



“十三五”职业教育国家规划教材

# 物联网C#程序开发 案例式教程

WULIANWANG C#CHENGXU KAIFA  
ANLISHI JIAOCHENG

○主编 吴建宁

江苏凤凰教育出版社 凤凰职教



“十三五”职业教育国家规划教材

# 物联网C#程序开发 案例式教程

WULIANWANG C#CHENGXU KAIFA  
ANLISHI JIAOCHENG

- 主 编 吴建宁
- 副主编 王灿田 程 涛
- 编 者 吴建宁 王灿田 程 涛  
林宗晖 郭健杰

## 图书在版编目(CIP)数据

物联网 C# 程序开发案例式教程 / 吴建宁主编. — 南京:江苏凤凰教育出版社, 2019. 7(2024. 8 重印)

ISBN 978-7-5499-8031-4

I. ①物… II. ①吴… III. ①C 语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 095205 号

书 名 物联网 C# 程序开发案例式教程

---

主 编 吴建宁  
责任编辑 杨小军  
出版发行 江苏凤凰教育出版社  
地 址 南京市湖南路 1 号 A 楼, 邮编:210009  
出 品 江苏凤凰职业教育图书有限公司  
网 址 <http://www.fhmooc.com>  
照 排 南京普胜印刷技术有限公司  
印 刷 河北铄柠印刷有限责任公司  
厂 址 河北省衡水市武邑县兴旺路以南, 邮编:053000  
电 话 0318-2212090  
开 本 787 毫米×1 092 毫米 1/16  
印 张 17.25  
版 次 2019 年 7 月第 1 版 2024 年 8 月第 4 次印刷  
标准书号 ISBN 978-7-5499-8031-4  
定 价 45.80 元  
批发电话 025-83677909  
盗版举报 025-83658893

---

如发现质量问题,请联系我们。

【内容质量】电话:025-83658873 邮箱:sunyi@ppm.cn

【印装质量】电话:025-83677905



本书是江苏省电子专业协作组物联网专业建设示范性教材之一,是中、高职电子与信息技术类专业必修课程。依据电子与信息技术类职业资格标准和职业岗位调研,参照江苏联合职业技术学院物联网专业课程标准,在教材内容和呈现形式等方面改革创新,策应中、高职课程教材衔接,体现职业教育信息化,有效推动项目教学、场景教学、岗位教学等教学模式改革。

本书从 C# 的开发环境开始介绍,循序渐进地讲解了 WPF 界面布局和元素、数据类型和运算符、流程控制、数组和集合、函数、面向对象编程等内容。每一章节包含案例描述、案例分析、必备知识、案例实现、目标检测等内容。本书内容由浅入深,尤其适合大中专院校的学生和其他立志学习 C# 开发的读者。本书具有以下特点:

1. 适于“案例驱动”教学模式。在案例设计上打破了传统教材枯燥无味的教学示例,采用全新的教学设计理念,设计了一系列生动有趣的教学案例,让读者在娱乐中不知不觉地领会知识要点,理解结构化的编程思想,自觉应用编程方法分析和解决问题,激发出学习兴趣,培养了学习的主动性和积极性,创新学习能力得到一定的提升。

2. 整合物联网专业课程教学需求。全书以 C# 应用程序开发、物联网的相关知识为主线,系统深入地讲解了物联网应用程序开发的相关知识和技能。

3. 充分发挥计算机、互联网等信息技术优势,适应个性化学习需求,为师生提供多介质、多媒体、满足不同教学需求的数字化教学资源,探索信息技术条件下教学模式和教学方法的改革。相关数字资源可登录 [www.](http://www.)

ppve.cn 下载。

本书由南京高等职业技术学校吴建宁任主编,王灿田、程涛任副主编。全书编写方案设计和统稿工作由吴建宁负责。吴建宁编写第 1 章,王灿田负责编写第 2 章、第 3 章,程涛负责编写第 4 章、第 5 章、第 6 章,林宗晖、郭健杰分别编写第 7 章和第 8 章。本书承蒙常州工程职业技术学院杨小来副教授和南京信息职业技术学院周波副教授仔细审阅,并提出了宝贵意见和建议,在此表示衷心的感谢!

本书在编写过程中,得到北京新大陆时代教育科技有限公司的大力支持,在此表示衷心的感谢!

本书编写过程中,编者尽可能做到把物联网的相关知识、技能,将最新、最准确的信息传递给读者,由于时间仓促,加上编者水平有限,书中难免有不妥和错漏之处,恳请读者批评指正。来信请发至 1092785565@qq.com 邮箱。

编 者

2019 年 4 月



# 目录

## 第一章

### C#语言概述

第一节 C#简介 .....	1
第二节 .NET Framework 简介 .....	2
第三节 Visual Studio2013 软件介绍 .....	4
第四节 WPF 工程创建 .....	4
案 例 如何才能成才? .....	8
案例描述 .....	8
案例分析 .....	8
案例实现 .....	9
目标检测 .....	12

## 第二章

### WPF界面布局与元素

案例一 三国猛将选择器 .....	13
案例描述 .....	13
案例分析 .....	14
必备知识 .....	14
案例实现 .....	17

<b>案例二 埃舍尔密铺</b> .....	24
案例描述 .....	24
案例分析 .....	25
必备知识 .....	25
案例实现 .....	26
<b>案例三 文臣与武将</b> .....	31
案例描述 .....	31
案例分析 .....	31
必备知识 .....	32
案例实现 .....	33
案例拓展 .....	37
目标检测 .....	40

## 第三章

### 数据类型与运算符

<b>案例一 追 星</b> .....	41
案例描述 .....	41
案例分析 .....	42
必备知识 .....	42
案例实现 .....	45
<b>案例二 跳一跳</b> .....	50
案例描述 .....	50
案例分析 .....	51
必备知识 .....	51
案例实现 .....	52
<b>案例三 穿越选择器</b> .....	54

案例描述 .....	54
必备知识 .....	55
案例实现 .....	56
<b>案例四 音乐播放器 .....</b>	<b>61</b>
案例描述 .....	61
案例分析 .....	62
必备知识 .....	62
案例实现 .....	63
<b>案例五 看我七十二变 .....</b>	<b>67</b>
案例描述 .....	67
案例分析 .....	68
必备知识 .....	68
案例实现 .....	69
目标检测 .....	74

## 第四章

### 流程控制

<b>案例一 一起来找茬 .....</b>	<b>76</b>
案例描述 .....	76
案例分析 .....	77
必备知识 .....	77
案例实现 .....	80
<b>案例二 柠檬播放器 .....</b>	<b>86</b>
案例描述 .....	86
案例分析 .....	86
必备知识 .....	86



案例实现 .....	88
<b>案例三 猜拳游戏 .....</b>	<b>91</b>
案例描述 .....	91
案例分析 .....	92
必备知识 .....	93
案例实现 .....	93
<b>案例四 冰雹猜想 .....</b>	<b>96</b>
案例描述 .....	96
案例分析 .....	97
必备知识 .....	97
案例实现 .....	98
<b>案例五 双 11 购物节 .....</b>	<b>100</b>
案例描述 .....	100
案例分析 .....	101
必备知识 .....	101
案例实现 .....	103
<b>案例六 博饼计算器 .....</b>	<b>105</b>
案例描述 .....	105
案例分析 .....	106
必备知识 .....	107
案例实现 .....	107
<b>案例七 橘子减肥法 .....</b>	<b>112</b>
案例描述 .....	112
案例分析 .....	113
必备知识 .....	113
案例实现 .....	113

目标检测 ..... 115

数组与集合

案例一 自助点餐平台 ..... 116

    案例描述 ..... 116

    案例分析 ..... 117

    必备知识 ..... 117

    案例实现 ..... 119

案例二 随机点名小程序 ..... 121

    案例描述 ..... 121

    案例分析 ..... 122

    必备知识 ..... 122

    案例实现 ..... 123

案例三 神探狄仁杰 ..... 126

    案例描述 ..... 126

    案例分析 ..... 126

    必备知识 ..... 127

    案例实现 ..... 128

案例四 新能源车平台 ..... 130

    案例描述 ..... 130

    案例分析 ..... 132

    必备知识 ..... 132

    案例实现 ..... 133

案例五 MAC 管理 ..... 138

    案例描述 ..... 138

## 第六章

### 函 数

案例分析 .....	139
必备知识 .....	140
案例实现 .....	141
目标检测 .....	144
<b>案例一 偷吃人参果 .....</b>	<b>145</b>
案例描述 .....	145
案例分析 .....	146
必备知识 .....	146
案例实现 .....	148
<b>案例二 精灵宝可梦 .....</b>	<b>150</b>
案例描述 .....	150
案例分析 .....	151
必备知识 .....	151
案例实现 .....	154
<b>案例三 汇率换算器 .....</b>	<b>161</b>
案例描述 .....	161
案例分析 .....	161
必备知识 .....	162
案例实现 .....	162
<b>案例四 个税计算器 .....</b>	<b>167</b>
案例描述 .....	167
案例分析 .....	168
必备知识 .....	169

案例实现 ..... 171  
目标检测 ..... 177

面向对象编程

**案例一 川菜之夫妻肺片** ..... 178  
    案例描述 ..... 178  
    案例分析 ..... 181  
    必备知识 ..... 181  
    案例实现 ..... 184

**案例二 秦始皇** ..... 192  
    案例描述 ..... 192  
    案例分析 ..... 192  
    必备知识 ..... 193  
    案例实现 ..... 194

**案例三 公交卡注册平台** ..... 199  
    案例描述 ..... 199  
    案例分析 ..... 200  
    必备知识 ..... 200  
    案例实现 ..... 202

**案例四 密林心脏** ..... 207  
    案例描述 ..... 207  
    案例分析 ..... 210  
    必备知识 ..... 210  
    案例实现 ..... 211

**案例五 长津湖之水门桥** ..... 220

## 第八章

案例描述 .....	220
案例分析 .....	221
必备知识 .....	221
案例实现 .....	222
目标检测 .....	226

### IO与JSON

<b>案例一 人生打印机</b> .....	228
案例描述 .....	228
案例分析 .....	230
必备知识 .....	230
案例实现 .....	231
<b>案例二 幸运大抽奖</b> .....	237
案例描述 .....	237
案例分析 .....	238
必备知识 .....	238
案例实现 .....	240
<b>案例三 智能家居</b> .....	246
案例描述 .....	246
案例分析 .....	247
必备知识 .....	248
案例实现 .....	252
目标检测 .....	259
<b>参考文献</b> .....	260



# 第一章

## C#语言概述



### 学习目标

- 了解 C# 语言的概念及其发展历程。
- 了解 .NET Framework。
- 熟悉 Visual Studio 2013 开发工具。
- 掌握简单 WPF 工程的开发。

## 第一节 C#简介

### 一、了解 C#

C# 读作 C Sharp。它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# 所开发的程序源代码并不是编译成能够直接在操作系统上执行的二进制本地代码。与 Java 类似，它被编译成为中间代码，然后通过 .NET Framework 的虚拟机——被称之为通用语言运行时 (CLR)——执行。所有的 .NET 编程语言都被编译成这种被称为 MSIL (Microsoft Intermediate Language) 的中间代码。因此，虽然最终的程序在表面上仍然与传统意义上的可执行文件都具有“.exe”的后缀名，但是实际上，如果计算机上没有



安装 .NET Framework, 那么这些程序将不能够被执行。在程序执行时, .NET Framework 将中间代码翻译成二进制机器码, 从而使它得到正确的运行。最终的二进制代码被存储在一个缓冲区中, 所以一旦程序使用了相同的代码, 那么将会调用缓冲区中的版本。这样如果一个 .NET 程序第二次被运行, 那么这种翻译不需要进行第二次, 速度明显加快。

## 二、C# 语言的发展

C# 语言的发展主要经历了以下阶段:

### 1. C# 1.0——纯粹地面向对象

在 2003 年 5 月, 微软推出了 Visual Studio .NET 2003, 同时也发布了 C# 的改进版本——C# 1.1。这一时期的 C# 提出了纯粹地面向对象概念。C++ 并非纯面向对象的, 为了和 C 兼容以及提供更高的执行效率, 它保留了很多模块化的东西。C# 还通过类类型、值类型和接口类型的概念形成了统一的类型系统。

### 2. C# 2.0——泛型编程新概念

微软在 2004 年 6 月份发布了 Visual Studio 2005 的第一个 Beta 版, 同时向开发者展示了 C# 语言的 2.0 版本。C# 2.0 为开发者带来的最主要的特性就是泛型编程能力, 另一个突出的特性就是匿名方法, 用来取代一些短小的并且仅出现一次的委托, 使得语言结构更加紧凑。

### 3. C# 3.0——研发代号“Orcas”

2005 年 9 月份的 PDC 大会——C# 3.0(研发代号“Orcas”——魔鬼)的技术预览版。说到 C# 3.0, 就不得不提一下微软的 LINQ 项目, LINQ(语言集成查询, Language Integrated Query)提出了一种通过面向对象语法来实现对非面向对象数据源的查询技术, 可查询的数据源从关系型数据库延伸到一般意义上的集合(如数组和列表)以及 XML。而 C# 3.0 则是率先实现了 LINQ 的语言。

### 4. C# 4.0——动态编程

C# 4.0 新增 dynamic 关键字, 提供动态编程(dynamic programming), 把既有的静态物件标记为动态物件, 类似 javascript、Python 或 Ruby。

### 5. C# 5.0

C# 5.0 增加了带参数的泛型构造函数、支持 null 类型运算、case 支持表达式、扩展属性和绑定运算符等。

## 第二节 .NET Framework 简介

Microsoft .NET Framework 是用于 Windows 的新托管代码编程模型。它将强大的功能与新技术结合起来, 用于构建具有视觉上引人注目的用户体验的应用程序, 实现跨

技术边界的无缝通信,并且能支持各种业务流程。

.NET Framework 具有两个主要组件:公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础,您可以将运行库看作一个在执行时管理代码的代理,它提供内存管理、线程管理和远程处理等核心服务,并且还强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要组件是类库,它是一个综合性的面向对象的可重用类型集合,您可以使用它开发多种应用程序,这些应用程序包括传统的命令行或图形用户界面(GUI)应用程序,也包括基于所提供的最新创新的应用程序(如 Web 窗体和 XML Web services)。

.NET Framework 使用的版本有:.NET Framework 1.0、.NET Framework 1.1、.NET Framework 2.0、.NET Framework 3.0、.NET Framework 3.5、.NET Framework 4.0、.NET Framework 4.5、.NET Framework 4.6、.NET Framework 4.7。新版本包含旧版本的功能,并且增加了新的特性。可参考图 1-1 所示。

目前.NET Framework 最高版本为.NET Framework 4.7.2,本教材所使用的.NET 框架为 4.5。

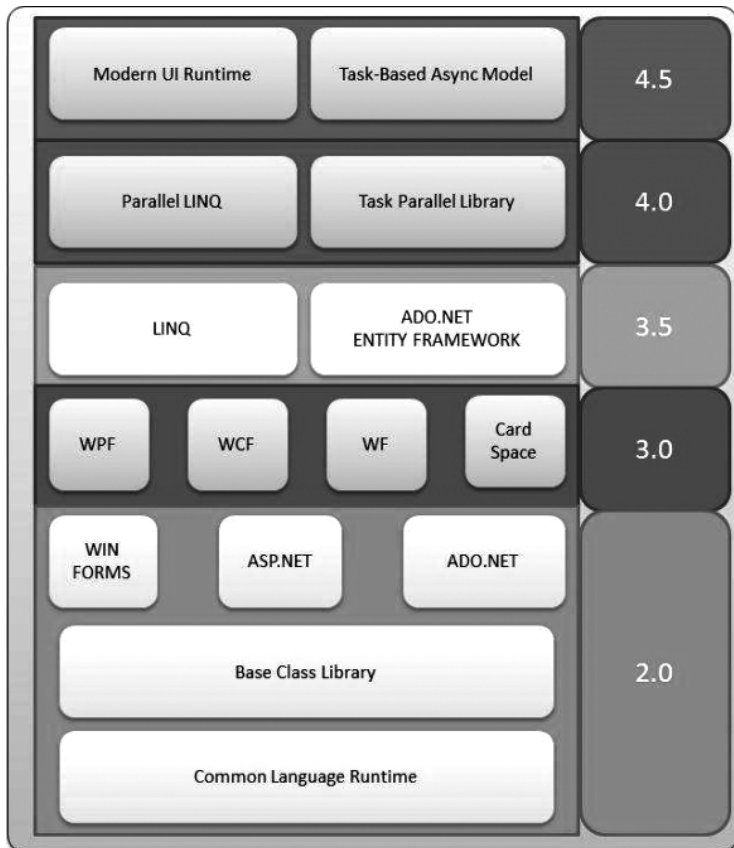


图 1-1 .NET Framework 各版本特性





### 第三节 Visual Studio2013 软件介绍

本书所有程序使用的开发工具为 Visual Studio2013。Visual Studio 是微软公司推出的开发环境,是最流行的 Windows 平台应用程序开发环境。安装 Visual Studio2013 建议使用 Windows7(32 位或 64 位)操作系统,内存应在 2GB 以上。Visual Studio2013 与 .NET Framework 4.5 学员可以从微软官方网站下载。安装步骤较为简单,本书将不做赘述。安装成功后,可以在“开始”菜单中看到 Visual Studio2013 的图标(图 1-2),点击进去后就可以开始我们的开发旅程了(图 1-3)。



图 1-2 “开始”菜单中的 Visual Studio2013 文件夹

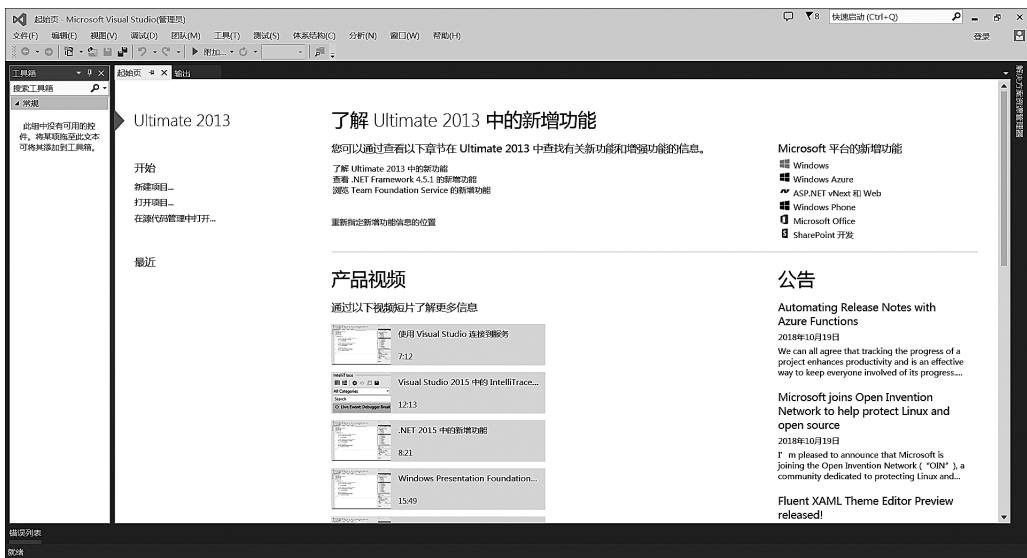


图 1-3 Visual Studio2013 开发环境

### 第四节 WPF 工程创建

WPF 为 Windows Presentation Foundation 的首字母缩写,中文译为“Windows 呈现基础”,其原来代号为“Avalon”,因与“我佩服”拼音首字母组合一样,国内有人调侃地称之为“我佩服”。它是微软新一代图形系统。

基于 DirectX 9/10 技术的 WPF 不仅带来了前所未有的 3D 界面,而且其图形向量渲染引擎也大大改进了传统的 2D 界面,比如 Vista 中的半透明效果的窗体等都得益于 WPF。程序员在 WPF 的帮助下,要开发出媲美 Mac 程序的酷炫界面已不再是遥不可及的奢望。WPF 相对于 Windows 客户端的开发来说,向前跨出了巨大的一步,它提供了超丰富的 .NET UI 框架,集成了矢量图形,丰富的流动文字支持(flow text support),3D 视觉效果和强大无比的元素模型框架。

分工与合作的关系存在于社会各个行业之间,也存在于行业内部。同样,在软件开发中也存在此关系,其最典型的分工合作就是设计师(Designer)与程序员(Programmer)之间的协作。

Microsoft 在引入 Windows Presentation Foundation 的同时,还引入了 XAML(读音为“zamel”),这是一种公开表示 Windows 应用程序用户界面的标记语言,可使开发人员和设计人员用来构建和重用 UI 的工具更加丰富。对于 Web 开发人员,XAML 提供了熟悉的 UI 说明模式。XAML 还使 UI 设计从基础代码中分离出来,从而使开发人员和设计人员之间的合作更加紧密。XAML 不像 HTML 和 XHTML 那样只应用在 Web 开发领域,它对于 Web 开发和桌面开发都是适应的。如今,为了支持 WPF 程序设计,微软推出了专门的、使用 XAML 语言进行 UI 设计工具——Expression Studio,使用它就像使用 PhotoShop 和 Dreamweaver 一样,设计出来的结果保存为 XAML 文件,程序员可以直接拿来用;当 UI 有变更时,程序员只需用新版 XAML 文件替换旧版即可。

WPF 开发中,XAML 标记语言的文件扩展名为 .xaml,而 C# 语言的文件扩展名为 .xaml.cs 文件,两者相辅相成。在创建完本章节 WPF 案例工程后可以在文件中看到图 1-4 所示文件结构:



图 1-4 WPF 项目结构

本教材的案例都将使用 WPF 工程项目。现在先来创建个 WPF 工程项目,看看它有哪些特点吧?

(1) 首先,选择“文件”→“新建”→“项目”,选择开发语言为 Visual C#,项目类型为 WPF 应用程序,并且为解决方案、应用程序取名并设置存储路径,如图 1-5 所示:

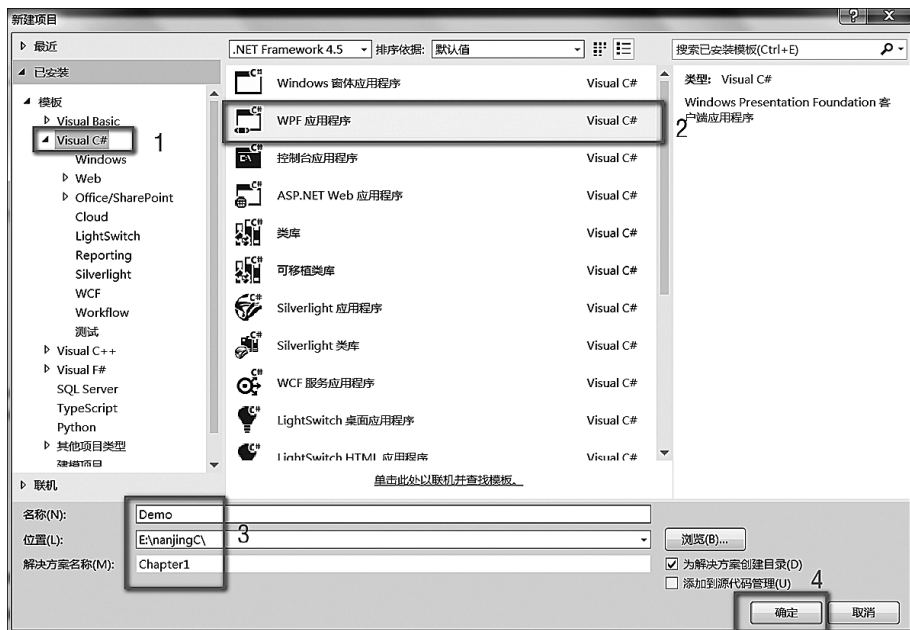


图 1-5 新建 WPF 应用程序

(2)设置完成后点击“确定”按钮,即可看到主窗体。如图 1-6 所示:

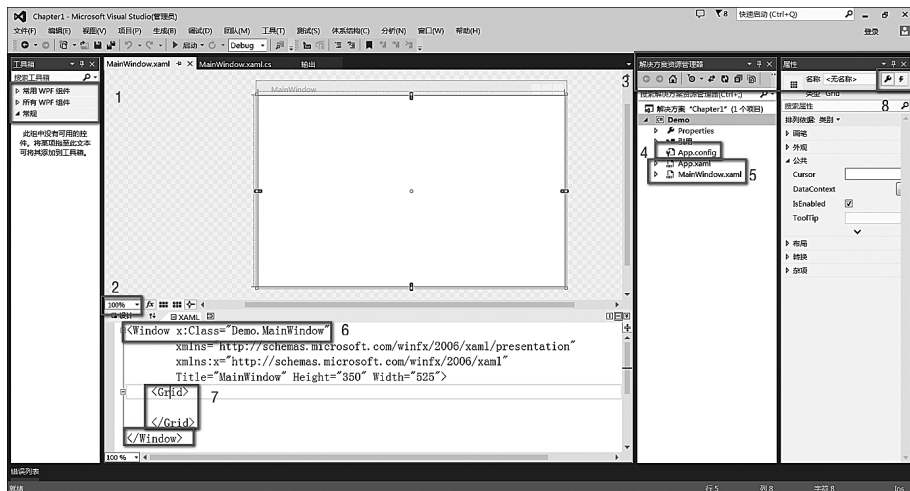




图 1-6 WPF 应用程序主窗体

在项目开发中,常用的界面有:工具箱、解决方案资源管理器、属性,如果各位学员的界面上并未看到这些界面,也可以在菜单栏中的“视图”中找到它们。

从上方 XAML 代码中,可以看到目前 WPF 的界面元素有:Window 和 Grid(网格),详见本章案例。成对的<Grid></Grid>或<Window></Window>写法类似于 HTML 中的标签,所以在 WPF 中,它们既是 WPF 的界面元素,也是标签。在 HTML 中,常见类似于

<body>之类的标签,它们的标签由<标签>开始,也由</标签>结束,这被称之为成对标签,也可见类似于<br/>标签的写法,“/”也代表了该标签的结束。

标签是具有属性和事件的,例如 Windows 标签中所见的 X : Name 属性,该属性除了为这个标签生成对应实例外,还为该实例声明一个引用变量,变量名就是 X : Name 的值。同学们可以在 XAML 代码中修改 X : Name 的值,也可以在属性界面上修改“名称”。

属性界面中,可以修改元素的属性,也可以编写元素的事件。属性可以按类型排列,也可以按名称排列,名称将按 A~Z 的顺序排列。

当前窗体中运行的 MainWindow 窗体为运行的首窗体,MainWindow 窗体包含 MainWindow.xaml 文件和 MainWindow.xaml.cs 文件。MainWindow.xaml 为设计界面文件,MainWindow.xaml.cs 为后台代码文件,两者就像一对筷子,任意一个文件代码出错,窗体都运行不起来。


右击“Chapter1”解决方案,选择 在文件资源管理器中打开文件夹(X),可以看到以工程名称命名的文件夹与以 .sln 为扩展名的解决方案文件(图 1-7)。需留意,解决方案名称不一定与工程名字相同。



图 1-7 WPF 文件夹结构

双击“Demo”文件夹,可以看到如图 1-8 的文件。

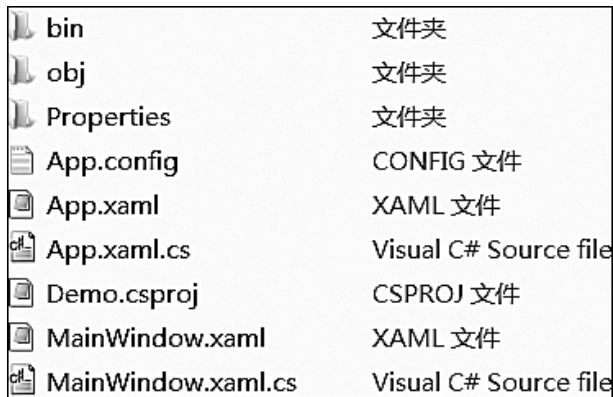


图 1-8 “Demo”文件夹结构

项目编辑完成后,点击菜单栏中的 启动 图标后,将在 bin\Debug 文件夹下生成 Demo.exe 可执行文件。



## 案例 如何才能成才?

### 案例描述

学生:我想学习 C 语言,Java 语言,Python,C#,C++,HTML5,JavaScript 等编程语言,老师你觉得可以成才嘛?

老师:贪多嚼不烂!

学生:如果我学好 C#,可以成才吗?

老师:学好了可以成才!

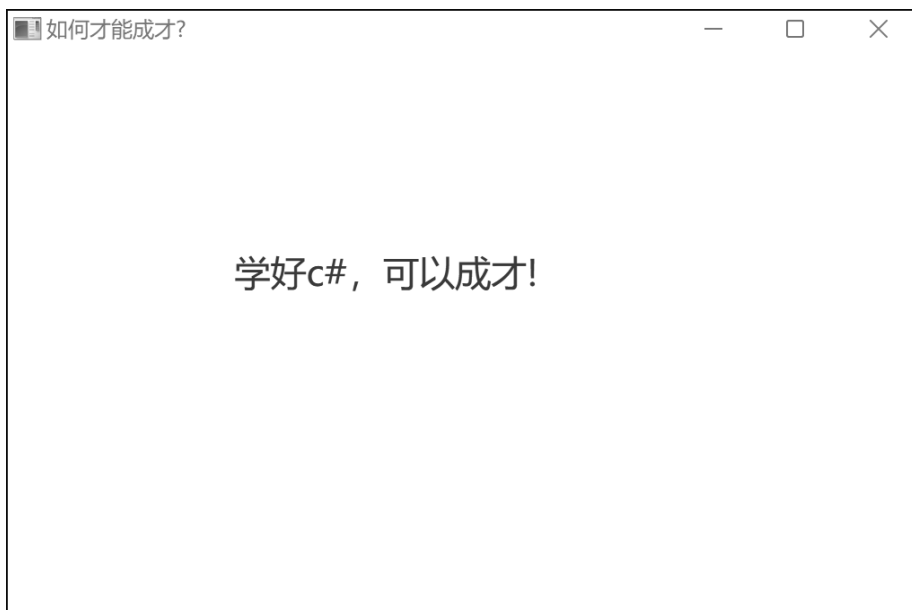


图 1-9 案例效果图

### 案例分析

该案例涉及的控件有:Label,但是需要修改的元素有 Window、Label,以及元素的属性。比如 Label 标签的 FontSize(字体大小)、Foreground(字体颜色)、Content(内容)。

## 案例实现

(1) 创建 Chapter1 解决方案→添加项目“Demo”，右击“Demo”→“设为启动项目”。  
如图 1-10 所示。

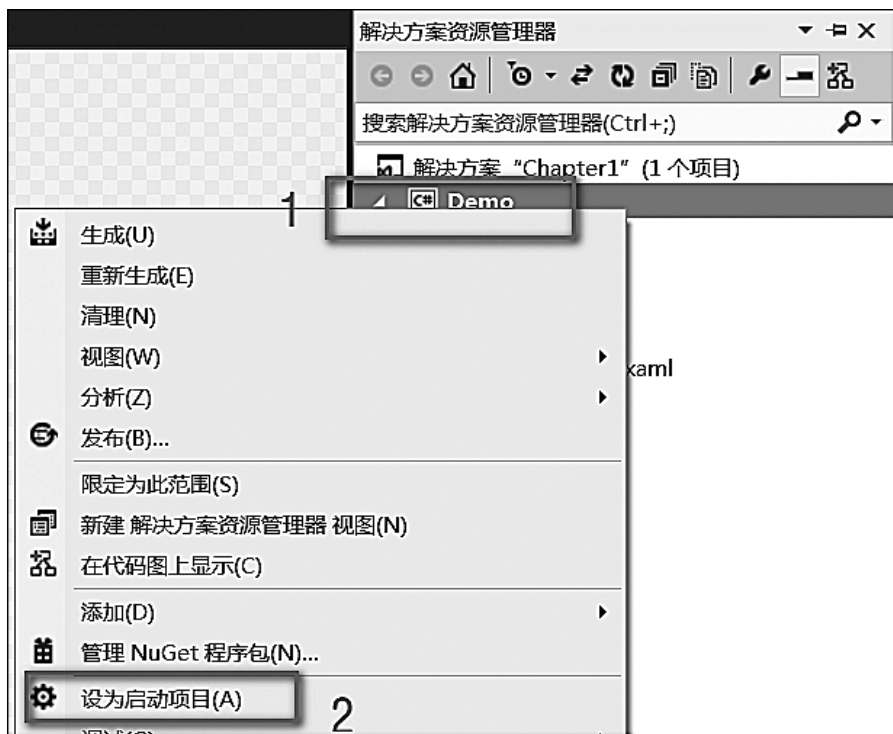


图 1-10 设置启动项目

(2) 拖动“工具箱”中的 Label 到 XAML 界面中,如图 1-11 所示。

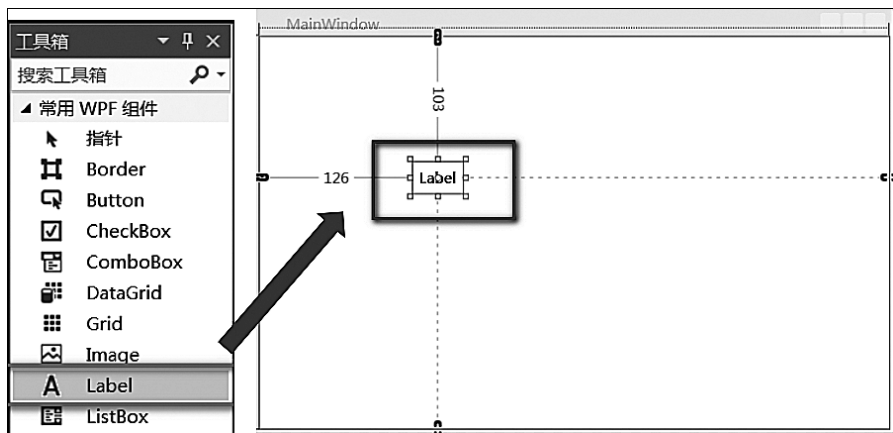


图 1-11 添加 Label 控件到窗体中



(3) 修改窗体标题为：“如何才能成才？”如图 1-12 所示。

```
<Window x:Class = "WpfApplication1.MainWindow"
    xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml"
    Title = "如何才能成才?" Height = "350" Width = "525">
<Grid>
    <Grid.ColumnDefinitions>
```

图 1-12 修改窗体标题

(4) 单击 Label 控件,在“属性”视图中修改控件名称,如图 1-13 所示。



图 1-13 修改名称

(5) 修改成功后,则可以在 XAML 文件中看到 Label 标签多了项 `x:Name` 属性,如图 1-14 所示。

```
<Grid>
    <Label x:Name = "lblName"
</Grid>
```

图 1-14 `x:Name` 属性

(6) 修改 Label 标签的 `Content` 属性值(`Content` 属性用于显示标签内容),如图 1-15 所示。

```
</Grid.ColumnDefinitions>
<Label x:Name = "lblName" Content = "学好 C#, 可以成才!"
```

图 1-15 `Content` 属性

(7) 修改 Label 标签的 `FontSize` 属性值(`FontSize` 属性用于修改字体大小),如图 1-16 所示。

(8) 单击 Label 控件,在“属性”视图中更换排列依据为:名称,如图 1-17 所示。

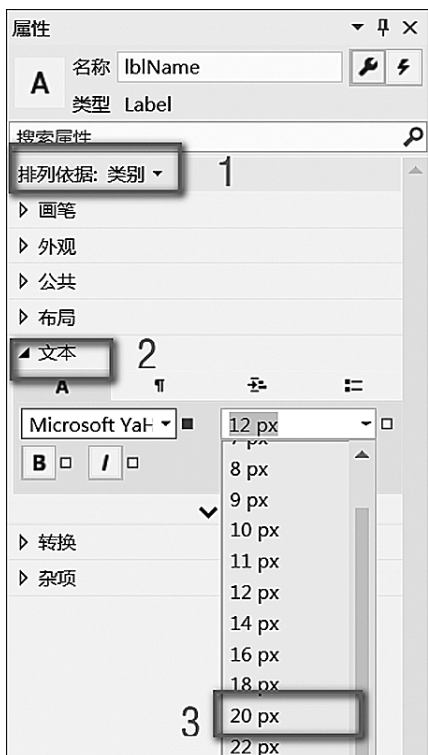


图 1-16 FontSize 属性



图 1-17 排列依据修改为:名称

(9) 在“属性”视图找到 Foreground 属性(Foreground 属性用于修改字体颜色), 选择“纯色画笔”并输入颜色值: #FFAC2222,如图 1-18 所示。

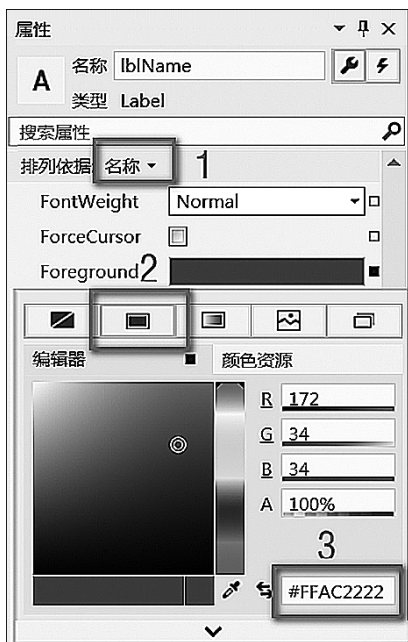


图 1-18 修改字体颜色





(10) 修改完 Label 标签的内容、字体大小、字体颜色后,将在 XAML 界面中看到如图 1-19 效果。

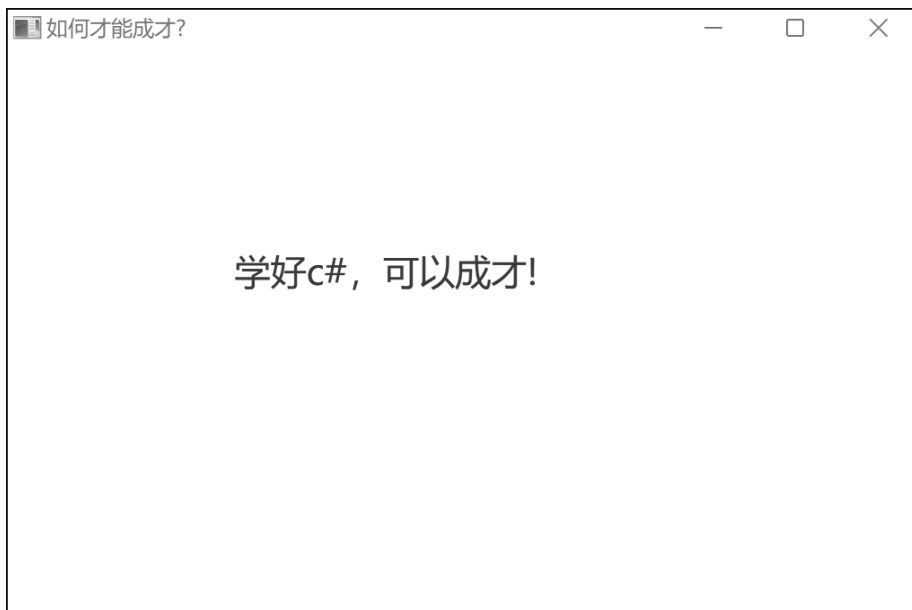


图 1-19 XAML 界面效果

(11) 点击菜单栏中的 ▶ 启动 ▾ 图标,即可看到图 1-9 所示效果。

## 目标检测

1. 请描述 XAML 文件的特点。
2. 请列举您在添加控件(元素)时,WPF 自动生成的属性有哪些? 它们分别代表什么意思?