



高等学校电子信息类“十二五”规划教材

# C#语言程序设计 基础教程

贾延明 张永涛 主 编  
徐好芹 聂 阳 卢朝华 副主编



西安电子科技大学出版社  
<http://www.xduph.com>

高等学校电子信息类“十二五”规划教材

# C#语言程序设计基础教程

贾延明 张永涛 主编

徐好芹 聂阳 卢朝华 副主编

西安电子科技大学出版社

## 内 容 简 介

本书分为三大部分,包括9个单元,共30个任务。第一部分(单元1~单元3)主要介绍C#基本语法与流程控制语句,第二部分(单元4~单元6)主要介绍使用WinForm设计Windows应用程序,第三部分(单元7~单元9)主要介绍ADO.NET实现数据库应用程序的开发。

本书可作为“C#语言程序设计”课程应用型本科生、专科生的教材,也可供从事C#语言程序设计工作的人员参考。

## 图书在版编目(CIP)数据

C#语言程序设计基础教程/贾延明,张永清主编. —西安:西安电子科技大学出版社,2015.7

高等学校电子信息类“十二五”规划教材

ISBN 978-7-5606-3723-5

I. ① C… II. ① 贾… ② 张… III. ① C语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆CIP数据核字(2015)第147292号



策 划 刘小莉

责任编辑 马武装 曹 锦

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xdpuh.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 北京京华虎彩印刷有限公司

版 次 2015年7月第1版 2015年7月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 14

字 数 327千字

定 价 25.00元

ISBN 978-7-5606-3723-5/TP

**XDUP 4015001-1**

\*\*\*如有印装问题可调换\*\*\*

# 前 言

C# 是微软公司基于 .NET 平台推出的一种全新、面向对象的高级程序设计语言。它充分吸收了 C/C++ 的优点，继承了 Visual Basic 的高效和 C++ 的强大功能，基于 .NET Framework 的有力支撑，提供了实现跨平台应用、开发的强有力的集成开发工具和方法。用微软公司的话来说：“C# 是从 C 和 C++ 派生出来的一种简单、现代、面向对象和类型安全的编程语言”。

C# 看起来与 Java 有着惊人的相似，它几乎与 Java 有相同的语法，也是先编译成中间代码，再加载到内存运行，但在底层实现中 C# 与 Java 却有着本质的区别。Java 程序编译后形成的字节代码需要在 Java 虚拟机 (JVM) 上运行；C# 程序编译成中间代码后则是通过 .NET Framework 中的公共语言运行时 (Common Language Runtime, CLR) 来执行的。C# 借鉴了 Delphi 的一些特点，与 COM (组件对象模型) 是直接集成的，同时，.NET Framework 还提供了内容丰富、功能强大的类库供 C# 调用，这使得 C# 变成一种功能十分强大的开发工具，可以实现几乎所有类型应用程序的开发。

如今，C# 已经成为微软 .NET 平台的主角，C# 程序员队伍也日益庞大。凭借着微软雄厚的技术实力和不容动摇的软件霸主地位，在今后可预见的未来，C# 必将得到进一步的加强和完善，受到更多程序员的青睐。可以说，要掌握软件开发的未来，就要先掌握基于 .NET 平台的 C# 开发方法。

本书是围绕一个“高校学生管理系统”项目而组织编写的，本着“学以致用”的原则，采用“边学边做、以做促学”的方式，按照任务驱动的教学方法，由浅入深地展开利用 C# 语言开发窗口应用程序的过程及其对应理论知识的讲解。每一个学习任务都附有与案例相关的理论知识及其对应的实际项目部分作为综合案例，并附有相应的实训内容。

在行文组织上，将每个任务都分为以下五部分，分别进行讲解。

- 任务描述：对任务学习完成后能够实现的具体任务的功能进行简要的描述。
- 预备知识：针对要实现的具体任务中的基础理论 (即知识点) 进行展开，并做详细讲解，以期让学生快速掌握 C# 语言的核心理论知识。
- 任务实施：在知识点讲解的基础上，依据任务描述中要求的任务需求，讲述任务实现的具体方法和步骤，侧重学习方法的培养。
- 知识拓展：结合本任务中涉及的知识，介绍前沿内容和拓展内容以开阔学生视野。
- 归纳总结：针对本任务进行简要归纳总结。

本书采用目标驱动和内容驱动相结合的行文方式，其中以内容驱动为主、以目标驱动为辅。具体来讲，总体上是按照 C# 语言教学内容逐层深入统稿全书，先讲解容易、基础的内容，然后讲解复杂、深入的内容，这与目前大多教材的行文方式相同；但在局部上则采用目标驱动的方法，即针对一个较大的知识点，一般都先设定一个具体的目标 (要解决的具体问题)，然后编写一个简要、容易实现、能达到该目标 (解决问题) 的应用程序，而该程

序涉及的知识尽可能覆盖讲解内容的所有知识点。这样，即使读者在内容上不知道“为什么”，但他知道该“怎么做”，由此可以快速获得对知识点的感性认识，这对理解和掌握接着要讲解的内容大有裨益。本书的行文方式有效吸收了内容驱动和目标驱动的优点，能让读者以最快的速度掌握 C# 语言的核心内容。

本书单元 1 和单元 2 由张永涛老师编写，单元 3 和单元 4 由徐好芹老师编写，单元 5 由聂阳老师编写，单元 6 和单元 9 由卢朝华老师编写，单元 7 和单元 8 由贾延明老师编写，全书由贾延明统稿。

最后，感谢对本书出版提供大力支持的史国永副校长、张煜星副校长、胡健主任、信息与电子工程学院汪新民院长、张广勇书记和王晓华院长。感谢信息与电子工程学院的全体教师，谢谢你们的帮助和指导。本书为商丘工学院教育质量工程资助立项项目。

由于编写水平有限，书中不可避免地存在不足之处，欢迎大家批评指正。

编 者

2015 年 2 月

# 目 录

## 第一部分 C# 基本语法与流程控制语句

|                                        |    |
|----------------------------------------|----|
| 单元 1 C# 语言简介 .....                     | 2  |
| 任务 1.1 C# 程序设计语言概述 .....               | 2  |
| 1.1.1 程序设计语言的发展 .....                  | 2  |
| 1.1.2 .NET Framework 与 C# 语言 .....     | 4  |
| 1.1.3 C# 语言开发环境 .....                  | 5  |
| 任务 1.2 创建一个简单的 C# 应用程序 .....           | 9  |
| 1.2.1 使用 Visual Studio 创建控制台应用程序 ..... | 9  |
| 1.2.2 Console 类 .....                  | 12 |
| 1.2.3 C# 的程序结构 .....                   | 14 |
| 实训练习 1 .....                           | 17 |
| 单元 2 C# 语言基础知识 .....                   | 18 |
| 任务 2.1 数据类型与表达式 .....                  | 18 |
| 2.1.1 C# 中的基本数据类型 .....                | 18 |
| 2.1.2 常量与变量 .....                      | 20 |
| 2.1.3 数据类型转换 .....                     | 21 |
| 2.1.4 运算符与表达式 .....                    | 25 |
| 任务 2.2 系统方法中字符串处理方法与用户自定义方法 .....      | 30 |
| 2.2.1 系统方法中字符串处理方法 .....               | 31 |
| 2.2.2 用户自定义方法 .....                    | 34 |
| 任务 2.3 值传递方式与引用传递方式 .....              | 38 |
| 2.3.1 值传递方式 .....                      | 39 |
| 2.3.2 引用传递方式 .....                     | 39 |
| 实训练习 2 .....                           | 42 |
| 单元 3 程序流程控制与数组 .....                   | 44 |
| 任务 3.1 C# 中流程控制语句 .....                | 44 |
| 3.1.1 顺序结构 .....                       | 44 |
| 3.1.2 选择结构 .....                       | 45 |
| 3.1.3 循环结构 .....                       | 53 |
| 任务 3.2 数 组 .....                       | 64 |
| 3.2.1 一维数组 .....                       | 65 |

|                  |    |
|------------------|----|
| 3.2.2 二维数组 ..... | 67 |
| 实训练习 3 .....     | 70 |

## 第二部分 使用 WinForm 设计 Windows 应用程序

|                                       |     |
|---------------------------------------|-----|
| 单元 4 Windows 窗体应用程序的创建 .....          | 74  |
| 任务 4.1 初识 Windows 窗体应用程序 .....        | 74  |
| 4.1.1 认识 Windows 应用程序 .....           | 74  |
| 4.1.2 Windows 窗体控件的常用属性 .....         | 76  |
| 4.1.3 Windows 窗体的跳转与关闭 .....          | 77  |
| 任务 4.2 事件驱动机制 .....                   | 80  |
| 4.2.1 事件驱动机制与窗体事件 .....               | 80  |
| 4.2.2 编写事件处理程序 .....                  | 81  |
| 实训练习 4 .....                          | 84  |
| 单元 5 窗体基本控件的使用与良好编程习惯的养成 .....        | 85  |
| 任务 5.1 设计“高校学生管理系统”的登录及创建学员用户窗体 ..... | 85  |
| 5.1.1 常用的基本控件 .....                   | 86  |
| 5.1.2 使用控件设计窗体的步骤 .....               | 88  |
| 任务 5.2 “高校学生管理系统”的主菜单设计 .....         | 90  |
| 5.2.1 菜单条控件简介 .....                   | 90  |
| 5.2.2 创建菜单的步骤 .....                   | 91  |
| 任务 5.3 “高校学生管理系统”提示功能的实现 .....        | 93  |
| 5.3.1 消息框的创建方法 .....                  | 94  |
| 5.3.2 消息框的返回值 .....                   | 95  |
| 任务 5.4 断点调试与良好编程习惯的养成 .....           | 99  |
| 5.4.1 断点调试 .....                      | 100 |
| 5.4.2 良好编程习惯的养成 .....                 | 103 |
| 实训练习 5 .....                          | 107 |
| 单元 6 窗体高级控件的使用 .....                  | 109 |
| 任务 6.1 “高校学生管理系统”工具栏、状态栏的实现 .....     | 109 |
| 6.1.1 工具栏 .....                       | 109 |
| 6.1.2 状态栏 .....                       | 110 |
| 任务 6.2 “高校学生管理系统”关于窗体图片动画效果的实现 .....  | 112 |
| 6.2.1 图片框 .....                       | 113 |
| 6.2.2 图片列表 .....                      | 114 |
| 6.2.3 定时器 .....                       | 114 |
| 任务 6.3 实现“关于”模式窗体与用户身份登录验证 .....      | 117 |



|                      |     |
|----------------------|-----|
| 6.3.1 模式窗体 .....     | 117 |
| 6.3.2 用户登录身份验证 ..... | 118 |
| 实训练习 6 .....         | 122 |

### 第三部分 ADO.NET 实现数据库应用程序的开发

|                                      |     |
|--------------------------------------|-----|
| 单元 7 使用 ADO.NET 实现数据库访问 .....        | 124 |
| 任务 7.1 ADO.NET 核心对象简介 .....          | 124 |
| 7.1.1 关系数据库简介 .....                  | 124 |
| 7.1.2 常用 SQL 语句 .....                | 125 |
| 任务 7.2 “高校学生管理系统”数据库连接实现 .....       | 131 |
| 7.2.1 SqlConnection 对象常用属性 .....     | 131 |
| 7.2.2 SqlConnection 常用方法 .....       | 132 |
| 7.2.3 DBHelper 类 .....               | 132 |
| 任务 7.3 “高校学生管理系统”数据打开时的异常处理 .....    | 135 |
| 7.3.1 程序错误类型 .....                   | 136 |
| 7.3.2 异常处理 .....                     | 138 |
| 任务 7.4 Command 对象简介 .....            | 141 |
| 7.4.1 Command 对象常用属性 .....           | 142 |
| 7.4.2 Command 对象常用方法 .....           | 142 |
| 任务 7.5 实现“高校学生管理系统”的登录功能 .....       | 149 |
| 7.5.1 用户登录功能需求分析 .....               | 149 |
| 7.5.2 用户登录功能实现方法 .....               | 150 |
| 任务 7.6 实现“高校学生管理系统”查询全部学生信息功能 .....  | 156 |
| 任务 7.7 实现“高校学生管理系统”模糊查询功能 .....      | 159 |
| 7.7.1 ListView 列表视图控件介绍 .....        | 159 |
| 7.7.2 ListView 控件简单应用 .....          | 161 |
| 任务 7.8 实现“高校学生管理系统”添加学员功能 .....      | 169 |
| 任务 7.9 实现学员状态修改及删除 .....             | 178 |
| 实训练习 7 .....                         | 184 |
| 单元 8 使用 DataSet 操作数据库 .....          | 186 |
| 任务 8.1 DataSet 结构及工作原理 .....         | 186 |
| 8.1.1 DataSet .....                  | 187 |
| 8.1.2 DataTable .....                | 187 |
| 8.1.3 DataColumn .....               | 188 |
| 8.1.4 DataRow .....                  | 188 |
| 任务 8.2 使用 DataAdapter 对象查看教师信息 ..... | 190 |
| 8.2.1 认识 DataAdapter 对象 .....        | 191 |



|                                           |     |
|-------------------------------------------|-----|
| 8.2.2 如何填充数据集 .....                       | 192 |
| 8.2.3 如何保存数据集中的数据 .....                   | 192 |
| 任务 8.3 实现“高校学生管理系统”教员信息列表显示 .....         | 195 |
| 8.3.1 认识 DataGridView 控件 .....            | 195 |
| 8.3.2 DataGridView 控件相关属性 .....           | 196 |
| 实训练习 8 .....                              | 201 |
| <b>单元 9 项目实训——机票预定系统的设计与实现</b> .....      | 203 |
| 实训练习 9 .....                              | 211 |
| <b>附录</b> .....                           | 212 |
| 附录 A C# 中的数据类型 .....                      | 212 |
| 附录 B C# 中关键字完整列表 .....                    | 213 |
| 附录 C C# 中的数据类型与 SQL Server 数据类型的对照表 ..... | 214 |
| <b>参考文献</b> .....                         | 215 |

# 第一部分 C#基本语法与流程控制语句

## 内容摘要

C# (C Sharp)是微软(Microsoft)为 .NET Framework 量身定做的程序语言, C# 拥有 C/C++ 的强大功能以及 Visual Basic 简易使用的特性, 是第一个面向组件(Component-oriented)的程序语言, 和 C++ 与 C 语言一样也是面向对象(Object-oriented)程序语言。

C#是一种最新的面向对象的编程语言。它使得程序员可以快速地编写各种基于 Microsoft .NET 平台的应用程序, 而 Microsoft .NET 提供了一系列的工具和服务来最大程度地开发计算与通信领域。

正是由于 C# 面向对象的卓越设计, 使它成为构建各类组件的理想之选——无论是高级的商业项目还是系统级的应用程序。使用简单的 C# 语言结构, 这些组件可以方便地转化为 XML 网络服务, 从而使它们可以由任何语言在任何操作系统上通过 Internet 进行调用。

最重要的是, C# 使得 C++ 程序员可以高效地开发程序, 而绝不损失 C/C++ 原有的强大功能。因为这种继承关系, C# 与 C/C++ 具有极大的相似性, 熟悉 C/C++ 语言的开发者可以很快地转向 C#。

任何一门程序语言课程, 都会涉及该语言的基本语法及逻辑结构, 本部分主要介绍 C# 语言中的基本语法及程序的逻辑结构, 帮助读者快速入门。

## 学习目标

- (1) 了解什么是 C# 以及 C# 相关概念。
- (2) 掌握 C# 中常用的数据类型、表达式、数组、变量定义方法。
- (3) 掌握选择结构与选择结构的语法。
- (4) 掌握自定义方法的定义与调用。
- (5) 了解值传递与引用传递的区别。
- (6) 掌握常用的字符串处理方法。
- (7) 掌握各种数据类型转换方式。
- (8) 养成规范化的编程习惯。

## 单元 1 C# 语言简介

### 任务 1.1 C# 程序设计语言概述

#### ► 任务描述

本节对程序设计语言及 C# 语言进行简要介绍, 主要目的是了解 .NET Framework(框架)、面向对象程序开发语言 C#、集成开发环境(IDE)Visual Studio.NET(简称 VS)等概念, 掌握安装 Visual Studio 2012 的方法。

#### ► 预备知识

##### 1.1.1 程序设计语言的发展

程序设计语言(Programming Language), 是用于书写计算机程序的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中, 这些记号串就是程序。程序设计语言有 3 个方面的要素, 即语法、语义和语用。语法表示程序的结构或形式, 亦即表示构成语言的各个记号之间的组合规律, 但不涉及这些记号的特定含义, 也不涉及使用者。语义表示程序的含义, 亦即按照各种方法所表示的各个记号的特定含义, 但不涉及使用者。语用表示程序与使用者的关系。

自 20 世纪 60 年代以来, 世界上公布的程序设计语言已有上千种, 但是只有很小一部分得到了广泛的应用。从发展历程来看, 程序设计语言可以分为 4 代。

##### 1. 第一代——机器语言

机器语言是由二进制 0、1 代码指令构成的, 不同的 CPU 具有不同的指令系统。机器语言程序难编写、难修改、难维护, 需要用户直接对存储空间进行分配, 编程效率极低。这种语言已经被渐渐淘汰了。

##### 2. 第二代——汇编语言

汇编语言指令是机器指令的符号化, 与机器指令存在着直接的对应关系, 所以汇编语言同样存在难学、难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点: 可直接访问系统接口, 汇编程序翻译成机器语言程序的效率高。从软件工程角度来看, 只有在高级语言不能满足设计要求或不具备支持某种特定功能的技术性能(如特殊的输入输出)时, 汇编语言才被使用。

### 3. 第三代——高级语言

高级语言是面向用户、基本上独立于计算机种类和结构的语言。其最大的优点是：形式上接近于算术语言和自然语言，概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此，高级语言易学、易用，通用性强，应用广泛。高级语言种类繁多，可以从应用特点和对客观系统的描述两个方面对其进行进一步分类。

从应用角度来看，高级语言可以分为基础语言、结构化语言和专用语言 3 类。

#### (1) 基础语言。

基础语言也称通用语言。它历史悠久，流传很广，有大量的已开发的软件库，拥有众多的用户，为人们所熟悉和接受。属于这类语言的有 FORTRAN、COBOL、BASIC、ALGOL 等。FORTRAN 语言是曾在国际上广为流行也是使用得最早的一种高级语言，从 20 世纪 90 年代起，在工程与科学计算中一直占有重要地位，备受科技人员的欢迎。BASIC 语言是在 20 世纪 60 年代初为适应分时系统而研制的一种交互式语言，可用于一般的数值计算与事务处理。BASIC 语言结构简单，易学、易用，并且具有交互能力，成为许多初学者学习程序设计的入门语言。

#### (2) 结构化语言。

20 世纪 70 年代以来，结构化程序设计和软件工程的思想日益为人们所接受和欣赏。在它们的影响下，先后出现了一些很有影响的结构化语言，这些结构化语言直接支持结构化的控制结构，具有很强的过程结构和数据结构能力。PASCAL、C、Ada 语言就是它们的突出代表。

PASCAL 语言是第一个系统地体现结构化程序设计概念的现代高级语言，软件开发的最初目标是把它作为结构化程序设计的教学工具。由于它模块清晰、控制结构完备，有丰富的数据类型和数据结构，语言表达能力强，移植容易，不仅被国内外许多高等院校定为教学语言，而且在科学计算、数据处理及系统软件开发中都有较广泛的应用。

C 语言功能丰富，表达能力强，有丰富的运算符和数据类型，使用灵活方便，应用面广，移植能力强，编译质量高，目标程序效率高，具有高级语言的优点。同时，C 语言还具有低级语言的许多特点，如允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作等。用 C 语言编译程序产生的目标程序，其质量可以与汇编语言产生的目标程序相媲美，具有“可移植的汇编语言”的美称，成为编写应用软件、操作系统和编译程序的重要语言之一。

#### (3) 专用语言。

专用语言是为某种特殊应用而专门设计的语言，通常具有特殊的语法形式。一般来说，这种语言的应用范围狭窄，移植性和可维护性不如结构化程序设计语言。随着时间的推移，被使用的专业语言已有数百种，应用比较广泛的有 APL 语言、Forth 语言、LISP 语言。

从描述客观系统来看，程序设计语言可以分为面向过程语言和面向对象语言两类。

#### (1) 面向过程语言。

以“数据结构 + 算法”程序设计范式构成的程序设计语言，称为面向过程语言。前面介绍的程序设计语言大多为面向过程语言。

(2) 面向对象语言。

以“对象 + 消息”程序设计范式构成的程序设计语言，称为面向对象语言。比较流行的面向对象语言有 Delphi、Visual Basic(简称 VB)、Java、C++ 等。

Delphi 语言具有可视化开发环境，提供面向对象的编程方法，可以设计各种具有 Windows 窗体的应用程序(如数据库应用系统、通信软件和三维虚拟现实等)，也可以开发多媒体应用系统。

C# 语言是为开发应用程序而提供的，拥有较好的开发环境与丰富的开发工具。它具有很好的图形用户界面，采用面向对象和事件驱动的新机制，把过程化和结构化编程集合在一起。它在应用程序开发中采用的图形化方式，无需编写任何程序，就可以方便地创建应用程序界面，且与 Windows 界面非常相似，有时甚至是一致的。

Java 语言是一种面向对象、不依赖于特定平台的程序设计语言，具有简单、可靠、可编译、可扩展、多线程、结构中立、类型显示说明、动态存储管理、易于理解等特点，是一种理想的用于开发 Internet 应用软件的程序设计语言。

#### 4. 第四代——非过程化语言

4GL 是非过程化语言，编码时只需说明“做什么”，不需描述算法的细节。

数据库查询和应用程序生成器是 4GL 的两个典型应用。用户可以用数据库查询语言(SQL)对数据库中的信息进行复杂的操作。用户只需将要查找的内容在什么地方、根据什么条件进行查找等信息告诉 SQL，SQL 将自动完成查找过程。应用程序生成器则是根据用户的需求“自动生成”满足需求的高级语言程序。真正的第四代程序设计语言应该说还没有出现。所谓的第四代非过程化语言大多是指基于某种语言环境的具有 4GL 特征的软件工具产品，如 System Z、PowerBuilder、FOCUS 等。

第四代非过程化语言是面向应用、为最终用户设计的一类程序设计语言。它具有缩短应用开发过程，降低维护代价，最大限度地减少调试过程中出现的问题以及对用户友好等优点。

### 1.1.2 .NET Framework 与 C# 语言

说起 C# 就不能不提到 .NET Framework，.NET Framework 又称 .NET 框架，它是由微软开发的，一个致力于敏捷软件开发(Agile Software Development)、快速应用开发(Rapid Application Development)、平台无关性和网络透明化的软件开发平台。自 2002 年发布第一版以来，.NET Framework 已经迅速占领企业应用市场，全球财富 100 强公司中有 90% 采用 .NET 技术构建信息系统。在这个平台环境中，可以开发出运行在 Windows 上的几乎所有的应用程序，而微软也将推出运行在其他操作系统上的版本。.NET 框架是一种采用系统虚拟机运行的编程平台，以公共语言运行时为基础，支持多种语言(C#、VB、C++、J#等)的开发，如图 1-1 所示。.NET 也为应用程序接口(API)提供了新功能和开发工具。简单地说，.NET Framework 是一个创建、部署和运行应用程序的多语言多平台环境，包含了庞大的代码库，各种 .NET 语言都可以共用这些代码库。

.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。.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。

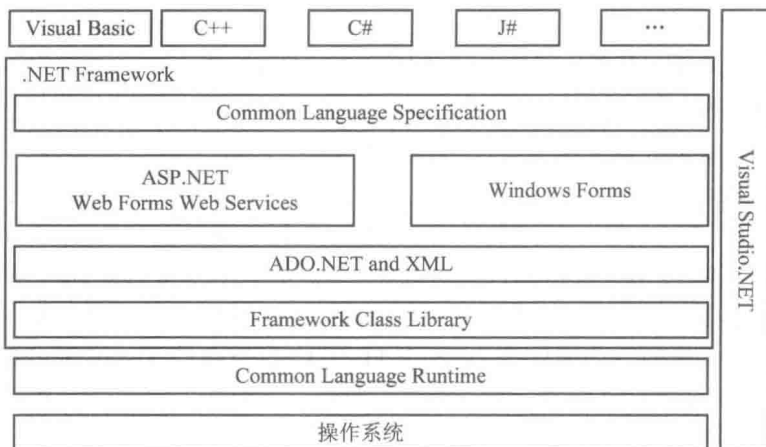


图 1-1 .NET 体系结构

.NET Framework 包括两部分内容, 一个是框架类库(Framework Class Library, FCL), 另一个是公共语言运行时。公共语言运行时是 .NET 框架的基础。框架类库是一个综合性的面向对象的可重用类型集合, 利用它可以开发从传统命令行或者 WinForm 应用程序到基于 ASP.NET 所提供的最新应用程序。

在 .NET 平台上可以使用 C#、VB 等多种语言来进行软件开发, 不同的语言可以使用相同的 FCL。“跨语言”的特性, 使得用某种语言编写的代码可以直接调用其他语言编写的方法, 就好像是这个方法是用该语言编写的一样。也就是说, 在一个项目小组中不同成员使用 .NET 支持的不同语言进行编程, 同样可以一起很好地工作, 不需重新学习新的语言。

C# 看起来像是“C++”中两个加号重叠在一起, 而且在音乐中“C#”表示 C 大调, 表示比 C 高一点的音节。微软借助这样的命名, 表示 C# 在一些语言特性方面是对 C++ 的提升。C# 是微软公司在 2000 年 6 月发布的一种新的编程语言, 是为 .NET Framework 量身定做的程序语言, C# 拥有 C/C++ 的强大功能以及 Visual Basic 简易使用的特性, 是第一个面向组件(Component-oriented)的程序语言, 它和 C++ 与 C 语言一样亦为面向对象(Object-oriented)程序语言。C# 看起来与 C 语言有着惊人的相似: 它包括了诸如单一继承、界面、与 C 语言几乎同样的语法, 以及编译成中间代码再运行的过程。但是 C# 与 C 语言有着明显的不同, 它借鉴了 Delphi 的一个特点, 与 COM(组件对象模型)是直接集成的, 而且它是微软公司 .NET Windows 网络框架的主角。使用 C# 可以编写出传统的 Windows 桌面应用程序(Win Forms)、Windows 服务程序(Windows Service)、Internet 应用程序(ASP.NET)、Web 服务程序(Web Service)等。因此, 在当前的软件开发行业中, C# 已经成为了绝对的主流语言, 可以说 C# 语言和 C 语言在当今企业应用中已经平分天下。

### 1.1.3 C# 语言开发环境

微软推出了这么强大的平台和技术, 当然也会有强大的集成开发环境(IDE)来支持, 那就是微软提供的 Visual Studio.NET。Microsoft Visual Studio(简称 VS)是美国微软公司的开发工具套件系列产品。VS 是一个基本完整的开发工具集, 它包括了整个软件生命周期所需要的大部分工具, 如 UML 工具、代码管控工具、集成开发环境等, 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 和 Microsoft Silverlight。

Visual Studio .NET 是用于快速生成企业级 ASP.NET Web 应用程序和高性能桌面应用程序的工具。Visual Studio 包含基于组件的开发工具(如 Visual C#、Visual J#、Visual Basic 和 Visual C++), 以及许多用于简化基于小组的解决方案的设计、开发和其他技术。

Visual Studio .NET 是一个完整的功能强大的集成开发环境(Integrated Development Environment, IDE), 可用开发控制台应用程序、桌面应用程序、ASP.NET Web 应用程序等多种类型的应用程序, 而且支持 C#、Visual Basic、Visual C++ 等多种 .NET 编程语言。

Visual Studio 最新版本为微软在 2013 年 11 月 13 日发布的 Visual Studio 2013, 在主机本下, 根据所包含的功能分为学习版、测试专业版、专业版、旗舰版等售价不等的版本。不同的版本如表 1-1 所示。

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

| 名 称                    | Visual Studio 版本 | .NET Framework 版本              | C# 版本 |
|------------------------|------------------|--------------------------------|-------|
| Visual Studio.NET 2002 | 7                | 1                              | 1.0   |
| Visual Studio.NET 2003 | 7.1              | 1.1                            | 1.5   |
| Visual Studio 2005     | 8                | 2.0                            | 2.0   |
| Visual Studio 2008     | 9                | 2.0, 3.0, 3.5                  | 3.0   |
| Visual Studio 2010     | 10               | 2.0, 3.0, 3.5, 4.0             | 4.0   |
| Visual Studio 2012     | 11               | 2.0, 3.0, 3.5, 4.0, 4.5        | 5.0   |
| Visual Studio 2013     | 12               | 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1 | 6.0   |

本书采用当前最为流行的 .NET 开发工具 Visual Studio 2012(简称 VS12)和 Microsoft .NET Framework 4.5 类库作为开发环境。Visual Studio 2012 支持开发面向 Windows 7 的应用程序, 还支持 Microsoft SQL Server, IBM DB2 和 Oracle 数据库。当然, 本书为 C# 基础教程, 教材中涉及的程序均可在 Visual Studio 2005 以上版本中调试与运行。

Visual Studio 2012 最简单的窗口结构如图 1-2 所示。



图 1-2 Visual Studio 2012 窗口结构界面



## ► 任务实施

### 任务 1-1 Visual Studio 2012 的安装与启动。

本书以 Visual Studio 2012 的安装为例,讲述 Visual Studio 的安装过程。如果下载的安装文件是 MSI 格式或 EXE 格式,对其直接双击即可开始安装。如果是 ISO 光盘镜像格式,则可以用虚拟光驱软件(比如 UltraISO、Daemon Tools 等)把它载入到虚拟光驱中,然后打开虚拟光驱,找到 Setup.exe 程序,双击该程序可启动安装过程。安装程序启动窗口欢迎界面如图 1-3 所示。

然后,出现安装位置选择和许可条款选项,如图 1-4 所示,点击“...”按钮可选择自定义安装位置。选择完成后,点击“下一步”按钮。



图 1-3 Visual Studio 2012 安装程序欢迎界面



图 1-4 Visual Studio 2012 安装位置选择

在如图 1-5 所示的可选功能界面中,选择所需要的功能模块,在选择完成后,点击“安装”按钮开始安装。

整个安装过程非常简单,但是需要花费一些时间,请耐心等待。安装完成后,选择“开始菜单”→“所有程序”→“Microsoft Visual Studio 2012”→“Visual Studio 2012”菜单命令即可启动。

首次启动 Visual Studio 2012 需要进行默认环境设置,如图 1-6 所示;然后进入 Visual Studio 2012 的开发界面(如图 1-2 所示)。其中,“解决方案资源管理器”面板是一个非常重要的工具,它管理解决方案中的所有资源。在 Visual Studio 中,一个项目可以编译产生一个程序集(一个 .exe 文件或 .dll 文件),一个解决方案可以包含多个项目,多个项目共同构成一个较大的应用程序。

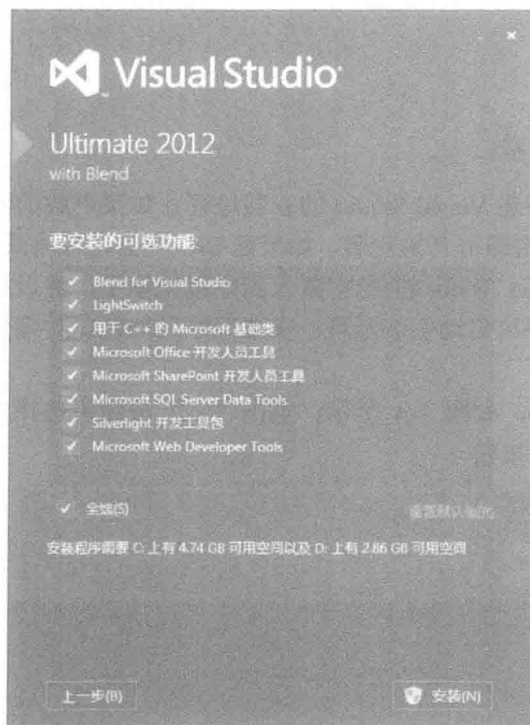


图 1-5 Visual Studio 2012 安装可选功能选择



图 1-6 Visual Studio 2012 设置默认环境

## ► 知识拓展

以上介绍了程序设计语言的概念与程序设计语言的分类,知道了 C# 是一种功能强大且非常流行的一种程序开发语言,而 .NET Framework 则是支持多种开发语言进行开发的基础框架,正是 .NET Framework 的存在,才能实现“跨语言”开发。Visual Studio 是由微软公司开发的一款功能强大的集成开发环境,利用这个开发环境和 C# 语言就可以开发出满足用户需求的软件。

在安装 Visual Studio 后,我们通常还会选择安装 MSDN(Microsoft Developer Network)。MSDN 库是微软公司向软件开发者提供的一种信息服务。MSDN 实际上是一个以 Visual Studio 和 Windows 平台为核心整合的虚拟开发社区,包括技术文档、在线电子教程、网络虚拟实验室、微软产品下载(几乎包含全部的操作系统、服务器程序、应用程序和开发程序的正式版和测试版,还包括各种驱动程序开发包和软件开发包)、Blog、BBS、MSDN WebCast、与 CMP 合作的 MSDN 杂志等一系列服务。

有兴趣的读者可以将 MSDN 安装到计算机中,尝试着做一些练习。

## ► 归纳总结

本节所介绍的内容是进行 C# 开发的第一步工作,认识 .NET 和 C#,了解 C# 的开发环境 VS 2012,是我们进一步深入学习的前提条件。