

高等院校教学改革示范教材 软件开发系列

C# 程序设计

项目教程

◎ 温一军 周洪斌 主编

- ✓ 通过Demo→Activity→Project，降低学习门槛，循序渐进培养编程能力，激发学习积极性
- ✓ 秉承“教学做合一”原则，详细讲解任务，手把手教学，注重编码规范，强调动手能力的培养
- ✓ 突出校企合作，内容选取上注重实用性，通过与企业专家合作，反映职业岗位技能要求
- ✓ 配套教学资源包括PPT课件以及Demo、Activity、Project源程序

登录华信教育资源网，获取电子教案及课件、案例素材、技能强化综合练习题等资源：
<http://www.hxedu.com.cn>



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

高等职业院校教学改革创新示范教材·软件开发系列

C#程序设计项目教程

温一军 周洪斌 主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书较为全面地介绍了基于 C# 的面向对象编程技术。在编写过程中创新性地采用“DAP 教学模式”，通过 Demo→Activity→Project，降低学习的门槛，循序渐进地培养学生的编程能力，激发学生的学习积极性；同时，秉承“教学做合一”的原则，对各个任务进行详细讲解，开展手把手教学，并在整个过程中注重编码规范，强调能力培养，使学生在不知不觉中养成企业开发规范，步入软件开发大门；在内容的选取上注重实用性，通过与企业专家合作，从高等职业院校学生的实际出发，强调实际、实用、实践，加强技能培养，突出工程实践，合理反映职业岗位技能要求。

本书可作为应用型本科、高等职业院校相关专业 C# 程序设计专业的教材，又可作为软件公司对新员工开展岗前培训和社会培训机构的培训用书，也可供编程爱好者学习和参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

C# 程序设计项目教程 / 温一军，周洪斌主编. —北京：电子工业出版社，2014.8

ISBN 978-7-121-24246-5

I. ①C… II. ①温… ②周… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 197728 号

策划编辑：程超群

责任编辑：郝黎明

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：15 字数：384 千字

版 次：2014 年 8 月第 1 版

印 次：2014 年 8 月第 1 次印刷

定 价：31.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前 言

C#是微软公司推出的业界主流的程序设计语言。掌握 C#程序设计语言,对于在校学生、应聘求职者具有极其重要的意义。本书以.NET 应用程序设计与开发这一职业岗位能力培养为核心,在编写过程中,本书创新性地采用了“**DAP 教学模式**”:

(1) Demo(基础训练):通过 Demo,教师示范讲解知识点、技能点,然后让学生重演 Demo,使学生初步理解知识、掌握技能。

(2) Activity(提高训练):安排在 Demo 之后,Activity 案例与 Demo 的技能点要求相同,但由学生在课堂上独立完成,主要训练学生融会贯通、灵活应用技能点的能力;同时,安排课外 Activity(巩固训练),安排学生在课外独立完成,使学生进一步巩固知识点、技能点。

(3) Project(项目实战):安排在课程设计(实训)阶段,布置学生完成若干个来源于企事业单位实际需求的项目,综合培养学生的项目开发能力。

因此,本书具备如下特色与创新之处:

(1) 通过 **Demo→Activity→Project**,降低学习的门槛,循序渐进地培养学生的编程能力,激发学生学习的积极性,使学生体会到软件项目开发的乐趣,享受成功进行软件开发的成就感。

(2) 秉承“教学做合一”的原则,书中包含大量案例,并对各个任务进行详细讲解,开展手把手教学,老师边讲,学生边做,讲练结合,在做中讲,在做中练,学以致用。在做的过程中理解概念,掌握技能;并在整个过程中,注重编码规范,强调能力培养,使学生在不知不觉中掌握企业开发规范,步入软件开发大门。

(3) 突出校企合作,在内容的选取上注重实用性。通过与企业专家合作,从高等职业院校学生的实际出发,强调实际、实用、实践,加强技能培养,突出工程实践,合理反映职业岗位技能要求。

本书是校企合作的成果,由沙洲职业工学院的温一军、周洪斌老师担任主编,董袁泉、陈立平、于淑香老师担任副主编,由苏州经贸职业技术学院李冬教授、沙洲职业工学院许礼捷副教授担任主审。其中,于淑香老师负责第 1~3 章的编写工作,董袁泉老师负责第 4 章、第 6 章的编写工作,陈立平老师负责第 8 章、第 9 章的编写工作,周洪斌老师负责第 5 章、第 7 章及附录的编写工作。江苏国泰新点软件有限公司高级工程师李强以及 NIIT 张家港软件与服务外包学院技术总监李经纬参与了本书的编写工作,为本书提供了众多来源于企业实际需求的案例。全书由温一军、周洪斌老师统稿。

本书凝聚了编者近几年来学习.NET 技术、开发.NET 软件、讲授.NET 课程的心得、体会,希望本书能为我国的软件人才培养起到推动作用,希望读者能凭借本书进入.NET 开发大门。

同时,请读者注意,软件开发是一项非常注重实践的工作,不可能只凭看书、看视频就学会软件开发,必须扎扎实实、一行一行地编写代码,不断积累项目经验,才能真正掌握编程技术。所以读者一定要亲自上机编写程序,才能真正掌握软件开发技术。读者如果能按照本书的要求,循序渐进地完成 Demo→Activity→Project,编程能力必将有一个质的飞跃。

本书是沙洲职业工学院 2013 年度教育教学改革重点课题“基于 DAP 高职教学模式改革的探索与实践”(课题编号:2013SGJG002)的阶段性的研究成果,同时受江苏省“青蓝工程”资

助。感谢电子工业出版社的大力支持，感谢所有在出版过程中给予编者帮助的人，谢谢你们让本书顺利付梓。

尽管编者在编写方面做了很大努力，但由于水平有限，不当之处在所难免，恳请读者批评指正（编者电子邮箱：itzhb@qq.com）。

编 者

目 录

第1章 初识C#	(1)
1.1 C#简介	(2)
1.2 Visual Studio 开发环境	(3)
1.3 C#程序结构	(4)
1.3.1 第一个C#程序	(4)
1.3.2 C#程序结构分析	(6)
1.4 输入/输出操作	(9)
本章小结	(11)
课外 ACTIVITY	(12)
第2章 C#编程基础	(13)
2.1 数据类型	(14)
2.1.1 值类型	(14)
2.1.2 引用类型	(19)
2.2 变量与常量	(20)
2.2.1 变量	(21)
2.2.2 常量	(24)
2.3 类型转换	(24)
2.3.1 隐式转换	(25)
2.3.2 显式转换	(25)
2.3.3 装箱和拆箱	(27)
2.4 运算符与表达式	(29)
2.4.1 算术运算符及其表达式	(29)
2.4.2 赋值运算符及其表达式	(32)
2.4.3 关系运算符及其表达式	(33)
2.4.4 逻辑运算符及其表达式	(33)
2.4.5 其他运算符	(34)
2.4.6 运算符的优先级	(36)
本章小结	(37)
课外 ACTIVITY	(37)
第3章 流程控制	(38)
3.1 算法的概念	(39)
3.2 顺序结构	(40)
3.3 选择结构	(41)
3.3.1 if 语句	(41)
3.3.2 switch 语句	(51)
3.4 循环结构	(54)
3.4.1 while 语句	(54)

3.4.2	do- while 语句	(56)
3.4.3	for 语句	(57)
3.4.4	foreach 语句	(59)
3.5	跳转语句	(60)
3.5.1	break 语句	(60)
3.5.2	continue 语句	(61)
3.6	循环嵌套	(62)
	本章小结	(64)
	课外 ACTIVITY	(65)
第4章	枚举、数组和字符串	(66)
4.1	枚举	(67)
4.2	数组	(69)
4.2.1	声明	(69)
4.2.2	初始化	(70)
4.2.3	一维数组的使用	(70)
4.3	字符串	(72)
4.3.1	声明字符串变量	(72)
4.3.2	字符串的比较	(72)
4.3.3	字符串的定位	(74)
4.3.4	提取子串	(76)
4.3.5	拆分字符串	(77)
4.3.6	连接字符串	(78)
4.3.7	修改字符串	(79)
4.3.8	格式化字符串	(79)
4.4	StringBuilder 类	(80)
4.5	正则表达式	(82)
4.5.1	通配符	(82)
4.5.2	Regex 类的常用方法	(84)
	本章小结	(86)
	课外 ACTIVITY	(86)
第5章	类的定义和使用	(87)
5.1	类和对象	(88)
5.2	字段	(88)
5.3	属性	(91)
5.4	方法	(96)
5.4.1	定义方法	(96)
5.4.2	方法参数	(101)
5.4.3	方法重载	(105)
5.5	构造方法	(106)
5.6	静态成员	(111)

5.7 this 关键字	(115)
5.7.1 访问当前对象成员	(115)
5.7.2 定义索引器	(117)
本章小结	(122)
课外 ACTIVITY	(122)
第 6 章 集合及泛型集合	(123)
6.1 集合	(124)
6.1.1 ArrayList 类	(124)
6.1.2 Hashtable 类	(128)
6.2 泛型集合	(131)
6.2.1 List<T>类	(132)
6.2.2 Dictionary<K,V>类	(134)
本章小结	(136)
课外 ACTIVITY	(136)
第 7 章 继承和多态性	(137)
7.1 类的继承	(138)
7.1.1 继承	(138)
7.1.2 继承的特性	(144)
7.1.3 类型转换	(145)
7.1.4 Object 类	(146)
7.1.5 访问权限	(147)
7.1.6 继承条件下对象创建过程	(148)
7.1.7 base 关键字	(149)
7.2 抽象类	(153)
7.2.1 引入抽象类	(153)
7.2.2 定义抽象类	(161)
7.3 多态性	(164)
7.3.1 多态	(164)
7.3.2 虚方法	(164)
7.4 接口	(170)
7.4.1 引入接口	(170)
7.4.2 自定义接口	(178)
7.4.3 泛型接口	(183)
本章小结	(191)
课外 ACTIVITY	(192)
第 8 章 文件读写及异常处理	(193)
8.1 文件读写	(194)
8.1.1 File 类和 FileInfo 类	(195)
8.1.2 Directory 类和 DirectoryInfo 类	(197)
8.1.3 Path 类和 Environment 类	(199)

8.1.4	FileStream 类	(201)
8.1.5	StreamReader 类和 StreamWriter 类	(204)
8.1.6	BinaryReader 类和 BinaryWriter 类	(208)
8.2	异常处理	(210)
	本章小结	(214)
	课外 ACTIVITY	(214)
第 9 章	项目训练	(219)
	项目训练 1 学院薪资管理系统	(219)
	项目训练 2 企业薪资管理系统	(221)
	项目训练 3 电器商场库存管理系统	(224)
	项目训练 4 校园速递管理系统	(226)
附录	C#编码规范	(228)
	参考文献	(229)

第 1 章 初识 C#

教学目标

通过本章的学习,使学生了解 Visual Studio 集成开发环境,掌握 C#程序设计的基础知识,包括新建、编译及执行 C#程序,认识 C#程序的基本结构,掌握控制台程序的输入/输出操作等。

知识点

1. C#程序的组成结构
2. 新建、编译及执行 C#程序
3. Console 类的输入/输出方法

技能点

1. 编写自定义的 C#程序
2. 调试 C#程序
3. 编译、执行 C#程序
4. 使用 Console 类的输入/输出方法

重点难点

1. C#程序的一般书写格式
2. Console 类的输入/输出方法
3. 在 Visual Studio 集成开发环境下建立和运行 C#程序

专业英文词汇

1. namespace: _____
2. solution: _____
3. project: _____
4. class: _____
5. program: _____
6. main: _____
7. console: _____

- 8. read: _____
- 9. readline: _____
- 10. write: _____
- 11. writeline: _____

1.1 C#简介

介绍 C#语言, 首先要了解一下它的诞生。人们先是设计出了高效灵活的 C 语言, 又在 C 语言中加入了面向对象技术, 产生了 C++语言。C/C++一直是最有生命力的编程语言, 它们提供了强大的功能和高度的灵活性, 拥有完整的底层控制能力。但是, 它们也有缺点, 即学习过程和开发周期较长, 而且内存泄漏和缓冲区溢出的问题也使很多初学者头痛不已。而许多开发效率更高的语言, 如 Visual Basic, 在功能方面又有局限性。因此, 软件行业同仁们在寻找一种全新的现代程序设计语言, 它既能在控制能力和生产效率之间达到平衡, 又能将高端应用开发与底层平台访问紧密结合在一起, 并与 Web 标准保持同步。于是, C#语言应运而生。

微软公司发布了 C# (读音: C Sharp) 的编程语言。C#语言源于 C++, 吸取了 C++的长处, 采用面向对象技术, 提供快捷的开发方式; 摒弃了 C++的缺点, 剔除了 C++中一些复杂和容易出错的语法, 这使 C#语言在功能强大的同时又简单易学, 可谓鱼和熊掌兼得。C#语言具有以下几方面的突出特点。

(1) 语法简洁流畅

C#虽源自 C/C++, 但与之相比, C#最大的特点是不允许直接操作内存, 去掉了指针操作。另外, C#使用有限的指令、修饰符和操作符简化了 C++中的语法冗余, 整个语言结构十分清晰, 简洁流畅。

(2) 面向对象设计

C#是面向对象的语言, 以类为基础构建所有类型, 并通过命名空间对代码进行层次化的组织和管理。同时, C#不但支持封装、继承、多态等面向对象技术的基本特征, 还增加了特性、属性、委托、可容类型等强大功能。

(3) 与 Web 紧密结合

C#语言既适合传统编程, 又适合 Web 编程, 支持绝大多数的 Web 标准, 使网络开发和本地开发几乎一样简单。利用 C#组件能快速地开发 Web 服务, 并通过 Internet 被运行于其他任何平台上的应用所调用。

(4) 完善的安全性

C#具有完善的安全机制, 符合通用类型系统的类型安全性要求。例如, 垃圾回收机制 (Garbage Collection) 自动回收无用对象占用的内存, 开发人员无需负担内存管理的任务, 应用程序的可靠性得到了提高。

(5) 强大的错误处理能力

错误的处理能力是衡量一种编程语言优秀与否的重要标准, C#提供了强大的错误和异常触发机制。异常处理 (Exception Handling) 提供了结构化的、可扩展的错误检测和处理方法; 安全类型 (Type-safe) 避免了“引用未初始化的变量”、“数组索引越界”、“执行未经检查的强制类型转换”等问题。

（6）灵活版本处理技术

C#语言中内置了版本控制功能，并通过接口和继承来实现应用程序的可扩展性。使程序设计人员更容易地开发和维护各种商业应用。

另外，C#中允许使用非托管代码来与其他程序进行集成和交互，具有良好的灵活性和兼容性。

总之，C#是一种简单易学、类型安全和完全面向对象的高级程序设计语言，在接下来的学习实践中，会更多地体会和领悟 C#的魅力。

1.2 Visual Studio 开发环境

Microsoft 推出集成开发环境 Visual Studio .NET，支持 C#等各种.NET 语言的可视化编程开发，使开发人员能够方便快速地创建、调试和发布程序，从而快速构建各类.NET 应用。本书进行代码演练所用的环境是 Visual Studio 2010，简单介绍如下。

选择“开始”→“所有程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”命令，就可以启动 Visual Studio 2010 .NET 开发环境。Visual Studio 2010 的起始页如图 1-1 所示。

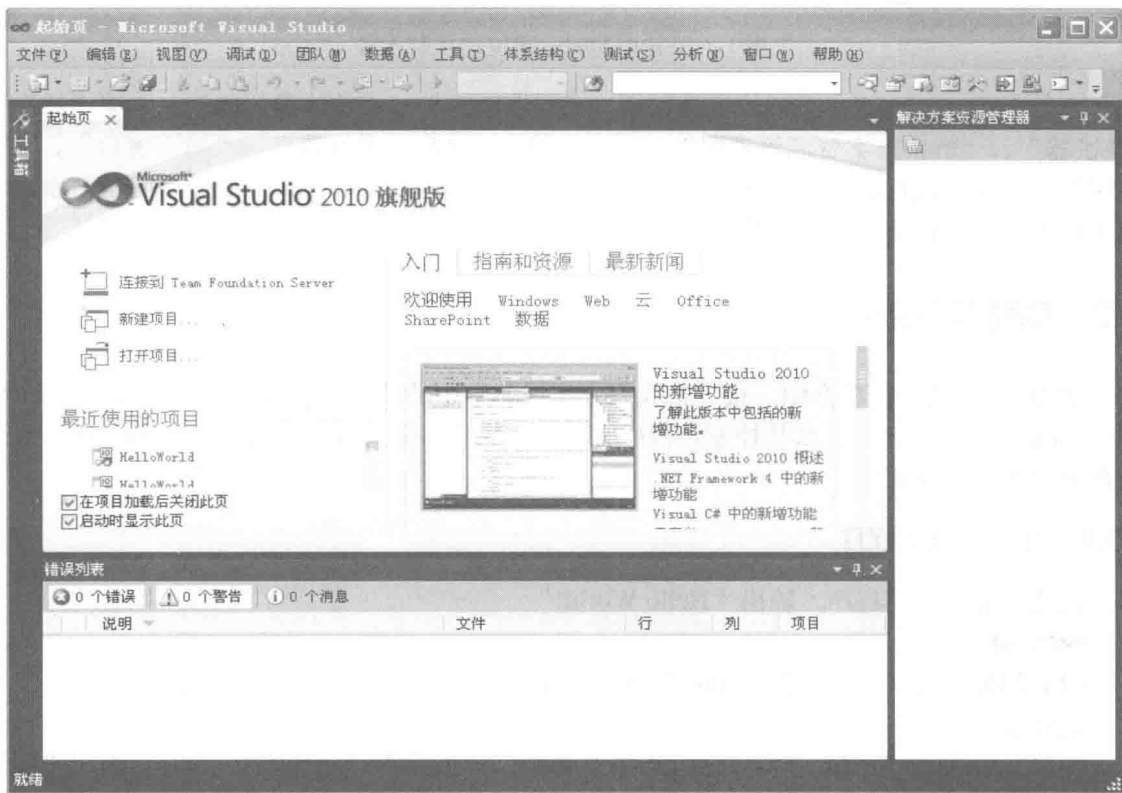


图 1-1 Visual Studio 2010 的起始页

起始页是集成开发环境内 Web 浏览器的默认主页，通过该界面可以创建新项目或打开现有项目，查询网上相关信息和设置配置文件，以有效地使用 Visual Studio 2010。Visual Studio 2010 开发环境如图 1-2 所示。

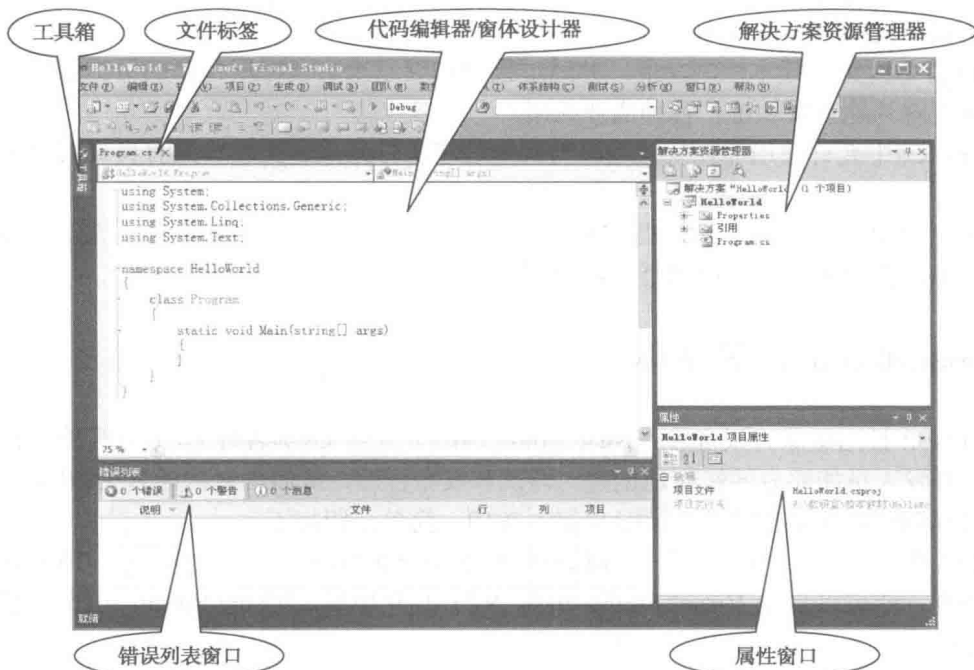


图 1-2 Visual Studio 2010 的开发环境

Visual Studio 2010 中重要的工具包括“代码编辑器”、“窗体设计器”、“编译器”、“调试器”、“解决方案资源管理器”、“工具箱”、“项目设计器”、“属性窗口”、“类视图”、“对象浏览器”等，这些窗口大多可通过“视图”菜单打开，在后面的学习中，将逐步深入地了解和使用这些工具。

1.3 C#程序结构

前面对 C#做了简单介绍，接下来我们来实践本书中的第一个 C#应用程序，这是一个控制台（Console）程序。什么是控制台程序呢？控制台程序就是在命令行下运行的程序，没有图形用户界面（GUI）。

1.3.1 第一个 C#程序

创建控制台应用程序，输出“Hello World!”

DEMO

(1) 创建项目（项目名称：HelloWorldDemo）

具体步骤如下。

- ① 启动 Visual Studio 开发环境，选择“文件”→“新建”→“项目”命令，弹出“新建项目”对话框，如图 1-3 所示。
- ② 在左侧“已安装的模板”中选择“Visual C#”选项。
- ③ 在右侧“模板”中选择“控制台应用程序”选项。
- ④ 在“名称”文本框中输入“HelloWorld”。
- ⑤ 在“位置”文本框中，通过“浏览”按钮选择项目所在的目录。

⑥ 单击“确定”按钮。

这样，我们就创建了一个名为“HelloWorld”的解决方案，包含一个 HelloWorld 项目。

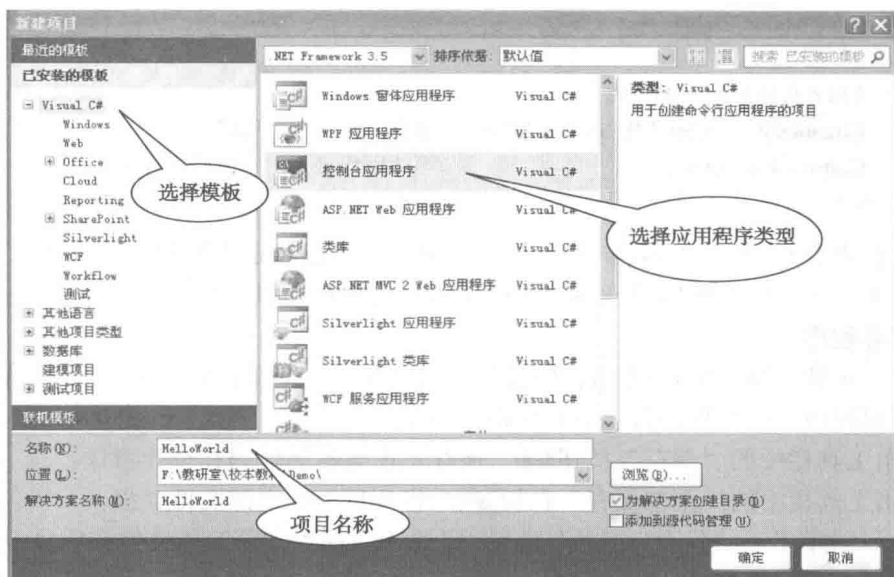


图 1-3 “新建项目”对话框

Visual Studio 中，用解决方案（Solution）组织应用程序，一个解决方案可以包含一个或多个项目（Project）。

(2) 编写代码

在成功创建解决方案后，Visual Studio 会自动生成控制台应用程序的代码框架，如图 1-4 所示。



图 1-4 控制台程序代码编辑窗口

“Program.cs”文件已自动在代码编辑器中打开，其中包含 Main 函数，该函数是 C#程序的主函数，程序执行的入口。接下来，在 Main 函数编写程序要执行的代码。

```
static void Main(string[] args)
{
    //写入程序要执行的代码
    Console.WriteLine("Hello World!"); //输出“Hello World!”
    Console.ReadLine();               //等待用户输入，结束应用程序
}
```

在控制台中，有一个 Console 类，这个类中有很多已经定义好的常用方法。WriteLine()方法就是其中一个，它的作用是把参数中的文本输出到计算机屏幕上。

(3) 保存程序

经过以上步骤，第一个 C#应用程序就创建成功了。在编译和运行应用程序之前，用户一般要保存应用程序。保存程序有以下几种操作方法。

- ① 单击工具栏中的“保存”按钮，保存正在编辑的代码和设计窗体。
- ② 单击工具栏中的“全部保存”按钮，将项目中的全部文件存盘。
- ③ 选择“文件”→“保存”或“全部保存”命令，保存当前窗体文件或项目中全部文件。

(4) 编译、运行程序

编译程序，按快捷键 F6，或者选择“生成”→“生成解决方案”命令。此时，Visual Studio 会编译链接程序，如果有语法、编译和链接错误，会在错误列表窗口中出现提示信息，如图 1-5 所示，双击错误提示行信息，就可直接跳转到出现错误的代码行进行修改；如果没有错误，则会直接生成一个可执行文件。

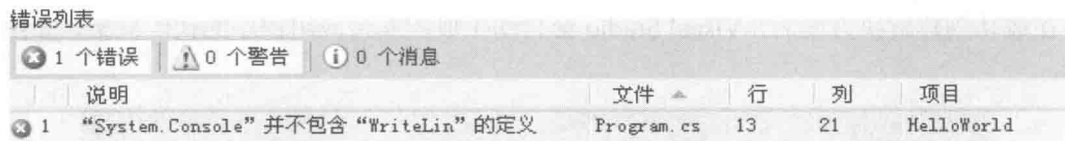


图 1-5 错误列表

运行程序有两种选择：调试执行和直接运行。

调试执行：按快捷键 F5，或者选择“调试”→“启动调试”命令，调试执行程序。

直接运行：按快捷键 Ctrl+F5，或者选择“调试”→“开始执行（不调试）”命令，直接运行程序。运行结果如图 1-6 所示。



图 1-6 HelloWorld 运行结果

1.3.2 C#程序结构分析

上面第一个 C#应用程序（HelloWorld）的 Demo 介绍了控制台应用程序的创建、编译和执行过程。接下来介绍应用程序的文件组成及其结构分析，这将有助于我们深入理解 C#应用

程序。

1. 应用程序的项目文件组成

首先，打开“HelloWorld”控制台应用程序的文件夹，其中有“HelloWorld”子文件夹和 HelloWorld.sln 和 HelloWorld.suo 两个文件，如图 1-7 所示。

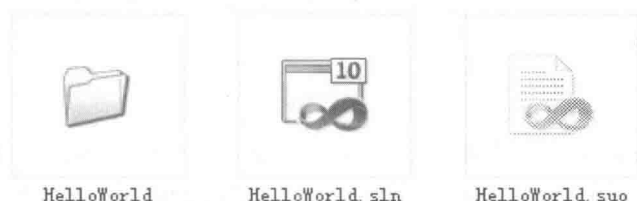


图 1-7 控制台应用程序的文件组成

HelloWorld 子文件夹中有 HelloWorld.csproj、program.cs 2 个文件和 bin、obj、properties 3 个文件夹。

(1) HelloWorld.csproj 包含项目的一些信息，如项目包含哪些文件，引用了哪些命名空间等，这些信息即是每次打开项目时读取的项目信息。

(2) program.cs 文件包含了 Visual C# 2010 最初创建的源代码，也称之为源文件。

(3) bin 文件夹存储编译好的文件和程序集。

(4) obj 文件夹存储临时文件和中间文件。

(5) properties 文件夹中有一个 AssemblyInfo.cs 文件，它包含程序集的描述信息和版本信息。

在后面的学习中，还会接触到窗体应用程序，它的文件组成和控制台应用程序相似，但也有区别，请大家在后续学习时注意观察并体会这些相似与区别。

2. C#程序的结构分析

回顾一下刚刚编写的第一个 C#程序，一个简单的 C#程序要涉及命名空间、类、Main() 方法、主程序等概念，通过分析 C#程序的结构，理解并掌握 C#语言的规范。下面是“HelloWorld”的完整程序。

```
using System;                                //导入 System 命名空间
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace HelloWorld                          //定义一个 HelloWorld 命名空间
{
    class Program                             //定义一个 Program 类
    {
        static void Main(string[] args)      //定义一个静态 Main()方法
        {
            //写入程序要执行的代码
            Console.WriteLine("Hello World!"); //输出语句“Hello World!”
            Console.ReadLine();               //等待用户输入，结束应用程序
        }
    }
}
```

```
}  
}  
}
```

(1) 命名空间

程序的第一条语句是“using System;”，功能是导入 System 命名空间。C#程序是用命名空间来组织代码的，要访问某个命名空间中的类或对象，必须要引用包含这些类或对象的命名空间，语法格式如下：

命名空间.类名

本程序中的 Console 类就包含在 System 命名空间中。所以，实际使用 Console 类时应该写 System.Console。但是，在程序的第一行有“using System;”这条导入语句，就可以直接使用 System 命名空间中的 Console 类了。

接下来，定义一个 HelloWorld 命名空间，即程序中的“namespace HelloWorld{ }”，用命名空间 HelloWorld 来组织控制台应用程序的 Program 类及其静态 Main()方法的代码。

(2) 类

```
class Program  
{  
...  
}
```

这是程序中定义的 Program 类，每个 C#语言编写的程序都是由多个类构成的。Class 是定义类的关键词，Program 是类的名称，这个类属于 HelloWorld 命名空间。此控制台应用程序的主要功能就是依靠 Program 类实现的。

(3) Main()方法

在 Program 类中，定义了静态方法 Main()，代码如下：

```
static void Main(string[] args)  
{  
...  
}
```

C#中规定，名称为 Main()的静态方法是程序的入口，而且 Main()方法必须包含在类中。方法的所有部分都包含在大括号中，即该方法所要执行的功能。

(4) 注释

注释是对程序中比较重要或需要注意的地方做的说明，但是这些说明不参与程序的编译执行。在编写程序的时候，要养成给出注释的好习惯，合理、详细的注释有利于代码后期的维护和阅读，使程序更清晰，有条理。

C#有 3 种注释方式。

① **单行注释：**单行注释符号是“//”，符号后面的文字是对语句、变量等的说明性文字，编译器在编译时会忽略双斜线后的文字。注意，“//”注释方法只对本行有效。

② **多行注释：**多行注释符号是以“/*”开始，以“*/”结束的，“/*”与“*/”所包含的内容作为注释内容，但是，它们必须成对使用。

③ **XML 文档说明：**实际上，前两种注释都是 C 语言的注释风格。C#中还提供了一个重