



“十三五”普通高等教育应用型规划教材

C# .NET CHENGXU SHEJI JIAOCHENG
—— JIYU ANLI QUDONG

C# .NET 程序设计教程

—— 基于案例驱动

主 编 赵佳华 杨 迎



北京邮电大学出版社
www.buptpress.com



“十三五”普通高等教育应用型规划教材

C#.NET 程序设计教程——基于案例驱动

主 编 赵佳华 杨 迎
副主编 王春明 付 杰
马宪敏 卢芳葳
主 审 周洪玉

北京邮电大学出版社

· 北京 ·

内 容 简 介

本书结合 Visual Studio 2010 开发环境介绍基于 C#.NET 项目开发的相关知识,全书包括 C# 的开发环境介绍;C# 中的变量、常量,如何使用数组及强大的 String 类,掌握不同数据类型之间的数据转换;C# 程序的流程控制语句、语句语法格式;面向对象程序设计中的基本知识,以及面向对象程序设计中的静态类、静态成员、继承性、多态性等相关知识;快速设计 Windows 应用程序的用户界面;ADO.NET 技术,主要涉及 ADO.NET 提供用来访问和处理数据的两个组件:.NET Framework 数据提供程序和 DataSet 数据集。本书以案例为基础进行理论知识的教授,通过一个个小的项目案例逐步引入相关理论知识,再通过解决项目过程中出现的问题对理论知识进行深入研究。在案例驱动下进行教学可以增强读者的求知欲,使其快速掌握 C# 的编程技术,理解更容易,上手更快。本书的第 8 章还提供了一个完整的应用案例——民航订票系统,以便读者能从整体上理解使用 C# 程序设计语言开发 Windows 应用系统。

本书既可作为高等学校计算机及相关专业的 C# 程序设计课程的教材与指导书,同时也适合作为广大学员开发人员的自学参考书。

图书在版编目(CIP)数据

C#.NET 程序设计教程:基于案例驱动/赵佳华,杨迎主编.——北京:北京邮电大学出版社,2016.9

ISBN 978-7-5635-4779-1

I. ①C… II. ①赵… ②杨… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 124868 号

书 名	C#.NET 程序设计教程——基于案例驱动
主 编	赵佳华 杨 迎
责任编辑	向 蕾
出版发行	北京邮电大学出版社
社 址	北京市海淀区西土城路 10 号(100876)
电话传真	010-82333010 62282185(发行部) 010-82333009 62283578(传真)
网 址	www.buptpress3.com
电子信箱	ctrd@buptpress.com
经 销	各地新华书店
印 刷	北京泽宇印刷有限公司
开 本	787 mm×1 092 mm 1/16
印 张	15.5
字 数	387 千字
版 次	2016 年 9 月第 1 版 2016 年 9 月第 1 次印刷

ISBN 978-7-5635-4779-1

定价: 35.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

前 言

C#是微软公司发布的一种面向对象的、运行于.NET Framework之上的高级程序设计语言。C#看起来与Java有着惊人的相似,它包括了诸如单一继承、接口及与Java几乎同样的语法和编译成中间代码再运行的过程。但是C#与Java有着明显的不同,它借鉴了Delphi的一个特点,与COM(组件对象模型)是直接集成的,而且它是微软公司.NET Windows网络框架的主角。C#吸收了C++,Visual Basic,Delphi,Java等语言的优点,体现了当今最新的程序设计技术的功能和精华。C#继承了C语言的语法风格,同时又继承了C++的面向对象特性。不同的是,C#的对象模型已经面向Internet进行了重新设计,使用的是.NET框架的类库;C#不再提供对指针类型的支持,使得程序不能随便访问内存地址空间,从而更加健壮;C#不再支持多重继承,避免了以往类层次结构中由于多重继承带来的可怕后果。.NET框架为C#提供了一个强大的、易用的、逻辑结构一致的程序设计环境。同时,公共语言运行时(common language runtime)为C#提供了一个托管的运行环境,使程序比以往更加稳定、安全。

本书是面向高等学校相关专业C#程序设计的教材,也可以用于自学C#程序设计的使用,重在培养读者利用面向对象的技术分析和解决实际问题的能力,指导读者在较短的时间内学会利用最先进的Visual Studio工具开发WinForm窗体。本书最大的特色在于以案例为基础进行理论知识的教授,本书通过一个个小的项目案例逐步引入相关理论知识,再通过解决项目过程中出现的问题对理论知识进行深入研究。在案例驱动下进行教学可以增强读者的求知欲,使其快速掌握C#的编程技术,理解更容易,上手更快。本书的第8章还提供了一个完整的应用案例,以便读者能从整体上理解WinForm应用程序的开发。

本书共分为9章,第1章将使学生对.NET和C#有一个初步的认识,了解C#的开发环境。通过动手编写第一个C#控制台程序,使学生了解C#程序的组成和特点。第2章将学习C#中的变量、常量、数组及强大的String类,掌握不同数据类型之间的数据转换。第3章将讲解C#程序的流程控制语句,学习条件判断、循环结构等程序逻辑处理语句,以及在C#中的语句语法格式。第4章将对面向对象程序设计中的基本知识进行详细讲解。第5章将继续学习面向对象的静态类、静态成员、继承性、多态性等相关知识,对面向对象有一个更深入的了解。第6章开始全新的Windows应用程序的开发。.NET Framework为开发Windows应用程序提供了完整的框架,借助于.NET Framework,开发人员可以快速设计Windows应用程序的用户界面。第7章将学习怎样在应用程序中操作数据库中的数据,也就是.NET Framework的另一个重要技术——ADO.NET技术,主要涉及ADO.NET提供用来访问和



处理数据的两个组件: .NET Framework 数据提供程序和 DataSet 数据集。第 8 章是对航空订票系统项目实战功能的进一步完善。第 9 章是各章习题与上机实战的参考答案。

为了方便教师使用本书进行教学,减轻教师备课负担,提高授课质量,本书为教师免费提供全部实例程序源代码。

本书由赵佳华、杨迎担任主编,王春明、付杰、马宪敏、卢芳葳担任副主编,主要分工如下:第 1、第 9 章由付杰编写,第 2、第 4 章由杨迎编写,第 3、第 7 章由赵佳华编写,第 5、第 6、第 8 章由王春明编写,各章习题与上机实战由马宪敏、卢芳葳编写。全书由马宪敏统稿,由周洪玉教授进行主审和制定修改方案,并进行了全书审校。

本书的编写得到了各编者单位有关领导的大力支持,在此深表谢意。

由于时间仓促,编者学识有限,书中难免会存在疏漏和不妥之处,恳请广大读者提出宝贵的意见和建议,以便修订时加以完善。

编者

2016 年 6 月

目 录

第 1 章 C# 概述	1
1.1 .NET 与 C# 概述	1
1.1.1 .NET 的概述	1
1.1.2 C# 的发展和特点	2
1.2 C# 程序的开发环境	3
1.2.1 IDE 概述	3
1.2.2 Visual Studio.NET 窗口结构简介	3
1.3 第一个 C# 应用程序	4
1.3.1 C# 程序案例	4
1.3.2 C# 程序的特点	6
1.3.3 控制台输入和输出	9
1.4 本书课程案例介绍	11
习题 1	13
上机实战 1	13
第 2 章 C# 程序设计基础	14
2.1 常量与变量	14
2.1.1 常量	14
2.1.2 变量	15
2.2 运算符和表达式	16
2.2.1 赋值运算符	17
2.2.2 算术运算符	18
2.2.3 自增和自减运算	18
2.2.4 关系运算符	19
2.2.5 逻辑运算符	19
2.2.6 条件运算符	20
2.3 C# 的数据类型	20
2.3.1 数据类型简介	20
2.3.2 数据类型转换	21
2.3.3 枚举类型和结构类型	25

2.4 数组和字符串	28
2.4.1 一维数组	29
2.4.2 多维数组	33
2.4.3 字符串 String	36
习题 2	45
上机实战 2	46
第 3 章 C# 程序的流程控制	47
3.1 C# 中的条件判断	47
3.1.1 if 语句	47
3.1.2 switch 语句	55
3.2 C# 程序的循环语句	58
3.2.1 while 语句	58
3.2.2 do...while 语句	61
3.2.3 for 语句	64
3.2.4 foreach 语句	69
3.2.5 循环嵌套语句	71
3.3 流程控制进阶	74
3.3.1 break 语句	74
3.3.2 continue 语句	75
习题 3	76
上机实战 3	77
第 4 章 面向对象程序设计入门	78
4.1 面向对象的基本概念	78
4.2 类与对象	79
4.2.1 类的声明	80
4.2.2 对象	81
4.2.3 类的成员	81
4.3 字段和属性	82
4.3.1 字段	82
4.3.2 静态字段	83
4.3.3 属性	84
4.4 类的方法	86
4.4.1 方法的声明	86
4.4.2 方法的调用	88
4.4.3 方法的参数传递	89
4.4.4 方法的重载	95



4.4.5 构造函数和析构函数	96
习题 4	98
上机实战 4	98
第 5 章 面向对象的高级程序设计	100
5.1 静态成员与静态类	100
5.1.1 静态成员	100
5.1.2 静态类	102
5.2 类的继承性与多态性	102
5.2.1 类的继承性	102
5.2.2 类的多态性	106
5.3 接口	110
5.4 使用集合组织相关数据	115
5.4.1 ArrayList	115
5.4.2 泛型	118
习题 5	120
上机实战 5	121
第 6 章 Windows 应用程序的界面设计	122
6.1 Windows 窗体应用程序	122
6.1.1 创建 Windows 窗体应用程序	122
6.1.2 认识 Windows 应用程序	124
6.1.3 基本控件	128
6.2 事件驱动编程	132
6.2.1 编写事件处理程序	132
6.2.2 事件函数	133
6.3 消息框	134
6.4 控件进阶	135
6.4.1 选项类控件	136
6.4.2 列表选择控件	141
6.5 菜单栏、工具栏和状态栏	149
6.5.1 菜单栏	149
6.5.2 工具栏	155
6.5.3 状态栏	156
6.6 窗体跳转和 MDI 应用程序	160
6.6.1 窗体间的跳转	160
6.6.2 创建 MDI 应用程序	163
习题 6	166

上机实战 6	166
第 7 章 数据库编程技术	169
7.1 ADO.NET 概述	169
7.1.1 ADO.NET 简介	169
7.1.2 ADO.NET 的主要组件	170
7.2 Connection 对象	171
7.2.1 Connection 对象概述	171
7.2.2 连接数据库	171
7.2.3 异常处理	173
7.3 Command 对象	174
7.3.1 Command 对象概述	175
7.3.2 Command 对象的常用方法	176
7.4 DataSet 数据集	185
7.4.1 DataSet 对象	185
7.4.2 DataAdapter 对象	186
7.4.3 数据绑定控件	188
习题 7	196
上机实战 7	196
第 8 章 项目实战——民航订票系统	198
8.1 航班查询及预订窗体	198
8.2 实现航班机票信息查询	201
8.3 实现航班机票预订功能	205
第 9 章 参考答案	210
参考文献	240

第 1 章

C# 概 述

本章学习目标

- ◇ 了解 .NET 和 C# 的特点。
- ◇ 掌握 C# 的开发环境。
- ◇ 熟练掌握简单 C# 程序的开发步骤。
- ◇ 熟练掌握 C# 中控制台输入和输出。

本章将使学生对 .NET 和 C# 有一个初步的认识,了解 C# 的开发环境。通过动手编写第一个 C# 控制台程序,使学生了解 C# 程序的组成和特点。

1.1 .NET 与 C# 概述

1.1.1 .NET 的概述

.NET 是微软公司自从创立以来进行的最大的一次改变,它是一整套平台,内容几乎涵盖程序开发的任何方面,使用微软技术建立的软件迟早有一天都要转化到 .NET 平台上来。在发行之初,.NET 曾经被称为一次“豪赌”,而现在看来,微软在这次“豪赌”中又下对了赌注。那么,.NET 究竟是什么呢?

.NET 是 Microsoft .NET 的简称,是基于 Windows 平台的一种技术,它包含了能在 .NET Framework 平台上运行的所有编程语言。.NET Framework 是微软为开发应用程序创建的一个富有革命性的新平台,在这个平台中,可以开发出运行在 Windows 上的几乎所有的应用程序。简单地说,.NET Framework 是一个创建、部署、运行应用程序的多语言多平台环境,它包含了一个庞大的代码库,各种能够运行在 .NET Framework 平台上的语言都可以共享这些代码库。.NET 的图标如图 1-1 所示。



图 1-1 .NET 的图标



1.1.2 C#的发展和特点

2000年,配合.NET平台的发布,微软公司发布了一门新语言——C#(读作C sharp),它是由微软公司的Anders Hejlsberg(安德斯·海尔斯伯格)领导的开发小组专门为.NET平台设计的语言。Anders Hejlsberg(见图1-2)是一位颇具传奇色彩的人物,他早年曾在著名的Borland公司工作,是Pascal和Delphi语言的缔造者。1996年,Anders离开Borland加盟微软,并在之后创造了C#。



图1-2 Anders Hejlsberg

C#是一种全新的语言,在设计它的时候,微软博采众长,将C,C++,Java等多种开发语言的优点集于一身。C#是专为.NET开发而设计的编程语言,是微软为奠定其下一代互联网霸主地位而打造的在Microsoft.NET平台上运行的主流语言。C#一经推出便以其强大的操作能力,优雅的语法风格,创新的语言特性,第一流的面向组件编程的支持而深受世界各地程序员的好评和喜爱。“它就是我梦寐以求的计算机语言!”——很多资深程序员拿到C#都是这样的惊喜。从C#的名字(C sharp,sharp有锐利、锋利的意思),我们也可见微软用其打造下一代互联网深度服务的勃勃雄心。

C#可以说是微软在不断地发展和实践过程中,锤炼出来的一个精品。它集众家之长,不仅安全,而且还非常地易于使用。C#主要具有以下特点。

①C#拥有C和C++的强大功能及Visual Basic简易使用的特性,而且看起来与Java有着惊人的相似,不仅安全,而且易于掌握和使用。

②C#提出了纯粹的面向对象的概念,并在语言特性中展现得淋漓尽致,C#还通过引用类型、值类型和接口类型的概念形成了统一的类型系统。

③强大的类库支持。C#有数量庞大、功能齐全类库支持,从而可以轻易地完成复杂的加密操作、网络操作等。使用C#可以轻松地构建功能强大、开发快捷、运用方便的应用程序。

④C#可以提供脚本语言所无法提供的强大功能,使用C#编写ASP.NET应用程序将成为Web开发的最佳选择。

⑤C#包含的内置特性,使任何组件都可以轻松转化为XML(extensible markup language,可扩展标记语言)网络服务,通过Internet被任何操作系统上运行的任何程序调用。

除了这些特点,C#也支持.NET平台本身具有的垃圾回收、语言自由、跨平台等特性。

1.2 C# 程序的开发环境

1.2.1 IDE 概述

IDE,即 integrated development environment,是“集成开发环境”的英文缩写。

IDE 是用于提供程序开发环境的应用程序,一般包括代码编辑器、编译器、调试器和图形用户界面工具。它是集成了代码编写功能、分析功能、编译功能、debug 功能等一体化的开发软件套。所有具备这一特性的软件或者软件套(组)都可以叫作 IDE。IDE 可以独立运行,也可以和其他程序并用。IDE 为用户使用 Visual Basic,Java 和 C# 等现代编程语言提供了方便。

IDE 多被用于开发 HTML 应用软件。例如,许多人在设计网站时使用 IDE(如 HomeSite,Dreamweaver,FrontPage 等),因为很多项任务会自动生成。IDE 集成代码编辑、代码生成、界面设计、调试、编译等功能,目前还融合了建模功能。

1.2.2 Visual Studio.NET 窗口结构简介

微软提供了强大的平台和技术,并提供了强大的 IDE 来支持,这就是 Visual Studio。Visual Studio 集成了 .NET Framework,是目前最流行的 .NET 平台开发环境。Visual Studio 的版本发展如表 1-1 所示。

表 1-1 Visual Studio 的版本发展

年代	1998	2002	2003	2005	2008	2010
开发工具版本	Visual Studio 6.0	Visual Studio	Visual Studio 2003	Visual Studio 2005	Visual Studio 2008	Visual Studio 2010
支持的 .NET Framework 版本	不支持	1.0	1.1	2.0	3.5	4.0

随着 .NET Framework 的不断升级,微软也在不断升级 Visual Studio。本书使用的是 Visual Studio 2010,它支持的是 .NET Framework 4.0 并向下兼容 .NET Framework 3.5 和 .NET Framework 2.0。

Visual Studio 2010 最简单的窗口结构如图 1-3 所示,随着学习的深入,我们会陆续认识 Visual Studio 的其他窗口。

- 解决方案资源管理器:Visual Studio 提供的有效管理应用程序的容器。



- 错误列表:用来显示程序编译错误、警告信息。



错误列表

图 1-3 Visual Studio 2010 最简单的窗口结构

利用 Visual Studio,我们可以创建 Windows 平台下的 Windows 应用程序和网络应用程序,也可以创建网络服务、智能设备应用程序和 Office 插件等。

1.3 第一个 C# 应用程序

1.3.1 C# 程序案例

对 .NET Framework,C# 和 Visual Studio 有了初步认识后,下面就让我们来看一看 C# 程序的庐山真面目吧。

首先,启动 Visual Studio 2010,选择“开始”→“程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”选项,如图 1-4 所示。

使用 Visual Studio 2010 创建并运行控制台应用程序包括 3 步:新建项目→生成可执行文件→运行。现在我们一起创建第一个控制台应用程序,输出一句话“Hello World!”。

“Hello World!”是一个很经典同时也很简单的程序,它表达了程序员用一门新语言创造出第一个程序的惊喜。



图 1-4 启动 Visual Studio 2010

1. 新建项目

①在 Visual Studio 2010 菜单栏中选择“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“HelloWorld”。

④为项目选择一个保存的位置,如 G:\visual studio 2010\,如图 1-5 所示。



图 1-5 “新建项目”对话框

单击“确定”按钮后,就创建了一个 C# 代码模板,如图 1-6 所示。

⑤在 Main() 方法中添加如下代码。

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

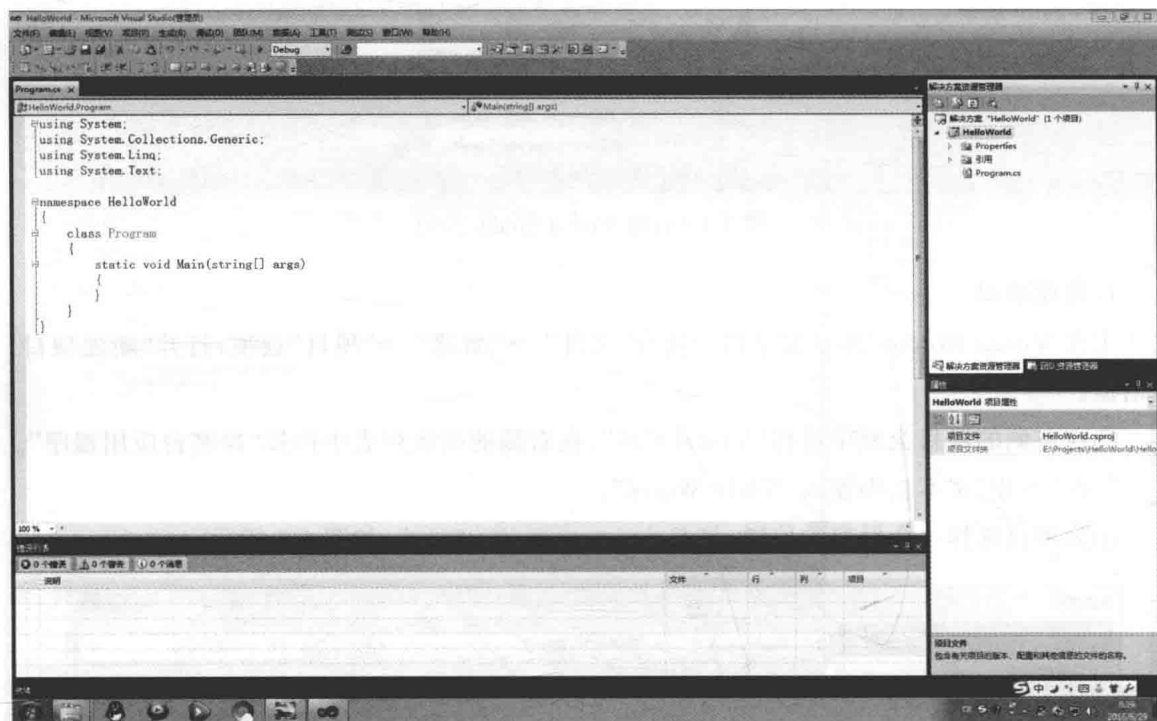


图 1-6 C# 代码模板

2. 生成可执行文件

在 Visual Studio 2010 菜单栏中选择“生成”→“生成解决方案”选项,或者按 F6 功能键。如果错误列表中没有显示错误和警告,并且 Visual Studio 2010 的状态栏中显示“生成成功”,就表示代码没有编译错误,可以开始运行了。

3. 运行

在 Visual Studio 2010 菜单栏中选择“调试”→“开始运行(不调试)”选项,或者按 Ctrl+F5 快捷键,如果得到如图 1-7 所示的结果,则表明运行成功。

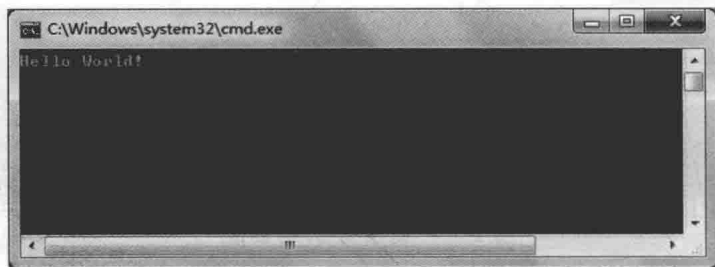


图 1-7 “HelloWorld”项目的输出结果

1.3.2 C# 程序的特点

成功运行了第一个 C# 程序后,让我们来看看 C# 程序是怎样组织的。

前面建立项目的时候,已经在 G:\visual studio 2010\下创建了一个与“HelloWorld”项目同名的文件夹,叫作解决方案文件夹。那么什么是解决方案呢?解决方案和项目都是 Visual



Studio 2010 提供的有效管理应用程序的容器。一个解决方案可以包含一个或多个项目,而每个项目都能够解决一个独立的问题。在本书中,我们的工作都是在某个解决方案的唯一一个项目中完成的。

Visual Studio 2010 提供了一个叫作解决方案资源管理器的窗口,在这里我们可以管理解决方案中包含的各类文件。在解决方案资源管理器中,单击  (显示所有文件)按钮,就可以看到项目的结构了,如图 1-8 所示。

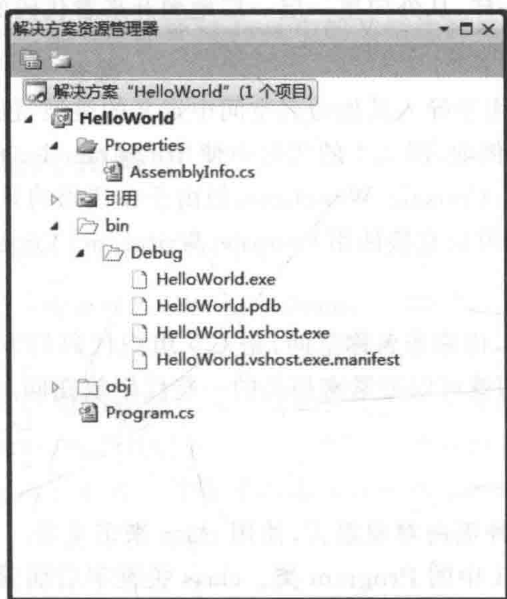


图 1-8 解决方案资源管理器

我们需要认识下面两个文件。

- Program.cs:该文件是项目的启动文件,在该文件中定义项目的启动入口,即 Main()方法。在 C# 中,程序源文件以“cs”作为扩展名。

- HelloWorld.exe:该文件位于 bin\Debug 目录下,是项目编译后生成的可执行文件,可以直接运行。

在解决方案资源管理器中双击 Program.cs,可以打开该文件,代码如例 1.1 所示。

例 1.1 控制台应用程序举例。

```
using System;                                //指示.NET 编译器引入 System.dll 类库中的代码
using System.Collections.Generic;             //引入泛型命名空间
using System.Linq;                            //引入语言集成查询 (LINQ)
using System.Text;                            //引入提供常用的字符串操作
/* 以上 4 个名称空间都是.NET 框架中的基础类库,用于实现一些基本的类的引入,在类创建时
   自动引入 */
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
```

```
{  
    Console.WriteLine("Welcome to the C# programming world!");  
    //输出“Welcome to the C# programming world!”  
    Console.ReadLine();  
}  
}
```

下面我们就像剥洋葱一样,从外向里一层一层地剥开来看代码的各个组成部分。

1. using

using 关键字的功能是用于导入其他命名空间中定义的类型,包括.NET 类库。它的作用和 Java 中的 import 类似。例如,例 1.1 的代码中使用的 Console.WriteLine()方法实际上是一个简写,其全称是 System.Console.WriteLine,但由于在代码的开始使用 using 指令引入了 System 命名空间,所以后面可以直接使用 Console.WriteLine()方法来进行输出。

2. namespace

namespace 即命名空间,也称作名称空间,是 C# 组织代码的方式,它的作用类似于 Java 中的 package(包),这样我们就可以把紧密相关的一些代码放在同一个命名空间中,从而大大提高管理和使用的效率。

3. class

C# 与 Java 一样,是一种面向对象语言,使用 class 来定义类。每一个 C# 的程序中至少包括一个自定义类,如例 1.1 中的 Program 类。class 关键字后面紧跟类名,类名后的左大括号“{”表示类定义的开始,右大括号“}”表示类定义的结束。C# 中的括号必须成对出现,否则会产生编译错误。

4. Main()方法

C# 创建的可执行程序中必须包含一个 Main()方法,Main()方法是程序运行的入口,程序从这里开始运行。要特别注意的是,C# 中的 Main()方法的首字母必须大写;Main()方法的返回值可以是 void 或者 int 类型;Main()方法可以没有命令行参数。因此,这样组合一下,C# 中的 Main()方法有 4 种形式。

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

这 4 种 Main()方法都是正确的,可以根据需要自行选择。代码模板自动生成的是第 1 种。

5. 关键代码

在例 1.1 中,Main()方法中添加的两行代码是这个程序的关键代码,是用来输入和输出的。

```
Console.WriteLine("Welcome to the C# programming world!");    /* 从控制台输出“Welcome  
to the C# programming
```