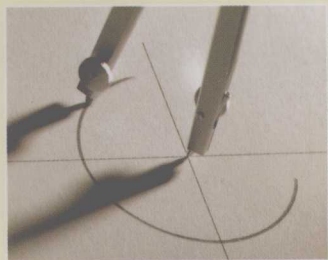


高 职 高 专 计 算 机 系 列

ASP.NET 程序设计 实验实训指导

吴海波 主 编
郭外萍 副主编



 人民邮电出版社
POSTS & TELECOM PRESS

ASP.NET 程序设计实训指导

高职高专计算机系列

ASP.NET 程序设计 实验实训指导

吴海波 主编

郭外萍 副主编

高职高专计算机系列

ASP.NET 程序设计实训指导

◆ 主 编 吴海波

副 主 编 郭外萍

责任编辑 刘雁斌

◆ 人民邮电出版社出版发行

地址 北京市丰台区右安门外大街 22 号

邮编 100061 电子邮件 312@ptpress.com.cn

网址 http://www.ptpress.com.cn

河北三河市新华印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 8.2

字数: 204 千字

印数: 1-3 000 册

 **人民邮电出版社**
POSTS & TELECOM PRESS

北 京

读者服务部热线: (010) 67170882 邮购部热线: (010) 67152523

图书在版编目 (CIP) 数据

ASP.NET 程序设计实验实训指导 / 吴海波主编. —北京:
人民邮电出版社, 2007.11
高职高专计算机系列
ISBN 978-7-115-16619-7

I. A… II. 吴… III. 主页制作—程序设计—高等学校:
技术学校—教学参考资料 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 114910 号

内 容 提 要

本书通过 15 个实验, 详细介绍了在 Visual Studio.NET 中设计基于 C# 编程语言的 Web 动态网页的方法和技巧。每个实验都给出了详细的指导, 可以使不同层次的读者由浅入深、逐步了解和掌握 ASP.NET。最后有一个综合实训项目, 详细讲述了使用 ASP.NET 开发电子商务网站的过程和方法。

本书的 15 个实验围绕电子商务网站中常见的功能网页设计而展开, 完成本书中的全部实验和实训项目的练习后, 读者即可以使用 ASP.NET 独立开发中等复杂程度的电子商务网站。本书是针对学习 ASP.NET 程序设计课程的高职学生编写的, 对于 ASP.NET 的爱好者, 也有较高的参考价值。

高职高专计算机系列

ASP.NET 程序设计实验实训指导

- ◆ 主 编 吴海波
副 主 编 郭外萍
责任编辑 刘雁斌
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
河北三河市海波印务有限公司印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 8.5
字数: 204 千字 2007 年 11 月第 1 版
印数: 1—3 000 册 2007 年 11 月河北第 1 次印刷

ISBN 978-7-115-16619-7/TP

定价: 14.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

编者的话

自 2000 年 7 月 Microsoft 公司展示 ASP.NET 以来, 这个新一代软件开发工具就备受 IT 业界的关注。它在应用程序开发领域开创了一个新时代, 并将推动下一代 Internet 的发展。作为 Microsoft.NET 平台的一部分, ASP.NET 是一个革命性的编程框架, 它实现了企业级 Web 应用程序的快速开发, 提供了最简便、最具有扩张能力的方式, 用于开发、部署和运行可以将任何浏览器或客户端设备作为目标的 Web 应用程序。目前, 使用 ASP.NET 来开发动态网页已成为 IT 业内的共识, 学习和使用 ASP.NET 的计算机软件爱好者和从业人员越来越多, 可见其技术的先进性和旺盛的生命力。

本书是与《ASP.NET 程序设计》(人民邮电出版社出版, 李德奇主编)一书配套的实验实训教程。针对《ASP.NET 程序设计》一书中的主要知识点设计了 15 个实验和 1 个实训项目。其结构安排如下。

实验 0 ASP.NET 编程环境的使用: 介绍了在 Visual Studio.NET 中编写 ASP.NET 应用程序的基本步骤。

实验 1 简单的 ASP.NET 网页设计: 练习简单的 ASP.NET 网页设计。

实验 2 HTML 语法和脚本应用: 练习 HTML 语法和 JavaScript 脚本的使用。

实验 3 ADO.NET 数据库访问技术应用: 练习在 Web Form 网页中访问数据库的技术。

实验 4 登录和注册页面设计: 练习 Label、Button 和 TextBox 控件的使用, 以及登录页面和注册页面的设计方法。

实验 5 信息浏览页面设计: 练习 DropDownList、ListBox 和 Image 控件的使用。

实验 6 信息收集页面设计: 练习 CheckBox、CheckBoxList、RadioButton 和 RadioButtonList 控件的使用。

实验 7 数据验证控件应用: 练习验证控件的使用。

实验 8 Repeater 控件应用: 练习 Repeater 控件的使用, 重点训练模板的编写。

实验 9 DataList 控件应用之一：初步练习 DataList 控件的使用。

实验 10 DataList 控件应用之二：练习 DataList 控件的使用，特别训练购物车技术。

实验 11 DataGrid 控件应用之一：初步练习 DataGrid 控件的使用。

实验 12 DataGrid 控件应用之二：练习 DataGrid 控件的使用，重点是分页和自定义分页导航。

实验 13 多网页项目设计：练习多网页项目的设计，处理页面间的关系。

实验 14 ASP.NET 内部对象应用：练习内部对象的使用，重点是 Session。

实验 15 TreeView 控件应用：练习 TreeView 控件的使用。

实训项目 电子商务网站设计实例——逸舞 PoPo 网上店铺系统开发：综合训练使用 ASP.NET 设计与实现 Web 应用项目开发的能力。

本书融合了作者多年的软件开发和教学经验以及近年来对 ASP.NET 的理解和实践。参加本书编写工作的有吴海波（主编）、郭外萍（副主编）、李德奇和李蓓蓓。在本书的策划和编写过程中，得到了彭勇及其他同行的帮助和支持，在此深表感谢。由于编者水平和经验有限，本书的不足和纰漏之处在所难免，恳请广大读者批评指正。

编者

2007年6月

目 录

ASP.NET 编程环境的使用	1
实验 1 简单的 ASP.NET 网页设计	8
实验 2 HTML 语法和脚本应用	11
实验 3 ADO.NET 数据库访问技术应用	14
实验 4 登录和注册页面设计	18
实验 5 信息浏览页面设计	24
实验 6 信息收集页面设计	30
实验 7 数据验证控件应用	35
实验 8 Repeater 控件应用	38
实验 9 DataList 控件应用之一	41
实验 10 DataList 控件应用之二	45
实验 11 DataGrid 控件应用之一	51
实验 12 DataGrid 控件应用之二	57
实验 13 多网页项目设计	61
实验 14 使用 ASP.NET 内部对象应用	66
实验 15 TreeView 控件应用	70
实训项目 电子商务网站设计实例——逸舞 PoPo 网上店铺平台	75

ASP.NET 编程环境的使用

一、起始页

启动 Visual Studio .NET 后，首先看到的是一个起始页，如图 0.1 所示，通过该页既可以使用本地计算机上的 .NET 项目，也可以访问本地计算机以外的信息和服务。起始页上的有些选项卡需要连接到 Internet 才能使用，这些选项卡提供了关于 Visual Studio .NET 的最新信息和最新下载的连接。此外，用户也可以简单地查找和注册 Web Service。若启动 Visual Studio .NET 后看不到起始页，可以单击帮助菜单中的【显示起始页】；若想隐去起始页，可以单击起始页上的【×】按钮关闭起始页。

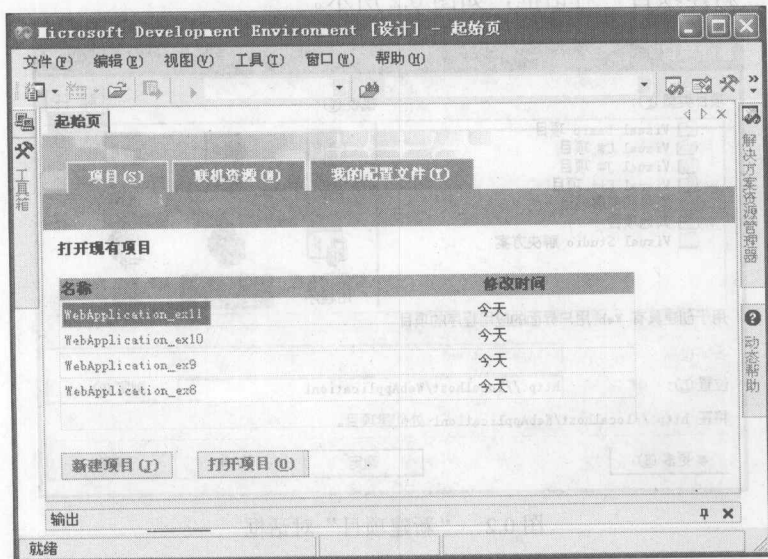


图 0.1 Visual Studio.NET 的起始页

1. “项目”选项卡

起始页的【项目】选项卡显示了最近修改过的 .NET 项目，将鼠标移动到某个项目的标题上，光标会变成手形，单击该项目即可打开它。

单击【新建项目(N)】链接按钮可以弹出“新建项目”对话框，通过该对话框可以建立一个 .NET 新项目。

单击【打开项目(O)】链接按钮可以弹出“打开项目”对话框,通过该对话框可以打开一个已经存在的.NET旧项目。

2. “联机资源”选项卡

当本地计算机连接到 Internet 后,可以使用【联机资源】选项卡来访问外部资源。例如,可以获得 Visual Studio.NET 新增功能的信息,联机搜索 MSDN 数据库,下载与 Visual Studio.NET 开发相关的内容和代码示例。特别地,可以从【XML Web Services】选项中搜索当前项目所需要的 XML Web Service。使用【Web 宿主】可以轻松地将自己的程序发布出去。

3. “我的配置文件”选项卡

通过该选项卡,开发人员可以按照自己的喜好来配置 Visual Studio .NET 的用户界面。通过选择不同的配置文件,开发人员可以根据自己对 C#、C++和 Visual Basic 等开发环境的熟悉程度,调整工具栏和其他窗口,也可以将键盘映射和窗口布局设定到 Visual Basic 6.0 或 Visual C++ 6.0 的风格。

二、创建 C#的 ASP.NET 应用程序项目

学习 ASP.NET 程序设计要从创建第一个项目开始。单击【文件】菜单→【新建】→【项目】,即可打开“新建项目”对话框,如图 0.2 所示。

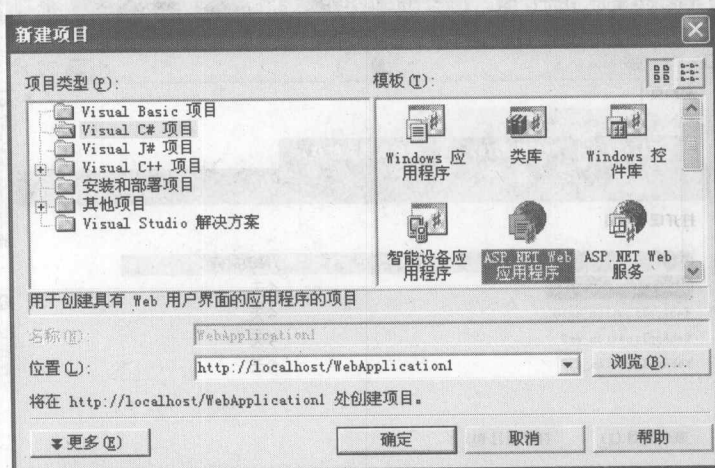




图 0.2 “新建项目”对话框

在左边的【项目类型】列表框中选择“Visual C#项目”,在右边的【模板】列表框中选择“ASP.NET Web 应用程序”。

在位置框中显示了项目的位置及名称“http://localhost/WebApplication1”,起步学习 ASP.NET 编程的人不要修改这个默认的位置及名称,待熟练后再去修改。单击【确定】按钮,一个 C#语言支持的 ASP.NET 应用程序项目就创建完毕。该项目默认的名称为“WebApplication1”,默认的虚拟路径为“http://localhost”,对应的物理路径为“C:\Inetput\wwwroot”。

如图 0.3 所示, 此时在 Visual Studio.NET 中显示了项目 WebApplication1 的开发状态。ASP.NET 应用程序可以在两种视图状态下进行设计: 设计视图和 HTML 视图。单击工作区下部的  设计  两个图标之一可以在两种视图间切换。在设计视图状态下工作时, 可以将工具箱中的控件拖到页面以组成用户界面; 在 HTML 视图状态下工作时, 可以通过修改页面的 HTML 标记和使用 HTML 语法来展示页面。两种设计方式是等价的。当前看到的是设计视图状态, 屏幕中不带网格的部分是应用程序的页面区, 其余部分为开发工具区。此后, 可以向应用程序的页面加入控件和代码, 进入编程状态。

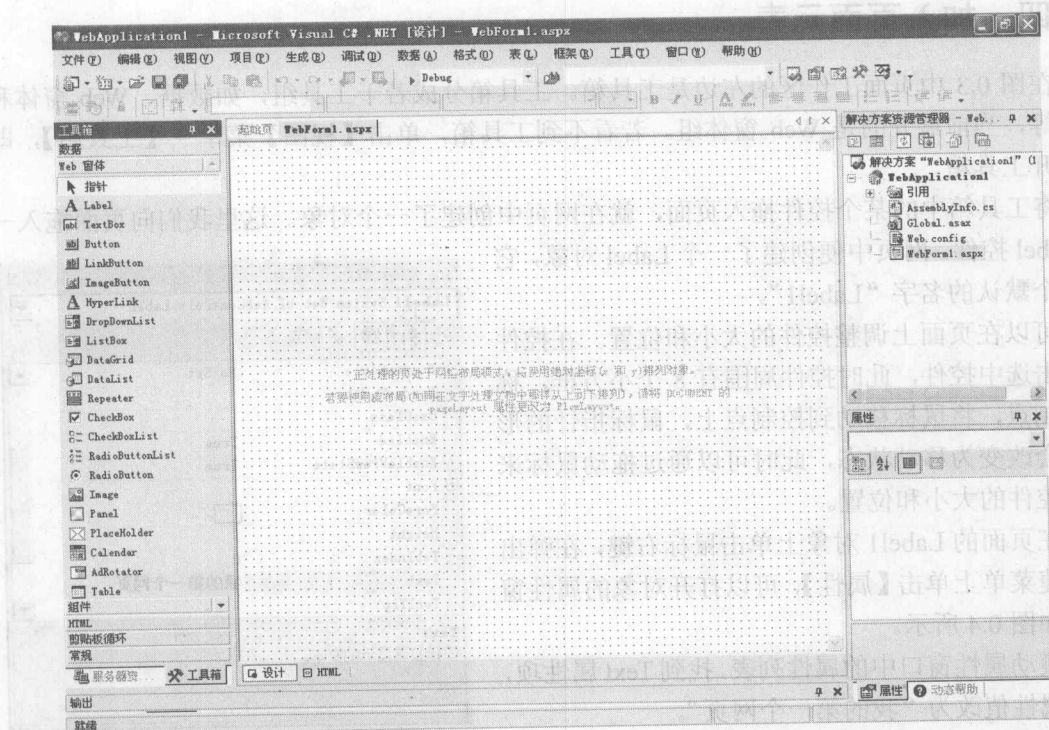


图 0.3 ASP.NET Web 应用程序的设计视图状态

三、设置网页设计状态的网格布局模式

ASP.NET Web 应用程序网页的设计视图有两种布局模式: 网格布局模式和流布局模式。当采用网格布局模式时, 页面区不会出现定位光标, 不可以在网页上输入文字, 页面上的所有内容必须使用控件来表现, 并且使用绝对坐标排列对象; 采用流布局模式时, 页面区有定位光标, 此时可以像在文字处理软件中一样对文字和控件进行处理, 这是两种布局模式在设计阶段的区别。在程序运行期间, 采用网格布局模式设计的网页在浏览器中看到的效果与设计视图完全相同, 由于该模式使用绝对坐标排列对象, 页面上的对象的位置是固定的, 不管这些对象是否可见。当某些对象不可见时, 这些对象的位置会空在那里。采用流布局模式时, 对象的位置是流动的, 当某些对象在某时刻不可见时, 后面的对象会依次上前占据那些空位, 使页面更加紧凑。当那些不可见对象变得可见时, 其后面的对象会依次后退, 以便各就其位。

程序员一般采用网格布局模式, 使网页在设计状态与运行状态的布局相同, 只有需要在

网页上动态创建对象致使网页元素时多时少、或元素的可见性时常发生变化、或网页上的文本时长时短的情况下才采用流布局模式，这样不会使网页运行时出现大块空白而开天窗，或者由于某些控件的文本增多而覆盖掉它后面的控件。

网页在设计状态默认为网格布局模式，若要改变网页的布局模式，在网页上单击右键，然后在属性对话框中改变网页的布局模式即可。目前我们不去改变网页的布局模式，仍然采用网格布局模式，其初始的状态就是在图 0.3 中看到的效果。

四、加入页面元素

在图 0.3 中页面工作区的左边是工具箱。工具箱分成若干工具组，如数据、Web 窗体和组件等，当前展开的是 Web 窗体组。若看不到工具箱，单击【视图】菜单→【工具箱】，即可打开工具箱。

将工具箱中的某个控件拖入页面，就在网页中创建了一个对象。这里我们向页面拖入一个 Label 控件，网页中便创建了一个 Label 对象，它有一个默认的名字“Label1”。

可以在页面上调整控件的大小和位置。在控件上单击选中控件，此时控件周围有 8 个小方框，称为控制点，将鼠标移动到控制点上，鼠标指针的形状就会改变为移动状态，此时可以通过拖动鼠标来改变控件的大小和位置。

在页面的 Label1 对象上单击鼠标右键，在弹出的快捷菜单上单击【属性】，可以打开对象的属性窗口，如图 0.4 所示。


滚动属性窗口中的属性列表，找到 Text 属性项，将其属性值改为“我的第一个网页”。

至此，一个最简单的网页就设计好了。



图 0.4 对象的属性窗口

五、编译和运行程序

单击【调试】菜单→【启动】或工具栏中的蓝三角，或者按下 F5 键，开始对程序进行调试。也可以单击【调试】菜单→【开始执行】，或者按下 Ctrl+F5 键，从而启动应用程序。如果程序没有错误，这两种方式都可以在浏览器中看到刚才设计的网页，如图 0.5 所示。

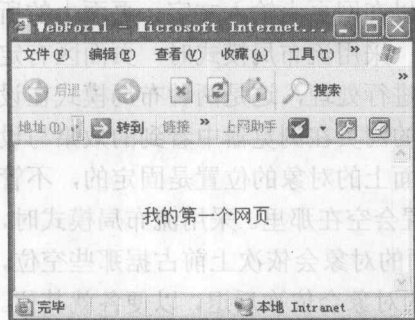


图 0.5 从浏览器中看到的网页

在【调试】菜单中单击【启动】或【开始执行】都可以编译程序。如果程序存在错误，则会在输出窗口显示错误信息，程序不能运行；如果程序没有错误，则会启动浏览器，在浏览器中输出网页。不同的是，【启动】是处于调试状态，一次只能调试一个程序；而【开

始执行】是处于执行状态，可以同时执行多个程序。

六、程序排错

设计程序时，错误总是难免的，即使是有经验的程序员也会犯错，学会调试改错才是最重要的。

程序错误通常有 3 种类型：语法错误、运行错误以及算法和逻辑错误。语法错误表示代码违反了编译器的语法规约，代码不能被编译器理解。运行错误是程序中试图完成不允许的操作。算法和逻辑错误是程序运行后得不到正确的结果。

1. 语法错误

当编译器不能理解代码行时，就发生语法错误。最常见的语法错误是关键字拼写错误、缺少语句结束符“;”、括号（圆括号（）、方括号[]和花括号{}）缺少一边、引用未定义的标识符或者使用一个不正确的语句结构等。

当发生语法错误时，编译器会停止编译，并弹出一个错误报告，如图 0.6 所示。

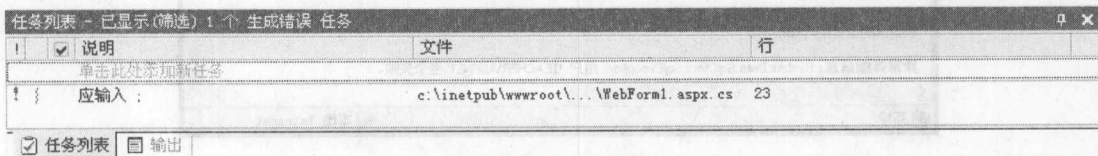


图 0.6 编译信息报告

在错误报告信息框中的某条错误信息上双击，光标会跳到编辑窗口中与之对应的出错语句上，以便进行修改。

修正语法错误的最好办法不是在出错之后，而是在代码编辑阶段就要把好关，以免语法错误太多，反复编译耽误时间。借助于 C# 编辑器的强大功能，有很多语法错误在编辑阶段就可以发现并予以纠正。

- **关键字拼写错误。**关键字在编辑器中是以蓝色显示的，若在编辑时某关键字未以蓝色显示，必定是拼写不正确，需及时改正。关键字通常用来表示数据类型和构成语句结构等，具有特定的含义。

- **缺少语句结束符“;”。**这个错误源于两点：忘记输入；输入的是中文的“;”。代码中最好是每一行写一个语句，这样借助于编辑器能够自动调整语句缩格功能，也能及时发现是否少了或多了分号。顺序结构中，若回车后光标不与上一行对齐，必定是上一行缺少分号，应及时补上；选择结构和循环结构中，回车后本该缩进的语句没有缩进，必定是上一行多了分号，应及时去掉。

- **各种括号缺少一边。**解决办法是，输入括号时，左右两边一并输入，然后再去填充括号中的内容。这样做似乎会降低输入速度，其实不然。当层次很多的花括号{}缺少一边时，等到编译发现后再去添加，不但浪费许多查找的时间，而且有时还会由于加入括号的位置不对而导致程序的逻辑混乱。

- **标识符错误。**标识符有两种来源：一是自定义，这时应检查该标识符此前是否定义

过, 或是否拼写错误; 二是来自 .NET 基类库, 这时必须检查是否用 `using` 关键字导入过它的名空间。标识符中间带有圆点操作符是最容易处理的, 若键入 “.” 后不跟随上下文菜单, 说明圆点前拼写有错; 若键入 “.” 后跟随上下文菜单, 则 “.” 后的内容让上下文菜单帮你写。

2. 运行错误

运行错误是在程序运行后出错, 这时会在浏览器中报告出错信息, 如图 0.7 所示。

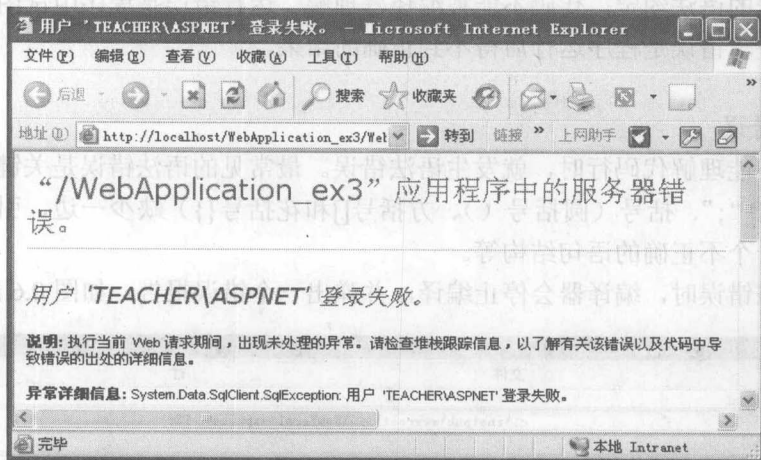


图 0.7 异常报告

有些异常是很容易纠正的, 例如“无法找到资源”等。但有些异常的原因就难于查找, 必须使用调试器仔细分析。有些异常可以预见, 但不可避免, 例如, 程序需要到 U 盘读取数据, 用户未准备好 U 盘必定导致异常。这时就必须采用异常处理的机制来编写程序。

3. 算法和逻辑错误

算法和逻辑错误是指程序能很好地运行, 没有任何错误报告和异常出现, 但程序得到的结果却不是程序员所需要的。

算法错误是指由于数学模型出了问题而导致的错误, 这时需要检查算法。

逻辑错误是程序代码逻辑混乱造成的, 多数情况是由选择结构或循环结构中的括号配对的错乱引起的。前述的花括号配对输入很重要, 养成良好的习惯会节约很多的调试时间。有些编辑器的 `}` 会自动成对出现, 也就是说, 当输入左 “`{`” 后, 右 “`}`” 会自动出现。C# 编辑器不采用这个功能, 但不管怎样同时输入 `{}` 的两边对于编程人员来说是个好习惯。

逻辑错误有时也出现在多余的分号上, 例如下段代码运行得很好, 在控件 `Label1` 上看到的结果是 101, 但实际上编程者想要的结果是 1 到 100 的累加和 5050。

```
int s=0;
for(int i=1;i<=100;i++);
s+=i;
Label1.Text=s.ToString();
```

错误出在 for 行多了一个“;”。其实,这个错误在编辑阶段就可以发现,当从 for 行回车后,光标未缩进而与 for 行对齐,表明编辑器认为这两行是顺序结构而不是循环结构,这时编程人员必须找到原因并予以改正,而不是让程序运行得出稀奇古怪的结果后再去找原因,要知道在一个十分复杂的程序中查找这样的错误是很费事的。

养成好的程序书写风格对于提高程序质量十分重要,建议在 C# 编辑器中编程时,编程人员不要用强制方法调整程序行的左右位置,只要程序逻辑正确,编辑器会自动理清阶梯缩进关系。如果程序行输入后不能按照逻辑关系排列层次,则必定是在语法或括号等环节出了问题。

一、目录索引

- 了解 Web Form 网页的构成
- 了解 ASP.NET 程序的处理过程
- 掌握事件和事件处理程序
- 掌握 Page 类的常用属性、事件和方法

二、主要内容索引

创建一个 ASP.NET Web 应用程序项目,在默认的网页 WebForm1 中加入一个 Label 标签和两个 Button 控件。当页面初次加载时,标签中显示“我的第一次实验”,当单击【实验】按钮时,标签中显示“页面回传了”,单击【重置】按钮时,标签中显示“重置”。

单击【重置】按钮时,标签中显示“重置”,单击【实验】按钮时,标签中显示“页面回传了”,单击【重置】按钮时,标签中显示“重置”。

单击【重置】按钮时,标签中显示“重置”,单击【实验】按钮时,标签中显示“页面回传了”,单击【重置】按钮时,标签中显示“重置”。

单击【重置】按钮时,标签中显示“重置”,单击【实验】按钮时,标签中显示“页面回传了”,单击【重置】按钮时,标签中显示“重置”。

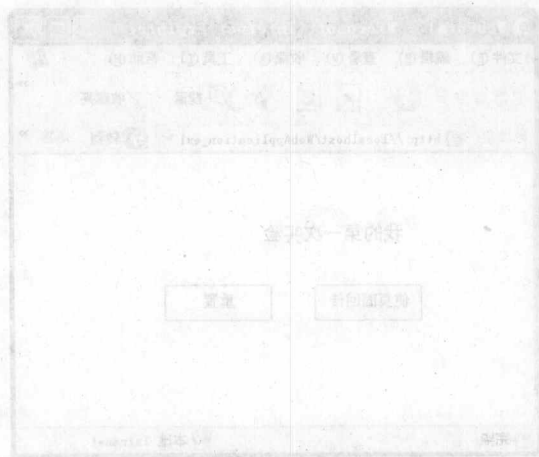


图 1.1-1 实验结果图

实验 1 简单的 ASP.NET 网页设计

一、实验目的

- 了解 Web Form 网页的构成
- 了解 ASP.NET 程序的处理过程
- 掌握事件和事件处理程序
- 掌握 Page 类的常用属性、事件和方法

二、实验内容和要求

创建一个 ASP.NET Web 应用程序项目，在默认的网页 WebForm1 中加入一个 Label 标签和两个 Button 按钮。当页面初次加载时，标签中显示“我的第一次实验”，当单击【使页面回传】按钮后，标签中显示“页面回传了”，单击【重置】按钮时，标签中显示“页面回传并被重置”。

不要为【使页面回传】按钮编写任何后台代码达到上述功能，以体会页面回传机制。程序运行后的效果如图 1.1 所示。



图 1.1 (a) 页面的初始状态

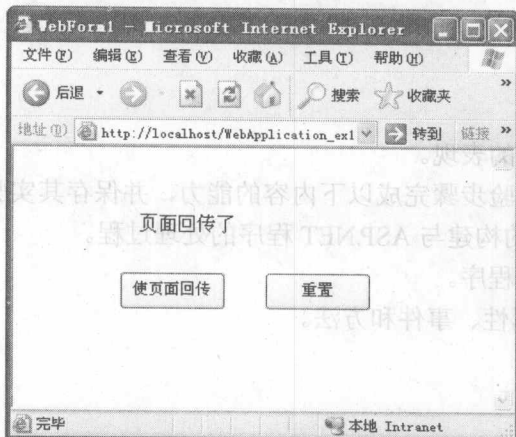


图 1.1 (b) 单击【使页面回传】按钮后的页面效果



图 1.1 (c) 单击【重置】按钮后的页面效果

三、实验步骤

(1) 创建 ASP.NET 的 Web 应用程序项目“WebApplication_ex1”。

(2) 在默认页面“WebForm1”加入以下控件。

- 1 个 Label 标签。属性 ID 为默认值“Label1”，属性 Text 为空，其他属性为默认值。
- 两个 Button 按钮。属性 ID 为默认值“Button1”和“Button2”，属性 Text 分别为“使页面回传”和“重置”，其他属性都为默认值。

(3) 为页面的 Page_Load 事件编写后台代码，实现页面加载时标签上显示“我的第一次实验”，如图 1.1 (a) 所示；页面回传后标签上显示“页面回传了”，如图 1.1 (b) 所示。

【提示】在该事件中通过判断属性“IsPostBack”来确定页面是否回传，以便使标签上显示不同的文本。

(4) 为“重置”按钮的 Button2_Click 事件编写后台代码，实现当它被单击时，标签 Label1 上显示“页面回传并被重置”。

四、考核内容

- (1) 考核学生实验时的表现。
- (2) 考核学生依照实验步骤完成以下内容的的能力，并保存其实验成果。
 - Web Form 网页的构建与 ASP.NET 程序的处理过程。
 - 事件和事件处理程序。
 - Page 类的常用属性、事件和方法。

五、评价标准

实验名称	实验模块	模块成果	学生课堂表现	学生互评
实验 1 简单的 ASP.NET 网页	1. Web Form 网页的构建与 ASP.NET 程序的处理过程 2. 事件和事件处理程序 3. Page 类的常用属性、事件和方法	教师根据学生各模块的完成情况以 A、B、C、D 档给出成绩 A 档：优秀 B 档：良好 C 档：及格 D 档：不及格	教师根据学生课堂表现情况以 A、B、C、D 档给出成绩 A 档：优秀 B 档：良好 C 档：及格 D 档：不及格	学生互评，以 A、B、C、D 档给出成绩 A 档：优秀 B 档：良好 C 档：及格 D 档：不及格

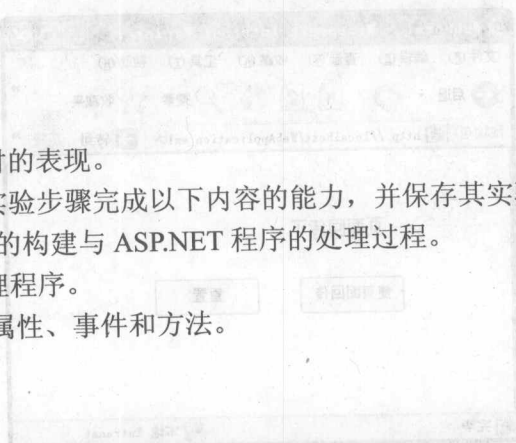


图 1.1 (c) 单击【重置】按钮后的页面效果

三、实验步骤

- (1) 创建 ASP.NET 的 Web 应用程序项目“WebApplication_ex1”。
- (2) 在默认页面“WebForm1”加入以下控件。
 - 1 个 Label 标签，属性 ID 为默认值“Label1”，属性 Text 为“空”，其他属性都为默认值。
 - 两个 Button 按钮，属性 ID 为默认值“Button1”和“Button2”，属性 Text 分别为“重置”和“重置”，其他属性都为默认值。
- (3) 为页面的 Page_Load 事件编写后台代码，实现页面加载时标签上显示“我的第一次实验”，如图 1.1 (a) 所示；页面回传后标签上显示“页面回传了”，如图 1.1 (b) 所示。
 - 【提示】在该事件中通过判断属性“IsPostBack”来确定页面是否回传，以便使标签上显示不同的文本。
- (4) 为“重置”按钮的 Button2_Click 事件编写后台代码，实现单击 Button2 标签 Label1 上显示“页面回传并重置”。

