Broadview
www.broadview.com.cn

.NET技术丛书

◆最主流的技术与平台
◆专为快速学习和就业而设计
◆详细的实验步骤和讲解
◆手把手带您熟悉微软技术
◆知识 + 实验 = 快速掌握 + 就业

# .NET平台下Windows程序设计

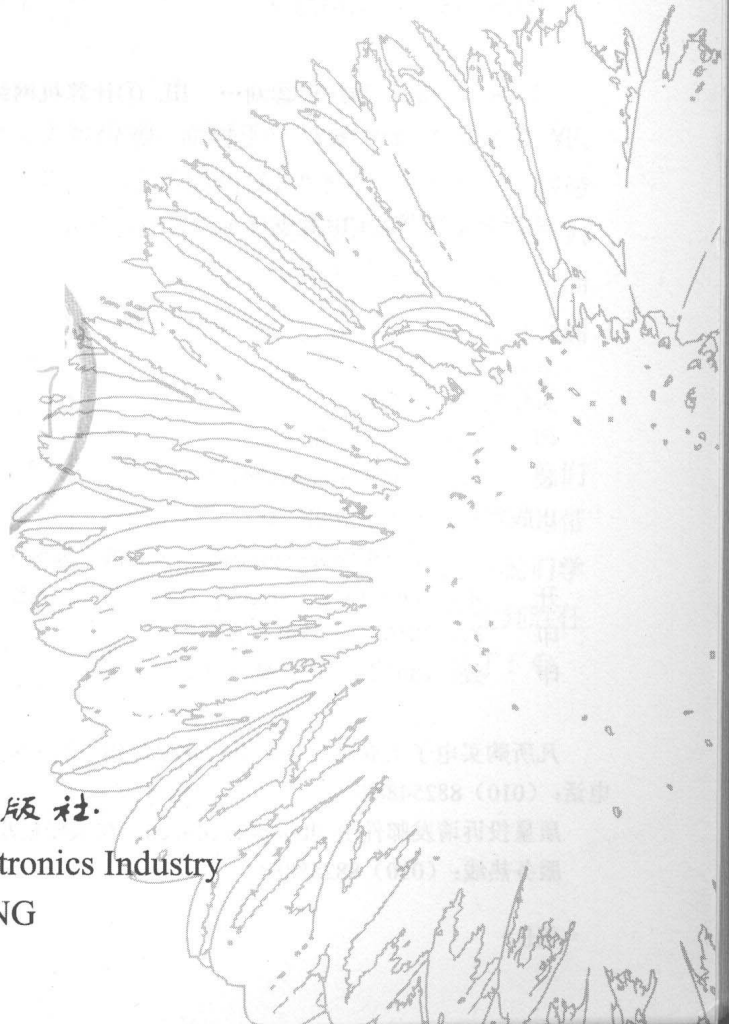
周羽明 刘元婷 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

.NET技术丛书

# .NET平台下Windows程序设计

周羽明　刘元婷　编著



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

## 内 容 简 介

微软公司一直引领IT行业的发展，平台占据市场绝大多数份额。而对于一个计算机的专业的从业人员，对微软整体技术的把握与发展，也是大多数IT从业人员的必然选择。

这本书籍就带我们全面地了解学习掌握微软.NET平台下的Windows程序设计、SQL Server与ADO.NET程序设计、以及XML的基础知识。相信通过学习，您可以全面地掌握.NET平台下的Windows程序设计。

按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验让我们以最快速度全面地掌握微软平台与技术。

本书可作为高校相关专业教材或培训班教材，也可供对.NET技术感兴趣的从业人员阅读和参考。



**图书在版编目（CIP）数据**

.NET平台下Windows程序设计 / 周羽明，刘元婷编著. —北京：电子工业出版社，2010.4
（.NET技术丛书）
ISBN 978-7-121-10333-9

Ⅰ. N… Ⅱ. ①周… ②刘… Ⅲ. ①计算机网络－程序设计 ②窗口软件，Windows－程序设计
Ⅳ. TP393.09 TP316.7

中国版本图书馆CIP数据核字（2010）第022696号

责任编辑：高洪霞
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编100036
开　　本：860×1092　1/16　印张：28　字数：783千字
印　　次：2010年4月第1次印刷
印　　数：4000册　定价：55.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

# 软件产业的未来是我们的

由于经济危机等不利因素的影响，世界经济处在一种不确定中。IT 行业也不能独善其身，同样面临着严峻的挑战。很多 IT 企业开始收缩产品线，裁减开发团队规模以应对这场危机。然而在这样的形势下，我们看到世界基础软件开发及中国的外包产业却逆风飞扬，呈现出一种前所未有的所谓“危机、危机、危中寻机”的态势。

十年寒窗，等我们毕业走向社会以后，却发现自己学到的知识与社会所有脱节。特别是计算机行业，技术发展日新月异。但是在学校所学知识真的就没有用么？不！这就像武侠小说，这十年我们已经练就了内功，但是却不会一套拳法、剑法，怎么能闯荡江湖？特别是计算机专业的学生，数学和计算机基础的学习，已经让我们有了不浅的内功，只需要把这些内功发挥出来。所以，我们可能需要的就是一套武林最正派的外家功夫！

.NET 技术丛书

微软公司一直引领 IT 行业的发展，平台占据市场绝大多数份额。而对于一个计算机专业的从业人员来说，对微软整体技术的把握与发展，也是大多数 IT 从业人员的必然选择。“.NET 技术丛书”将带领我们从基础开始进入微软平台开发领域，本套丛书包含：《.NET 平台与 C#面向对象程序设计》、《.NET 平台下 Windows 程序设计》、《.NET 平台下 Web 程序设计》。三本书分别面向基础的语言与面向对象的思想，Windows 平台与 Web 平台。提供最实用的市场主流知识和技术实训试验，让我们全面掌握微软开发平台的方方面面。本套丛书全部作者均来自一线开发人员，具有多年的实践项目经验，除封面署名作者外，其他参与编写人员有：杨小兰、王超、陈波、梁建红。

按照学习的顺序和技术的难易程度，每一个知识点都配套详细的实训实验，通过实训实验让我们以最快的速度学习所有技术的一招一式。除了知识点以外，详细地讲解了 150 多个实验，手把手地带领读者从零开始，进入到.NET 开发的各个方面的知识点。200 多个基础项目实验的源码，而当我们学习知识和试验后，还有四个不同方向的中小型真实项目源码供我们理解，掌握它们以后就可以达到胜任著名外企开发职位或一般企业初级项目经理职位的水准。到此，我们可以真正下山，闯荡江湖了！☺

## 关于本书实验部分的源码

本书中涉及的所有实验都有完整的代码文件及工程文件供读者下载。

下载网站是：www.broadview.com.cn。

除此之外，我们还给读者提供了 4 个晋级的项目源代码，分别针对不同的方向，涉及 Windows 窗体、Web、网络通信、移动设备、游戏等。

希望读者通过对这 4 个晋级项目的自学，能成长为一名微软技术的高手。

| 项目名称 | 项 目 简 介 |
| --- | --- |
| SMTP Client | SMTP 邮件客户端。通过此项目学习，让学生掌握一般的 Windows Form 项目开发。包含技术有：.NET Framework Windows 基本的控件使用，多线程编程，I/O 流，网络功能（mail），字体编码及文件格式定义保存使用 |
| Club Site Starter Kit | 入门级的 ASP.NET 2.0 站点。通过学习，学生对网络程序的开发有一定认识，对基本的数据库连接、页面与代码逻辑的结构及服务器控件编程有一定掌握 |
| Pocket Sudoku | 趣味性的 Windows Mobile 游戏。通过学习，学生熟悉掌握一般 Mobile 程序开发流程，对 Mobile 设备上的图形绘制、设备的使用、用户界面及简单的网络功能有一定认识 |
| RSS Reader | RSS 阅读器。通过此项目学习让学生认识智能客户端的要素和一般结构，学习掌握 XML 和 RSS 技术，进一步提高.NET 开发技术。可以尝试做 RSS Reader 的 Web 版本和 Mobile 版本 |

**适用读者**

- 如果你是计算机专业的毕业生，这套书能最快地把我们大学的知识与积累，转换成为就业的资本和能力，让我们最快地发挥出我们的积累，创造机会。
- 如果你想进入计算机行业，这套书能让我们最快地学到最实用的技术，给我们带来更多的发展与工作机会，以及以后的方向。

未来是我们的！

# 目　录

# 第1章

# Windows Form 程序设计介绍

## 1.1 .NET 平台下开发 Windows Form 简介

读者学习了.NET Framework，C#语言及软件设计的基本知识后，是不是想开发 Windows 应用程序，既能方便自己的工作，又能让朋友使用，或者展示自己的独特创意？好，下面一起进入 Windows Form 开发。

### 1.1.1 Windows Form 简介

Windows Form 即 Windows 窗体，使用 Windows 窗体可以开发智能客户端。“智能客户端”是易于部署和更新的图形丰富的应用程序，无论是否连接到 Internet 它都可以工作，并且可以比传统的基于 Windows 的应用程序以更安全的方式访问本地计算机中的资源。

Windows 窗体是.NET Framework 的智能客户端技术，.NET Framework 是一组可简化常用应用程序任务（如读写文件系统）的托管库。使用类似 Visual Studio 的开发环境，用户通过该开发环境可以创建 Windows 窗体智能客户端应用程序，以显示信息、请求用户输入，以及通过网络与远程计算机通信。

在 Windows 窗体中，“窗体”是向用户显示信息的可视界面。通常情况下，通过向窗体上添加控件并开发对用户操作（如单击或按下鼠标按键）的响应，生成 Windows 窗体应用程序。“控件”是显示数据或接受数据输入的相对独立的用户界面（UI）元素。

当用户对窗体或其中的某个控件进行操作时，将生成事件。应用程序使用代码对这些事件进行响应，并在事件发生时处理事件。

Windows 窗体包含可添加到窗体上的各式控件，如用于显示的文本框、按钮、下拉列表、单选按钮，以及网页的控件。有关可在窗体上使用的所有控件的列表，请参见在 Windows 窗体上使用的控件。如果现有控件不满足需要，Windows 窗体还支持使用 UserControl 类创建用户自己的自定义控件。

Windows 窗体具有丰富的 UI 控件，可模拟像 Microsoft Office 这样的高端应用程序中的功能。使用 ToolStrip 和 MenuStrip 控件，可以创建包含文本和图像、显示子菜单及承载其他控件（如文本框和组合框）的工具栏和菜单。

使用 Visual Studio 的具有拖放功能的 Windows 窗体设计器，可以轻松创建 Windows 窗体应用程序。只需使用光标选择控件并将控件添加到窗体上所需的位置即可。设计器提供类似网格线和对齐线的工具，可简化对齐控件的操作。无论使用 Visual Studio 还是在命令行编译，都可以使用 FlowLayoutPanel、TableLayoutPanel 和 SplitContainer 控件以较短的时间创建高级窗体布局。

最后，如果用户必须创建自己的自定义界面元素，则可使用 System.Drawing 命名空间，其中包含了大量的类，可供选择用于直接在窗体上呈现线条、圆和其他形状。

许多应用程序必须从数据库、XML 文件、XML Web Services 或其他数据源显示数据。Windows 窗体提供了一个名为 DataGridView 的灵活控件，用于使用传统的行和列格式显示此类表格数据，以使每条数据都占据自己的单元格。使用 DataGridView 时，可以自定义各个单元格的外观、将任意行和列锁定在现有位置，以及在单元格内显示复杂控件等。

使用 Windows 窗体智能客户端，通过网格连接数据源成为了一个简单的任务。Visual Studio 2005 和.NET Framework 2.0 中随 Windows 窗体提供的新组件 BindingSource 可以表示到数据源的连接，并公开了将数据绑定到控件、导航至上一条和下一条记录、编辑记录，以及将更改保存回原始数据源的方法。BindingNavigator 控件提供一个与 BindingSource 组件的简单接口，供用户在记录间导航。

使用“数据源”窗口可以轻松地创建数据绑定控件。该窗口可显示项目中的数据源，例如数据库、Web 服务和对象。通过将项从此窗口拖动到项目中的窗体可以创建数据绑定控件；还可以通过将对象从“数据源”窗口拖动到现有控件，来将现有控件数据绑定到数据。

“设置”是另一种可在 Windows 窗体中管理的数据绑定。大多数智能客户端应用程序必须保留一些关于其运行时状态的信息（如窗体的上次已知大小），并保留用户首选项数据（如所保存文件的默认位置）。应用程序设置功能提供了一种简单的方法，可将这两种设置保存在客户端计算机上，从而满足了这些需要。使用 Visual Studio 或代码编辑器定义设置后，这些设置将保存为 XML 并在运行时自动读回内存中。

编写完应用程序后，必须将它发送给用户，以便他们可以在自己的客户端计算机上安装并运行该应用程序。使用 ClickOnce 技术，只需几次单击即可从 Visual Studio 中部署应用程序，并为用户提供指向网站中应用程序的 URL。ClickOnce 管理应用程序中的所有元素和依赖项，并确保应用程序正确地安装在客户端计算机上。

ClickOnce 应用程序可以配置为仅在用户连接到网络时运行，或者在联机和脱机时都可以运行。如果指定应用程序支持脱机操作，则 ClickOnce 将在用户的“开始”菜单中添加指向应用程序的链接。随后，用户不必使用 URL 即可打开应用程序。

更新应用程序时，需将新的部署清单和应用程序的新副本发布到 Web 服务器。ClickOnce 将检测到存在可用更新并升级用户的安装，用户无须自定义编程来更新旧的程序集。

Windows 窗体中提供的许多其他功能可以快速方便地实现一些常规任务，如对一些任务的支持，包括创建对话框、打印、添加帮助和文档，以及将应用程序本地化为多种语言。此外，Windows 窗体依赖于.NET Framework 的可靠安全系统。通过这一系统，可以向客户发布更安全的应用程序。

### 1.1.2 Windows Form 开发技术

Windows Form 是用于生成利用公共语言运行库的 Windows 客户端应用程序的框架。可用公共语言运行库支持的任何语言编写 Windows 窗体应用程序。使用 Windows 窗体的优点如下。

- 简单而且功能强大：Windows 窗体是用于开发 Windows 应用程序的编程模型，它融合了 Visual Basic 6.0 编程模型的简单性与公共语言运行库的强大功能和灵活性。
- 所属权总成本较低：Windows 窗体利用公共语言运行库的版本控制和部署功能，可提供随时间流逝降低的部署成本和更高的应用程序可靠性。这显著降低了用 Windows 窗体编写的应用程序的维护成本（TCO）。
- 控件的结构：Windows 窗体提供用于控件和控件容器的结构，该结构基于控件和容器类的具体

实现。这显著减少了控件和容器间的交互问题。

- 安全性：Windows 窗体充分利用公共语言运行库的安全功能。这意味着 Windows 窗体可用于实现所有内容，从在浏览器中运行的不受信任的控件到安装在用户硬盘上的完全受信任的应用程序，应用范围十分广泛。
- XML Web 服务支持：Windows 窗体为快速轻松地连接到 XML Web 服务提供全面支持。
- 丰富的图形：Windows 窗体是 GDI+的第一批载体之一，GDI+是一种新版本的 Windows 图形设备接口（GDI），支持 Alpha 混合效果、纹理画笔、高级转换、多格式文本支持等。
- 灵活的控件：Windows 窗体提供一组丰富的控件，其中包含 Windows 提供的所有控件。这些控件还提供新功能，如用于按钮、单选按钮和复选框的“平面”样式。
- 数据识别功能：Windows 窗体对 ADO.NET 数据模型提供全面支持。
- ActiveX 控件支持：Windows 窗体对 ActiveX 控件提供全面支持。可以轻松地在 Windows 窗体应用程序中承载 ActiveX 控件。还可以将 Windows 窗体控件作为 ActiveX 控件承载。
- 授权：Windows 窗体利用公共语言运行库的增强授权模型。
- 打印：Windows 窗体提供打印框架，使应用程序能够提供各种打印报表。
- 辅助功能：Windows 窗体控件实现由 Microsoft 活动访问（MSAA）定义的接口，这使得生成支持辅助功能（如屏幕读取器）的应用程序变得简单。
- 设计时支持：Windows 窗体充分利用公共语言运行库提供的元数据和组件模型功能，为控件用户和控件实施者提供全面的设计时支持。

## 1.2　Visual Studio.NET 开发环境介绍

- 默认情况下，用户的 Visual Studio 2005 IDE 布局如图 1-1 所示，“解决方案管理器”、“类视图”窗口显示在右侧，“工具栏”和“服务器资源”管理器窗口显示在左侧，“属性”窗口显示在“解决方案管理器”窗口的下方，“输出”、“错误列表”、“任务列表”、“查找结果”和“测试结果”等窗口显示在 IDE 主窗口的下方。

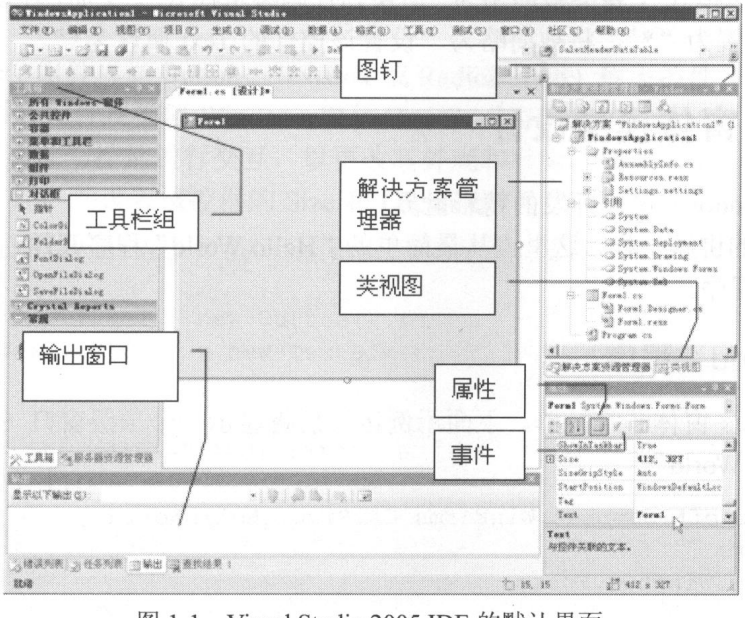



图 1-1　Visual Studio 2005 IDE 的默认界面

- 可以通过单击“图钉”按钮来隐藏或重新显示“工具栏”、“解决方案管理器”和“输出”等窗口。
- 如果上述基本窗口没有在 Visual Studio 2005 IDE 中显示出来，可以使用相应的菜单命令将其调出来，如图 1-2 所示。

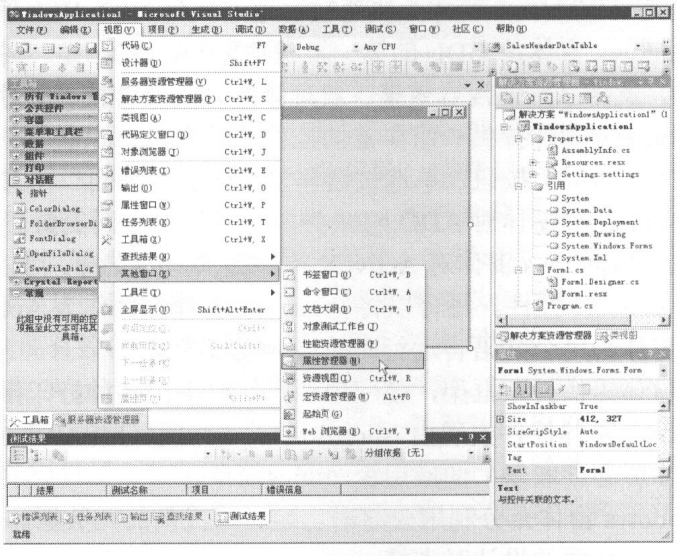


图 1-2　Visual Studio IDE 的简单介绍

- 在“属性”窗口中切换到“事件”选项卡，是指单击“属性”窗口中的表示“事件”的闪电形状小图标。
- 窗体控件的位置表示为：“控件所在的工具箱分类”+“控件名称”，如将“对话框”工具栏中的 FontDialog 控件拖放到窗体中。
- 部分基本操作仅在本书中第一次出现时才进行详细的操作指导，如果在后面的章节中出现相同的操作，将简略描述。如在窗体中单击鼠标右键，在弹出的上下文菜单中选择“查看代码”切换到“代码编辑模式”，将会被简略为“切换到代码编辑模式”。又如，在属性窗口中找到“Text”属性并将其设置为“*”，将被简略为“设置该控件的 Text 属性为*”。

## 1.3　Windows 窗体模型设计

下面开始进入 Windows 窗体开发的精彩世界。

像很多编程语言的讲解一样，这里也从最简单的“Hello World”程序开始，下面是 Windows Form 版的“Hello World”程序。

### 1.3.1　“Hello World”程序

开始使用 Windows 窗体非常简单。下面示例运行后将显示一个顶级窗口（称为窗体），并将标题栏文本设置为“Hello World”。

```
namespace Microsoft.Samples.WinForms.Cs.SimpleHelloWorld {
    using System;
    using System.Windows.Forms;
```

```
    public class SimpleHelloWorld : Form {

        [STAThread]
        public static int Main(string[] args) {
            Application.Run(new SimpleHelloWorld());
            return 0;
        }

        public SimpleHelloWorld() {
            this.Text = "Hello World";
        }
    }
}
```

没错，就是这么简单。OK，读者已经学会了最简单的 Windows 窗体程序，下面探讨 Windows 窗体应用程序模型。

### 1.3.2　Windows 窗体应用程序模型

Windows 窗体的应用程序编程模型主要由窗体、控件及其事件组成。本主题涉及 Windows 窗体应用程序模型的以下方面。

#### 1．窗体

在 Windows 窗体中，Form 类是在应用程序中显示的任何窗口的表示形式。可以使用 Form 类的 BorderStyle 属性创建标准窗口、工具窗口、无边框窗口和浮动窗口。还可使用 Form 类创建有模式窗口，如对话框。通过设置 Form 类的 MDIContainer 属性，可以创建一种特殊类型的窗体 MDI 窗体。MDI 窗体可以在其工作区内包含名为 MDI 子窗体的其他窗体。Form 类为键盘处理（Tab 键顺序）和滚动窗体的内容提供内置的支持。

当为应用程序设计用户界面时，通常创建一个从 Form 派生的类。然后可以添加控件、设置属性、创建事件处理程序，以及向窗体添加编程逻辑。

#### 2．控件

添加到窗体中的每个组件（如 Button、TextBox 或 RadioButton）称为控件。Windows 窗体包括通常与 Windows 关联的所有控件及类似 Windows 窗体 DataGrid 的自定义控件。

通常可以通过设置属性与控件进行交互，以更改其外观和行为。例如，下面的 Form 的派生类向窗体添加一个 Button 控件，并设置该控件的 Size 和 Location。

```
public class HelloWorldForm : System.Windows.Forms.Form {

    private Button button1 = new Button() ;
    private TextBox textBox1 = new TextBox();

    [STAThread]
    public static int Main(string[] args) {
        Application.Run(new HelloWorldForm());
        return 0;
    }

    public HelloWorldForm() {
```

```
        this.Text = "Hello Windows Forms World";
        this.AutoScaleBaseSize = new Size(5, 13);
        this.ClientSize = new Size(392, 117);

        this.MinimumSize = new Size(392, (117 + SystemInformation.CaptionHeight));

        this.AcceptButton=button1;

        button1.Location = new Point(256, 64);
        button1.Size = new Size(120, 40);
        button1.TabIndex = 2;
        button1.Text = "Click Me!";

        button1.Click += new System.EventHandler(button1_Click);

        textBox1.Text = "Hello Windows Forms World";
        textBox1.TabIndex = 1;
        textBox1.Size = new Size(360, 20);
        textBox1.Location = new Point(16, 24);

        this.Controls.Add(button1);
        this.Controls.Add(textBox1);
    }
}
```

窗体对于何时可设置控件的属性提供有限的限制。控件没有阻止更新其状态的模式。创建控件的新实例后，可以立即更改其状态。例如，下面的代码提供两个示例，演示创建 Button 控件的有效方法。

```
Button button1 = new Button();
button1.Location = new Point(256, 64);
button1.Size = new Size(120, 40);
button1.TabIndex = 1;
button1.Text = "Click Me!";
this.Controls.Add(button1);

Button button1 = new Button();
this.Controls.Add(button1);
button1.Location = new Point(256, 64);
button1.Size = new Size(120, 40);
button1.TabIndex = 1;
button1.Text = "Click Me!";
```

Windows 窗体确保用户创建的代码是有效的。例如，如果设置一个 Windows 控件的 Windows 样式位的属性（该属性仅可在创建控件时设置），则 Windows 窗体控件放弃基础 Windows 控件，并创建一个新控件。只有在不使用 Windows 窗体而直接访问控件的基础 HWND 时，此功能才有可能不合需要。用户无法保持对 HWND 的引用，因为代码中所设置的属性可能使其无效。

### 3. 事件

Windows 窗体编程模型基于事件。当控件更改状态，如当用户单击按钮时，它引发一个事件。为了处理事件，应用程序为该事件注册一个事件处理方法。在 Visual Basic 中，有以下两种途径可以注

册事件处理方法。

- 如果使用 WithEvents 关键字声明控件变量，可以在方法的声明中使用 Handles 关键字，将该方法注册为事件处理方法。
- 可使用 AddHandler 在运行时注册事件处理方法。

下面的代码阐释注册事件处理方法的两种途径。

```
......
//Create the button
Button button1 = new Button() ;
button1.Location = new Point(256, 64);
button1.Size = new Size(120, 40);
button1.Text = "Click Me!";
this.Controls.Add(button1);

//Register the event handler
button1.Click += new System.EventHandler(button1_Click);
......

//The event handling method
private void button1_Click(object sender, EventArgs evArgs) {
   MessageBox.Show("Hello Windows Forms World!");
}
```

只对特定控件的特定事件调用某个事件处理方法。这使得可以避免窗体中出现处理所有控件的所有事件的单个方法。此功能还使代码更易理解和维护。而且，因为 Windows 窗体事件结构基于委托，所以事件处理方法是类型安全的并可以声明为私有的。此功能使编译器得以在编译时检测方法签名不匹配情况。它还使 Form 类的公共接口不与公共事件处理方法相混淆。读者可在.NET 框架 SDK 文档中找到关于委托的更多信息。

### 4. 事件类

每个事件有两个支持类，如下所示：

- 用于注册事件处理方法的 EventHandler 委托类。EventHandler 的签名指示事件处理方法的签名。
- 包含有关引发的事件的数据的 EventArgs 类。

EventHandler 的签名为：第一个参数包含对引发事件的对象（发送方）的引用，第二个参数包含关于该事件（EventArgs 的一个实例）的数据。例如，Button 上的 Click 事件使用以下事件处理程序。

```
public delegate void EventHandler(object sender, EventArgs e);
```

因此，Click 事件的任何事件处理方法都必须具有以下签名。

```
<access> void <name>(object sender, EventArgs evArgs)
```

对于强类型语言，如果事件处理方法的签名与委托签名不匹配，将发生编译时错误。

很多事件使用一般的 EventHandler 和 EventArgs 类。但是，一些事件要求包含针对所引发事件的类型的附加信息。例如，鼠标移动事件包括有关鼠标指针或鼠标按钮位置的信息。这些事件定义其自己的类，这些类必须从 EventHandler 和 EventArgs 类继承。例如，MouseDown 事件使用 MouseEventHandler 和 MouseEventArgs 类。