



全国高职高专教育“十一五”规划教材

WinForm 应用开发与实践教程

主 编 宋楚平 周建辉
副主编 沈学华 朱建东



高等教育出版社
HIGHER EDUCATION PRESS

计算机应用基础教程 (Windows XP+Office 2003)	贺 军	向子明
计算机应用基础实训教程	周 威	龚京民
IT职业素养	雷 瑛	
高等职业教育计算机英语	李志勇	
计算机数学基础	叶东毅	
C语言程序设计	冯茂岩	孙炯宁
Visual Basic程序设计语言入门与提高	吕日宝	邱晓华
Visual Basic程序设计语言入门与提高——实训教程与试题精解	吕日宝	刘鲁平
Visual FoxPro程序设计 (第二版)	李珍香	
Visual FoxPro程序设计实训指导 (第二版)	李珍香	
WinForm应用开发与实践教程	宋楚平	周建辉
数据库应用基础	蒋文蓉	
SQL Server 2005数据库技术及应用	赵俊荣	
网络互联通信技术基础教程	丁慧洁	付 勤
网页设计与制作——Photoshop版面构图+Dreamweaver网站实现	李瑞强	朱建东
Photoshop CS3图像处理任务式案例教程	张武卫	
多媒体技术基础	黄荣怀	

全国高职高专教育“十一五”规划教材

WinForm 应用开发 与实践教程

主 编 宋楚平 周建辉
副主编 沈学华 朱建东

高等教育出版社

内容简介

本书以开发能力为主线,采用典型案例引导、任务驱动的模式来编写。本书在内容的安排上遵循“实用、够用、拓展能力”的原则,从基本知识、应用技能出发,介绍了使用.NET 2.0 技术开发 WinForm 应用程序的基本知识、基本技能、编程技能、案例及项目开发经验。在内容的组织上,除了介绍 WinForm 编程基础、常用组件、文件处理、数据库访问与数据显示这些常见内容外,还重点介绍了其他教材中鲜见的绘图与打印、水晶报表、多线程和 Windows 服务、网络编程、单元测试和多层结构开发模式技术。

本书以案例和项目作为载体介绍 WinForm 应用程序设计与开发的主要内容,并且配以大量的图表说明、编码提示和编程方法介绍,使内容深入浅出、容易理解。由于这些案例、项目及编程方法来源于作者多年的实际开发经验,所以具有很强的实用性。为配合教学,本书每章后面都配有与该章内容紧密相关的实验及练习题,以提高学生的职业技能。

本书可作为应用性、技能型人才培养的 WinForm 或 C# 高级开发的课程教材,也可供.NET 开发人员和程序设计爱好者参考使用。

图书在版编目(CIP)数据

WinForm 应用开发与实践教程/宋楚平,周建辉主编.

北京:高等教育出版社,2009.8

ISBN 978-7-04-027816-3

I. W… II. ①宋…②周… III. 窗口软件,Windows—程序设计—教材 IV. TP316.7

中国版本图书馆 CIP 数据核字(2009)第 133563 号

责任编辑	张朶林	特约编辑	韩东娜	封面设计	顾凌芝	责任印制	蔡敏燕
出版发行	高等教育出版社	购书热线	010-58581118				
社 址	北京市西城区德外大街 4 号		021-56717287				
邮政编码	100011	免费咨询	400-810-0598				
总 机	010-58581000	网 址	http://www.hep.edu.cn				
传 真	021-56965341		http://www.hep.com.cn				
			http://www.hepsh.com				
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landracom.com				
排 版	南京理工出版信息技术有限公司		http://www.landracom.com.cn				
印 刷	江苏如皋市印刷有限公司	畅想教育	http://www.widedu.com				
开 本	787×1092 1/16	版 次	2009 年 8 月第 1 版				
印 张	19.25	印 次	2009 年 8 月第 1 次				
字 数	445 000	定 价	26.50 元				

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

物料号 27816-00

案例引领和项目驱动教学法是目前应用型本科和高职院校比较流行的教学方法。在当前教学改革、课程开发的背景下,编写一本贯彻工学结合思想,采用典型案例和工作任务来承载教学内容的教材就变得迫切而具有现实意义。

本书的编写思路如下:在内容选取上充分考虑工学结合的要求,重点选择在 WinForm 应用程序开发中广泛使用的技术,强调案例和任务的典型性;摒弃或者简单介绍那些相对比较落后或者用得比较少少的技术。在内容的编排上做到循序渐进、由浅入深,既考虑 WinForm 应用程序的开发过程和步骤,同时也考虑学生的认知规律。

本书共 11 章,可分为 5 个单元。

第 1 单元:第 1~3 章,主要讲述 WinForm 编程基础和开发过程中常用的组件和文件处理等内容。

第 2 单元:第 4、5 章,主要讲述访问数据库技术以及利用控件显示和操作数据。

第 3 单元:第 6、7 章,主要讲述绘图技术以及报表预览和打印。

第 4 单元:第 8、9 章,主要讲述多线程与 Windows 服务技术、网络编程。

第 5 单元:第 10、11 章,主要讲述多层结构的应用系统开发技术以及应用程序的单元测试和发布。

本书的主要特色体现在以下几个方面:

1. 以案例和工作任务承载教学内容。

本书所选取的案例和工作任务均来源于已投入使用的实际开发的 WinForm 应用程序,是在开发过程中经常碰到的,具有典型性和教学价值。在介绍相关的基本概念、基础知识和必要的理论后,紧接着就配以一个涵盖概念、知识和理论,但又高于概念、知识和理论的案例,在案例和实例中体现设计理念、编程方法和实现步骤。

2. 具有先进性和实用性。

本书的水晶报表、多线程和 Windows 服务、网络编程、多层结构的应用系统开发、单元测试和应用程序发布等内容在一般书中鲜有介绍,而本书紧密结合实际,在深度上基本满足实际岗位的能力要求,同时这些技术也是目前软件行业广泛使用和正在流行的技术。通过本书的学习,可为读者将来适应工作岗位打下一个良好的基础。

3. 增强学生的编程体验,注重学生的职业技能和动手能力的培养。

学生的应用程序开发技能在很大程度上要通过上机的反复练习和实际操作来掌握。因此,本书各章节中,每个案例、每个具体的工作过程都尽可能描述得详细清楚,以便学生容易理解和操作,同时配备了适量的上机练习题,使学生能学中做、做中学,不断提高实践能力。在每章结束处还配有一个综合性较强的实验,对这些实验的实验目的、实验内容、实验步骤以及重点难点都做了必要的文字说明和代码提示,在老师的指导下学生一般能独立完成,可增强学生的成就感。

学习本书的读者应该已学习过 C# 语言程序设计基础。建议本书的教学学时为 64 学



时,学时分配如下表所示,各校可以根据学生实际情况进行必要的调整。

参考学时分配表

序 号	授 课 内 容	学时分配	
		理论	实践
1	WinForm 编程基础、常用组件、文件处理	6	6
2	访问数据库、利用控件显示和操作数据	8	8
3	绘图和打印、水晶报表	6	6
4	多线程和 Windows 服务、网络编程	6	6
5	开发多层结构的应用系统、应用程序的单元测试和发布	6	6
合 计		32	32

为方便教学,本书配套有相关教学资源,教师可填写书后所附的《教学资源索取单》,依照相关方式索取。

本书由宋楚平、周建辉任主编,沈学华、朱建东任副主编,宋楚平编写了第 4~6 章和第 10 章,周建辉编写了第 1~3 章,沈学华编写了第 8 章和第 9 章,朱建东编写了第 7 章和第 11 章,最后由宋楚平统稿。

在本书初稿的使用和编写过程中,部分老师和学生提出了许多宝贵的意见,在此表示衷心的感谢。

由于时间仓促,加之作者水平有限,书中难免有错漏之处,敬请读者批评指正。编者的 E-mail 为:ntscp@sina.com。欢迎联系和交流。

编 者

2009 年 4 月

CONTENTS

第1章 WinForm 编程基础 1

1.1	Windows 窗体简介	1
1.1.1	创建一个简单的 Windows 应用程序	2
1.1.2	设置窗体属性	7
1.1.3	窗体的常用方法和事件	9
1.1.4	一个简单的启动窗口	10
1.2	多文档应用程序	12
1.2.1	创建 MDI 应用程序	13
1.2.2	子窗体的激活和排列	14
1.2.3	多文档记事本	16
1.3	多窗体应用程序	19
1.3.1	窗体之间的连接	19
1.3.2	消息框窗口	19
1.3.3	窗体之间的数据传递	21
	实验	25
	思考与练习	29

第2章 常用组件 30

2.1	组件的基本概念	30
2.2	基本控件	31
2.2.1	显示控件	31
2.2.2	输入控件	32
2.2.3	按钮控件	35
2.2.4	选择控件	38
2.2.5	图形控件	40
2.2.6	进度控件	41
2.2.7	录入学生信息示例	42
2.3	容器控件	44



2.3.1	布局控件	45
2.3.2	分组控件	46
2.3.3	选项卡控件	47
2.3.4	班级管理示例	48
2.4	菜单和工具栏	49
2.4.1	主菜单	50
2.4.2	快捷菜单	51
2.4.3	工具栏	52
2.4.4	状态栏	52
2.4.5	学生管理系统主窗体示例	53
2.5	对话框控件	55
2.5.1	文件对话框	56
2.5.2	颜色对话框	61
2.5.3	字体对话框	61
2.5.4	简易写字板应用程序	63
2.6	其他高级组件	64
2.6.1	监视文件系统	65
2.6.2	图像列表	66
2.6.3	进程控制	67
2.6.4	定时器	69
实验		72
思考与练习		76
 第3章 文件处理		78
3.1	文件系统的基本操作	78
3.1.1	文件及流的基本概念	78
3.1.2	目录的操作	79
3.1.3	文件的操作	80
3.1.4	列举目录下的文件	82
3.2	读写文本文件	84
3.2.1	使用 FileStream 读写文本文件	84
3.2.2	使用 StreamReader 读取文本数据	87
3.2.3	使用 StreamWriter 写入文本数据	89
3.3	读写二进制文件	91
3.3.1	读取二进制数据	91
3.3.2	写入二进制数据	92

3.3.3 读写课程基本参数	93
3.4 读写 ini 文件	95
3.4.1 ini 文件介绍	95
3.4.2 读写 ini 文件示例	96
实验	99
思考与练习	100

第4章 访问数据库 102

4.1 ADO.NET 概述	102
4.1.1 ADO.NET 的结构	102
4.1.2 ADO.NET 的优势	104
4.1.3 ADO.NET 的基本组件	104
4.2 数据的增删改查	107
4.2.1 建立数据库连接	107
4.2.2 检索数据	108
4.2.3 增加、删除和更改数据	110
4.3 数据集	113
4.3.1 创建数据集	113
4.3.2 数据表、数据列和数据行	113
4.3.3 利用数据集批量更新数据	116
4.3.4 DataView	118
4.4 数据组件和数据绑定	120
4.4.1 数据组件	120
4.4.2 简单的数据绑定	124
4.4.3 复杂的数据绑定	124
4.5 使用存储过程	126
4.5.1 返回记录的存储过程	127
4.5.2 返回值的存储过程	128
4.5.3 行为存储过程	129
4.5.4 存取大对象数据	131
4.6 使用事务处理	135
4.6.1 在数据库级使用事务处理	135
4.6.2 在 ADO.NET 级使用事务处理	137
实验	138
思考与练习	143



第 5 章 利用控件显示和操作数据 144

5.1 利用 TreeView 显示数据	144
5.2 利用 ListView 显示数据	147
5.3 利用 ReportViewer 显示数据	149
5.4 利用 DataGridView 显示和操作数据	154
5.4.1 在 DataGridView 中显示数据	155
5.4.2 在 DataGridView 中操作数据	156
实验	159
思考与练习	164

第 6 章 绘图和打印 166

6.1 图形设备接口简介	166
6.2 绘制基本图形	167
6.2.1 绘制文本和直线	167
6.2.2 绘制矩形和椭圆	168
6.2.3 绘制曲线	169
6.2.4 翻转、缩放和移动图形	170
6.3 自定义打印和预览	173
6.3.1 打印自定义票据	174
6.3.2 打印自定义报表	179
实验	182
思考与练习	188

第 7 章 水晶报表 189

7.1 水晶报表简介	189
7.2 创建水晶报表	190
7.2.1 拉模型创建水晶报表	190
7.2.2 推模型创建水晶报表	198
7.3 预览水晶报表	200
7.4 水晶报表的高级应用	203
实验	208
思考与练习	212

第 8 章 多线程和 Windows 服务 214

8.1 多线程	214
8.1.1 线程类的属性和方法	214
8.1.2 创建线程	215
8.1.3 线程的调度	217
8.1.4 线程同步	219
8.2 Windows 服务	221
8.2.1 创建服务	223
8.2.2 安装和卸载 Windows 服务	224
8.2.3 调试 Windows 服务	227
实验	228
思考与练习	231

第 9 章 网络编程 233

9.1 Internet 访问	233
9.1.1 利用 WebClient 类访问 Internet	233
9.1.2 利用 WebRequest 类和 WebResponse 类 访问 Internet	236
9.2 套接字编程	239
9.2.1 TCP 协议类简介	239
9.2.2 建立 TCP 服务器和客户端	241
实验	246
思考与练习	252

第 10 章 开发多层结构的应用系统 254

10.1 系统的功能需求分析	254
10.1.1 项目背景说明	254
10.1.2 业务流程	255
10.1.3 功能描述	255
10.2 数据库设计	256
10.2.1 系统的 ER 图	256
10.2.2 系统的关系表	257



10.3 多层结构的应用系统设计	258
10.3.1 多层结构的概念及优势	258
10.3.2 搭建多层结构的开发框架	260
10.3.3 实现实体层	263
10.3.4 实现数据访问层	264
10.3.5 实现业务逻辑层	268
10.3.6 实现表示层	269
实验	272
思考与练习	279
第 11 章 应用程序的单元测试和发布	280
11.1 程序的单元测试	280
11.1.1 单元测试的基本概念	280
11.1.2 单元测试类	281
11.1.3 创建单元测试项目	283
11.1.4 运行单元测试项目	287
11.2 应用程序的发布	289
11.2.1 使用 ClickOnce 技术发布应用程序	289
11.2.2 使用 Windows Installer 技术发布应用 程序	291
实验	294
思考与练习	295
主要参考文献	296

第 1 章 WinForm 编程基础

本章要点

本章是 .NET Framework WinForm 编程的基础,主要介绍 Windows 窗体、多文档应用程序和多窗体应用程序的基本概念和设计基础。WinForm 中的属性、方法和事件是构成 WinForm 应用程序的基础。在 Visual Studio 2005 中可以创建多种样式的窗体,如普通窗体、透明变形窗体以及各种类型的应用程序,如简单的应用程序、多文档应用程序和多窗体应用程序等。

通过本章的学习,读者应该能够:

- 了解 WinForm 的编程模型,熟练窗体的属性、方法和事件的使用
- 了解多文档应用程序的概念,能够创建多文档应用程序
- 掌握创建多窗体应用程序的方法,能合理有效地在窗体间传递数据

1.1 Windows 窗体简介

经常使用 Windows 操作系统的用户,每天都会接触大量 Windows 窗体。Windows 窗体又称为 Windows 窗口,一个标准的 Windows 窗口包括 6 个部分:标题栏、菜单栏、工具栏、客户区、状态栏和边框。其中客户区的主要作用是容纳各种各样的控件。图 1.1 所示的是 Windows 系统中写字板程序的主界面,它是一个标准的 Windows 窗口,其中箭头指向的区域为窗口的各个组成部分。

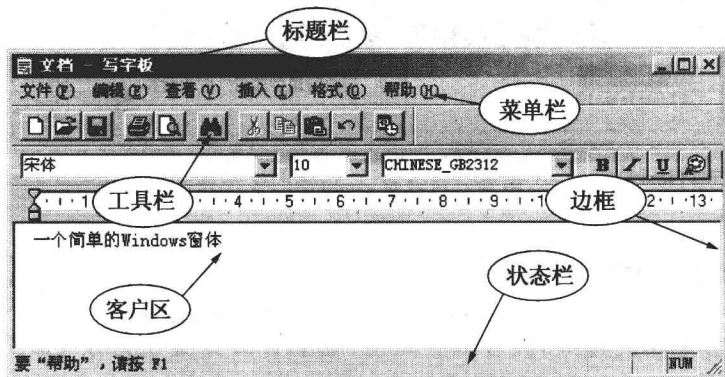


图 1.1 写字板程序的主界面

Windows 应用程序的窗口在实际应用中也有很多变化,例如有些窗口没有菜单栏或者工具栏,有时甚至没有标题栏;有些窗口的客户区并不像图 1.1 那样只容纳少量控件(文本框控件);有些窗口甚至使用了透明、变形等效果,从而使得窗体变得丰富多样。



总之,构成 Windows 应用程序 GUI(图形用户界面)的基本元素就是窗体和控件。对于 C#.NET 程序设计人员而言,要构造不同的窗口应用程序,通常不需要自己再独立编写这些界面元素,.NET 框架 2.0 提供了 Windows 窗体以及窗体内常用的一些控件,以供开发人员直接使用,从而高效地开发应用程序。

1.1.1 创建一个简单的 Windows 应用程序

创建一个 C# 的 Windows 应用程序的基本过程为:

- ① 创建项目。
- ② 添加控件及设置控件属性。
- ③ 编写事件处理代码。
- ④ 调试运行程序。
- ⑤ 保存程序。

以下是创建一个简单的 Windows 应用程序(除自动产生的代码外,不编写任何代码)的详细步骤:

① 启动 Microsoft Visual Studio 2005,此时显示的是 Visual Studio 2005 应用程序主界面窗口,如图 1.2 所示。

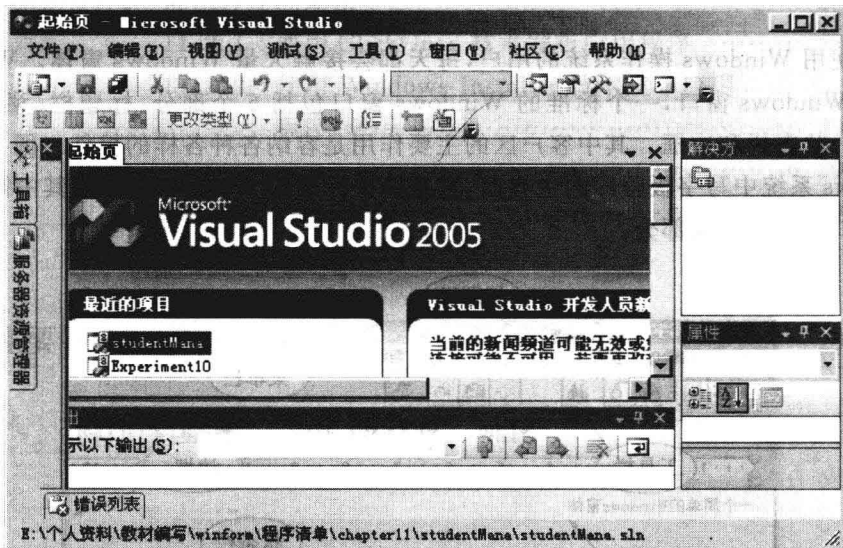


图 1.2 Visual Studio 2005 应用程序主界面窗口

② 单击“文件”→“新建”→“项目”菜单项,此时弹出“新建项目”对话框,如图 1.3 所示。

③ 在图 1.3 中的“项目类型”视图中选中 Visual C#,展开后再选中 Windows(如果默认设置已经是上述选择,则省去前面的两个选择),然后在“模板”区域中选中“Windows 应用程序”,并把该应用程序命名为 chapter1,指定其存放的位置和解决方案的名称,然后单击“确定”按钮,此时就会显示 Windows 项目设计视图,如图 1.4 所示。

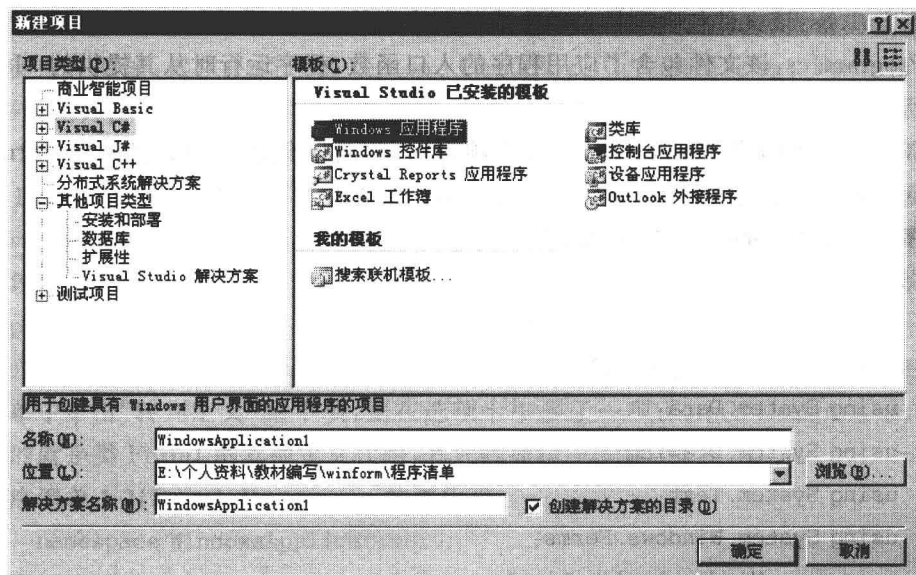


图 1.3 “新建项目”对话框

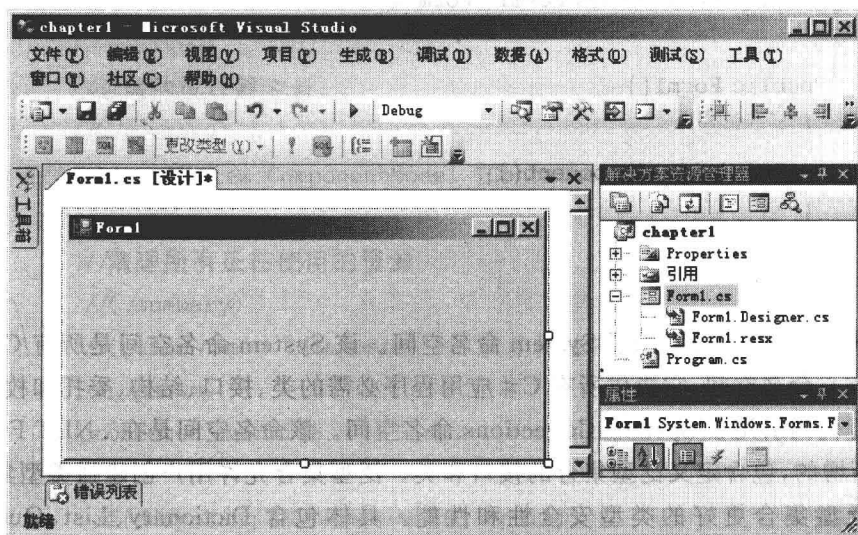


图 1.4 Windows 项目设计视图

要修改项目的默认存储位置,可以单击“工具”→“选项”菜单项,在弹出的对话框中重新设定,重新启动 Visual Studio 2005 后,该设置生效。

从图 1.4 所示的“解决方案资源管理器”窗口可以看到,该解决方案包含了一个项目 chapter1(解决方案是针对一个开发需求而提出的项目设计计划,一个解决方案中可以包含多个项目)。该项目由以下几个部分组成:

- Properties:项目属性,它包含了项目的应用程序、生成、生成事件、调试和资源等相关信息。

- 引用:主要包括该项目引用的一些默认系统命名空间。

➤ Form1.cs:该文件包含项目自动生成的一个窗体的定义代码。

➤ Program.cs:该文件包含了应用程序的入口函数,程序运行时从其提供的 Main 函数开始执行。

用 Visual C# 创建 Windows 应用程序项目时,将自动创建一个类名称为 Form1 的窗体。Visual Studio 2005 为该窗体生成的代码如程序清单 1-1 所示:

程序清单 1-1(Form1.cs 的类代码)

```
1: using System;
2: using System.Collections.Generic;
3: using System.ComponentModel;
4: using System.Data;
5: using System.Drawing;
6: using System.Text;
7: using System.Windows.Forms;
8: namespace WindowsApplication1
9: {
10:     public partial class Form1: Form
11:     {
12:         public Form1()
13:         {
14:             InitializeComponent();
15:         }
16:     }
17: }
```

上述代码中,第 1 行引用了 System 命名空间。该 System 命名空间是所有 C# 项目使用的基础核心命名空间,它提供所有 C# 应用程序必需的类、接口、结构、委托和枚举。

第 2 行代码引用了 System.Collections 命名空间。该命名空间是在 .NET Framework 2.0 版中新增的,包含定义泛型集合的接口和类。泛型集合允许用户创建强类型集合,并能提供比非泛型集合更好的类型安全性和性能。具体包含 Dictionary、List、Queue、SortedList 等及相关的接口和结构。

第 3 行引用了 System.ComponentModel 命名空间。该命名空间支持以下类:ArrayConverter、ByteConverter、DateTimeConverter、Int16Converter、Int32Converter 和 Int64Converter 等。它对于大量事件处理程序委托也提供了委托支持。

第 4 行引用了 System.Data 命名空间。该命名空间主要提供了使用了 ADO.NET 技术访问数据库的类。在第 4 章和第 5 章将对这些类进行详细探讨。

第 5 行引用了 System.Drawing 命名空间。该命名空间提供了使用大量绘图工具的途径,例如 Brushes、Font 和 Pens 等类,命名空间也包括 Color、Point 和 Rectangle 等结构。

第 6 行引用了 System.Text 命名空间。该命名空间包含表示 ASCII、Unicode、UTF-7 和 UTF-8 字符编码的类。

第7行引用了 System.Windows.Forms 命名空间。该命名空间对大量窗体和控件提供了类支持。例如,提供的类包括 Button、CheckBox、CommonDialog、Form 和 ListBox 等。此外,还为窗体和控件提供了委托支持,并包括窗体和控件的样式、状态的枚举等。

第10~16行定义了一个名为 Form1 的项目的基本类。每个应用程序至少使用一个窗体。类 Form1 派生于 System.Windows.Forms 命名空间。所有的定义均包含在命名空间 WindowsApplication1 中,但分布在两个不同的文件中。第10行中的关键字 partial 意味着此处是该类定义的一部分。第12~15行是 Form1 类的构造函数,其中调用了方法 InitializeComponent()。

将光标定位在第14行代码的 InitializeComponent()方法上,单击鼠标右键,在弹出的快捷菜单中单击“转到定义”菜单项,则在选项卡中多了一项 Form1.Designer.cs。该选项卡包含的就是类 Form1 的其他定义代码,其代码如程序清单 1-2 所示:

程序清单 1-2 (Form1.Designer.cs 的类代码)

```

1: namespace WindowsApplication1
2: {
3:     partial class Form1
4:     {
5:         ///

```