

1.1 .NET 简介

1.1.1 .NET 简述

Microsoft.NET，简称.NET，是 Microsoft 的新一代技术平台，构建于开放的 Internet 协议和标准上，以新的方式提供计算和通信等服务。除 Visual C#外，Visual Basic、Visual C++等多种开发语言均被集成到.NET 平台中，为开发者提供了统一的用户界面和安全机制。.NET 的公用语言子集（Common Language Subset，CLS）提供了无缝的集成给符合其规范的语言和类库，类库提供了对 XML（Extensible Markup Language，可扩展标记语言）的完全支持。

在.NET 平台上编写的源代码被编译成与处理器无关的中间语言（Intermediate Language，IL）代码，程序运行时在即时编译器（Just-In-Time，JIT）的编译下、由中间语言代码编译成本机机器代码，以实现程序开发的跨平台性和可移植性。

.NET 开发平台如图 1.1 所示。

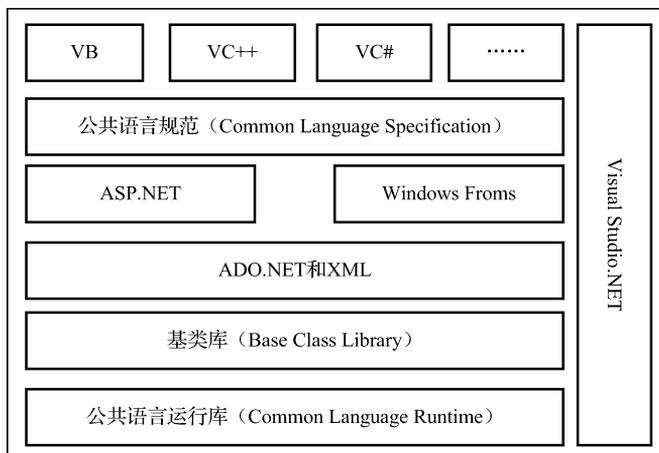


图 1.1 .NET 开发平台

.NET 开发平台包括底层操作系统、.NET 企业服务器、XML Web 服务构件、.NET 框架、.NET 开发工具五个部分。其中，.NET 框架是核心部分。

1. 底层操作系统

Web 服务程序和使用 Web 服务的应用程序都运行在计算机上，如 Windows 7、Windows 8、Windows 10 等操作系统是必需的。

2. .NET 企业服务器

.NET 企业服务器（Enterprise Servers）是 Microsoft 公司推出的基于 Web 服务应用进行企业集成和管理的系列产品，如 Microsoft SQL Server 2014 为企业的信息化和信息集成提供了帮助。

3. XML Web 服务

XML Web 服务提供了一系列高度分布可编程的公用性网络服务，只要客户端支持 XML、SOAP（Simple Object Access Protocol，简单对象访问协议），就可以访问该服务，与之进行交互，不受操作系统、编程语言等的限制。

4. .NET 框架

.NET 框架（Framework）支持包括 Windows 窗体、Web 窗体、XML Web 服务等在内的各种应用程序，是 .NET 平台的一个运行和执行环境。 .NET 框架主要包括公共语言运行库（Common Language Runtime, CLR）和 Framework 基本类库（Framework Class Library, FCL），如图 1.2 所示。



图 1.2 .NET 框架

5. .NET 开发工具

.NET 开发工具主要包括集成开发环境 Visual Studio.NET 和 .NET Framework，用于创建、设计、运行和部署 .NET 解决方案。在 Visual Studio.NET 中，可以用 Visual C#、Visual C++、Visual Basic、Visual J# 和 JScript.NET 等语言进行开发。

1.1.2 .NET 框架

.NET 框架主要包括以下几种组件。

1. 公共语言运行库 (CLR)

CLR 解决了各种不同编程语言之间相互调用同一个程序的问题, 其管理工作如下:

- ① 确定加载代码的方式和时机, 管理对象在内存中的分配。代码执行时, CLR 将决定代码所需要的类和方法, 并按需要将原先编译过的中间语言 (IL) 进行即时编译 (JIT)。
- ② 管理托管所占用的内存, 用垃圾收集器 (Garbage Collection, GC) 实现对无用内存的回收。
- ③ 使用基于异常处理机制的通用错误处理方式, 对托管代码中的错误进行处理和传递。
- ④ 维护 CLR 和应用程序的安全性。在 .NET 框架中, 安全性有两种形式: 代码访问的安全性和角色的安全性。前者用于确保在安全的访问中执行代码, 后者用于控制对系统资源的访问。

2. .NET 框架基本类库 (FCL)

面向对象编程语言的函数库是由一些预先定义好的类构成, 函数库里主要存放一些预先定义好的常用功能函数, 用户编程时可以直接调用函数库里的函数。不同的编程语言都有自己的函数库, 函数库的内容和存取方式都各不相同, 这就给不同的编程语言之间的相互调用造成了巨大的障碍。

.NET 框架的基本类库就是为了解决这个问题而建立的公共类库 (函数库), 各种符合公共语言规范的编程语言都可以调用其中的内容。基本类库将系统的基本功能, 如窗口对象、网络存取、通信协议等分成几个基本类, 程序语言可以直接调用基本类, 也可以在基本类的基础上增加一些功能, 通过继承建立的新类来调用。

3. 数据库访问组件 (ADO.NET 和 XML)

ADO.NET 是一组用于和数据源进行交互的面向对象类库。一般情况下, 数据源可以是数据库, 也可以是文本文件、Excel 表格或者 XML 文件。ADO.NET 允许和不同类型的数据源进行交互, 不同的数据源采用相应的协议, 可以使用 ODBC 协议, 也可以使用 OLE-DB 协议, 这些数据源都可以通过 .NET 的 ADO.NET 类库来进行连接, 这些类库称为 Data Providers, 通常以与之交互的协议和数据源的类型来命名。

4. Windows 桌面应用界面编程组件 (Windows Form)

Windows Form 是一种用户界面, Word、Excel 等的界面就是 Windows Form。

5. 基于 ASP.NET 编程框架的网络服务 (Web Services) 和网络表单 (Web Forms)

Web Forms 是另一种用户界面, ASP.NET 用于开发 Web 应用程序, 应用程序中包括验证、缓存、状态管理、调试和部署等功能。

1.1.3 .NET 特性

1. 一次编译, 多处运行

.NET Framework 具有平台无关的特性, 程序员开发的 .NET 应用程序经编译后得到中间语言代码交由 .NET Framework 完成机器码的即时编译工作。程序员无须考虑硬件的不同或系统需要指令的不同或程序运行的系统平台环境, 只要获得 .NET Framework 支持就可以正常运

行程序，程序代码的移植和运行都由.NET Framework 自动完成。

2. 自动内存管理，安全编程

.NET 中的 CLR 为应用程序提供了高性能的垃圾收集环境。垃圾收集器自动追踪应用程序操作的对象，程序员再也用不着和复杂的内存管理打交道。

3. 强大的基本类库

.NET Framework 中的基本类库提供了非常丰富的类，使程序开发变得简单。例如，字符串处理、数据收集、数据库连接，以及文件访问等，.NET Framework 都提供了完善的类可以直接使用。Microsoft .NET 框架基本类库是一组广泛的、面向对象的、可重用类的集合，为应用程序提供各种开发支持，包括命令行程序、Windows 窗体程序及 Web 程序。

1.2 Visual Studio .NET 集成开发环境

1.2.1 解决方案管理器

解决方案资源管理器以树状的结构显示整个解决方案中包括哪些项目、每个项目具体组成等信息，如图 1.3 所示。

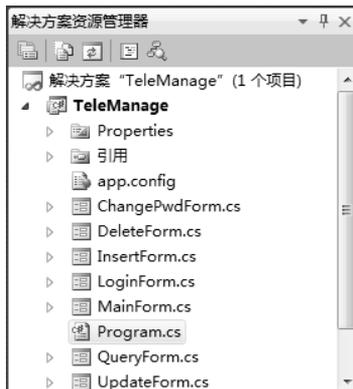


图 1.3 解决方案资源管理器

解决方案资源管理器像一个列表，列出了解决方案中的所有窗体文件和其他相关文件，双击其中任何一个文件都可以在编程环境的工作区中显示出这个文件，从而进行编辑。当工作区中显示的文件过多时，可以关掉暂时不用的文件，等到要用时再从解决方案资源管理器中双击将其打开。

1.2.2 控件工具箱

控件工具箱列出所有可用控件，如按钮、标签、文本框等。C#提供的控件分别放在“数

据”、“组件”、“所有 Windows 窗体”、“常规”等选项卡中，如图 1.4 所示。



图 1.4 工具箱

选项卡中的控件不是一成不变的，可以根据需要增加或删除。在工具箱窗口相应控件选项卡上右击，在弹出的快捷菜单中选择“选择项”对话框，弹出一个包含所有可选择控件的对话框，勾选所需使用的控件前面的复选框，单击“确定”按钮，就可以在当前选中的选项卡下面添加这个控件。展开相应控件选项卡，在所需删除的控件上右击，在快捷菜单中选择“删除”即可。

1.2.3 属性窗口

所谓属性，就是对象的一些性质。我们常用到的控件属性，如文本框控件的宽度、高度、位置等都是属性。

属性窗口用于显示和设置选中的控件或窗体等对象的属性，如图 1.5 所示。可以直接在属性窗口中设置各属性值，也可以通过代码设置。当选中属性窗口的某一栏时，属性窗口下部的窗格中会显示当前选中属性的简要介绍。

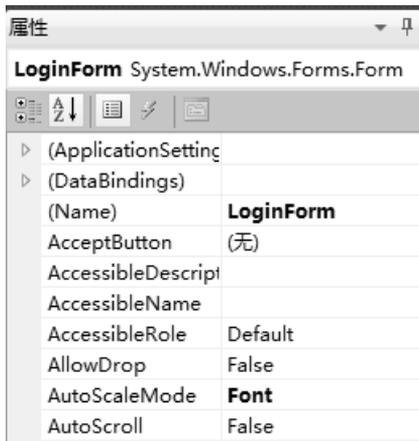


图 1.5 属性窗口

1.3 创建一个简单的 C#程序

1.3.1 C#程序的文件类型

在 Visual Studio.NET 集成开发环境下编写的 C#应用程序称作解决方案 (solution)，可以由一个或多个项目组成，每个项目可以包含一个或多个程序文件，一个项目可以只有一个窗体，也可以有多个窗体和附加文件。

以项目名称为 Hello 的 Windows 窗体应用程序为例，C#窗体应用程序中的文件类型及其说明如表 1.1 所示。

表 1.1 文件类型

类 型	图 标	说 明
Hello.sln		解决方案文件——解决方案的主文件，记录与解决方案及所含项目有关的信息，打开该文件才能处理或运行项目
HelloForm.cs		.cs (C#) 文件——包含为窗体编写的方法代码，可以在任何编辑器中打开
HelloForm.resx		窗体资源文件——定义窗体使用的所有资源，包括文本串、数字及图形等
Hello.csproj		项目文件——描述项目，列出项目中包括的文件
HelloForm.Designer.cs		.cs (C#) 文件——包含窗体及其控件的定义
Program.cs		.cs (C#) 文件——包含自动生成的、在执行程序时首先运行的代码

1.3.2 应用程序开发步骤

不考虑问题分析和算法分析的前期工作阶段，C#的 Windows 窗体应用程序开发包括设计用户界面与设置属性、编写代码、调试程序、运行程序等四个步骤，重复按这四个步骤来规划和创建项目，具体如下所示。

(1) 设计用户界面与设置相应属性：创建窗体对象，在其上添加所需的控件对象，对每个对象进行命名，并设置其相应属性值。

(2) 编写代码：编写程序功能实现所需的代码。

(3) 调试程序：对出现的语法错误或逻辑错误进行调试。

(4) 运行程序：运行程序以实现程序功能。

C#的 Web 应用程序开发步骤与 Windows 窗体应用程序开发步骤类似。

C#的控制台应用程序开发步骤是编写代码、调试程序、运行程序。

1.3.3 三种常用应用程序

1. Windows 窗体应用程序

Windows 窗体应用程序是图形化用户界面程序。Visual Studio 2010 中开发 Windows 窗体应用程序是使用图形用户界面开发工具来进行设计的。

【例 1-1】 在 Windows 窗体中显示文字。要求：创建一个 Windows 窗体应用程序，运行时在消息框中显示“Hello, China!”；程序运行界面如图 1.6 所示。



图 1.6 程序运行界面

具体步骤：

(1) 启动 Visual Studio 2010。

(2) 选择“文件”→“新建”→“项目”选项，弹出“新建项目”对话框，选择 Visual C# 开发项目类型，选择“Windows 窗体应用程序”选项，选择项目文件存放位置，输入项目名称“helloWindows”，单击“确定”按钮，进入设计界面，如图 1.7 所示。

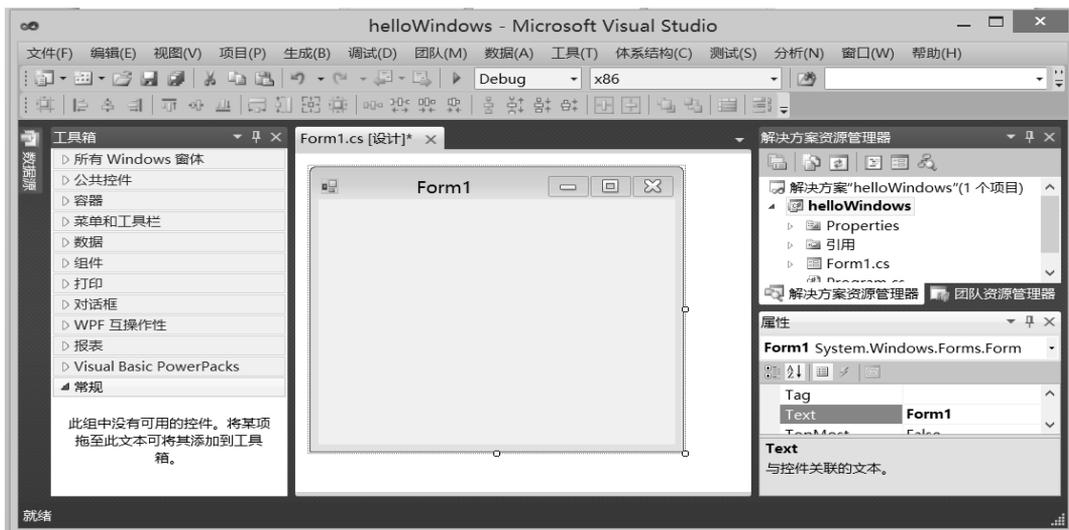


图 1.7 设计界面

(3) 在工具箱中将 Button 控件拖到 Form1 中生成一个按钮对象 button1，调整其大小和位置。右击 button1，在弹出的快捷菜单中选择“属性”选项，在“属性”窗口中将按钮的 Text 属性改为“显示”。

(4) 双击“显示”命令按钮，切换到代码窗口，在 button1_Click 事件中添加一行代码

“Message Box.Show("Hello, China!");”。

代码窗口的内容如下：

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

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

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Hello, China!", "提示");
        }
    }
}
```

大部分 C#语句是自动生成的

(5) 单击“启动调试”按钮或按 F5 键运行程序，显示一个窗体界面，单击窗体中的“显示”按钮，弹出消息框，内容为“Hello, China!”，如图 1.6 所示。

分析与归纳：

1) 命名空间 (NameSpace)

命名空间用关键字 `NameSpace` 定义，用来限定名称的解析和使用范围。一个 C#文件中可以包含多个命名空间，类似于文件夹，用来组织管理代码。

使用关键字 `using` 自动导入由 .NET 框架提供的名为 `System` 的命名空间，`System` 命名空间包含了若干个类和二级命名空间。本程序开始时用 `using` 指令引用了一系列给定的命名空间，后面就可以直接使用命名空间的成员。

例如，`System.Windows.Forms` 是用于创建基于 `Windows` 的应用程序类的命名空间，其大多数类是从 `Control` 类派生而来的，`Control` 类是定义控件或可视化组件的基类，为在 `Form` 中显示的所有控件或组件提供基本功能。通过该命名空间可以实现 `Microsoft Windows` 操作系统提供的用户界面功能。

除 .NET 框架提供的命名空间外，还可以自定义命名空间。例如，程序中 `namespace helloWindows`，`helloWindows` 就是一个命名空间。

创建项目时，会自动创建一个以项目名命名的命名空间。

2) 类 (Class)

类用关键字 `class` 定义，C#程序中的每个对象都必须属于一个类。本例中声明了一个名叫 `Form1` 的类。其中，`public` 是访问修饰符，说明类 `Form1` 具有公共访问权限；`partial` 说明类 `Form1` 是分部类，可以将类的定义拆分到两个或多个源文件中；`class Form1: Form` 说明类

Form1 继承自 Form 类；Form 类用以表示应用程序内的窗口。

3) private void button1_Click(object sender, EventArgs e)

private 是访问修饰符，说明 button1_Click 事件方法的访问权限是私有的，仅能在 Form1 类的内部使用；当单击 button1 时，便会触发一个 Click（单击）事件，程序自动调用相应的事件方法进行处理；事件方法的基础代码由系统自动提供，具体内容由用户自己编写。

本例中，由于 Click 事件是 button1 的默认事件，直接在 button1 上双击就可以进入该事件方法进行代码编写。对于非默认事件，如 MouseClick 事件，则需要选择 button1 后，在属性窗口的工具栏中单击“事件”按钮，从列出的所有事件中，双击 MouseClick 才能进入 button1 的 MouseClick 事件方法进行代码编写。

4) 消息框 (MessageBox)

MessageBox 是 .NET 框架基本类库中的类，位于 System.Windows.Forms 命名空间中，用于显示包含文本、按钮、图标等内容的消息框，其 Show() 方法中的参数决定消息框内需要显示的内容，本例中消息框内有标题“提示”和显示内容“Hello, China!”。

5) 注释语句

单行注释，即写在每一行中“//”的后面，需注释的内容不跨行；多行注释，即以“/*”开始，以“*/”结束，需注释的内容可跨多行。

使用注释语句，可以提高程序的可读性和可维护性。编译程序时，注释语句不参与编译。

6) 其他

C# 程序区分大小写；所有语句以“;”（分号）作为结束；“{”和“}”用来对语句进行分组，分别标示某个代码块的开始和结束；书写代码时尽量用缩进来表示代码的结构层次。

2. Web 应用程序

ASP.NET Web 应用程序，即网站，是由 Web 窗体（页）、控件、代码模块和服务组成的集合。Visual Studio 2010 中开发 Web 应用程序也是使用图形用户界面开发工具来进行设计的。

【例 1-2】 Web 窗体中显示文字。要求：创建一个 Web 应用程序，运行时在标签中显示“Hello, China!”；程序运行界面如图 1.8 所示。



图 1.8 运行界面

具体步骤：

(1) 启动 Visual Studio 2010。

(2) 选择“文件”→“新建”→“网站”选项，弹出“新建网站”对话框，选择要建立的网站类型为“ASP.NET 网站”，输入网站文件系统的文件夹名称 WebSite，如图 1.9 所示。单击“确定”按钮，系统自动完成项目的配置，生成 Web 窗体文件 Default.aspx 及与之相关

联的源代码文件 Default.aspx.cs 等。



图 1.9 新建 ASP.NET 网站

(3) 在解决方案管理器中右击文件 Default.aspx，在弹出的快捷菜单中选择“重命名”选项，将文件名 Default.aspx 修改为 HelloWeb.aspx，系统将 Default.aspx.cs 自动改为 HelloWeb.aspx.cs，如图 1.10 所示。

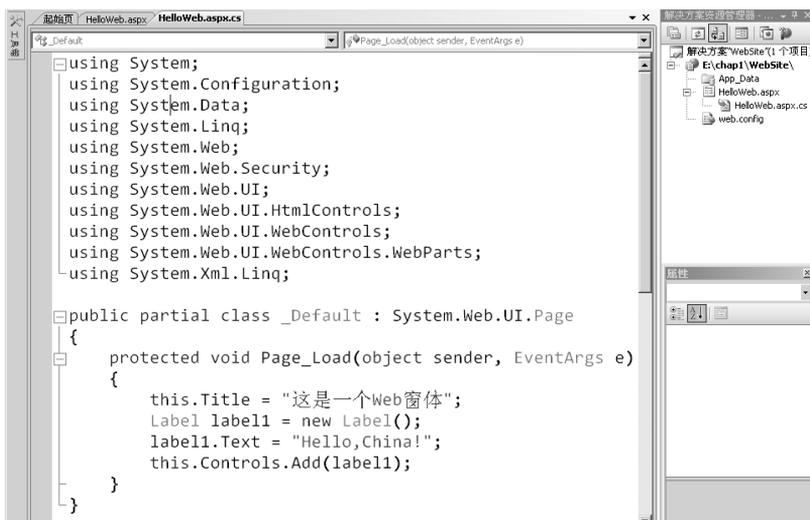


图 1.10 代码界面

(4) 右击 Visual Studio 2010 设计区，在弹出的快捷菜单中选择“查看代码”选项，将 Web 窗体的设计或编辑视图切换为源程序代码文件 HelloWeb.aspx.cs 的编辑视图，在 HelloWeb.aspx.cs 文件的源代码窗口添加代码，如图 1.10 所示。

(5) 在解决方案管理器中，右击文件“HelloWeb.aspx”，在弹出的快捷菜单中选择“在浏览器中查看”选项，Visual Studio 2010 将启动 C#语言编辑器编译程序，执行 Web 窗体应用程序，将结果输出到浏览器，如图 1.8 所示。

3. 控制台应用程序

控制台应用程序是指非图形化用户界面，使用命令行方式进行人机交互的程序。

【例 1-3】 控制台输出一行文字“Hello, China!”。

具体步骤：

(1) 启动 Visual Studio 2010。

(2) 选择“文件”→“新建”→“项目”选项，弹出“新建项目”对话框，如图 1.11 所示。在“已安装的模板”中选择模板 Visual C#，选择“控制台应用程序”选项，选择项目存放位置为“E:\chap1\”，输入项目名称“helloConsole”，单击“确定”按钮，进入编程界面。



图 1.11 “新建项目”对话框

(3) 在代码窗口的 Main 方法中输入相应代码，Program.cs 文件的代码如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace helloConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello,China!");
            Console.Read();
        }
    }
}
```

(4) 单击“启动调试”按钮或按 F5 键运行程序，输出结果如图 1.12 所示。

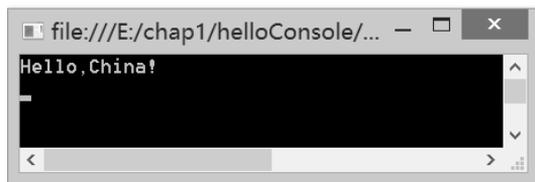


图 1.12 运行界面

分析与归纳：

(1) 控制台应用程序中会声明一个名为 **Program** 的类

(2) 控制台应用程序中会声明一个 **Main()** 方法。

每个 C# 程序都必须含有且只能含有一个 **Main()** 方法，用于指示编译器从此处开始执行程序。**Main()** 方法在类或构造的内部声明，是一个静态方法，用 **static** 修饰符进行声明，静态方法在类对象创建之前可以被调用。

声明 **Main()** 方法时，可以有参数，也可以没有参数，例如：

```
static void Main()  
static void Main(string[] args)
```

void 表示 **Main()** 方法无返回值。

(3) 输入和输出。

程序的输入和输出功能由 **Console** 类的不同方法来完成。

Console.ReadLine() 方法与 **Console.Read()** 方法都用于从键盘读入信息，返回值是字符串类型，区别在于前者读取一行字符，后者读取一个字符。

Console.WriteLine() 方法与 **Console.Write()** 方法都用于在输出设备上输出，区别在于前者输出后会自动换行而后者不换行。

1.4 操作使用技巧及开发调试技巧

1.4.1 拆分代码窗口

当需要对同一文档的不同部分代码进行查看的时候，可以通过拆分代码窗口来查看代码的不同部分。将鼠标移动到代码窗口右上角滚动条的上方，出现双向箭头时向下拖至想拆分的位置，或选择“窗口”→“拆分”选项，就可以在不同窗口通过移动滚动条来查看代码的不同部分，如图 1.13 所示。

1.4.2 设置断点，跟踪调试

在开发过程中使用得最多的就是断点调试方法。

将鼠标定位在需要设置断点的语句上，右击，在弹出的快捷菜单中，选择“断点”→“插入断点”选项，当代码变为红色背景时，该行即设置了一个断点，如图 1.14 所示。



图 1.13 拆分代码窗口

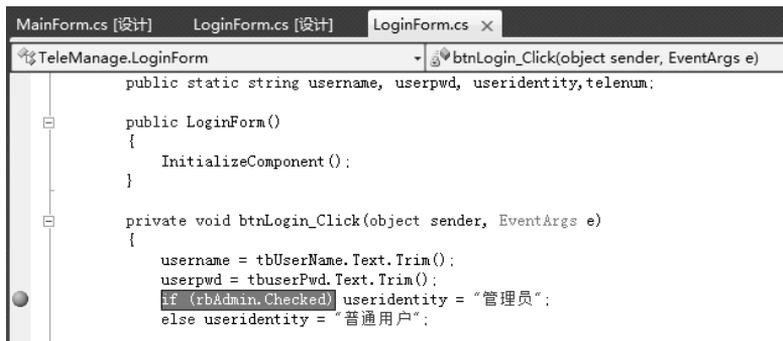


图 1.14 设置断点

当程序执行到该语句时便停下来，如图 1.15 所示，把控制权交给调试程序。此时，可以按 F10 键逐行进行试调，也可以按 F11 键逐语句进行调试。

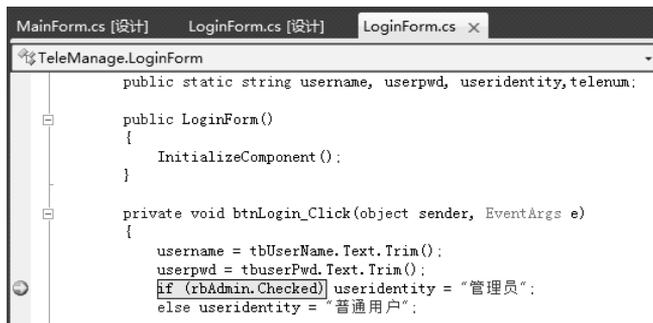


图 1.15 调试程序

条件断点是指在满足一定条件的情况下，程序才停下来等待调试，如图 1.16 所示。将鼠标定位在待设置条件断点的语句行上，右击，在弹出的快捷菜单中选择“断点”→“条件”选项，弹出如图 1.17 所示的“断点条件”对话框。

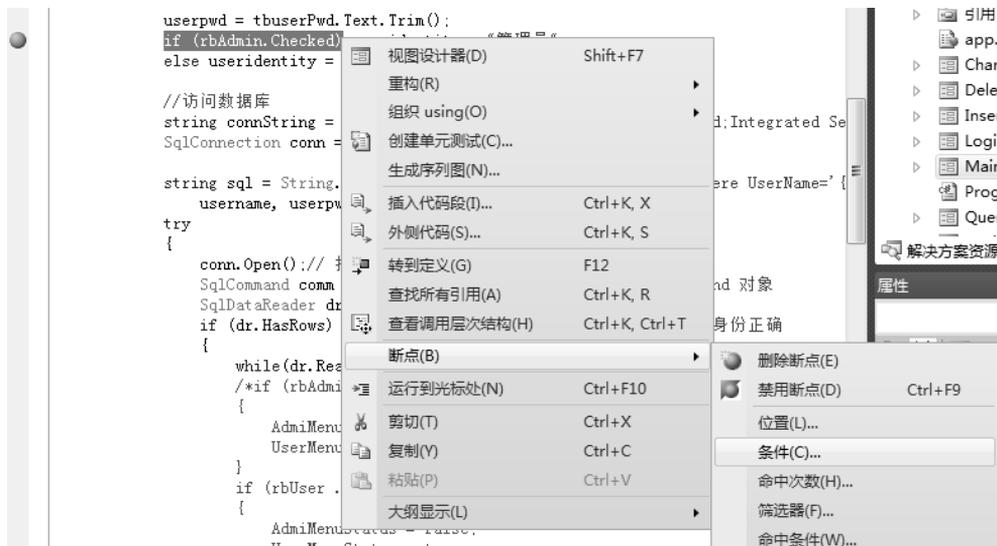


图 1.16 条件断点

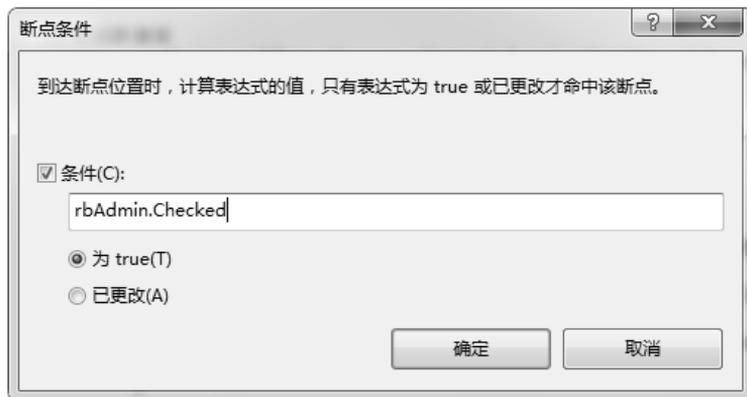


图 1.17 “断点条件”对话框

输入断点条件，如 rbAdmin.Checked 为 True，单击“确定”按钮，运行程序。

1.4.3 变量监视，堆栈观察

在程序调试过程中监视变量的值时，将光标停放在变量的位置上，右击，弹出快捷菜单，如图 1.18 所示。选择“添加监视”选项，就可以看到变量的值在监视对话框中显示出来了，如图 1.19 所示。

在断点调试的过程中还可以查看语句的调用过程，如图 1.20 和图 1.21 所示。

1.5 本章小结

本章主要介绍了.NET 相关技术、Visual Studio.NET 2010 中 C#的集成开发环境、操作使用技巧、开发调试技巧，并以示例方式在实训中讲解了如何创建 C#的三种常用应用程序。



1. 填空题

- (1) C#语言源代码文件的扩展名是_____。
- (2) 当进入 VS 集成环境时，如果没有显示“工具箱”窗口，应选择_____菜单中的“工具箱”选项，以显示“工具箱”窗口。
- (3) 在 Visual Studio.NET 窗口，_____窗口显示了当前 Visual Studio 解决方案的树状结构。
- (4) .NET Framework 主要包括_____和_____。

2. 选择题

- (1) 公共语言运行库是指_____。
A. CRL B. CLR C. CRR D. CLS
- (2) .NET 平台是一个新的开发框架，_____是.NET 的核心部分。
A. C# B. VB.NET C. 操作系统 D. .NET Framework
- (3) 解决方案文件的扩展名是_____。
A. .csproj B. .cs C. .sln D. .suo
- (4) 利用 C#可以开发 3 种类型的应用程序，不包括_____。
A. SQL 程序 B. 控制台程序 C. Windows 程序 D. Web 程序
- (5) .NET 框架是.NET 战略的基础，是一种新的便捷的开发平台，它具有两个主要的组件，分别是_____和 Framework 类库。
A. 公共语言运行库 B. Web 服务 C. 命名空间 D. Main()函数
- (6) 运行 C#程序可以通过_____键实现。
A. Alt+F5 B. Ctrl+F5 C. Alt+ Ctrl+F5 D. F5

3. 判断题

- (1) .Net 包含两个部分，即公共语言运行库和框架类库。
- (2) .NET Framework 运行环境除了支持 VB.net 和 C#两种编程语言，还支持 Perl、C++、.NET、J#、Jscript.NET、ActionScript。
- (3) Windows 应用程序和 Web 应用程序都是通过事件触发的。
- (4) C#源代码的扩展名为.cs。

(5) DotNet 包含两部分，即公共语言运行库和框架类库。

4. 简答题

- (1) 一个完整的 C#控制台程序包括哪些必要的组成部分？
- (2) 简述创建 Windows 窗体应用程序的步骤。

5. 上机实操题

- (1) 创建一个 Windows 窗体应用程序，在窗体上输出字符串“C#.NET 程序设计”。
- (2) 创建一个 Web 应用程序，在窗体上输出字符串“C#.NET 程序设计”。
- (3) 创建一个控制台应用程序，输出字符串“C#.NET 程序设计”。