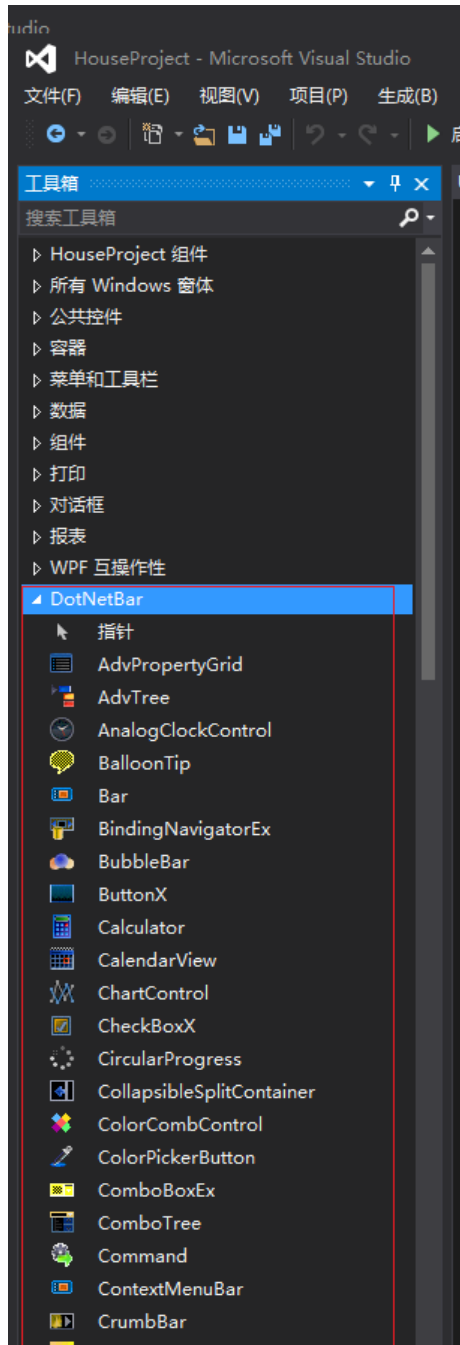


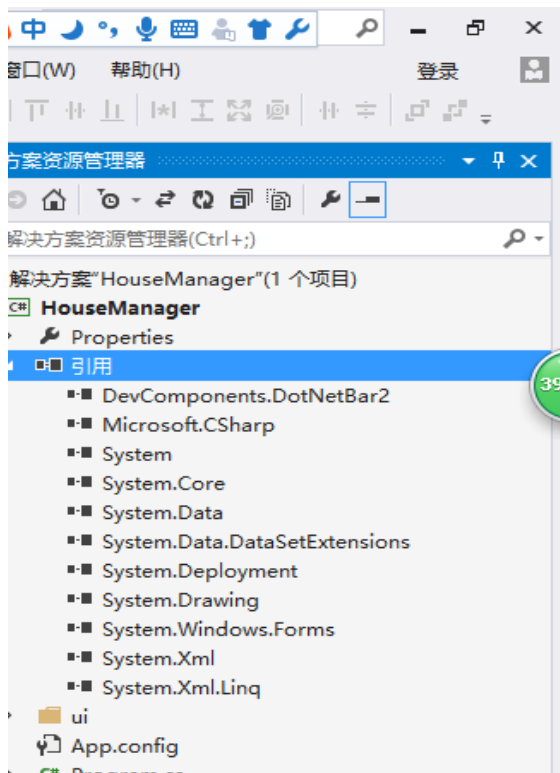
一、安装 DotNetBar 控件

- 1、安装 “DotNetBar for Windows Forms 12.9.0.0.exe” 之前请关闭 VS 2013
- 2、双击 “DotNetBar for Windows Forms 12.9.0.0.exe” 进行安装
- 3、如果在运行过程中提示需要安装.NET Framework 3.0 或者 .NET Framework 3.5 之类的版本，请选择安装
- 4、提示安装成功之后，启动 VS 2013，无操作等待两到三分钟
- 5、创建 Winforms 项目后，就能在工具箱中看到"DotNetBar"分类，下面是所有的新控件：



二、让窗体变成 office 风格

- 0、首先往任意窗口拖入一个 DotNetBar 控件（如 TextBoxX），然后就能在解决方案资源管理器中看到如下内容



- 1、接着窗体全部继承自 DevComponents.DotNetBar.Office2007Form
- 2、然后要在构造函数里面写上 this.EnableGlass = false;

三、登录窗口的设计

- 1、把所有的图片文件都放入项目的_res 文件夹
- 2、窗体的图标要使用 ico 格式
- 3、窗体的标题使用 Text 属性来更改
- 4、将 MaximizeBox 属性改为 false
- 5、图片使用 vs 自带的 PictureBox 控件
- 6、文本标签使用 DotNetBar 的 LabelX 控件，设置控件的 TextAlignment 属性为 Far，表示文字靠右
- 7、输入文本框使用 DotNetBar 的 TextBoxX 控件，注意命名规范，txt 开头，使用骆驼命名法
- 8、新用户注册 和 找回密码 使用 DotNetBar 的 LabelX 控件，注意命名规范，lbl 开头，使用骆驼命名法

为了实现鼠标移入移出事件，写两个事件方法：

8.1 鼠标移入事件是 MouseEnter，事件方法如下：

```
/// <summary>
/// 【事件】鼠标移入
/// </summary>
/// <param name="sender">表示发生事件的那个控件</param>
/// <param name="e">事件发生时的一些细节信息</param>
private void lblFindPwd_MouseEnter(object sender, EventArgs e)
{
    LabelX lbl = (LabelX)sender; // 因为两个控件公用这个方法，所以这里才用到 sender
```

```
lbl.ForeColor = Color.Blue;
}
```

8.2 鼠标移出事件是 MouseLeave，事件方法略

9、按钮使用 DotNetBar 的 ButtonX 控件，注意命名规范，btn 开头，使用骆驼命名法

四、选项卡式的主界面，按选项卡设计用户控件

1、首先主界面不能改变大小，其次，主界面有一个最低目标分辨率，如 1024*768

除了设置 MaximizeBox 为 false，还要设置 FormBorderStyle 为 FixedSingle

2、菜单栏使用 VS 自带的 MenuStrip 控件，一级菜单项按模块分

3、采用选项卡式的界面，选项卡使用 DotNetBar 的 TabControl 控件，命名规范是以 tabCtl 开头

将 TabControl 的 Dock 属性设置为 Fill，即正中间全部区域

【注意】选项卡控件叫 TabControl，里面有很多的选项卡页，选项卡页由头 TabItem 和内容 TabControlPanel 构成

4、选中选项卡控件，在控件右上角的三角形箭头下拉中，选 Create New Tab 能追加新选项卡页

5、选项卡控件的四个属性 CloseButtonXXX 都选择成 True 就有关闭按钮

6、选项卡控件的属性 AutoCloseTabs 设置为 True，则选项卡页就能在点击关闭按钮后自动关闭

要实现“首页”不能关闭，对选项卡控件做 TabItemClose 事件的处理

```
/// <summary>
/// 【事件】整个选项卡控件中，某个选项卡页被关闭
/// </summary>
/// <param name="sender">发生事件的选项卡</param>
/// <param name="e">发生事件时的一些相关信息</param>
private void tabCtlMain_TabItemClose(object sender, DevComponents.DotNetBar.TabStripActionEventArgs
e)
{
    TabItem ti = e.TabItem; // 从 e 中获得当前被关闭的选项卡页的头是哪个

    if (ti.Text == "首页")
        e.Cancel = true; // 将该事件取消
    else
    {
        e.Cancel = true; // 将该事件取消，不关了
        ti.Visible = false; // 然后将选项卡隐藏而已
    }
}
```

7、在界面刚加载时，选项卡控件只显示“首页”，在界面的构造函数里加入如下代码：

```

    TabsCollection tc = this.tabCtlMain.Tabs; // 得到选项卡控件的所有选项页头
    for (int i = 1; i < tc.Count; i++) // 遍历这些页头
    {
        TabItem ti = tc[i];
        ti.Visible = false; // 隐藏
    }

```

8、点击相应的菜单，能显示对应的选项卡页，为菜单项编写 Click 事件的处理方法

```

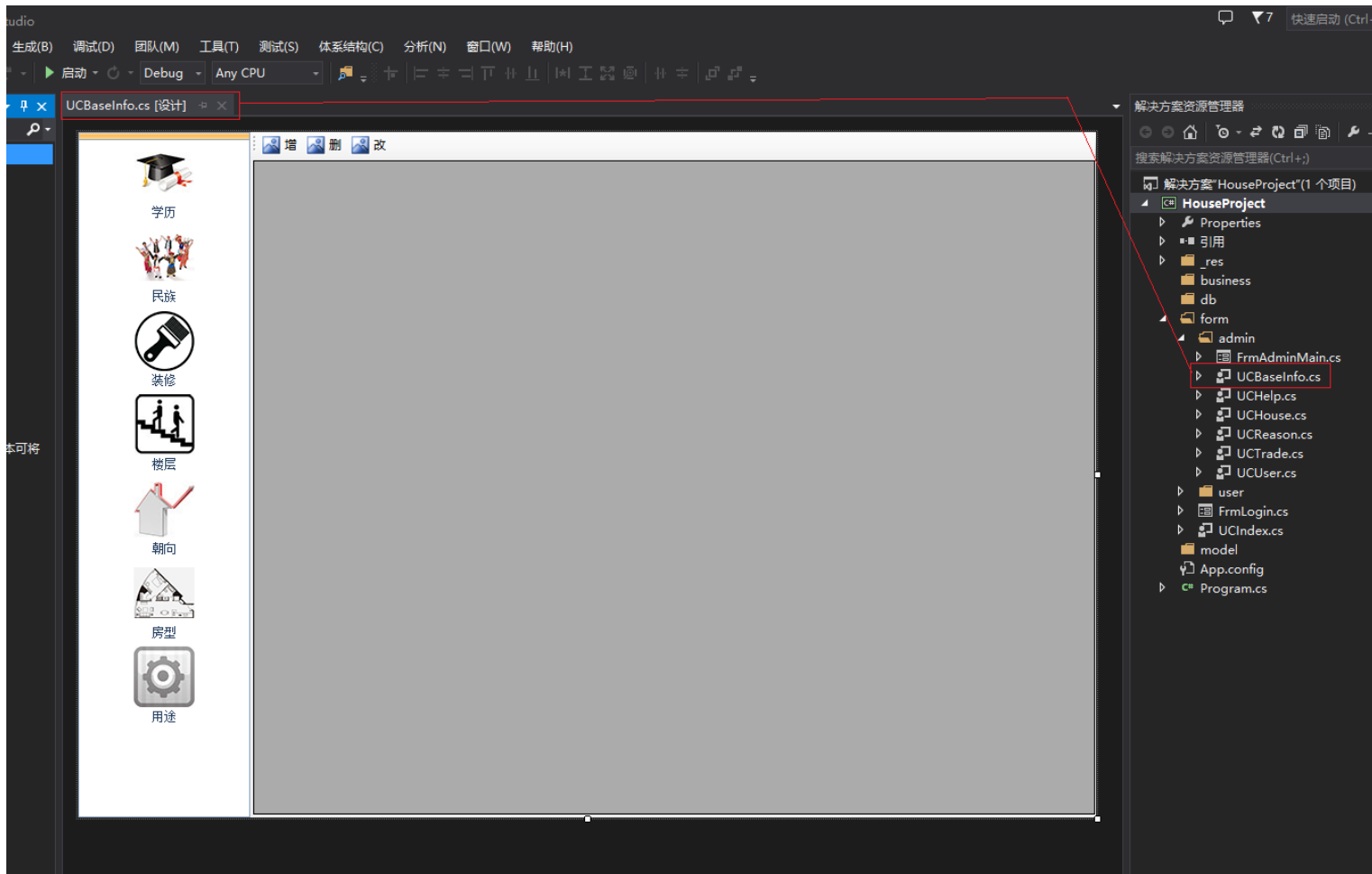
/// <summary>
/// 【事件】基础信息管理下，任意一个子菜单的点击事件
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void 房型管理 ToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.tabCtlMain.Tabs[1].Visible = true; // 显示对应的选项卡页
    this.tabCtlMain.SelectedTab = this.tabCtlMain.Tabs[1]; // 而且把这个页激活，显示在最前面
}

```

9、以下开始为每个选项卡页设计用户控件，一页设计一个用户控件

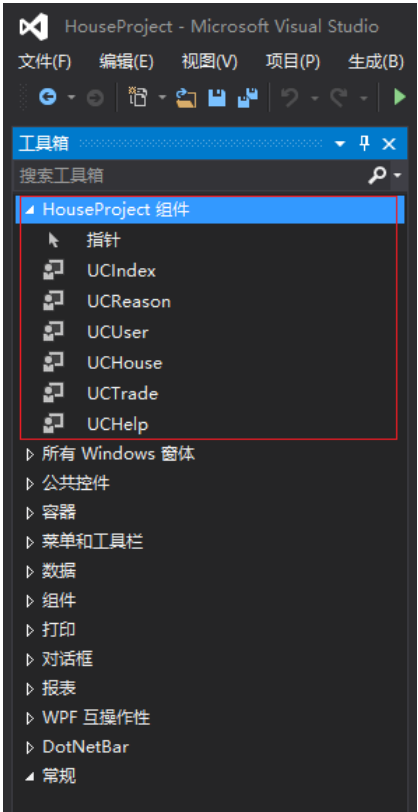
9.1、在项目中的某个文件夹上右击，选择“添加”——>“用户控件”，如“基本信息管理”选项卡的用户控件：

用户控件的大小略小于选项卡内容的大小，比如选项卡是 1008*679，则用户控件是 1006*677



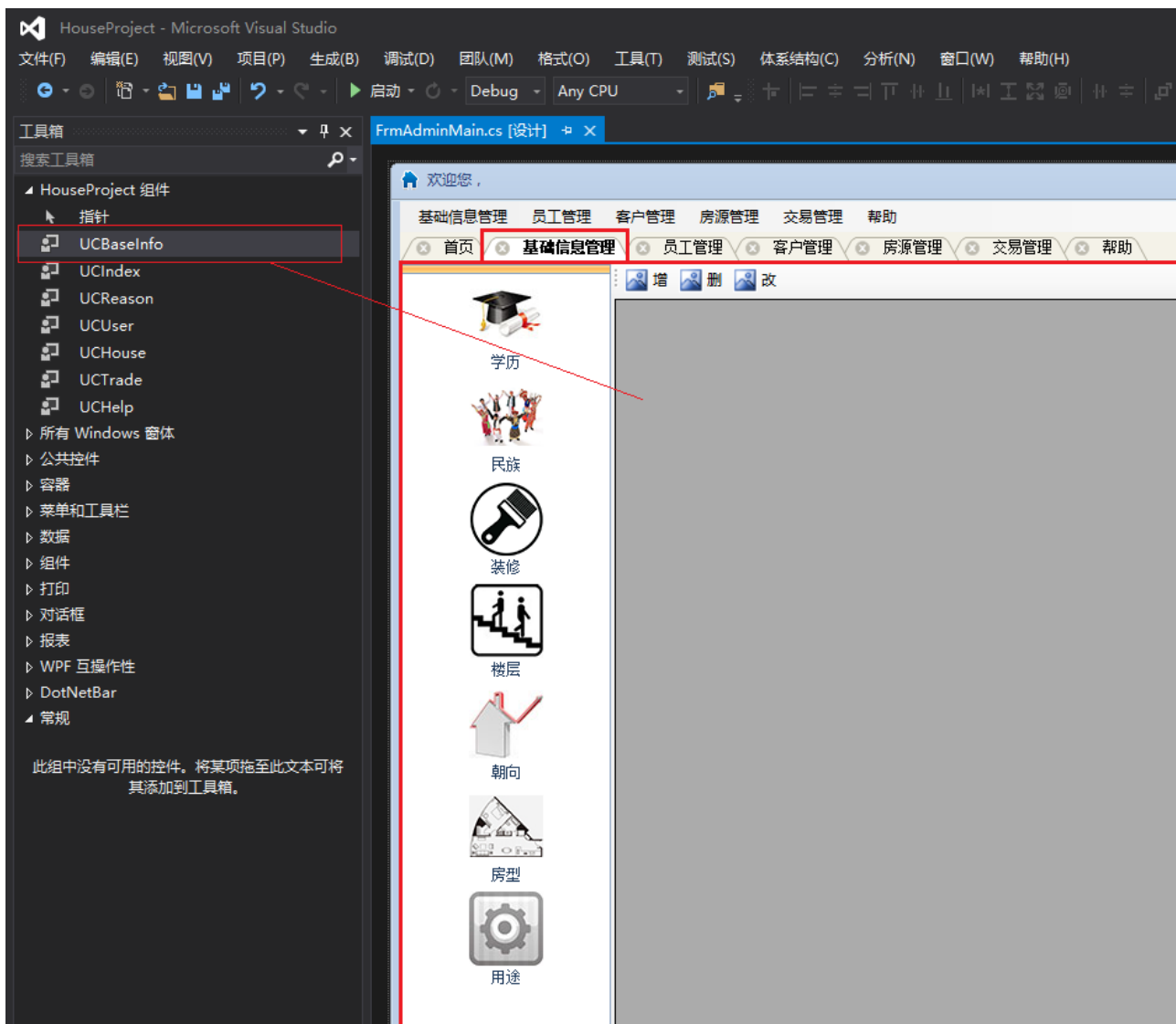
9.2、在解决方案资源管理器的项目上右击，选择“重新生成”

9.3、在工具箱中应该能看到多了一个项目控件，如：



9.4、打开主界面，选定对应的选项卡，将上图中的用户控件拖入选项卡即可

将拖入的用户控件的 Dock 属性选择为 Fill



9.5、【注意】

不能在主界面的选项卡中写选项卡内容控件的事件代码

只能在用户控件中写事件代码

五、某些选项卡的内容是用户控件，用户控件中的左右结构

1、左边使用的是 DotNetBar 的 SideBar 控件，设置 SideBar 控件的 Dock 停靠属性为左

SideBar 控件中有图文按钮，是在 SideBar 上右击，在弹出菜单选“Create Button”

选中任意的 Button，可以为他设置图片和文字

2、右边根据左边的按钮显示不同的内容，右边也嵌套一个选项卡，DotNetBar 的 TabControl 控件

2.1 右边选项卡只默认显示第一页，其他页隐藏

2.2 为左边的按钮写点击事件，点击了哪个按钮就显示对应的右边内容，且隐藏其他的内容

六、完成简单的登录功能

1、在登录界面，步骤分为四大步：

1.1 从界面接收数据

1.2 检查数据

1.3 数据库操作

1.4 根据数据库操作结果进行反馈

2、数据库操作分为三大类型的对象

// 3. 进行数据库操作

// 3.1 创建数据库连接

```
SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=HouseProject;User ID=sa;Pwd=123");
```

// 3.2 创建数据库语句对象

```
string sql = "select * from login where loginAccount='{0}' and loginPwd='{1}'";
```

```
sql = string.Format( sql, loginAccount, loginPwd );
```

```
SqlCommand cmd = new SqlCommand(sql,conn);
```

// 3.3 执行查询语句获得查询结果

```
conn.Open();
```

```
SqlDataReader datareader = cmd.ExecuteReader();
```

3、对数据库查询结果的处理（处理完一定要关闭连接）

// 4. 根据数据库操作结果反馈

```
if (datareader != null && datareader.Read())
```

```
{
```

```
    this.Hide(); // 因为当前登录窗口是主线程窗口，所以只能隐藏，不能 Close
```

```
    FrmAdminMain frmAdminMain = new FrmAdminMain();
```

```
    frmAdminMain.Show();
```

```
}
```

```
else
```

```
{
```

```
    MessageBox.Show("很遗憾，登录失败！");
```

```
}
```

```
conn.Close();
```

4、管理主界面的关闭要实现退出程序

为管理主界面的“FormClosing”事件写处理方法

方法只有一条语句：

```
/// <summary>
```

```
/// 【事件】窗体关闭
```

```
/// </summary>
```

```
/// <param name="sender"></param>
```

```
/// <param name="e"></param>
```

```
private void FrmAdminMain_FormClosing(object sender, FormClosingEventArgs e)
```

```
{
```

```
    System.Environment.Exit(0);
```

```
}
```