

21世纪大学计算机规划教材

# C#

## 大学实用教程

匡 松 张淮鑫 主编

杜小丹 王 超 何 嘉 刘永红 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



21 世纪大学计算机规划教材

# C#大学实用教程

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
|     | 匡 松 | 张淮鑫 | 主 编 |     |
| 杜小丹 | 王 超 | 何 嘉 | 刘永红 | 副主编 |
| 廖柯熹 | 夏雪梅 | 罗 琴 | 鄢 涛 |     |
| 古沐松 | 魏 春 | 周 蓓 | 徐 静 | 编 委 |

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书介绍 C#程序设计的基础知识及语法、C#面向对象程序设计、基于 Windows Forms 的应用程序开发和 ADO.NET 数据库应用开发等内容。以编写简单 Windows 应用程序的可视化编程方法为切入点, 全书采用语法介绍和实例驱动的方式进行讲解, 由浅入深, 层层引导, 以案例带动知识点的学习, 并注意保证知识的相对完整性和系统性, 使读者快速掌握 C#, 提高程序设计技能。本书为教师免费提供教学资料包(含电子课件及习题解答)。

本书可作为高等院校计算机及相关专业教材, 也可作为工程技术人员的自学用书。

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#大学实用教程 / 匡松, 张淮鑫主编. —北京: 电子工业出版社, 2009.7  
21 世纪大学计算机规划教材  
ISBN 978-7-121-09059-2

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

中国版本图书馆 CIP 数据核字(2009)第 096809 号

策划编辑: 章海涛

责任编辑: 章海涛

印 刷: 北京京师印务有限公司  
装 订:

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: 787×1 092 1/16 印张: 22 字数: 563 千字

印 次: 2009 年 7 月第 1 次印刷

印 数: 5 000 册 定价: 29.80 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。

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

服务热线: (010) 88258888。

# 前 言

从 FORTRAN 语言开始，人类发明了众多的高级计算机程序设计语言。到目前为止，C/C++、Java 和 C#代表了最主流的三种程序设计语言。其中，C#程序设计语言推出时间最晚，但兴起的速度最快，已成为我们值得学习和研究的程序设计语言之一。

## 1. C#是最近 20 年软件开发技术积累的成果

20 年前，程序设计的主要方法是，调用操作系统提供的各种功能元素，以实现比较特殊的功能。面向对象程序设计方法兴起之后，Visual Basic 提供了一套功能库来封装操作系统提供的功能，Visual C++提供了程序库来封装操作系统提供的功能。各种程序设计语言都各自维护了一套专用的功能库。因而程序员使用一种新的语言，就需要重新学习一套新的功能库，苦不堪言，造成大量的人力浪费。.NET Framework 系统库便是在这种背景下开发出来的，使各种程序设计语言都可以通过调用这套库来实现自己的功能。同时，在.NET Framework 上可以开发传统型应用程序，也可以开发网站类应用，满足了当代程序设计的需要。这使得 C#——这个专为 .NET Framework 开发的程序设计语言——很快成为了主流的程序设计语言。

## 2. C#非常适合作为程序设计学习语言

C#是非常精巧的面向对象程序设计语言，学习它，容易理解面向对象开发的基本方法。

C#主要吸收了 C++和 Java 等语言的经验和教训，是目前最成熟的面向对象程序设计语言。一旦掌握了 C#语言，利用对比学习法，可以轻松学会 C++和 Java 语言。

C#发明时，考虑了各类程序开发的需要，用它既可以进行传统的 Windows 程序设计，也可以开发大型数据库相关类程序，结合 ASP.NET 技术还可以开发 Web 网站。C#语言的适应面特别广，因此投入精力学好 C#语言是非常值得的。

## 3. 推荐的 C#程序设计学习路径

根据近几年 C#程序设计的教学经验，结合 C#程序设计的特点，我们设计了一个基本的 C#程序设计学习路径，也是本书的内容安排顺序，全书按这个顺序分为五部分。

**第一部分：C#程序设计基础**，包括第 1 章和第 2 章，让读者了解 C#语言的特点，同时让读者学会 C#开发环境的安装，并掌握 C#程序设计的基本流程。为后面一步一个台阶地学习基本概念和实践打下基础。

**第二部分：简单 C#程序设计**，包括第 3~7 章，重在掌握 C#程序设计的基本语法。任何一种程序设计语言，变量的定义和使用，顺序、选择和循环三种基本结构，程序模块（函数）的定义和使用等都是最基本的语法，C#语言也不例外。打好基础，熟悉语法，正确地建立基本的程序设计概念，以后将处处受益。

**第三部分：面向对象 C#程序设计**。面向对象程序设计是 C#语言的精华，第 8~13 章重点介绍 C#语法知识。从建立类和对象的概念开始，到继承，到方法，构建完整面向对象程序设计知识库；同时结合 C#的特点，介绍了 C#特有的属性、委托和事件，以及接口等语法特性。而这些特性，不但是现代程序设计语言的精华，也是编写 Windows 程序的基础。

**第四部分：基本 Windows 程序设计。**有了前面的基础，我们可以用 C#语言建立起一整套程序设计基本库 .NET Framework，可以轻松地通过拖放基本库中的对象来设置属性，进而简化程序设计。第 14~18 章主要介绍这套 Windows 程序设计库的使用。不难发现，有了面向对象程序设计的基础，当我们来使用这一套现成的类库时，Windows 程序设计就变得非常简单了。

**第五部分：专业程序设计基础**，包括第 19~20 章。学会使用 Windows 基本类库之后，我们编写较大规模的比较专业的应用程序时，需要用到数据库开发技术和应用程序的部署。因此，这一部分通过介绍和专业数据的连接及应用方法为读者打下良好的基础，同时通过一个简单应用安装实例，使读者明白一个应用程序软件安装包的开发过程。

在教学中，可以按照上述过程进行实施。根据学生基础不同，教学时间不同，专业需求不同，可以在这个基础上增加学习材料，或者减少部分学习内容，供学生自学使用，以达到因材施教的教学效果。

#### 4. 后续的学习建议

C#语言及其平台是众多优秀软件工程师的智慧结晶。读者可以根据自己的职业发展需要，以及毕业设计或者其他课程的学习需要，还有自己的研究兴趣来选择学习。一般来说，有三个学习方向。

① 专业 Windows 程序设计。C#正在逐步替代 Visual Basic、Visual FoxPro、Delphi 和 Visual C++，成为 Windows 上应用程序的首选语言。中大型的 Windows 程序设计的积累，需要一定的项目经验，读者一边学习更专业的类库知识，一边实践积累项目开发经验。

② 专业 Web 程序设计。如前面所说，C#是建立在一个统一的开发平台上的，除了 Windows 程序，通过学习 ASP.NET 技术，还能开发专业 Web 网站，而且不必从头学起，因为我们已经具备了基本的 C#语法。

③ 研究 .NET Framework 库。如果希望成为高级开发人员，研究一套运行完美的库，体会大型程序架构设计精髓，是一条捷径。这会让读者积累大型程序建构的经验，以后即使模仿编写，也会使程序的稳定性大增。在研究的同时，如果结合面向对象程序设计方法，学些软件工程课程会更好。

#### 5. 本书编写团队

本书由匡松、张淮鑫主编，杜小丹、王超、何嘉、刘永红担任副主编，廖柯熹、夏雪梅、罗琴、鄢涛、古沐松、魏春、周蓓、徐静等参加编写。虽然本书体现了我们近几年 C# 程序设计教学经验的积累，但因为知识面和开发经验有限，书中难免存在不足之处，请使用本书的老师和读者不吝指正。我们的 E-mail 是 beone2000@126.com。

本书为任课教师提供配套的教学资源（包含电子课件及习题解答），需要者可登录到华信教育资源网（<http://www.huaxin.edu.cn> 或 <http://www.hxedu.com.cn>），注册之后免费下载，或发邮件到 [unicode@phei.com.cn](mailto:unicode@phei.com.cn) 咨询。

作 者

# 目 录

|                                   |    |
|-----------------------------------|----|
| <b>第 1 章 C#简介</b>                 | 1  |
| 1.1 什么是.NET Framework             | 1  |
| 1.2 什么是 C#                        | 2  |
| 1.2.1 C#概述                        | 2  |
| 1.2.2 C#的主要功能                     | 3  |
| 1.2.3 C#的主要特点                     | 3  |
| 1.3 Visual Studio 2008 简介         | 5  |
| 1.3.1 环境要求                        | 5  |
| 1.3.2 安装步骤                        | 5  |
| 1.3.3 Visual Studio 2008 新功能介绍    | 7  |
| 1.4 如何学习 C#语言                     | 9  |
| 1.5 小结                            | 10 |
| 练习 1                              | 10 |
| <b>第 2 章 编写简单的 C#程序</b>           | 12 |
| 2.1 Visual Studio 2008 开发环境       | 12 |
| 2.1.1 Visual Studio 2008 常用功能窗口介绍 | 12 |
| 2.1.2 配置设置                        | 18 |
| 2.1.3 类设计器                        | 21 |
| 2.1.4 帮助系统                        | 22 |
| 2.2 编写控制台 Hello Word 程序           | 26 |
| 2.3 创建 Windows Forms 应用程序         | 28 |
| 2.4 小结                            | 31 |
| 练习 2                              | 32 |
| <b>第 3 章 变量和表达式</b>               | 34 |
| 3.1 C#的基本语法                       | 34 |
| 3.1.1 C#程序结构                      | 34 |
| 3.1.2 C#程序入口                      | 36 |
| 3.1.3 程序区块                        | 37 |
| 3.1.4 C#程序的注释方法                   | 38 |
| 3.1.5 命名空间                        | 39 |
| 3.2 变量                            | 41 |
| 3.2.1 变量的命名                       | 41 |
| 3.2.2 数据类型                        | 42 |
| 3.2.3 变量的声明                       | 44 |
| 3.3 常量                            | 45 |
| 3.4 表达式                           | 47 |

|       |                       |    |
|-------|-----------------------|----|
| 3.4.1 | 操作符 .....             | 47 |
| 3.4.2 | 算术表达式 .....           | 48 |
| 3.4.3 | 赋值表达式 .....           | 50 |
| 3.5   | 小结 .....              | 51 |
| 练习 3  | .....                 | 51 |
| 第 4 章 | 基本程序流程控制 .....        | 54 |
| 4.1   | 布尔逻辑 .....            | 54 |
| 4.1.1 | 位运算符 .....            | 54 |
| 4.1.2 | 关系运算 .....            | 56 |
| 4.1.3 | 逻辑表达式 .....           | 57 |
| 4.2   | 分支 .....              | 58 |
| 4.2.1 | if 语句 .....           | 58 |
| 4.2.2 | 三元运算符 .....           | 62 |
| 4.2.3 | switch 语句 .....       | 63 |
| 4.2.4 | break 语句 .....        | 65 |
| 4.2.5 | default 语句 .....      | 67 |
| 4.3   | goto 语句 .....         | 68 |
| 4.4   | 小结 .....              | 68 |
| 练习 4  | .....                 | 70 |
| 第 5 章 | 循环控制 .....            | 73 |
| 5.1   | 程序中的循环结构 .....        | 73 |
| 5.2   | 位运算符 .....            | 74 |
| 5.2.1 | while 循环语句的语法格式 ..... | 74 |
| 5.2.2 | while 循环语句的应用 .....   | 74 |
| 5.3   | do 循环 .....           | 76 |
| 5.3.1 | do 循环语句的语法格式 .....    | 76 |
| 5.3.2 | do 循环语句的应用 .....      | 76 |
| 5.4   | for 循环 .....          | 78 |
| 5.4.1 | for 循环语句的语法格式 .....   | 78 |
| 5.4.2 | for 循环语句的应用 .....     | 78 |
| 5.5   | 循环的嵌套 .....           | 81 |
| 5.6   | 循环的中断 .....           | 84 |
| 5.6.1 | break 语句 .....        | 84 |
| 5.6.2 | continue 语句 .....     | 85 |
| 5.6.3 | goto 语句 .....         | 86 |
| 5.7   | 小结 .....              | 87 |
| 练习 5  | .....                 | 87 |
| 第 6 章 | 复杂的变量 .....           | 91 |
| 6.1   | 类型转换 .....            | 91 |
| 6.2   | 复杂的变量类型 .....         | 94 |

|                             |            |
|-----------------------------|------------|
| 6.2.1 枚举 .....              | 94         |
| 6.2.2 结构 .....              | 96         |
| 6.2.3 数组 .....              | 98         |
| 6.3 字符串的处理 .....            | 101        |
| 6.3.1 创建字符串 .....           | 101        |
| 6.3.2 操作字符串 .....           | 101        |
| 6.4 小结 .....                | 103        |
| 练习 6 .....                  | 104        |
| <b>第 7 章 函数 .....</b>       | <b>106</b> |
| 7.1 定义和使用函数 .....           | 106        |
| 7.1.1 返回值 .....             | 107        |
| 7.1.2 参数 .....              | 108        |
| 7.2 变量的作用域 .....            | 112        |
| 7.3 Main()函数 .....          | 116        |
| 7.4 委托简介 .....              | 116        |
| 7.5 小结 .....                | 118        |
| 练习 7 .....                  | 118        |
| <b>第 8 章 面向对象程序设计 .....</b> | <b>121</b> |
| 8.1 什么是面向对象编程 .....         | 121        |
| 8.2 类和对象 .....              | 122        |
| 8.3 对象之间的关系 .....           | 123        |
| 8.4 继承 .....                | 123        |
| 8.5 多态性 .....               | 126        |
| 8.6 接口 .....                | 128        |
| 8.7 事件 .....                | 129        |
| 8.8 引用类型和值类型 .....          | 130        |
| 8.9 小结 .....                | 132        |
| 练习 8 .....                  | 134        |
| <b>第 9 章 类和对象 .....</b>     | <b>136</b> |
| 9.1 C#中类的定义和使用 .....        | 136        |
| 9.2 类成员 .....               | 138        |
| 9.3 构造函数和析构函数 .....         | 141        |
| 9.3.1 构造函数 .....            | 142        |
| 9.3.2 析构函数 .....            | 143        |
| 9.3.3 this 访问 .....         | 144        |
| 9.4 静态成员和实例成员 .....         | 145        |
| 9.5 常数和字段 .....             | 146        |
| 9.6 System.Object 类 .....   | 148        |
| 9.6.1 装箱 .....              | 148        |
| 9.6.2 拆箱 .....              | 149        |

|                                           |            |
|-------------------------------------------|------------|
| 9.7 小结 .....                              | 149        |
| 练习 9.....                                 | 150        |
| <b>第 10 章 方法 .....</b>                    | <b>152</b> |
| 10.1 方法的定义与调用.....                        | 152        |
| 10.1.1 方法的定义 .....                        | 152        |
| 10.1.2 方法的调用 .....                        | 154        |
| 10.1.3 方法的返回类型 .....                      | 156        |
| 10.2 虚拟方法.....                            | 157        |
| 10.3 重写方法.....                            | 158        |
| 10.4 参数.....                              | 160        |
| 10.5 方法重载.....                            | 163        |
| 10.6 小结.....                              | 165        |
| 练习 10 .....                               | 166        |
| <b>第 11 章 属性 .....</b>                    | <b>168</b> |
| 11.1 属性的定义.....                           | 168        |
| 11.2 属性的使用 .....                          | 169        |
| 11.3 索引器.....                             | 175        |
| 11.4 小结.....                              | 177        |
| 练习 11 .....                               | 178        |
| <b>第 12 章 委托和事件 .....</b>                 | <b>180</b> |
| 12.1 委托.....                              | 180        |
| 12.1.1 C# "Hello World!"委托版 .....         | 181        |
| 12.1.2 委托的特点 .....                        | 182        |
| 12.1.3 C# "Hello World!"委托扩展版之给方法签名 ..... | 183        |
| 12.1.4 C# "Hello World!"委托扩展版之自定义方法 ..... | 184        |
| 12.2 多路广播.....                            | 185        |
| 12.2.1 多路广播委托的组合 .....                    | 185        |
| 12.2.2 C# "Hello World!"委托扩展版之多路广播 .....  | 186        |
| 12.3 委托的使用.....                           | 188        |
| 12.3.1 委托中的命名方法 .....                     | 188        |
| 12.3.2 命名方法简单示例 .....                     | 189        |
| 12.4 委托的相等性.....                          | 190        |
| 12.5 委托综合示例.....                          | 190        |
| 12.6 事件.....                              | 193        |
| 12.6.1 事件概述 .....                         | 193        |
| 12.6.2 事件的声明 .....                        | 194        |
| 12.6.3 简单事件声明示例 .....                     | 195        |
| 12.6.4 如何引发事件 .....                       | 196        |
| 12.6.5 如何订阅事件 .....                       | 196        |
| 12.6.6 是否所有委托都可以应用到事件中 .....              | 197        |

|               |                                 |            |
|---------------|---------------------------------|------------|
| 12.7          | EventHandler .....              | 198        |
| 12.7.1        | 如何声明事件访问器 .....                 | 198        |
| 12.7.2        | 创建响应事件的控件 .....                 | 199        |
| 12.7.3        | 创建响应事件的控件 .....                 | 199        |
| 12.8          | GUI 事件 .....                    | 201        |
| 12.8.1        | 鼠标单击事件 .....                    | 201        |
| 12.8.2        | KeyPress 事件 .....               | 203        |
| 12.8.3        | 鼠标按下、释放和拖动事件 .....              | 205        |
| 12.9          | 小结 .....                        | 206        |
| 练习 12         | .....                           | 206        |
| <b>第 13 章</b> | <b>接口和异常处理 .....</b>            | <b>210</b> |
| 13.1          | 接口 .....                        | 210        |
| 13.1.1        | 接口和类的区别 .....                   | 211        |
| 13.1.2        | 接口和类的选择使用 .....                 | 211        |
| 13.1.3        | 定义接口 .....                      | 211        |
| 13.1.4        | 接口成员 .....                      | 213        |
| 13.1.5        | 接口成员的全权名 .....                  | 216        |
| 13.2          | 接口的实现 .....                     | 216        |
| 13.3          | 接口的查询 .....                     | 218        |
| 13.3.1        | 使用 is 运算符实现查询 .....             | 218        |
| 13.3.2        | 使用 as 运算符实现查询 .....             | 219        |
| 13.4          | 异常处理 .....                      | 220        |
| 13.5          | 小结 .....                        | 223        |
| 练习 13         | .....                           | 223        |
| <b>第 14 章</b> | <b>Windows 编程基础 .....</b>       | <b>226</b> |
| 14.1          | 控件 .....                        | 226        |
| 14.1.1        | 控件的通用属性 .....                   | 226        |
| 14.1.2        | 窗体 .....                        | 227        |
| 14.1.3        | 控件的创建 .....                     | 232        |
| 14.1.4        | 控件的事件 .....                     | 232        |
| 14.2          | Button 控件 .....                 | 233        |
| 14.3          | Label 和 LinkLabel 控件 .....      | 234        |
| 14.4          | TextBox 控件 .....                | 236        |
| 14.5          | RichTextBox 控件 .....            | 237        |
| 14.6          | RadioButton 和 CheckBox 控件 ..... | 238        |
| 14.7          | ListBox 和 ComboBox 控件 .....     | 240        |
| 14.8          | ListView 控件 .....               | 243        |
| 14.9          | 小结 .....                        | 247        |
| 练习 14         | .....                           | 248        |

|                                |     |
|--------------------------------|-----|
| 第 15 章 Windows 表单的高级功能 .....   | 251 |
| 15.1 菜单栏 .....                 | 251 |
| 15.1.1 菜单控件的使用 .....           | 252 |
| 15.1.2 什么是快捷菜单 .....           | 253 |
| 15.1.3 快捷菜单的创建 .....           | 253 |
| 15.2 工具栏 .....                 | 255 |
| 15.2.1 工具栏的创建 .....            | 255 |
| 15.3 状态栏 .....                 | 257 |
| 15.3.1 状态栏控件的使用 .....          | 257 |
| 15.4 常用控件 .....                | 258 |
| 15.4.1 滚动条控件 .....             | 258 |
| 15.4.2 进程条控件 .....             | 259 |
| 15.4.3 跟踪条控件 .....             | 261 |
| 15.4.4 计时器控件 .....             | 262 |
| 15.4.5 模拟跑表计时器 .....           | 262 |
| 15.5 单文档和多文档应用程序 .....         | 264 |
| 15.5.1 基于对话框的应用程序 .....        | 264 |
| 15.5.2 单文档应用程序 .....           | 265 |
| 15.5.3 多文档应用程序 .....           | 265 |
| 15.5.4 多文档应用程序示例 .....         | 266 |
| 15.5.5 MDI 文档编辑器示例 .....       | 268 |
| 15.6 小结 .....                  | 269 |
| 练习 15 .....                    | 270 |
| 第 16 章 通用对话框 .....             | 273 |
| 16.1 消息框 .....                 | 273 |
| 16.1.1 MessageBox 显示信息 .....   | 274 |
| 16.1.2 MessageBox 接受用户输入 ..... | 275 |
| 16.2 对话框的使用 .....              | 276 |
| 16.3 文件对话框 .....               | 277 |
| 16.3.1 打开文件对话框 .....           | 277 |
| 16.3.2 保存文件对话框 .....           | 279 |
| 16.4 打印 .....                  | 280 |
| 16.5 打印预览 .....                | 282 |
| 16.6 字体对话框和颜色对话框 .....         | 284 |
| 16.6.1 字体对话框 .....             | 284 |
| 16.6.2 颜色对话框 .....             | 285 |
| 16.7 浏览文件夹对话框 .....            | 285 |
| 16.8 小结 .....                  | 286 |
| 练习 16 .....                    | 286 |

第 17 章 容器类控件 ..... 288

17.1 面板控件..... 288

17.2 分组控件..... 290

17.3 分页控件..... 292

17.4 小结..... 296

练习 17 ..... 296

第 18 章 输入/输出类 ..... 298

18.1 Stream 类 ..... 298

18.2 File 类和 FileInfo 类 ..... 299

18.3 StreamReader 类和 StreamWriter 类 ..... 302

18.4 BinaryReader 类和 BinartWriter 类 ..... 306

18.5 小结..... 310

练习 18 ..... 311

第 19 章 数据库应用程序 ..... 313

19.1 ADO.NET ..... 313

19.2 DataSet 的概念及作用 ..... 315

19.2.1 DataSet 和 DataReader 的区别 ..... 317

19.2.2 DataTable..... 318

19.2.3 BindingSource ..... 320

19.2.4 BindingNavigator ..... 322

19.2.5 使用 BindingNavigator 完成数据导航功能 ..... 323

19.3 数据访问的基本步骤..... 324

19.3.1 简单数据访问示例 ..... 327

19.3.2 DataGridView 控件..... 329

19.3.3 利用 DataGridView 控件实现单元格合并 ..... 331

19.4 小结..... 334

练习 19 ..... 335

参考文献 ..... 339

# 第 1 章

## C# 简介

本章主要了解什么是 .NET Framework、什么是 C#，并简要介绍 Visual Studio 2008 和如何学习 C#语言。

| 关键点                   | 基本要求和学习方法介绍               | 难度指数  |
|-----------------------|---------------------------|-------|
| .NET 框架               | 了解.NET 框架的工作原理            | ★★★☆☆ |
| C#简介                  | 掌握 C#语言的主要特征              | ★★★☆☆ |
| 安装 Visual Studio 2008 | 掌握如何安装 Visual Studio 2008 | ★★★☆☆ |

### 1.1 什么是.NET Framework

介绍 C#之前，必须了解 C#运行的环境.NET Framework。

在 Windows 之前的时代，DOS 或者 UNIX，都只是提供普通的函数编程接口，所有应用程序都直接调用操作系统的函数接口。Windows 刚刚发布的时候，也是通过 Windows API 提供系统函数调用，当时编写 Windows 应用程序时也需要紧密研究 Windows API。

但这种软件开发的方法属于面向过程的结构化程序设计方式，伴随面向对象程序设计开发技术的成熟，在 Windows API 上提供一套面向对象程序设计类库，使得编写 Windows 应用程序变得更为简单。

一时间，Windows 上的开发工具，Visual Basic、Visual C++、Delphi、Visual FoxPro、PowerBuilder 等都各自提供了一套开发库。虽然这些开发库大大简化了应用程序的编写，我们只需要拖放控件即可实现应用程序的界面设计，但这些开发库各自成为一个体系，相互之间差异颇大，各有特长，应用程序设计人员在开发不同的应用程序时，需要使用不同的开发工具，这就需要重新学习一套库的使用，既浪费时间，也使得项目开发的进度变慢。

如图 1-1 所示，聪明的工程师们便想到，吸收这些控件库的优点，开发出一套中间库，让所有的开发语言都调用一套相同的库，各种开发语言都使用统一的开发库，这样可以简化开发语言的设计 and 应用程序的编写。这就是 .NET Framework 这套面向对象应用程序库开发的最初目的。

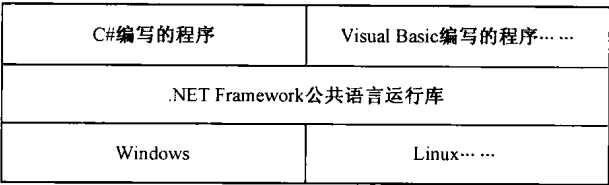


图 1-1 .NET Framework 所处的中间位置

同时，这套库屏蔽了应用程序和操作系统之间的直接连接，操作系统可以根据自己的需要进化，变得更加稳定、强壮和安全，这样操作系统的开发也可以从这套库的运行中得益。

并且，在开发这套 .NET Framework 时充分考虑到未来发展的需要，只要新出现一种开发模式，便在库中添加新的应用程序开发模式即可。这几年，Web 网站成为一个火热的新课题，人们就可以轻松地通过 ASP.NET 运行库，编写差不多的网络类应用程序。这使得人们发现，伴随 .NET Framework 而生的 C#语言，既能编写控制台应用，又能编写 Windows 应用，还能编写 ASP.NET 网站类应用，几乎是一种全能语言。学习好了 C#，就可以轻松进入各个领域，这大大减轻了程序员的学习难度，提升了他们的技能空间。

所以，吸收了多种开发语言优点的 C#，以非常快的速度普及了。

## 1.2 什么是 C#

21 世纪初，Microsoft 推出了新一代程序开发环境 Visual Studio.NET。为了配合 .NET 应用程序的开发，促进 .NET 战略的发展，Microsoft 同时推出了该环境下的主要编程语言 C#（读作 C Sharp）。C#语言是针对 .NET 框架平台开发的一种新型编程语言，完全建立在.NET 框架平台基础之上。Microsoft 宣称，C#是开发.NET 框架应用程序的最好语言。

### 1.2.1 C#概述

C#是 Visual Studio.NET 中包含的一种全新的开发语言。它从 C/C++语言发展而来，具有与 C/C++语言相似的语法，继承和发扬了 C/C++语言的优点。同时，由于是 Microsoft 公司的产品，它又同 Visual Basic 一样简单，在创建图形用户界面和事件驱动的应用程序方面的容易程度可以媲美 Visual Basic。对于 Web 开发而言，C#又像 Java/C++语言一样，具有强大的功能和灵活性，使用起来更加方便和直观。

C#语言虽然从 C/C++语言发展而来，但是它摒弃了 C/C++语言中容易出错的指针、多重继承等概念，是一种完全面向对象、类型安全的语言，与使用新的.NET 框架的其他许多语言兼容，允许与具有 C/C++语言风格的需要传递指针类型参数的 API 进行交互操作，允许 C#语言组件与其他语言组件间的交互操作。C#是一种完全建立在.NET 框架平台之上的高级语言，其所有类型都是从 .NET Framework 提供的统一类库中派生的，并且所有的功能实现都依赖于这个类库。C#中没有多重继承并且没有模块类。C#中的简单类型仅仅是底层数据类型是直接映射。C#中的类型不是由语言本身确定的，而是由公共类型系统确定的。例如，在 C#中，枚举、属性和事件都已内建到语言中，虽然所有面向 .NET Framework 的语言都可以使用该框架，但 C#是完全针对这个框架的，它可以充分使用这个框架所提供的类、接口和对象。

## 📖 延伸学习——C#与 Java 比较

C#语言与 Java 语言有诸多相似之处，只是在细节上有一些差别。实际上，C#与 Java 的主要区别不是在语言本身，而是所执行的平台。Java 程序需要一个运行环境 JRE (Java Runtime Environment) 来执行代码，但 JRE 只限于在 Java 语言中使用。C#语言也需要一个运行环境 CLR (Common Language Runtime)，但是 CLR 提供了对多种语言的支持。Java 源代码可以被编译成字节代码的一种中间状态，然后由已提供的虚拟机来执行这些代码。C#代码也被编译成一种中间代码，称为中间语言 (IL)。但是 IL 代码则被传输到由 CLR 管理的执行进程上，然后通过 CLR 的 JIT 编译器编译成本地代码执行。另外，C#语言还提供了一些 Java 语言中没有的功能，如运算符重载、封箱和装箱以及方法隐藏等。

由于 C#语言一开始就是完全按照面向对象的思想来设计的，而且使用的全部是 .NET Framework 定义的语法规则，不存在考虑与 .NET 之前版本兼容的问题。同时，C#语言作为一种高级语言标准，其基本的内部实现形式是公开的，容易被更多种平台接受。因此，C#语言给人的感觉是结构清晰、语法简洁、方便直观和应用广泛。

### 1.2.2 C#的主要功能

C#是一种最新的、面向对象的编程语言，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，Microsoft .NET 提供了一系列的工具和服务，来最大程度地开发利用计算与通信领域。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——无论是高级的商业对象还是系统级的应用程序。

C#的功能主要表现在以下几方面：

- ◎ 设计 Windows 应用程序。
- ◎ 自定义 Windows 控制库。
- ◎ 设计控制台应用程序。
- ◎ 设计智能设备应用程序。
- ◎ 设计 ASP.NET Web 应用程序。
- ◎ 设计 ASP.NET Web 服务。
- ◎ 设计 ASP.NET 移动 Web 应用程序。
- ◎ 自定义 Web 控件库。

ASP.NET 正是以 C#为基础所开发出来的控件、标记。在中间语言的领域里，C#是最具亲和力的一种语言，它拥有 C 语言和 Java 语言的主要特点，同时拥有功能强大的函数库和方便的模板等，是目前最理想的语言之一。

### 1.2.3 C#的主要特点

C#语言几乎综合了目前高级语言的优点，具有以下主要特点。

(1) 简洁的语法

C#语言与 Java 语言一样，使用了统一的操作符，淘汰了 C++语言中复杂的表达式和伪关

键字，使用最简单、最常见的形式进行描述。

## (2) 卓越的面向对象设计

C#语言完全按照面向对象的思想来设计，因此具有面向对象所应具有的一切特性，即：封装性、继承性和多态性。

C#语言只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。在 C# 语言中，每种类型都是一个对象，因此不存在全局函数、全局变量和全局常数等概念。所有常量、变量、属性、方法、索引和事件等都必须封装在类中，从而使代码具有更好的可读性，也减少了发生命名冲突的可能。

## (3) 与 Web 的紧密结合

在 C#中，对于复杂的 Web 编程和其他网络编程更像是对本地对象进行操作，从而简化了大规模、深层次的分布式开发。用 C#语言构建的组件能够方便地为 Web 服务，并可以通过 Internet 被运行在任何操作系统上的任何语言所调用。

## (4) 完全的安全性和错误处理

语言的安全性和错误处理能力是衡量一种语言是否优秀的重要依据。C#语言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。

在默认情况下，从 Internet 和 Intranet 下载的代码都不允许访问任何本地文件和资源；C#语言不允许使用未初始化的变量，并提供了便捷检查和溢出检查等功能。内存管理中的垃圾回收机制大大减轻了开发人员对内存管理的负担。

## (5) 版本处理技术

C#语言内置了版本控制功能，如对函数重载和借口的处理方式以及特性支持等，从而保证方便地开发和升级复杂的软件。

## (6) 灵活性和兼容性

在托管状态下，C#语言不能使用指针，而是用委托（Delegate）来模拟指针的功能。如果确实需要在类或者类的方法中使用指针，只需要声明这些内容为非安全即可。另外，虽然 C#语言不支持类的多继承，但是可以通过继承接口来实现多继承。

兼容性是指 C#语言允许与具有 C/C++语言风格的需要传递指针类型参数的 API 进行交互操作，允许 C#语言组件与其他语言组件间的交互操作。

## (7) 商业过程和软件实现的更好对应

企业的商业计划要付诸现实，必须在抽象的商业过程和实际的软件实现之间建立紧密地对应。大多数语言都很难做到这点。

C#语言允许类型定义的、扩展的元数据。这些元数据可以应用于任何对象。项目构建者可以定义领域特有的属性并把他们应用于任何语言元素，如类、接口等。

总之，C#是一种现代的面向对象语言，使程序员可以快速、便捷地创建基于 Microsoft .NET 平台的解决方案。这种框架使 C#组件可以方便地转化为 XML 网络服务，从而使任何平台的应用程序都可以通过 Internet 调用它。C#增强了开发者的效率，同时也致力于消除编程中可能导致严重结果的错误。C#使 C/C++程序员可以快速进行网络开发，同时也保持了开发者所需要的强大性和灵活性。

## 1.3 Visual Studio 2008 简介

Visual Studio 2008 是一套完整的开发工具集，其中的 C# 是使用最广泛的开发语言，用于开发 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C# 和 Visual J# 都使用相同的集成开发环境（Integrated Development Environment, IDE），方便用户快速有效地建立应用程序。

### 1.3.1 环境要求

Visual Studio 2008 是一个比较庞大的系统软件，它对计算机的要求较高。下面给出了安装和运行 Visual Studio 2008 的系统需求。

- ⊙ 处理器：配备 1.6 GHz 或速度更快的处理器。
- ⊙ 内存：384 MB 或更大内存（Windows Vista 至少需要 768 MB）。
- ⊙ 硬盘空间：完全安装 Visual Studio 2008 大致需要 4 GB 的硬盘空间，再加上用户开发的项目文件占用的空间，最小需要 5 GB 以上的硬盘空间。
- ⊙ 显示设备：1024×768 及以上分辨率，16 位颜色。
- ⊙ 光驱：在安装 Visual Studio 2008 时，DVD 光驱是必需的。如果需要的所有组件都已安装在硬盘上，在开发过程中可以不再使用光驱。
- ⊙ 操作系统：建议使用 Windows XP 或更新版本的操作系统。

### 1.3.2 安装步骤

下面主要介绍在 Windows Vista 操作系统下安装 Visual Studio 2008 的步骤。在其他操作系统中安装 Visual Studio.NET 的过程大致相同。

（1）将 Visual Studio 2008 的安装光盘放入 DVD 光驱，自动出现如图 1-2 所示的界面。选择第 1 项“安装 Visual Studio 2008”，安装程序将对 Windows 组件进行更新。因为将要安装的 Visual Studio 2008 的组件与原有的 Windows 组件不匹配，系统要求进行更新，以便进行 Windows 的升级工作。



图 1-2 “Visual Studio 2008 安装程序”对话框