

高职高专计算机教学改革 新体系 规划教材

C#程序设计基础

陶 南 主 编
曾文权 唐懿芳 副主编

清华大学出版社



高职高专计算机教学改革 **新体系** 规划教材

C#程序设计基础

陶 南 主 编

曾文权 唐懿芳 副主编

清华大学出版社
北 京

内 容 简 介

本书由微软金牌讲师主编。作者结合多年的教学经验,从真实项目中汲取丰富的实训素材后编写而成。本书是一本专门介绍 C# 编程语言基础的书籍。

本书系统介绍了 C# 编程语言的运行环境、基本数据类型和控制结构、方法和数组、对象初级编程等传统知识,还穿插设计了诸如软件流程图的制作、软件测试等软件工程方面的知识。在教学环节中设计了工作任务、归纳总结、技能拓展、课后习题等板块,不仅可以培养学生基本的软件开发和系统设计能力,还特别强调了自主学习拓展本专业新技术、新开发工具,获取新知识的能力。

本书既可作为高职院校计算机专业、NET 方向的专业基础课程用书,也可作为其他相关专业的编程入门课程用书,还可成为编程爱好者的入门指导书。

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# 程序设计基础/陶南主编.--北京:清华大学出版社,2013

高职高专计算机教学改革新体系规划教材

ISBN 978-7-302-32970-1

I. ①C… II. ①陶… III. ①C 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 147761 号

责任编辑:陈砺川

封面设计:傅瑞学

责任校对:袁 芳

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62795764

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:13 字 数:298 千字

版 次:2013 年 8 月第 1 版 印 次:2013 年 8 月第 1 次印刷

印 数:1~3000

定 价:27.00 元

产品编号:053589-01

前言

FOREWORD

毋庸置疑,程序设计课程作为软件专业的入门课程,在计算机科学教育中起着非常重要的作用。第一门程序设计课应该教授什么?学生能学到什么?其教材内容应如何组织?

在长期的教学过程中作者发现,一本好的教材,需要对各种问题做出合理和充分的解释,无论是语言本身,还是关于程序设计的过程,使学生在在学习时能了解问题的本质,找到解决问题的方式,而不是在模糊中自我感悟。我们认为,学生通过这门课程应该掌握以下内容。

(1) 算法式的思维及表达方式:能够读写一种正式的编程语言。

(2) 抽象能力:能简单表达复杂的概念,并有逻辑性地分解一个问题的能力。

(3) 尽善尽美:认识到程序设计并没有要记住的标准答案,但肯定有一些代码是更好的。

目前一些大专院校的程序设计课程使用的是 C 或 C++ 语言,也有的学校使用 C# 和 Java 语言。从作为入门教学语言的角度看,没有一种语言具有无可比拟的天然优势,选择时都应该综合考虑其优缺点。选择 C# 作为教学语言的主要理由是,其面向对象的编程方式及其相应的开发技术已得到了广泛的应用。C# 作为微软公司 .NET 技术的主要语言,既包含了编程需要理解的基本原理和主要机制,能满足讨论程序与程序设计问题的实际需要,又具备面向对象的特点,帮助学生建立面向对象的工程思维。

本书的内容分为以下几个部分,较为全面地涵盖了程序设计语言的知识目标和技能目标。

(1) 以程序设计为基本线索,同时对 C# 语言做深入介绍,强调如何认识程序和编写程序。通过实例讨论问题的分析和分解、设计求解过程,找出主要步骤,确定函数抽象,找出循环,选择语言结构,最后写出程序过程。

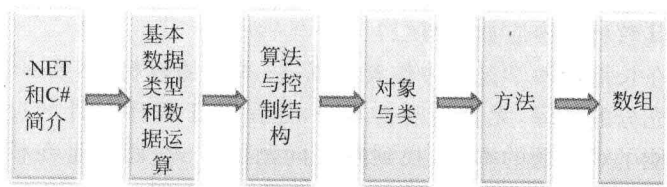
(2) 建立面向对象的程序设计思维,理解通过方法抽象建立起清晰结构的重要性。尽可能早地引进面向对象的程序设计方法,将传统的“基础先行”(fundamental-first)或时髦的“对象先行”(object-first)教学方式结合起来,形成“基础—对象并行”的内容安排模式。

(3) 强调良好的程序设计风格,从类的定义、函数的书写方式等方面,强调程序的结构性、可读性和易修改性。

(4) 培养学生对某一问题的自我扩展能力。常言道,兴趣是最好的老师。具备良好的自学能力,能更好地激励学生的学习兴趣,提高学生主动学习的积极性。

本书的内容体现了以下教学方式。一是以学习者为中心,按照循序渐进的认知规律,从易到难,分阶段安排学习和训练内容;二是注重行动能力培养,以软件项目为载体,用任务驱动的方式组织学习过程,方便做中学、学中做;三是将软件工程的相关技术融入学习和训练环节中。

本书共分为六大学习情境,由易到难地进行任务驱动教学。每个学习情境都设计了若干个工作任务,将知识体系都分布到各个工作任务中,通过一个个任务的完成让学生达到理解和消化知识点和掌握编程技能的目的。



建议本书学时为 60 学时,也可根据具体情况选择讲授的内容。

本书难度适中,内容由浅入深,主题鲜明,案例丰富,结构紧凑,实用性强,可面向所有的 C# 程序设计人员。本书可作为普通高职院校 C# 程序设计课程的教材,也可作为对 .NET 编程感兴趣的读者的参考资料。

由于时间仓促,加之作者水平有限,书中疏漏之处在所难免,欢迎广大读者批评指正,作者将不胜感激。

编 者

2013 年 4 月

目 录

CONTENTS

学习情境 1 .NET 和 C# 简介	1
任务 1.1 Hello, World! Hello C#!	1
1.1.1 进入 .NET 和 C# 世界	2
1.1.2 第一个 C# 程序	3
1.1.3 什么是“项目”	7
1.1.4 C# 打印字符图案	8
1.1.5 多项目解决方案	8
任务 1.2 熟悉 C# 的集成开发环境	10
1.2.1 开启 Visual Studio 2010 之匙	10
1.2.2 Visual Studio 2010 窗口	12
1.2.3 Visual Studio 2010 Team Suite 的安装	17
归纳总结	21
技能拓展	22
课后习题	23
学习情境 2 基本数据类型和数据运算	25
任务 2.1 求简单代数式的值	25
2.1.1 数据的分类	26
2.1.2 常量和变量	26
2.1.3 数据的输入和输出	30
2.1.4 基本数据类型	32
2.1.5 根据公式求简单代数式的值	36
任务 2.2 密码的破解	38
2.2.1 运算符与表达式	38
2.2.2 数据类型转换	46
2.2.3 对称加密技术的简单原理	48
2.2.4 程序运行结果	49
任务 2.3 数学函数和复杂公式的求解	50
2.3.1 常用数学函数	50

2.3.2 三角形面积公式的求解	52
2.3.3 一元二次方程组的求解	53
任务 2.4 随机数生成器	54
2.4.1 产生指定范围的随机数	55
2.4.2 验证码的生成	56
归纳总结	58
技能拓展	58
课后习题	61
学习情境 3 算法与控制结构	63
任务 3.1 使用 Visio 画程序流程图	63
3.1.1 程序设计的灵魂——算法	64
3.1.2 算法的最佳表示方法——流程图	65
3.1.3 Visio 工具介绍	69
3.1.4 制作基本流程图	70
3.1.5 典型算法分析	72
任务 3.2 带简单登录功能的控制台运算器	76
3.2.1 if 语句的三种形式	76
3.2.2 if 语句的嵌套	79
3.2.3 用 switch 实现多分支选择结构	81
3.2.4 控制台运算器的实现	84
3.2.5 程序运行结果	85
任务 3.3 可重复登录和计算的控制台运算器	86
3.3.1 循环语句的三种形式	86
3.3.2 多重循环	94
3.3.3 转移控制语句	97
3.3.4 可重复登录功能的实现	100
3.3.5 可重复计算功能的实现	102
归纳总结	105
技能拓展	105
课后习题	108
学习情境 4 对象与类	110
任务 4.1 使用 OOP 技术制作 ATM 取款机软件	110
4.1.1 对象	111
4.1.2 对象与类	112
4.1.3 需求分析和类的设计	115
4.1.4 类的实现	116

4.1.5 程序运行结果	117
任务 4.2 ATM 软件 Windows 版本的开发	118
4.2.1 使用属性升级 ATM 项目	118
4.2.2 Windows 窗体应用程序和常用控件	123
4.2.3 ATM 的 Windows 版本实现	127
归纳总结	131
技能拓展	131
课后习题	135
学习情境 5 类中的方法	137
任务 5.1 算术小测试	137
5.1.1 搭积木,建程序	138
5.1.2 方法的定义	139
5.1.3 Main 方法	141
5.1.4 从算法到方法	142
5.1.5 基本流程和模块实现	144
5.1.6 主函数的实现	145
任务 5.2 方法进阶	146
5.2.1 奇怪的数据交换	147
5.2.2 方法的递归调用	148
任务 5.3 人机猜拳游戏的实现	151
5.3.1 对象操作——方法	151
5.3.2 静态方法	153
5.3.3 项目分析	154
5.3.4 玩家和电脑角色类的实现	155
5.3.5 游戏类的实现	156
5.3.6 程序调用和运行	158
归纳总结	159
技能拓展	160
课后习题	165
学习情境 6 数组	167
任务 6.1 考试成绩处理系统的制作	167
6.1.1 为什么要定义数组变量	168
6.1.2 数组变量的使用	169
6.1.3 向方法传递数组	172
6.1.4 程序功能模块划分	176
6.1.5 主函数的实现	176

6.1.6 子功能模块的实现	178
任务 6.2 冒泡排序排名次	181
6.2.1 冒泡排序的原理	182
6.2.2 改进的冒泡排序	184
6.2.3 排名次方法的实现	185
6.2.4 程序运行结果	186
任务 6.3 多学科考试成绩处理系统	187
6.3.1 二维数组的定义与使用	187
6.3.2 多学科考试成绩处理系统的实现	191
归纳总结	195
技能拓展	195
课后习题	196
参考文献	198

.NET 和 C# 简介

亲爱的同学们,从现在开始,我们将进入 .NET 系列课程的学习阶段。在微软课程的学习体系中,本课程是入门级的课程,也是非常基本和重要的课程。在第一个学习情境里,你将对 .NET 和 C# 有个初步的认识,了解 .NET 的开发环境,通过动手编写第一个 C# 控制台程序,了解 C# 程序的组成,熟悉 Visual Studio 工具的安装和使用方法。

工作任务

- Hello, World! Hello C# !
- 熟悉 C# 的集成开发环境

技能目标

- 熟悉 .NET 和 C# 的历史
- 理解 C# 语言的基本特点
- 学会创建一个简单的控制台程序
- 理解基本的 C# 程序元素
- 掌握控制台工程的文件夹结构
- 熟悉 Visual Studio 2010 的各项功能
- 学会正确安装 Visual Studio 2010

任务 1.1 Hello, World! Hello C# !



任务描述

通过创建一个控制台应用程序——Hello World,掌握 C# 程序的组成和各种程序元素的含义。通过分析该程序的文件夹组成理解工程的作用。



任务分析

为了创建出第一个 C# 程序,首先应了解 C# 语言和 .NET 技术的历史和发展趋势,并掌握相应的开发工具——Visual Studio .NET。然后,同学们只需要跟着课本的步骤,循序渐进地学习就可以基本掌握 C# 编程技术了。

1.1.1 进入.NET 和 C# 世界

.NET 是微软于 2000 年推出的一种全新的开发平台。对 .NET 的需求源于 Internet 的高速发展。应用程序的概念不再单一地是桌面运行的可执行文件,应用程序的模式已经发生了根本性的变化,这些变化主要体现在以下方面。

- (1) 应用程序应随时、随地并且在任何设备中都可用。
- (2) 软件作为一种服务来提供,例如电子邮件服务。
- (3) 应用程序必须具有互操作性,应用程序之间协同工作,数据交换的需求也越来越高。

Microsoft .NET 就是在这样的大环境下诞生的。.NET 刺激了下一代计算技术的发展,加速了新一代 Internet 的来临,并为后一代应用程序的开发提供了便利。它为创建、部署以及管理安全、强大、高效的应用程序提供了前所未有的最大支持。

通过 .NET 平台,用户不仅可以开发出功能更强大的应用程序,而且还可以开发出丰富的 Web 应用和服务。

.NET 有一个庞大的技术体系,读者如何制定自己的学习步骤呢?图 1.1 为读者显示了 .NET 技术学习全景图。

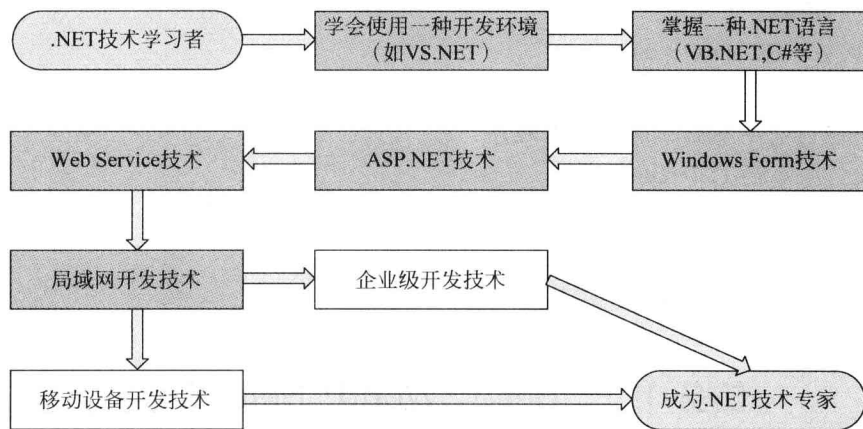


图 1.1 .NET 技术学习全景图

通过图 1.1 可以看出, .NET 目前主要的开发方向是 Web 开发、桌面系统开发、移动开发。

(1) Web 开发通俗地说就是开发网站, Web 开发是目前 .NET 开发的主要方向, 目前大部分招聘 .NET 的职位都是做 Web 开发。

(2) 桌面系统一般是指使用 Windows 窗体技术开发 Windows 软件, 比如以 Windows 软件形式运行的仓库管理系统、图书管理系统等。

(3) 移动开发主要指的是手机开发。 .NET 手机开发目前主要是 Windows Mobile 的开发。微软在 2012 年 6 月正式推出了 Windows Phone 8(简称 WP8)手机操作系统。对移动开发感兴趣的用户可以关注 Windows Phone 的开发。

学习需要循序渐进,在掌握上述核心开发技术之前,必须先学会使用基于 .NET 平台的开发环境——Visual Studio .NET 并熟练掌握一种支持 .NET 的核心开发语言 C#。

作为 .NET 平台下的静态强类型语言,C# 在过去 10 年的发展历程中披荆斩棘,已经逐渐成为应用开发语言中的佼佼者。C# 是专门为 .NET 应用而开发出的语言,这从根本上保证了 C# 与 .NET 框架的完美结合。在 .NET 运行库的支持下,.NET 框架的各种优点在 C# 中表现得淋漓尽致。同学们可以先来看看 C# 的一些突出的特点,相信在以后的学习过程中将会深深地体会到“#”——“SHARP”(锋利)的真正含义。

C# 具有如下特点。

(1) 面向对象编程。C# 实现对属性、事件、委托、方法、索引器、构造器的全面支持,为面向对象的封装、继承、多态和接口提供了语言级别的支持。以继承为例,C# 支持单实现继承和多接口继承,摒弃了 C++ 中多继承带来的复杂性。

(2) 跨平台运行时支持。CLR 是 .NET 平台下应用程序的通用语言运行时,是 C# 程序赖以生存的跨平台环境,因此 C# 具有 .NET 平台语言的所有优势,通用类型系统、自动内存管理、统一异常处理、完全的 FCL 访问权,都成为 C# 无与伦比的优势所在。

微软推出了这么强大的平台和技术,当然也会有强大的集成开发环境(IDE)来支持,那就是 Visual Studio .NET(简称 VS),它集成了 .NET Framework。利用 Visual Studio .NET,用户可以快速开发 .NET 应用程序。

1.1.2 第一个 C# 程序

对 .NET Framework、C# 和 VS 有了初步的认识后,我们来看看 C# 程序的庐山真面目吧。首先启动 VS,单击“开始”→“程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”选项,如图 1.2 所示。

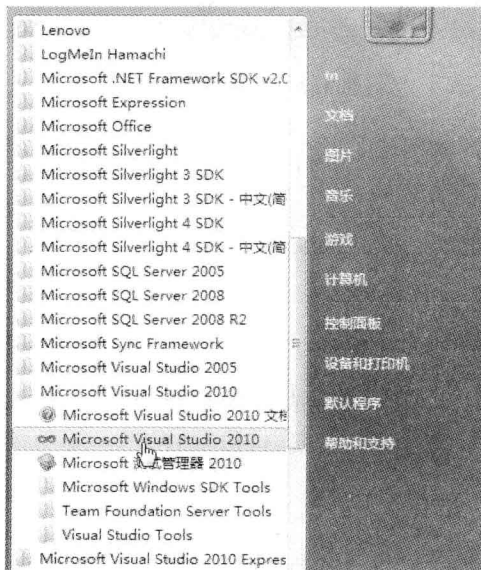


图 1.2 启动 Visual Studio 2010

使用 VS 2010 创建并运行控制台应用程序包括 3 个步骤：新建项目→生成解决方案→调试。

1. 新建项目 (Project)

- (1) 在 VS 菜单栏中单击“文件”→“新建”→“项目”选项，打开“新建项目”对话框。
- (2) 在左侧的项目类型中选择“Visual C#”，在右侧的模板列表中选择“控制台应用程序”。
- (3) 在“名称”文本框中输入“HelloCSharp”。
- (4) 为项目选择一个保存位置，如“D:\MyFirstApp”，如图 1.3 所示。



图 1.3 新建项目窗口

单击“确定”按钮之后，就创建了一个 C# 代码模板，如图 1.4 所示。

在 Main 方法中添加如下代码。

```
Console.WriteLine("Hello, World! Hello, C#!");  
Console.ReadLine();
```

2. 生成解决方案 (Build Solution)

在 VS 菜单栏中单击“生成”→“生成解决方案”选项。如果在 VS 状态栏底部显示“生成成功”，就表示代码不存在编译错误，可以运行了。

3. 调试 (Debug)

在 VS 菜单栏中单击“调试”→“启动调试”选项。如果看到如图 1.5 所示的结果，就表示已经成功运行出自己的第一个 C# 程序了！

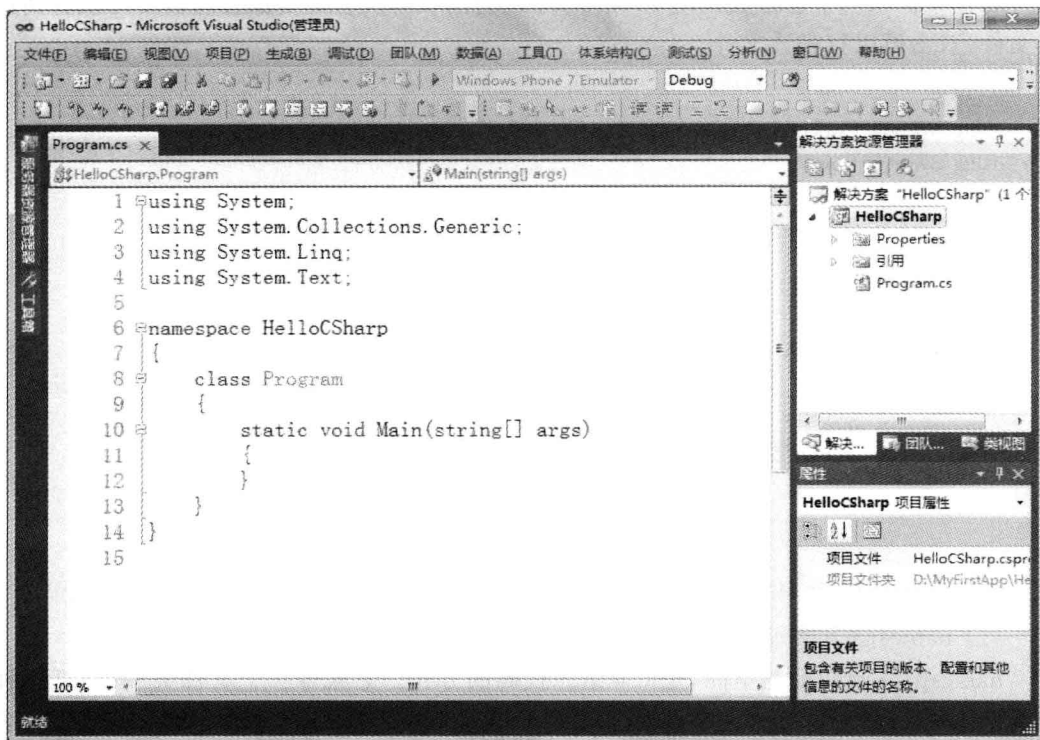


图 1.4 C# 代码模板

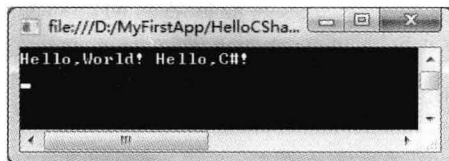


图 1.5 HelloCSharp 项目输出

Tips: Hello World 的由来

“Hello,World”程序指的是只在计算机屏幕上输出“Hello,World!”(意为“世界,你好!”)这行字符串的计算机程序。一般来说,这是每一种计算机编程语言中最基本、最简单的程序,通常也是初学者所编写的第一个程序。它还可以用来确定该语言的编译器、程序开发环境以及运行环境是否已经安装妥当。

“Hello World”示例程序最早出现于1972年由贝尔实验室成员 Brian Kernighan 撰写的内部技术文件《Introduction to the Language B》之中。不久同作者于1974年所撰写的《Programming in C: A Tutorial》也沿用这个示例。本书也保留了这个示范程序。

现在来认识一下存放代码的 Program.cs 文件,在解决方案资源管理器中可以看到这个文件,双击可以打开该文件。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace HelloCSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World! Hello, C#!");
            Console.ReadLine();
        }
    }
}
```

在这个代码段中,有很多代码是自动生成的。先来逐个看看这些自动生成的代码。

(1) namespace 关键字

namespace 也叫命名空间,是 C# 组织代码的方式。它就像一个文件夹一样,把各种相关的 class 类放在一起,大大提高了编码的效率。在上述代码中,VS 自动以项目的名称 HelloCSharp 作为命名空间的名称。

(2) using 关键字

在 C# 中,使用 using 关键字来导入其他命名空间。如果要在代码中使用到某一个命名空间中的类 A,则需要告诉这段代码类 A 所在的命名空间,此时可以在程序的开始位置导入这个命名空间。在上述代码中,系统自动添加了 4 条 using 语句。

(3) class 关键字

C# 是一种面向对象的语言。一个类可以创建多个不同的对象。在 C# 中使用 class 关键字表示类。用户编写的任何代码都要放在一个类中,而类必须包含在命名空间中。在程序模板生成时,VS 自动为用户起了一个类名 Program。如果不喜欢,可以改掉它。

(4) Main 方法

Main 方法是用户程序的大门,应用程序从这里开始执行。系统将自动为用户创建一个 Main 方法。关于 Main 方法的知识将在学习情境 5 中予以阐述。

(5) 关键代码

在程序中添加如下代码。

```
Console.WriteLine("Hello World!Hello C#!")
Console.ReadLine();
```

第一句话表示在控制台窗口上输出一个字符串。Console.WriteLine 是用于输出的方法。准确地说,方法名称叫作 WriteLine,而该方法所在的类名是 Console。调用方法时,使用“类名.方法名()”的调用格式。要输出的内容写在方法后面的小括号里,因为是字符串,需要用双引号""进行包含。

第二句话也是执行一个方法。该方法等待用户的输入,如果用户在键盘上输入一个字符串,该方法将获得这个字符串。这句话在这里的作用是使窗口停留,直到用户按下任意键。

关于这两个方法的详细说明将在学习情境2中描述。

1.1.3 什么是“项目”

一个真实的商业程序可能需要几十万行代码,将这些代码放在一个文件中进行编辑是不可行的。VS将这些代码按照功能的划分分别保存到不同的文件夹和文件中。在这些文件中,只有一个文件能包含 Main 方法。

项目就是将这些文件组织起来的一个手段。项目本身就是一个文件,它的后缀名为 .csproj。但这个文件本身并不代表程序,而仅仅是起到组织多个文件的作用。由于 .csproj 文件不是一个文本文件,因此不能用 Open File 打开它,而只能用 Open Project 打开它。

项目文件里的 XML 信息会告诉编译器完整的程序由哪些文件组成,这些文件的位置在哪里,程序的版本以及默认的输出/输入路径等。所以,要想将一个项目从一台计算机复制到另一台计算机,仅仅复制 .csproj 文件是不行的,必须复制全部的文件。最佳的做法是将包含这些文件的文件夹整体复制过去,如图 1.6 所示。



图 1.6 HelloCSharp 项目文件夹

在上面的图里还可以看到 Program.cs 文件。该文件就是该项目的源代码文件,扩展名为 .cs。用户将在 .cs 文件中编写 C# 的核心代码。

此外,项目文件夹中还有 3 个子目录,分别是 bin、obj 和 Properties。

bin 目录用来存放编译的结果。bin 是二进制 binary 的英文缩写,因为最初 C 编译的程序文件都是二进制文件,它有 Debug 和 Release 两个版本,分别对应文件夹 bin/Debug 和 bin/Release,这两个文件夹是默认的输出路径,用户可以通过“项目属性”→“配置属性”→“输出路径”命令来修改。

obj 是 object 的缩写,用于存放编译过程中生成的中间临时文件。其中也有 debug 和 release 两个子目录,分别对应调试版本和发行版本,在 .NET 中,编译是分模块进行的,编译完成后会将各模块合并为一个 .dll 或 .exe 文件保存到 bin 目录下。

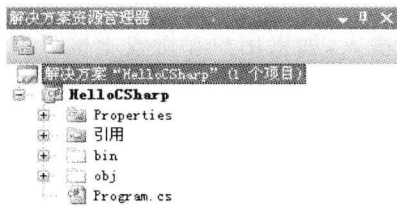


图 1.7 HelloCSharp 项目结构

Properties 文件夹定义用户程序集的属性。一般只有一个 AssemblyInfo.cs 类文件,用于保存程序集的信息,如名称、版本等。这些信息一般与项目属性面板中的数据对应,不需要用户手动编写。

上面的内容介绍了项目的物理文件组成,其实,在 VS 中的解决方案资源管理器中可以更好地管理这些项目。图 1.7 所示就是 HelloCSharp 在解决方案资源管理器中的组织结构。一个解决方案可以包含多个项目,解决方案对应的文件扩展名为 .sln。

1.1.4 C# 打印字符图案

新建一个项目 PrintShape,打印如图 1.8 所示的图案。

这个问题可以直接使用 Console.WriteLine() 方法来实现。这个图案有 7 行,调用一次 Console.WriteLine() 方法输出一行,一共调用这个方法 7 次。

代码如下。

```
static void Main(string[] args)
{
    Console.WriteLine(" * ");
    Console.WriteLine(" *** ");
    Console.WriteLine(" ***** ");
    Console.WriteLine(" ***** ");
    Console.WriteLine(" ***** ");
    Console.WriteLine(" *** ");
    Console.WriteLine(" * ");
}
```



图 1.8 侧三角形字符图案

1.1.5 多项目解决方案

在上述的“解决方案资源管理器”窗口,在根节点上右击,单击“添加”→“新建项目”选项,为该方案新建一个项目,如图 1.9 所示。

在弹出的窗口中为项目指定名称: PrintShape2,如图 1.10 所示。

单击“确定”按钮。解决方案资源管理器如图 1.11 所示。