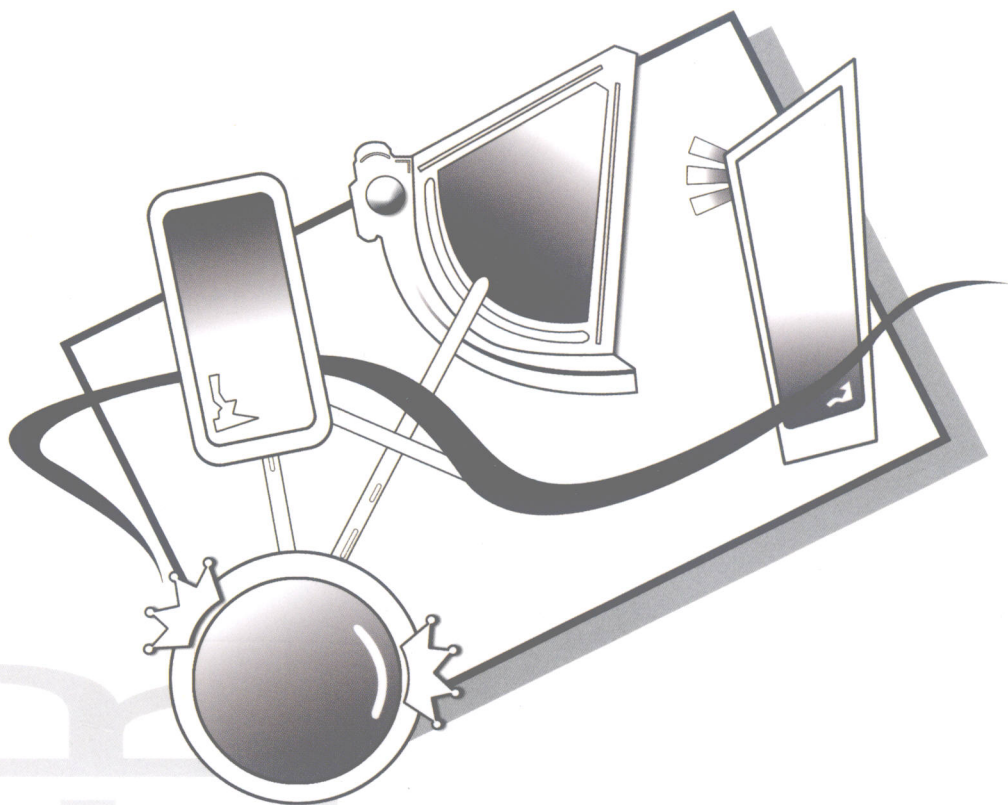


高等职业教育计算机专业推荐教材



网上免费提供  
电子教案

# Visual C#.NET

## 面向对象 程序设计教程

赵卫伟 何集体 刘瑞光 等编著

高等职业教育计算机专业推荐教材

# Visual C#.NET 面向对象 程序设计教程

赵卫伟 何集体 刘瑞光 等编著



机械工业出版社

本书重点讲述了C#语言关键字、语法和程序结构。书中不仅介绍了C#语言的数据类型、运算符和运算表达式、常量、变量、数组、程序顺序结构、选择结构以及循环结构等传统的程序基本元素,还讲述了类和面向对象的基本概念,并重点介绍了进行C#程序开发的集成开发环境的安装和使用、C#数组、C#的字符串、C#窗体开发中常用的对话框、图形、多媒体、文件、数据库访问技术等高级知识。

本书内容丰富、全面、系统,并列举了大量的程序实例,课后附有习题,不仅适合C#的初中级读者,还可帮助C和C++的用户顺利过渡到C#。本书既可作为高职高专学校计算机软件技术课程的教材,也可作为管理信息系统开发人员的技术参考书。

### 图书在版编目(CIP)数据

Visual C#.NET 面向对象程序设计教程/赵卫伟等编著. —北京:机械工业出版社,2006.5

(高等职业教育计算机专业推荐教材)

ISBN 7-111-18764-4

I. V... II. 赵... III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2006)第025370号

机械工业出版社(北京市百万庄大街22号 邮政编码 100037)

策 划:胡毓坚

责任编辑:李馨馨

责任印制:洪汉军

北京汇林印务有限公司印刷

2006年5月第1版·第1次印刷

184mm×260mm·14印张·342千字

0001—5000册

定价:20.00元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010)68326294

编辑热线电话:(010)88379739

封面无防伪标均为盗版

# 高等职业教育计算机专业推荐教材

## 编委会成员名单

主 任 王元元

编 委	丁跃潮	黄陈蓉	黄国兴
	李咏梅	逯燕玲	王爱梅
	奚李峰	杨世平	张桂芸

## 编者的话

根据有关部门对我国信息产业发展的客观需求及劳动力市场现状的调查,在计算机应用和软件专业领域培养技能型紧缺人才,是当务之急。近年来,不仅高等职业技术类院校,而且相当一部分本科类工程技术院校(特别是相当数量高等学校的二级学院、民办院校),都把招收和培养计算机专业技能型紧缺人才列为教育改革的重要举措。为一些院校提供“适时、适度、优选、优质”的计算机专业的高等职业教育系列教材,正是我们组织编写这套“高等职业教育计算机专业推荐教材”(以下简称“推荐教材”)的目标。“推荐教材”由四个模块的30多本教材组成。这些模块是:基础知识模块、程序设计模块、实用技术模块、实践模块。

这套“推荐教材”是“适时”的,因为它努力适应我国信息产业发展和劳动力市场的客观需求,适应计算机行业技术的现状,强调教学内容的先进性和实用性。这套教材十分关注信息技术的最新发展,突出本专业领域的新知识、新技术、新流程和新方法。其中程序设计模块和实用技术模块充分体现了这一特色,所涉及的19本教材既有基础的平台、语言,如《Linux操作系统》、《C语言程序设计与实践》,也有最新的《Visual C#.NET面向对象程序设计教程》、《XML实用教程》、《JSP应用教程》等工具,还有十分接近实际工作需要的《Oracle数据库应用教程》、《计算机网络管理》、《电子商务概论》等实用教材。

这套“推荐教材”是“适度”的,因为它不是简单地摒弃基础理论,而是注意强调理论联系实际,努力做到专业技能型人才能从中学学习到必要和相对系统的基础理论知识,把各种能力的培养和全面素质的提高放在首要的位置。“推荐教材”中基础知识模块的设置,充分体现了这一特色,它囊括了从数学基础、电子基础、硬件技术基础到系统软件基础、应用技术基础、网络技术基础、信息安全基础等10本教材。

这套“推荐教材”是“优选”的,因为充分考虑了现有高中毕业生的认知水平和已有知识,为学生提供适应劳动力市场需要和有职业发展前景的、模块化的教材体系。在学习内容、教学组织等方面留给教师和学生选择和创新的空間,便于教师组织和构建开放式的课程体系,适应学生个性化发展的需要,在灵活的模块化课程结构中自由发展。“推荐教材”的四个模块对重要内容都安排了看似重复的多种教材,供教师和学生去选择。例如,可以在《C语言程序设计与实践》、《Java程序设计教程》中任意选择一到两门,也可以在《ASP基础及应用教程》、《JSP应用教程》中任选一门。

这套“推荐教材”是“优质”的,因为它们的作者多数是从事高等职业教育的计算机专业教师,具有长期的计算机实际工作和教育工作经验。这套教材的优质,还体现在它的改革和创新精神上。其中《计算机电路基础》对传统的电路、模拟电路和数字电路课程教材作了重大的改变,《计算机组装与维修教程》则是一门纯实践的课程教材。我们欢迎使用这套教材的师生,指出教材中存在的问题并提出修改意见。

高等职业教育计算机专业推荐教材  
编委会

# 前 言

C# 是基于微软新一代操作平台 .NET、面向对象的全新的开发语言,它集中了其他语言的优点,具有 Visual C++ 的灵活性和 Visual Basic 的易用性,还具有 Java 的跨平台特性。它可以使程序员快速地编写各种基于 Microsoft .NET 平台的应用程序。由于 C# 面向对象的卓越设计功能,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。

本书从开发 C# 应用程序所用到的基本概念讲起,由浅入深,逐步介绍使用 C# 语言开发应用程序所必须掌握的知识。全书主要分为两部分:第一部分是第 1~6 章,主要介绍 C# 程序设计语言基础,包括 .NET 开发环境和 C# 语言的特点,C# 的集成开发环境,C# 语言的基本语法、语句和方法,以及面向对象编程的基本知识;第二部分是第 7~14 章,主要介绍 C# 程序开发中所使用的各种技术,包括数组的操作、字符串的操作、简单 Windows GUI 编程的操作、对话框的使用和操作、图形和多媒体的使用和操作、数据库访问和编程中的调试技术等内容。

本书在编写过程中注重对基础知识的掌握和运用,特别是对理论的理解和运用,为了更好地帮助读者理解这些知识,提高读者的编程能力,在讲述知识点时,都配合了典型的例子进行讲解。这些例子是经过精心挑选的、能够反映知识点的经典实例。为帮助读者巩固学过的知识,并提高运用知识解决实际问题的能力,在每章末尾有针对性地给出了习题。

本书适用于没有任何编程语言基础的初学者。但本书后面的一些章节也涉及到了一些高级专题,因此有一定 C# 语言基础的读者也可以把本书作为进阶的教材。

本书主要由赵卫伟、何集体、刘瑞光编写,参加编写及整理工作的还有张俊峰、王改梅、闫宏伟、田绪安、辛应志、向阳。

由于时间仓促,加之水平所限,书中不足之处在所难免,敬请读者批评指正。

为了配合本书的教学,机械工业出版社为读者提供了电子教案,读者可以在 [www.cmpbook.com](http://www.cmpbook.com) 上下载。

作 者

# 目 录

编者的话

前言

<b>第 1 章 Microsoft.NET 和 C# 语言简介</b> .....	1
1.1 Microsoft.NET 概述 .....	1
1.1.1 Microsoft.NET 的设计理念 .....	1
1.1.2 Microsoft.NET 的组成 .....	2
1.1.3 .NET 框架和 C# .....	2
1.2 C# 语言介绍 .....	2
1.2.1 C# 的语言特点 .....	3
1.2.2 C# 的运行模型 .....	4
1.3 小结 .....	5
1.4 习题 .....	5
<b>第 2 章 Visual Studio.NET 集成开发环境</b> .....	6
2.1 Visual Studio.NET 集成开发环境的安装 .....	6
2.1.1 硬件和系统需求 .....	6
2.1.2 安装必要的组件 .....	6
2.1.3 在 Windows XP 平台上安装 Visual Studio.NET .....	8
2.2 Visual Studio.NET 集成开发环境介绍 .....	10
2.2.1 启动 Visual Studio.NET 集成开发环境 .....	10
2.2.2 Visual Studio.NET 的窗口 .....	12
2.2.3 Visual Studio.NET 的菜单 .....	14
2.3 .NET 集成开发环境中的 C# 程序开发 .....	23
2.3.1 C# 应用程序的创建 .....	23
2.3.2 C# 程序的典型布局 .....	25
2.4 小结 .....	27
2.5 习题 .....	27
<b>第 3 章 C# 的基本语法</b> .....	28
3.1 词法结构 .....	28
3.1.1 标识符 .....	28
3.1.2 关键字 .....	28
3.2 数据类型 .....	29
3.2.1 值类型 .....	29
3.2.2 引用类型 .....	33
3.2.3 装箱和拆箱 .....	37
3.3 变量和常量 .....	38
3.3.1 变量 .....	38

3.3.2 常量 .....	39
3.4 运算符和表达式 .....	40
3.4.1 运算符的分类和优先级 .....	40
3.4.2 算术运算符和算术表达式 .....	41
3.4.3 赋值运算符和赋值表达式 .....	43
3.4.4 关系运算符和关系表达式 .....	43
3.4.5 逻辑运算符和逻辑表达式 .....	45
3.4.6 位运算符 .....	45
3.4.7 特殊运算符 .....	46
3.5 小结 .....	48
3.6 习题 .....	48
<b>第4章 C# 语句</b> .....	<b>50</b>
4.1 C#的基本语句 .....	50
4.1.1 块语句 .....	50
4.1.2 空语句 .....	50
4.1.3 标签语句 .....	50
4.1.4 声明语句 .....	51
4.1.5 表达式语句 .....	51
4.2 选择语句 .....	51
4.2.1 if 语句 .....	51
4.2.2 switch 语句 .....	54
4.3 循环语句 .....	55
4.3.1 for 循环语句 .....	55
4.3.2 foreach 循环语句 .....	57
4.3.3 while 循环语句 .....	57
4.3.4 do-while 语句 .....	58
4.4 跳转语句 .....	59
4.4.1 goto 语句 .....	59
4.4.2 break 语句 .....	59
4.4.3 continue 语句 .....	60
4.4.4 return 语句 .....	60
4.5 异常处理语句 .....	60
4.6 小结 .....	62
4.7 习题 .....	62
<b>第5章 C# 方法</b> .....	<b>65</b>
5.1 概述 .....	65
5.2 方法的声明 .....	65
5.3 方法中的参数 .....	66
5.3.1 值参数 .....	66



5.3.2	引用参数	67
5.3.3	输出参数	68
5.3.4	数组参数	68
5.3.5	变量持续周期和作用范围	69
5.3.6	递归	70
5.3.7	方法的重载	72
5.4	C#命名空间	73
5.4.1	命名空间的使用	74
5.4.2	命名空间的别名	76
5.5	类方法的使用	76
5.5.1	Console类方法	77
5.5.2	Math类方法	79
5.6	小结	80
5.7	习题	81
<b>第6章</b>	<b>C#面向对象编程基础</b>	<b>83</b>
6.1	面向对象基础知识	83
6.1.1	对象(object)	83
6.1.2	消息(message)	83
6.1.3	类(class)	84
6.2	声明类	84
6.2.1	添加/声明类	84
6.2.2	修饰符	86
6.2.3	添加/声明类的成员	86
6.3	构造函数和析构函数	88
6.3.1	构造函数	88
6.3.2	构造函数的重载	89
6.3.3	析构函数	90
6.4	面向对象的基本特征	90
6.4.1	封装性(encapsulation)	90
6.4.2	继承性(inheritance)	91
6.4.3	多态性(polymorphic)	92
6.5	访问修饰符	95
6.6	属性和字段	96
6.6.1	字段	96
6.6.2	字段的初始化	97
6.6.3	属性	97
6.7	方法的修饰符	97
6.8	小结	98
6.9	习题	98

<b>第 7 章 数组</b> .....	100
7.1 概述 .....	100
7.2 数组的声明、分配和使用 .....	100
7.2.1 声明和分配数组 .....	100
7.2.2 数组初始化函数 .....	103
7.2.3 数组应用举例 .....	104
7.3 System.Array 类型 .....	105
7.4 数组的参数传递 .....	105
7.4.1 值传递 .....	106
7.4.2 引用传递 .....	107
7.5 数组的查找和排序 .....	109
7.5.1 查找数组元素 .....	109
7.5.2 数组排序 .....	111
7.6 多维数组 .....	112
7.7 小结 .....	114
7.8 习题 .....	114
<b>第 8 章 字符串</b> .....	115
8.1 概述 .....	115
8.2 字符和字符串基础 .....	115
8.3 String 类构造函数 .....	116
8.4 String 类的索引器、属性和方法 .....	117
8.5 字符串比较 .....	118
8.6 字符串定位、查找 .....	120
8.7 String 类的子串提取 .....	122
8.8 String 类中的其他方法 .....	123
8.9 小结 .....	124
8.10 习题 .....	124
<b>第 9 章 简单 Windows 编程</b> .....	125
9.1 概述 .....	125
9.2 标签、按钮和文本框 .....	125
9.2.1 建立项目 .....	125
9.2.2 标签的创建和属性的设置 .....	126
9.2.3 按钮的创建和属性的设置 .....	126
9.2.4 文本框控件 .....	127
9.3 复选框和单选按钮 .....	129
9.3.1 复选框 .....	129
9.3.2 单选按钮 .....	129
9.4 下拉列表框和列表框 .....	134
9.4.1 下拉列表框 .....	134

9.4.2 列表框 .....	134
9.5 图片框 .....	136
9.6 TreeView 控件 .....	137
9.7 ListView 控件 .....	139
9.8 鼠标事件处理 .....	140
9.9 键盘事件处理 .....	140
9.10 小结 .....	141
9.11 习题 .....	142
<b>第 10 章 对话框 .....</b>	<b>143</b>
10.1 模态对话框 .....	143
10.1.1 添加对话框 .....	143
10.1.2 编辑对话框属性 .....	144
10.1.3 添加控件 .....	145
10.1.4 实现对话框自身功能 .....	145
10.1.5 定义对话框的数据访问 .....	147
10.1.6 显示对话框 .....	148
10.2 非模态对话框 .....	150
10.2.1 添加、设置对话框 .....	150
10.2.2 添加对话框属性 .....	150
10.2.3 实现控制功能 .....	151
10.2.4 显示、隐藏非模态对话框 .....	153
10.3 通用对话框 .....	154
10.3.1 打开文件对话框 .....	154
10.3.2 调色板对话框 .....	157
10.4 小结 .....	158
10.5 习题 .....	158
<b>第 11 章 图形与多媒体 .....</b>	<b>159</b>
11.1 Graphics 图形对象 .....	159
11.2 绘制线段 .....	159
11.3 绘制多边形 .....	161
11.4 绘制矩形 .....	163
11.5 绘制弧、椭圆、圆 .....	164
11.6 设置绘图属性 .....	168
11.6.1 设置画线的宽度和颜色 .....	168
11.6.2 绘制文本字符串 .....	169
11.7 多媒体编程 .....	170
11.7.1 处理图像 .....	170
11.7.2 Windows 媒体播放器 .....	172
11.8 小结 .....	173

11.9 习题 .....	174
<b>第 12 章 文件和流</b> .....	<b>175</b>
12.1 概述 .....	175
12.2 文件和流的概念 .....	175
12.3 文件存储管理 .....	176
12.3.1 目录管理 .....	176
12.3.2 文件操作 .....	178
12.3.3 目录下的文件操作 .....	180
12.4 读写文件 .....	181
12.5 小结 .....	183
12.6 习题 .....	183
<b>第 13 章 Visual C# 数据库访问技术</b> .....	<b>185</b>
13.1 数据库访问的基础知识 .....	185
13.1.1 数据库的基本概念 .....	185
13.1.2 关系数据库 .....	186
13.1.3 创建数据库 .....	186
13.1.4 结构化查询语言(SQL) .....	187
13.2 ADO.NET .....	191
13.2.1 ADO.NET 概述 .....	191
13.2.2 ADO.NET 的结构 .....	192
13.2.3 ADO.NET 的开发环境 .....	193
13.2.4 利用 ADO.NET 对象模型获取数据的流程 .....	193
13.3 应用 ADO.NET 访问数据库 .....	193
13.3.1 设计时连接 .....	193
13.3.2 以编程方式创建连接 .....	194
13.3.3 创建和使用 OleDbDataAdapter 对象 .....	195
13.3.4 创建和使用 DataSet 对象 .....	197
13.4 数据绑定 .....	199
13.4.1 单一的数据绑定 .....	199
13.4.2 复合的数据绑定 .....	200
13.4.3 实现对数据记录的浏览 .....	200
13.5 小结 .....	201
13.6 习题 .....	201
<b>第 14 章 Visual C# 调试技术</b> .....	<b>202</b>
14.1 Visual Studio.NET 调试功能简介 .....	202
14.2 设置断点 .....	202
14.3 程序执行状态的观察 .....	204
14.3.1 监视窗口 .....	204
14.3.2 自动窗口、局部变量和 This .....	205

14.3.3 查看寄存器 .....	206
14.3.4 查看函数堆栈 .....	206
14.3.5 查看内存区 .....	207
14.3.6 输出窗口 .....	208
14.4 小结 .....	208
14.5 习题 .....	208
<b>参考文献</b> .....	<b>209</b>

# 第 1 章 Microsoft.NET 和 C# 语言简介

本章简单介绍了 Microsoft.NET 和 C# 语言,包括 C# 语言的基础,Microsoft.NET 平台的设计理念、主要构成和设计目标,以及 C# 语言的特点和运行模型。

## 1.1 Microsoft.NET 概述

Microsoft.NET 是微软公司推出的面向下一代互联网软件和服务战略,它的出现标志着新的软件设计理念和服务理念的产生。Microsoft.NET 使得用户、企业和服务商三者联系更加紧密,使得计算和通信从目前广泛应用的单个网站环境变成丰富、合作和交互的网站环境,从而大大方便了用户,提高了企业的效率,丰富了服务提供商的服务内容。

### 1.1.1 Microsoft.NET 的设计理念

Microsoft.NET 的设计理念主要是把 Internet 作为构建新一代操作系统的基础,将 Internet 和操作系统的设计思想合理延伸,使得开发人员创建出摆脱硬件束缚的应用程序,从而方便地使用 Internet。Microsoft.NET 使得 Web 技术不仅适用于传统的 PC,而且能够满足日益增长的新设备的需要,例如个人数字助理(PDA)、移动电话和智能家电等。同时,开发商可以利用 Microsoft.NET 开发环境开发出新的 Internet 服务,从而使 Microsoft.NET 得到更多的认同。

Microsoft.NET 面向用户、企业和服务商,为三者提供了使用和创建服务的高效环境。基于 Microsoft.NET 的服务将系列的应用程序、服务程序和相关设备集成起来,创建成一种个性化的数字体验。Microsoft.NET 努力改变人类适应机器、适应技术的现状,为各类用户提供一种简单的操作信息的能力,使得商业用户和个人用户都能方便地查找和使用其中的信息,任何规模的公司都可以使用相同的工具与他们的供应商、商业伙伴和客户高效地沟通和分享信息,从而创建一种全新的协同工作模式。

- 对于用户来说,Microsoft.NET 意味着更简单化的整体服务:统一的信息浏览、编辑和授权,查看资料,工作,联机/脱机媒体。通过这个整体服务,用户可以通过任何一种联网设备(例如个人电脑、PDA、手机,甚至是智能家电)在任何一个地点及时和自动地实现任何个人信息的获取和修改。
- 对于企业来说,Microsoft.NET 意味着统一的信息浏览、编辑和授权;丰富的同步传播;密切的移动通信联系;得力的信息管理和电子商务工具,它可以在基于 Internet 和 Intranet 的服务程序之间灵活地切换,为动态商务伙伴关系的建立提供支持。
- 对独立软件开发商来说,Microsoft.NET 意味着创造基于 Internet 的更多新型高级服务的机会,这些服务可以借助 Internet 自动地从本地或远程获取和利用所需的信息,而无需为不同的工作环境重新编写程序。

### 1.1.2 Microsoft.NET 的组成

Microsoft.NET 主要包括以下 4 方面的内容:

- 1) 构造和管理新一代服务的基本结构和工具,包括:.NET 框架(Framework)、Visual Studio.NET、.NET 企业服务器和 Windows.NET。.NET Framework 是.NET 平台的基础,该框架最大的优点是消除了各类编程框架之间的差异,只用统一的命令集支持任何的编程语言,从而提供了强大的跨语言能力。用户可以自由地选择喜欢的编程语言,不同的编程语言可以互操作,从而共同完成项目开发。Visual Studio.NET 是微软公司推出的用于开发.NET 应用程序的强大的集成开发环境,该开发环境支持 Visual C#.NET、Visual Basic.NET、Visual C++.NET 等多种开发语言,以方便用户选择。
- 2) 创建互联网操作系统的构建块,包括用于用户认证的 PassPort.NET、用于文件存储服务、用户首选项服务、日历管理以及众多的其他服务。
- 3) 能够启用新型智能互联网设备的.NET 设备的软件。
- 4) .NET 用户体验。

### 1.1.3 .NET 框架和 C#

.NET 框架是.NET 平台基础架构,它消除了各类编程语言之间的差别,从而实现了跨语言平台编程的能力。.NET 的这种能力主要来源于公共语言运行时(Common Language Runtime,CLR)和类库。公共语言运行时和类库(Windows Forms,ADO.NET 和 ASP.NET)统一了各类语言类型,从而使得各种编程语言间无缝集成成为可能。

.NET 框架支持各种编程语言,因而开发人员可选择一种熟悉的语言进行.NET 开发。目前人们改进、开发的各种编程语言都或多或少以牺牲编程人员的灵活性作为代价,这些解决方案限制了开发人员能力的发挥,而且这些解决方案并不能很好地与当前流行的 Web 应用结合。

众所周知,近 20 年来,C 和 C++ 一直是最有生命力的编程语言,为开发人员提供了丰富的功能、高度的灵活性和强大的底层控制能力。但是这一切都不得不在效率上作出不同程度的牺牲,一方面,开发人员必须忍受较长时间的开发周期,另一方面,开发人员在享用 C 和 C++ 高度灵活性的同时,也必须付出长时间的学习过程。

C# 语言就是微软公司为解决上述问题而设计的,它是微软公司为.NET 计划开发推出的核心编程语言,该语言几乎综合了目前所有编程语言的优点。C# 是一种现代的面向对象的程序开发语言,开发人员能够利用它在.NET 平台上快速开发种类丰富的应用程序,并可以转换为 Web 服务。作为一流的面向对象设计语言,C# 从构建组件形式的高层商业对象到构造系统级应用程序都是最合用的,不但如此,C# 还为开发人员提供了快捷的开发方式和强大的控制能力,由于 C# 语法上与 C 和 C++ 有很大程度的相似,熟悉 C 和 C++ 的开发人员能够很快精通 C#。

## 1.2 C# 语言介绍

正如上节所述,C# 语言是专门为.NET 应用而开发的语言,并且在设计过程中综合了其

他编程语言的优点,既满足了应用程序的快速开发能力,又没有牺牲各种灵活性,同时它忠实地继承了 C 和 C++ 的优点。熟悉 C 和 C++ 的开发人员能够很快地掌握该语言,即使刚刚接触 C# 编程,C# 语言快速应用程序开发(Rapid Application Development, RAD)和简洁语法的特点也能使之很快成为一名熟练的开发人员。

C# 语言和 .NET 框架的完美结合,保证了使用 C# 语言开发程序能够享受到 .NET 框架提供的各种优点。

### 1.2.1 C# 的语言特点

C# 语言作为一种新生语言与其他编程语言相比有以下几个突出的特点:

- 简洁的语法。
- 面向对象的特点。
- 与 Web 的紧密结合。
- 安全性和错误处理。
- 版本处理。
- 灵活性与兼容性。

#### 1. 简洁的语法

虽然 C# 语言采用了 C 和 C++ 的语法结构,但是在设计过程中对 C 和 C++ 的语法进行了裁减,保留了其良好语法结构,去掉不良的语法结构。C# 最大的特点是没有指针,不能直接操作内存,当然与此相关的一些操作符也统统去掉;C# 用真正的关键字取代了较为混乱的伪关键字;C# 对 C++ 中的语法冗余问题也进行了简化,只保留了常见的形式。

#### 2. 面向对象的特点

C# 语言是一种纯面向对象语言,所谓纯面向对象语言是指除了面向对象所具有的封装、继承与多态外,其所有的类型也都采用了对象形式实现。以 C++ 语言作为对比,在 C++ 中整形变量类型是一个简单数据类型,如果要对其进行操作必须依赖于其他的处理函数,而在 C# 语言中整形类型是一个类,对它的操作仅仅通过类的方法调用就可以了,从而大大简化了编程难度。

#### 3. 与 Web 的紧密结合

目前随着网络的发展,许多应用程序开发模型都需要与 Web 标准相统一,例如超文本标记语言(Hypertext Markup Language, HTML)和扩展标记语言(eXtended Markup Language, XML),但是由于历史原因,很多开发工具不能很好地与 Web 紧密结合。C# 语言采用了简单对象访问协议(Simple Object Access Protocol, SOAP),使得采用 C# 开发大规模的分布式项目成为可能,有了 Web 服务的帮助,网络服务的开发就像开发本地应用程序一样,从而极大地方便了开发人员。

#### 4. 安全性和错误处理

C# 采用了 .NET 运行时提供的代码安全特性,从而提高了程序的安全性。在代码运行时,.NET 运行时将该代码的允许权和用户规定权限进行比对,如果允许权在用户的规定权限范围内,那该程序可以运行,否则不能运行。例如,在默认情况下,从网络上下载的代码都不能访问本地的文件和资源。

由于开发人员的特点,难免会出现错误,因而一个开发语言具有错误处理能力是非常重要的



的。C#的先进设计思想可以消除软件开发中的许多错误,减轻开发人员的工作量。在软件开发过程中,最让开发人员头痛的就是内存的管理,有时处理不好就会造成内存泄漏,甚至系统崩溃,C#借助.NET框架提供的垃圾收集器(Garbage Collector,GC),从而很好地解决了内存管理问题。

### 5. 版本处理

在软件升级中,系统组件中的代码修改很可能会对现存的软件产生影响,甚至可能会导致程序的崩溃。例如,Windows系统中dll版本控制问题就让微软伤透了脑筋。C#提供内置的版本支持来减少软件升级过程中可能出现的错误,使得软件的维护更加容易。

### 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。虽然它不是一种无限制的语言,比如不能用它来开发硬件驱动程序,默认情况下没有指针,但是在使用过程中,它仍然很灵活,你可以在需要的时候通过 delegates 模拟指针,同样也可以通过接口来实现类的多继承。

正是由于其灵活性,C#才允许与具有C风格的API函数进行互操作,C#遵守.NET公共语言规范(Common Language Specification,CLS),从而保证了C#组件与其他组件间的互操作性。元数据(Metadata)概念的引入在保证兼容性的同时,也实现了类型安全。

## 1.2.2 C#的运行模型

目前用高级语言编写的程序有两种形式,一种是被编译成机器语言在CPU上执行,如Visual C++,Pascal等。另外一种是在边解释边执行的,称为解释性语言,如Visual FoxPro,Visual Basic 3.0等。

随着软件规模的不断扩大,软件运行需要更多的系统资源,因而原有的编译型方式已经不能满足软件规模的发展。为了更好地管理、调用系统资源,程序就必须运行在一个平台或环境中,例如ASP程序需要在IIS环境中运行,Java程序运行在Java虚拟机中(Java Virtual Machine,JVM)。因此,理解一种语言的运行原理和它的运行平台就变得很重要。下面我们主要介绍C#程序的运行模型。

编写完一个C#程序后,首先需要通过C#编译器——CSC.exe,将C#程序编译成EXE或者DLL(Dynamic Link Library)等可执行文件,然而这些可执行文件并不能直接送入CPU运行,它们只是一种中间语言(Intermediate Language,IL),中间语言是一种与硬件平台无关的指令集(Instruction Set),这就使得C#编写的程序可以在任何支持.NET的系统上运行,实现了微软“一次编写,到处运行”的目标。

除了产生IL之外,CSC.exe还会将元数据(Metadata)添加到编译过的程序文件中。所谓元数据是指描述数据的数据,元数据主要用来描述程序的类型函数库(Type Library),比如类(Class)、属性(Property)、方法(Method)等信息。

编译后的可执行文件通过操作系统载入器(OS Loader)装入内存中初始化.NET环境后通过类加载器载入IL代码,通过及时编译器(Just-In-Time Compiler)转换为CPU可以执行的程序代码,当然根据C#安全性的特点,在执行前要进行安全性检查,C#程序的运行过程如图1-1所示。

图中阴影部分是.NET的运行环境,应用程序通过类加载器载入内存后,在执行过程中是通过JIT编译器边编译边运行的,与Visual C++、Pascal等一次编译全部程序的运行方式相比,无疑节约了很多资源。