



高等学校计算机科学与技术教材

Visual C# 应用程序设计

COMPUTER Science and Technology

□ 叶 华 王婉茹 王作为 马 欣 编著

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精练，实例丰富
- 可操作性强，实用性突出



清华大学出版社 ● 北京交通大学出版社

高等学校计算机科学与技术教材

Visual C#应用程序设计

叶 华 王婉茹 王作为 马 欣 编著

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

本书详细介绍了 C#程序设计及其应用技术,内容涵盖了 C#语言程序设计基础、面向对象程序设计思想、Windows 应用程序设计、Web 应用程序设计、异常处理和程序调试等方面。本书在介绍程序设计方法时,使用针对性强的例子进行说明,实例代码短小精悍,具有一定的实用价值。第 2 章至第 8 章每章最后都综合利用本章关键知识点开发了一款实用软件供读者学习。书中的编程实例步骤详细,所有代码全部在 Visual Studio .NET 2012 开发环境中编译、运行通过。

本书可以作为高等学校计算机导论和 C 语言程序设计课程的后续课程,或作为使用 C#语言开发 Windows 应用程序和 Web 应用程序的培训教材,也适合使用 C#语言开发项目的程序员作为参考。

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 13501256678 13801310933

图书在版编目(CIP)数据

Visual C#应用程序设计 / 叶华等编著. —北京: 北京交通大学出版社: 清华大学出版社, 2014.1

(高等学校计算机科学与技术教材)

ISBN 978-7-5121-1757-0

I. ①V… II. ①叶… 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(2013)第 317494 号

责任编辑: 谭文芳 特邀编辑: 李晓敏

出版发行: 清华大学出版社 邮编: 100084 电话: 010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编: 100044 电话: 010-51686414 <http://www.bjtup.com.cn>

印刷者: 北京交大印刷厂

经 销: 全国新华书店

开 本: 185×260 印张: 17.25 字数: 442 千字

版 次: 2014 年 2 月第 1 版 2014 年 2 月第 1 次印刷

书 号: ISBN 978-7-5121-1757-0/TP·775

印 数: 1~3 000 册 定价: 34.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

C#程序设计语言是一种完全面向对象的基于.NET的编程语言,已被国际标准化组织批准为高级语言开发标准(ISO/IEC 23270)。随着.NET技术的普及,C#语言已成为开发基于.NET的企业级应用程序的首选语言。

本书是作者在讲授 Visual C# .NET 程序设计课程的讲义和教案的基础上,经过整理、编辑而成的。本书详细介绍了 C#程序设计及其应用技术,内容涵盖了 C#语言程序设计基础、面向对象程序设计思想、Windows 应用程序设计、Web 应用程序设计、异常处理和程序调试等方面。全书共 9 章。第 1 章介绍.NET 技术和 C#语言特点;第 2 章介绍 C#语言的基本知识;第 3 章讲解面向对象编程的思想;第 4 章介绍 Windows 窗体应用程序中的窗体和常用控件的设计;第 5 章介绍文件和流的编程技术;第 6 章介绍 GDI+编程;第 7 章介绍 ADO.NET 技术;第 8 章介绍使用 ASP.NET 技术设计简单的 Web 应用程序;第 9 章介绍异常处理和程序调试的方法。附录为实验指导。

本书讲解力求深入浅出,循序渐进,讲解过程中步骤详尽,让读者在阅读时一目了然。在介绍程序设计方法时,本书选择典型实例进行说明,实例代码短小精悍,具有一定的实用价值。部分代码可以直接应用到实际项目中。第 2 章至第 8 章每章的最后一节都综合利用本章关键知识点开发了一款实用性较强的实用软件供读者学习。书中的例子全部在 Visual Studio.NET 2012 开发环境中编译、运行通过。

本书由叶华、王琬茹、王作为、马欣编写。感谢天津工业大学的李兰友教授、徐国伟副教授对书稿提出了宝贵的意见和建议。参与本书编写、程序设计与调试、文字校对工作的还有刘伟、范博文、郑渊、房明磊、王颖、刘京京、庞剑、刘星、王超、王程皓、程启航等。由于时间仓促,加之水平有限,书中的缺点和不足之处在所难免,敬请读者批评指正。联系方式: yhua2005@sohu.com。

作 者
2013 年 10 月

目 录

第 1 章 概述	1
1.1 .NET 平台和.NET 框架	1
1.2 C#语言	2
1.3 浏览 Visual Studio.NET 2012	2
1.4 Visual Studio.NET 解决方案和项目文件的组织结构	4
习题	4
第 2 章 C#语言基本语法	5
2.1 控制台应用程序	5
2.1.1 创建控制台应用程序	5
2.1.2 控制台应用程序基本结构	7
2.2 数据类型与表达式	9
2.2.1 内置数据类型	9
2.2.2 变量和常量	12
2.2.3 运算符与表达式	13
2.3 流程控制语句	16
2.3.1 顺序结构	16
2.3.2 选择结构	16
2.3.3 循环结构	19
2.3.4 跳转语句	20
2.4 数组	21
2.4.1 数组的定义与初始化	21
2.4.2 数组应用	22
2.5 综合实例之计算缴税金额程序	24
2.5.1 计算缴税金额	24
2.5.2 为多名职员计算缴税金额	25
习题	27
第 3 章 面向对象的编程基础	28
3.1 定义类	28
3.1.1 类和对象	28
3.1.2 定义类	28
3.1.3 创建对象	30
3.1.4 方法	31
3.1.5 属性	35

3.2	封装和数据隐藏	36
3.3	继承性	37
3.4	多态性	40
3.5	综合实例之获取商品信息	44
3.5.1	定义商品类 Product	44
3.5.2	测试商品类 Product	46
	习题	47
第4章	窗体和常用控件编程	48
4.1	初识 Windows 窗体	48
4.2	常用控件设计	50
4.2.1	标签控件 (Label)	51
4.2.2	文本框控件 (TextBox)	51
4.2.3	下压按钮控件 (Button)	52
4.2.4	组框控件 (GroupBox) 和面板控件 (Panel)	54
4.2.5	单选按钮控件 (CheckBox) 和复选框控件 (RadioButton)	55
4.2.6	列表框控件 (ListBox)	59
4.2.7	组合框控件 (ComboBox)	63
4.2.8	菜单	65
4.2.9	工具栏	68
4.2.10	状态栏	69
4.2.11	对话框	70
4.2.12	通用对话框	71
4.3	综合实例之仿 QQ 登录界面的设计	74
4.3.1	创建登录窗体	74
4.3.2	为登录窗体添加控件	75
4.3.3	创建“网络设置”对话框	77
4.3.4	为“网络设置”对话框添加控件	78
4.3.5	为“网络设置”对话框的控件编写事件处理函数	79
4.3.6	登录窗体调用“网络设置”对话框	81
	习题	81
第5章	文件和流	83
5.1	文件和流概述	83
5.2	File 类和 FileInfo 类	83
5.3	Directory 类和 DirectoryInfo 类	87
5.4	FileStream 类	92
5.4.1	FileStream 对象实例化	92
5.4.2	读取数据和写入数据	92
5.5	StreamWriter 类和 StreamReader 类	94
5.5.1	StreamWriter 类	94

5.5.2	StreamReader 类	96
5.6	BinaryWriter 类和 BinaryReader 类	100
5.6.1	BinaryWriter 类	100
5.6.2	BinaryReader 类	103
5.7	综合案例之文件分割与文件合并工具软件	105
5.7.1	TabControl 控件、ProgressBar 控件和 FolderBrowserDialog 对话框	105
5.7.2	字符串处理	106
5.7.3	文件分割合并工具软件界面设计	108
5.7.4	文件分割功能设计	111
5.7.5	文件合并功能设计	116
	习题	120
第 6 章	GDI+程序设计	121
6.1	GDI+概述	121
6.2	GDI+中常用结构	122
6.3	Graphics 类	122
6.4	绘制图形	123
6.4.1	绘图步骤	123
6.4.2	Paint 事件	123
6.4.3	Pen 类	124
6.4.4	Brush 类	126
6.4.5	基本图形的绘制和填充	128
6.5	图像处理	131
6.5.1	显示图像	131
6.5.2	保存图像	133
6.5.3	图像处理	133
6.6	文字处理	136
6.6.1	创建字体	136
6.6.2	绘制文本	136
6.7	综合实例之画板	137
6.7.1	画板程序界面设计	138
6.7.2	实现文件新建、打开及保存功能	140
6.7.3	绘图参数的读取	143
6.7.4	拖动鼠标绘图的实现	146
6.7.5	图像缩放处理的实现	150
	习题	152
第 7 章	ADO.NET 数据库访问技术	153
7.1	数据库基础	153
7.1.1	关系数据库	153
7.1.2	使用 SQL Server Express 创建数据库	154

7.1.3	结构化查询语言	155
7.2	ADO.NET 架构	159
7.2.1	ADO.NET	159
7.2.2	连接方式和非连接方式	159
7.2.3	ADO.NET 对象模型	160
7.2.4	.NET 框架数据提供程序	161
7.2.5	DataSet	170
7.3	综合实例之作者信息管理系统	184
7.3.1	设计数据库	184
7.3.2	作者信息管理系统界面设计	185
7.3.3	作者信息管理系统功能设计	187
	习题	199
第 8 章	Web 应用程序设计	200
8.1	Web 应用程序设计基础知识	200
8.1.1	静态网页与动态网页	200
8.1.2	ASP.NET 简介	201
8.1.3	使用 Visual Studio.NET 开发简单的网站并测试	201
8.1.4	Web 应用程序开发环境搭建	204
8.2	Web 服务器控件之标准控件	209
8.2.1	简单的 Web 原理	209
8.2.2	Web Form 服务器控件简介	210
8.2.3	Web Form 服务器控件示例	210
8.3	ASP.NET 中的对象	212
8.3.1	Page 类	213
8.3.2	ASP.NET 核心对象	214
8.3.3	状态管理	219
8.4	综合实例之用户登录模块设计	227
8.4.1	设计数据库	227
8.4.2	设计登录页面	227
8.4.3	登录功能的实现	229
	习题	232
第 9 章	异常处理与程序调试	233
9.1	异常处理	233
9.1.1	Exception 类	233
9.1.2	异常处理语句	234
9.1.3	自定义异常类	240
9.2	程序调试	242
9.2.1	程序错误类型	243
9.2.2	程序调试	244

习题..... 257

附录 A 实验指导 258

 实验 1 C#编程基础实验 258

 实验 2 面向对象程序设计实验 258

 实验 3 窗体和常用控件设计 259

 实验 4 文件和流编程 261

 实验 5 GDI+图形图像编程..... 262

 实验 6 ADO.NET 编程 263

 实验 7 Web 应用程序设计 263

 实验 8 异常处理与程序调试 264

参考文献..... 265

第 1 章 概 述

1.1 .NET 平台和.NET 框架

微软公司表示，Microsoft.NET 平台代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程框架。它是一种进行各种应用程序开发的集成开发和执行环境。.NET 平台提供了多种核心的技术和服务，运用这些技术和服务能够简化应用程序开发。

.NET 框架的主要组件包括.NET 框架类库和公共语言运行库（Common Language Runtime，CLR）。

.NET 框架类库提供了一个可以由所有 Web 应用程序、Windows 应用程序及 XML Web Service 应用程序访问的类库。.NET 框架不仅仅是一种类库集。已经有多种编程语言的编译器使用了.NET 框架，运用这些编程语言可以创建各种各样的应用程序，.NET 框架为这些应用程序提供一个可共享的基本平台，监视应用程序的执行，使得用不同编程语言所编写的软件可以安全地使用类和其他数据类型。

公共语言运行库（CLR）负责管理和执行.NET 框架类库开发的所有应用程序。在 Visual Studio.NET 中集成的所有编程语言编写的应用程序均需通过公共语言运行库才能运行。

Visual Studio.NET 在编译应用程序时，不是立即创建操作系统特定的本机代码，而是把代码编译为 Microsoft 中间语言代码（Microsoft Intermediate Language，MSIL），这些代码不专用于任何一种操作系统。即时（Just In Time，JIT）编译器将 MSIL 编译为专用于操作系统和目标机器结构的本机代码。

Visual Studio.NET 2012 是微软公司于 2012 年正式发布的最新一代 Visual Studio.NET 开发平台。Visual Studio.NET 2012 开发平台提供了针对 Windows 应用、Web 应用、WPF（Windows Presentation Foundation）应用、Office 应用、Silverlight 应用等很多应用程序模板。表 1-1 介绍了一些主要模板的作用。

表 1-1 Visual Studio.NET 2012 开发平台提供的若干应用程序模板

模 板	描 述
控制台应用程序	控制台应用程序使用统一的重定向文本输出形式，在命令行方式下运行，用于交互性操作不多、偏重于内部功能实现的场合
Windows 窗体应用程序	Windows 窗体应用程序主要用于创建具有 Windows 窗体用户界面，且交互性操作较多的项目，如网络游戏、办公软件等
Windows 窗体控件库	Windows 窗体控件库用来开发 Windows 应用程序的用户界面控件，这些控件具有可重用性
WPF 应用程序	WPF 客户端应用程序是基于 Windows 的图形界面处理。它提供了统一的编程模型、语言和框架，真正做到了分离界面设计人员与开发人员的工作；同时它提供了全新的多媒体交互用户图形界面
ASP.NET Web 窗体应用程序	ASP.NET 应用程序为基于 Web 的应用程序
ASP.NET Web 服务器控件	ASP.NET Web 服务器控件模板用于开发 Web 应用程序的用户界面控件，这些控件具有可重用性

1.2 C#语言

C#是可用于创建运行在.NET CLR 上的应用程序语言之一，是 Microsoft 专门为使用.NET 平台而创建的，能使用.NET 框架类库提供的每种功能。

C#语言是一种现代的、面向对象的语言，它简化了 C++语言在类、命名空间、方法重载和异常处理等方面的操作，它摒弃了 C++的复杂性，更易使用，更少出错。

1.3 浏览 Visual Studio.NET 2012

Visual Studio.NET 2012 开发环境拥有大量的窗口、工具栏和工具箱窗口。默认的 Visual Studio.NET 2012 开发环境如图 1-1 所示。

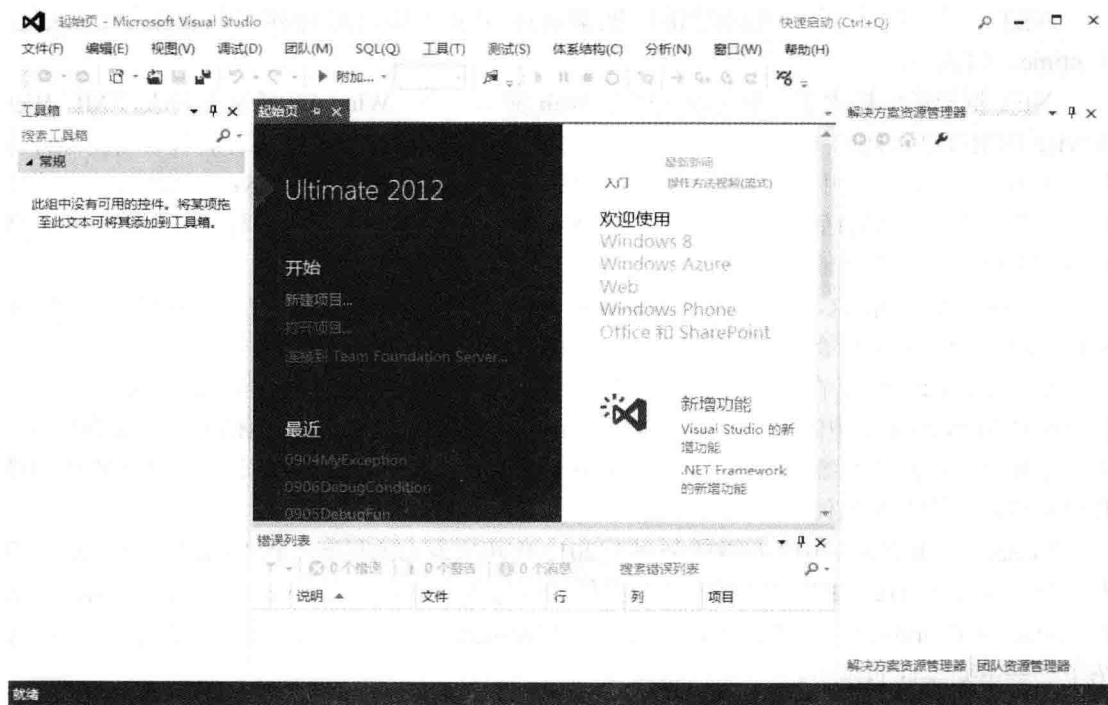


图 1-1 Visual Studio.NET 2012 初始界面

1. “解决方案资源管理器”窗口

“解决方案资源管理器”窗口如图 1-2 所示。该窗口显示了当前 Visual Stdio.NET 解决方案的树型结构。通过“解决方案资源管理器”窗口，开发人员既可以浏览组成当前解决方案的所有项目，又可以浏览属于每个项目的文件。双击某个项目文件可以将该文件打开，然后可以对该文件进行编辑。打开一个文件后，菜单和工具栏的可用项将随之改变。

2. “类视图”窗口

“类视图”窗口如图 1-3 所示。该窗口显示项目的类结构。



图 1-2 “解决方案资源管理器”窗口



图 1-3 “类视图”窗口

3. “资源视图”窗口

“资源视图”窗口用于查看与项目相关的资源文件。正常情况下，只有当项目存在相关资源（.rc）文件时，“资源视图”窗口才会显示。

4. “属性”窗口

“属性”窗口如图 1-4 所示。该窗口用于设置解决方案中不同子项的属性。该窗口的内容由当前所设置的子项类型来决定。

5. “错误列表”窗口

“错误列表”窗口如图 1-5 所示。该窗口显示程序的错误信息。



图 1-4 “属性”窗口



图 1-5 “错误列表”窗口

6. “服务器资源管理器”窗口

“服务器资源管理器”窗口如图 1-6 所示。该窗口提供访问本地或远程计算机上可用系统服务的功能。

7. “工具箱”窗口

“工具箱”窗口如图 1-7 所示。该窗口按照控件的不同类型放置在“服务器资源管理器”窗口的不同选项卡中。

Visual Studio.NET 2012 还包括很多非常有用的窗口，如对象浏览器窗口、调用堆栈窗口、输出窗口、调试窗口等。若要显示这些窗口，可以选择主菜单“视图”中的相应子菜单。

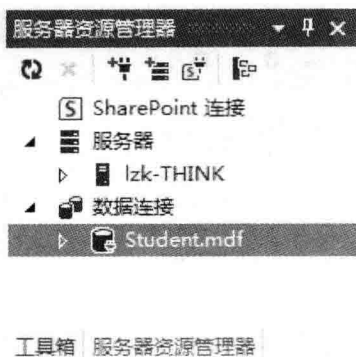


图 1-6 “服务器资源管理器”窗口

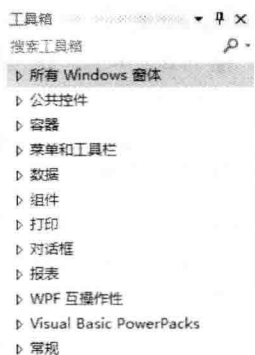


图 1-7 “工具箱”窗口

1.4 Visual Studio.NET 解决方案和项目文件的组织结构

在使用 Visual Studio.NET 开发应用程序时，可用通过创建解决方案来完成。解决方案是把多个 Visual Studio.NET 项目组织在一起的项目集合。项目是解决方案中的一个容器，用户可以通过它可以对其组成部分进行编译、调试，以及进行逻辑上的管理。

解决方案允许在开发环境的同一实例上管理多个项目，允许同时处理一组项目的子项，允许统一管理解决方案或项目的其他各种外部文件。

Visual Studio.NET 支持多种文件类型。Visual Studio.NET 应用程序的常用文件类型如表 1-2 所示。

表 1-2 Visual Stdio.NET 应用程序的常用文件类型

名 称	扩 展 名	说 明
Visual Studio 解决方案文件	.sln	.sln 文件通过提供解决方案的项目、项目子项及其他子项位置的环境引用，实现对它们的组织管理
C#项目文件	.cs	.cs 文件表示项目的窗体文件、用户控件文件、类文件和模块文件等
C#项目文件	.csproj	.csproj 文件表示属于多项目的窗体文件、用户控件文件、类文件和模块文件。借助该扩展名，可以把 C#程序文件和其他.NET 兼容语言的程序文件区别开
Web 项目文件	.aspx .asmx 等	.aspx 是 Web 窗体文件；.asmx 是 XML Web Service 文件；.Web 项目也使用.cs 文件扩展名来表示类文件和模块文件
资源文件	.resx	.resx 文件是 XML 格式的资源文件。

习题

1. 什么是.NET 平台？什么是.NET 框架？
2. Visual Studio.NET 2012 开发平台提供了哪些应用程序模板？
3. 什么是解决方案？什么是项目？
4. 扩展名为.sln 的文件是什么类型的文件？
5. 扩展名为.cs 和.csproj 的文件是什么类型的文件？

第 2 章 C#语言基本语法

2.1 控制台应用程序

本书第 2 章至第 3 章将频繁使用控制台应用程序，本节将首先介绍如何创建一个控制台应用程序。

2.1.1 创建控制台应用程序

【例 2-1】 创建一个简单的控制台应用程序，在控制台显示“Hello World!”。

本例题的设计、操作步骤如下：

① 打开 Visual Studio.NET 2012，选择“文件”→“新建”→“项目”菜单命令，弹出如图 2-1 所示的“新建项目”对话框。在对话框的“项目类型”中选择“Windows”，在“模板”中选择“控制台应用程序”，在“位置”中输入项目所在路径，在“名称”中输入项目名称，本例题项目名称为“0201Hello”。设置完毕上述内容后，单击“确定”按钮，弹出如图 2-2 所示的界面。



图 2-1 “新建项目”对话框

② 在 Program.cs 文件的 Main 函数中输入如下代码。

```
static void Main(string[] args)
{
    Console.WriteLine("Hello World!");
    Console.ReadKey();
}
```



图 2-2 创建控制台应用程序后的界面

③ 选择“调试”→“启动调试”菜单命令，得到如图 2-3 所示的运行结果。

④ 按下任意键，退出应用程序。

注意，控制台窗口在应用程序执行完毕后立即关闭，这样就无法看到运行的结果。为了解决这个问题，本例使用 `Console.ReadKey();` 语句告诉代码在结束前等待按键。

下面介绍开发环境中的各个组成部分，例如“解决方案资源管理器”窗口、“类视图”窗口、“属性”窗口和“错误列表”窗口。

1. “解决方案资源管理器”窗口

“解决方案资源管理器”窗口显示了组成 0201Hello 项目的文件，如图 2-4 所示。项目文件包括代码文件 `Program.cs` 和 `AssemblyInfo.cs` 等文件。所有的 C# 文件使用 `.cs` 扩展名。

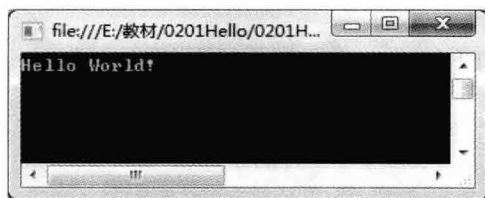


图 2-3 例 2-1 运行结果



图 2-4 “解决方案资源管理器”窗口

使用“解决方案资源管理器”窗口可以双击.cs 文件以编写代码，或者对文件进行重命名、删除等操作。

2. “类视图”窗口

“类视图”窗口显示项目的类结构，如图 2-5 所示。

3. “属性”窗口

“属性”窗口显示“解决方案资源管理器”窗口中所选项目的其他信息，如图 2-6 所示。



图 2-5 “类视图”窗口



图 2-6 “属性”窗口

4. “错误列表”窗口

“错误列表”窗口显示程序的错误信息，如图 2-7 所示。如果双击该窗口中显示的错误信息，光标会跳到源代码中出现错误的地方。

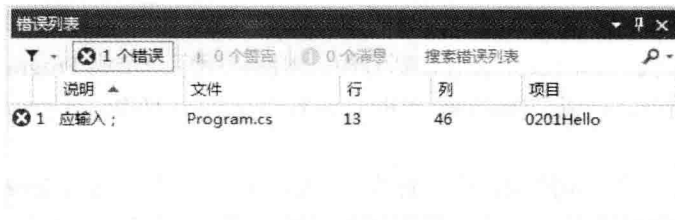


图 2-7 “错误列表”窗口

2.1.2 控制台应用程序基本结构

例 2-1 创建的项目 0201Hello 中的文件 Program.cs 的具体代码如下所示：

```
//Program.cs, A first program in C#.
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace 0201Hello
{
    class Program
    {
        static void Main(string[] args)
```



```
    {  
        Console.WriteLine("Hello World!");  
        Console.ReadLine();  
    }  
}  
}
```

现对上述代码中的元素和组织原则作简要介绍。

1. using 关键字

using 关键字可以引用 Microsoft .NET 框架类库中的现有资源。通常，在程序文件的开头使用这个关键字。通过多次使用该关键字，可以为程序引用多种资源。

2. System 命名空间

System 命名空间提供了对构建应用程序所需的所有系统功能的访问。标准的 System 命名空间包含了最常用的 .NET 类型。本例使用了 System 命名空间中的 Console 类，以写入控制台窗口。

3. namespace 关键字

namespace 关键字声明了应与类相关的命名空间，其后花括号中的所有代码都被认为是在这个命名空间中。编译器在 using 指令指定的命名空间中查找没有在当前命名空间中定义、但在代码中引用类。

4. 类

所有的 C# 代码都必须包含中类中。类的声明包括 class 关键字，其后是类名和一对花括号。与类相关的所有代码都应放在这对花括号中。本例声明了一个类 Program，因为该类位于 0201 Hello 命名空间中，所以，Program 类的完整名称是 0201 Hello.Program。

5. Main 方法

每个 C# 可执行文件，如控制台应用程序、Windows 应用程序或 Windows 服务，都必须有一个入口点，即 Main 方法。这个方法在程序启动时调用。本例中，Main 方法有两个修饰符 public 和 static。修饰符 public 表示可以在任何地方访问该方法，所以可以在类的外部调用。修饰符 static 表示方法不能在类的特定实例上执行，不必先实例化类再调用。每个 C# 应用程序都必须在组成程序的某一个类中包含 Main 方法。

6. 语句

C# 语言要求每个语句都必须用一个分号结尾。语句可以写在多个代码行上。

7. 注释

C# 语言使用 // 符号进行单行注释；多行注释以一个 “/*” 开头，以一个 “*/” 结尾。

本例的主要代码语句如下：

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

本例调用了 System.Console 类的 WriteLine() 方法，把 "Hello World!" 这个字符串写到控制台窗口上。WriteLine() 方法是一个静态方法，在调用之前不需要实例化 Console 对象。

Console.ReadLine() 读取用户的输入，该行代码让应用程序等待用户按下回车键，之后退出应用程序。在 Visual Studio 2012 中，控制台窗口会消失。