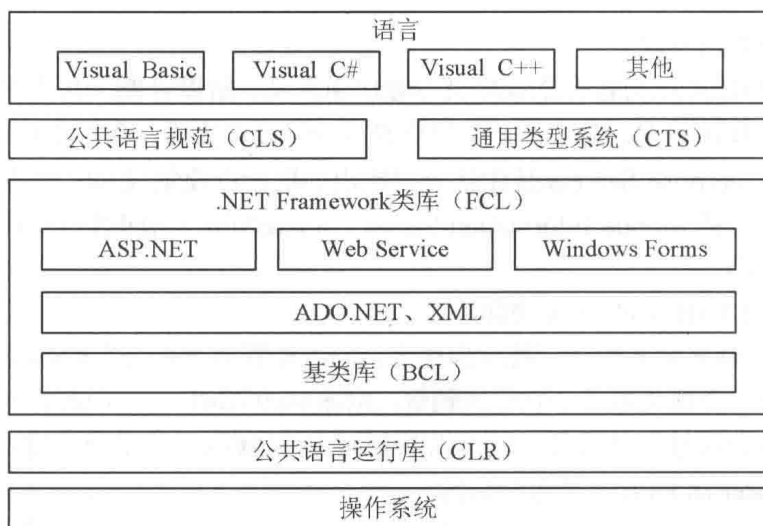


图 1-1 .NET Framework 体系结构

#### 1. 公共语言运行库

公共语言运行库 (Common Language Runtime, CLR) 是 Microsoft .NET Framework 的核心, 用于执行和管理任何针对 Microsoft .NET 平台的所有代码。

CLR 包括公共语言规范 (Common Language Specification, CLS) 和通用类型系统 (Common Type System, CTS)。

### (1) CLS。

各种编程语言除了在数据类型上不同外,在语法上也有非常大的区别,因此需要定义CLS。CLS 定义了所有编程语言必须遵守的共同标准,包含了函数调用方式、参数传递方式、数据类型和异常处理等。

CLS 是一个最低标准集,所有面向.NET 的编译器必须支持它。只有遵守这个标准编写的程序,才可能在只安装有.NET Framework 运行环境的计算机中运行,也才可以在.NET Framework 下实现互相操作。

### (2) CTS。

CTS 定义了一套可以在中间语言中使用的预定义数据类型,所有面向.NET Framework 的语言都可以生成最终基于这些类型的编译代码,即通用类型系统用于解决不同编程语言的数据类型不同的问题,从而实现跨语言功能。

## 2. .NET Framework 类库 (.NET Framework Class Library, FCL)

.NET Framework 类库是一个由类、接口和值类型组成的库,通过该库中的内容可访问系统功能。它是建立.NET Framework 应用程序、组件和控件的基础。

它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,这些应用程序包括传统的命令行或图形用户界面应用程序,也包括基于 ASP.NET 所提供的最新的应用程序(如 Web Services)。

## 3. 基类库 (Base Class Library, BCL)

除 CLR 和 CTS/CLS 规范之外,.NET 平台提供了一个适用于全部.NET 程序语言的基类库。这个基类库不仅封装了各种基本类型,如线程、文件输入/输出(I/O)、图形绘制以及与各种外部硬件设备的交互,还支持在实际应用中用到的一些服务。

## 4. ADO.NET 和 XML

ADO.NET (ActiveX Data Objects for the .NET Framework) 是.NET 框架中的一组类和工具的集合,可用于创建强大的、灵活可靠的数据驱动的应用程序。ADO.NET 的各个类位于 System.Data.dll 中,并且与 System.Xml.dll 中的 XML 类相互集成。

ADO.NET 提供对 Microsoft SQL Server、Oracle 等数据源以及通过 OLE DB 和 XML 公开的数据源的一致访问。数据共享使用者可以通过 ADO.NET 来连接这些数据源,并检索、操作和更新数据库中的数据。

XML 是可扩展标记语言,它与 Access、Oracle 和 SQL Server 等数据库不同,数据库提供了更强有力的数据存储和分析能力,而 XML 仅仅是展示数据,XML 的简单性使其易于在任何应用程序中读写数据。

ADO.NET 和 XML 类提供了一种统一的中间 API,程序员可通过同步的双编程接口来使用它。可以使用 XML 基于节点的分层法或基于列的表格式数据集关系法来访问和更新数据,同时,也可以在任何时间从数据的数据集表示形式切换到 XML 文档对象模型。

## 5. Windows Forms

Windows Forms (Windows 窗体) 是微软的.NET 开发框架的图形用户界面部分,它使用了许多新技术,包括一个公共应用程序框架、受控的执行环境、集成的安全性和面向对象的设计原则。

Windows Forms 完全支持快速、容易地连接 XML 网络服务和在 ADO.NET 数据模型基础

上创建丰富的、数据感知的应用程序。开发人员可以使用任何支持.NET 平台的语言,包括 Microsoft Visual Basic 和 C#创建 Windows 窗体应用程序。

## 6. Web Service

Web Service (Web 服务) 是一种通过 HTTP 使用 XML 接收请求和数据的应用程序。

Web Service 的主要目标是跨平台的可互操作性。为了实现这一目标, Web Service 完全基于 XML、XSD (XML 结构定义) 等独立于平台、独立于软件供应商的标准, 是创建可互操作的、分布式应用程序的新平台。

### 1.2.4 .NET 程序执行过程

托管代码 (Managed Code) 是中间语言代码, 在公共语言运行库 (CLR) 控制下运行。托管代码应用程序可以获得公共语言运行库服务, 例如自动垃圾回收、运行库类型检查和安全支持等。

创建执行一个.NET 程序分以下 3 个步骤:

(1) 当用户编译一个.NET 程序时, 需要把源代码编译成与机器无关的 Microsoft 中间语言 (MSIL), 中间语言被封装在一个叫程序集 (assembly) 的文件中, 程序集中包含了所创建的类、方法和属性的所有元数据, 如图 1-2 所示。

(2) 由于 MSIL 独立于特定的 CPU, 不能直接被机器执行, 因此必须由 CLR 中的 JIT 编译器将 MSIL 翻成本机代码, 因为 JIT 编译器能确定程序运行在什么类型的处理器上, 可以利用该处理器提供的特性或特定的机器指令来优化最后的可执行代码, 如图 1-3 所示。



图 1-2 .NET 编译器编译



图 1-3 JIT 编译器编译

(3) 在托管 CLR 环境中运行本机代码以及其他应用程序, 如图 1-4 所示。

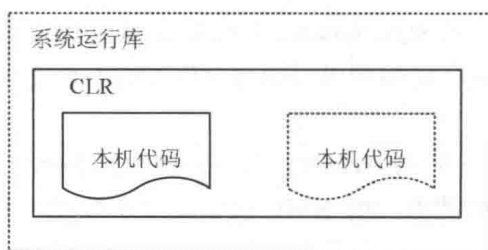


图 1-4 运行本机代码

## 1.3 Visual Studio 2017 集成开发环境

Visual Studio 是一套完整的开发工具集, 可以生成 ASP.NET Web 应用程序、Web 服务应用程序、Windows 应用程序和移动设备应用程序。它为 Visual Basic、Visual C++、Visual C# 等提供了统一的集成开发环境 (IDE), 在 IDE 中可以共享工具和创建混合语言解决方案。目前最新版本为 Visual Studio 2017, 基于 .NET Framework 4.5.2。

Visual Studio 2017 (简称 VS 2017) 可支持 C#、C++、Python、Visual Basic、Node.js、HTML、JavaScript 等各大编程语言, 不仅可编写 Windows 10 UWP 通用程序, 还能开发 iOS、Android 移动平台应用。

VS 2017 集成开发环境如图 1-5 所示。

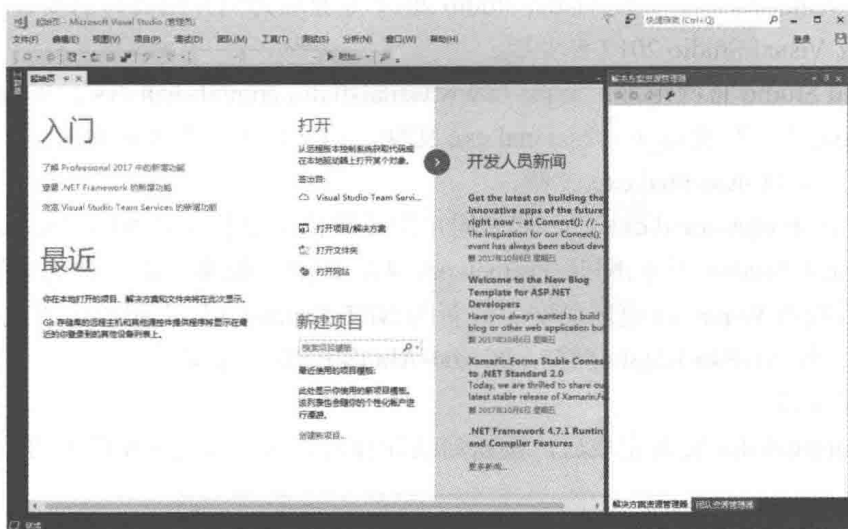


图 1-5 VS 2017 集成开发环境

### 1.3.1 Visual Studio 2017 的安装

Visual Studio 2017 有 3 个版本: Visual Studio 2017 社区版、Visual Studio 2017 专业版和 Visual Studio 2017 企业版。

社区版是适用于学生、开放源代码和个体开发人员的免费、全功能型 IDE。

专业版和企业版功能强大, 可供专业用户和企业购买使用。企业版是可以满足任何规模团队的生产效率和协调性需求的 Microsoft Davos 解决方案。而对于专业用户来说, 可能用不到企业版那样全面的功能, 但社区版又不能满足需求, 所以专业版中提供的专业开发者工具、服务和订阅就成了最佳选择。

#### 1. 安装 Visual Studio 2017 的系统要求

##### (1) 硬件。

- 处理器: 频率在 1.8 GHz 或以上, 推荐使用双核或更好的内核。
- 系统内存: 2 GB 或以上, 建议采用 4 GB (如果在虚拟机上运行, 则最低 2.5 GB)。
- 硬盘空间: 1 GB~40 GB, 具体取决于安装的功能。
- 显示器: 视频卡支持最小显示分辨率 1280×720, Visual Studio 最适宜的分辨率为 WXGA (1366×768) 或更高。

##### (2) 支持的操作系统。

Visual Studio 2017 可在以下操作系统上安装并运行:

- Windows 10 1507 版或更高版本: 家庭版、专业版、教育版和企业版 (不支持 LTSB 和 Windows 10 S)。
- Windows Server 2016: Standard 和 Datacenter。

- Windows 8.1 (带有更新 2919355): 核心版、专业版和企业版。
- Windows Server 2012 R2 (带有更新 2919355): Essentials、Standard、Datacenter。
- Windows 7 SP1 (带有最新 Windows 更新): 家庭高级版、专业版、企业版、旗舰版。

## 2. 安装步骤

下面以在 Windows 7 上安装 Visual Studio 2017 专业版为例介绍具体的安装步骤。

### (1) 下载 Visual Studio 2017 专业版。

进入 Visual Studio 官网地址: <https://www.visualstudio.com/zh-hans/vs/>, 单击“下载 Visual Studio\Professional”, 下载 vs\_Professional.exe 文件, 该文件大小为 1.02MB。

### (2) 运行 vs\_Professional.exe 文件。

运行文件 vs\_Professional.exe 后, 如果用户的计算机上没有安装 .NET Framework 4.6, 系统会提示“Visual Studio 要求 .NET Framework 4.6 或更高版本。请从此处安装最新的 .NET Framework, 或检查 Windows 更新以安装最新的 .NET Framework。”可单击“此处”链接到相应页面, 下载文件 NDP46-KB3045557-x86-x64-AllOS-ENU 并安装。

### (3) 在线安装。

.NET Framework 4.6 安装完成后, 重新启动计算机, 进入如图 1-6 所示的界面。

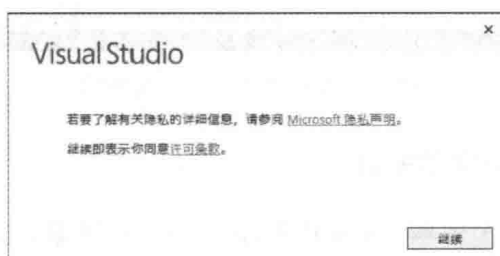


图 1-6 继续界面

单击“继续”按钮, 进入安装设置界面, 如图 1-7 所示。

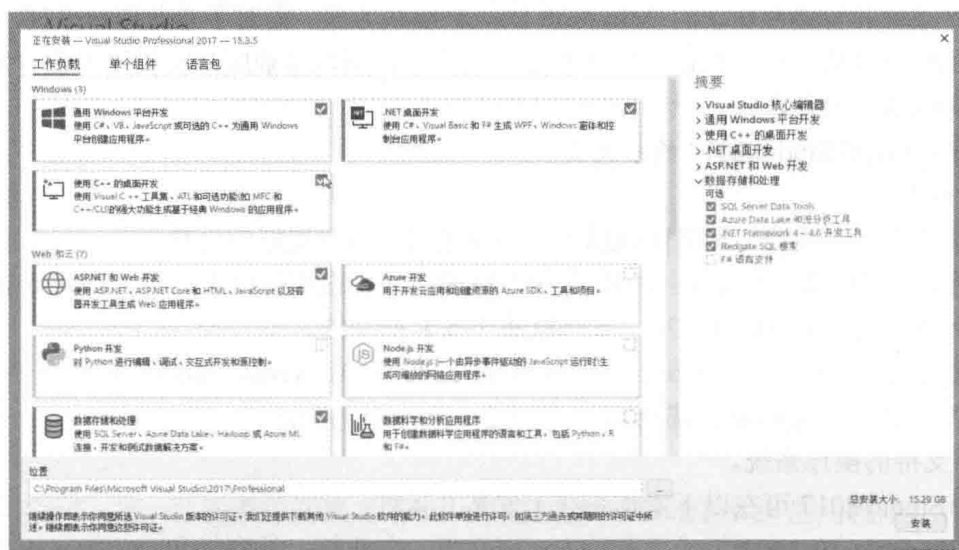


图 1-7 安装设置