

高等院校计算机实验与实践系列示范教材

基于C#的Windows 应用程序开发 实验与实践教程

李莹 吴晓艳 编著

清华大学出版社

清华大学出版社 清华大学计算机系列教材

基于C#的Windows 应用程序开发 实验与实践教程

冯 建 编著

013179282X

高等院校计算机实验与实践系列示范教材

基于C#的Windows 应用程序开发 实验与实践教程

清华大学出版社
北京

内 容 简 介

本书介绍如何使用 C# 语言开发 Windows 应用程序。全书共 9 章：第 1 至第 3 章介绍了 Windows 程序中窗体和控件的使用；第 4 章介绍了面向对象编程；第 5 章介绍了 ADO.NET 对象模型；第 6 章介绍了如何在 Windows 中调用外界的功能模块；第 7 章介绍了 GDI+ 在 WinForm 中的使用；第 8 章介绍了如何创建报表；第 9 章介绍了如何提高 Windows 程序的可访问性。

本书内容实用、循序渐进、实践性强，适用于高等职业院校计算机软件相关专业的学生，以及其他 C# 程序设计的学习者。

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

基于 C# 的 Windows 应用程序开发实验与实践教程/李莹, 吴晓艳编著. —北京：清华大学出版社, 2011. 1

(高等院校计算机实验与实践系列示范教材)

ISBN 978-7-302-22789-2

I. ①基… II. ①李…②吴… III. ①C 语言—程序设计—高等学校—教材 ②窗口软件, Windows—程序设计—高等学校—教材 IV. ①TP312 ②TP316.7

中国版本图书馆 CIP 数据核字(2010)第 090425 号

责任编辑：梁 颖 薛 阳

责任校对：李建庄

责任印制：李红英

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62795954, jsjjc@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：北京市清华园胶印厂

经 销：全国新华书店

开 本：185×260 印 张：12.75 字 数：315 千字

版 次：2011 年 1 月第 1 版 印 次：2011 年 1 月第 1 次印刷

印 数：1~3000

定 价：22.00 元

产品编号：033448-01

出版说明

当前,重视实验与实践教育是各国高等教育界的发展潮流,我国与国外教学工作的差距也主要表现在实践教学环节上。面对新的形式和新的挑战,完善实验与实践教育体系成为一种必然。为了培养具有高质量、高素质、高实践能力和高创新能力的人才,全国很多高等院校在实验与实践教学方面进行了大力改革,在实验与实践教学内容、教学方法、教学体系、实验室建设等方面积累了丰富的宝贵经验,起到了教学示范作用。

实验与实践性教学与理论教学是相辅相成的,具有同等重要的地位。它是在开放教育的基础上,为配合理论教学、培养学生分析问题和解决问题的能力以及加强训练学生专业实践能力而设置的教学环节;对于完成教学计划、落实教学大纲,确保教学质量,培养学生分析问题、解决问题的能力 and 实际操作技能更具有特别重要的意义。同时,实践教学也是培养应用型人才的重要途径,实践教学质量的好坏,实际上也决定了应用型人才培养质量的高低。因此,加强实践教学环节,提高实践教学质量,对培养高质量的应用型人才至关重要。

近年来,教育部把实验与实践教学作为对高等院校教学工作评估的关键性指标。2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“高等学校要强化实践育人的意识,区别不同学科对实践教学的要求,合理制定实践教学方案,完善实践教学体系。要切实加强实验、实习、社会实践、毕业设计(论文)等实践教学环节,保障各环节的时间和效果,不得降低要求。”“要不断改革实践教学内容,改进实践教学方法,通过政策引导,吸引高水平教师从事实践环节教学工作。要加强产学研合作教育,充分利用国内外资源,不断拓展校际之间、校企之间、高校与科研院所之间的合作,加强各种形式的实践教学基地和实验室建设。”

为了配合开展实践教学及适应教学改革的需要,我们在全中国各高等院校精心挖掘和遴选了一批在计算机实验与实践教学方面具有潜心研究并取得了富有特色、值得推广的教学成果的作者,把他们多年积累的教学经验编写成教材,为开展实践教学的学校起一个抛砖引玉的示范作用。

为了保证出版质量,本套教材中的每本书都经过编委会委员的精心筛选和

严格评审,坚持宁缺毋滥的原则,力争把每本书都做成精品。同时,为了能够让更多、更好的实践教学成果应用于社会和各高等院校,我们热切期望在这方面有经验和成果的教师能够加入到本套丛书的编写队伍中,为实践教学的发展和取得成效做出贡献;也衷心地期望广大读者对本套教材提出宝贵意见,以便我们更好地为读者服务。

清华大学出版社

联系人:索梅 suom@tup.tsinghua.edu.cn

C#是微软公司为.NET平台量身定做的语言,是基于.NET平台的最佳语言。2002年Microsoft公司发布了C#语言的第一个版本,2005年发布了C#语言的2.0版本。

本书论述了在实际开发Windows应用程序中所涉及的C#特性,适合已掌握C#基本语法的读者使用。本书包含了大量的代码,使用实例来突出要点,希望读者在阅读本书后,会觉得本书生动有趣,能给您使用C#开发Windows程序带来帮助,并能编写真正的代码来解决实际的问题。

本书是在全国进行教育课程教学改革大环境下,为适应高等职业院校项目化教学改革编写的,宗旨是培养学习者的实际编程能力。

本书共分9章,内容安排如下:

第1到第3章,介绍了Windows窗体和常用控件的使用,及创建控件的三种常用方法;

第4章介绍了面向对象的编程技术,以及Windows Forms应用程序与面向对象编程的关系;

第5章介绍了ADO.NET在Windows程序中的使用;

第6章介绍了Windows程序中的调用,包括调用XML Web服务、COM组件及Windows API函数;

第7章介绍了GDI+的编程技术;

第8章介绍了Crystal Reports在Windows程序中的使用,以及使用“报表专家”来创建和格式化报表;

第9章介绍了如何提高Windows程序的可访问性。

本书可以作为高等职业院校软件技术及相关专业的教材,也可以作为其他C#程序设计学习者的参考用书。

本书主要由李莹和吴晓艳编著,另外参与本书编著的人员还有陈汉卿、王鑫、马旭东、刘洋等。

由于时间仓促,编者水平有限,书中难免存在不足之处,敬请广大读者指正。编者的E-mail为:liying0000@sohu.com,将在以后的版本中更正读者指出的错误。

作 者

2010年10月

第 1 章 Windows 窗体	1
1.1 Windows 窗体介绍	1
1.2 创建 Windows 程序	2
1.3 事件处理	5
1.4 多文档界面设计	13
1.4.1 创建多文档界面应用程序	13
1.4.2 排列子窗体	14
1.5 小结	14
1.6 习题	14
第 2 章 Windows 常用控件的使用	15
2.1 Label、TextBox 和 Button 控件	16
2.2 RadioButton 和 CheckBox 控件	19
2.3 其他选项类控件和 ListBox 控件	21
2.3.1 CheckedListBox 控件	21
2.3.2 ComboBox 控件	23
2.3.3 DateTimePicker 控件	25
2.3.4 ListBox 控件	26
2.4 MenuStrip、ToolStrip 和 StatusStrip 控件	28
2.4.1 MenuStrip 控件	28
2.4.2 ToolStrip 控件	30
2.4.3 StatusStrip 控件	31
2.4.4 综合示例	32
2.5 TreeView 和 ListView 控件	38
2.5.1 TreeView 控件	38
2.5.2 ListView 控件	40
2.5.3 SplitContainer 控件	43
2.5.4 综合示例	44
2.6 通用对话框的使用	45

2.6.1	打开和保存文件对话框	47
2.6.2	字体和颜色对话框	49
2.6.3	打印对话框	50
2.7	其他常用控件	54
2.7.1	计时器	54
2.7.2	进度条	56
2.7.3	任务栏图标	57
2.7.4	页面浏览器	58
2.8	实验	60
2.8.1	实验 1: 控件的使用 1	60
2.8.2	实验 2: 控件的使用 2	60
2.8.3	实验 3: 控件的使用 3	60
2.9	小结	61
2.10	习题	61
第 3 章	创建控件	62
3.1	复合控件	62
3.2	扩展控件	67
3.3	自定义控件	71
3.4	实验 4: 创建控件	73
3.5	小结	73
3.6	习题	73
第 4 章	面向对象编程简介	74
4.1	对象和类	74
4.1.1	对象	74
4.1.2	类	74
4.1.3	类成员	75
4.1.4	对象的创建和实例化	77
4.1.5	方法的参数	78
4.2	封装	83
4.3	继承	83
4.4	多态	86
4.4.1	虚方法的重载	86
4.4.2	抽象类与抽象方法	87
4.5	接口	88
4.6	Windows 程序中的面向对象	99
4.7	小结	101
4.8	习题	101

第 5 章 使用 ADO.NET 访问数据	102
5.1 ADO.NET 概述	102
5.1.1 ADO.NET 对象模型	102
5.1.2 使用命名空间	103
5.2 Connection 对象的使用	103
5.2.1 选择.NET 数据提供程序	103
5.2.2 使用 SqlConnection 对象	104
5.2.3 使用 OleDbConnection 对象	105
5.3 Command 对象的使用	105
5.3.1 插入、修改、删除数据	106
5.3.2 读取数据	107
5.3.3 执行存储过程	111
5.4 DataAdapter 和 DataSet 对象的使用	112
5.4.1 填充 DataSet	112
5.4.2 更新 DataSet	113
5.5 ADO.NET 与 XML	115
5.6 实验 5：在 Windows 程序中使用 ADO.NET	115
5.7 小结	116
5.8 习题	116
第 6 章 Windows 程序中的调用	117
6.1 调用 XML Web Service	117
6.1.1 创建一个简单的 XML Web Services	118
6.1.2 使用 C# 开发 XML Web Services 客户端	120
6.1.3 实验 6：调用 XML Web Services	122
6.2 调用 COM 组件	123
6.2.1 什么是 COM 组件	123
6.2.2 在.NET 中调用 COM 组件	123
6.2.3 实例 1：把文本写入 Word 文档	124
6.2.4 实例 2：将数据库中数据导出为 Word 文档	125
6.3 调用 Windows API	129
6.3.1 在 Windows 程序中调用 API	129
6.3.2 实例 3：运行可执行文件	129
6.3.3 实例 4：播放声音	130
6.4 小结	132
6.5 习题	132
第 7 章 GDI+ 的简单编程	133
7.1 GDI+ 和 Graphic 类	133

7.1.1	创建 Graphics 对象	135
7.1.2	基本的 GDI+对象	137
7.1.3	基本图形的绘制	142
7.2	实例 1: 绘制不规则连线	146
7.3	实例 2: 分栏显示文字	148
7.4	实例 3: 复制图像指定区域	150
7.5	实例 4: 使用 GDI+	152
7.6	小结	156
7.7	习题	156
第 8 章	使用 Crystal Reports 创建报表	157
8.1	Crystal Reports 概述	157
8.2	创建报表的基本步骤	158
8.3	使用查看器查看报表	159
8.4	实例 1: 利用向导创建报表	160
8.5	实例 2: 加载水晶报表文件	165
8.6	实例 3: 在报表中筛选数据记录	167
8.7	实例 4: 在报表中分组排序记录	171
8.8	实例 5: 在报表中分组汇总记录	173
8.9	实例 6: 在报表中制作数据图表	175
8.10	小结	178
8.11	习题	178
第 9 章	提高程序的可访问性	179
9.1	界面设计原则	179
9.2	添加辅助功能特性	180
9.2.1	.NET Framework 的辅助功能支持	180
9.2.2	为 Windows Form 应用程序添加辅助特性	182
9.2.3	测试辅助功能	182
9.3	为应用程序添加帮助	183
9.3.1	.NET 支持的帮助类型	183
9.3.2	为窗体和控件添加区分上下文帮助	183
9.3.3	将帮助主题链接到菜单	185
9.3.4	显示 ToolTip 帮助	186
9.4	应用程序的本地化	186
9.5	小结	191
9.6	习题	191
参考文献		192

Windows 操作系统将程序员和用户带到了图形用户界面中。使用 C# 语言和 Visual Studio.NET 工具不仅可以开发具有传统的 Windows 应用程序功能的程序,还可以开发高效强大的 Windows 应用程序。因为 .NET Framework 为 Windows 应用程序增加许多新技术与新功能。如使用 GDI+ 技术可以扩展 Windows 控件的外观和功能,使应用程序界面变得丰富多彩;利用面向对象的编程技术可以提高代码的复用性;利用 ADO.NET 技术可以快速开发数据领域的应用程序;利用资源文件可以开发多语言的应用程序;利用智能客户端部署方案可以快速部署应用程序,应用程序可以跨城市、地域、国家。

Windows Forms(Windows 窗体)是一个新的窗体包,完全基于 .NET Framework,同时应用了许多新技术,能够开发强大的客户端应用程序。

本章主要介绍在 Visual Studio 2005 环境下创建 Windows 程序、Windows 事件处理机制和多文档界面设计。

1.1 Windows 窗体介绍

Windows 窗体是 .NET Framework 的一部分,它应用了许多新技术,包括通用应用程序框架、托管执行环境、集成安全性和面向对象的设计理念等。

窗体是为 Windows 操作系统所创建的应用程序用户界面的基本元素。它提供了一个框架,可以使应用程序呈现统一的外观和样式。在基于 Windows 的应用程序中,窗体向用户提供信息和获取用户输入。

窗体是应用程序的基础,每一个窗体都是一个类,可以为一个窗体创建多个实例或继承窗体。

窗体也是应用程序的基本单元。窗体实质上是一块空白板,开发人员可通过添加控件来创建用户界面,并通过编写代码来操作数据,从而填充这个空白板。Visual Studio 提供的 Windows 窗体设计器为创建基于 Windows 的应用程序提供了一种快速的开发解决方案,在该设计器中可以进行可视化的、基于客户端的窗体设计。

窗体提供了定义窗体外观的属性、定义行为的方法和定义与用户交互的事件,通过设置窗体的属性和编写代码来响应窗体的事件,可以定制满足应用程序需求的窗体。

窗体是从 Form 类派生的控件,而 Form 类是从 Control 类派生的,这种框架也便于继承现有的窗体来增加功能,或修改现有的行为。当为项目添加一个窗体时,可以选择从 .NET Framework 提供的 Form 类继承,或者从先前创建的窗体继承。

设计具有用户界面的应用程序时有两种选择:Windows 窗体和 Web 窗体。两者在开发环境中都具有完整的设计时支持,并且提供了丰富的用户界面和高级的应用程序功能用于解决业务问题。面对多种选择时,了解在什么情况下使用哪种选择是很重要的。

(1) Windows 窗体用于开发需要客户端承担大量处理任务的应用程序,包括典型的基于 Windows 应用程序接口的桌面应用程序,如绘图或图形应用程序、数据输入系统、销售点系统和游戏等都属于这类应用程序。这些应用程序都依赖桌面计算机的能力来进行信息的处理和高效的内容显示。

(2) ASP.NET Web 窗体用于创建主用户界面是浏览器的应用程序,公众可通过万维网使用它们,如电子商务应用程序。

1.2 创建 Windows 程序

在 Visual Studio 2005 中,窗体就是应用程序中使用的窗口。当创建一个新的 Windows 应用程序项目时,Visual Studio.NET 提供了一个默认窗体,包含一些基本元素:一个标题栏、一个控件框以及“最小化”、“最大化”和“关闭”按钮。

大多数应用程序需要不止一个窗口,必须为应用程序所需要的每一个窗口添加一个窗体。下面介绍创建 Windows 应用程序的步骤。

(1) 启动 Visual Studio.NET,选择菜单“文件”→“新建”→“项目”命令,打开“新建项目”对话框,如图 1.1 所示。



图 1.1 “新建项目”对话框

(2) 在“项目类型”中选择 Visual C# → Windows, 在“模板”中选择“Windows 应用程序”, 可以在“名称”文本框中输入所希望的应用程序名称, 也可以选择应用程序的保存位置。

(3) 单击“确定”按钮完成创建。此时的 Visual Studio.NET 开发环境如图 1.2 所示, 包括“工具箱”、“窗体设计器”、“解决方案资源管理器”和“属性”等窗口, 在“窗体设计器”中已经创建了一个默认窗体 Form1。

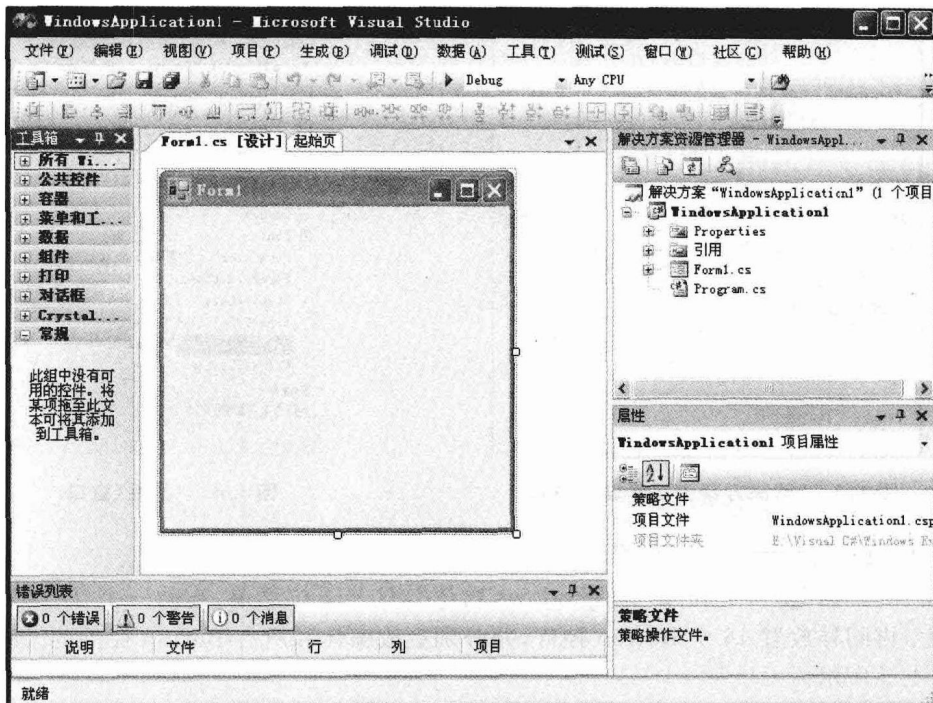


图 1.2 Visual Studio.NET 开发环境

(4) 在“解决方案资源管理器”窗口中, 可以将 Form1 改名。在 Form1.cs 上右击, 从快捷菜单中选择“重命名”命令, 如图 1.3 所示。

(5) 设置窗体属性。可以通过编写代码来设置窗体的属性; 也可以直接使用“属性”窗口来设置(如图 1.4 所示), 此时设置的任何属性都会作为每次应用程序运行的初始设置。

- 如果“属性”窗口没有打开, 则选择菜单“视图”→“属性窗口”命令将其打开。
- 在设计视图中, 单击想要设置其属性的窗体, 窗体的名称出现在“属性”窗口顶部的“对象”列表中。
- 在“属性”窗口中, 通过单击“按字母顺序”或“按分类顺序”按钮, 根据个人的习惯, 选择窗体属性的排列方式。
- 在“属性”窗口中, 单击想要设置的属性, 此时, 属性的描述便出现在“属性”窗口底部的说明窗格中。
- 输入或者选择想要的属性设置值。

下面通过一个简单的计算器界面设计, 来认识如何通过 C# 来创建 Windows 应用程序界面。

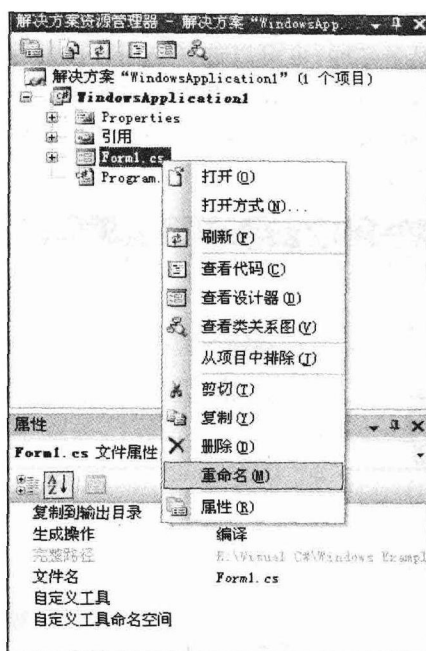


图 1.3 “解决方案资源管理器”窗口

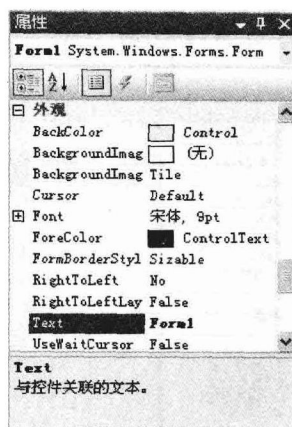


图 1.4 “属性”窗口

【例 1.1】 简易计算器界面设计。

单击“工具箱”窗口上的“所有 Windows 窗体”，打开下拉菜单，拖动一个 TextBox 控件到窗体上，再同样放置 16 个 Button 控件，并分别更改窗体和所有 Button 控件的 Text 属性值，如图 1.5 所示。



图 1.5 “简易计算器”设计界面

可以看到界面上的按钮摆放得很整齐，这是由于使用了“工具箱”中的 tableLayoutPanel 控件。使用此控件与在 Word 文档中添加一个表格很相似，只需从“工具箱”中拖放到窗体上，根据具体的需要设置行数和列数，然后根据设计好的布局摆放控件。

这样一个简易计算器的窗体就设计好了，但是它还不能完成任何功能。想要让这个计算器可以使用，必须要添加控件的事件处理程序。1.3 节将会介绍 Windows 应用程序的事件处理机制。

窗体默认继承自 System.Windows.Forms.Form，可以通过修改基类来创建继承窗体，并可设置 Modifiers 属性，使子窗体按照设计者的意愿继承窗体。

Modifiers 属性决定控件的可访问级别，例如，当一个控件的窗体作为基窗体时，该控件如何运行以及具有哪些功能。根据 Modifiers 属性值不同，继承窗体会以不同的灰色阴影显示控件。Modifiers 属性包括 public、private 和 protected 等属性值。

通过继承基窗体来创建新 Windows 窗体是利用已有的工作成果的一种高效方法，不必

每次需要窗体时都重新创建。这种从现有的窗体继承的过程称为可视化继承。

不仅可以从基窗体上继承控件和属性,还可以继承代码,这意味着可以创建代码库以增强代码重用性。使用继承的主要好处是当代码不适用于某种特定情况时,可以重写代码。要想在子窗体中重写一个事件,则该事件对于子窗体来说必须是可见的。它必须被定义为 `public` 或 `protected`,而且它的 `Modifiers` 关键字不能为 `private`。

子控件的属性值与其父对象的对应控件的属性值相同,当在子窗体上改变这些值时,在“属性”窗体上这些属性会变成粗体。要将所有属性的值重置为父类所持有的值,可以右击“属性”窗口空白处,然后选择快捷菜单“重置”命令。

1.3 事件处理

事件处理是 Windows 程序设计的基础,包括鼠标事件、键事件和焦点事件等。这些事件虽然看上去非常不同,但是它们有着相似的原理。

1. 事件处理基础

【例 1.2】 以按钮处理事件为例,从窗体控件、按钮控件和事件机制剖析 3 个方面来说明事件处理的功能和原理。

(1) 窗体控件

窗体(`Form`)是 Windows 程序的重要部分,即以窗口的形式显示用户界面。窗体是一个容器,上面可以放置其他的控件,如按钮。

按钮(`Button`)是可以被用户单击的控件,在 Windows 系统中常见的“确定”、“取消”和“应用”等按钮都是典型的实例。

新建一个 Windows 应用程序,在默认生成的窗体 `Form1` 的设计视图上右击,从快捷菜单中选择“查看代码”命令,打开“代码”视图,可以看到 Visual Studio .NET 已经自动生成了创建窗体的所有代码,其中,`Form1.cs` 文件的代码主要分为以下两部分。

- 命名空间。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

- 构造函数。

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

Program.cs 文件的代码如下：

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
namespace 第一个 Windows 程序
{
    static class Program
    {
        /// <summary>
        /// 应用程序的主入口点
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

C# 程序中的注释语句主要有 3 种：第 1 种是添加“//”，后面是注释的内容；第 2 种主要是针对注释内容多于 1 行的情况，添加“/* ... */”，中间是注释的内容；第 3 种就是例 1.2 程序中用到的，专门针对 C# 中的方法提供注释，在方法声明语句前添加“///”，系统会自动添加一个注释模板。如例 1.2 程序中的 Main() 方法上面的注释就是一个简单的例子，可以在<summary>和</summary>中间写注释，如“应用程序的主入口点”。

其中的 Main() 函数是应用程序的主入口点。Application.Run(new Form1()) 语句，创建并运行 Form1 实例。

在“属性”窗体中单击“事件”按钮（“闪电”图标），可以切换到“事件”列表。根据事件类型的不同，Form1 的事件被分为若干类，包括键事件、鼠标事件和焦点事件等。双击对应事件后面的空白，就可以添加对该事件的处理函数。

（2）按钮控件

从“工具箱”窗口中拖拽一个按钮控件到 Form1 上。找到按钮的“事件”视图，双击 Click 行的空白处添加 Click 事件。此时，“代码”窗口显示的是自动创建的 Click 事件处理函数。为 Click 事件处理函数添加处理代码，使得单击按钮时，打开一个信息提示框，此时 Form1.cs 文件的代码如下：

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace 第一个 Windows 程序
{
```