21 世纪高职高专规划教材·计算机系列

C#程序设计易懂易会教程

袁开鸿　主编
彭　勇　主审

清 华 大 学 出 版 社
北京交通大学出版社

·北京·

内 容 简 介

C＃程序设计语言是21世纪才开发出来的语言，近几年来相关的书籍主要面向有一定程序设计基础的读者。本书为C＃程序设计的基础教材，可以从零起点开始学习。本书始终围绕易懂易会构思内容结构和细节，主要内容有程序设计基础、类和对象、继承和多态性、委托和事件、接口和异常处理等。

本书适合作为高等院校特别是高职高专计算机及其他相关专业面向对象程序设计课程教材，也适合作为初、中级程序员的C＃面向对象程序设计的参考书。本书还是程序设计爱好者自学C＃面向对象程序设计的理想教材。

图书在版编目（CIP）数据

C＃程序设计易懂易会教程/袁开鸿主编．—北京：清华大学出版社；北京交通大学出版社，2007.12
（21世纪高职高专规划教材·计算机系列）
ISBN 978－7－81123－071－0

Ⅰ.C…　Ⅱ.袁…　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（2007）第173639号

责任编辑：刘　洵
出版发行：清华大学出版社　邮编：100084　电话：010－62776969
　　北京交通大学出版社　邮编：100044　电话：010－51686414
印 刷 者：北京市梦宇印务有限公司
经　　销：全国新华书店
开　　本：185×260　印张：22.5　字数：543千字
版　　次：2008年1月第1版　2008年1月第1次印刷
书　　号：ISBN 978－7－81123－071－0/TP·392
印　　数：1～4 000册　定价：32.00元

前　言

C＃程序设计语言是微软开发基于 .NET 平台的程序设计语言。.NET 共支持四种程序设计语言：C＃，J＃，C＋＋和 VB。C＃是专门用于 .NET 的程序设计语言，被称为 .NET 的母语。C＃具有功能强大、简单易用的特点。作为 21 世纪才开发出来的新一代程序设计语言，C＃汇集了各种程序设计语言的优点，有着其他程序设计语言无法比拟的优势。

C＃程序设计语言及其相关环境 .NET Framework 是近年来最重要的新技术。.NET 提供了一种新环境，在这个环境中，可以开发出运行在 Windows 上的所有应用程序，也可使用 C＃编写动态 Web 页面、XML Web 服务、分布式应用程序的组件、数据库访问组件等。

.NET 对编写程序的方式进行了革新，可以进行可视化（Visual）程序设计。所谓可视化程序设计是一种全新的程序设计方法，它允许程序设计人员利用软件本身所提供的各种控件，像搭积木式地构造应用程序的各种界面。可视化程序设计可以使编程者只编写少量的程序代码，就能完成应用程序的设计，极大地提高了编程人员的工作效率。充分利用可视化程序设计开发程序的前提是有 C＃面向对象程序设计的扎实基础。

本书为学习 C＃程序设计、打好程序设计基础的理想教材。本书始终围绕易懂易会构思内容结构和细节。全书分为两部分：C＃程序设计基础部分和 C＃面向对象程序设计部分。

C＃程序设计基础部分包括第 1～6 章，主要介绍程序设计的基本结构、数据类型、方法（函数）使用、字符串、数组。

第 1 章程序设计简述，介绍程序设计的发展历程、C＃编程环境及简单的 C＃程序；

第 2 章程序设计基础，介绍基本知识，主要内容有变量、常量、运算符、表达式等；

第 3 章数据类型，讲述与数据存储紧密相关的内容，主要包括值类型和引用类型；

第 4 章程序流控制，讲述程序设计的基本语句结构，主要介绍选择、循环和跳转结构；

第 5 章方法，讲述方法的定义和调用及与方法参数相关的关键字；

第 6 章字符串和数组，主要讲述字符串的定义和基本操作及数组的基本定义和使用。

C＃面向对象程序设计部分包括第 7～12 章，主要介绍类与对象、属性与索引器、继承和多态性、委托和事件、接口，以及异常处理和文件操作等。

第 7 章类和对象，主要讲述面向对象的基本概念、类和对象、构造方法及属性和索引器；

第 8 章继承和多态性，主要讲述继承、抽象类、装箱与拆箱及多态性；

第 9 章委托和事件，主要讲述委托与事件及相互关系；

第 10 章接口，主要讲述接口的定义和使用；

第 11 章异常处理，主要讲述异常、异常处理及异常类；

第 12 章文件操作，主要介绍文件的存储形式及文件的读写操作。

本书注重对学习者面向对象程序设计思维的培养，对难点内容采用理论联系实际与简单实例启发相结合的方式入手，由浅入深，深入浅出地完成面向对象程序设计的阐述。全书 140 多个例题，全部在 Microsoft Visual Studio .NET 2003 编程环境下运行通过，为方便读者学

习，每一例题都给出运行结果。

本书适合作为高等院校计算机及其他相关专业面向对象程序设计课程教材，初、中级程序员的C＃面向对象程序设计参考书。本书还是程序设计爱好者自学C＃面向对象程序设计的理想教材。

本书的编著和出版得到了许多专家和学者的大力支持和帮助。编写任务的圆满完成是一个团队共同努力的结果，其中第1～9章、第11章由袁开鸿编写，第10章由熊锡义编写，第12章由何帆编写。本书的编写得到了李德奇教授的热情指导。副教授、系统分析师翁建红和副教授、系统分析师刘志成及北京交通大学出版社的刘洵老师在编写过程中都给予了极大的支持和帮助。在这里对他们表示衷心的感谢。

由于时间仓促及作者水平有限，加上程序设计技术发展很快，尽管作者作了极大的努力，书中错误仍在所难免，恳切希望读者对本书提出宝贵的批评和建议。欢迎读者与我们联系，E－mail联系地址为ykh6512@sina.com。

编　者

2007.10

目　　录

第一篇　C#程序设计基础

第二篇 C#面向对象程序设计

21世纪高职高专规划教材·计算机系列
编审委员会成员名单

主 任 委 员　李兰友　边奠英

副主任委员　周学毛　崔世钢　王学彬　丁桂芝　赵　伟
韩瑞功　汪志达

委　　员　（按姓名笔画排序）

马春荣　马　辉　万志平　万振凯　王一曙
王永平　王建明　尤晓玮　丰继林　尹绍宏
左文忠　叶　华　叶　伟　叶建波　付晓光
付慧生　冯平安　江　中　佟立本　刘　炜
刘建民　刘　晶　刘　颖　曲建民　孙培民
邢素萍　华铨平　吕新平　陈国震　陈小东
陈月波　陈跃安　李长明　李　可　李志奎
李　琳　李源生　李群明　李静东　邱希春
沈才梁　宋维堂　汪　繁　吴学毅　张文明
张宝忠　张家超　张　琦　金忠伟　林长春
林文信　罗春红　苗长云　竺士蒙　周智仁
孟德欣　柏万里　宫国顺　柳　炜　钮　静
胡敬佩　姚　策　赵英杰　高福成　贾建军
徐建俊　殷兆麟　唐　健　黄　斌　章春军
曹豫莪　程　琪　韩广峰　韩其睿　韩　劼
裘旭光　童爱红　谢　婷　曾瑶辉　管致锦
熊锡义　潘玫玫　薛永三　操静涛　鞠洪尧

第一篇

C#程序设计基础

第 1 章　程序设计简述

本章要点：

- 程序设计的发展历程
- 使用 C#创建、编译和执行简单的应用程序
- 掌握 C#程序的简单结构
- 了解窗体、控件
- 熟悉 Visual Studio .NET 可视化集成开发环境（IDE）

学习本章内容的重点是了解程序设计的发展历程，熟悉 Visual Studio .NET 可视化集成开发环境，并掌握简单的控制台应用程序的设计及简单的可视化 Windows 窗体程序的设计。

对本章内容要认真学习，按步骤练习操作，熟悉 C#程序设计开发环境是本章最重要的内容。可以考虑多次按操作顺序练习，基本达到可以独立不看提示，完成简单 C#控制台应用程序和 Windows 窗体程序设计的要求。

1.1　程序设计的发展历程

计算机由硬件和软件构成，计算机硬件技术的发展促进了计算机软件技术的发展。现在的计算机速度越来越快，为程序设计高级语言的发展创造了条件。总的来看，程序设计的发展经历了四个阶段：机器语言、汇编语言、面向过程程序设计语言和面向对象程序设计语言。

1．机器语言

机器语言是计算机可以直接识别的语言，机器语言的数据和指令都是二进制形式。这是由于计算机发送的指令和使用的数据都是二进制形式的。程序语句分为两部分：指令部分和数据部分。

下面以字长为 8 位的 CPU（指令和操作数都是 8 位）为例介绍完成 5+7 算术运算的语句。

指令部分	数据 1	数据 2
01110010	00000101	00000111
加法	第一操作数	第二操作数

机器语言是所有程序设计语言的根，各程序设计语言编写的程序最后都要转换成机器语言形式。将人能识别的程序设计语言转换成计算机能识别的机器语言的过程称为编译。常见的后缀名为.exe 和.com 及.DLL 等的文件都是编译产生的结果。

2．汇编语言

随着机器语言的发展，为了编程方便，人们在机器语言的旁边注上一些标记（助记符），以帮助阅读程序，例如：

add	5	7
01110010	00000101	00000111
加法	第一操作数	第二操作数

由于这些助记符和机器语言的指令、操作数有一一对应的关系，时间长了人们就不用机器语言编写程序了，而改用助记符编写程序，程序编写完之后，再一次性地将程序翻译成机器语言形式，这就有了汇编语言。

例如，编写汇编程序完成 5+7 算术运算：

```
MOV     AX , 05
MOV     BX , 07
ADD     AX , BX
```

其中 AX、BX 称为寄存器。寄存器是在 CPU 内部保存数据的存储单元。MOV 是数据传送指令，表示将 5 和 7 分别传送给寄存器 AX 和 BX。ADD 是加法指令，语句：

```
ADD AX , BX
```

表示将 BX 中的数和 AX 中的数相加，结果放入 AX 中。

汇编语言相对机器语言而言前进了一大步。汇编语言与机器语言的语句之间基本上有一一对应的关系，因此，汇编语言的代码效率非常高。所以，微软 Windows 操作系统的核心部分仍有部分内容采用汇编语言编写，以提高软件的执行速度。

从前面的程序可以看到，汇编语言程序设计无法回避对寄存器的访问。CPU 内部的寄存器的使用有严格规定，不能随便使用，这给汇编语言程序的设计和学习带来了很大困难。

高级程序设计语言避开了寄存器，寄存器由编译系统自动选择使用。高级程序设计语言经历了面向过程的程序设计语言到面向对象的程序设计语言的发展阶段。

3．面向过程程序设计语言

面向过程的程序设计将程序分为两大块，即数据和处理数据的方法（有很多教材也称为函数）。数据对方法是公开的，每一个方法都可以访问数据。例如，完成 5+7 的算术运算的程序为

```
…
int   a=5;
int   b=7;
int   c;
AddMethod()
{
     c = a + b;
}
…
```

程序中有一个 AddMethod()方法，有三个整数类型的数据空间 a,b,c。a 保存十进制数 5，b 保存十进制数 7，c 用于保存 a 和 b 的和。语句：

```
c = a + b;
```

完成加法运算，结果保存在变量 c 中。

高级程序设计语言回避了对 CPU 内部寄存器的访问，使程序设计得以采用人类的自然语言形式，极大地方便了对程序的编写和阅读。高级程序设计语言编写的程序编译生成的机器代码相对于汇编语言编写的程序要长得多。

面向过程程序设计语言常用的有 Basic 语言、Fortran 语言、PASCAL 语言、C 语言等。

面向过程程序设计语言在程序设计的发展过程中起到了很好的作用，相对于汇编语言而言，程序能处理的问题的规模得到了很大的提高。

随着高级程序设计语言的发展，要求计算机完成的任务越来越多，任务的规模越来越大，面向过程程序设计将数据和处理数据的方法分离，数据的访问是公开的，所有方法都可以访问。当程序处理的问题的规模达到一定程度后，程序设计人员很难把握可公开访问的数据何时被修改了及方法调用的数据到底是不是它期望的数据等，这样就带来了数据不稳定的问题。所以，随着程序设计的发展，面向过程的程序设计已不能满足程序设计的需要。

人们首先想到的是对数据和处理数据的方法的规模进行有效的控制。初步的想法是，将数据和处理数据的方法分成几块，每一块包含相应的数据和只用来处理该块数据的方法，块与块之间通过发送消息进行相互之间的联系。这就有了面向对象程序设计的雏形。但面向对象程序设计不是简单的分块，它还有着严密的科学性和组织性。

4．面向对象程序设计语言

面向对象程序设计语言是由面向过程程序设计语言发展而来。面向过程的程序设计以具体的解决问题的过程为主体“就事论事”。面向过程的程序设计方法是针对问题定义若干数据成员和若干方法成员，展开对程序的设计。

面向对象的程序设计是以具体的解决问题过程中所涉及的各对象为主体。面向对象程序设计中的对象要通过类来定义生成，通过一个类定义的多个对象自然表现出该类对象的相同的特征。

面向对象程序设计对事件的抽象认为，类是由数据成员和方法成员构成。例如，狗类可以抽象为

- 数据成员——脚的个数、毛色、身高、体重、年龄、雌雄等；
- 方法成员——看家护院（）、吠叫（）、跑（）、扑（）、忠实主人（）等。

面向对象程序可以由多个类构成，每一个类可以由多个数据成员和多个方法成员构成。类对外公开哪些数据成员和方法成员可以根据需要决定。而且面向对象程序设计能够虚拟与问题相关的各客观对象，使现实问题在计算机中得到复现，使得处理问题的过程遵循人类认识世界的规律，极大地发挥人的认识潜能。面向对象程序设计是当前程序设计的主流。

1.2　给一个要求计算机完成的任务

直接讲解面向对象的程序设计 OOP（Object Oriented Programming）理论，是不容易理解的，也不必马上理解，任何事情都有一个开头，C#程序设计语言的学习，需要的是先操作、感受，再逐步理解。面向对象程序设计的理念之一就是通俗、易懂易学。可以把“好东西”先拿来使用，觉得好下次再用。C#是一种当前最好的面向对象程序设计语言，要学好 C#必须逐步感受、逐步理解，最终才能达到完全掌握的目的。

Visual Studio .NET 提供了大量的设计完善的程序代码（类库等），它们以程序集的形式被很好地组织在 Visual Studio .NET 中，如果需要能够很方便地引用。它们对编程人员如同日常生活中的电饭煲、洗衣机、电冰箱一样，功能强、使用方便，开发人员不需要完全理解里面的代码，就如同不用掌握电饭煲、洗衣机、电冰箱是如何设计出来。

Visual Studio .NET 有着功能强大的帮助系统，利用它可以方便地查阅有关 C#程序设计的

任何问题。

那是不是有帮助系统就不需要 C#教材了呢？不是这样的，首先帮助系统适合所有使用 Visual Studio .NET 的人员，所以会有相当多的专业术语，会阻碍初学者的学习；再就是帮助系统一般会用很精辟的语言进行描述，文字不太多，不可能长篇大论，这样也不方便初学者学习理解；另外利用帮助系统进行学习，需要单击的次数太多，不利于系统掌握。所以还是要有一本系统全面的教材。

C#程序设计语言的学习要从实例入手。下面先通过一个实例，让读者感受一下 C#程序。

例 1.1　在控制台方式下输出一行文本。

任务：显示一行文本："我在学习 C#。"

在编辑器中输入下面的内容：

```
//程序清单 P1_1.cs:
using System;
public class P1_1
{
    public static void Main()
    {
        Console.WriteLine("我在学习 C#.");
    }
}
```

运行以后，效果如图 1.1 所示。

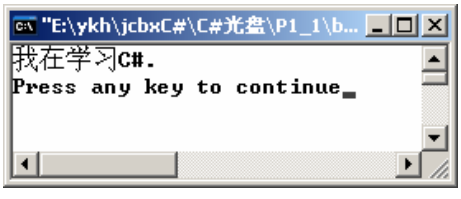

图 1.1　在控制台方式下显示一行文本

1.3　编写简单的 C#程序

1．操作步骤

① 启动 Visual Studio .NET。如果是第一次使用 Visual Studio .NET 集成开发环境，需要进行一下系统配置，这样有利于 C#程序的设计。可单击【起始页】|【我的配置文件】进行相应的设置，如图 1.2 所示。先考虑选 C#相关项，再考虑选默认值。（如果在程序设计时，界面调乱了，感觉不"友好"了，可再进入【起始页】重新配制。）

图 1.2　配置 C#编程环境

② 单击【文件】|【新建】|【项目】，打开【新建项目】对话框。

③ 在【项目类型】窗格中选择【Visual C#项目】，然后在【模板】窗格中选择【空项目】。

④ 在【名称】文本框中，键入“P1_1”作为该项目的名称；在【位置】文本框中，输入项目将要保存的目录，或单击【浏览】选择目录。

⑤ 单击【确定】按钮，创建一个“P1_1”项目，并显示解决方案资源管理器。若没有显示，可选择【视图】|【解决方案资源管理器】，或按 Ctrl+Alt+L 键。

⑥ 在解决方案资源管理器中，右击 P1_1 项目，在弹出的快捷菜单中选择【添加】|【添加新项】，打开【添加新项】对话框。

⑦ 在【添加新项】对话框的【类别】窗格中，选择【本地项目】，在【模板】窗格中选择【代码文件】，然后在【名称】文本框中，键入“P1_1”作为 C#源文件的文件名(全名为“P1_1.cs”)。

⑧ 单击【打开】按钮，输入例 1.1 的 C#源程序代码，如图 1.3 所示。

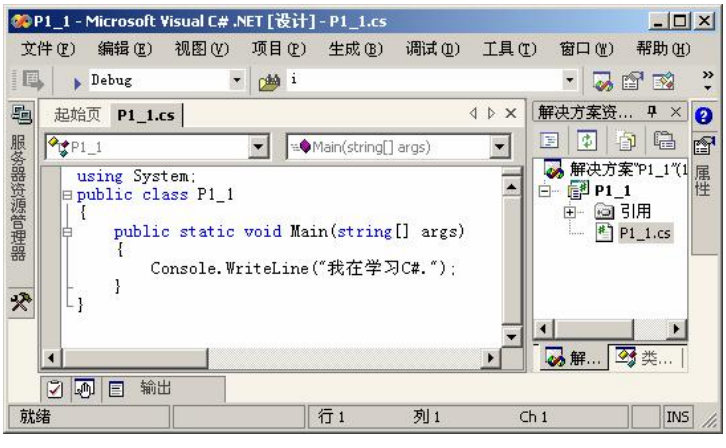

图 1.3　C# .NET 编程环境