

2009-2010 学年第二学期

基于项目的 C#实训指导书

——基于 VC#的浏览器程序设计

熊 坤

郑州轻工业学院
计算机与通信工程学院网络工程系

2010 年 6 月 2 日

一 实训目的

通过实训使学生在掌握 VC#编程理论的基础上，推动学生初步掌握基于 VC#的网络浏览器程序的编程方法，为今后的网络协议编程和 Web 服务编程打下基础。

二 实训题目

使用 C#语言在 VS2005 平台上编程实现基于浏览器程序。

三 实训要求

1. 能够熟练使用 VS2005 编程平台，熟悉平台菜单栏、工具栏和各种窗口的使用，掌握不同项目类型创建步骤和平台类库的使用。
2. 熟练掌握 C#语法与程序结构，能够熟练创建程序功能所需的类、对象、接口等，并通过派生与继承实现代码重用，通过事件与消息实现对象之间的交互。
3. 熟练掌握权利浏览器程序所需的各种部件的使用，能够正确使用菜单、工具栏和窗口控件完成浏览器程序界面设计，并能够完成这些部件的相应响应函数代码编写，完成部件与函数之间的消息映射。
4. 按照软件工程的要求的功能分析、程序设计、代码编写、程序调试、功能测试等步骤，完成实训项目要求。
5. 提交“基于 VC#的浏览器器程序设计”设计方案与程序源代码。

四 实训步骤

4.1 掌握浏览器工作原理

浏览器工作原理如图 4-1 所示。

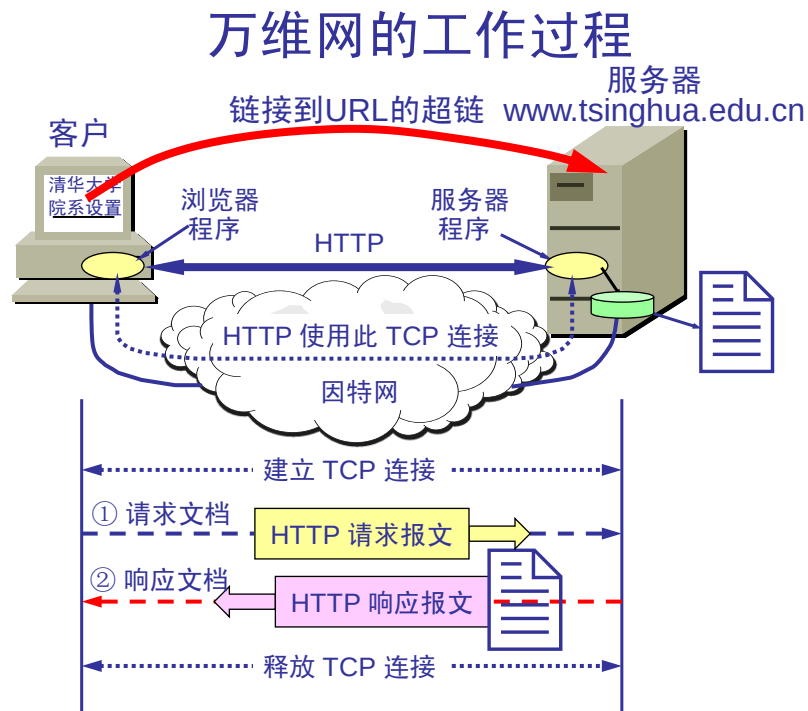


图 4-1 浏览器工作原理

- (1) 浏览器分析超链指向页面的 URL。
- (2) 浏览器向 DNS 请求解析 `www.tsinghua.edu.cn` 的 IP 地址。
- (3) 域名系统 DNS 解析出清华大学服务器的 IP 地址。
- (4) 浏览器与服务器建立 TCP 连接
- (5) 浏览器发出取文件命令：`GET /chn/yxsx/index.htm`。
- (6) 服务器 给出响应，把文件 `index.htm` 发给浏览器。
- (7) TCP 连接释放。
- (8) 浏览器显示文件 `index.htm` 中的所有文本

4.2 学习浏览器功能结构

浏览器程序功能结构如图 4-2 所示。

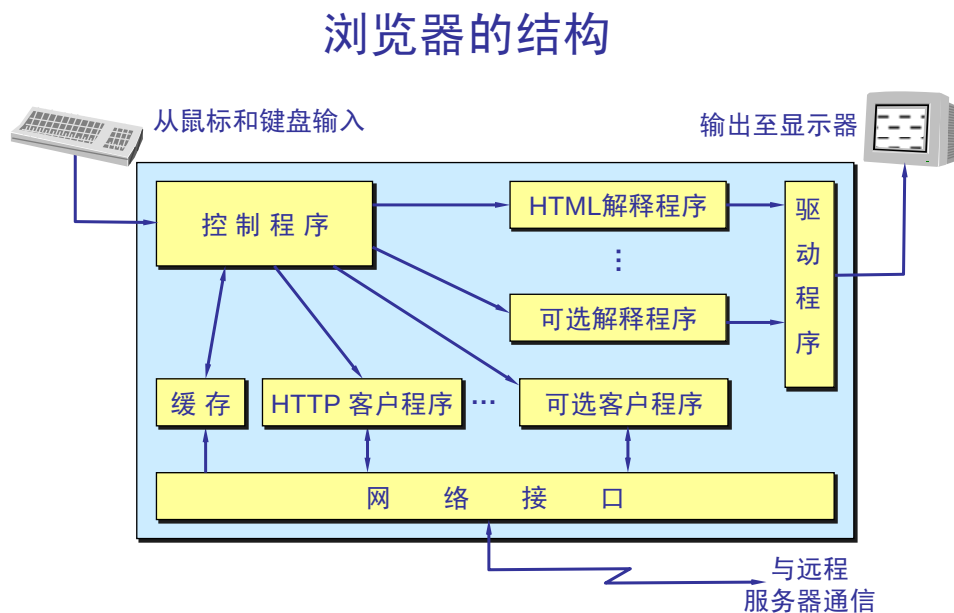


图 4-2 浏览器功能结构

- 1) 浏览器有一组客户、一组解释程序，以及管理这些客户和解释程序的控制程序。
- 2) 控制程序是其中的核心部件，它解释鼠标的点击和键盘的输入，并调用有关的组件来执行用户指定的操作。
- 3) 例如，当用户用鼠标点击一个超链的起点时，控制程序就调用一个客户从所需文档所在的远地服务器上取回该文档，并调用解释程序向用户显示该文档
- 4) HTML 解释程序是必不可少的，而其他的解释程序则是可选的。
- 5) 解释程序把 HTML 规格转换为适合用户显示硬件的命令来处理版面的细节。

- 6) 许多浏览器还包含 FTP 客户，用来获取文件传送服务。
- 7) 一些浏览器也包含电子邮件客户，使浏览器能够发送和接收电子邮件
- 8) 浏览器将它取回的每一个页面副本都放入本地磁盘的缓存中。
- 9) 当用户用鼠标点击某个选项时，浏览器首先检查磁盘的缓存。
若缓存中保存了该项，浏览器就直接从缓存中得到该项副本而不必从网络获取，这样就明显地改善浏览器的运行特性。
- 10) 但缓存要占用磁盘大量的空间，而浏览器性能的改善只有在用户再次查看缓存中的页面时才有帮助。
- 11) 许多浏览器允许用户调整缓存策略。

4.3 浏览器设计

(1) 浏览器功能设计

浏览器功能设计主要确定设计好的浏览器需要具备哪些功能，以及这些功能是以什么方式进行实现，这个步骤相当于软件设计中的“需求分析”，其中必须保证基本功能的具备，即浏览器必须能够浏览 Web 页面。

(2) 浏览器界面设计

浏览器界面设计主要确定浏览器的页面布局，设定浏览器窗口上设置哪些功能菜单项、设置哪些功能按钮、设置哪些状态栏，它们各自的位置如何、属性如何、对应的变量如何等等。

(3) 浏览器代码设计

浏览器代码设计主要是确定对上述的设计的功能在代码上怎样实现，包括选用的程序语言和编程平台选择、程序代码的编写、程序代码的调试、程序功能的测试等等。

4.4 浏览器编程设计平台

- (1) VS2005 编程平台
- (2) WebBrowser 类及其使用
- (3) 一些浏览器程序的代码分析

4.5 浏览器开发步骤

- (1) 熟悉 VS2005 平台，建立项目（半天）

打开 VS2005 选择“创建项目”，进入“新建项目”，如图 4-1 所示。

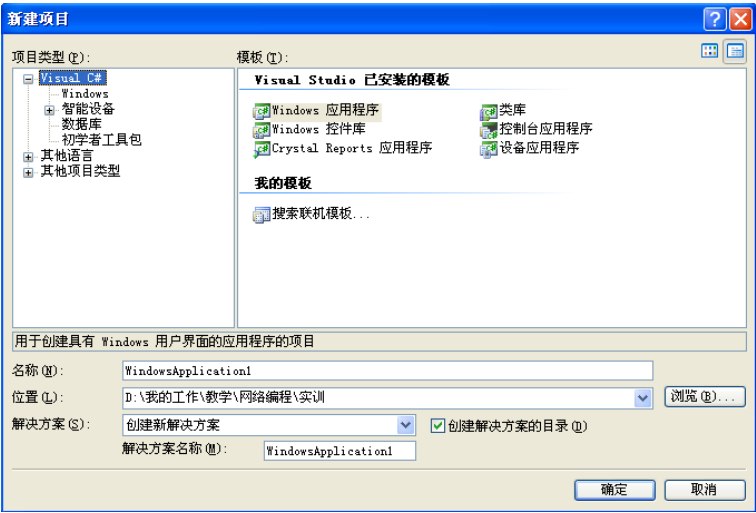


图 4-1 “新建项目”对话框

在“新建项目”对话框的项目类型窗口中选中“Visual C#”作为项目开发语言，在模板窗口中选中“Windows 应用程序”作为项目开发模板，在“名称”、“位置”编辑框中输入自己设定的项目名字和项

目存储位置，本讲义使用“CSharp 浏览器”作为项目名称，如图 4-2 所示。

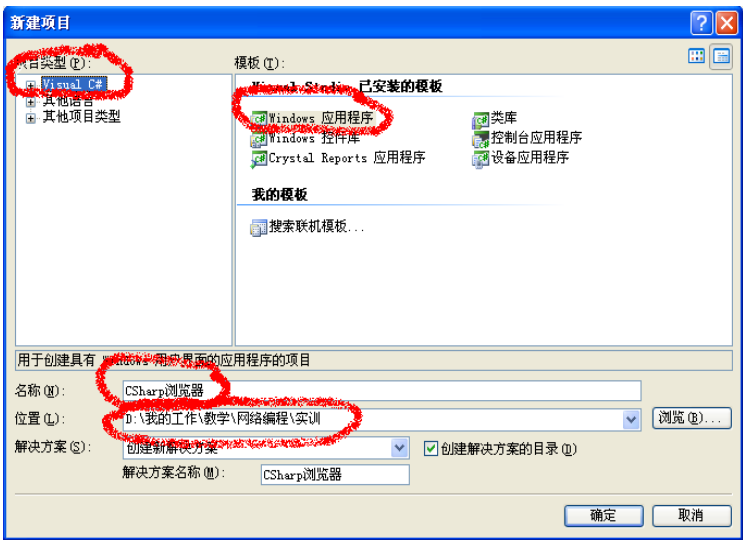


图 4-2 输入新建项目的属性

完成输入内容后，单击“确定”按钮，进入项目 VC#开发窗口“Form.cs [设计]”，如图 4-3 所示。

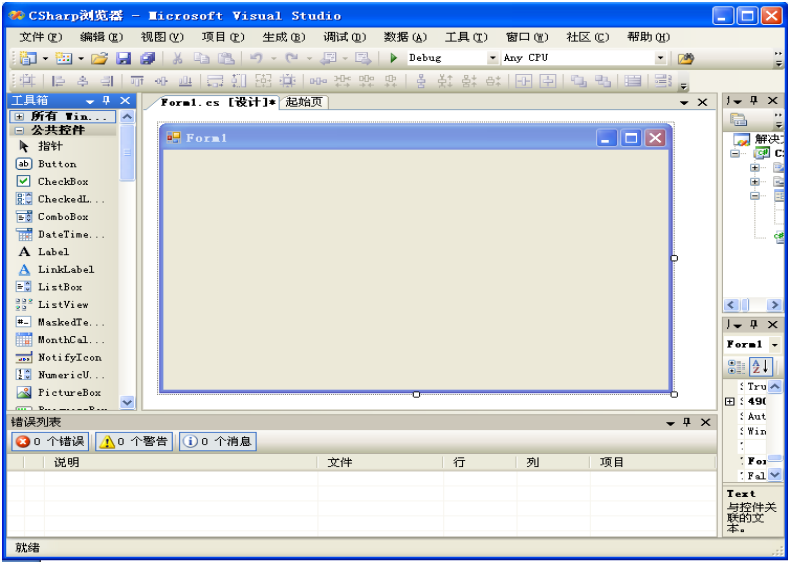


图 4-3 VC#开发窗口“Form.cs[设计]”

(2) 界面设计

① 设计菜单栏如图 4-4 所示。(半天)

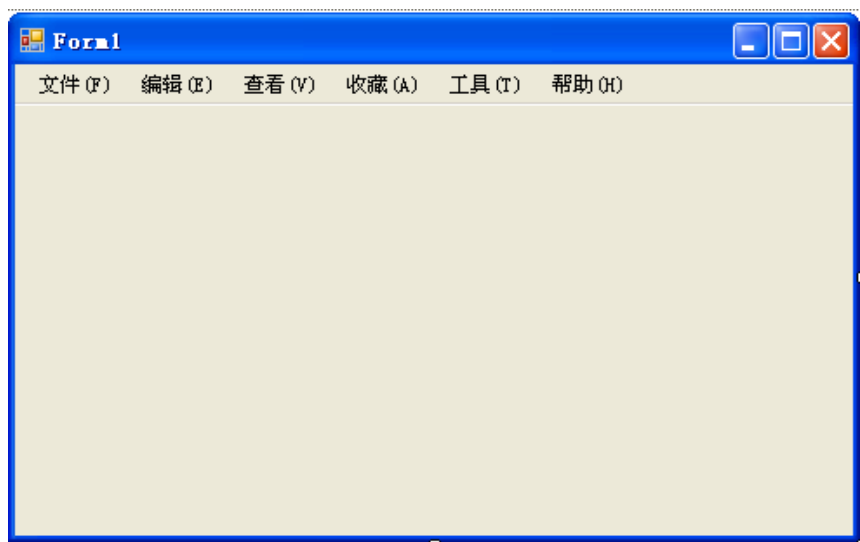


图 4-4 设计菜单栏

设计“文件”菜单如 4-5 所示。

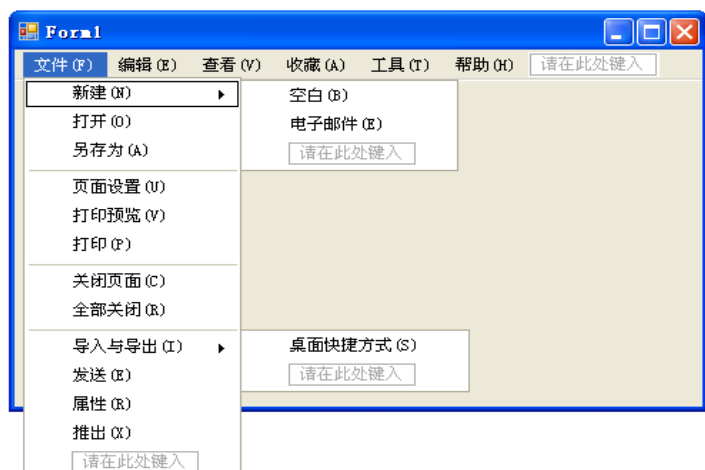


图 4-5 设计“文件”菜单

设计“编辑”菜单如图 4-6 所示。

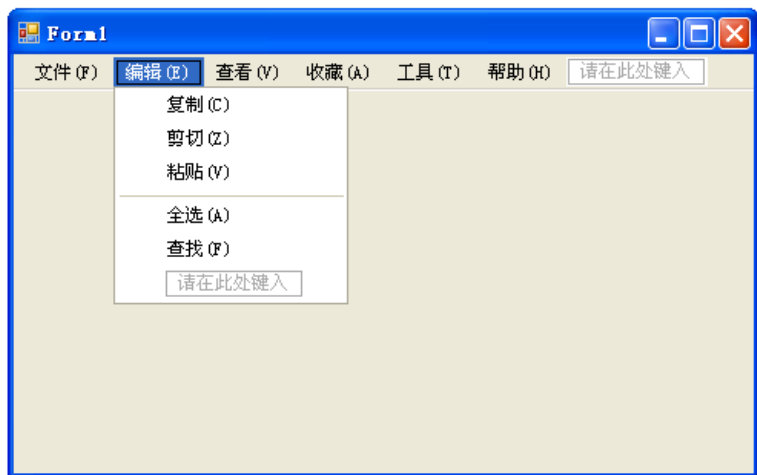


图 4-6 设计“编辑”菜单

设计“查看”菜单如图 4-7 所示。

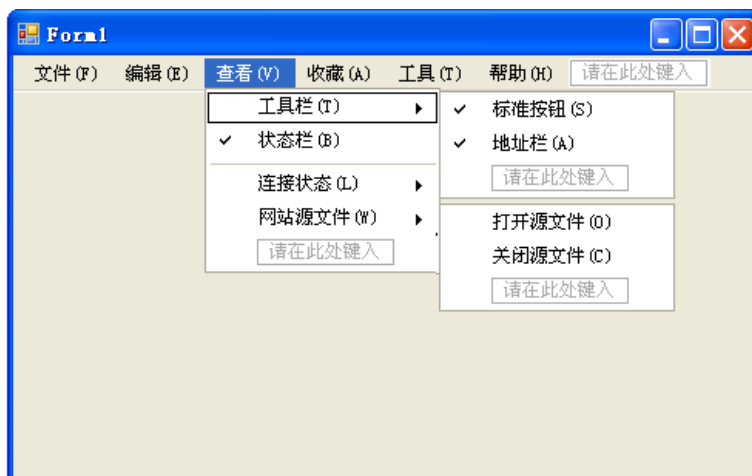


图 4-7 设计“查看”菜单

设计“收藏”菜单如图 4-8 所示。



图 4-8 设计“收藏”菜单

设计“工具”菜单如图 4-9 所示。



图 4-9 设计“工具”菜单

② 设计工具栏、用户区和状态栏如图 4-10 所示（半天）

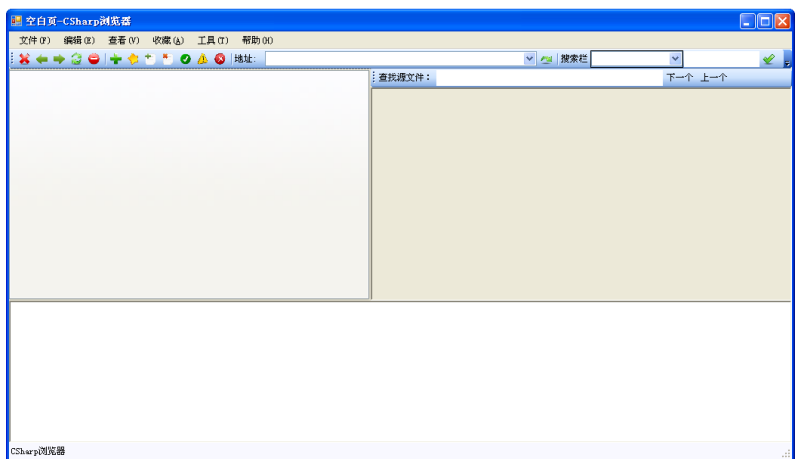


图 4-10 工具栏和状态栏

（3）功能代码编辑

① 初始化代码

双击“Form”设计页面进入代码编写页面，如图 4-11 所示。

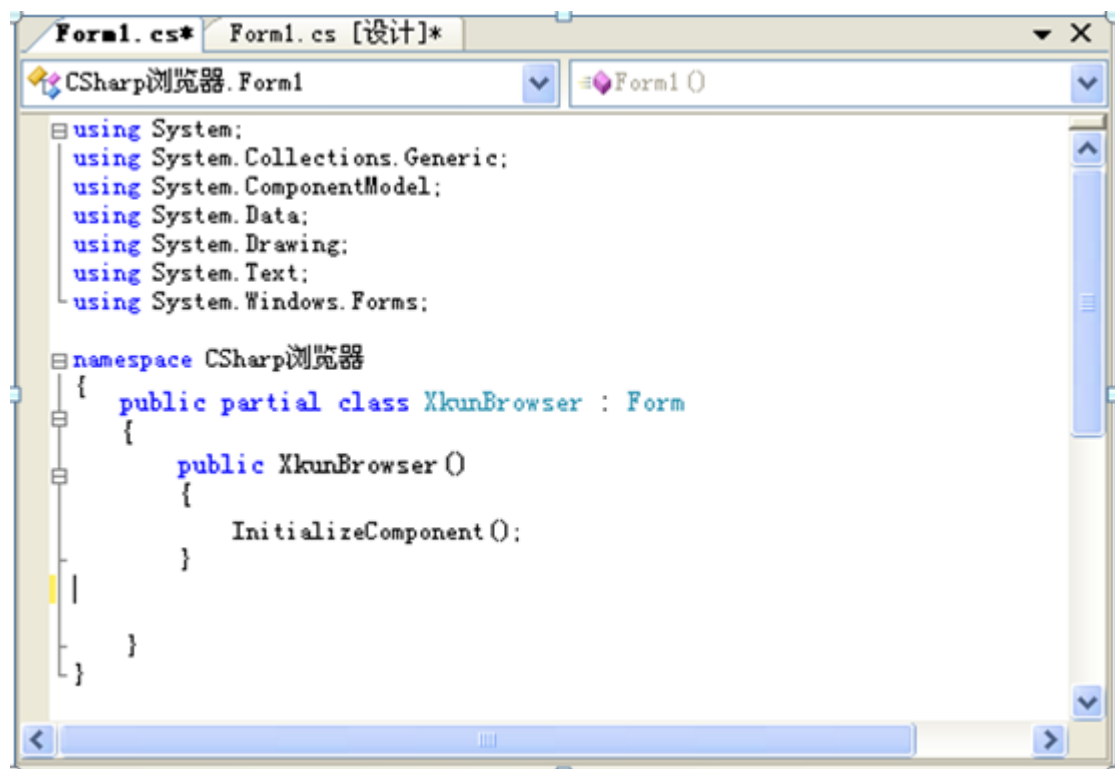


图 4-11 代码编写页面

初始化页面代码如下：

```

////////////////////////////////////
//                               CSharp浏览器                               //
//                               //                                           //
//                               版权所有-熊坤编程                           //
//                               //                                           //
//                               2009-8-28日                               //
////////////////////////////////////

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Threading;
using mshtml;
using System.Net;
using System.Runtime.InteropServices;
using SHDocVw;

//*****//
//                               此类是一个主窗口类                               //

```



```

}

//*****//
//                                     //
//             启动程序运行时             //
//                                     //
//*****//
private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        this.WindowState = FormWindowState.Maximized;           //窗口最大化显示
        Combo_FindType.Text = Combo_FindType.Items[0].ToString(); //初始化搜索类型
        Thread thdProcess = new Thread(new ThreadStart(Thread_NewNullTabPage)); //启动生成空白页线程

        thdProcess.Start();
        cishu.Add(0);           //初始化，对cishu列表添加添加一个元素（整数0）
        timer1.Start();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}

```

② 编写基础功能类

编辑浏览器webBrowser控件处理浏览器基本功能:

```

////////////////////////////////////
//             CSharp浏览器             //
//                                     //
//             版权所有-熊坤编程             //
//                                     //
//             2009-8-28日             //
//                                     //
////////////////////////////////////
//*****//
//                                     //
//             此类是一个继承TabPage的页面类，包含webBrowser控件             //
//                                     //
//*****//
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Text;
using System.Windows.Forms;

namespace CSharp浏览器
{
    class CustomTabPage : TabPage
    {
        public System.Windows.Forms.WebBrowser webBrowser;
        private string _Navigate;
        private XkunBrowser pParentWin = null;
        //*****//
        //                                     //
        //          初始化CustomTabPage类          //
        //                                     //
        //*****//
        public CustomTabPage(XkunBrowser pParentWinForm)
        {
            _Navigate = "about:blank";
            pParentWin = pParentWinForm;
        }
        //*****//
        //                                     //
        // 把WebBrowser的Navigate方法转变为TabPage类 //
        // 的Navigate属性                               //
        //                                     //
        //*****//
        public string Navigate
        {
            get { return _Navigate; }
            set
            {
                _Navigate = value;
                this.Text = value;
                if (webBrowser != null)
                {
                    if (String.IsNullOrEmpty(_Navigate)) return;
                    if (_Navigate.Equals("about:blank")) return;
                    if (!_Navigate.StartsWith("http://")) _Navigate = "http://" + _Navigate;
                    try
                    {
                        webBrowser.Navigate(new Uri(_Navigate));
                    }
                    catch (System.UriFormatException)
                    {

```

```

        return;
    }
}

}

//*****//
//                                     //
// 初始化WebBrowser控件，并添加到TabPage控件中  //
//                                     //
//*****//
protected override void OnCreateControl()
{
    //添加webBrowser 相关的响应事件
    webBrowser = new System.Windows.Forms.WebBrowser();
    webBrowser.Dock = System.Windows.Forms.DockStyle.Fill;
    webBrowser.ScriptErrorsSuppressed = true;
    webBrowser.DocumentCompleted += new
System.Windows.Forms.WebBrowserDocumentCompletedEventHandler(webBrowser_DocumentCompleted);
    webBrowser.NewWindow += new
System.ComponentModel.CancelEventHandler(webBrowser_NewWindow);
    webBrowser.Navigating += new
System.Windows.Forms.WebBrowserNavigatingEventHandler(webBrowser_Navigating);
    webBrowser.StatusTextChanged += new EventHandler(webBrowser_StatusTextChanged);
    webBrowser.Navigated += new
System.Windows.Forms.WebBrowserNavigatedEventHandler(webBrowser_Navigated);
    webBrowser.ProgressChanged += new
WebBrowserProgressChangedEventHandler(webBrowser_ProgressChanged);
    webBrowser.CanGoForwardChanged += new
EventHandler(webBrowser_CanGoForwardChanged);
    webBrowser.CanGoBackChanged += new EventHandler(webBrowser_CanGoBackChanged);
    webBrowser.DocumentTitleChanged += new
EventHandler(webBrowser_DocumentTitleChanged);
    //显示webBrowser
    webBrowser.Visible = true;
    //把webBrowser控件添加到TabPage控件上
    this.Controls.Add(webBrowser);
    base.OnCreateControl();
}

//*****//
//                                     //
//          加载中事件          //
//                                     //
//*****//
void webBrowser_Navigating(object sender,

```

```

System.Windows.Forms.WebBrowserNavigatingEventArgs e)
{
    ///当点击网页跳转时,使地址栏重新获得跳转后的地址
    string strurl = this.webBrowser.Document.ActiveElement.GetAttribute("href");
    //string strurl = webBrowser.StatusText;
    if (!strurl.Contains("http://"))
    {
        strurl = "http://" + strurl;
    }
    this.pParentWin.urlBox.Text = strurl;
    this.pParentWin.toolStripProgressBar1.Visible = true;
    //this.Text = strurl;
    //this.pParentWin.Text = strurl;
}

//*****//
//                                     //
//             打开新窗口事件             //
//                                     //
//*****//

void webBrowser_NewWindow(object sender, System.ComponentModel.CancelEventArgs e)
{
    //阻止用IE新开窗口
    e.Cancel = true;
    webBrowser.AllowNavigation = false;
    //为父容器添加新的选项卡
    CustomTabPage tabPage = new CustomTabPage(pParentWin);
    this.pParentWin.tabControl1.TabPages.Add(tabPage);
    this.pParentWin.tabControl1.SelectedTab = tabPage;
    //获得当前截获的新开窗口的URL
    string url = webBrowser.StatusText;
    //如果非链接而是表单提交及脚本的话,此时会取不到值,所以需要下面的判断
    if (!string.IsNullOrEmpty(url))
    {
        tabPage.Navigate = url;
        this.pParentWin.urlBox.Text = url;
    }
    webBrowser.AllowNavigation = true;
}

//*****//
//                                     //
//             文档加载完事件             //
//                                     //
//*****//

void webBrowser_DocumentCompleted(object sender,

```



```

System.Windows.Forms.WebBrowserDocumentCompletedEventArgs e)
{
    //页面加载完成后给窗口及选项卡加标题
    this.pParentWin. toolStripProgressBar1.Visible = false;
}
//*****//
//                                     //
//          页面状态事件                //
//                                     //
//*****//
void webBrowser_StatusTextChanged(object sender, EventArgs e)
{
    this.pParentWin. toolStripStatusLabel1.Text = webBrowser.StatusText;
    if (this.pParentWin. richTextBox1.Visible)
    {
        if (webBrowser.StatusText != "" && webBrowser.StatusText != "完毕")
        {
            this.pParentWin. richTextBox1.AppendText(webBrowser.StatusText + "\n");
        }
    }
}
//*****//
//                                     //
//          文档加载前事件                //
//                                     //
//*****//
private void webBrowser_Navigated(object sender, WebBrowserNavigatedEventArgs e)
{
    this.pParentWin. urlBox.Text = webBrowser.Document.Url.ToString();
}
//*****//
//                                     //
//          文档加载进度事件                //
//                                     //
//*****//
private void webBrowser_ProgressChanged(object sender,
WebBrowserProgressChangedEventArgs e)
{
    this.pParentWin. toolStripProgressBar1.Maximum = (int)e.MaximumProgress;
    this.pParentWin. toolStripProgressBar1.Value = (int)e.CurrentProgress;
}
//*****//
//                                     //
//          判断页面后退键是否有效                //
//                                     //

```

```

//                                     //
//*****//
private void webBrowser_CanGoBackChanged(object sender, EventArgs e)
{
    this.pParentWin.tsBut_Back.Enabled = webBrowser.CanGoBack;
}
//*****//
//                                     //
//          判断页面前进键是否有效          //
//                                     //
//*****//
private void webBrowser_CanGoForwardChanged(object sender, EventArgs e)
{
    this.pParentWin.tsBut_Forward.Enabled = webBrowser.CanGoForward;
}
//*****//
//                                     //
//          浏览器文档标题改变事件          //
//                                     //
//*****//
private void webBrowser_DocumentTitleChanged(object sender, EventArgs e)
{
    string str_DocumentTitle = webBrowser.DocumentTitle;
    if (str_DocumentTitle.Equals(""))
    {
        //this.t = webBrowser.Document.Url.ToString();
        this.pParentWin.Text = webBrowser.Document.Url.ToString();
        this.Text = GetTabPageText(webBrowser.Document.Url.ToString());
    }
    else
    {
        this.ToolTipText = str_DocumentTitle;
        this.pParentWin.Text = str_DocumentTitle;
        this.Text = GetTabPageText(str_DocumentTitle);
    }
}
//*****//
//                                     //
//          获取页面选项卡的text值          //
//                                     //
//*****//
public string GetTabPageText(string strTabPageText)
{
    String TabPageText = strTabPageText;

```



```

//*****//
public ShouCangJiaMenuItem()
{

}

//*****//
//                                     //
//          初始化收藏夹文件名          //
//                                     //
//*****//
public ShouCangJiaMenuItem(string fileName)
{
    this.Text = fileName;
}

//*****//
//                                     //
//          初始化收藏夹全路径文件名      //
//                                     //
//*****//
public ShouCangJiaMenuItem(string fileName, string filePath)
{
    this.Text = GetMenuItemtext(fileName);
    this.ToolTipText = GetMenuItemToolTipText(fileName, filePath);
    UrlText = GetMuneItemUrl(filePath);
}

//*****//
//                                     //
//          获取收藏夹菜单的text值        //
//                                     //
//*****//
public string GetMenuItemtext(string strMenuItemtext)
{
    String MenuText = strMenuItemtext;
    int MenuTextLength = 30;
    MenuText = MenuText.Remove(MenuText.Length - 4);
    if (MenuText.Length > MenuTextLength)
    {
        MenuText = MenuText.Remove(MenuTextLength - 4) + "...";
    }
    return MenuText;
}

```

```

//*****//
//
//      获取收藏夹菜单的ToolTipText值      //
//
//*****//
public string GetMenuItemToolTipText(string strMenuItemText, string path)
{
    string MenuItemToolTipText;
    strMenuItemText = strMenuItemText.Remove(strMenuItemText.Length - 4);
    string strpath = GetMuneItemUrl(path);
    MenuItemToolTipText = "标题：" + strMenuItemText + "\n";
    MenuItemToolTipText += "网址：" + strpath;
    return MenuItemToolTipText;
}

//*****//
//
//      获取指定系统收藏夹里面的URL地址      //
//
//*****//
public string GetMuneItemUrl(string path)
{
    StreamReader sr = new StreamReader(path);
    string urltet;
    string first_str = sr.ReadLine();
    if (first_str != "[InternetShortcut]")
    {
        urltet = sr.ReadLine();
        //得到指定的URL地址
        urltet = urltet.Substring(8, urltet.Length - 8);
    }
    else
    {
        urltet = sr.ReadLine();
        //得到指定的URL地址
        urltet = urltet.Substring(4, urltet.Length - 4);
    }
    return urltet;
}
}
}

```

④ 编辑 XkunTabPage 功能类完成某些附属功能

编写 XkunTabPage 功能类代码如下:

```
////////////////////////////////////
//                          CSharp浏览器                          //
//                                                                //
//                          版权所有-熊坤编程                      //
//                                                                //
//                          2009-8-28日                          //
////////////////////////////////////

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

namespace CSharp浏览器
{
    public partial class XkunTabPage : System.Windows.Forms.TabPage
    {
        public XkunTabPage()
        {
            Button b = new Button();
            b.Location = new Point(0, 0);
            b.Text = "1";
            this.BorderStyle = System.Windows.Forms.BorderStyle.None;
            this.Font = new Font(this.Font,
            this.Font.Style | FontStyle.Bold);
        }
    }
}
```

⑤ 编辑浏览器基本功能

输入下列基本功能代码:

```
/**//
//
//      普通在地址栏输入地址浏览的方法
//
//
/**//

public void LiuLan()
{
    try
```

```

    {
        if (this.tabControl1.TabPages != null)
        {
            if (this.tabControl1.SelectedTab.Text == "空白页")
            {
                ((CustomTabPage)(this.tabControl1.SelectedTab)).Navigate = url_text;
            }
            else
            {
                Thread thdProcess = new Thread(new
ThreadStart(Thread_NewNavigateTabPage));
                thdProcess.Start();
            }
        }
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}

//*****//
//                                     //
//          新建空白页方法          //
//                                     //
//*****//
public void NewNullTabPage()
{
    CustomTabPage tabpage = new CustomTabPage(this);
    this.urlBox.Text = "about:blank";
    this.tabControl1.TabPages.Add(tabpage);
    this.tabControl1.SelectedTab = tabpage;
    this.Text = "空白页-CSharp浏览器";
    tabpage.Text = "空白页";
    this.toolStripStatusLabel1.Text = "CSharp浏览器";
}

//*****//
//                                     //
//          新建空白页线程          //
//                                     //
//*****//
private void Thread_NewNullTabPage()
{
    MethodInvoker mi = new MethodInvoker(this.NewNullTabPage);
    this.BeginInvoke(mi);
}

```

```

        Thread.Sleep(20);
    }

    //*****//
    //                                     //
    //          新建有导航地址的页面          //
    //      (用于普通的输入网址浏览和回车浏览)      //
    //                                     //
    //*****//
    public void NewNavigateTabPage()
    {

        CustomTabPage tabpage = new CustomTabPage(this);
        this.tabControl1.TabPages.Add(tabpage);
        tabpage.Navigate = url_text;
        this.tabControl1.SelectedTab = tabpage;
    }

    //*****//
    //                                     //
    //          新建有导航地址的线程          //
    //      (用于普通的输入网址浏览和回车浏览)      //
    //                                     //
    //*****//
    private void Thread_NewNavigateTabPage()
    {

        MethodInvoker mi = new MethodInvoker(this.NewNavigateTabPage);
        this.BeginInvoke(mi);
        Thread.Sleep(20);
    }

    //*****//
    //                                     //
    //          按回车按钮进行浏览          //
    //                                     //
    //*****//
    private void urlBox_KeyDown(object sender, KeyEventArgs e)
    {

        if (e.KeyCode == Keys.Enter)
        {
            url_text = this.urlBox.Text;
            LiuLan();
        }
    }

    //*****//
    //                                     //

```



```

//          双击鼠标删除页面          //
//          //
//*****//
private void tabControl1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.Dispose();
    tabControl1.TabPages[tabControl1.SelectedIndex].Dispose();
    if (tabControl1.TabPages.Count>=1)
    {
        tabControl1.SelectedTab = tabControl1.TabPages[tabControl1.TabPages.Count -
1];
    }
}
//*****//
//          //
//          进行页面切换          //
//          //
//*****//
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (this.tabControl1.SelectedTab != null)
    {
        this.urlBox.Text = ((CustomTabPage) (this.tabControl1.SelectedTab)).Navigate;
        this.Text = ((CustomTabPage) (this.tabControl1.SelectedTab)).ToolTipText;
        if (((CustomTabPage) (this.tabControl1.SelectedTab)).Text == "空白页")
        {
            this.Text = "空白页-CSharp浏览器";
        }
    }
    else
    {
        Thread thdProcess = new Thread(new ThreadStart(Thread_NewNullTabPage));
        thdProcess.Start();
    }
}

```

⑥ 编辑菜单栏功能代码

对应设置的菜单输入下列代码：

```

//*****//
//          //
//          打开链接状态          //
//          //
//*****//

```

```

private void 查看连接状态ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Visible = true;
    panel2.Visible = true;
}
//*****//
//                                     //
//          清除链接状态               //
//                                     //
//*****//
private void 清除ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
}
//*****//
//                                     //
//          关闭链接状态栏             //
//                                     //
//                                     //
//*****//
private void 关闭链接状态栏ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Visible = false;
    panel2.Visible = false;
    richTextBox1.Clear();
}
//*****//
//                                     //
//          清除链接状态栏（快捷菜单）   //
//                                     //
//                                     //
//*****//
private void 清除状态栏信息ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
}
//*****//
//                                     //
//          关闭链接状态栏（快捷菜单）   //
//                                     //
//                                     //
//*****//
private void 关闭链接状态栏ToolStripMenuItem1_Click(object sender, EventArgs e)
{
    richTextBox1.Visible = false;
    panel2.Visible = false;
    richTextBox1.Clear();
}

```

```

    }
    //*****//
    //                                     //
    //          查看源文件方法          //
    //                                     //
    //*****//
    private void GetHtmlSource()
    {
        string Myurl =
((CustomTabPage)(this.tabControl1.SelectedTab)).webBrowser.Url.ToString();
        System.IO.Stream stream = null;
        WebClient client = new WebClient();
        stream = client.OpenRead(Myurl);
        StreamReader readerOfStream = new StreamReader(stream,
System.Text.Encoding.GetEncoding("GB2312"));
        string aa = readerOfStream.ReadToEnd();

        // Close the stream.
        stream.Close();
        richTextBox2.Text = aa;
    }
    //*****//
    //                                     //
    //          查看源文件按钮          //
    //                                     //
    //*****//
    private void toolStripSplitButton2_ButtonClick(object sender, EventArgs e)
    {
        try
        {
            TextBox_FindNeiRong.Focus();
            richTextBox2.Visible = true;
            panel3.Visible = true;
            GetHtmlSource();
        }
        catch
        {
            MessageBox.Show("无法连接到该网站，或没打开网站，请检查您的网络是否正常！ ", "
提示 ");
            richTextBox2.Visible = false;
            panel3.Visible = false;
        }
    }
}

```

```

//*****//
//
//          查看源文件菜单
//
//*****//
private void 查看当前网站源文件ToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        TextBox_FindNeiRong.Focus();
        richTextBox2.Visible = true;
        panel3.Visible = true;
        GetHtmlSource();
    }
    catch
    {
        MessageBox.Show("无法连接到该网站，或没打开网站，请检查您的网络是否正常！", "
提示 ");

        //panel1.Dock = DockStyle.Fill;
        panel3.Visible = false;
        richTextBox2.Visible = false;

    }
}

//*****//
//
//          关闭窗口文件子菜单
//
//*****//
private void 关闭源文件ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox2.Visible = false;
    panel3.Visible = false;
    richTextBox2.Clear();
}

//*****//
//
//          查看源文件快捷菜单
//
//*****//
private void 关闭ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox2.Visible = false;
    panel3.Visible = false;

```

```

        richTextBox2.Clear();
    }
    //*****//
    //                                     //
    //             查找源文件（下一步）           //
    //                                     //
    //*****//
private void toolStripButton7_Click(object sender, EventArgs e)
{
    //richTextBox2.SelectionColor = Color.Black;
    richTextBox2.Focus();
    string RichText, FindText;
    RichText = richTextBox2.Text;
    FindText = toolStripTextBox1.Text;
    if (RichText.Contains(FindText))
    {
        i++;
        cishu.Add(richTextBox2.Find(FindText, cishu[i - 1], RichText.Length, 0));
        cishu[i] += FindText.Length;
        // MessageBox.Show(cishu[i].ToString());
        //richTextBox2.SelectionColor = Color.Blue;
    }
}
//*****//
//                                     //
//             查找源文件（上一步）           //
//                                     //
//*****//
private void toolStripButton8_Click(object sender, EventArgs e)
{
    //richTextBox2.SelectionColor = Color.Black;
    richTextBox2.Focus();
    string RichText, FindText;
    RichText = richTextBox2.Text;
    FindText = toolStripTextBox1.Text;
    if (RichText.Contains(FindText) && i > 0)
    {
        int y = cishu[i - 1];
        y -= FindText.Length;
        if (y < 0)
        {
            y += FindText.Length;
            MessageBox.Show("到最上面，没有该查询", "提示");
        }
    }
}

```

```

        richTextBox2.Find(FindText, y, RichText.Length, 0);
        //richTextBox2.SelectionColor = Color.Blue;
        i--;
    }
    else
    {
        MessageBox.Show("到最上面，没有该查询", "提示");
        cishu.Clear();
        cishu.Add(0);
        i = 0;
    }
}

//*****//
//                                     //
//          查找源文件（文本框变化）          //
//                                     //
//*****//
private void toolStripTextBox1_TextChanged(object sender, EventArgs e)
{
    richTextBox2.SelectionColor = Color.Black;
    cishu.Clear();
    cishu.Add(0);
    i = 0;
}

//*****//
//                                     //
//          新建空白页（菜单栏）          //
//                                     //
//*****//
private void 空白BToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    Thread thdProcess = new Thread(new ThreadStart(Thread_NewNullTabPage));
    thdProcess.Start();
}

//*****//
//                                     //
//          关闭当前页（菜单栏）          //
//                                     //
//*****//
private void 关闭页面ToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    ((CustomTabPage)(this.tabControl1.SelectedTab)).webBrowser.Dispose();
    tabControl1.TabPages[tabControl1.SelectedIndex].Dispose();
    if (tabControl1.TabPages.Count >= 1)

```

```

        {
            tabControl1.SelectedTab = tabControl1.TabPages[tabControl1.TabPages.Count -
1];
        }
    }

//*****//
//                                     //
//          关闭全部页（菜单栏）          //
//                                     //
//*****//
private void 全部关闭RToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    tabControl1.TabPages.Clear();
}

//*****//
//                                     //
//          退出当前页（菜单栏）          //
//                                     //
//*****//
private void 退出XToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

//*****//
//                                     //
//          查看连接状态（菜单栏）          //
//                                     //
//*****//
private void 查看连接状态VToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Visible = true;
    panel2.Visible = true;
}

//*****//
//                                     //
//          清空连接状态（菜单栏）          //
//                                     //
//*****//
private void 清空连接状态信息ToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
}

//*****//
//                                     //

```

```

//          关闭连接状态（菜单栏）          //
//          //
//*****//
private void 关闭连接状态CToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Visible = false;
    panel2.Visible = false;
    richTextBox1.Clear();
}
//*****//
//          //
//          打开当前源文件（菜单栏）          //
//          //
//*****//
private void 打开源文件OToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        TextBox_FindNeiRong.Focus();
        richTextBox2.Visible = true;
        panel3.Visible = true;
        GetHtmlSource();
    }
    catch
    {
        MessageBox.Show("无法连接到该网站，或没打开网站，请检查您的网络是否正常！ ", "
提示 ");

        //panel1.Dock = DockStyle.Fill;
        panel3.Visible = false;
        richTextBox2.Visible = false;

    }
}
//*****//
//          //
//          关闭当前源文件（菜单栏）          //
//          //
//*****//
private void 关闭源文件CToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox2.Visible = false;
    panel3.Visible = false;
    richTextBox2.Clear();
}

```



```

//*****//
//                                     //
//          另存为（菜单栏）          //
//                                     //
//*****//
private void 另存为AToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {

((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.ShowSaveAsDialog();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }

}

//*****//
//                                     //
//          属性（菜单栏）            //
//                                     //
//*****//
private void 属性RToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {

((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.ShowPropertiesDialog();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }

}

//*****//
//                                     //
//          页面设置（菜单栏）        //
//                                     //
//*****//
private void 页面设置UToolStripMenuItem_Click(object sender, EventArgs e)
{
    try

```

```

    {

((CustomTabPage) (this.tabControll1.SelectedTab)).webBrowser.ShowPageSetupDialog();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }

}

//*****//
//                                     //
//          打印预览（菜单栏）          //
//                                     //
//*****//
private void 打印预览VToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {

((CustomTabPage) (this.tabControll1.SelectedTab)).webBrowser.ShowPrintPreviewDialog();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }

}

//*****//
//                                     //
//          打印（菜单栏）          //
//                                     //
//*****//
private void 打印PToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {

((CustomTabPage) (this.tabControll1.SelectedTab)).webBrowser.ShowPrintDialog();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }

}

```

```

}
//*****//
//                                     //
//             添加到收藏夹（菜单栏）             //
//                                     //
//*****//
private void 添加到收藏夹AToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        FindInvokeIEServerCommand(IDM_ADDFAVORITES);
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}

//*****//
//                                     //
//             整理收藏夹（菜单栏）             //
//                                     //
//*****//
private void 整理收藏夹OToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        ShellUIHelper helper = new ShellUIHelper();
        object o = null;
        helper.ShowBrowserUI("OrganizeFavorites", ref o);
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}

//*****//
//                                     //
//             导入与导出（菜单栏）             //
//                                     //
//*****//
private void 导入导出IToolStripMenuItem_Click(object sender, EventArgs e)
{
    try

```

```

        {
            FindShellDocObjectViewCommand(ID_IE_FILE_IMPORTEXPORT);
        }
        catch (System.Exception ex)
        {
            throw ex;
        }
    }

    //*****//
    //                                     //
    //          桌面快捷方式（菜单栏）          //
    //                                     //
    //*****//
    private void 桌面快捷方式SToolStripMenuItem_Click_1(object sender, EventArgs e)
    {
        try
        {
            FindShellDocObjectViewCommand(ID_IE_FILE_SENDDESKTOPSHORTCUT);
        }
        catch (System.Exception ex)
        {
            throw ex;
        }
    }

    //*****//
    //                                     //
    //          电子邮件（菜单栏）          //
    //                                     //
    //*****//
    private void 电子邮件EToolStripMenuItem_Click(object sender, EventArgs e)
    {
        try
        {
            FindShellDocObjectViewCommand(ID_IE_FILE_NEWMAIL);
        }
        catch (System.Exception ex)
        {
            throw ex;
        }
    }
}

```

```

//*****//
//                                     //
//          Internet选项（菜单栏）    //
//                                     //
//*****//
private void internet选项IToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process.Start("C://WINDOWS//system32//inetcpl.cpl");
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}

//*****//
//                                     //
//          标准按钮（菜单栏）        //
//                                     //
//*****//
private void 标准按钮SToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (标准按钮SToolStripMenuItem.Checked == false)
    {
        tsBut_NewNullTab.Visible = false;
        tsBut_Back.Visible = false;
        tsBut_Forward.Visible = false;
        tsBut_refurbish.Visible = false;
        tsBut_Stop.Visible = false;
        tsBut_HomePage.Visible = false;
    }
    else
    {
        tsBut_NewNullTab.Visible = true;
        tsBut_Back.Visible = true;
        tsBut_Forward.Visible = true;
        tsBut_refurbish.Visible = true;
        tsBut_Stop.Visible = true;
        tsBut_HomePage.Visible = true;
    }
}

```

```

//*****//
//                                     //
//          地址栏（菜单栏）          //
//                                     //
//*****//
private void 地址栏AToolStripMenuItem_Click(object sender, EventArgs e)
{

    if (地址栏AToolStripMenuItem.Checked == false)
    {
        toolStrip1.Visible = false;
        // btn_LiuLan.Visible = false;

    }
    else
    {
        toolStrip1.Visible = true;
        //btn_LiuLan.Visible = true;

    }
}

//*****//
//                                     //
//          状态栏（菜单栏）          //
//                                     //
//*****//
private void 状态栏BToolStripMenuItem_Click(object sender, EventArgs e)
{

    if (状态栏BToolStripMenuItem.Checked == false)
    {
        statusStrip1.Visible = false;

    }
    else
    {
        statusStrip1.Visible = true;

    }
}

```

⑦ 编辑工具栏功能代码

```

//*****//
//                                     //
//          按浏览按钮进行浏览          //
//                                     //
//*****//

```

```

private void toolStripButton9_Click(object sender, EventArgs e)
{
    url_text = this.urlBox.Text;
    LiuLan();
}
//*****//
//                                     //
//          工具栏新建空页面          //
//                                     //
//*****//
private void toolStripButton4_Click(object sender, EventArgs e)
{
    Thread thdProcess = new Thread(new ThreadStart(Thread_NewNullTabPage));
    thdProcess.Start();
}
//*****//
//                                     //
//          后退, 返回先前页面        //
//                                     //
//*****//
private void toolStripButton1_Click(object sender, EventArgs e)
{
    ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.GoBack();
}
//*****//
//                                     //
//          前进, 浏览新页面          //
//                                     //
//*****//
private void toolStripButton2_Click(object sender, EventArgs e)
{
    ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.GoForward();
}

//*****//
//                                     //
//          刷新当前页面              //
//                                     //
//*****//
private void toolStripButton5_Click(object sender, EventArgs e)
{
    if
(!((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.Url.Equals("about:blank"))

```

```

        {
            ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.Refresh();
        }
    }

    //*****//
    //                                     //
    //             停止当前页面             //
    //                                     //
    //*****//
    private void toolStripButton3_Click(object sender, EventArgs e)
    {
        ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.Stop();
    }

    //*****//
    //                                     //
    //             回到当前主页             //
    //                                     //
    //*****//
    private void toolStripButton6_Click_1(object sender, EventArgs e)
    {
        ((CustomTabPage) (this.tabControl1.SelectedTab)).webBrowser.GoHome();
    }

    //*****//
    //                                     //
    //             单击搜索按钮             //
    //                                     //
    //*****//
    private void Button_Find_Click(object sender, EventArgs e)
    {
        search();
    }

    //*****//
    //                                     //
    //             回车进行搜索             //
    //                                     //
    //*****//

    private void TextBox_FindNeiRong_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            search();
        }
    }
}

```


⑧ 编辑其他功能代码

编辑IE命令功能代码如下:

```
//*****//
//
//          调用IE隐藏命令          //
//      (即调用Internet Explorer_Server的命令) //
//          //
//*****//

private void FindInvokeIEServerCommand(int nID)
//调用IE隐藏命令，既调用Internet Explorer_Server的命令
{
    IntPtr vHandle =
((CustomTabPage)(this.tabControll.SelectedTab)).webBrowser.Handle;
    vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Shell Embedding", null);
    vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Shell DocObject View", null);
    vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Internet Explorer_Server", null);
    SendMessage(vHandle, WM_COMMAND, nID, (int)Handle);
}

//*****//
//
//          调用IE隐藏命令          //
//      (即调用Shell DocObject View的命令) //
//          //
//*****//

private void FindShellDocObjectViewCommand(int nID)
{
    IntPtr vHandle =
((CustomTabPage)(this.tabControll.SelectedTab)).webBrowser.Handle;
    vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Shell Embedding", null);
    //vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Internet Explorer_Server", null);
    vHandle = FindWindowEx(vHandle, IntPtr.Zero, "Shell DocObject View", null);
    SendMessage(vHandle, WM_COMMAND, nID, (int)Handle);
}
```

编辑收藏夹处理功能代码:

```
//*****//
//
//          动态生成收藏夹          //
//          //
//*****//

private void ShouCangMenuItem()
{
}
```

```

        try
        {
            string favorfolder =
Environment.GetFolderPath(Environment.SpecialFolder.Favorites);           //获取系统收藏夹路径
            ListFiles(收藏夹ToolStripMenuItem, new DirectoryInfo(favorfolder));
        }
        catch (System.Exception ex)
        {
            throw(ex);
        }
    }

    //*****//
    //                                     //
    //          生成收藏夹的函数（递归）          //
    //                                     //
    //*****//
    public void ListFiles(ShouCangJiaMenuItem toolstripmenuItem, FileSystemInfo info)
    {
        if (!info.Exists) return;
        DirectoryInfo dir = info as DirectoryInfo;
        //不是目录
        if (dir == null) return;
        FileSystemInfo[] files = dir.GetFileSystemInfos();
        ShouCangJiaMenuItem[] ShouCangsMenuItem = new ShouCangJiaMenuItem[files.Length];
        for (int i = 0; i < files.Length; i++)
        {
            FileInfo file = files[i] as FileInfo;
            // ShouCangsMenuItem[i].AutoSize = false;
            //是文件
            if (file != null)
            {
                ShouCangsMenuItem[i] = new ShouCangJiaMenuItem(file.Name, file.FullName);
                ShouCangsMenuItem[i].Click += new EventHandler(ShouCangsMenuItem_Click);
                //把获得的URL赋值给菜单工具提示文本
                toolstripmenuItem.DropDownItems.Add(ShouCangsMenuItem[i]);
            }
            //对于子目录，进行递归调用
        }
        else
        {
            DirectoryInfo Direct = files[i] as DirectoryInfo;
            ShouCangsMenuItem[i] = new ShouCangJiaMenuItem(files[i].Name);
            //ShouCangsMenuItem[i].Click += new
            EventHandler(ShouCangsMenuItem_Click);
        }
    }

```

```

        toolStripMenuItem.DropDownItems.Add(ShouCangMenuItem[i]);
        ListFiles(ShouCangMenuItem[i], new DirectoryInfo(Direct.FullName));    //
递归使用，生成子菜单
    }
}

}

//*****//
//                                     //
//          生成收藏夹的进程          //
//                                     //
//*****//
private void Thread_ShouCangMenuItem()
{
    MethodInvoker mi = new MethodInvoker(this.ShouCangMenuItem);
    this.BeginInvoke(mi);
    Thread.Sleep(20);
}
//*****//
//                                     //
//          单击收藏夹事件          //
//                                     //
//*****//
private void ShouCangMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        ShouCangJiaMenuItem menuItem = (ShouCangJiaMenuItem) sender;
        url_text = menuItem.UrlText;
        LiuLan();
    }
    catch (System.Exception ex)
    {
        throw ex;
    }
}
//*****//
//                                     //
//          启动收藏夹菜单线程          //
//                                     //
//*****//
private void timer1_Tick(object sender, EventArgs e)
{
    Thread ShouCangMenuItemProcess = new Thread(new

```

```
ThreadStart(Thread_ShouCangMenuItem));
    ShouCangMenuItemProcess.Start();
    timer1.Stop();
}
```

编辑搜索功函数:

```
//*****//
//                                     //
//          搜索功能函数               //
//                                     //
//*****//
private void search()
{
    string SearchUrl = null;
    if (!TextBox_FindNeiRong.Text.Equals(""))
    {
        if (Combo_FindType.SelectedIndex.Equals(0))
        {
            SearchUrl = "http://www.baidu.com/s?&word=" + TextBox_FindNeiRong.Text;
        }
        else if (Combo_FindType.SelectedIndex.Equals(1))
        {
            SearchUrl =
            "http://www.google.com/search?hl=zh-CN&inlang=zh-CN&ie=GB2312&oe=GB2312&q=" +
            TextBox_FindNeiRong.Text;
        }
        else if (Combo_FindType.SelectedIndex.Equals(3))
        {
            SearchUrl = "http://www.yahoo.com.cn/search?p=" +
            TextBox_FindNeiRong.Text;
        }
        else if (Combo_FindType.SelectedIndex.Equals(2))
        {
            SearchUrl =
            "http://search.iask.com/cgi-bin/search/search.cgi?_searchkey=" + TextBox_FindNeiRong.Text;
        }
        else if (Combo_FindType.SelectedIndex.Equals(4))
        {
            SearchUrl = "http://baigoohoo.com/search.asp?catalog=web&word=" +
            TextBox_FindNeiRong.Text;
        }
        else
        {

```

```
        SearchUrl = "http://www.iciba.com/" + TextBox_FindNeiRong.Text +  
        "/?uid=29314&sid=1";  
    }  
    url_text = SearchUrl;  
    LiuLan();  
}  
}
```

(3) 程序调试与错误处理

比如若调试出现下列错误，如图4-12所示。

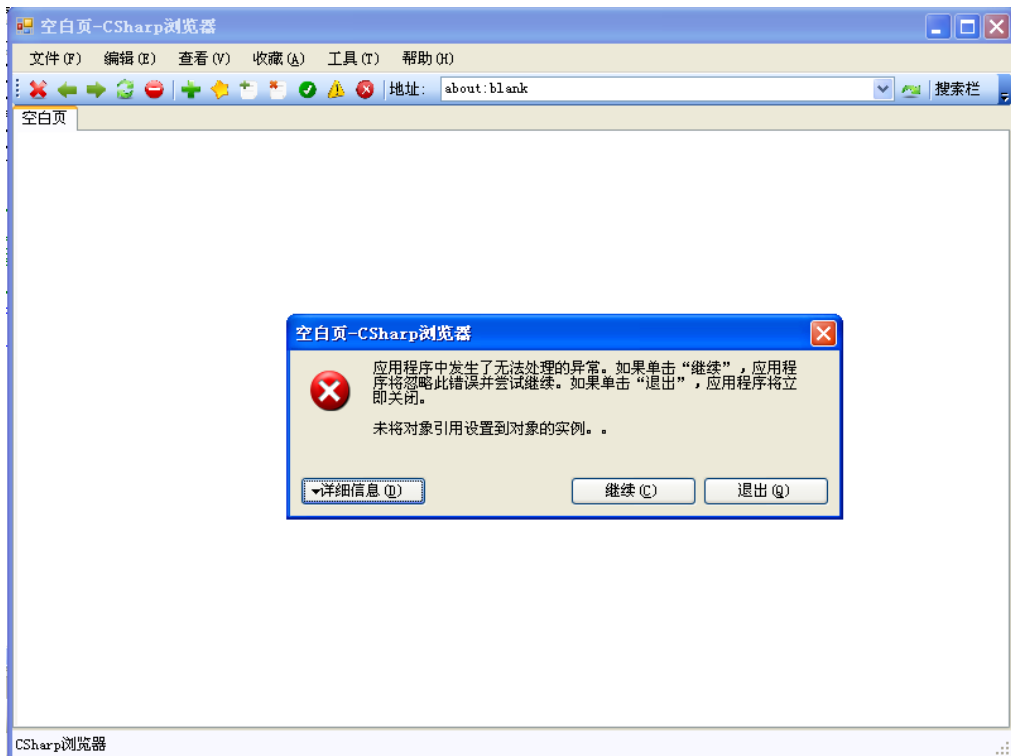


图4-12 出现运行错误

分析可知：由于c#是严格语言，所有变量（尤其是引用基础类空间的变量），必须声明并设置其初值，完成变量初值设置后，这类错误后可以消除（当然，也可能与平台版本相关）。处理提示如图4-13所示。



图13 运行错误处理提示

4.6 程序运行测试

②测试浏览网页功能（新浪新闻）



图14 测试浏览网页功能

③测试网络连接查看功能

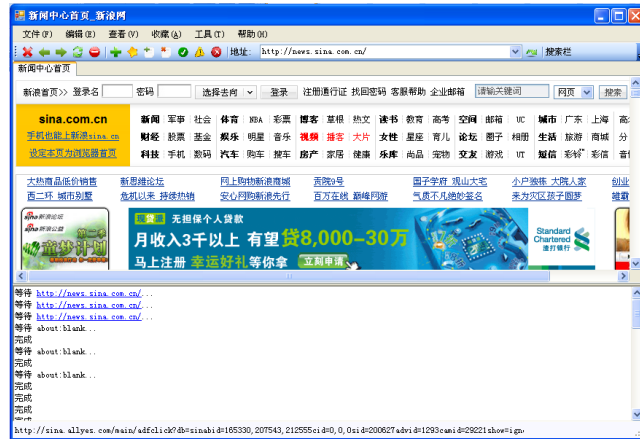


图15 测试网络连接查看功能

④测试源文件查看功能



图16 测试源文件查看功能

⑤测试功能菜单Internet选项



图16测试功能菜单“Internet选项”

五 项目小结

写出自己关于实训项目的心得、经验、教训、收获等

六 项目验收

验收内容：程序设计文档+程序代码（可执行）

1. 设计文档，包括功能描述与分析、功能模块划分、项目架构、界面设计、模块功能设计、核心数据结构、核心功能函数流程，以及实训小结等；

2. 完整的程序代码，要求代码正确执行。