

- 详细讲述C#语言的Windows应用程序设计方法
- 内容安排和叙述风格采用“教与学面对面”方式
- 理论结合实际，通过具体示例巩固所学知识
- 免费提供PPT电子课件以及书中示例的源程序

C#语言

Windows程序设计

于国防 李 剑◎编著

Microsoft

Visual Studio.NET



清华大学出版社

- 详细讲述C#语言的Windows应用程序设计方法
- 内容安排和叙述风格采用“教与学面对面”方式
- 理论结合实际，通过具体示例巩固所学知识
- 免费提供PPT电子课件以及书中示例的源程序

C#语言

Windows程序设计

于国防 李 剑◎编著

Microsoft

清华大学出版社
北 京

内 容 简 介

本书以 Visual Studio.NET 为开发平台,针对新一代面向对象、使用简单、表达力丰富的 C#编程语言,重点讲述了其功能最强、应用最广的 Windows 应用程序设计。书中内容的编排前后贯通、由浅入深;语言叙述力求通俗易懂,避免使用晦涩的专业术语,从而营造轻松、自然的学习环境;特别注重理论与实际的结合,随时通过适量的具体应用示例,对所学的知识加以巩固提高。每章后面布置了相应的习题和实验,从而更好地融“教、学、练”于一体,使学习者快速入门并具备良好的 C#编程实战技能。

本书不仅可以作为高等学校计算机应用、通信工程以及电子与信息工程等相关专业的本、专科学生的 C#教学用书,也可以作为数字图像处理与分析专业本、专科学生学习的基础教程。

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#语言 Windows 程序设计/于国防,李剑编著. —北京:清华大学出版社,2010.9

ISBN 978-7-302-23375-6

I. ①C… II. ①于… ②李… III. ①C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 152947 号

责任编辑:朱英彪

封面设计:张 岩

版式设计:王世情

责任校对:王 云

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

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

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:21.25 字 数:491 千字

版 次:2010 年 9 月第 1 版 印 次:2010 年 9 月第 1 次印刷

印 数:1~4000

定 价:35.00 元

前 言

自 2000 年 6 月微软公司推出 Microsoft.NET 以来, .NET 技术及其相关产品在信息技术领域得到了广泛应用。而为 .NET Framework 量身定做的新一代面向对象的 Visual C# 语言, 也随之成为业界主流的程序设计语言之一。C# 不仅功能强大, 而且其程序设计简捷灵活、易学易用。

本书以优秀的 C# 开发工具 Visual Studio.NET (简称 VS.NET) 为平台, 在以控制台应用程序设计作为 C# 语言入门学习的基础上, 重点讲述功能最强、应用最广的 C# 语言 Windows 应用程序设计。

本书的内容安排和叙述风格采用“教与学面对面”的方式, 而非刻板的纯书面传授方式, 所以可使读者在一种轻松自然的学习环境中学习掌握新的知识和技术。

全书共分为 12 章, 每章章首的学习要点便于读者合理分配学习精力; 内容讲解过程中列举了适量的示例, 并给出了包含详细注释的源代码, 非常有利于读者理解和分析程序。书中“说明”部分帮助读者加深理解重点和难点问题; 而简明扼要的应用技巧或引申知识“提示”更有利于拓展读者的知识面; 章后的习题和实验, 可供读者进一步上机练习使用。各章基本内容概括如下。

第 1 章: 引领读者初步认识 C#, 知道它的起源, 并通过对 C# 特点的介绍以及与其他编程语言的比较, 坚定读者学习 C# 的决心和学好 C# 的信心。

第 2 章: 因为 VS.NET 是利用 C# 对程序进行设计、开发、调试和部署的开发工具, 所以本章简单介绍了 .NET 技术, 使读者对 .NET 及 VS.NET 有一个初步的认识, 并在此基础上学会如何安装 Visual Studio 2005。

第 3 章: 通过一个简单的 C# 程序设计, 使读者熟悉 C# 编程的入门步骤, 也使其初步体验到 C# 的简单易学; 并在考虑到读者通常已具备一定的其他编程语言的基础上, 引导读者以一种类似复习的形式, 利用较少的学时快速掌握 C# 的变量、数据类型、运算符以及流程控制等基础知识。

第 4 章: 通过类概念的引入, 开始了面向对象的学习, 其中命名空间的应用和几种常用类的介绍, 使读者进一步领略 C# 面向对象的易学易用, 而方法和事件的介绍则使学习渐入 C# 程序设计的进阶时段。

第 5 章: 以一个基于 Win Forms 的简单上机自测系统的开发, 作为 C# 语言程序设计学习的入门示例, 形象直观地结合 3 种上机操作基本题型, 在 Windows 窗体的基础上, 依次分析讲解几种常用 Win Forms 控件的功能和开发方法; 然后, 适时应用菜单控件, 将全部窗体管理分类归纳; 最后, 引入工具栏设计, 使所设计的系统更趋完善。

第 6 章: 进程和线程是开发实时或多任务应用程序的技术基础, 非常重要但具有一定的学习难度。为此, 本章结合形象、直观的示例程序, 采用类比对照式的学习方法, 深入

浅出地重点讲述了多线程开发技术，并为后续网络通信编程中的线程应用奠定了基础。

第7章：首先介绍文件和流的基本概念以及与文件和流操作相关的路径和目录等知识，然后结合一个“三合一”的示例分析文件流读/写的3种基本方法，为第8章的网络编程奠定相关的技术基础。

第8章：网络编程是重点，也是本书学习的难点所在。遵循由浅入深的原则和示例讲解的方式，首先介绍基本知识，并逐渐引入重点内容，然后，基于此前所学相关知识，结合两个不同版本的网络聊天和文件上传与下载的程序设计示例，分别讲解基于TCP协议和FTP协议的网络编程技术；最后结合一个基础示例，讲解基于UDP协议的网络编程技术。

第9章：考虑到有些读者可能缺乏数据库方面的基础知识，本章首先以最精简的篇幅对数据库加以介绍，并结合一个简单、易理解的成绩管理系统示例，分别介绍Access和SQL Server的应用和开发知识。后面继续结合该示例，逐步引导读者综合运用ADO.NET的相关技术以及SQL语句，开发C/S模式的C#应用程序。

第10章：通过一个图文并茂的GDI+绘图基本步骤的形象化比喻，使读者在运用GDI+之前就对其基本用法有一个清晰、直观的整体认识；然后遵循该开发步骤，结合不同的示例，分门别类地讲解GDI+的图形绘制、文本呈现以及简单的图像处理技术。最后，结合一个基于鼠标轨迹画线的示例和一个转轮动画设计的示例，将GDI+的学习引向深入。

第11章：为实用扩展篇。目前基于C#语言的图像处理的参考书远不如C++丰富，众多想用C#进行图像处理的初学者，非常渴望能获得更多的相关参考资料。为此，本章选择了最基础的10个图像处理的典型示例，结合其程序设计流程图进行分析和编程，使读者快速入门。所以，本章既可作为C#学习的扩展练习，也可与此前相关章节统一学时安排，作为C#图像处理课程的基础教材。

第12章：本章利用有限的篇幅介绍如何将所开发的应用程序进行部署和发布。

本书由于国防和李剑编著。李剑编写了第1~4章和第12章，于国防编写了第5~11章，全书由于国防统稿。另外，在本书的编写过程中还得到了杜文龙、张玉杰、庄立运、贾栋清几位老师以及编者的研究生谭相录、寇立强、付丽娜的支持和帮助，在此表示衷心的感谢！

作为本书的编者，我们虽有多年的计算机语言编程的教学与工程应用经验，但也深知，在这一领域我们仍有许多技术或知识尚未融会贯通并正确运用。所以，尽管在编写此书的过程中，我们虚心请教了多位同行教师或专业人士，广泛征求了学生的建议，并参考了多部相关教材和参考书，但由于水平所限，书中不妥、错误之处仍在所难免，殷切希望广大读者批评指正。

为了便于读者测试和分析书中示例，随书提供了书中所有示例的源程序；同时，为了便于教师进行多媒体课堂教学，随书还提供了相应的PPT电子教案。

编者

2010年5月

教（学）计划

章 名	C#语言 Windows 程序设计		C#数字图像处理基础	
	教 学 学 时	实 验 学 时	教 学 学 时	实 验 学 时
第 1 章	2		2	
第 2 章	2		2	
第 3 章	4		4	
第 4 章	6	2	6	2
第 5 章	6	2	6	2
第 6 章	4	2	4	2
第 7 章	2	2		
第 8 章	6	2		
第 9 章	4	2	2（选讲）	2（选做）
第 10 章	6	2		
第 11 章	4（选讲）	2（选做）	6	4
第 12 章	自学			
小计	42（46）	14（16）	30（32）	10（12）
总计	56（62）		40（44）	

注：

- ① 实际教学学时和上机实验学时可根据教学或自学计划适当调整。
- ② 数字图像处理的基本原理可根据大纲要求另外增加学时（16~20 学时），并在此基础上学习 C#数字图像处理的程序设计。

目 录

第 1 章 C#概述	1
1.1 C#语言的由来	1
1.2 C#的特点	2
1.3 C#与其他编程语言的比较	3
1.3.1 C#语言与 C++语言的比较	3
1.3.2 C#语言与 Java 语言的比较	4
1.3.3 C#语言与 VB.NET 语言的比较	5
习题	5
第 2 章 .NET 简介	6
2.1 .NET 平台	6
2.2 .NET Framework	7
2.2.1 公共语言运行库 CLR	8
2.2.2 .NET Framework 类库	10
2.3 Visual Studio.NET 简介	10
2.3.1 Visual Studio.NET 的发展历史和优点	10
2.3.2 Visual Studio 2005 的安装	11
2.4 Visual Studio 2005 开发环境	14
2.4.1 进入 Visual Studio 2005 开发环境	14
2.4.2 创建与打开应用程序项目	15
2.4.3 设计与编辑应用程序项目	16
2.4.4 调试应用程序项目	18
习题	18
第 3 章 C#编程基础	19
3.1 一个简单的 C#程序设计	19
3.1.1 命名空间	20
3.1.2 Main()方法	22
3.1.3 程序代码注释	22
3.1.4 控制台的输入与输出	22
3.2 标识符和关键字	24
3.2.1 标识符	24
3.2.2 关键字	25
3.3 值类型	25

3.3.1 简单值类型	26
3.3.2 结构类型	28
3.3.3 枚举类型	30
3.4 引用类型	31
3.4.1 object 类型	31
3.4.2 类类型	32
3.4.3 字符串类型	33
3.4.4 数组类型	35
3.5 类型转换	39
3.6 常量和变量	40
3.6.1 常量	40
3.6.2 变量	41
3.7 表达式和运算符	42
3.7.1 表达式	42
3.7.2 运算符	43
3.8 流程控制语句	45
3.8.1 条件语句	46
3.8.2 循环语句	48
3.8.3 跳转语句	51
3.9 异常处理语句	52
习题	54
第4章 面向对象程序设计	56
4.1 面向对象编程的概述	56
4.1.1 类和对象	56
4.1.2 面向对象程序设计的相关概念	56
4.2 类	57
4.2.1 类的声明与实例	58
4.2.2 类成员	59
4.3 方法	62
4.3.1 方法的声明与调用	62
4.3.2 方法的参数	63
4.3.3 方法重载	63
4.4 事件	65
习题	66
第5章 Windows 窗体与控件程序设计	67
5.1 窗体与控件概述	67
5.1.1 窗体	67

5.1.2 控件	68
5.2 项目与解决方案概述	68
5.3 一个基于 Win Forms 的上机自测系统设计	69
5.3.1 Windows 窗体分析与应用	71
5.3.2 PictureBox 控件分析与应用	73
5.3.3 Label 控件分析与应用	75
5.3.4 添加新窗体	75
5.3.5 TextBox 控件分析与应用	77
5.3.6 RadioButton 与 Panel 控件分析与应用	77
5.3.7 ListBox 与 ComboBox 控件分析与应用	79
5.3.8 CheckBox 控件分析与应用	81
5.3.9 Timer 控件分析与应用	82
5.3.10 Button 控件分析与应用	84
5.3.11 启动窗体设置	86
5.3.12 MenuStrip 控件分析与应用	87
5.3.13 ToolBar 控件分析与应用	89
5.4 Windows 窗体的文档界面设置及其管理	90
5.4.1 基于单文档 (SDI) 的应用程序	90
5.4.2 基于多文档 (MDI) 的应用程序	90
5.4.3 自测系统多文档管理	91
5.5 Windows 窗体的显示模式分析与应用	93
5.5.1 模态窗体	93
5.5.2 非模态窗体	94
5.6 Windows 通用对话框概述	95
5.6.1 MessageBox 消息对话框	95
5.6.2 OpenFileDialog 对话框控件	96
5.6.3 SaveFileDialog 对话框控件	98
习题	98
第 6 章 进程与线程开发程序设计	100
6.1 进程与线程概述	100
6.2 进程开发技术	101
6.3 线程开发基础知识	104
6.4 多线程开发技术	109
6.4.1 多线程概述	109
6.4.2 多线程互斥与同步概述	109
6.4.3 多线程互斥程序开发	110

6.4.4 多线程同步程序开发	121
6.5 Windows 窗体控件的跨线程访问	125
6.5.1 非线程安全的窗体控件访问概述	125
6.5.2 线程安全的窗体控件访问概述	125
6.5.3 线程安全的窗体控件访问程序开发	126
习题	128
第 7 章 文件存取程序设计	129
7.1 文件与流概述	129
7.2 System.IO 命名空间与 System.Net.Sockets 命名空间	129
7.3 路径类与目录类	130
7.4 文件对话框类	133
7.4.1 OpenFileDialog 类	133
7.4.2 SaveFileDialog 类	133
7.5 文件读写与流操作	133
7.5.1 文件流操作	134
7.5.2 网络流操作	139
习题	140
第 8 章 网络通信程序设计	141
8.1 网络应用系统的体系结构	141
8.2 Socket 与网络通信	142
8.2.1 定义和获取主机消息	142
8.2.2 网络协议概述	143
8.2.3 Socket 概述	144
8.3 TCP/IP 与网络通信程序设计	145
8.3.1 TCP/IP 概述	145
8.3.2 基于 TCP/IP 的网络通信原理	145
8.3.3 基于 TCP 的网络通信程序设计	146
8.4 UDP 与网络通信程序设计	162
8.4.1 UDP 概述	162
8.4.2 基于 UDP 的网络通信原理	163
8.4.3 利用 UdpClient 类实现 UDP 网络通信	163
8.5 FTP 与网络通信程序设计	168
8.5.1 FTP 概述	168
8.5.2 FTP 服务的工作原理	168
8.5.3 网络文件传输的 FTP 程序设计	169
习题	186

第 9 章 数据库访问程序设计	187
9.1 数据库基础知识	187
9.1.1 数据库概述	187
9.1.2 Access 数据库概述及其应用	188
9.1.3 SQL Server 数据库概述及其应用	195
9.1.4 SQL 语言概述	201
9.2 ADO.NET 概述	204
9.2.1 ADO.NET 体系结构	204
9.2.2 ADO.NET 数据提供程序	205
9.2.3 ADO.NET 命名空间及其常用类	205
9.3 ADO.NET 数据库访问	206
9.3.1 ADO.NET 数据库访问步骤	207
9.3.2 SqlConnection 类与数据库连接	208
9.3.3 DataSet 数据集与 DataAdapter 数据适配器	211
9.3.4 SqlCommand 类与 SQL Server 数据库操作	214
9.3.5 利用 DataReader 类获取只读数据	220
9.3.6 利用 DataTable 类和 DataView 类更新数据及控制视图	221
习题	226
第 10 章 GDI+图形图像程序设计	228
10.1 GDI+概述	228
10.1.1 .NET 图形图像处理命名空间	229
10.1.2 GDI+绘图的基本步骤	229
10.1.3 Graphics 类	230
10.1.4 颜色	230
10.1.5 坐标系统	231
10.1.6 画笔与画刷	232
10.2 利用画笔或画刷绘制图形	233
10.2.1 绘制直线	233
10.2.2 绘制多边形	234
10.2.3 绘制曲线	235
10.2.4 绘制矩形	236
10.2.5 绘制椭圆(或圆)	237
10.3 利用画刷填充图形	238
10.3.1 HatchBrush 画刷	238
10.3.2 TextureBrush 画刷	240
10.3.3 LinearGradientBrush 画刷和 PathGradientBrush 画刷	242
10.4 字体与文本绘制	244

10.4.1	Font 类与字体	244
10.4.2	绘制文本	245
10.5	基于鼠标轨迹画线的程序设计示例	246
10.6	GDI+图像处理基础	249
10.6.1	GDI+图像处理概述	249
10.6.2	图像的显示与保存	251
10.6.3	动画程序设计示例	254
	习题	256
第 11 章	C#数字图像处理基础程序设计	258
11.1	C#与数字图像处理概述	258
11.2	C#数字图像像素操作的 3 种方法	258
11.3	C#数字图像处理程序设计	263
11.3.1	图像灰度化	266
11.3.2	灰度图像直方图	268
11.3.3	图像二值化	273
11.3.4	图像滤波	276
11.3.5	图像锐化	279
11.3.6	图像边缘检测	283
11.3.7	图像轮廓链码提取	288
11.3.8	图像细化	294
11.3.9	图像腐蚀	302
11.3.10	图像膨胀	307
	习题	310
第 12 章	Windows 应用程序的部署	312
12.1	应用程序部署概述	312
12.2	Windows Installer 部署	313
12.2.1	使用安装向导制作安装程序	314
12.2.2	部署应用程序	318
12.2.3	卸载应用程序	318
12.3	ClickOnce 部署	318
12.3.1	发布到共享文件及其安装	318
12.3.2	发布到 Web	320
12.3.3	发布到 CD-ROM/DVD-ROM	320
	习题	321
附录	实验参考	322
实验 1	C#语言程序设计基础	322

实验 2 窗体及控件程序设计	322
实验 3 多线程开发程序设计	323
实验 4 文件读/写	323
实验 5 网络编程	324
实验 6 数据库访问	324
实验 7 GDI+与图形图像	325
实验 8 图像处理	325
参考文献	326

第 1 章 C#概述

学习要点

- ☐ 了解C#语言的由来
- ☐ 了解C#语言的特点
- ☐ 了解C#语言和其他高级编程语言的差异

C#（读作“C sharp”）是目前比较流行的一种编程语言。作为一种类型安全、面向对象的语言，C#简单却功能强大，可让程序设计人员设计出范围广泛的应用程序。本章主要介绍C#的由来、C#的特点及其与其他编程语言的比较。

1.1 C#语言的由来

对C#的由来有两种解释：从字面意义来解释，C#是C语言的开发利器；而微软公司给出的解释是，C#是C++的升级语言，具有比C++更优越的开发特性。

随着软件开发技术的不断发展，程序员们不但要解决各种技术平台上的组件兼容问题，还需要解决由不同编程语言开发的组件间的集成问题；同时Web应用已成为一种趋势，编程语言还应该能够快速地进行网络应用的开发。为了满足这些需求，2000年微软公司发布了C#编程语言。这种语言作为微软公司.NET战略的一部分，是.NET平台应用的首选语言。C#语言是由Anders Hejlsberg和Scott Wiltamuth领导的小组专门为.NET平台设计的，运行于.NET Framework之上的编程语言。

C#由C/C++演化而来，它在表达式、运算符和语句等方面沿用了C/C++的许多特性，而在类型安全、错误处理、版本转换、事件和垃圾回收等方面做了很大的改进和创新。

自2000年C#诞生以来，其版本不断更新，改进和新增了很多特性。

- ☒ 2000年，微软公司发布了C#的第一个预览版。
- ☒ 2003年，微软公司发布了C#规范1.2（简称C#1.2）。
- ☒ 2005年，微软公司发布了C#规范2.0（简称C#2.0）。
- ☒ 2007年，微软公司发布了C#规范3.0（简称C#3.0）。

C#从诞生起就很好地体现了面向对象的设计精神，在C#语言中不仅使用了类、继承、多态的概念，还使用了属性、方法、接口、事件等概念，很好地描述了对对象。同时，C#还通过值类型和引用类型的概念形成了统一的类型系统，所有的类型只有实例化为对象才能被使用。

C#2.0增加了泛型、匿名方法、迭代等新特性，增强了语言的表达能力。C#2.0已经被国际标准化组织定为高级语言开发标准。

C#3.0 又增加了查询关键字、Lambda 表达式、匿名类型、扩展方法、自动实现的属性、隐式类型的本地变量和数组等功能。这些新的语言构造可以分别用在各种上下文中，并且可以共同完成语言集成查询（LINQ）功能。

本书以 C#2.0、.NET Framework 2.0 和 Visual Studio.NET 2005 为范本和开发环境，所有案例均在此开发环境下调试运行。

1.2 C#的特点

相对于其他编程语言，C#的特点概括如下。

1. 简洁的语法

C#语言相对 C++语言而言，简单易学、容易入门。C#语言淘汰了 C++语言中繁乱的表示符号和伪关键字，使用了有限的、统一的操作符、修饰符和运算符。另外，C#语言很少使用 C++语言中功能强大却让人头疼的指针。

2. 面向对象的设计

作为一种完全按照面向对象的思想而设计的新语言，C#语言支持面向对象的所有关键特性，如封装、继承和多态等，是真正纯粹的面向对象的编程语言。而 C++、Java 等语言并不是纯粹的面向对象的编程语言。

C#语言以类和结构为基础构建所有的类型，每种类型都是一个对象。C#语言通过命名空间对代码进行层次化的管理，所有的常量、变量、属性、方法、事件等都被封装在类中，从而使程序具有更好的可读性，并且避免了发生命名冲突的可能。

3. 完备的安全性

默认情况下，代码工作在一种受托管的环境中（将在第 2 章介绍），在托管环境下不允许进行类似直接存取内存的不安全操作。C#语言可以消除许多软件开发中常见的编程错误，如忘记初始化变量和数组越界等，提供包括类型安全在内的严格的安全检查。此外，不允许网络上下载的代码直接访问本地文件和资源；提供了边界检查和溢出检查等功能；使用垃圾回收机制减轻内存管理的负担；通过使用 CLR（公共语言运行库，Common Language Runtime）提供的代码，在程序中配置安全等级和用户权限等。这些安全机制保证了 C#语言具有完备的安全性，使得程序员能够把注意力集中在其他方面。

4. 版本控制

版本问题一直就是比较棘手的问题。例如，由于多个应用程序都安装了名字相同的 DLL 的不同版本，有时应用程序能够正常运行，更多的时候会中断运行。C#语言内置了版本控制功能，可以很好地支持版本管理，从而使得 C#语言开发复杂的软件可以不断地进行更新和升级。同时，C#语言的标准开发环境 Visual Studio 集成的发布和其部署功能，能够确保安装和卸载应用程序时不会影响其他应用程序的使用。

5. 良好的兼容性

C#语言凭借.NET Framework 平台对 COM+组件、XML Web 服务和 MSMQ（微软消息队列）服务的支持，能够跨语言、跨平台交互操作，能够实现不同软件技术开发的组件之间的互相调用，还能实现组件之间跨互联网的调用。作为.NET Framework 的首推语言，C#在很大程度上保持了对外界技术的兼容。

6. 支持快速开发

C#语言增强了开发效率，借助于 Visual Studio 开发工具，可以通过拖放的形式自动添加组件并生成相应的代码。而自动生成的代码和手动添加的代码又相隔离，便于程序员检查自己的设计。

7. 面向组件的开发

面向组件的设计方法是继面向对象的设计方法之后又一流行的趋势。C#语言提供生成持久系统级组件的能力。在 C#语言中，组件可以在开发中直接使用，也可以通过调用对象所提供的方法来进行操作。数据访问组件是 C#语言（或者说是 Visual C#）中最具特色的组件。

1.3 C#与其他编程语言的比较

C#语言是专为.NET 平台设计的，在.NET Framework 中，既可以用 C#语言开发运行在 Windows 操作系统上的应用程序，也可以开发基于智能设备的应用程序；既可以开发 B/S 和 C/S 应用程序，也可以开发 Web 应用程序。而且在同一个项目的不同设计文件中可以分别使用 C#和 Visual Basic 等语言。从语法规则和功能上看，C#和 C++非常相似；对于 Web 应用开发而言，C#像 Java 语言，又具有 Delphi 语言的一些优点；同时，由于是微软公司的产品，借助 Visual Studio 开发工具，C#和 Visual Basic 一样简单易用。

本节将 C#语言与目前常用的高级编程语言 C++、Java 和 Visual Basic.NET 做了简单的比较，以便读者更好地了解 C#语言的特点，更快地掌握 C#语言。

1.3.1 C#语言与 C++语言的比较

C#语言虽然是从 C++语言发展而来的，但是它们之间却有很大的区别。

1. 编译

C++程序直接编译成标准的二进制可执行代码，而 C#的源代码由.NET 兼容编译器产生，称为程序集的输出文件，程序集的代码不是二进制的本机代码，而是一种称为 CIL（公共中间语言，Common Intermediate Language）的中间语言（IL），程序集还包括代码的类型信息和安全信息。执行时，程序集通过实时（Just-In-Time）编译器将需要的模块临时编译成本机代码。

2. 类的继承

C++语言允许类的多继承，即一个类可以有多个基类。而 C#语言只允许类的单继承，

只在需要时通过接口实现多继承。此外，C++语言没有接口的概念。

3. 指针

C++语言中大量使用指针，没有托管代码的概念。作为 C++ 的一个优点，使用指针允许开发人员针对底层机器编写优化的代码。但这也是 C++ 的缺点，因为使用指针可能会带来内存泄漏以及管理漏洞等不安全因素，这将使开发和维护的难度大大增加。而 C# 语言中操作的基本上是实例的对象，只有部分类类型支持指针，而且不建议使用指针。

4. 字符串处理

C++语言中对字符串的处理比较麻烦，需要调用函数处理。而 C#语言中，字符串是作为一种基本的数据类型来使用的。

5. 预处理器指令和操作符重载

C#语言中减少了 C++ 中大量预编译的伪指令（如 `#include` 伪指令和宏定义），赋值运算符都不能被重载。C#语言还支持其他的运算符，并引入了某些逻辑运算符的不同功能。C++ 中操作符重载的运用也降低了程序的可读性。

6. 内存管理

C++ 不提供垃圾回收功能，在进程终止前，未显式释放的内存仍保持已分配的状态。C#语言提供垃圾回收功能，能够自动地在合适的时间回收不再使用的内存。

7. 输入/输出

C#语言的输入/输出服务和格式设置依赖于 .NET Framework 的公共语言运行库 CLR。在 .NET Framework 类库中提供了大量针对控制台、Windows 窗体、Web 页面的输入/输出方法，开发人员可以方便地调用这些方法完成各种输入/输出功能。

C#语言与 C++语言在语法上的不同还有很多，这里不再一一说明，毕竟 C#是独立的、全新的、完全面向对象的高级编程语言。

1.3.2 C#语言与 Java 语言的比较

C#语言和 Java 语言有很多相似之处，也有很多不同之处。

1. 相似处

C#语言和 Java 语言都被编译成跨平台、跨语言的代码，并且代码只能在一个受控制的环境中运行；在网络应用方面，两种语言都支持 Web Server；两种语言在内存管理方面都是用垃圾回收机制管理内存；两种语言都没有头文件，都支持命名空间；两种语言都没有全局变量和全局常量，一切都属于类；两种语言中数组和字符串都能自动进行边界检查和长度计算。

2. 不同处

C#语言毕竟是从 C++继承而来的，所以在语法细节上和 Java 语言有很多不同之处，如 C#语言中有结构、枚举等类型，而 Java 语言没有。但 C#语言和 Java 语言的最主要差别是