

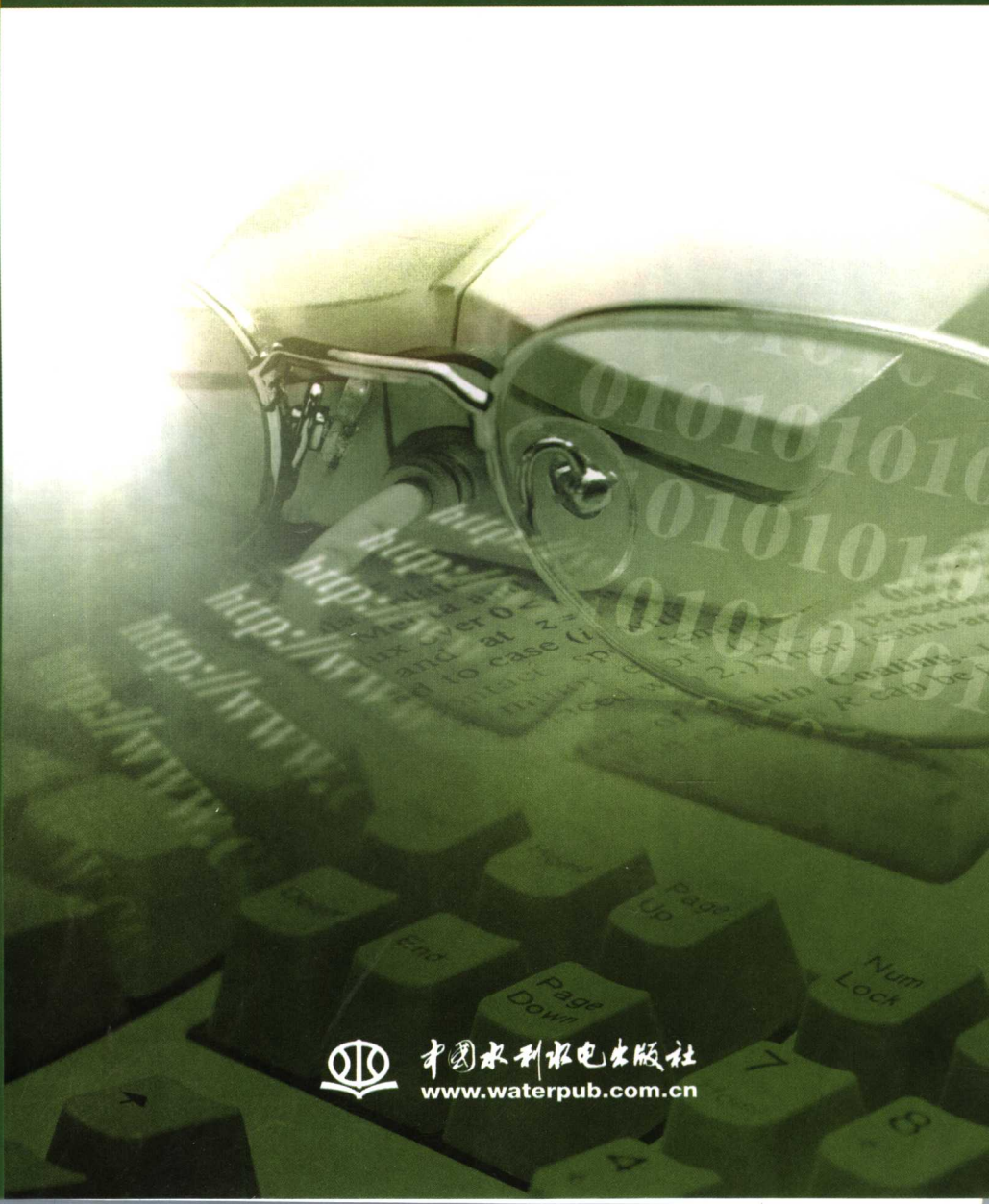
21

世纪 高职高专规划教材

C#语言程序设计

李继武 彭德林 主 编 张 珑 赵 松 周建辉 副主编

21SHIJIGAOZHIGAOZHIGUANGUIHUAJIAOCAI



中国水利水电出版社
www.waterpub.com.cn

21 世纪高职高专规划教材

C#语言程序设计

李继武 彭德林 主 编

张 琬 赵 松 周建辉 副主编

中国水利水电出版社

内 容 提 要

C#语言是近两年来十分流行的编程语言,它兼有 C++ 的强大功能和 VB 的易用性,是未来非常有前途的编程语言。

本书主要内容包括 .NET 开发平台简介、C#语言基础、C#面向对象初级编程、C#面向对象高级编程、C#进行文件 I/O 操作、C#开发 Windows 应用程序、C#开发 ASP.NET Web 应用程序等内容。

本书是计算机专业及其他相近专业的一门专业技能课的教材。本教材以“案例驱动式”教学模式为导向,在每节都先讲解 C#语言的基本概念和编程技巧,然后通过精心设计的案例来实践讲过的理论。在充分考虑高职高专人才教育特点的基础上,着重对学生进行基本 Windows 应用程序和 Web 应用程序的初步开发能力的培养。作为应用性程序开发类课程,本书在强调先进编程思想和先进编程技术的基础上,着重讲述面向对象编程的基本技术和 C#语言编程的基本技能。学完本书,要求学生能够利用 C#语言进行初步的 Windows 应用程序开发和 Web 应用程序开发。

本书不仅适合没有编程经验的读者学习,也适合有一定程序语言基础的读者自学,还可作为高职高专和各类计算机培训班的教材。

本书配有电子教案和源程序代码,读者可以从中国水利水电出版社网站上免费下载,网址为: www.waterpub.com.cn/softdown。

图书在版编目(CIP)数据

C#语言程序设计 / 李继武, 彭德林主编. —北京: 中国水利水电出版社, 2006

(21 世纪高职高专规划教材)

ISBN 7-5084-3947-3

I. C… II. ①李…②彭… 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 082978 号

书 名	C#语言程序设计
作 者	李继武 彭德林 主 编 张 珑 赵 松 周建辉 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 13.5 印张 299 千字
版 次	2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷
印 数	0001—4000 册
定 价	20.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

近年来,软件开发平台之争非常激烈。以 Java 为核心的 SUN 阵营和以 .NET 为核心的 Microsoft 阵营都竭尽全力完善自己的开发平台。作为 Windows 平台的开发者和桌面软件的垄断者,Microsoft 推出的 .NET 开发平台无疑会成为未来主流的开发平台之一。

高等职业教育作为我国高等教育的重要组成部分,它的主要培养目标就是为社会输送符合新时期社会需求的技能型、应用型人才。本书即着眼于当前最新的 .NET 软件开发平台和最新的 C# 语言编程技术,结合高职高专的培养目标,打造满足未来社会需求的、有竞争力的技能型软件开发人才。

本书第 1 章首先介绍了 .NET 平台的基础知识,然后重点讲解了 .NET 框架的主要内容和 Visual Studio.NET 开发环境,第 2 章讲解了 C# 语言编程的基础知识,这部分是学习编程的入门知识,主要为后续章节奠定基础。第 3 章和第 4 章主要讲解如何通过 C# 语言进行面向对象编程,这部分对建立学生面向对象的编程思维非常重要,面向对象编程的三大特征:封装、继承和多态都得到了很好的讲解。第 5 章对 .NET 框架中有关文件 I/O 操作的类作了详细讲解,这非常有助于帮助学生建立流的概念。第 6 章和第 7 章是本书的重头戏,正所谓“学以致用”,第 6 章通过大量的案例讲解如何通过 C# 语言在 VS.NET 集成开发环境下开发 Windows 应用程序,而第 7 章则讲解如何开发 ASP.NET Web 应用程序。

纵观全书可以知道,本书的特色在于以 C# 语言为核心,以 VS.NET 为开发环境,以基本编程理论和编程技巧为线索,以精心设计的案例为驱动,由浅入深、循序渐进地讲解了 C# 语言程序设计技术。

本书由李继武、彭德林任主编,张珑、赵松、周建辉任副主编。全书由李继武和彭德林统稿。

第 1 章由刘妍编写;第 2 章的 2.1 节由周建辉编写,2.2~2.5 节由张珑编写,2.6 节和 2.7 节由迟国栋编写;第 3 章由赵松编写;第 4 章的 4.1~4.4 节由赵松编写,4.5 节和 4.6 节由赵淑英编写,4.7 节由钱英军编写;第 5 章的 5.1 节和 5.2 节由张珑编写,5.3 节由李继连编写,5.4 节由刘妍编写;第 6 章的 6.1~6.3 节由李继武编写,6.4 节由赵淑英编写;第 7 章由彭德林编写;每章的小结和实训由刘妍编写。

由于编者水平有限,书中的错误和疏漏之处在所难免,敬请广大读者和同仁批评指正。

编 者

2006 年 5 月

目 录

前言

第 1 章 .NET 平台简介	1
1.1 .NET 平台	1
1.2 .NET 框架	2
1.2.1 公共语言运行库 (CLR)	2
1.2.2 .NET 框架类库	2
1.3 Visual Studio .NET	3
1.3.1 Visual Studio .NET 简介	3
1.3.2 用 Visual Studio .NET 开发 C# 应用程序	3
本章小结	5
本章实训	5
第 2 章 C# 语言基础	6
2.1 基本数据类型	6
2.1.1 值类型	6
2.1.2 引用类型	10
2.2 变量和常量	12
2.2.1 变量	12
2.2.2 常量	17
2.3 数组	18
2.3.1 一维数组	18
2.3.2 二维数组	21
2.3.3 数组的一些特性	23
2.4 表达式	24
2.5 语句	25
2.5.1 条件语句	25
2.5.2 循环语句	30
2.5.3 跳转语句	35
2.5.4 异常处理语句	38
2.6 运算符	42
2.6.1 算术运算符	42
2.6.2 赋值运算符	43
2.6.3 关系运算符	44

2.6.4	逻辑运算符	45
2.6.5	三元运算符	46
2.6.6	运算符优先级	46
2.7	名称空间	47
	本章小结	48
	本章实训	49
第 3 章	C#面向对象初级编程	50
3.1	面向对象的基本概念	50
3.1.1	对象和类	50
3.1.2	继承	51
3.1.3	封装	51
3.1.4	多态	51
3.2	C#中的类与对象	51
3.2.1	在 C#中定义类	51
3.2.2	访问修饰符	52
3.2.3	实例化对象与构造函数	55
3.2.4	方法重载	59
3.2.5	销毁对象与析构函数	64
3.2.6	在方法调用中传递参数	65
3.2.7	静态对象成员	70
3.2.8	用属性封装数据	71
	本章小结	74
	本章实训	74
第 4 章	C#面向对象高级编程	76
4.1	类的继承与多态	76
4.1.1	继承	76
4.1.2	多态	80
4.1.3	抽象和密封	83
4.2	操作符重载	87
4.2.1	问题的提出	87
4.2.2	使用成员方法重载操作符	88
4.3	类型转换	91
4.3.1	隐式类型转换	92
4.3.2	显式类型转换	92
4.3.3	类的引用转换	94
4.3.4	装箱与拆箱	94
4.4	结构和接口	97

4.4.1	结构	97
4.4.2	接口	101
4.5	集合与索引器	108
4.5.1	集合	108
4.5.2	索引器	111
4.6	异常处理	112
4.6.1	异常类	112
4.6.2	抛出和捕获异常	114
4.7	委托和事件	120
4.7.1	委托	120
4.7.2	事件	122
	本章小结	124
	本章实训	124
第 5 章	C#进行文件 I/O 操作	126
5.1	用于文件操作的类	126
5.2	目录和路径操作	127
5.2.1	Directory 类	127
5.2.2	DirectoryInfo 类	129
5.2.3	Path 类	132
5.3	创建文件	134
5.3.1	File 类	134
5.3.2	FileInfo 类	136
5.3.3	FileStream 类	138
5.3.4	创建文件的几种方法	140
5.4	读写文件	141
5.4.1	使用 FileStream 类读写文件	141
5.4.2	使用 StreamReader 和 StreamWriter 类读写文本文件	144
	本章小结	147
	本章实训	147
第 6 章	C#开发 Windows 应用程序	149
6.1	建立 Windows 应用程序	149
6.2	使用 Windows Forms 控件	153
6.2.1	Button 控件	153
6.2.2	Label 控件	154
6.2.3	LinkLabel 控件	156
6.2.4	TextBox 控件	157
6.2.5	CheckBox 控件	159

6.2.6	RadioButton 控件	161
6.2.7	GroupBox 控件	163
6.2.8	ComboBox 控件	164
6.2.9	ListBox 控件	166
6.2.10	StatusBar 控件	167
6.3	使用菜单	170
6.3.1	创建主菜单	170
6.3.2	创建上下文菜单	172
6.4	对话框	172
6.4.1	模态和非模态对话框	172
6.4.2	通用对话框	174
本章小结		179
本章实训		179
第 7 章	C#开发 ASP.NET Web 应用程序	181
7.1	Web 应用程序基础	181
7.1.1	HTML 页面	181
7.1.2	动态 Web 页面	183
7.2	ASP.NET 简介	184
7.2.1	支持编译语言	185
7.2.2	程序代码与页面内容的成功分离	185
7.2.3	引入服务器端控件	185
7.3	建立 ASP.NET 应用程序	186
7.3.1	启动 IIS 服务器	186
7.3.2	用 Visual Studio.NET 创建 ASP.NET 应用程序	187
7.4	Web 窗体和控件	190
7.4.1	Web 窗体	190
7.4.2	Web 控件	191
7.4.3	HTML 控件	202
本章小结		204
本章实训		204
参考文献		206

第 1 章 .NET 平台简介

.NET 平台是微软公司推出的一个新的开发平台,按照微软的说法,.NET 是一个“革命性的新平台,构建于开放的 Internet 协议和标准之上,并提供工具和服务,以新的方式融合计算和通信。”更通俗的说法应该是:.NET 平台既是一个软件开发环境,也是一个软件运行环境。

.NET 平台的主要内容是.NET 框架,本章重点介绍.NET 框架的主要内容。

1.1 .NET 平台

.NET 平台主要包含四部分内容:底层操作系统、.NET 企业服务器、.NET 框架和 Visual Studio .NET 集成开发环境。

1. 底层操作系统

这部分包括微软提供的几种操作系统: Windows 2000、Windows XP、Windows 2003 等。

2. .NET 企业服务器

Microsoft SQL Server 2000——微软提供的.NET 平台的数据存储解决方案,一个功能强大、使用方便的数据库管理系统。

Microsoft Exchange 2000——微软推出的通信和协作服务器,可作为 Internet 上的邮件服务器。对那些需要在应用程序中增加通信或 workflow 特性的开发者而言,它提供了大量方便易用的功能。

Microsoft BizTalk Server 2000——它是 Windows 企业服务器中最重要的产品之一,程序开发者使用它可以迅速创建高效率的、健壮的、高可互操作性的 B2B 电子商务应用程序。

Microsoft Application Center 2000——它用来简化 Web 服务器或应用程序服务器群集的创建和管理任务,通过管理控制台,使得创建服务器群集、向群集中添加新服务器、关闭或启动服务器以进行升级等操作更加简单。

Microsoft Mobile Server——它是微软公司 Pocket PC 和 Smartphone 的软件平台。Windows Mobile 把熟悉的桌面系统 Windows 扩展到了个人手持设备上。

3. .NET 框架

.NET 框架包含两个主要部分:公共语言运行库(Common Language Runtime, CLR)和.NET 框架类库(FCL)。

4. Visual Studio .NET 集成开发环境

它是微软耕耘多年的开发工具,在构建基于 Windows 平台的各种程序时,它为程序员提供了非常出色的开发环境。

1.2 .NET 框架

1.2.1 公共语言运行库 (CLR)

公共语言运行库 (CLR) 为执行用 .NET 语言编写的代码提供了一个运行环境。它管理 .NET 代码的执行, 包括对内存和对象生存期的管理。具体来说, CLR 主要负责以下几项任务:

(1) 创建和执行托管代码。

用 .NET 框架编写的代码就是托管代码, 它在 CLR 的控制下运行。为了利用 CLR 的功能, 我们首先选择一种面向 CLR 的编程语言, 如 C# 或 VB.NET, 然后利用该语言进行软件开发, 开发完的程序将被编译成微软中间语言 (Microsoft Intermediate Language, MSIL)。在执行 MSIL 之前, 由 .NET 框架的即时 (Just-In-Time) 编译器将 MSIL 转换成适合特定 CPU 结构的本地代码。

(2) 自动内存管理。

我们在用 C++ 进行软件开发时, 对内存的使用完全由程序员来决定, 内存的申请和释放也完全由程序员自己来做。实践表明, 程序员在使用完内存后常常忘记释放内存, 从而造成内存泄漏, 为程序的稳定性埋下了隐患。

CLR 提供了一种垃圾回收机制。也就是说, 程序员只管申请内存, 不管释放内存。CLR 后台有一个专门的垃圾回收线程, 它会按照自己的算法不定时地检查它托管的内存, 如果发现没有引用的对象在占据内存, 它就会把内存收回。

(3) 跨语言的互操作性。

所谓跨语言的互操作性就是指用一种语言编写的代码同用另一种语言编写的代码进行交互操作的能力, 这种能力有助于提高代码的重用性, 提高软件的开发效率。

所有面向 CLR 的编程语言都必须支持通用语言规范 (Common Language Specification, CLS), CLS 是支持 .NET 的编译器的最低标准集, 它可以确保多种语言的互操作性。

1.2.2 .NET 框架类库

.NET 框架类库 (FCL) 通过提供可以继承的基类为开发者提供支持。开发者在自己的应用程序中可以使用一套分层次的 .NET 类, 这些类由被称为命名空间的容器组织起来, 能够为开发者提供可重复使用的基本功能或者高级功能。

.NET 框架类库 (FCL) 主要包括如下内容:

(1) 系统框架服务。

例如, 集合、输入/输出、字符串操作、图像、网络等。

(2) ADO.NET 组件。

通过 ADO.NET 组件为程序提供到数据库的访问服务。

(3) XML 数据组件。

.NET 框架对 XML 数据提供操作支持,并且还支持 ADO.NET 数据同 XML 数据之间的相互转换。

(4) Windows Forms 组件。

Windows Forms 组件为程序员开发 Windows GUI 程序提供了强大的支持,程序员通过它能够开发出功能丰富、界面美观的 Windows 程序。

(5) Web Forms 组件。

.NET 框架中的 ASP.NET 为程序员开发 Web 程序提供了丰富的服务器端组件和客户端组件,使开发人员可以非常直观地进行 Web 开发。

1.3 Visual Studio .NET

1.3.1 Visual Studio .NET 简介

Visual Studio .NET 是微软开发的针对 .NET 平台的集成开发环境,它包含一套完整的开发工具,可以开发桌面应用程序、ASP.NET Web 应用程序、XML Web Services 和移动应用程序。该环境包含 Visual Basic .NET、Visual C++ .NET、Visual C# .NET 和 Visual J# .NET 四种编程语言,并允许它们共享工具,这有助于创建混合语言解决方案。另外,这些语言充分利用了 .NET 框架的功能,该框架提供了对简化 ASP.NET Web 应用程序和 XML Web Services 开发的关键技术的访问。作为快速创建和集成 XML Web 服务和应用程序的单一综合工具,Visual Studio .NET 在改善操作的同时极大地提高了开发人员的效率。Microsoft Visual Studio .NET 正在快速地发展成为 Windows 平台上的关键的集成软件开发环境。

1.3.2 用 Visual Studio .NET 开发 C# 应用程序

Visual Studio .NET (本书简称 VS.NET) 是微软针对 .NET 平台提供的集成开发环境,它为开发人员提供了功能强大的调试器和各种各样的设计器。下面通过一个案例来学习 VS.NET 开发程序的基本步骤。

案例: 学习 VS.NET 开发控制台程序的过程。

目标: 掌握 VS.NET 开发程序的基本步骤。

步骤:

1) 单击“开始”→“程序”→Microsoft Visual Studio .NET 2003→Microsoft Visual Studio .NET 2003 命令,启动 VS.NET 开发环境,界面如图 1-1 所示。

2) 单击 VS.NET 开始界面上的“新建项目”按钮,打开“新建项目”窗口,界面如图 1-2 所示。

3) 在“新建项目”窗口中,项目类型选择“Visual C# 项目”,模板选择“控制台应用程序”,名称默认为 ConsoleApplication1,位置改为“C:\CSharpSamples\chp1”,然后单击“确定”按钮,打开“代码编辑器”窗口,如图 1-3 所示。

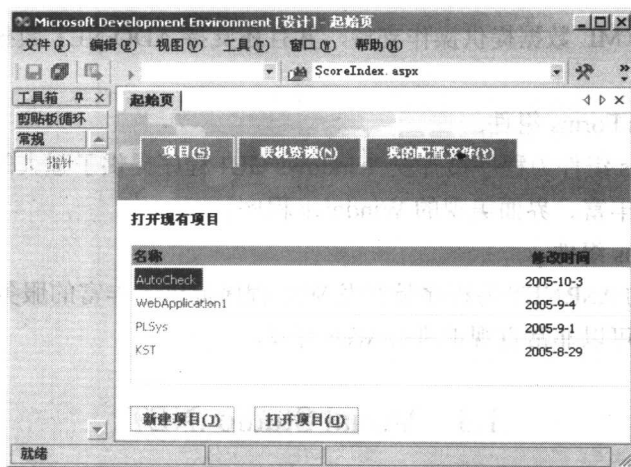


图 1-1 VS.NET 开始界面

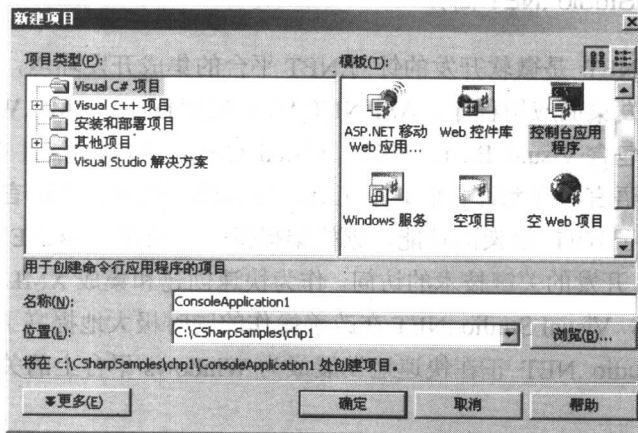


图 1-2 VS.NET “新建项目”窗口

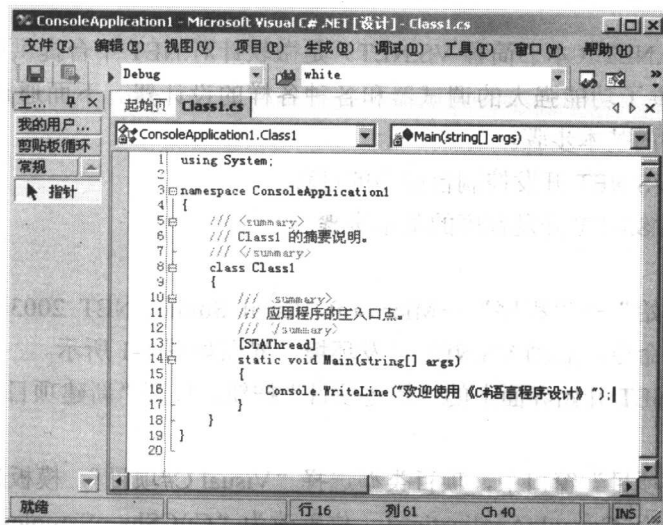


图 1-3 VS.NET “代码编辑器”窗口

4) 单击“调试->开始执行(不调试)”菜单或按 Ctrl+F5 组合键, 编译并运行程序, 结果如图 1-4 所示。

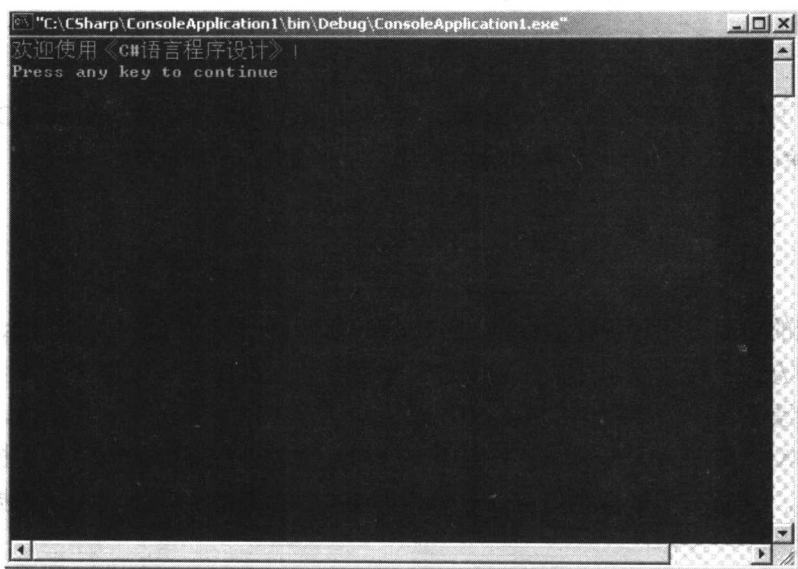


图 1-4 程序运行结果

说明: 上述 4 步即是利用 VS.NET 开发程序的基本步骤, 其中, 关于“代码编辑器”所自动生成的代码框架知识将在后续章节介绍, 本案例只要求学生初步熟悉 VS.NET 开发程序的基本步骤。

本章小结

本章详细介绍了 .NET 平台的主要内容, 重点阐述了公共语言运行库 (CLR) 和 .NET 框架类库的工作原理, 对 Visual Studio .NET 集成开发工具进行了介绍, 并通过一个案例演示了通过 VS.NET 开发程序的基本步骤, 希望学生重点熟悉 VS.NET 开发程序的基本步骤。

本章实训

案例名称: ConsoleTest

存放位置: C:\CSharpSamples\Exercise\chp1

案例要求: 开发一个控制台程序, 运行后在控制台显示两行内容, 一行是“我们伟大祖国”, 另一行是“万里锦绣河山”。

第 2 章 C#语言基础

在了解了.NET 平台的内容后, 现在就开始学习.NET 平台的原生语言——C#, 只有深入地掌握了 C#语言, 才能开发出各种应用程序。下面先从基本数据类型讲起。

2.1 基本数据类型

程序处理的对象是各种各样的数据, 因此, 必须让计算机了解需要处理什么样的数据, 以及采用哪种方式进行处理、按什么格式进行保存等。比如, 在编码程序中需要处理单个字符, 在订购票系统中需要打印货币金额, 在科学运算中需要不同精度的小数, 这些都是不同的数据类型。每一种编程语言都有它所支持的数据类型, 不同的编程语言提供的数据类型不尽相同。C#语言的数据类型可以分为两大部分: 值类型和引用类型。数据类型分类如图 2-1 所示。

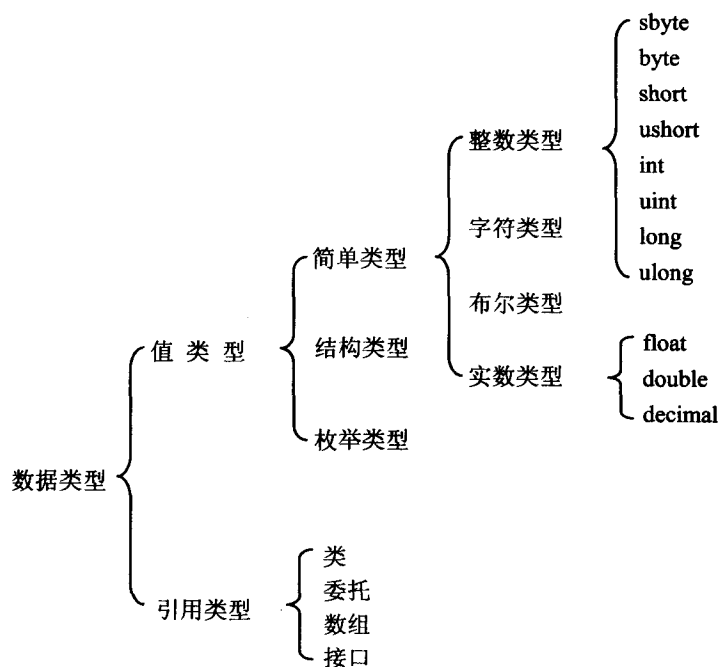


图 2-1 C#语言的数据类型

2.1.1 值类型

在具体讲解各种数据类型之前, 先提一下变量的概念, 下一节将对变量做进一步的讨

论。从用户角度来看,变量就是存储信息的基本单元;从系统角度来看,变量就是计算机内存中的一个存储空间。

值类型的变量其内含为变量的值本身,C#语言的值类型可以分为以下几种:

- 简单类型 (Simple types)。
- 结构类型 (Struct types)。
- 枚举类型 (Enumeration types)。

简单类型,有时人们也称其为内置数据类型,是C#语言中提供的、无法再分解的一种具体类型。从计算机的表示角度来看,这些简单类型可以分为整数类型、实数类型、字符类型和布尔类型。

2.1.1.1 整数类型

顾名思义,整数类型的变量的值为整数。数学上的整数可以从负无穷大到正无穷大,但是由于计算机的存储单元是有限的,所以计算机语言提供的整数类型的值总是在一定的范围之内。根据变量在内存中所占的二进制位数不同和是否有符号位,C#语言中整数类型有8种:字节型(sbyte)、无符号字节型(byte)、短整型(short)、无符号短整型(ushort)、整型(int)、无符号整型(uint)、长整型(long)、无符号长整型(ulong)。所占的二进制位数不同,表示的数值的取值范围也不同,所占的二进制位数越多,表示的数值的取值范围越大。例如8位整数,它可以表示2的8次幂个数值,即256个不同的数值,如果用来表示有符号8位整数(sbyte),其取值范围就是在-128到127之间,而如果用来表示无符号8位整数(byte),其取值范围就是在0到255之间。具体各整数类型及其取值范围如表2-1所示。

表 2-1 整数类型及其取值范围

数据类型	特征	取值范围	示例
sbyte	有符号 8 位整数	-128~127	sbyte val=16 ;
byte	无符号 8 位整数	0~255	byte val=16 ;
short	有符号 16 位整数	-32,768~32,767	short val=16 ;
ushort	无符号 16 位整数	0~65,535	ushort val=16 ;
int	有符号 32 位整数	-2,147,483,648~2,147,483,647	int val=16 ;
uint	无符号 32 位整数	0~4,294,967,295	uint val1=16 ; uint val2=16U ;
long	有符号 64 位整数	-9,223,372,036,854,775,808~ 9,223,372,036,854,775,807	long val1=16 ; long val2=16L ;
ulong	无符号 64 位整数	0~18,446,744,073,709,551,615	ulong val1=16 ; ulong val2=16U ; ulong val3=16L ; ulong val4=16UL ;

2.1.1.2 实数类型

数学中的实数不仅包括整数,而且包括小数。在C#语言中实数类型包括单精度浮点类

型 (float)、双精度浮点类型 (double) 和十进制类型 (decimal)。它们的差别主要在于取值范围和精度不同。计算机对浮点数的运算速度大大低于对整数的运算，在对精度要求不是很高的情况下，最好采用 float 型，如果对精度要求很高时，应该采用 double 型，但这样将会占用更多的内存单元，处理速度也会相对较慢。Decimal 型的取值范围比 double 型的范围要小得多，但它更精确，非常适合金融和货币方面的计算。具体各实数类型及其取值范围与精度如表 2-2 所示。

表 2-2 实数类型及其取值范围与精度

数据类型	特征	取值范围	示例
float	单精度浮点类型	$\pm 1.5 \times 10^{-45} \sim \pm 3.4 \times 10^{38}$ 精度为 7 位数	float val=6.18F ;
double	双精度浮点类型	$\pm 5.0 \times 10^{-324} \sim \pm 1.7 \times 10^{308}$ 精度为 15 到 16 位数	double val=6.18 ;
decimal	十进制类型	$\pm 1.0 \times 10^{-28} \sim \pm 7.9 \times 10^{28}$ 精度为 28 至 29 位有效数字	decimal val=6.18M ;

2.1.1.3 字符类型

除了数字以外，计算机处理的信息，主要就是字符了。在 C#语言中字符类型采用国际上公认的 16 位 Unicode 字符集表示形式，用它可以表示世界上大多数语言。其取值范围为'\u0000'~'\uFFFF'，即 0~65535。对于一个字符类型变量，可以采用如下方式对它进行赋值。

```
char c = 'H' ;           // 字符 H
char c = '\x0048' ;      // 字符 H，十六进制转义符（前缀为\x）
char c = '\u0048' ;      // 字符 H，Unicode 表示形式（前缀为\u）
char c = '\r' ;          // 回车，转义字符（用于在程序中指代特殊的控制字符）
```

常见转义字符如表 2-3 所示。

表 2-3 转义符 (Escape Sequences)

转义符	字符名
\'	单引号
\"	双引号
\\	反斜杠
\n	新行
\r	回车
\t	水平 tab
\v	垂直 tab

2.1.1.4 布尔类型

布尔类型是用来表示“真”和“假”这两个概念的。这虽然看起来很简单，但实际应用非常广泛。布尔类型表示的逻辑变量只有两种取值：“真”或“假”，主要用来在程序中

做“判断”和“分支”。在C#语言中，这两种取值分别采用 `true` 和 `false` 来表示。

这里需要说明的是，在C#语言中，不能像在C语言和C++语言中那样，用 `0` 来表示“假”，其他任何非 `0` 的式子都表示“真”。任何整数类型和布尔型之间都不能进行转换，将任何整数类型转换成布尔型都是不合法的。

```
bool x = 1;           // 错误，不存在这种写法
bool y = 1;           // 错误，不存在这种写法
bool x = true;        // 正确，可以被执行
```

2.1.1.5 结构类型

完成一些常见的数据运算和文字处理工作，使用前面介绍过的简单类型似乎已经足够。但是我们会经常碰到一些更为复杂的数据类型。例如，学生成绩单的记录中一般包含学生的学号、姓名和成绩。如果按照简单类型来处理，每一条记录都要存放到3个不同的变量当中，这样工作量很大，也不够直观。有没有更好的办法呢？

正如上面的例子，在实际生活中，我们经常把一组相关的信息放在一起。把一系列相关的变量组织成为一个整体的过程称为生成结构的过程。这个整体的类型就叫做结构类型，每一个变量称为结构的成员。结构类型采用 `struct` 进行声明。定义学生成绩单的记录结构的代码如下：

```
struct StudentScoreList
{
    public string id ;
    public string name ;
    public int score ;
}
StudentScoreList s1 ;
```

其中，`s1` 是一个 `StudentScoreList` 结构类型的变量。上面声明中的 `public` 表示对结构类型的成员的访问权限，详细内容参见 3.2.2 节访问修饰符。对结构成员的访问通过结构变量名加上访问符“`.`”，再跟成员名称：

```
s1.name = "Tom" ;
```

结构类型包含的成员类型可以是任何类型，包括可以是另一种结构类型。

```
struct StudentScoreList
{
    public string id;
    public string name;
    public struct Scores
    {
        public int mathematics ;
        public int chinese ;
        public int english ;
        public int biology ;
    }
}
```

这里，`StudentScoreList` 这个结构中又包括了 `Scores` 这个结构，`Scores` 结构类型中包括：