



普通高等教育“十一五”国家级规划教材

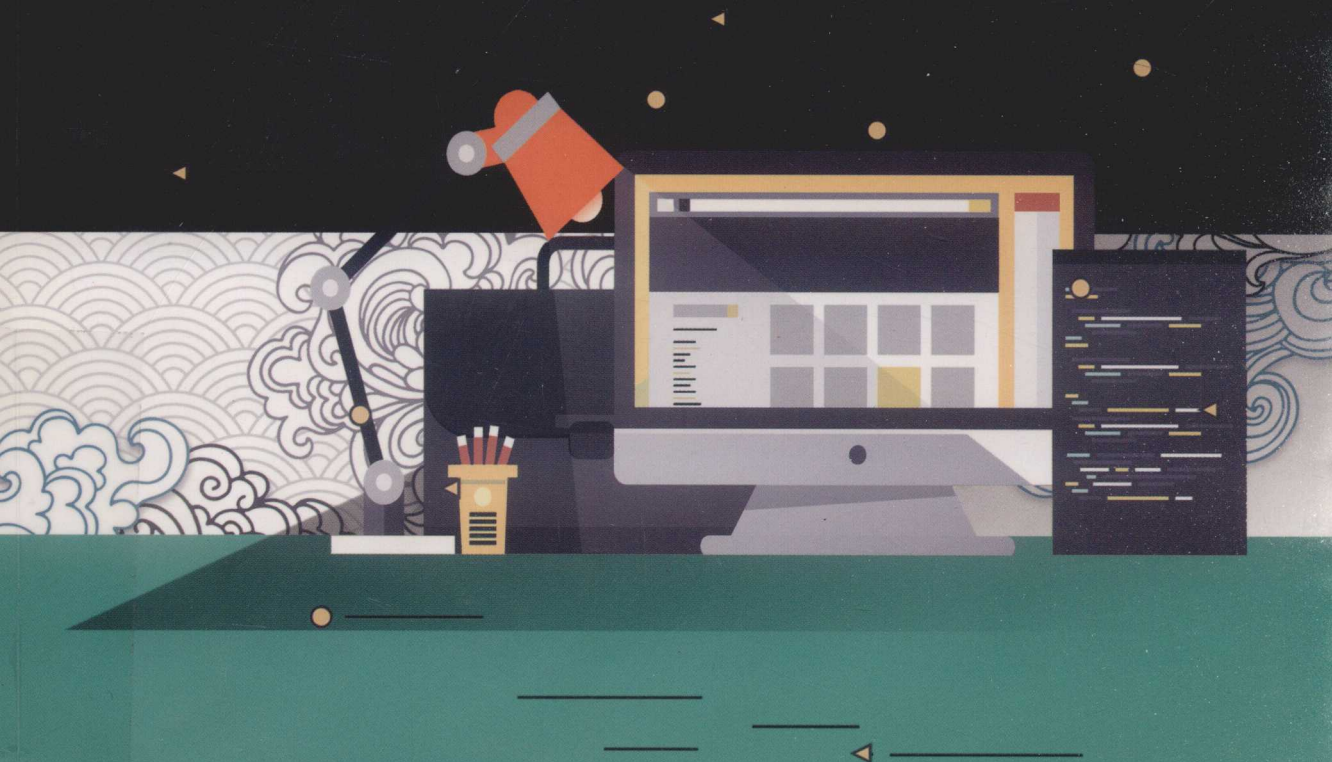
高等院校“十三五”规划教材·软件技术系列

C#程序设计教程

(第5版)

◎ 刘甫迎 主编

◎ 刘光会 王蓉 刘焱 黄敏 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



普通高等教育“十一五”国家级规划教材

高等院校“十三五”规划教材·软件技术系列

C#程序设计教程

(第5版)

刘甫迎 主 编
刘光会 王 蓉 刘 焱 黄 敏 副主 编



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

微软的C#是从C++演变而来的、少有的在互联网诞生后才推出的、简单易学且功能强大的编程语言，适合开发本地及Web应用，移动、虚拟现实和云计算应用开发。C#代表当前一个流行方向，备受用户青睐。

本书讲解C#的结构化程序设计及算法，数组、结构、枚举和集合，面向对象程序设计（类、委托、继承和接口），可视化应用程序设计，文件和流，C/S模式编程，C#的多线程应用等。本书理论与实践相结合，突出应用，有大纲、习题、实验指导、模拟试题等，并通过一个真实应用系统案例进行了实战。

本书可作为应用型本科院校、高等职业院校及培训机构的教材，也可供从事软件开发和应用的人员参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

C#程序设计教程 / 刘甫迎主编. —5版. —北京：电子工业出版社，2019.7

ISBN 978-7-121-36755-7

I. ①C... II. ①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（2019）第106584号

策划编辑：薛华强

责任编辑：程超群 特约编辑：田学清

印刷：三河市华成印务有限公司

装订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱

邮编：100036

开本：787×1092 1/16 印张：18.5 字数：533千字

版次：2005年10月第1版

2019年7月第5版

印次：2019年7月第1次印刷

定价：55.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254569，xuehq@phei.com.cn，QQ1140210769。

/ 前 言 /

C#是从 C 和 C++演变而来的一种新的面向对象的编程语言。结合 C#的强大功能和 Visual Studio.NET 平台的环境和类的支持,形成了功能强大的开发工具——C#.NET。C#是目前主流语言中不多见的在 Internet 出现后推出的语言,故它的设计很适合在互联网上使用。C#可以编译成跨平台的代码,它避免了 C 语言中复杂的指针和多继承等难于控制的编程方案,简单易学且功能强大。C#结合 ASP.NET、Windows Phone 和 Unity 游戏引擎等平台,开发本地及 Web 应用,移动、虚拟现实和云计算应用,无疑代表了当前广泛使用的一个编程方向,备受用户青睐。

本书的前 4 版教材被众多高校使用,是国内高校较早使用的 C#语言编程课程教材,有的版本印刷达 10 多次,受到读者欢迎。本书是《C#程序设计教程》的第 5 版,主要特点如下:

(1) 本书以 Visual Studio 2017/2019 (以下简称 VS 2017/2019) 为平台,介绍了 VS 2019 预览版下 C# 8.0 等的新特点,并以使用较普遍的 VS 2017(C# 7.0)平台为基础进行了内容更新和添加。在保持 C#先前版本的优势外,还增加了基于 Tuple 的多返回值和本地方法等内容。

(2) 在 C#操作 SQL Server 等数据库的基础上,增加了 C#操作 MySQL 数据库的内容(MySQL 是当前云计算、大数据普遍运用的开源关系数据库)。

(3) 严格限定本课程在专业中的培养目标界限,以确定课程内容(例如,把 Access 作为数据库类教材等)。本课程目标是,让学生掌握 C#程序设计的基本知识及技术,为今后成为软件工程师或程序设计员奠定基础。课程重点是 C#的结构化程序设计、面向对象编程和可视化程序设计、客户机/服务器(C/S)模式编程、多线程应用等。

(4) 除附录 A《C#程序设计课程》教学大纲、附录 B《C#程序设计课程》实验指导书外,本书还增加了附录 C 模拟试题。

(5) 在选材上,本书继续强调“理论与实践相结合”,删去了不常用的语法、语义解释,在系统性和实用性方面寻求平衡;降低了个别章节的难度,力求深入浅出,专注于常用功能使用。

(6) 书中实例使用了许多经典算法,弥补了有些读者未学“数据结构”的不足。

(7) 本书由校际联盟中多个学校的教师和“校企合作”的专家参与编写,用真实案例增强实践性,并以一个综合的实际应用系统(用 UML、CASE、MVC 等技术)进行了实战。

(8) 本书教学资源丰富,力求打造成立体化教材:在其精品资源在线开放课程网站上,有配套的 PPT、教学大纲、习题、实验指导书、课程设计、教师在线辅导、CAI、网络课程、试题库、考试系统、相关素材等,这些资源生动、具体、形象、直观,便于教与学。

本书由刘甫迎教授担任主编,刘光会、王蓉、刘焱、黄敏担任副主编。刘甫迎编写第 1 章、附录 A 和附录 C;刘光会编写第 2 章至第 6 章;王蓉编写第 7 章、第 9 章、第 10 章;刘焱编写第 8 章、第 6.2.4 节、第 11 章;黄敏编写第 7.4 节、第 12 章和附录 B。全书由刘甫迎统稿。其他参编人员包括:周绍敏等。在编写和出版的过程中,得到电子工业出版社薛华强编辑的大力支持,在此谨表示感谢!由于编者水平有限,书中难免有不足之处,请斧正。

刘甫迎
2019 年 4 月

目 录

CONTENTS

第 1 章 C#概述	1	3.1.4 装箱和拆箱	36
1.1 C#简介	1	3.2 常量与变量	38
1.1.1 C#的缘起	1	3.2.1 常量	38
1.1.2 C#的特点	2	3.2.2 变量	38
1.1.3 C#的发展及 VS2019 下 C#8.0 等的新功能	4	3.3 运算符和表达式	40
1.1.4 Microsoft .NET 平台	7	3.3.1 运算符	41
1.1.5 C#的运行环境及安装	9	3.3.2 表达式	48
1.2 C#集成开发环境	12	习题	50
1.2.1 C#的启动	12	第 4 章 结构化程序设计	51
1.2.2 C#集成开发环境	13	4.1 结构化设计的概念	51
习题	18	4.1.1 结构化设计的概念及算 法的概念	51
第 2 章 简单的 C#程序设计	19	4.1.2 流程图	52
2.1 C#程序结构	19	4.2 顺序结构	53
2.1.1 第一个 C#程序	19	4.2.1 顺序结构的概念	53
2.1.2 编译和执行程序	20	4.2.2 顺序结构的实例	53
2.1.3 C#程序结构分析	21	4.3 选择结构	54
2.1.4 标识符	22	4.3.1 选择结构的概念	54
2.2 输入/输出操作	24	4.3.2 条件语句	55
2.2.1 Console.WriteLine() 方法	24	4.3.3 分支语句	59
2.2.2 Console.Write() 方法	27	4.3.4 选择结构的实例	62
2.2.3 Console.ReadLine() 方法	27	4.4 循环结构	64
2.2.4 Console.Read() 方法	28	4.4.1 循环结构的概念	64
习题	29	4.4.2 while 语句和 do-while 语句	65
第 3 章 数据类型、运算符与表达式	30	4.4.3 for 语句和 foreach 语句	67
3.1 数据类型	30	4.4.4 跳转语句	70
3.1.1 值类型	30	4.4.5 循环语句的算法——循环 结构的实例	73
3.1.2 引用类型	32	4.5 异常处理	75
3.1.3 类型转换	35	4.5.1 异常处理的概念	75

4.5.2 异常的处理.....	77	6.5 属性.....	131
习题.....	80	6.5.1 属性的声明.....	131
第5章 数组、结构和枚举.....	84	6.5.2 属性的访问.....	132
5.1 数组.....	84	6.6 索引指示器.....	134
5.1.1 数组的概念.....	84	6.6.1 定义索引指示器.....	134
5.1.2 一维数组.....	84	6.6.2 使用索引指示器访问对象.....	135
5.1.3 二维数组.....	89	6.7 委托与事件.....	136
5.1.4 “冒泡排序”算法——数组的 实例.....	92	6.7.1 委托.....	137
5.2 结构类型.....	93	6.7.2 事件.....	139
5.2.1 结构的声明.....	93	6.8 C#常用的基础类.....	141
5.2.2 结构成员的访问.....	94	6.8.1 Math 类与 Random 类.....	141
5.2.3 顺序查找算法——结构的 实例.....	95	6.8.2 字符串.....	142
5.3 枚举.....	97	6.9 命名空间.....	145
5.3.1 枚举类型的定义.....	98	6.9.1 命名空间的声明.....	145
5.3.2 枚举成员的赋值.....	98	6.9.2 命名空间的成员.....	146
5.3.3 枚举成员的访问.....	100	6.9.3 命名空间的使用.....	147
习题.....	102	习题.....	148
第6章 C#的面向对象程序设计.....	106	第7章 继承和接口.....	151
6.1 面向对象程序设计的基本概念.....	106	7.1 C#的继承机制.....	151
6.2 类和对象.....	109	7.1.1 继承的基本知识.....	151
6.2.1 类的声明.....	109	7.1.2 base 关键字.....	153
6.2.2 类的成员.....	110	7.1.3 覆盖.....	154
6.2.3 对象的声明.....	112	7.2 多态性.....	156
6.2.4 对象初始化器.....	114	7.2.1 多态性概述.....	156
6.3 构造函数和析构函数.....	115	7.2.2 虚方法.....	156
6.3.1 构造函数.....	116	7.2.3 多态性的实现.....	157
6.3.2 析构函数.....	119	7.3 接口.....	158
6.4 方法.....	120	7.3.1 接口的定义.....	159
6.4.1 方法的定义及调用.....	120	7.3.2 接口的成员.....	160
6.4.2 方法的参数类型.....	122	7.3.3 接口的实现.....	163
6.4.3 方法的重载.....	124	7.4 泛型、泛型接口和委托中的协变、 逆变.....	178
6.4.4 静态方法与非静态方法.....	125	7.4.1 泛型、泛型集合 List<T>、 IEnumerable<T>接口及 yield... ..	178
6.4.5 运算符的重载.....	126	7.4.2 协变和逆变.....	182
6.4.6 递归.....	129	7.4.3 泛型接口和委托中的协变和 逆变.....	182
6.4.7 基于 Tuple 的“多”返回值.....	130	7.5 本地方法.....	185

习题	186	10.3.1 Connection 对象	235
第 8 章 可视化应用程序设计	188	10.3.2 Command 对象	237
8.1 Windows 窗体	188	10.3.3 DataReader 对象	238
8.1.1 Windows 窗体的基本属性	189	10.3.4 DataAdapter (数据适配器) 对象	239
8.1.2 创建窗体	190	10.3.5 DataSet 对象	239
8.1.3 设置启动窗体属性	191	10.4 数据绑定控件	241
8.2 控件概述	192	习题	243
8.2.1 控件的基本属性	192	第 11 章 C#的多线程应用	244
8.2.2 控件共有的属性、事件和 方法	193	11.1 多线程简介	244
8.3 命令按钮控件 (Button)	200	11.2 多线程编程	248
8.4 标签控件 (Label)	202	11.2.1 线程的优先级和执行状态	248
8.5 文本框控件 (TextBox)	204	11.2.2 线程同步	249
8.6 单选按钮 (RadioButton) 和复选框 (CheckBox 和 CheckedListBox)	205	习题	252
8.7 面板控件 (Panel) 和分组框 (GroupBox) 控件	210	第 12 章 C#案例——FoxOA 的人力资源 培训成绩管理系统	253
8.8 图形框控件 (PictureBox)	212	12.1 系统概述	253
8.9 列表框 (ListBox)、带复选框的列表 框 (Checked ListBox) 和组合框 (ComboBox) 控件	213	12.1.1 办公自动化系统 FoxOA	253
8.9.1 列表框控件 (ListBox)	213	12.1.2 系统功能	253
8.9.2 带复选框的列表框控件 (CheckedListBox)	215	12.2 系统分析与设计 (UML)	254
8.9.3 组合框控件 (ComboBox)	216	12.2.1 计算机辅助软件工程 (CASE) 工具——PD	254
习题	218	12.2.2 系统建模	254
第 9 章 C#的文件和流	220	12.3 主要关键技术	256
9.1 文件和流的概念	220	12.3.1 MVC 设计模式简介	256
9.2 文件的读和写	225	12.3.2 MVC 设计模式的实现	257
9.2.1 读文件	225	12.4 数据库结构	257
9.2.2 写文件	226	12.5 用户使用说明书	258
9.3 文件操作实例——链表算法	227	12.5.1 系统平台	258
习题	232	12.5.2 登录和进入系统	258
第 10 章 客户机/服务器编程访问数据	233	12.6 关键源程序	259
10.1 客户机/服务器 (C/S) 模式编程	233	习题	279
10.2 ADO.NET 概念	234	附录 A 《C#程序设计课程》教学大纲	280
10.3 ADO.NET 对象	235	附录 B 《C#程序设计课程》实验 指导书	281
		附录 C 模拟试题	285

本章介绍 C# 的由来和发展、特点、VS 2017 等的新功能、Microsoft.NET 平台及 C# 的运行环境，使读者能够初步了解 C#，并能够进行 C# 的安装、启动和熟悉 C# 的界面。

1.1 C#简介

1.1.1 C#的缘起

美国微软公司的 Visual Studio.NET 已发展到 2017 版，Visual C#（简称 VC# 或 C#，本书统一称 C#）包含在其中。然而 C# 是如何发展而来的呢？

1995 年，SUN 公司正式推出了面向对象的开发语言 Java，它具有跨平台、跨语言的功能特点，Java 逐渐成了企业级应用系统开发的首选工具，而且使得越来越多的基于 C/C++ 的应用开发人员转向于从事基于 Java 的应用开发。

在 Java 势头很猛的软件开发领域可观前景的冲击下，作为世界上最大的软件公司微软立即做出了迎接挑战的反应。很快，微软也推出了基于 Java 语言的编译器 Visual J++。Visual J++ 在最短的时间里由 1.1 版本升级到了 6.0 版本。Visual J++ 6.0 集成在 Visual Studio 6.0 中，不但 Java 虚拟机（Java Virtual Machine, JVM）的运行速度大大加快，而且增加了许多新特性，同时支持调用 Windows API，这些特性使得 Visual J++ 成为强有力的 Windows 应用开发平台，并成为业界公认的优秀 Java 编译器。

Visual J++ 虽然具有强大的开发功能，但主要应用在 Windows 平台的系统开发中，SUN 公司认为 Visual J++ 违反了 Java 的许可协议，即违反了 Java 开发平台的中立性，因而，对微软提出了诉讼，这使得微软处于极为被动的局面。为了改变这种局面，微软另辟蹊径，决定推出其进军互联网的庞大 .NET 计划，该计划中包括重要的开发语言——Visual C#。

微软的 .NET 是一项非常庞大的计划，也是微软发展的战略核心。Visual Studio.NET 则是微软 .NET 技术的开发平台，C# 就集成在 Visual Studio.NET 中。.NET 代表了一个集合、一个环境、一个编程的基本结构，作为一个平台来支持下一代的互联网（本书 1.1.4 节将详细介绍 .NET 平台）。为了支持 .NET 平台，Visual Studio.NET 在原来的 Visual Studio 6.0 的基础上进行了极大的修改和变更。在 Visual Studio.NET 测试版中 Visual J++ 消失了，取而代之的就是 C# 语言。

美国微软公司在 2000 年 6 月份举行的“职业开发人员技术大会”上正式发布了 C# 语言，其英文名为 Visual C-Sharp。微软公司对 C# 的定义是：“C# 是一种类型安全的、现代的、简单的，由 C 和 C++ 衍生出来的面向对象的编程语言，它是牢牢根植于 C 和 C++ 语言之上的，并可立即被 C

和 C++ 开发人员所熟悉。C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力。”

目前使用 C# 进行 C/S (客户机/服务器) 结构编程或用 C# 与 ASP.NET 结合进行 B/S (浏览器/服务器) 结构编程的人员越来越多, 用 C# 进行编程已成为今后程序设计的趋势之一, 而且将逐步超越其他主流编程语言的地位。为什么会这样呢? 请看下节所讲述的 C# 的特点。

1.1.2 C# 的特点

作为微软新一代面向对象的语言产品, C# 语言自 C/C++ 演变而来, 它是给那些愿意牺牲 C++ 一点底层功能, 以获得更方便和更产品化的企业开发人员而创造的。C# 具有现代、简单、完全面向对象和类型安全等特点。

如果读者是 C/C++ 程序员, 学习将会变得很容易。许多 C# 语句直接借用程序员所喜爱的语言, 包括表达式和操作符。假如不仔细看, 就会把它当成 C++。

关于 C# 最重要的一点: 它是现代的编程语言。它在类、名字空间、方法重载和异常处理等领域简化了 C++。摒弃了 C++ 的复杂性, 使它更易用、更少出错。

对 C# 的易用有贡献的是减少了 C++ 的一些特性, 不再有宏、模板和多重继承。特别对企业开发者来说, 上述功能只会产生更多的麻烦而不是效益。

使编程更方便的新功能是严格的类型安全、版本控制、垃圾收集 (garbage collect) 等。所有这些功能的目标都是瞄准了开发面向组件的软件。

1. 可避免使用指针等, 语法更简单、易学

在 C# 中可避免复杂的令人头痛的在 C++ 中流行的指针, 禁止直接进行内存操作, 不能使用 “::” 和 “—>” 运算符, 整型数据 0 和 1 也不再是布尔值, “==” 被用于比较操作而 “=” 被用于赋值操作, 从而减少了运算符错误。C# 使用统一的类型系统, 摒弃了 C++ 中多变的类型系统。

2. 支持跨平台

由于网络系统错综复杂, 使用的硬件设备和软件系统各不相同, 开发人员所设计的应用程序必须具有强大的跨平台性, C# 编写的应用程序就具有强大的跨平台性, 这种跨平台性也包括了 C# 程序的客户端可以运行在不同类型的客户端上, 比如 PDA、手机等非 PC 设备。

3. 面向对象且避免了多重继承

C# 支持所有关键的面向对象的概念, 如封装、继承和多态性。完整的 C# 类模式构建在 NGWS 运行时的虚拟对象系统 (VOS, Virtual Object System) 的上层。对象模式只是基础的一部分。

在 C# 中, 不存在全局函数、变量或者常量。所有的内容都封装在类中, 包括事例成员 (通过类的事例——对象可以访问) 或静态成员 (通过数据类型)。这些使 C# 代码更加易读且有助于减少潜在的命名冲突。

定义类中的方法默认是非虚拟的 (它们不能被派生类改写)。主要特点是, 这样会消除由于偶尔改写方法而导致另外一些原码出错。要改写方法, 必须具有显式的虚拟标志。这种行为不但缩减了虚拟函数表, 而且还确保正确版本的控制。

使用 C++ 编写类, 用户可以使用访问权限给类成员设置不同的访问等级。C# 同样支持 private, protected 和 public 三种访问权限, 而且还增加了第四种访问权限: internal。

C# 仅允许一个基类, 如果程序需要多重继承, 可以运用接口。

4. 现代快速应用开发 (RAD) 功能

支持快速应用开发 (Rapid Application Development) 是目前开发语言重要的功能之一, 也正是 C/C++ 的致命伤。网络时代应用系统的开发必须按照网络时代的速度来进行, 支持快速开发可以使得开发人员的开发效率倍增, 从而使得他们可以从繁重的重复性劳动中解放出来。C# 的 RAD 功能主要表现在如垃圾收集、委托等众多特性上。垃圾收集机制将减轻开发人员对内存的管理负担, 而委托功能更是可以让程序员不经过内部类就调用函数。利用 C# 的这些功能, 可以使开发者通过较少的代码来实现更强大的应用程序, 并且能够更好地避免错误的发生, 从而缩短了应用系统的开发周期。许多用 C++ 很费力实现的功能, 在 C# 中不过是一部分的基本功能而已。

5. 语言的兼容、协作交互性

用 C# 编写的程序能最大程度地实现与任何 .NET 的语言互相交换信息, 为开发人员节省了大量的时间。C# 与其他 .NET 语言有很好的协作, 这点对开发人员非常重要。.NET 让各种语言可以真正地互相交流, 开发者不必把一种语言强行改成另一种语言。在全球的程序员中, 用 Visual Basic 作为基本的编程工具的人占了很大一部分, 在跨入 .NET 编程时代的时候, 这些人能轻松地使用 Visual Basic.NET 开发 Web 应用程序, C# 可与之很好地兼容、协作交互。

6. 与 XML 的天然融合

由于 XML 技术真正融入到 .NET 和 C# 中, C# 编程变成了真正意义上的网络编程, 甚至可以说 .NET 和 C# 是专为 XML 而设计的。C# 程序员可以轻松通过 C# 内含的类使用 XML 技术。和其他编程语言相比, C# 为程序员提供了更多的自由和更好的性能来使用 XML。

7. 对 C++ 的继承且类型安全

C# 集成并保留了 C++ 强大的功能。例如, C# 保留了类型安全检测和重载功能, 还提供了一些新功能取代原来 C++ 中的预处理程序的部分功能, 提高了语言的类型安全性。

类型安全可以将指针作为一个例子。在 C++ 中使用指针, 程序员能自由地把它强制转换成为任何类型, 包括可以执行把一个 `int*` (整型指针) 强制转换成一个 `double*` (双精度指针) 这样的不安全操作。但只要内存支持这种操作, 它就可以执行, 这并不是程序员所想象的企业级编程语言的类型安全。

C# 实施最严格的类型安全, 以保护自己及垃圾收集器。因此程序员必须遵守 C# 中一些相关变量的规则。

C# 取消了不安全的类型转换。不能把一个整型强制转换成一个引用类型 (如对象), 而当向下转换时, C# 验证这种转换是正确的, 也就是说, 派生类真的是从向下转换的那个类派生出来的。

8. 版本可控

在过去的几年中, 几乎所有的程序员都难免涉及众所周知的 “DLL 地狱”, 该问题起因于多个应用程序都安装了相同 DLL 名字的不同版本。有时, 老版本的应用程序可以很好地和新版本的 DLL 一起工作, 但是更多的时候它们会中断运行。

NGWS 运行时将对程序员所写的应用程序提供版本支持。C# 可以最好地支持版本控制。尽管 C# 不能确保正确的版本控制, 但是它可以为程序员保证版本控制成为可能。有了这种支持, 一个开发人员就可以确保当他的类库升级时, 仍保留着对已存在的客户应用程序的二进制兼容。

综上所述, 我们可以认为 C# 是派生于 C 语言和 C++ 语言的一种程序设计语言, 它使程序员能够更快速、更容易地为微软 .NET 平台开发应用程序。但 C# 也有一些弱点, 例如, C# 程序设计和编译程序级的优化不能在非微软的平台上充分利用, 想在非微软平台上展开 .NET, 再充分运用它

们也是不现实的。

1.1.3 C#的发展及 VS2019 下 C#8.0 等的新功能

1. C#及 VS 的发展

包含 C#语言的 Visual Studio.NET (简称 VS) 已发展到 VS2017 版本和 VS2019 预览版。

C# 1.0 于 2000 年亮相, 随着 C# 2.0 和 Visual Studio 2005 的问世, C#等语言新增了几个重要的功能, 其中包括泛型、迭代器和匿名方法等。随同 Microsoft Visual Studio 2008 发布的 C# 3.0 添加了更多功能, 例如, 扩展方法、lambda 表达式、自动属性, 对象初始化和集合初始化器, 以及最有名的语言集成查询 (Language Integrated Query, LINQ) 工具。

Visual C# 4.0 (Visual C#2010) 改善了与其他语言 and 技术的互操作性。新增的功能包括命名和可选参数; dynamic 类型, 它告诉语言的“运行时”要实现一个对象的晚期绑定, 以及协变性和逆变性, 它们解决了泛型接口的定义方式所造成的一些问题。C# 4.0 利用了新版本的 .NET Framework, 版本也是 4.0。在这个版本中, .NET Framework 添加了许多新东西, 但最重要的就是构成“任务并行库” (Task Parallel Library, TPL) 的类和类型。现在可以使用 TPL 构建具有良好伸缩性的应用程序, 从而快速和简单地利用多核处理器的强大能力。对 Web 服务和 Windows Communication Foundation (WCF) 的支持也得到了扩展, 现在可以遵循 REST 模型和较传统的 SOAP 方案构建服务。Microsoft Visual Studio.NET 2010 提供的开发环境使得这些强大的功能变得易于使用, Visual Studio 2010 新增的大量向导和增强措施也显著提高了开发人员的工作效率。Visual C# 2010 中新的及加强的主要功能与特性可有效改善 Microsoft Office 等的编程能力和对代码的分析、研究能力, 并对测试驱动开发提供支持; 增加了类型等价支持 (Type Equivalence Support), 相比从主互操作程序集 (Primary Interop Assembly) 中导入类型信息, 现在可以部署一款带有内嵌类型信息的应用, 通过内嵌的类型信息, 应用可在运行时间内使用类型而无须参考运行程序集; 新的命令行选项: /langversion 命令行选项可让编译器接受只在特定 C#版本中有效的语法, /appconfig 编译器选项可让 C#应用指定程序集的应用配置文件位置; Visual C# 2010 加强了集成开发环境 (IDE): 有了调用层次结构 (Call Hierarchy), 能通过用户的代码进行导航; 可以使用 Navigate To 功能来搜索包含在字符中的关键字, 通过使用驼峰式大小写风格 (Camel casing) 和下划线来将这些符号分割成关键字; 当单击源代码中的一个字符时, 该字符所有的实例都会被高亮显示; 使用中生成 (Generate From Usage) 可让用户在定义 classes 和 members 之前就使用它们, 而无须在代码中留下当前的位置, 用户可以生成一个想要使用但仍未定义的 class、构造函数、方法、属性、栏目等存根 (stub), 这将对工作流程产生最小的影响; IntelliSense 现在为 IntelliSense 声明完成提供了两种选择, 即完成模式和建议模式, 当 class 和 member 在被定义前使用时, 建议模式会被使用; 在 Visual C# 2010 中, 实时语义误差 (Live Semantic Error) 功能得到了加强, 它使用波浪下划线来发出错误信号与提示。

Visual Studio.NET 2012 (含有 C# 2012) 有以下特性。Visual Studio.NET 2012 的代号为 Ark (方舟)。众所周知, 人们认为 2012 年有 67.58% 的可能性为世界末日 (据中国国家统计局调查结果推算)。为了使大家能够平安度过世界末日, 微软将新的 VS 代号命名为 Ark, 可见 MS 的社会责任感。新的 VS 增加了触摸事件。随着触摸设备的普及, 原有的鼠标键盘事件将不能满足需要了, 配合 Windows 8 的上市, VS 将增加大量的触摸事件在 WPF、SL 里面。值得大家关注的是, Visual Studio.NET 2012 为了体现博爱和自然和谐, 首次导入了 Miao 系统, 一种专为猫咪设计的触摸系统。猫咪的爪和人类的手指有很大不同, 为了顾及猫咪的触摸需要, 特地开发了 Miao 系统。据项

目负责人表示, Wang 系统, 为狗定制的触摸系统也在开发中。

混合的编码环境。有人批评微软为 C#投入了大量的资源, 导致了 VB、F#的关注度不够。为此新版本的 VS 推出了后缀为 .mix 的新代码。可以在同一份代码里面, 混合各种语言的代码。

众多技术社区的编辑器的代码着色功能表示压力很大。

VS2012、Microsoft .NET Framework 4.5 是一个针对 .NET Framework 4 的高度兼容的就地更新。通过将 .NET Framework 4.5 与 C#4.5 (C# 2012) 或 Visual Basic4.5 编程语言结合使用, 可以编写 Windows Metro 风格的应用程序 (为触摸而设计的最新卡片风格界面, 能向用户显示重要信息, 这个界面同时支持鼠标和键盘, 并适用于平板设备)。
.NET Framework 4.5 包括针对 C#4.5 和 Visual Basic 4.5 的重大语言和框架改进, 以便能够利用异步性、同步代码中的控制流混合、可响应 UI 和 Web 应用程序可扩展性。

2. Visual Studio.NET 2015 的特性

Visual Studio.NET 2015 (简称 VS2015) 中文旗舰版是微软重磅推出的一款软件开发平台, 可以帮助开发人员打造跨平台的应用程序, 从 Windows 到 Linux, 甚至 iOS 和 Android 都可以跨平台使用。软件现在可以轻松开发 Android、iOS、WP 应用程序, 还可以开发能运行在 Mac、Linux 上的 ASP.NET 网站, 并支持云服务。另外无论团队规模或项目复杂程度如何, 由 Team Foundation Server 支持的 VS2015 旗舰版均可帮助用户将创意变为软件, 用它所写的目标代码适用于微软支持的所有平台, 包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework、Microsoft Silverlight 及 Windows Phone。

VS2015, 这是一款由开发人员工作效率工具、云服务和扩展组成的集成套件, 让你和你的团队可以创建适用于 Web、Windows 商店、桌面、Android 和 iOS 的强大的应用程序和游戏。

(1) 自定义窗口布局。如果你需要在多个设备上开发应用, 这个功能就能让你开发起来得心应手了。举个例子, 如果你想在回家的火车上用 Surface Pro 开发, 上班的时候在 23 英寸的显示器上开发, 可以使用 Window → Apply Window Layout 来快速切换开发环境的布局, 以适应当前的设备类型。如果登录了 VS 2015, 你还可以使用快捷键来切换开发环境布局, 非常方便。

(2) 更优的代码编辑器。代码编辑器已经替换成 “Roslyn”, 这将会给你带来不一样的代码编辑体验。当要修复代码的时候, 界面中将会出现一个小灯泡, 它会提供一系列修复代码的方案, 你只需选择即可。

(3) Shared Project 集成。之前有多少开发者想在 Visual Studio 之外使用 Shared Project 功能却未能实现, 但是现在可以了, 你只需要搜索 shared, 在出现的界面中选择 Visual C# Shared Project, 然后新建一个名为 Person.cs 的类, 编写代码。你也可以创建一个 WPF 应用程序, 引用 Shared Project 项目。

(4) Bower 和 NPM 中的代码智能提示。如果你想创建一个 ASP.NET 5 Web 应用程序, 系统将会自动生成一个目录结构。这里将生成一个名为 Dependencies 的文件夹, 里面包含了 Bower 和 NPM, 一般来说, 你可以将 Bower 看作客户端的开发包, 如 jQuery 和 Angular 脚本库; 可以把 NPM 看作开发工具, 如 Grunt 和 Gulp。这些开发包都是通过一个 JSON 格式的文件来进行统一管理的, 如下所示。

bower.json for Bower

config.json for NPM

如果你想在 Bower 中添加一个类库, 可以打开 bower.json 文件, 加入自己的类库即可。一旦添加成功, 你将会看到安装/更新/删除的菜单选项, 这将使我们在开发 Web 应用程序时更加得心应手。

(5) 调试 Lambdas 表达式。现在我们可以调试 lambda 表达式了，如果想看每步的执行结果，可以添加监视器和断点，这样就可以更方便、更直接地调试应用程序了。

3. VS2017、C#7.0 及 VS2019 预览版下 C#8.0 的新功能

2017 年 3 月 7 日，最强 IDE Visual Studio 2017 正式版发布，该版本可支持 C#、C++、Python、Visual Basic、Node.js、HTML、JavaScript 等各大编程语言，它不仅可编写 Windows 10 UWP 通用程序，甚至还能开发 iOS、Android 移动平台应用。其不仅添加了实时单元测试、实时架构依赖关系验证等新特性，还对许多实用功能进行了改进，如代码导航、IntelliSense、重构、代码修复和调试等。无论使用哪种语言或平台，都能节省开发者在日常任务上花费的时间和精力。

此外，该版本还带来了一个新的轻量化和模块化的安装体验，使用者可根据需要量身定制安装。多个增强功能汇集在一起，使 Visual Studio 2017 的启动速度比 Visual Studio 2015 快 3 倍，解决方案加载时间缩短 2~4 倍。该版本有如下亮点。

(1) 导航增强：

Visual Studio 2017 极大地改善了代码导航，并对结果进行着色，提供自定义分组、排序、过滤和搜索。强大的 Go to All（快捷键为 Ctrl + T 或 Ctrl + ,），能完成对解决方案中的任何文件、类型、成员或符号声明的快速、完整搜索。

(2) 无须解决方案加载文件：

Visual Studio 2017 可以直接打开并处理 C#、C++、Ruby、Go 等一系列语言的任何文件。

(3) 智能过滤：

IntelliSense 现在提供过滤器，帮助筛选你所需要的，而不必涉及过多的步骤。

(4) 语言改进：

添加了新的 C# 语言重构命令，可以使代码符合最新标准。新的风格分析器和对 EditorConfig 的支持能够协调整个团队的编码标准。

(5) CMake support for C++：

可以通过在 Visual Studio 中直接加载 CMake 项目来开始编码。

(6) Linux support for C++：

Visual C++ for Linux 开发现是在 Visual Studio 2017 的一部分。

(7) Live unit testing：

顾名思义，实时告诉你单元测试将通过或失败，而不用离开代码编辑器。

(8) Run to Click：

当调试器停止在某个中断状态时，将鼠标悬停在一行代码上，你会看到 Run to Click glyph 提示。单击可在该行停止并下次继续以此执行。

(9) Exception Helpers：

可立即查看异常的根本原因，即时访问内部异常。此外，可以在抛出异常停止时通过单击复选框添加条件来排除从指定模块抛出的异常类型。

(10) 小而轻的安装：

Visual Studio 2017 新的安装程序更容易启动和运行。最小安装所占空间只有以前版本的十分之一，只需一两分钟即可完成安装。

C#7.0 等的新功能主要有基于 Tuple 的“多”返回值方法、本地方法、模式匹配、返回引用等。前两者在本教材“6.4 方法”中有详细介绍。“模式匹配”也许能算得上 C#本次更新重要的升级，也是最受关注的特性，通过模式匹配，可以简化大量的条件代码（包括 Switch 语句、Match 表达式、Is 表达式等）。“返回引用”早在最初的发行版 C# 1.0 中，就借鉴并延续了 C/C++中的指针参

数,原生允许将值类型数据的引用(指针)通过标记 `ref` 参数的形式,传递到方法体中,但对于方法内的值类型引用,该如何以引用的方式返回,却一直以来没有一个非常完美的解决方案,尽管这种用例非常少见。`C#7.0` 解决了此问题。

`C# 8.0` 将与 `.NET Core 3.0` 同时发布。然而,随着 `Visual Studio 2019` 预览版的推出,这些特性将开始活跃起来,其新特性如下。

Nullable reference types 可空引用类型:此特性的目的是帮助人们处理无处不在的空引用异常,这种异常已经困扰了半个世纪的面向对象编程。这个特性阻止系统将 `null` 放入普通引用类型中(如字符串),从而使这些类型不可为 `null`,程序员可以在项目、文件甚至行级别执行此操作。

Async streams 异步流:如果想要消费(或生产)连续的结果流(例如,可能从物联网设备或云服务获得),异步流就是为此而存在的。`IAsyncEnumerable<T>`,即 `IEnumerable<T>` 的异步版本。允许 `await foreach` 消费它们的元素,并以 `yield return` 生产元素。

Ranges and indices 范围和索引:添加一个类型 `Index`,用于索引。程序员可以创建一个整型来表示从头开始的索引,或者一个 `^` 前缀的从结尾表示的索引。

Default implementations of interface members 接口成员的默认:现在一旦发布了一个接口,就不能在不破坏它的现有实现的情况下向它添加成员。在 `C# 8.0` 中,没有实现该成员(可能因为他们编写代码时还没有该成员),将允许为接口成员提供一个默认实现。另外,**Recursive patterns** 递归的模式匹配、**Switch expressions** `Switch` 表达式、**Target-typed new-expressions** 已知目标类型的新表达式等新功能在此不赘述了。

大多数 `C# 8.0` 语言特性都可以在任何版本的 `.NET` 上运行。但是,其中一些语言特性具有平台依赖性。`Async streams`、`Index` 和 `Range` 都依赖于 `.NET Standard 2.1` 的新类型。正如 Immo 在他的文章《公布.NET Standard 2.1》所说的那样,`.NET Core 3.0`、`Xamarin`、`Unity` 和 `Mono` 都将实现 `.NET Standard 2.1`,但 `.NET Framework 4.8` 不会。这意味着当将 `C# 8.0` 在 `.NET Framework 4.8` 中运行时,使用这些功能所需的类型将不可用。

1.1.4 Microsoft .NET 平台

2000年6月22日微软公司公布了其下一代基于互联网平台的软件开发构想——`.NET`,在IT业界引起了广泛反响。什么是 `Microsoft .NET`? 微软公司前首席执行官兼总裁史蒂夫·鲍尔默说:“`.NET` 代表了一个集合、一个环境、一个编程的基本结构,作为一个平台来支持下一代的互联网。`.NET` 也是一个用户环境,是一组基本的用户服务,可以作用于客户端、服务器或任何地方,与改编成的模式具有很好的一致性,并有新的创意。因此,它不仅是一个用户体验,而且是开发人员体验的集合,这就是对`.NET`的概念性描述。”由此可以看出,`.NET`是微软公司为适应Internet发展的需要,所提供的特别适合网络编程和网络服务(Web Service)的开发平台。`.NET`就是将一切都Internet/Web化,让应用程序通过互联网来互相沟通,并同时共享彼此的资源。对于软件开发人员来说,`.NET`是继DOS开发平台、Windows开发平台之后,以互联网为应用程序开发平台的所谓第三波的改变。这一波是以互联网为基础,通过互联网上标准的通信协议来沟通,以全新的开发环境来开发应用程序。在传统的开发环境下,在不同的程序设计语言间进行代码复用和应用集成,以及应用程序的跨平台运行和沟通往往是比较困难的,但是,在`.NET`环境下,程序设计人员不必担心程序设计语言之间的差异。不同语言开发出来的程序,彼此可直接利用对方的源代码,一种语言与另一种语言之间还可以通过原始代码相互继承。这样,在程序开发设计中,设计人员可根据功能需求的不同,随心所欲地选择不同的语言,大大提高了软件开发的效率。另外,在`.NET`

环境下，由于采用了标准通信协议，可以实现应用程序在不同平台上的沟通。

.NET 的核心是.NET 框架（.NET Framework），它是构建于以互联网为开发平台的基础工具。.NET 框架的结构如图 1-1 所示。

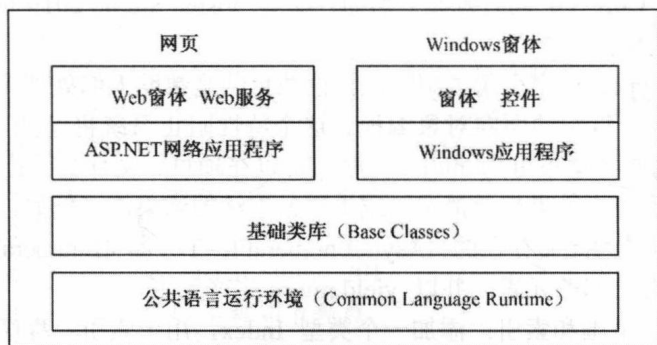


图 1-1 .NET 框架的结构

.NET 框架的最上层是应用程序，这些应用程序大致可以分为面向网络应用的 ASP.NET 网络应用程序和面向 Windows 系统的 Windows 应用程序，这两类应用程序均可使用 VC#.NET、VC++.NET、VB.NET 等来编写。

.NET 框架的中间一层是基础类库，它提供一个可供不同编程语言调用的、分层的、面向对象的函数库。在传统的开发环境中，各种程序设计语言都有自己的函数库，但由于各种语言的编程方式不同，各函数库及对其的调用方法都不同，这样就使得跨语言编程比较困难。.NET 框架提供了一个各种基于.NET 的程序设计语言都可以调用的基础类库，使得各种不一样的编程有了一致性的基础，减少了语言间的界限。在.NET 框架的基础类库中，提供了大量基础类，如窗体控件、通信协议、网络存取等，并以分层的结构来分类。使用这些基础类非常简单方便，只要利用继承或直接调用就可以完成，而且各种基于.NET 的程序设计语言的调用方式都相同（如 C#.NET、VC++.NET、VB.NET 等都可以以同样的方式调用）。这样，在应用程序设计中，不再直接调用底层的系统 API，有效地简化了应用程序设计，减少了应用程序设计中的问题。

.NET 框架的最底层是公共语言运行环境（CLR，Common Language Runtime），它提供了程序代码可以跨平台执行的机制。通常，当使用一种程序设计语言编写程序代码后，编译系统将程序代码与语言本身提供的函数库结合，编译成机器可以直接执行的本地代码（Native Code）。但是，当使用.NET 程序设计语言编写好程序代码后，它会被编译两次，第一次是将程序代码和基础类组合编译成中间语言（IL，Intermediate Language），第二次是在执行时，.NET 的公共语言运行环境（CLR）会将中间语言（IL）代码载入内存，然后及时地将其编译成运行平台的 CPU 可以执行的本地代码。这样，这些应用程序将可以在任何具有 CLR 的平台上执行。正是这样的运行模式，使得应用程序可在不同环境下执行。

.NET 的公共语言运行环境（CLR）还提供了系统资源统一管理和统一安全机制。以前，在互联网结构下开发应用程序时，虽然互联网的本质是开放的，但是由于在不同的系统平台之间互不相容，使得各系统平台之间的合作也仅限于特定的功能。例如，浏览器与网站服务器之间，通过 HTTP（超文本传输协议，Hyper Text Transport Protocol）协议来通信；电子邮件收发程序与服务器通过 SMTP（Simple Mail Transfer Protocol）与 POP3（Post Office Protocol Version 3）协议存取信件等。而现今.NET 要建立的是各式各样的网络应用程序和网络服务，让各种系统的应用程序通过互联网沟通，.NET 拥有这样强大的功能，除内部结构的特点外，最重要的一点是.NET 框架采用

了 XML 和 SOAP 两项关键技术,保证了各种系统的应用程序通过互联网方便地进行沟通,并同时共享彼此的资源。XML (可扩展标记语言, eXtensible Markup Language) 是当今热门技术之一,它是一种 Word Wide Web Consortium (W3C 协会) 标准下的结构化数据表达语言。它提供了跨程序、跨平台数据交换的公共格式。.NET 对所有应用程序和系统的输入/输出均以 XML 来封装数据,这样就可以方便地交换数据,而不必再进行特定的转换。SOAP (Sample Object Access Protocol) 是一种以 XML 为基础的分布式对象通信协议。在 SOAP 基础上,应用程序可以利用现有的互联网结构可以实现彼此沟通,而不会被防火墙阻碍。但一般的程序设计者,不必过多了解底层的通信协议,这是.NET 开发环境提供的功能。

1.1.5 C#的运行环境及安装

C#是 Visual Studio.NET 的一部分,作为一个强大的集成开发工具, Visual Studio.NET 对系统环境有较高的要求。因此,在安装 C#7.0 之前要全面确定所使用计算机的软、硬件配置情况,看看是否能达到基本配置的要求,以便正确地安装并全面地使用其强大的功能。

1. 硬件要求

Visual C#7.0 用户计算机的配置需要达到 Windows 10 (或者 Windows 8、Windows 7) 的硬件要求。

- 中央处理器 (CPU): 建议采用 2.4GHz 或以上 64 位 (X64) 处理器。
- 内存 (RAM): 当系统运行时 Visual Studio.NET 的 IDE 和操作系统都要占用不少的内存空间。推荐 8GB (64 位) 内存以上。
- 硬盘: 推荐 500GB (64 位) 或以上硬盘空间。

2. 软件要求

- 操作系统: Windows 10 (或者 Windows 8、Windows 7)。
- 后台数据库: 推荐 MySQL、SQL Server Express, Access 2008 或 SQL Server 2008 以上版本。

3. C#的安装

C#是 Visual Studio.NET 的一部分,同其他的.NET 语言一样,都必须在.NET 框架环境下运行。因此,要建立一个完整的 C#开发平台,必须安装 Visual Studio.NET 和.NET Framework SDK (.NET 框架软件开发工具包)。

无论安装的是 Windows 10 操作系统还是 Windows 8 操作系统,只要硬件配置满足 C#的要求,都可以安装 C#。Visual Studio 2017 拥有全新的开发界面和诸多新功能, Visual Studio2017 是一个卓越版本。它的目的就是帮助用户在贵在创意、重在速度的市场中发展壮大。让我们来看一看它帮助用户将创意快速转化为应用程序的一些方法。

(1) 下载 Visual Studio 2017 之后,解压安装包,打开安装的应用程序,弹出安装界面,单击“继续”按钮开始进行 VS2017 的安装,如图 1-2 所示。



图 1-2 Visual Studio 2017 安装启动

(2) 稍等片刻。选择安装的磁盘位置，并同意条款和条约，然后单击“下一步”按钮安装继续，如图 1-3 所示。

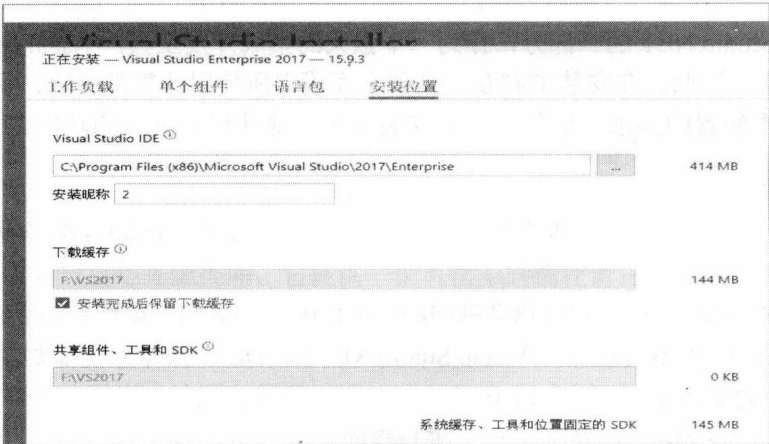


图 1-3 安装位置

(3) 选择安装功能与组件，单击“安装”按钮，开始安装，如图 1-4 所示。

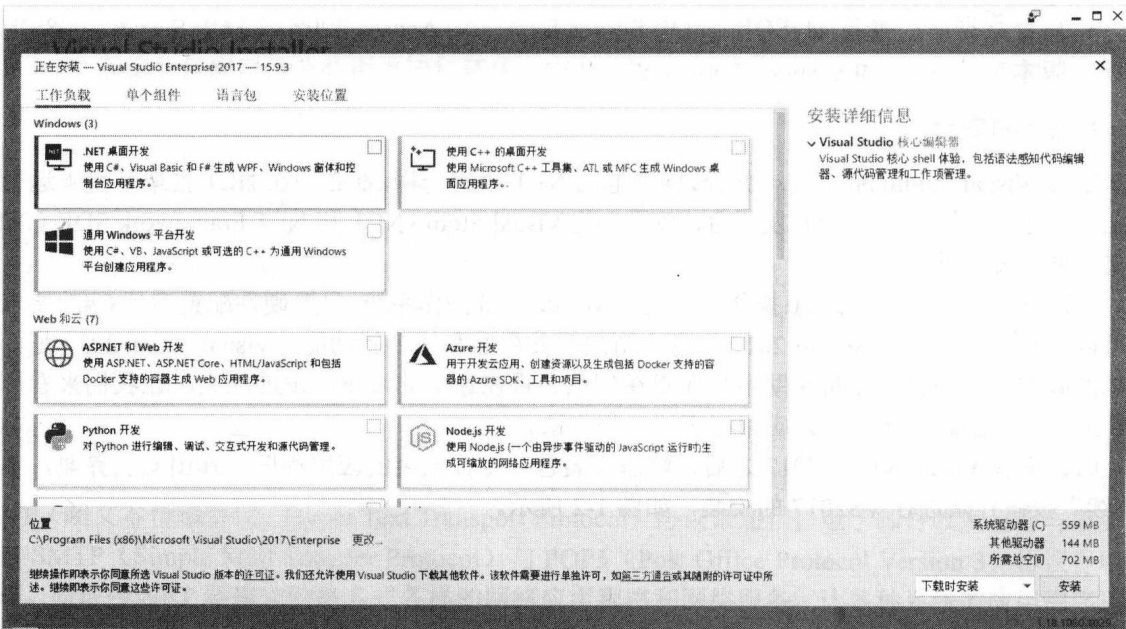


图 1-4 选择安装功能与组件