

C#

程序设计案例教程

向燕飞 主 编
魏菊霞 彭之军 副主编



+



+



+



- ◆ 以基础理论—实用技术—实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置了“小型案例实训”，旨在培养学生的实践能力
- ◆ 配备丰富的免费教学资源——电子课件、习题答案



向燕飞 主 编

魏菊霞 彭之军 副主编

清华大学出版社
北 京

内 容 简 介

本书采用“案例引导，任务驱动”的编写方式，深入浅出地讲解了C#程序设计的基本方法。

本书重点讲解C#程序设计语言的基本语法和常用方法，去掉一些不常用的语法细节，采用语法与实例相结合的形式，将C#基本语言以及各种编程对象融合到具体实例中，侧重培养学生解决实际编程问题的能力，同时又可增强学生对C#编程思想的理解。

全书分为18个单元，涵盖四大专题，包括C#语言基础、Windows编程、数据访问、网络编程。每个单元的案例按照“案例描述→知识链接→案例分析与实现→拓展实训→习题”的顺序编写，实例丰富，内容详尽，难易适中，重点突出，降低了读者学习的难度。

本书适合作为高等院校计算机专业学生的教材使用，也可作为自学C#语言程序设计的参考用书。

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

C#程序设计案例教程/向燕飞主编. —北京：清华大学出版社，2018（2018.8重印）

（全国高等院校应用型创新规划教材·计算机系列）

ISBN 978-7-302-48564-3

I. ①C… II. ①向… 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(2017)第241123号

责任编辑：梁媛媛

封面设计：杨玉兰

责任校对：宋延清

责任印制：沈 露

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>, 010-62791865

印装者：北京鑫海金澳胶印有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：22 字 数：532千字

版 次：2018年1月第1版 印 次：2018年8月第2次印刷

定 价：49.80元

产品编号：075893-01

前言

首先，感谢您翻阅这本书。

本书的背景

在 .NET 被正式发布之前，NGWS 这个术语一直被用于微软的某些计划，这些计划旨在创建一个下一代视窗服务的基于因特网的平台(Internet-based platform of Next Generation Windows Services)。

史蒂夫·鲍尔默于 2000 年指出：“交付一个基于因特网的下一代视窗服务平台是我们公司的重中之重。我们在此谈论的突破包括对编程模型、用户界面、应用程序整合模型、文件系统、新的 XML Schema 等的改变。”

作为微软最重要的开发者工具集，新版 Visual Studio 2017 旨在帮助开发人员围绕微软核心产品和服务打造高效完美的解决方案。当前 Visual Studio 2017 已经扩展到 Mac 平台，这意味着开发者可以在 Mac 上使用 C#、F#、.NET Core、ASP.NET Core、Xamarin 和 Unity 来搭建应用程序。而 C#正是.NET 平台主流的开发语言，也是一个现代的、通用的、面向对象的编程语言。因此，它的应用范围十分广泛。

本书由作者将过去的教学经验和实验指导经验整理而成，主要针对大学本科或专科相关专业学生的程序设计学习，亦可供相关的开发人员及爱好者入门学习使用。

本书的特色

编程属于利用抽象概念来思维，通常可以通过简单的例程形象地进行学习。本书采用“案例描述→知识链接→案例分析与实现→拓展实训→习题”这样的编写体例。通过精心选取案例，一改传统案例的枯燥性，挑选更符合应用，而且生动有趣的案例帮助读者提高学习效率。在每个单元中，针对初学者容易忽略、出错，或难于理解的知识点及代码块，采用注意、说明等特别形式重点提醒读者。

本书的组织

在学习面向对象技术的过程中，既需要学习编程的抽象概念，又需要培养编程的实际技能，二者都是非常重要的。学习理论知识时，要防止沉迷于语言细节。如果过多地关注细节，就会分散注意力，导致对某些概念“知其然，而不知其所以然。”

本书面向爱好 C#的零基础读者，全面讲解 C#的基础知识。在实例的选取上注重实用性，内容详尽，难易适中，重点突出，降低了读者学习的难度。

本书涵盖四大专题，包括 C#语言基础、Windows 编程、数据访问、网络编程，共 18



个单元。

第一篇 C#语言基础

包括 1~12 单元。

单元 1 从 C#的基础知识开始进行简要的介绍,其中包括 .NET Framework、Common Language Runtime 以及 C#的特点等。该单元的目的,是让读者快速了解 C#的基本概念,让读者对 C#有一个基本的认识。

单元 2 全面介绍 C#中的数据类型,以及数据类型间的转换。

程序中变量的值不断变化,最终产生人们想要的处理结果,而变量值的变化是通过运算符和表达式来实现的。在单元 3 中,详细阐述表达式中每种运算符的功能、优先级、结合性及在使用中的注意事项。

在实际的任务中,大多数问题的求解步骤(也就是通常所说的算法)往往都会有若干分支选项或重复执行的情况。所以单元 4 讲解选择结构程序设计,单元 5 讲解循环结构的程序设计。

随着程序代码的增多,任何软件都难免会出错,在软件开发的过程中,进行错误捕捉显得尤为重要,因为有的错误会导致软件功能失常,有的错误甚至会造成破坏性损失。在单元 6 中,主要讲解异常处理方法和调试方法。

有时需要存储多个相同类型的值,这就要用到单元 7 中介绍的数组和集合的知识。

C#是面向对象的语言,所以在单元 8~12 中,全面介绍面向对象的知识,包括类和对象的概念及其使用、类的属性和方法、类的继承与多态性(虚方法和重写、抽象类、接口)、委托与事件、泛型程序设计。

第二篇 Windows 编程

单元 13 是 Windows 编程基础,单元 14 是 Windows 编程进阶。这部分结合一些 Windows 应用程序实例,介绍最常用控件的属性、方法、事件及应用,让读者对 Windows 应用程序的设计有进一步的了解和认识。

第三篇 数据访问

主要分为 ADO.NET 数据访问和磁盘文件的读写。

ADO.NET 是一组用于和数据源进行交互的面向对象类库,因此,要掌握信息管理系统软件的开发,就必须掌握 ADO.NET 数据库的编程。单元 15~16 以简单通俗的例子,阐述 C#语言中进行数据库编程的几个基本核心对象。

文件是系统的重要组成部分。在物联网系统中,如何将数据永久保存为文件?如何读取已有的数据文件?单元 17 介绍涉及到的文件读写技术。

第四篇 网络编程

计算机技术发展到现在,从应用服务器到 PC 机,再到手持设备,几乎都要通过网络连接起来。现在的应用软件,大都需要通过网络来进行通信。所以在单元 18 中,简单地引入 C#在网络编程中的应用,而更多的网络编程技术将在其他教材和课程里深入学习。感兴趣的读者也可以自学。

本书所有案例均在 Visual Studio 2017 环境下编译通过。

由于书籍篇幅有限,通常不能将代码完全列出。强烈建议读者多编写代码实践,可以运行、调试、修改、补充各个案例与练习。编程技能的训练目的是掌握足够多的语言细节以便完成有意义的程序。而真实软件开发项目中的代码,则需要更为周全的考虑,例如统一的编程风格、完善的注释和文档、各种修饰符的选择、方法的参数检查、完整的异常处理和防御编程、有弹性的类层次设计等,所有这些,都需要在编程练习中逐步学会。

本书由向燕飞担任主编,魏菊霞、彭之军担任副主编。具体分工是:第1~14单元由向燕飞编写,15~16单元由魏菊霞编写,17~18单元由彭之军编写。在出版过程中,我们得到了所在学校领导、同事以及出版社编辑的帮助和指教,特别是奚建清教授、曹琳教授对本书的编写提出了许多宝贵意见。谨向他们表示衷心感谢。

由于作者水平有限,书中的错误和不妥之处在所难免,希望读者批评指正。有关本书的意见反馈和咨询,读者可在清华大学出版社网站的相关版块中与作者进行交流。

向燕飞

1.1.5 C#程序结构	12	1.3.3 案例拓展	43
1.1.6 命名空间	13	1.4 拓展训练:判断计算结果	44
1.2.7 程序的运行与调试	13	习题三	44
1.3 案例拓展:计算水仙花数	14		
1.3.1 案例拓展	14	单元4 顺序和选择结构程序的设计	45
1.3.2 案例拓展	15	4.1 案例描述	47
习题一	16	4.2 知识链接	47
单元2 变量与数据类型	17	4.2.1 程序结构	48
2.1 案例描述	18	4.2.2 顺序文	49
2.2 知识链接	18	4.2.3 switch分支	51
2.2.1 数据类型	18	4.3 案例拓展:判断奇偶	55
2.2.2 常量与变量	25	4.3.1 案例描述	56
2.2.3 数据类型间的转换	28	4.3.2 案例拓展	56
2.2.4 DateTime	30	4.4 拓展训练:判断价格计算	57
2.3 案例拓展:大数	32	习题四	58
2.3.1 案例拓展	32	单元5 循环结构程序设计	60
2.3.2 案例拓展	32	5.1 案例描述	61
2.4 知识链接	34	5.2 知识链接	61
2.4.1 拓展训练1:使用变量存储	34	5.2.1 循环结构概述	61
2.4.2 拓展训练2:数字加密器	34	5.2.2 while循环	63
习题二	35	5.2.3 do-while 循环	65
单元3 运算符和表达式	36	5.2.4 for 循环	66
3.1 案例描述	37	5.2.5 循环跳转:continue 语句	68
3.2 知识链接	37	5.2.6 提前结束循环:break 语句	69
		5.3 案例拓展与实训	70
		5.3.1 案例拓展	70
		5.3.2 案例拓展	70

目录

单元 1 我的第一个 C#程序.....1

1.1 案例描述.....2

1.2 知识链接.....2

1.2.1 .NET 概述.....2

1.2.2 .NET Framework.....3

1.2.3 开发环境和 C#语言.....4

1.2.4 使用 Visual Studio 创建项目.....9

1.2.5 C#程序结构.....12

1.2.6 命名空间.....13

1.2.7 程序的运行与调试.....13

1.2.8 了解 MSDN.....14

1.3 案例分析与实现.....15

1.3.1 案例分析.....15

1.3.2 案例实现.....15

习题一.....16

单元 2 变量与数据类型.....17

2.1 案例描述.....18

2.2 知识链接.....18

2.2.1 数据类型.....18

2.2.2 变量与常量.....25

2.2.3 数据类型间的转换.....28

2.2.4 DateTime.....30

2.3 案例分析与实现.....32

2.3.1 案例分析.....32

2.3.2 案例实现.....32

2.4 拓展训练.....34

2.4.1 拓展实训 1: 使用变量存储

一部手机的信息.....34

2.4.2 拓展实训 2: 数字加密器.....34

习题二.....35

单元 3 运算符和表达式.....36

3.1 案例描述.....37

3.2 知识链接.....37

3.2.1 算术运算符与算术表达式.....38

3.2.2 赋值运算符与赋值表达式.....40

3.2.3 关系运算符与关系表达式.....40

3.2.4 逻辑运算符与逻辑表达式.....40

3.2.5 条件运算符与条件表达式.....41

3.2.6 运算符的优先级与结合顺序.....42

3.3 案例分析与实现.....42

3.3.1 案例分析.....42

3.3.2 案例实现.....43

3.4 拓展训练: 判断计算是否正确.....44

习题三.....44

单元 4 顺序和选择结构程序的设计.....46

4.1 案例描述.....47

4.2 知识链接.....47

4.2.1 顺序结构.....48

4.2.2 if 分支.....49

4.2.3 switch 分支.....53

4.3 案例分析与实现.....56

4.3.1 案例分析.....56

4.3.2 案例实现.....56

4.4 拓展训练: 旅游价格计算.....57

习题四.....59

单元 5 循环结构程序设计.....60

5.1 案例描述.....61

5.2 知识链接.....61

5.2.1 循环结构概述.....61

5.2.2 while 循环.....63

5.2.3 do-while 循环.....65

5.2.4 for 循环.....66

5.2.5 循环跳转: continue 语句.....68

5.2.6 提前结束循环: break 语句.....69

5.3 案例分析与实现.....70

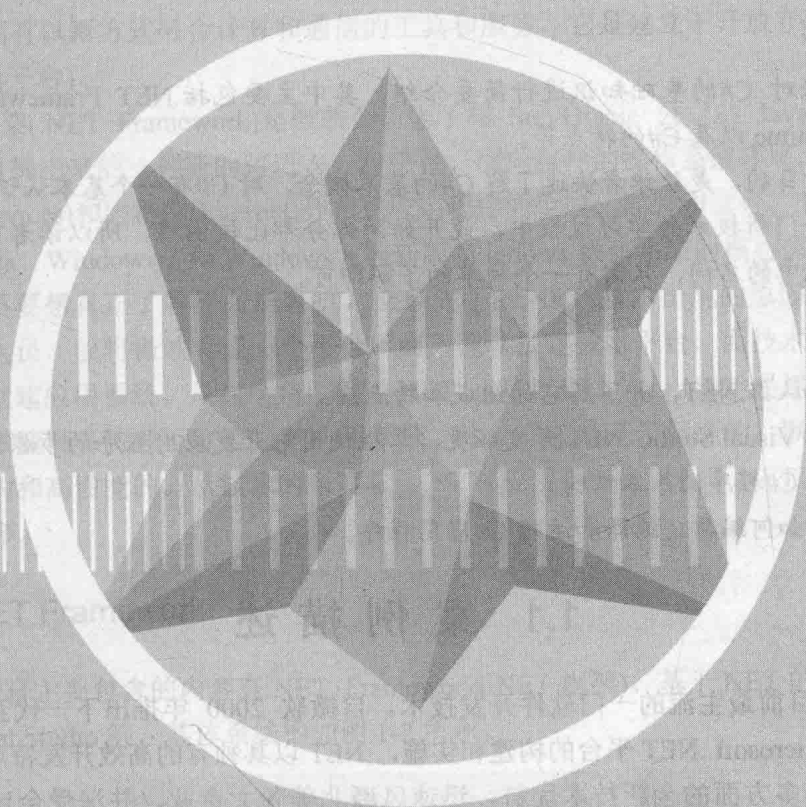
5.3.1 案例分析.....70

5.3.2 案例实现.....70

5.4 拓展训练.....	73	8.2.2 类的声明和成员组织以及访问 修饰符.....	111
5.4.1 拓展实训 1: 直到型循环.....	73	8.2.3 创建类实例.....	114
5.4.2 拓展实训 2: 嵌套循环.....	73	8.2.4 类的构造函数和析构函数.....	115
习题五.....	74	8.3 案例分析与实现.....	119
单元 6 程序调试与异常处理	76	8.3.1 案例分析.....	119
6.1 案例描述.....	77	8.3.2 案例实现.....	120
6.2 知识链接.....	78	8.4 拓展训练.....	121
6.2.1 异常的概念.....	78	8.4.1 拓展实训 1: 定义长方 体类.....	121
6.2.2 Visual Studio 中的调试方法.....	79	8.4.2 拓展实训 2: 识别静态 成员.....	122
6.2.3 异常处理.....	84	习题八.....	123
6.3 案例分析与实现.....	87	单元 9 类的方法和属性	125
6.3.1 案例分析.....	87	9.1 案例描述.....	126
6.3.2 案例实现.....	87	9.2 知识链接.....	127
6.4 拓展训练: 创建用户自定义异常.....	89	9.2.1 方法的定义.....	127
习题六.....	90	9.2.2 方法的调用.....	128
单元 7 数组与集合	92	9.2.3 方法中的参数传递.....	130
7.1 案例描述.....	93	9.2.4 方法重载.....	135
7.2 知识链接.....	94	9.2.5 属性.....	137
7.2.1 数组的定义和使用.....	94	9.3 案例分析与实现.....	139
7.2.2 用 foreach 语句遍历数组.....	97	9.3.1 案例分析.....	139
7.2.3 数组的常用排序算法.....	97	9.3.2 案例实现.....	141
7.2.4 集合类.....	99	9.4 拓展训练: 设计一个类的定义和 封装.....	143
7.3 案例分析与实现.....	103	习题九.....	145
7.3.1 案例分析.....	103	单元 10 类的继承与多态性	147
7.3.2 案例实现.....	104	10.1 案例描述.....	148
7.4 拓展训练.....	105	10.2 知识链接.....	149
7.4.1 拓展实训 1: 二维数组的 使用.....	105	10.2.1 类的继承.....	149
7.4.2 拓展实训 2: 利用 Array 进行 排序.....	106	10.2.2 多态性: 虚方法和重写.....	155
习题七.....	106	10.2.3 多态性: 抽象类.....	159
单元 8 类和对象	108	10.2.4 多态性: 接口.....	161
8.1 案例描述.....	109	10.3 案例分析与实现.....	164
8.2 知识链接.....	109	10.3.1 案例分析.....	164
8.2.1 面向对象的基本思想.....	109		

10.3.2 案例实现.....	165	13.2.7 Label 控件.....	217
10.4 拓展训练: 从接口继承.....	167	13.2.8 LinkLabel 控件.....	218
习题十.....	170	13.2.9 TextBox 控件.....	219
单元 11 委托与事件	171	13.2.10 PictureBox 控件.....	220
11.1 案例描述.....	172	13.2.11 Timer 控件.....	221
11.2 知识链接.....	172	13.2.12 GroupBox 控件.....	222
11.2.1 委托和方法.....	172	13.2.13 Panel 控件.....	222
11.2.2 事件处理.....	176	13.2.14 RadioButton 控件.....	222
11.3 案例分析与实现.....	178	13.2.15 CheckBox 控件.....	224
11.3.1 案例分析.....	178	13.2.16 TabControl 控件.....	224
11.3.2 案例实现.....	180	13.3 案例分析与实现.....	226
11.4 拓展训练: 使用委托和事件.....	182	13.3.1 案例分析.....	226
习题十一.....	184	13.3.2 案例实现.....	226
单元 12 泛型	186	13.4 拓展训练: 简易计算器的设计与实现.....	228
12.1 案例描述.....	187	习题十三.....	232
12.2 知识链接.....	187	单元 14 Windows 编程进阶	233
12.2.1 泛型的引入.....	187	14.1 案例描述.....	234
12.2.2 使用泛型类.....	190	14.2 知识链接.....	234
12.2.3 泛型方法.....	192	14.2.1 RichTextBox 控件.....	234
12.2.4 泛型约束.....	193	14.2.2 TreeView 控件.....	236
12.3 案例分析与实现.....	196	14.2.3 MenuStrip 控件.....	240
12.3.1 案例分析.....	196	14.2.4 ToolStrip 控件.....	242
12.3.2 案例实现.....	197	14.2.5 StatusStrip 控件.....	244
12.4 拓展训练: 使用 List<T>类.....	198	14.2.6 OpenFileDialog 控件.....	244
习题十二.....	200	14.2.7 SaveFileDialog 控件.....	245
单元 13 Windows 编程基础	201	14.3 案例分析与实现.....	246
13.1 案例描述.....	202	14.3.1 案例分析.....	246
13.2 知识链接.....	202	14.3.2 案例实现.....	246
13.2.1 可视化编程原理与技巧.....	202	14.4 拓展训练: 摇奖程序的设计与实现.....	248
13.2.2 Windows 窗体.....	205	习题十四.....	252
13.2.3 控件的常用属性、方法和事件.....	209	单元 15 创建数据库连接与数据操作命令	254
13.2.4 Button 控件.....	213	15.1 案例描述.....	255
13.2.5 ListBox 控件.....	213	15.2 知识链接.....	256
13.2.6 ComboBox 控件.....	217		

15.2.1 数据库基础.....	256	16.4 拓展训练.....	294
15.2.2 ADO.NET 基础.....	258	16.4.1 拓展实训 1: 销售前台.....	294
15.2.3 用 Connection 对象连接数据库.....	260	16.4.2 拓展实训 2: 三层架构的设计与实现.....	300
15.2.4 用 Command 对象执行命令.....	262	习题十六.....	308
15.2.5 用 DataReader 对象读取数据.....	263	单元 17 C#读写文件	309
15.3 案例分析与实现.....	265	17.1 案例描述.....	310
15.3.1 案例分析.....	265	17.2 知识链接.....	310
15.3.2 案例实现.....	265	17.2.1 System.IO 命名空间介绍.....	310
15.4 拓展训练.....	270	17.2.2 File 类的重要方法.....	311
15.4.1 拓展实训 1: 绑定下拉框数据.....	270	17.2.3 文本文件的读写.....	313
15.4.2 拓展实训 2: 用户注册.....	274	17.2.4 二进制文件的读写.....	315
习题十五.....	276	17.3 案例分析与实现.....	316
单元 16 数据集 DataSet 与数据绑定组件	278	17.3.1 案例分析.....	316
16.1 案例描述.....	279	17.3.2 案例实现.....	317
16.2 知识链接.....	280	17.4 拓展训练: 图片查看器.....	318
16.2.1 数据集 DataSet.....	280	习题十七.....	321
16.2.2 数据适配器 DataAdapter.....	281	单元 18 网络编程概述	322
16.2.3 数据表 DataTable.....	282	18.1 案例描述.....	323
16.2.4 命令生成器 Command Builder.....	282	18.2 知识链接.....	323
16.2.5 DataGridView 控件.....	284	18.2.1 Socket 的使用.....	324
16.3 案例分析与实现.....	287	18.2.2 Socket 连接实例.....	324
16.3.1 案例分析.....	287	18.3 案例分析与实现.....	329
16.3.2 案例实现.....	287	18.3.1 案例分析.....	329
		18.3.2 案例实现.....	329
		18.4 拓展训练: 基于 TCP 编程的实例..	334
		习题十八.....	338
		参考文献	339



单元 1

我的第一个 C#程序

单元导读

本单元将对 C# 的基础知识进行简要介绍, 其中主要包括 .NET Framework、Common Language Runtime 以及 C# 的特点等。

本单元的目的, 是让读者快速了解 C# 的基本概念, 对 C# 有一个基本认识。

在任何一门新技术的学习过程中, 最开始的部分都比较困难, 所以读者可以不必过分纠结于本单元中的名词, 只需有一个简单的了解即可。

学习目标

- 初步认识 .NET, 并了解它的组成和特点。
- 熟悉 Visual Studio .NET 开发环境, 掌握使用它开发应用程序的步骤。
- 学习 C# 程序的基本结构。
- 掌握如何编辑、编译和运行 C# 应用程序。

1.1 案例描述

.NET 是目前最主流的一门软件开发技术。自微软 2000 年推出下一代互联网构想以来, 伴随着 Microsoft .NET 平台的构建和实施, .NET 以其独有的高效开发特点、简单易行的版本控制等多方面的全新技术优势, 迅速风靡北美各大企业, 并深受全球开发者的喜爱。NET 是世界上最大的软件公司——微软花费 300 亿美元精心打造的开发平台, 可以开发 Web 程序、Windows 应用程序和 WAP 无线网络应用程序等, 其在大型系统开发中的份额越来越重。

本案例中, 我们想要初步了解 C# 和 .NET。将编写一个应用程序, 能根据用户输入的名字, 显示 “Hello, XXX, 欢迎来到 C# 的世界!” 欢迎词。执行结果如图 1-1 所示。



图 1-1 第一个程序

1.2 知识链接

1.2.1 .NET 概述

微软对 .NET 的定义是: .NET is a revolutionary new platform, built on open Internet protocols and standards, with tools and services that meld computing and communications in new

ways(.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)。

.NET 框架(.NET Framework)是微软公司为了与 Sun/Oracle 公司的 Java(J2EE)竞争,于 2000 年 6 月提出来的一种新的跨语言、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 框架也是 Windows Vista、Windows 7 和 Windows 8 等新版 Windows 操作系统的核心部件。

技术人员要想真正了解什么是 .NET,必须先了解 .NET 技术出现的原因和它想解决的问题。技术人员一般将微软看成一个平台厂商。微软搭建技术平台,而技术人员在这个技术平台之上创建应用系统。从这个角度看,.NET 也可以定义为:.NET 是微软的新一代技术平台,为敏捷商务构建互联互通的应用系统,这些系统是基于标准的、联通的、适应变化的、稳定的和高性能的。从技术的角度看,.NET 应用是一个运行于 .NET Framework 之上的应用程序。

1.2.2 .NET Framework

.NET 平台主要包含的内容有 .NET Framework(.NET 框架)、基于 .NET 的编程语言及开发工具 Visual Studio 等,其体系结构如图 1-2 所示。

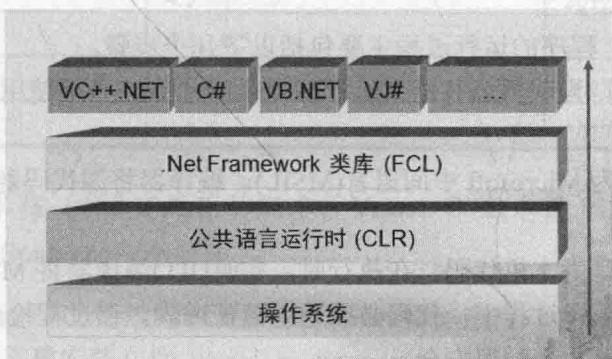


图 1-2 .NET 的体系结构

.NET 平台的基础和核心是 .NET Framework,.NET 平台的各种优秀特性都要依赖它来实现。.NET Framework 包括两部分内容:一是框架类库(Framework Class Library, FCL);二是公共语言运行时(Common Language Runtime, CLR),也译作“公共语言运行库”。

1. FCL(框架类库)

从图 1-2 中可以看出,在 .NET 平台上可以使用 C#、VB.NET 等多种语言来编写程序,不同的语言可以使用相同的 FCL。

FCL(Framework Class Library, 框架类库)为开发人员定义并提供了统一的、面向对象的、分层的和可扩展的类库集,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。从 JScript、Visual Basic 到 Visual C++、C#、F#的所有编程语言(通过托管扩展)具有

对框架的相似访问，开发人员可以自由选择它们要使用的语言。

相对于贫乏的 C++ 类库和丰富的 Java 类库，.NET 框架类库非常庞大，包含数百个命名空间、数千个类、接口和值类型。该库提供对系统功能的访问，是建立 .NET 框架应用程序、组件和控件的基础。框架类库采用命名空间来组织和使用，如 .NET 4.5 版的类库提供了 588 个命名空间，其中用于应用编程的 BCL(System 型)命名空间有 407 个。

2. CLR(公共语言运行时)

与 Java 虚拟机(Java Virtual Machine, JVM)相似，CLR 也是一个运行时环境。CLR 负责内存分配和垃圾回收，也就是通常所说的资源分配，同时保证应用和底层系统的分离。总而言之，它负责 .NET 库所开发的所有应用程序的执行。

CLR 所负责的应用程序在执行时是托管的。托管代码带来的好处是跨语言调用、内存管理、安全性处理等。CLR 隐藏了一些与底层操作系统打交道的环节，使开发人员可以把注意力放在代码所实现的功能上。非 CLR 控制的代码即非托管(unmanaged)代码，如 C++ 等，这些语言可以访问操作系统的功能，直接进行硬件操作。

垃圾回收(Garbage Collection)是 .NET 中一个很重要的功能，尽管这种思想在其他语言中也有实现。这个功能保证应用程序不再使用某些内存时，这些内存就会被 .NET 回收并释放。这种功能被实现以前，这些复杂的工作主要由开发人员来实现，而这正是导致程序不稳定的主要因素之一。

一个典型的 .NET 程序的运行过程主要包括以下几个步骤。

(1) 选择编译器。为获得公共语言运行库提供的优点，必须使用一个或多个针对运行库的语言编译器。

(2) 将代码编译为 Microsoft 中间语言(MSIL)。编译器将源代码翻译为 MSIL 并生成所需的元数据。

(3) 将 MSIL 编译为本机代码。在执行时，实时(JIT)编译器将 MSIL(微软中间语言)翻译为本机代码。在此编译过程中，代码必须通过验证过程，该过程检查 MSIL 和元数据以查看是否可以将代码确定为类型安全。

(4) 运行代码。公共语言运行库提供使执行能够发生以及可在执行期间使用的各种服务的结构。

1.2.3 开发环境和 C#语言

1. Visual Studio 2017

Visual Studio 是目前最流行的 Windows 平台应用程序的集成开发环境，最新的版本是 Visual Studio 2017，其官方发布时给出的“*What's new in VS 2017*”如图 1-3 所示，其特点是适用于开发 Android、iOS、Windows、Web 和云的应用，支持创建 .NET Core 程序，可以选择 .NET Core 1.0 或 .NET Core 1.1。

本书的所有范例是在 Visual Studio 2017 Professional 环境下调试的。

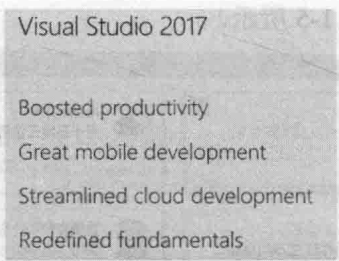


图 1-3 VS 2017 的新特性

VS(C#)不同的版本如表 1-1 所示。

表 1-1 Visual Studio、.NET Framework 和 C#版本

名 称	.NET Framework 版本	发布日期	C#版本
Visual Studio 2002	1.0	2002-02-13	1.0
Visual Studio 2003	1.1	2003-04-24	1.5
Visual Studio 2005	2.0	2005-11-07	2.0
Visual Studio 2008	3.5	2007-11-19	3.0
Visual Studio 2010	4.0	2010-04-12	4.0
Visual Studio 2012	4.5	2012-08-25	5.0
Visual Studio 2013	4.5、4.5.1	2013-10-17	5.0
Visual Studio 2015	4.6	2015-07-21	6.0
Visual Studio 2017	4.7	2017-03-07	7.0

 **说明：** C#版本指 C#语言规范的版本；Visual C#版本是开发工具的版本，它是 Visual Studio 的一个组件。

安装 Visual Studio 2017 的过程如下。

(1) 从微软网站下载安装文件 `visual_studio_professional_2017_x86_x64.exe`，只有不到 1MB 大小。这只是一个引导程序(Web Installer)，启动之后，将看到安装界面变得更可视化了，总共有三个选项卡。

① “工作负载”选项卡。工作负载总共分 3 类，拥有 13 个具体选项。
Windows 类(共 3 项)，如图 1-4 所示。

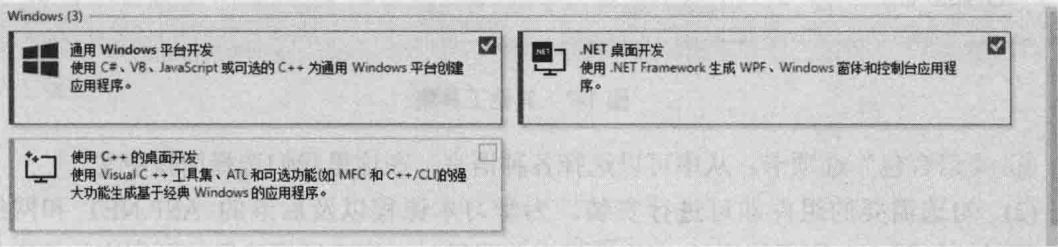


图 1-4 Windows 类

Web 和云类(共 5 项), 如图 1-5 所示。



图 1-5 Web 和云类

移动与游戏类(共 5 项), 如图 1-6 所示。

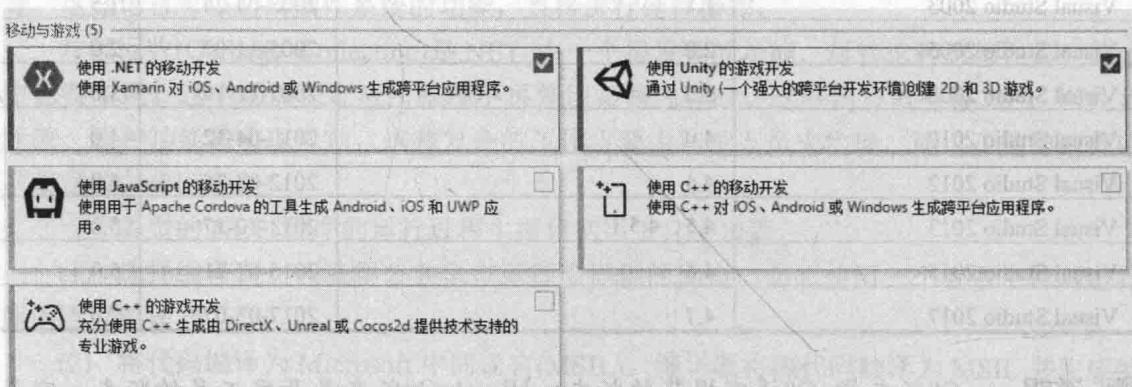


图 1-6 移动与游戏类

② “单个组件”选项卡。包含其他工具集(共 3 项), 如图 1-7 所示。

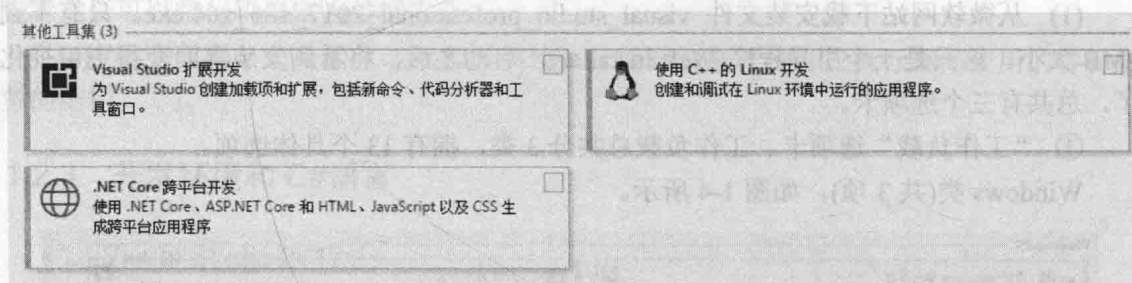


图 1-7 其他工具集

③ “语言包”选项卡。从中可以选择各种语言, 在这里我们选择简体中文。

(2) 勾选需要的组件即可进行安装。为学习本课程以及后继的 ASP.NET 和网络编程, 可以按如图 1-8 所示勾选(自定义选择安装路径时, 注意所属路径的预留空间要充足, 否则安装会失败)。然后进行安装。

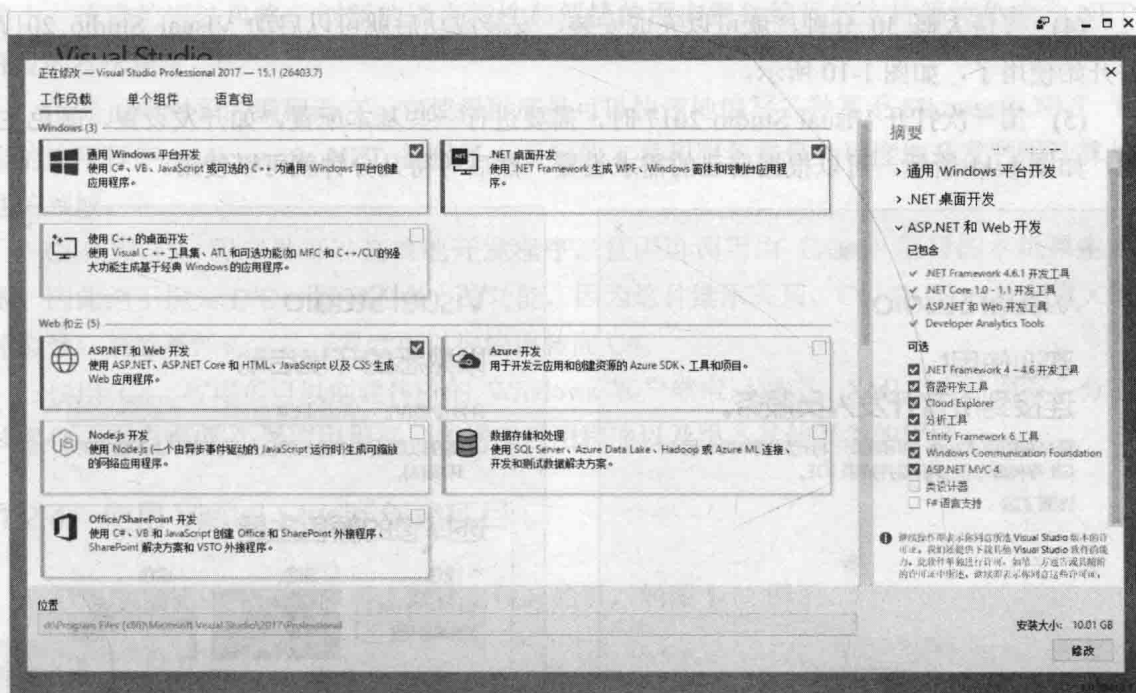


图 1-8 勾选需要的功能并安装

(3) 进入下载和安装界面。安装过程中，Visual Studio 2017 会占用很多的系统资源，所以最好不要开启太多其他软件，等待安装完成，如图 1-9 所示。

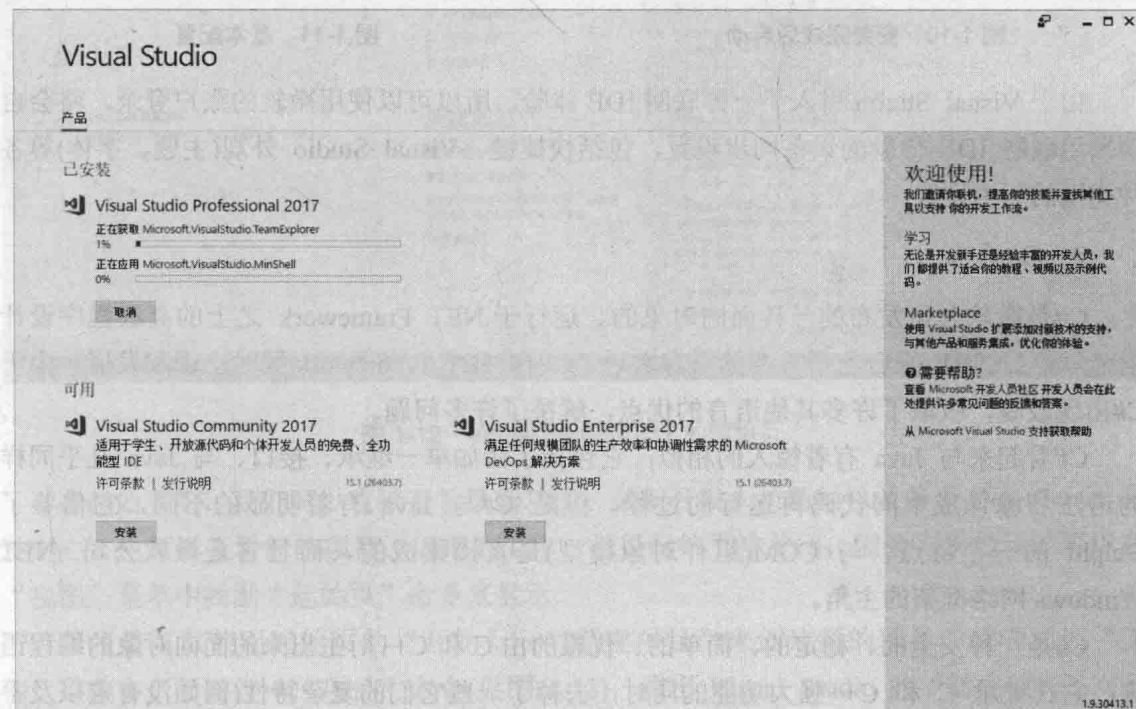


图 1-9 正在安装