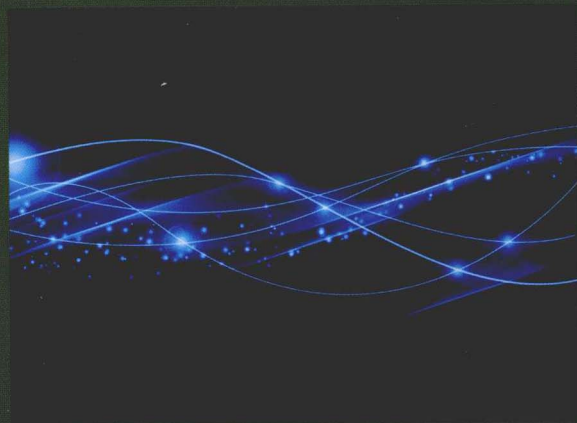




可视化程序设计与 C#.NET语言

何明昌 张笑钦 高利新 编著



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

可视化程序设计与 C#.NET 语言

何明昌 张笑钦 高利新 编著

上海交通大学出版社

内 容 提 要

Visual C#.NET 是面向对象的可视化程序设计语言,具有简洁、高效、安全、性能优良等特点,是面向对象程序设计、可视化程序设计教学的主干语言之一。全书共分 14 章,全面阐述了程序设计的基本思想、结构化程序设计方法、面向对象的程序设计概念、可视化程序设计方法与原理、控件窗体与界面设计技术、图形编程、数据库程序设计和 Web 程序设计、文件操作等内容。本书的所有程序都在 Windows 7 平台和 Visual Studio.NET2008 环境下调试通过,在 Studio.NET2010 下也可以正常运行,另外提供了 15 个实验和一套练习集。

本书从第一门高级编程语言的角度出发,让学生全面了解程序设计思想及可视化编程技术,全书理论分析透彻严谨,实例丰富生动,内容由浅入深,能快速引导学生进入 Visual C#.NET 编程世界。本书可作为高等院校相关专业的程序设计类课程教材,也可作为初学 Visual C#.NET 编程人员的参考用书。

图书在版编目(CIP)数据

可视化程序设计与 C#.NET 语言/何明昌,张笑钦,高利新编著. —上海:上海交通大学出版社,2012
ISBN 978-7-313-09384-4

I. ①可… II. ①何…②张…③高… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 316765 号

可视化程序设计与 C#.NET 语言

何明昌 张笑钦 高利新 编著

上海交通大学 出版社出版发行

(上海市番禺路 951 号 邮政编码 200030)

电话: 64071208 出版人: 韩建民

浙江云广印业有限公司印刷 全国新华书店经销

开本: 787 mm×1092 mm 1/16 印张: 17.25 字数: 413 千字

2012 年 12 月第 1 版 2012 年 12 月第 1 次印刷

ISBN 978-7-313-09384-4/TP 定价: 49.00 元

版权所有 侵权必究

告读者:如发现本书有印装质量问题请与印刷厂质量科联系
联系电话: 0573-86577317

Contents

目 录

1	Visual C#.NET 简介及集成开发环境	1
1.1	Visual Studio.NET 简介	1
1.2	Visual Studio.NET 开发平台的组成	2
1.3	创建应用程序	3
1.4	C# 集成开发环境	7
1.5	Visual Studio2010 简介	15
1.6	习题	17
2	C# 语言基础知识	18
2.1	C# 软件项目的组成	18
2.2	C# 程序结构	19
2.3	C# 基本数据类型	21
2.4	变量和常量	24
2.5	运算符与表达式	26
2.6	文本框、按钮对象相关操作	31
2.7	习题	35
3	顺序结构与选择结构程序设计	37
3.1	赋值语句	37
3.2	输入与输出	39
3.3	图片框与图片列表控件	42
3.4	选择结构	45
3.5	分组框与 Panel 控件	50
3.6	单选按钮与复选框控件	51

3.7	常用数学函数方法	53
3.8	习题	53
4	循环结构程序设计	55
4.1	for 循环语句	55
4.2	while 循环语句	58
4.3	do... while 语句	59
4.4	三种循环控制语句的区别	60
4.5	循环的嵌套	61
4.6	break 与 continue 语句	63
4.7	列表框与组合框	65
4.8	计时(Timer)控件	68
4.9	习题	70
5	数组	72
5.1	数组概述	72
5.2	数组与数组元素	72
5.3	一维数组	72
5.4	多维数组	76
5.5	foreach 语句	79
5.6	控件数组	80
5.7	字符串与字符	81
5.8	数组与数组列表属性	87
5.9	鼠标和键盘事件	88
5.10	习题	91
6	方法	93
6.1	方法概述	93
6.2	方法的声明及定义	93
6.3	方法的调用	95
6.4	参数的传递	96
6.5	变量的生命期与作用域	99
6.6	方法的递归调用	101
6.7	数组作为方法参数	103
6.8	参数数组	106
6.9	习题	107

7	面向对象程序设计	108
7.1	面向对象概述	108
7.2	类与对象的声明	109
7.3	类的字段与属性	113
7.4	类的方法及方法重载	116
7.5	静态成员	119
7.6	构造函数	121
7.7	类的继承	122
7.8	多态性	123
7.9	C#.NET 常用数据类型转换方法	125
7.10	习题	126
8	Windows 高级界面设计	128
8.1	主菜单的设计	129
8.2	RichTextBox 控件	131
8.3	弹出式菜单的设计	134
8.4	工具栏设计	135
8.5	状态栏设计	137
8.6	通用对话框	139
8.7	消息框	142
8.8	多选项卡	145
8.9	习题	146
9	数据库应用	147
9.1	数据库的基本概念	147
9.2	Access 数据库操作	148
9.3	SQL 语言	151
9.4	SQL Server 2005 简介	152
9.5	ADO.NET 概述	154
9.6	利用 ADO.NET 控件编写数据库应用程序	155
9.7	利用 ADO.NET 对象编程	156
9.8	习题	165
10	多媒体应用程序开发	166
10.1	多媒体概念	166
10.2	Animation 控件	167

10.3	Windows Media Player 控件	168
10.4	ShockwaveFlash 控件	170
10.5	WebBrowser 控件	172
10.6	习题	174

11

Web 应用程序开发	175
11.1 ASP.NET 简介	175
11.2 ASP.NET 应用程序的组成	176
11.3 ASP.NET 常用服务器控件	180
11.4 ASP.NET 常用内置对象	188
11.5 Web 服务的创建和使用	195
11.6 习题	201

12

C#.NET 图形编程	202
12.1 GDI+绘图基础	202
12.2 Graphics 对象	203
12.3 Pen 及 Brush 对象	204
12.4 常用图形的绘制方法	206
12.5 习题	214

13

文件	215
13.1 文件的相关概念	215
13.2 文件夹管理	216
13.3 File 类	220
13.4 用 FileStream 读写文件	224
13.5 用 BinaryReader、BinaryWriter 类读写数据	228
13.6 用 StreamReader 和 StreamWriter 类读写字符串	229
13.7 习题	231

14

实验设计	232
14.1 实验目的与要求	232
14.2 实验计划	233
实验 1 Visual C#.NET 及窗体设计	233
实验 2 基本数据类型	234
实验 3 运算符与表达式	235
实验 4 选择结构应用	236

实验 5 复选框、组合框、switch 语句等使用	237
实验 6 循环结构应用	238
实验 7 餐厅点菜系统	239
实验 8 面向对象程序设计	239
实验 9 Windows 应用程序设计	240
实验 10 Windows 应用程序设计 2	241
实验 11 Windows 高级界面设计	242
实验 12 对话框使用及图形绘制	242
实验 13 数据库操作	243
实验 14 多媒体应用程序设计	244
实验 15 Web 应用程序设计	245
附录 1 综合练习题	246
附录 2 C# 关键字	262
参考文献	265

Visual C#.NET 简介及集成开发环境

Visual C#.NET 是微软公司近年来主推的可视化开发工具 Visual Studio.NET 平台中的主流产品,是目前在视窗产品开发、网络产品开发、手机产品开发最重要的编程语言之一,也是最简便最快捷的开发工具之一。Visual Studio.NET 平台是微软公司开发的建立在开放互联网络协议标准之上,采用新的工具和服务来满足人们的计算和通信需求的革命性的新型 XML Web 智能计算服务平台,通过该平台,程序员可以用 C#.NET、VB、C++ 等不同的语言开发控制台应用程序、Windows 应用程序、网络程序、手机及智能化程序。

1.1 Visual Studio.NET 简介

2000 年,微软向全球宣布其革命性的软件和服务平台 Microsoft.NET 平台,通过先进的软件技术和众多的智能设备,提供更简单、更个性化、更有效的互联网服务。

Microsoft.NET 对于用户来说非常重要,因为计算机功能将会得到大幅度的提升,同时计算机的操作也会变得简单。更重要的是,用户将完全摆脱硬件束缚,可以自由冲浪于因特网中,自由访问,自由查看,自由使用自己的数据,而不是束缚在 PC 的分寸空间。用户可以通过任何桌面系统,任何便携式计算机,任何移动电话或者 PDA 访问数据及资源。.NET 的战略目标就是在任何时候(when)、任何地方(where)、任何工具(what)都能通过.NET 的服务获得网络上的任何信息,享受网络带给人们的便捷与快乐。

通俗地说,.NET 是一个连接系统、人员、设备和信息的平台。从某种意义上,可以将其看作是通往协同工作的一种捷径。也许可以这样理解,它是一个打破不同系统之间的障碍、打破信任障碍、打破人与人的沟通障碍、打破知识分享的障碍、打破日常使用的障碍,从而创造彼此联系的用户体验、改变人们生活和工作方式的平台。

Microsoft.NET 开发平台包括.NET 框架和.NET 开发工具等组成部分。

Microsoft.NET 框架是整个开发平台的基础,包括公共语言运行库(Common Language Runtime, CLR)和框架类库。

1.2 Visual Studio.NET 开发平台的组成

Visual Studio.NET 是 .NET 平台下最为强大的开发工具,无论是软件服务商,还是企业应用程序的部署与发布,Visual Studio.NET 都可以提供近乎完美的解决方案。Visual Studio.NET 提供了包括设计、编码、编译调试、数据库连接操作等基本功能和基于开放架构的服务器组件开发平台、企业开发工具和应用程序重新发布工具以及性能评测报告等高级功能。Visual Studio.NET 自从微软公司 2000 年宣布开发以来,其推出的版本有 Visual Studio.NET2003, Visual Studio.NET2005, Visual Studio.NET2008, Visual Studio.NET2010。后续的版本不断改正前面版本的一些错误,并增加了一些功能,性能上也不断提升。本书编译环境以 Visual Studio.NET2008(简称 VS2008)为主,书中所有的代码均可以在 Visual Studio.NET2008 以及 Visual Studio.NET2010 下运行。

Visual Studio.NET 是一套完整的开发工具,用于生成 ASP Web 应用程序、XML Web services、桌面应用程序和移动应用程序。在该集成开发环境下可以用 Visual Basic.NET、Visual C++.NET、Visual C#.NET 和 Visual J#.NET 语言单独编程与混合编程。另外,这些语言利用了 .NET Framework 的功能,此框架提供对简化 ASP Web 应用程序和 XML Web services 开发的关键技术的访问。.NET 开发工具包括 Visual Studio.NET 集成开发环境和 .NET 编程语言。它们的关系如图 1.2.1 所示:

VB	C#	托管 C++	J#	其他语言		
公共语言规范(CLS)						
ASP. NET/Web 应用/Web 服务			Windows 窗体应用			
ADO. NET 与 XML						
. NET 框架基础类库						
公共语言运行库						
操作系统						

图 1.2.1 .NET 关系图

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类型集合。可使用 .NET Framework 开发下列类型的应用程序和服务:

- (1) 控制台应用程序及 Windows 服务。
- (2) Windows GUI 应用程序(Windows 窗体)。
- (3) ASP.NET 应用程序,XML Web services。

C#.NET 是从 C、C++ 和 Java 发展而来,它采用了这三种语言最优秀的特点,并加入了自己的特性。C#.NET 是事件驱动的,完全面向对象的可视化编程语言,我们可以使用集成开发环境来编写 C# 程序,程序员可以非常方便地建立、运行、测试和调试 C# 程序。

C#.NET 是一种全新的简单、安全、面向对象的程序设计语言。它是专门为 .NET 的应用而开发的语言。它吸收了 C++、Visual Basic、Delphic、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。其主要特点有：

- 语言简洁,并保留了 C++ 的强大功能。
- 语言的自由性,具备快速应用开发功能。
- 强大的 Web 服务器控件。
- 支持跨平台,与 XML 相融合。

1.3 创建应用程序

在 Visual Studio.NET 中,可以建立控制台应用程序、Windows 应用程序、ASP.NET 动态网站及智能设备(手机等)应用程序。由于 C#.NET 语言必须在 .NET 中运行,因此, C#.NET 也常简称 C#,在本书中 C#.NET 有时也简称 C#,后面章节的叙述中不再进行区分。

1.3.1 创建控制台应用程序

在控制台应用程序下,输入输出都是纯文字信息,这种方式是传统 C 语言的工作方式。下面我们通过实例来创建一个最简单的控制台应用程序。

例 1.1 建立一个控制台应用程序,程序运行后首先让用户通过键盘输入自己的名字,然后程序在屏幕上输出一条欢迎信息。

设计步骤如下：

(1) 启动 VS2008 程序,单击菜单“文件(F)|新建项目(P)...”菜单项,打开“新建项目”,选择编程语言为 Visual C#,出现对话框如图 1.3.1 所示：

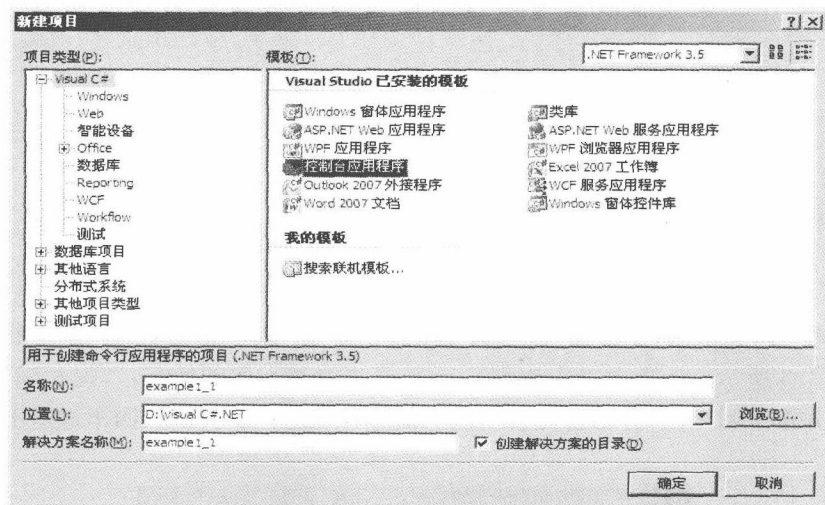


图 1.3.1 建立控制台应用程序

(2) 在“模板(T)”列表框中选择“控制台应用程序”,在“名称(N)”编辑框中键入 example1_1,在“位置(L)”中选择该项目代码存放的路径,单击“确定”按钮,创建项目,出现如图 1.3.2 界面：

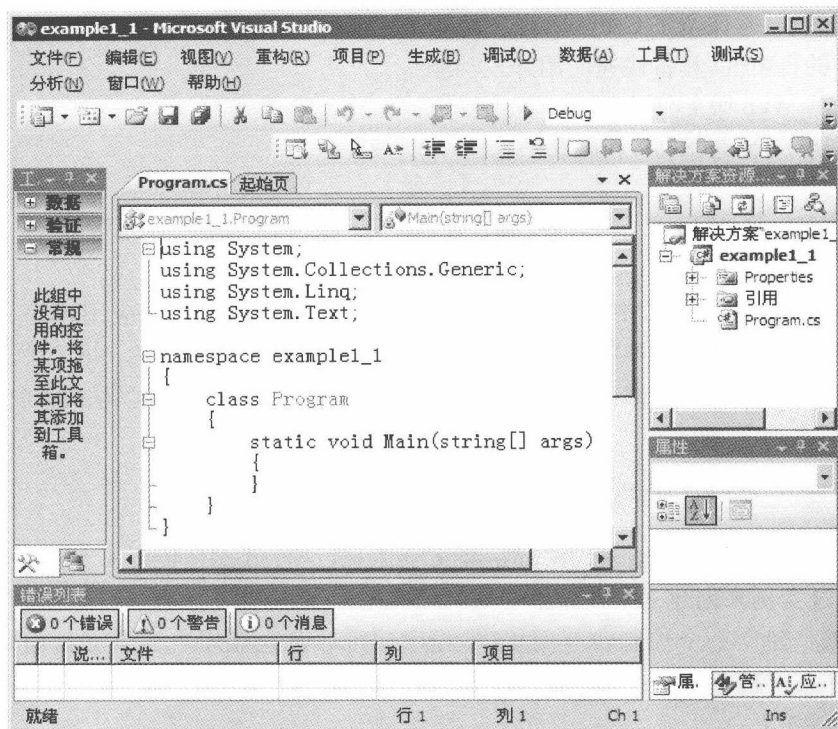


图 1.3.2 控制台应用程序

(3) 在编写代码工作区的 `static void Main(string[] args)` 一行下面的大括号内输入如下代码：

```
{
    Console.WriteLine("请输入你的姓名:");
    Console.ReadLine();
    Console.WriteLine("欢迎你!");
}
```

(4) 代码输入完毕后,点击工具栏上的运行按钮“▶”或 CTRL+F5 键,运行程序,运行结果如图 1.3.3 所示。屏幕上首先出现一行字符:“请键入你的姓名:”,提示输入姓名。输入任意字符并按下回车键,屏幕将打印出欢迎信息:“欢迎你!”。键入回车键退出程序。

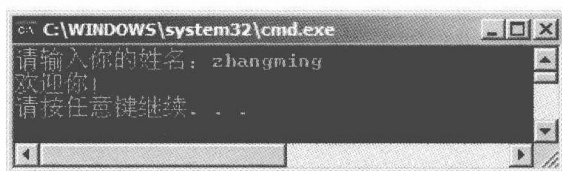


图 1.3.3 控制台应用程序运行界面

上面代码中 `Console.ReadLine()` 表示要输入一行信息,而 `Console.WriteLine("欢迎你!")` 则表示要输出“欢迎你!”这一行信息。

(5) 保存项目。

只要运行了该项目,则该项目的所有文件会自动保存,如果在编程过程中要保存项目,可以选择菜单中的“文件”下的“全部保存”,系统把正在编辑的项目中的所有文件都保存到相应的文件夹下。该项目下主要的文件夹和文件有下面几个:

bin 文件夹:该文件夹下包含 `debug` 子文件夹,含有与项目名对应的可执行文件 `example1_1.exe`,该文件可单独执行,另外该文件夹下还包含一个 `.pdb` 的文件,主要包含完整的调试信息。

obj 文件夹:该文件夹下也包含一个 `debug` 子文件夹,含有编译过程中生成的中间代码。

Properties 文件夹:该文件夹下包含 `AssemblyInfo.cs` 文件,它是在创建项目过程中自动添加的,该文件包含程序集属性的设置。

Program.cs 文件:该文件是应用程序文件,它包含用户自己编写的代码。

example1_1.csproj 文件:该文件是项目文件,是建立项目的时候自动生成的文件。

example1_1.sln 文件:该文件是自动生成的以项目名命名的解决方案文件,它通过为环境提供对项目、项目项和解决方案项在磁盘上位置的引用,可将它们组织到解决方案中。

example1_1.suo 文件:也是一个解决方案文件,它是我们建立项目时自动以项目名命名的文件,主要解决方案用户选项,记录所有将与解决方案建立关联的选项,以便在每次打开时,它都包含用户所做的自定义设置。

编写一个应用程序可能包含多个文件,才能生成可执行文件,所有这些文件的集合叫做一个项目。项目名称可以是任何标识符,这里用本项目所在的章节号,本书所有的项目名称(项目所在目录)、文件名及以后网站所在目录都采用本方法,以方便查找。

(6) 打开项目。

如果要打开已经存在的项目,可以选择菜单中的“文件”下的“打开”项,再从中选择“项目/解决方案(P)”,找到某个项目文件下的 `.sln` 文件,双击它即可打开已经存在的项目。

1.3.2 创建 Windows 应用程序

目前,在 Windows 操作系统下开发的项目基本上以窗口图形界面为主的软件,这些软件在 C#.NET 开发中称为 Windows 窗体应用程序,下面我们用 C#.NET 语言开发第一个 Windows 应用程序实例。

例 1.2 建立一个 Windows 应用程序,要求运行后,从窗体的文本框中输入一个名字,点击确定按钮后,屏幕出现欢迎信息。

设计步骤如下:

(1) 在集成开发环境的文件菜单中选择“新建”,“项目”,在打开新建工程对话框我们选择类型为“Visual C#”,在模板中选择“Windows 窗体应用程序”,在名称处输入工程名称“example1_2”,在位置中选择保存的路径,如图 1.3.4 所示:

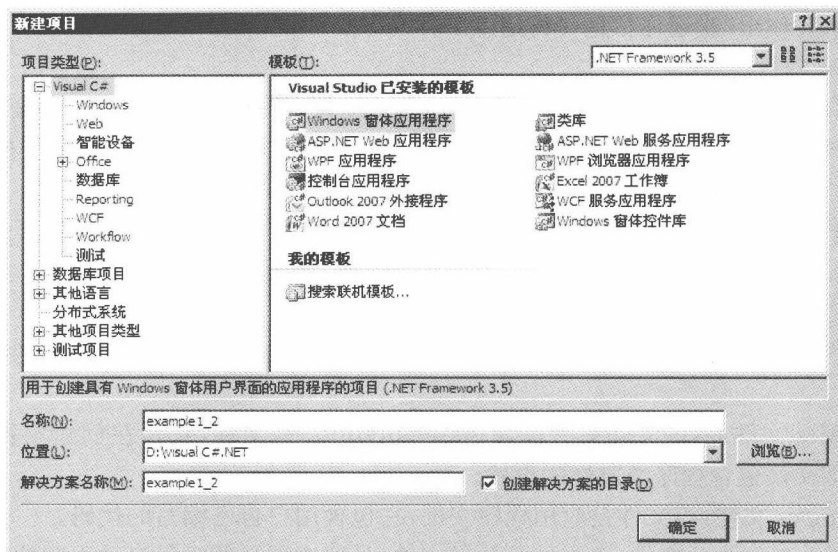


图 1.3.4 建立 Windows 应用程序

(2) 确定后,系统自动生成了程序框架,进入 Windows 窗体设计界面,如图 1.3.5 所示:

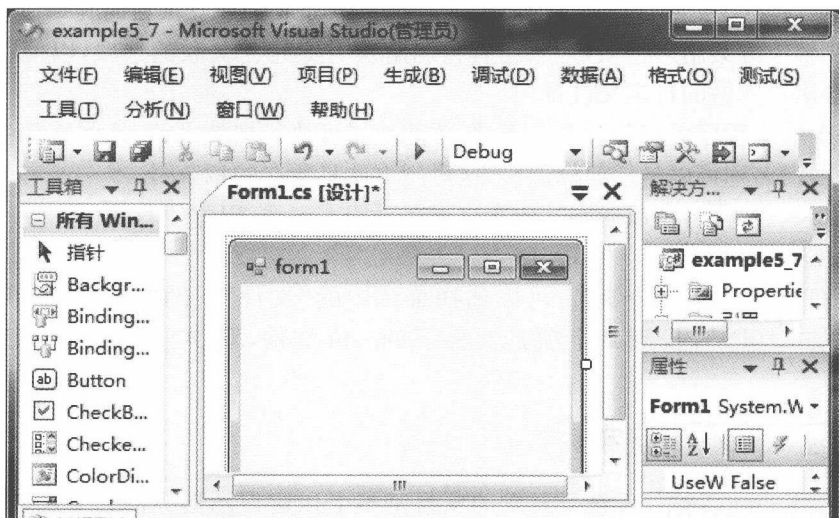


图 1.3.5 Windows 应用程序设计界面

窗体以及窗体内的任何一个控件都可以根据需要调整其大小,只需把鼠标移至窗体或控件边框周围的圆点上,按住鼠标右键拖放即可改变其大小。

(3) 接着分别从工具箱中拖入两个标签、一个文本框、一个按钮到窗体上,然后设置好它们的 text 属性,设置好的界面如图 1.3.6 所示。

(4) 然后,双击窗体上的 button1 按钮(即确定按钮),进入代码设计界面,在下列函数中输入如下代码:

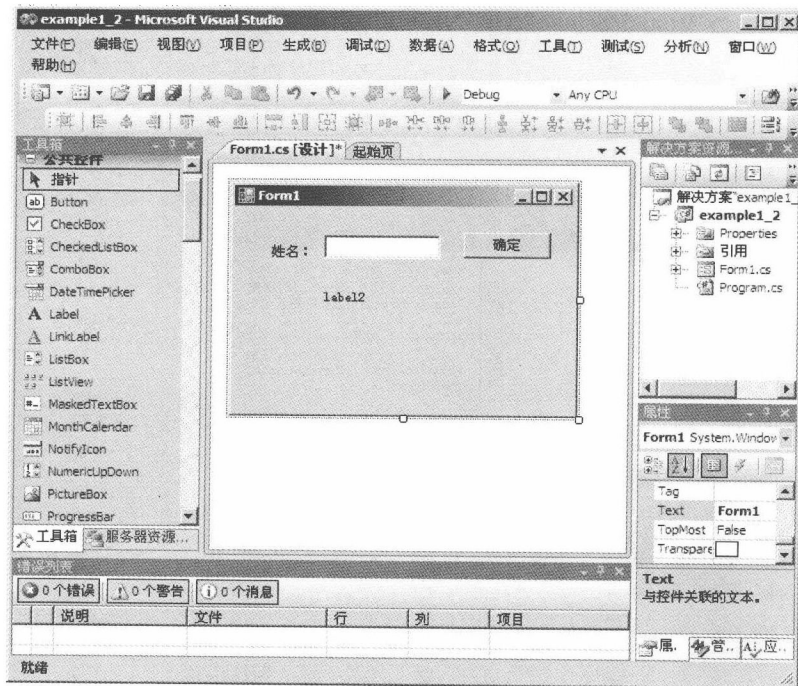


图 1.3.6 设计好的 Windows 应用程序界面

```
private void button1_Click(object sender, EventArgs e)
{
    //以下一行是用户写入的代码
    label2.Text = textBox1.Text + ",欢迎你!";
}
```

(5) 点击工具栏上的运行按钮“▶”或 CTRL + F5 键,运行程序,从文本框中输入文字后,点击确定按钮,得到运行结果如图 1.3.7 所示。

Windows 应用程序项目对应的文件夹除了上面所说的控制台项目文件外,还有窗体的相关文件。其中 program.cs 是程序入口文件,form1.cs 是一个 C# 代码文件,包含用户编写的代码。Form1.designer.cs 是对应 form1.cs 的文件,实际上是 Form1 那个

Form1 类的一部分,里面大部分代码是自动生成的,主要包括窗体控件的相关设置。Form1.resx 是资源导入使用文件,如一些图片和音乐之类的,可以通过它导入到项目中,这样,生成的时候就不需要引用外部的文件了。

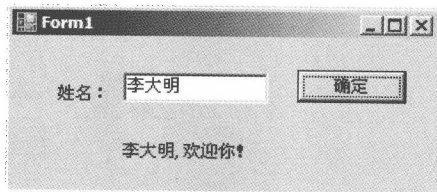


图 1.3.7 Windows 应用程序运行结果

1.4 C# 集成开发环境

上面两个例子分别介绍了控制台应用程序和 Windows 应用程序,下面再专门讨论 C#

集成开发环境。运行 Visual Studio2008.NET 后,启动的窗体如图 1.4.1 所示:

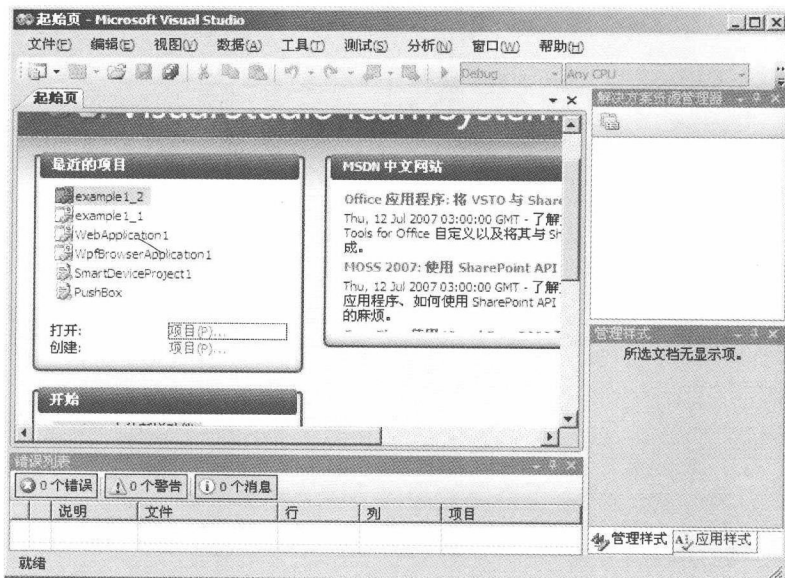


图 1.4.1 Visual Studio2008.NET 初始窗体

在该窗体下,可以新建项目,也可以打开已建好的项目。项目可以是控制台应用程序、Windows 应用程序、ASP.NET 应用程序(动态网站)等。其中控制台应用程序只有代码视图。

新建或打开 Windows 应用程序后,进入 Windows 应用程序视图设计窗体。设计窗体一般被分成几个部分,分别是解决方案资源管理器、属性、设计视图(与代码视图共用一个窗口,它们之间可以转换)及状态窗等,如图 1.4.2 所示:

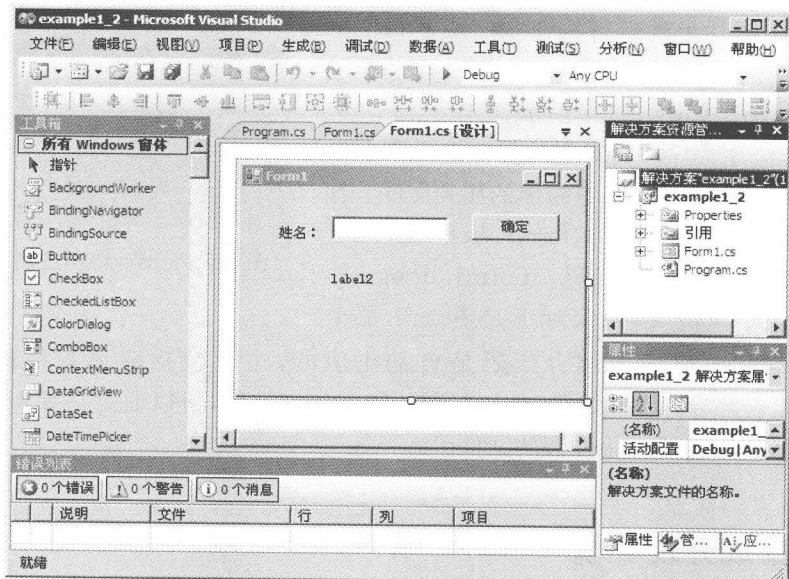


图 1.4.2 Visual Studio2008.NET 的 Windows 应用程序设计窗体

1.4.1 解决方案资源管理器

使用 Visual Studio.NET 开发的每一个应用程序叫解决方案,每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块,一个项目可以有多个项。每个项下可以有一个或多个文件夹及文件。其中解决方案资源管理文件的扩展名为. sln,而 C#.NET 程序代码文件的扩展名为. cs。

由于一个项目中可以有很多窗体,那么程序运行后先启动那一个窗体呢? 这个由解决方案资源管理下的文件 program. cs 中的代码决定,在该文件代码中有一个方法 static void Main(),它是整个程序第一个执行的方法,该方法下有一行代码:

```
Application.Run(new Form1());
```

这行代码决定了程序运行后启动的是 Form1 窗体,如果把它改为:

```
Application.Run(new Form2());
```

则程序运行后,第一个显示的是 Form2 窗体。

“解决方案资源管理器”子窗口显示 Visual Studio.NET 解决方案的树形结构,主要用于用户管理与选择项目中的文件。在“解决方案资源管理器”中可以浏览组成解决方案的所有项目和每个项目中的文件,可以对解决方案的各元素进行组织和编辑,即在解决方案资源管理窗口中,选中相应的文件双击后,会在设计视图(代码视图)窗口中出现相应文件的代码或窗体设计。解决方案资源管理器窗口如图 1.4.3 所示。

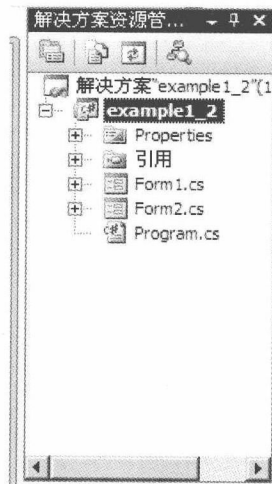


图 1.4.3 解决方案资源管理器

1.4.2 设计器视图与代码视图

设计器视图与代码视图是程序设计最常用的两个子窗口,这也是用户设计窗体或编写代码的工作区。设计器视图用来设计 Windows 窗体或 Web 窗体,代码视图实质上是一个纯文本编辑器,用于设计程序源代码。设计器视图则可以设计 Windows 或 Web 项目的界面,可以添加修改用户控件。这两个子窗体可以通过菜单或在解决方案资源管理器子窗体中的两个按钮操作进行转换。它们的窗口如图 1.4.4,图 1.4.5 所示。

1.4.3 工具箱与服务器资源管理器

“工具箱”主要用于用户在设计窗体时向 Windows 应用程序或 Web 应用程序的设计窗口上添加控件。“工具箱”使用选项卡分类管理其中的控件,打开“工具箱”将显示 Visual Studio 项目中使用的各个不同的控件列表。根据当前正在使用的设计器或编辑器,“工具箱”中可用的选项卡和控件会有所变化。

“服务器资源管理器”是 Visual Studio.NET 的服务器管理控制台。当开发数据应用程