# 四 HALCON联合C#编程

HALCON的开发使用的是自带的编译环境HDevelop，对于开发工作者来说是个很好的使用环境，但是使用HALCON开发的程序最终还是要交付给客户使用的，在HDevelop中调试完的程序并不能直接交付客户使用，还需要经过一层封装，将HDevelop中调试完的程序封装成一个界面应用软件，方便用户使用的软件程序。

本书联合编程所用到的编程语言为C#及C++，涉及的用户界面封装软件为Winform和Qt。涉及内容包括：软件的安装、软件的简单使用、环境配置、联合编程等。

## 第10章winform入门

在HALCON联合C#编程中，最常用也是相对比较简单的用户界面封装软件就是Winform，WinForm是Windows Form的简称，是基于.NETFramework平台的客户端（PC软件）开发技术，一般使用C#编程。WinForm支持可视化设计，简单易上手，并可以接入大量的第三方UI库或自定义控件，给桌面应用开发带来了无限可能。

HALCON联合C#编程目前主要是三种方式：

1. 将HALCON代码导成C#代码后放到C#软用户界面封装的程序中。
2. 直接在C#软件用户界面封装的程序中用HALCON的C#库语句进行程序编写。

（相当于第一种方式的进阶版本，适合对C#和HALCON代码比较熟悉的老手使用。）

1. 利用HALCON引擎直接在C#软件用户界面封装的程序中调用HALCON程序。

（必须安装完整的HALCON软件，但可以在HALCON软件中修改代码后在C#软件界面中直接观察到修改后的效果）

考虑到本书所面向的用户群体，本书将着重介绍第一种编程方式。

### 10.1 winform的安装

本书以visual studio 2022的Community版本为例，演示下载安装过程。首先打开浏览器，访问visual studio官网，网址为：<https://visualstudio.microsoft.com/zh-hans/downloads/>，网站页面如图所示。

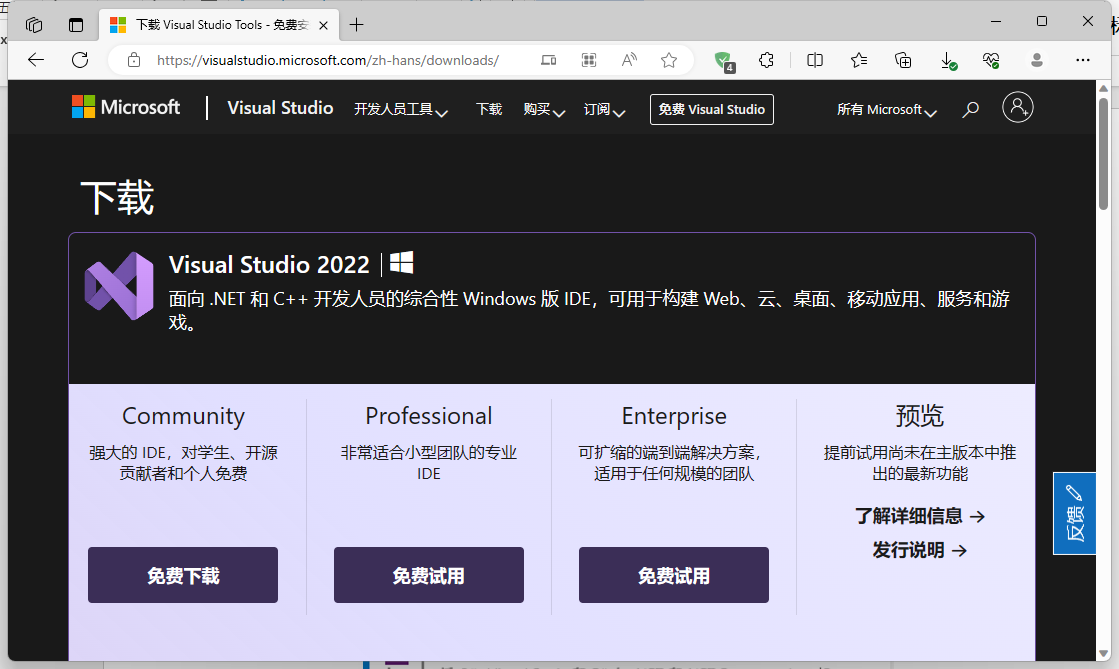


图10.1 visual studio 2022的下载页面

这里选择社区版（Community）进行下载，得到安装包。双击打开安装包，出现如图界面，单击“继续”按钮，然后等待一会，等其自动完成一些设置。

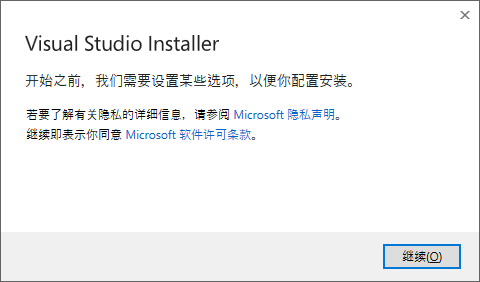


图10.2 visual studio 2022安装程序的启动界面

完成设置后会弹出如图界面，勾选上.NET桌面开发功能，在“安装位置”可选择安装路径，建议选择在C盘以外的地方进行安装，安装路径最好不要含有中文，一切准备就绪后单击底部右下方的“安装”按钮。

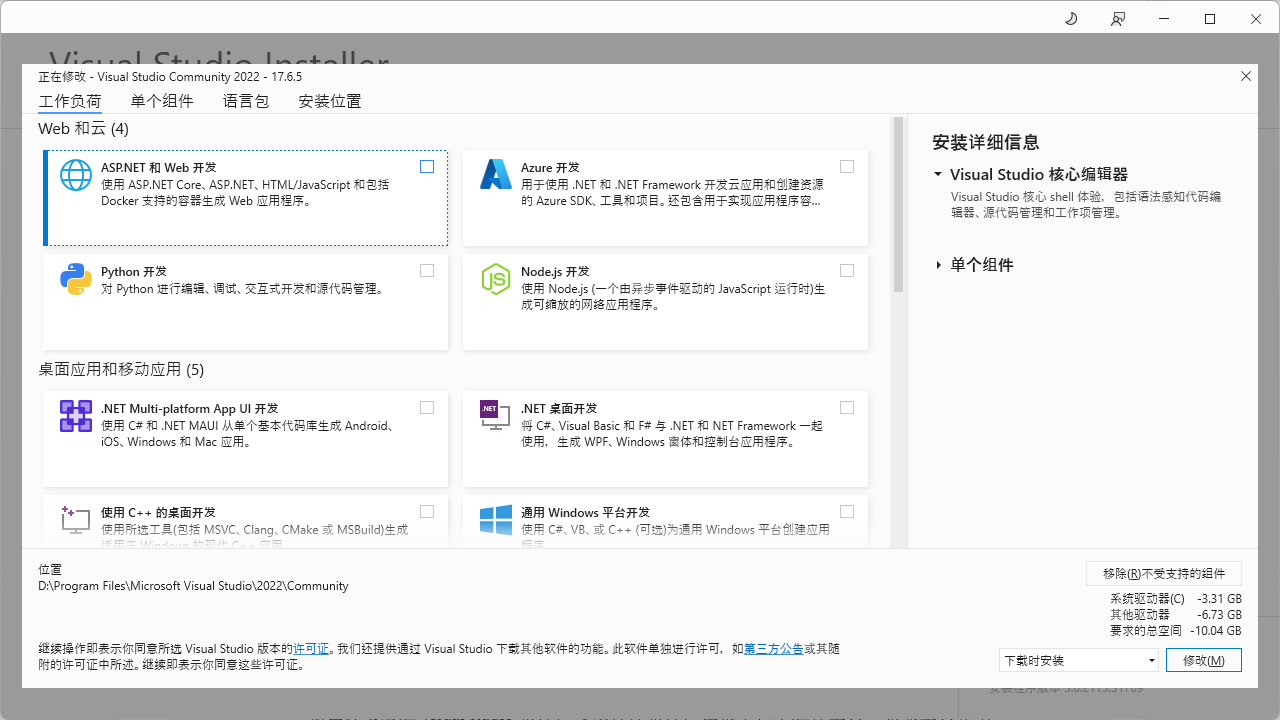


图10.3 visual studio 2022安装程序的主界面

如果已经安装过visual studio软件，而且没有安装.NET桌面开发功能，可以打开软件，如图所示，在菜单栏中选择【工具】→【获取工具和功能】中，勾选上.NET桌面开发功能，然后单击“修改”按钮，等待下载安装完成。

软件安装完成后，创建一个新项目试运行。

1. 运行visual studio 2022，出现如图10.所示界面，点击创建新项目。



图10.4 visual studio 2022 运行初始界面

1. 如图所示，在左边项目模板列表中找到windows窗体应用(.NET Framework)，选中之后点击“下一步”按钮。

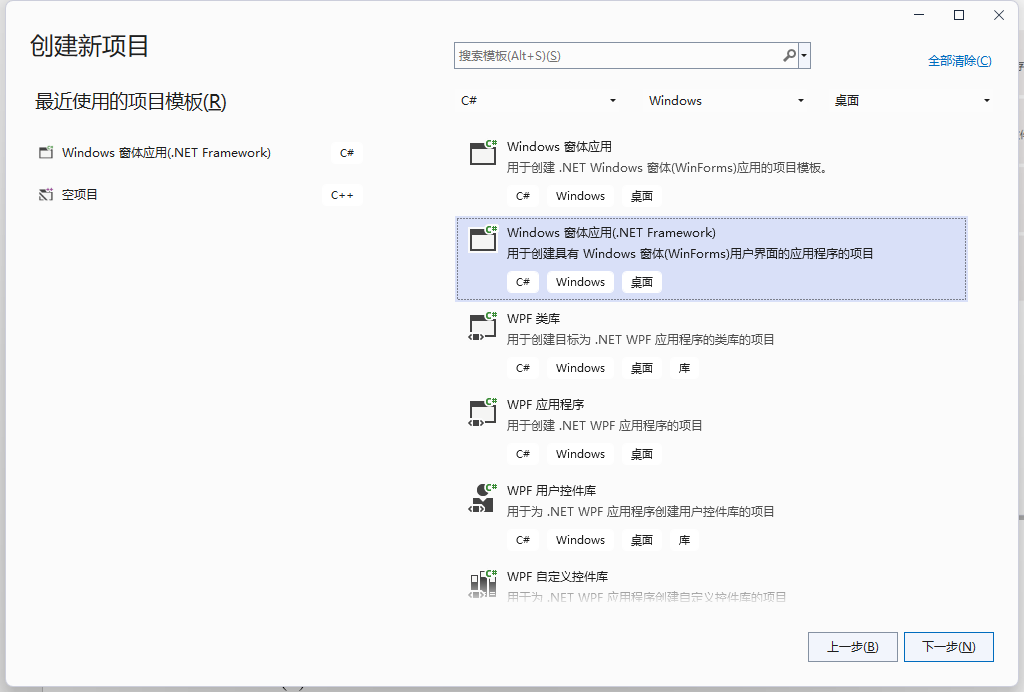


图10.5 visual studio 2022创建新项目界面

1. 如图所示，填写项目名称及解决方案名称，修改文件存放位置，注意最好不要出现中文。完成后点击“创建”按钮。

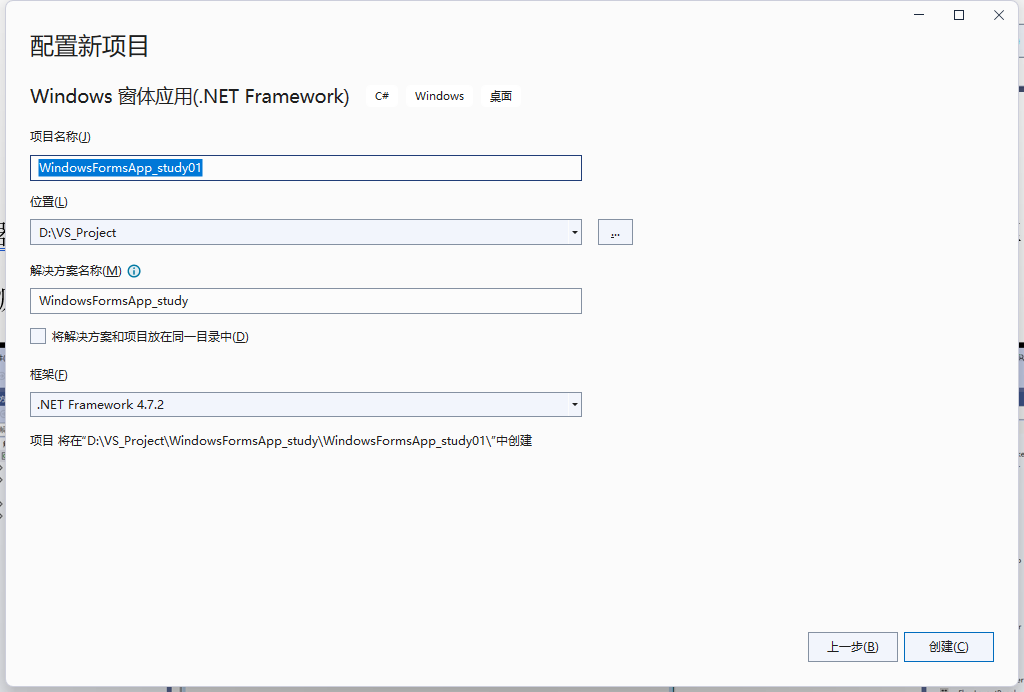


图10.5 visual studio 2022新项目配置界面

1. 项目创建完成后出现如图所示界面，可以点击“启动”按钮或者按下“F5”试着启动，没有报错并且出现一个空白的窗体就表示已经成功创建完成啦！

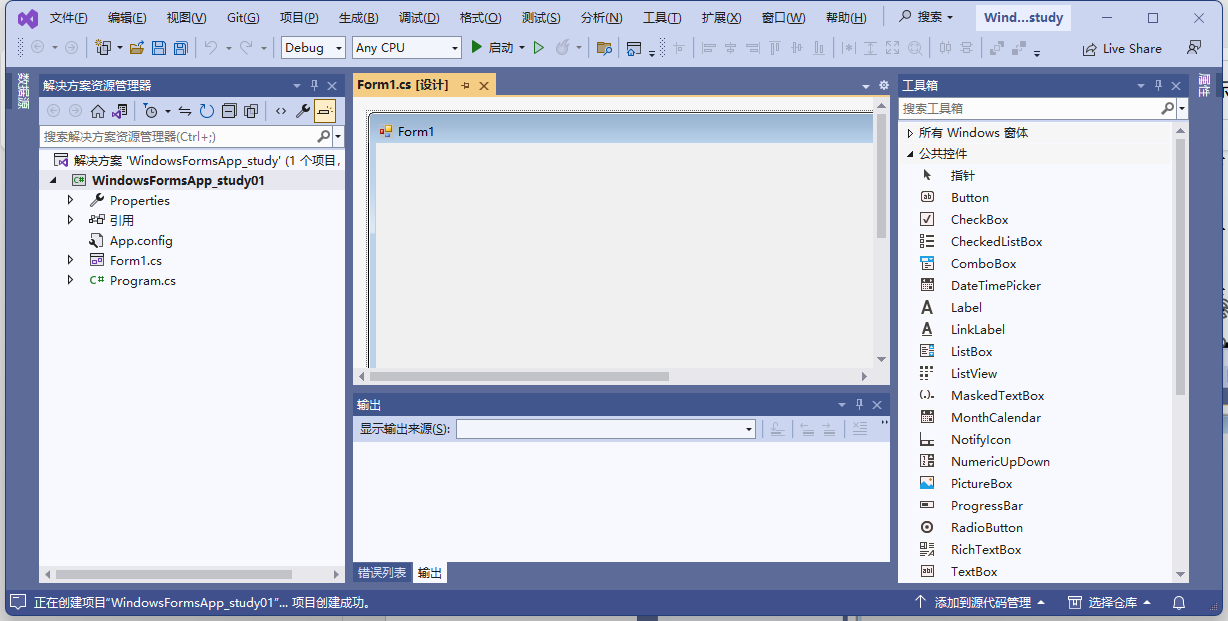


图10.1 新项目的初始界面

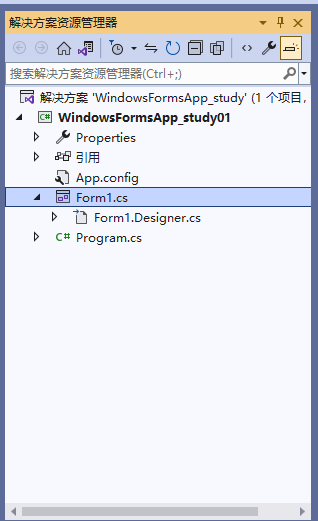
### 10.3 常用工具

简单介绍一下几个接下来学习需要用到的工具，有些工具默认并不会显示出来，可以在菜单栏中点击【视图】来打开需要的窗口，如图所示，



#### 10.3.1解决方案资源管理器

首先是“解决方案资源管理器”，如图所示，可在菜单栏中的【视图】→【解决方案资源管理器】中打开工具。



该工具是对Winform里的文件资源进行整合管理，我们项目的所有文件资源都可以在这里找到，简单介绍一下WinForm项目结构：

1. 引用：包括所有的系统库文件的引用依赖。当引用第三方库时也会显示在这里。
2. App.config：当前项目的配置文件。
3. Form1.cs：当前窗体的事件逻辑源码。后面我们的联合HALCON的代码基本都写在这个文件中。双击该文件可以打开窗体设计器。

-Form1.Designer.cs：当前窗体的控件布局源码文件，一般“不需要用户修改”，只要通过窗体设计器进行界面设计，代码就会自动生成。

注意：查看Form1.cs和Form1.Designer.cs文件时，会发现两个文件都定义了Form1类，该类使用了Partial关键词声明，其定义的类可以在多个地方被定义，最后编译的时候会被当作一个类来处理。因此两个文件各司其职，最后合并为一个类编译。要手动实现自定义窗体，可以添加自己的类，然后继承Form类即可

1. Program.cs：当前项目程序的主入口Main，启动项目，运行初始窗口。

#### 10.3.2 窗体设计器

如图所示，窗体设计器就是用来对窗体进行设计工作的，可以实现我们在设计过程对窗体的可视化。可以在上面讲的“解决方案资源管理器”中的“From1.cs”双击打开，在以后的设计过程中可能会用到两个以上的窗体，双击相应的窗体源码文件就可以打开对应的窗体设计器。

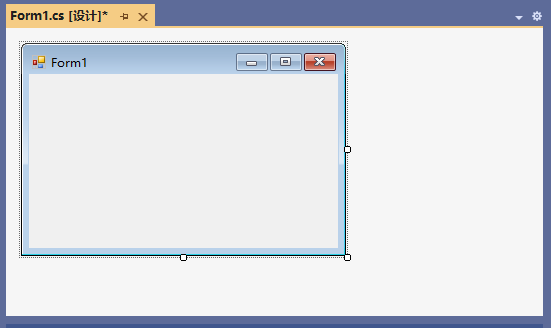


图10.5 窗体设计器

#### 10.3.3 工具箱

工具箱在界面设计中是最常用的工具，可在【视图】→【工具箱】中打开，我们在界面设计时所用到的大部分控件都来着于此，如图所示，里面包含了许多windows风格的控件。

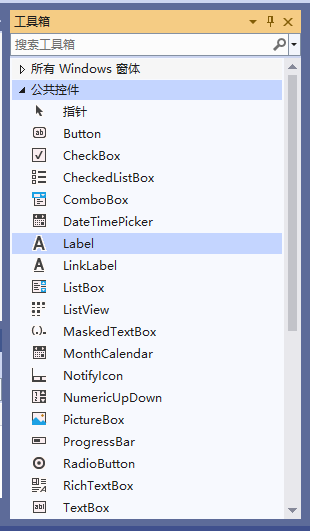
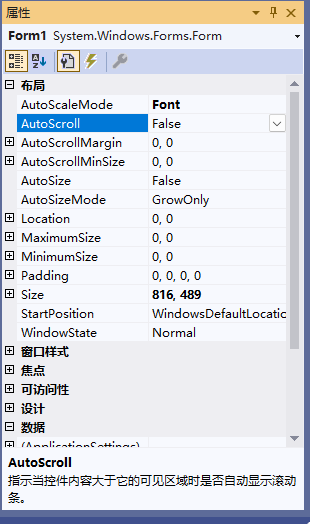


图10.6 工具箱

#### 10.3.3控件属性

我们可以通过调整控件属性来修改控件的外观、字体、显示样式等，也可以在控件属性中添加控件的触发事件，控件属性工具可在菜单栏中的【视图】→【属性】中打开，也可以按快捷键“F4”来打开。



### 10.4 winform常用控件介绍

Winform提供了许多Windows风格的现成控件，这些控件放在上一节提到的工具箱中，刚开始学习只要学那几个比较常用的就够了，其他的基本都大同小异，以后项目中需要用到再学习就行。

这里会介绍控件的基本使用方法，包括常用控件属性设置及事件的使用方法。

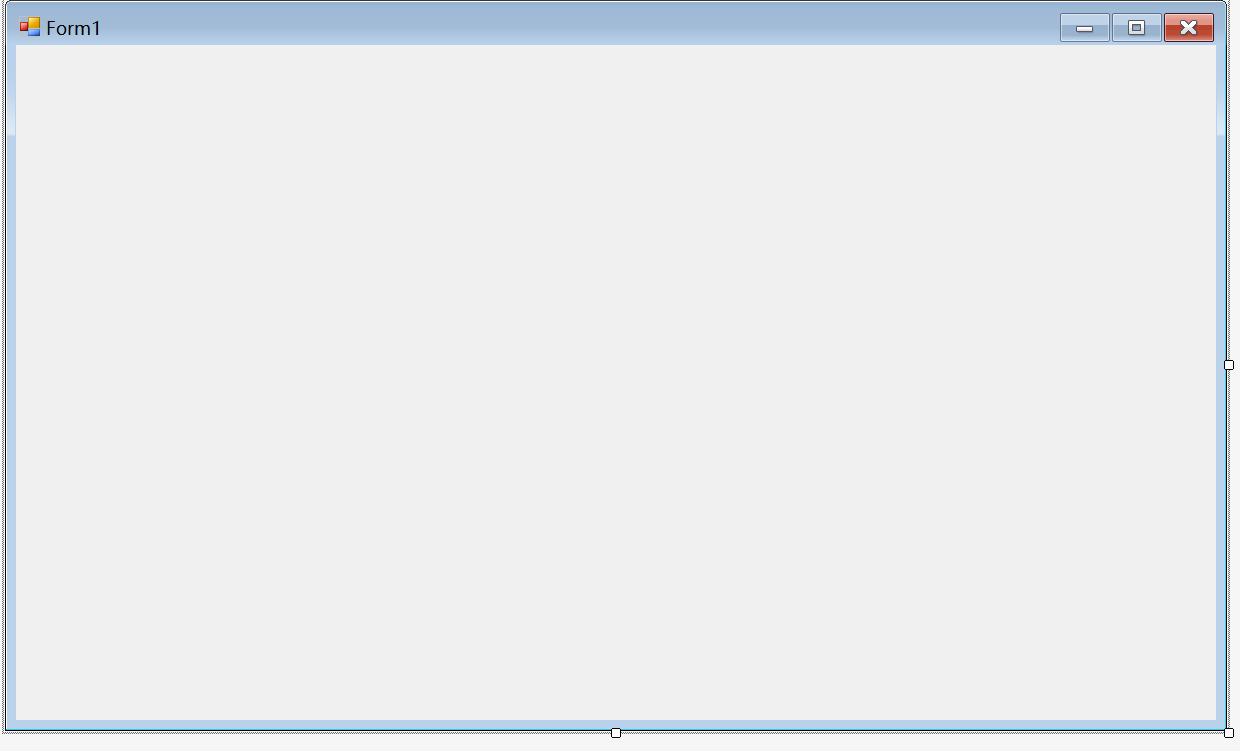
控件属性可以在设计时通过属性窗口进行设置，也可以在代码中通过窗体对象的属性进行设置。这里只讲解常用的功能性属性，一些外观的属性的设置比较简单可观，比如背景颜色、字体大小、显示样式等等，自行摸索即可。

下表是常用控件的功能，可在工具箱中先行找到其位置。

|  |  |
| --- | --- |
| 控件名称 | 控件介绍 |
| Form （窗体） | 可视化容器，用以承载其他控件 |
| Label（标签） | 用于显示文本内容 |
| TextBox（文本框） | 用于接收用户输入的文本 |
| Button（按钮） | 用于触发操作或事件。 |
| ComboBox（下拉列表框） | 用于选择一个选项 |
| RadioButton（单选按钮） | 用于选择一个选项 |
| ListBox（列表框） | 用于显示一个选项列表 |
| ListView（列表视图） | 用于显示和管理数据的列表视图。 |
| DataGridView（数据表格） | 用于显示和编辑数据表格 |
| PictureBox（图片框） | 用于显示图像 |
| CheckBox（复选框） | 用于选择一个或多个选项 |

#### 窗体

窗体是整个WinForm的核心，是一个可视化容器，用以承载其他控件，相当于一个平台让其他的控件能在这上面展现给用户使用。它是Windows应用程序的主要界面，用于显示和布局其他控件，并处理用户交互事件。



窗体常用属性：

name：窗体的名称（用于在代码中，标识全局唯一的这个窗体）

Text：显示在左上角的窗体名称

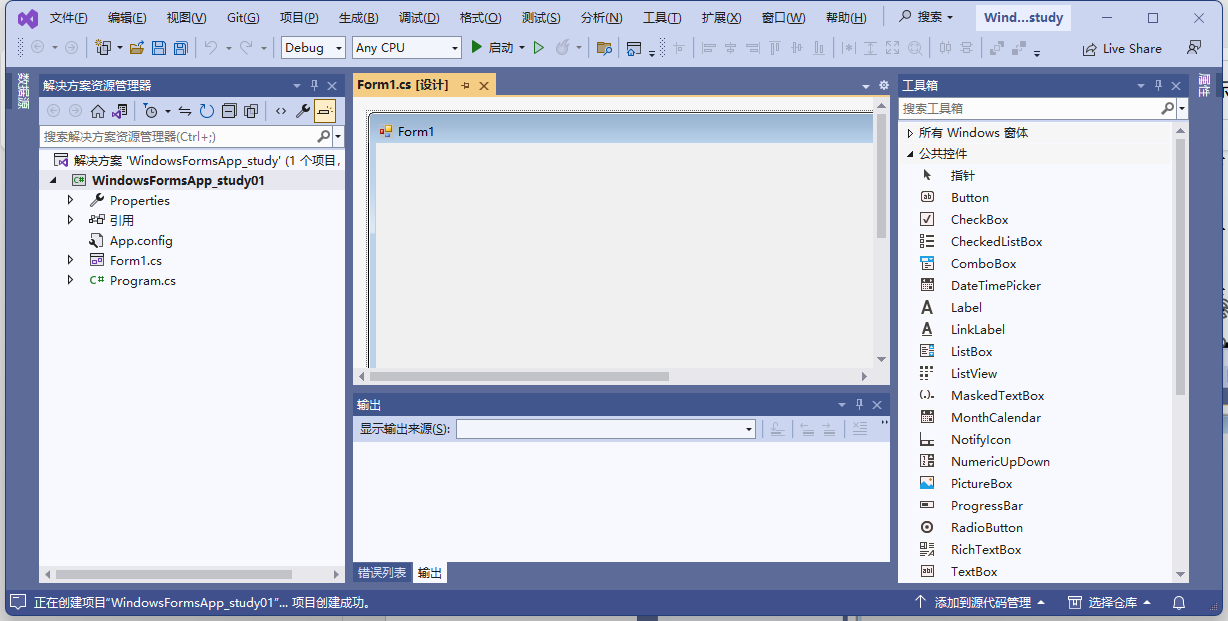
FormBorderStyle：窗体的[边框样式](https://so.csdn.net/so/search?q=%E8%BE%B9%E6%A1%86%E6%A0%B7%E5%BC%8F&spm=1001.2101.3001.7020)及窗体的大小是否可以调整（运行后）

### 案例：学生信息录入系统

接下来进行一个案例学习一下这些常用控件的用法，首先说明一下，因为我们的项目一般属于工程应用类，所以界面并不需要像我们常用的一些软件或者网页那般美观，简洁大方、功能俱到就可以了，所以本书并不会过多的涉及到控件外观的内容，有需要请自行学习。

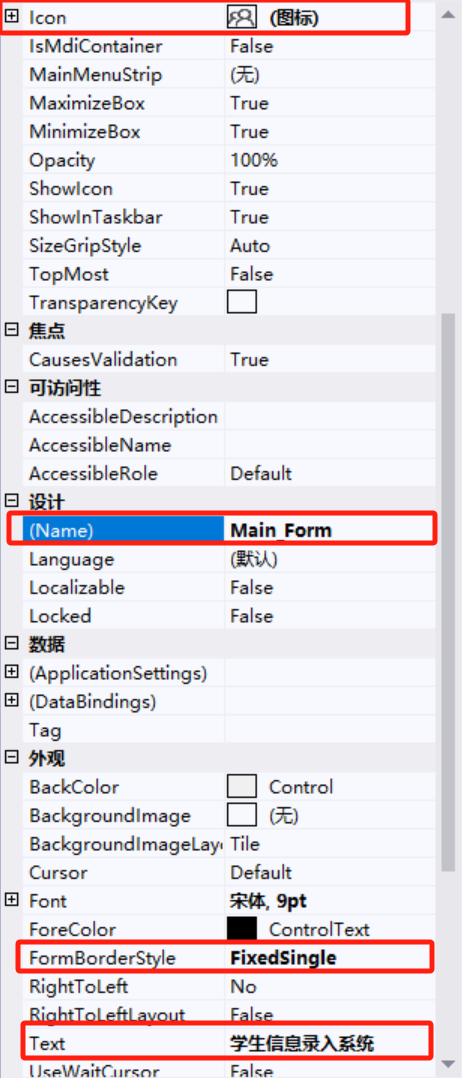
1. 创建新项目

首先按照上文所说创建一个新项目，创建完成后默认会有一个窗体。



1. 修改窗体属性

点击窗体，在窗体的属性窗口中找到Icon、（Name）、FormBorderStyle以及Text。如图所示进行修改。

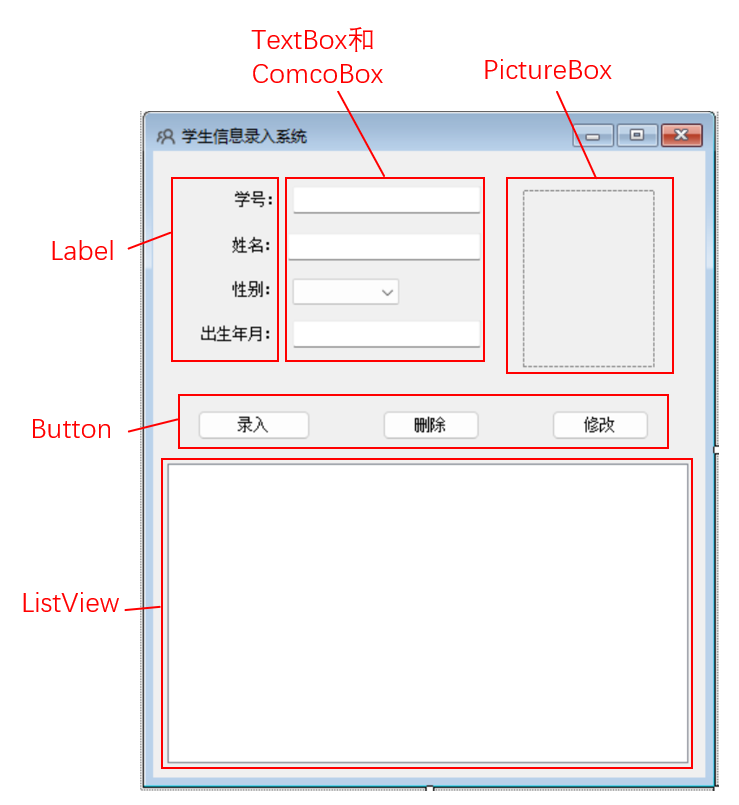


**说明：**

* Icon可以修改窗体右上角的图标，需要.ico格式的文件。.ico文件可以通过将其他图片格式的文件进行格式转换获得。
* （ Name）是控件在代码中的唯一标识符，在代码中可以通过这个来调用控件。根据控件的功能用途修改相应的名称，虽说支持中文，但是最好还是修改成英文格式，养成编程的好习惯。
* FormBorderStyle可以设置窗体的边框样式，这里并不想用户随意的缩放窗体的大小，所以修改为FixedSingle。
* Text是窗体左上角的标题文字，不同的控件显示的位置不同，可以通过修改Font、Fontcolor等选项来修改文字的样式、颜色等等。这里修改为“学生信息录入系统”。

1. 添加控件

如图所示，将各个控件从工具箱中拖出来，根据图中的控件文本显示修改控件的“Text”属性，调整控件大小以及位置。控件的大小及位置没有严格要求，整洁即可。所用到的控件有Button、Label、TextBox、ComboBox、PictureBox及ListView。

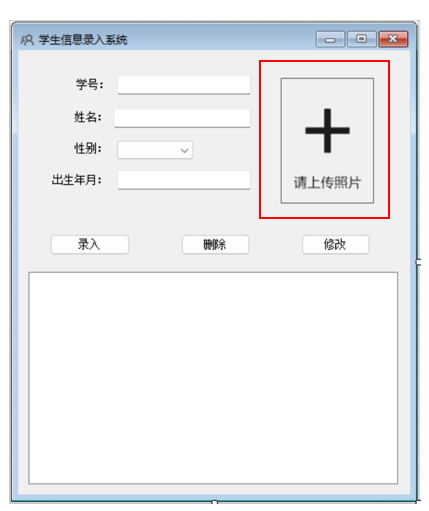


1. 修改控件属性

修改各个控件的（Name）属性，根据控件的功能进行修改，可以参考如图所示进行修改。Label控件在这里是为了提示用户其他控件的功能，不需要在代码中调用，所以不需要修改（Name）。



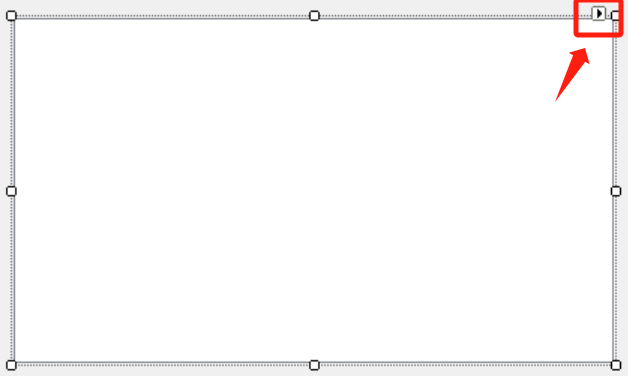
将pictureBox控件的“BorderStyle”修改为FixedSingle，增加一个边框，Image中选择一张图片用以提示该控件的功能（图片可在电脑自带的画图软件中进行制作），SizeMode修改为StretchImage，使得加载的图片平铺满控件。修改后效果如图所示。



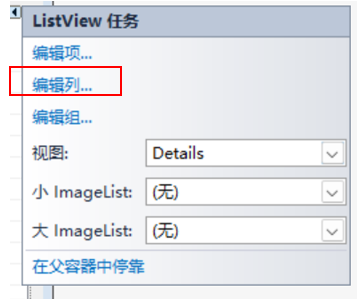
listView控件主要是用于显示和管理数据的列表视图。需要修改的属性有：

1. FullRowSelect修改为True，使得在列表中选中某项时，一整行都能高亮显示；
2. Gridhight修改为ture，显示表格线；
3. View修改为Details，显示的数据风格为详细，也就是所有数据都显示。

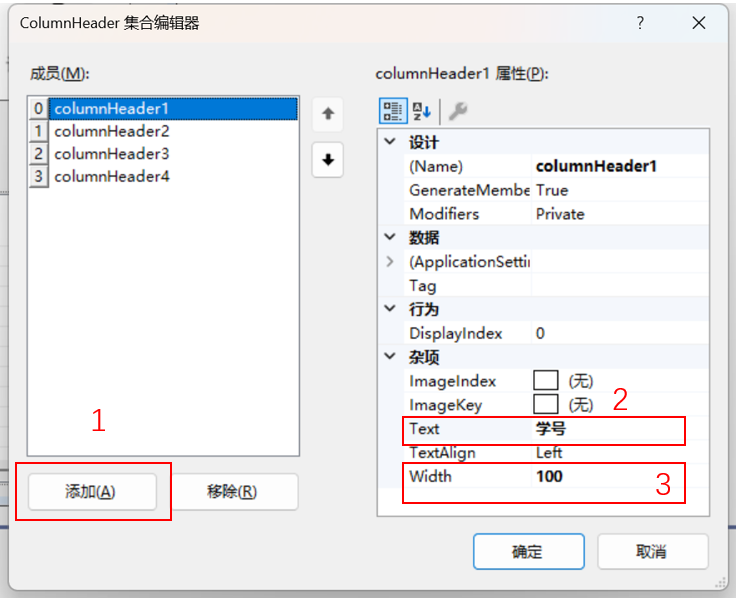
修改完属性后，还需要添加表头，当然也可以在代码中进行添加，这里展示在设计器中进行添加。如图所示，在选中listView控件的情况下，控件的右上角会出现一个三角按钮，点击按钮。



出现ListView任务栏，点击编辑列。



点击添加按钮，添加4个成员，分别修改其属性中的Text为学号、姓名、性别、出生年月。 然后修改Width，根据listview控件的长度进行分配，可以在listview的属性中的Size查看，这里的长度为400，所以每一列的长度分配为100。



完成操作后的效果如图所示。



1. 添加录入按钮的点击事件

双击录入按钮，将会生成一个以按钮（Name）属性加上事件名为命名的函数void button\_add\_Click(object sender, EventArgs e)，这个函数是按钮的点击事件响应函数。当用户在点击录入按钮时，就会执行该函数。

在函数中添加以下代码

//判断文本输入框中是否有数据，有数据才能执行添加

//没有数据则弹出提示框

if (textBox\_num.Text.Length > 0 && textBox\_name.Text.Length > 0

&& comboBox\_sex.Text.Length > 0 && textBox\_time.Text.Length > 0)

{

//判断图片保存的文件夹是否存在，不存在则创建文件夹

string path = Application.StartupPath + "\\img\\";

if (!Directory.Exists(path))

Directory.CreateDirectory(path);

//判断图片是否存在，不存在则弹出提示框提醒

if (Ispic\_ok != false)

{

//将需要添加的数据中的学号与表中已有的学号进行对比

//如存在相同学号则弹出提示框提示

bool Is\_repeat = false;

for (int i = 0; i < listView\_message.Items.Count; i++)

{

if (textBox\_num.Text ==

listView\_message.Items[i].SubItems[0].Text)

{

MessageBox.Show("学号重复了");

Is\_repeat = true;

break;

}

}

if (!Is\_repeat)

{

listView\_message.Items.Add(new ListViewItem(new string[] { textBox\_num.Text, textBox\_name.Text, comboBox\_sex.Text, textBox\_time.Text }));

PictureBox\_show.Image.Save(path + textBox\_num.Text + ".png", System.Drawing.Imaging.ImageFormat.Png);

}

}

else

{

MessageBox.Show("请上传照片！");

}

}

else

{

MessageBox.Show("数据不能为空");

}

1. 添加删除按钮的点击事件响应函数

双击删除按钮以添加点击事件响应函数，在函数中添加以下代码：



1. 添加修改按钮的点击事件响应函数

双击删除按钮以修改点击事件响应函数，在函数中添加以下代码：



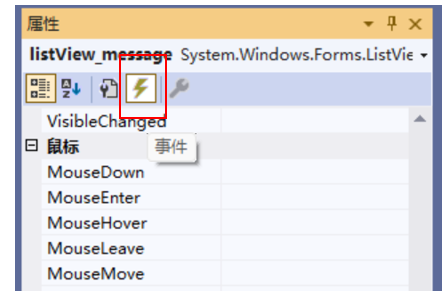
1. 添加pictureBox控件的点击事件响应函数

双击pictureBox控件以修改点击事件响应函数，在函数中添加以下代码：

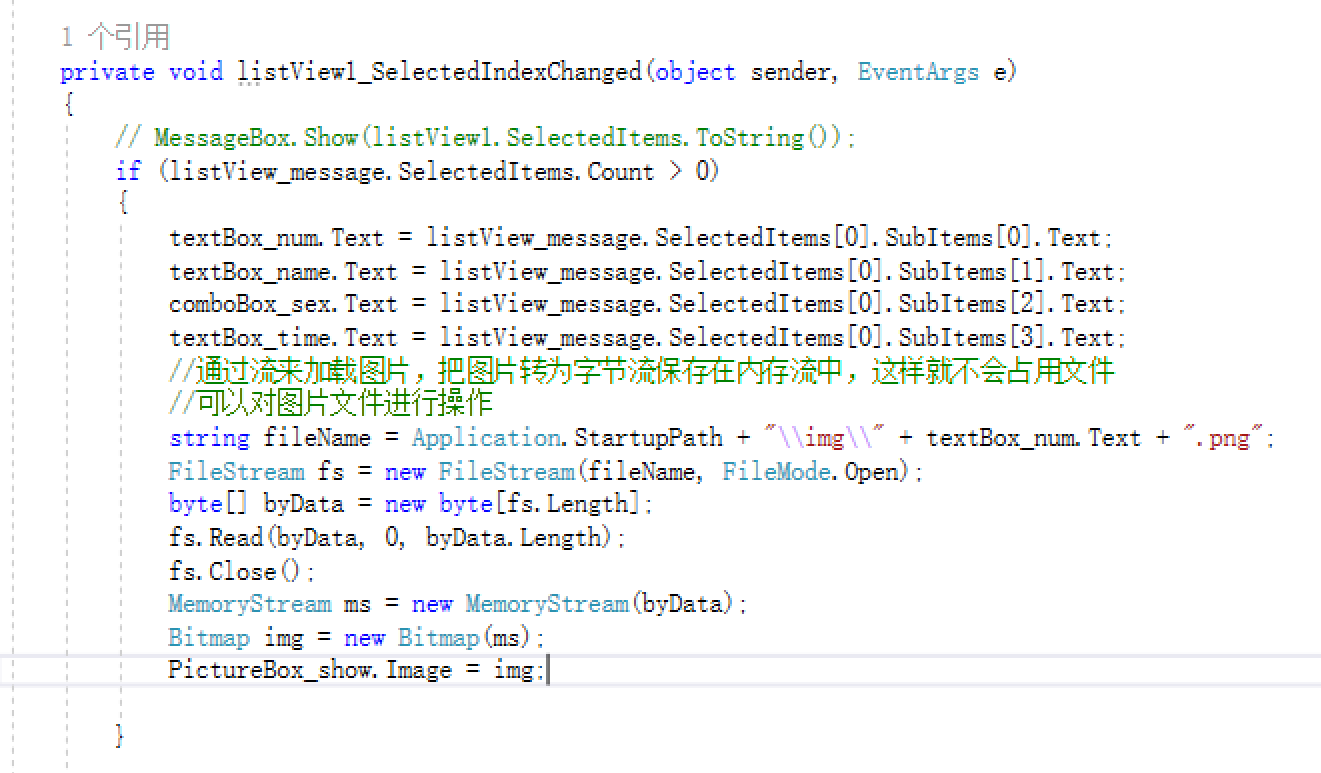


1. 添加listView控件的SelectedIndexChanged事件响应函数

如图所示，在listView控件的属性栏的上方找到一个闪电状的按钮



点击之后会出现该控件默认所能响应的事件，找到SelectedIndexChanged事件，双击事件以添加事件响应函数。在函数中添加以下代码。



到此就完成了所有工作，点击运行试着运行代码，并尝试各种功能是否正常。

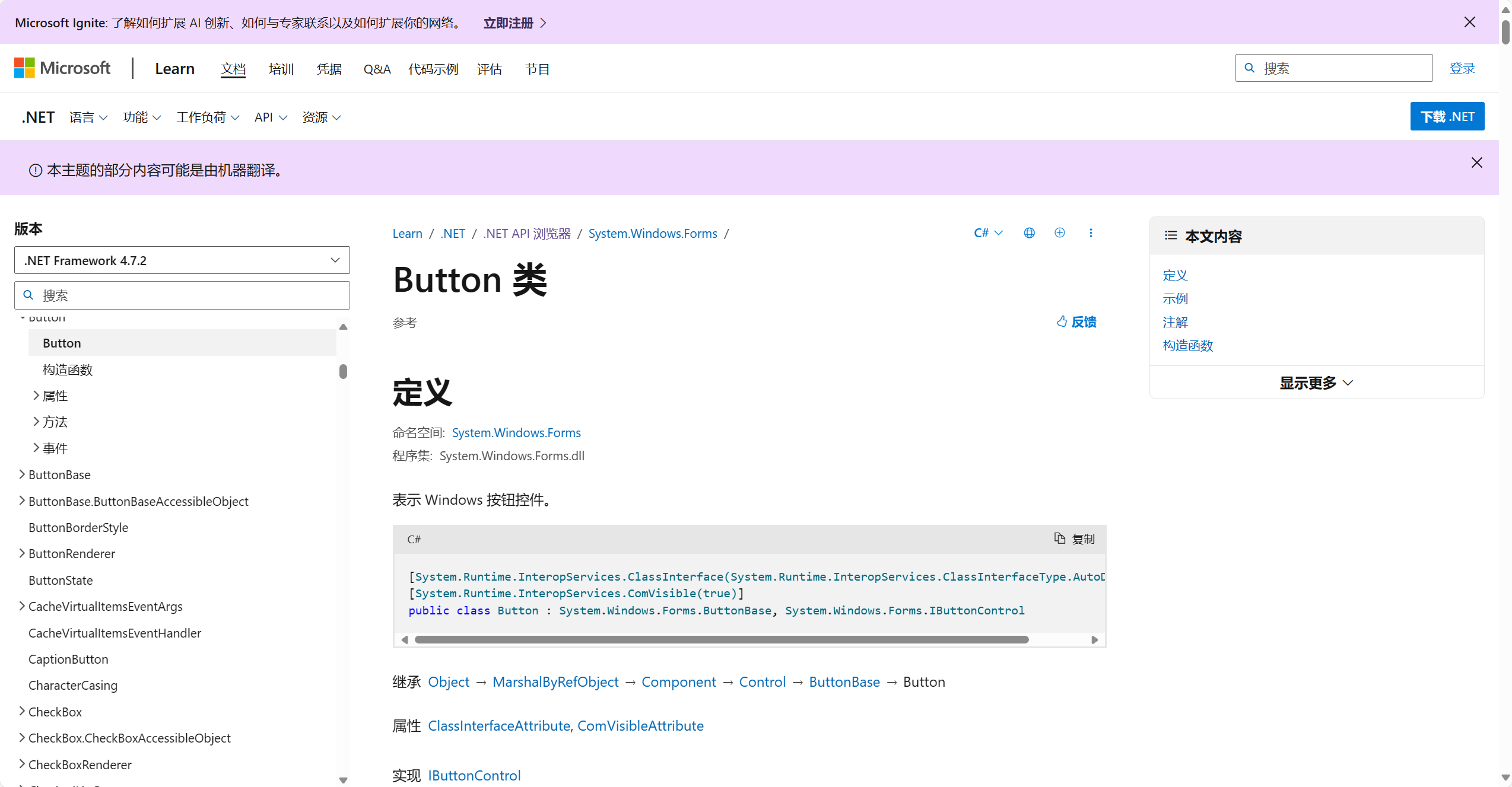
小技巧：我们点击窗体，这里出现很多的属性，我们可以在浏览器中搜索“.NET API 浏览器”，选择对应的框架进行搜索，这里我们用的是.NET Framework 4.7.2的。



选择完成后输入想要查看的控件英文名称，比如搜索button，就会出现有关button的资料，我们选择查看button类，也就是第一个。



这里面就有关于button类的详细说明，像属性、事件等函数都会有说明，当然在winform中编写代码也会有说明，只是没有这里的这么详细，在编写代码过程中遇到不懂的都可以来“.NET API 浏览器”中进行查找。



## 第11章 Halcon联合C#

### 11.1 halcon代码导出

首先介绍一下如何将halcon的程序变成winform能识别的编程语言也就是C#。打开我们在halcon中已经写好的代码，这里以水果分类的推理过程的代码为例子。如图所示，在菜单栏中选择【文件】→【导出程序】，这里记得在导出程序之前最好保存一下文件，不然导出的就是未保存之前的代码了。



选择导出的程序存放路径，然后导出程序的类型，这里选择“C#-HALCON/.NET”，编码选择“UTF-8”，确认无误后点击【导出】。



完成以上操作后打开我们导出的文件，里面就是用C#语言编写的代码了，这里面的代码很长，但是我们刚开始接触，只需要知道几个点就够了，下面是代码的结构，大家对照着导出来的代码进行比对，我们主要关注外部函数和action（）函数就可以了，所以只要知道action（）函数的位置，在其之上除了重构函数之外就都是外部函数。action（）函数位置可以用搜索来查找，快捷键一般都是ctrl+F。

public partial class HDevelopExport

{

#if !(NO\_EXPORT\_MAIN || NO\_EXPORT\_APP\_MAIN)

public HDevelopExport()

{

类的重构函数内容，用不上，不用管

}

#endif

接下来到action（）函数前的一大串都是外部函数

#if !NO\_EXPORT\_MAIN

// Main procedure

private void action()

{

变量申明

try

{

halcon代码导出内容

}

catch

{

释放内存

抛出捕获到的错误信息

}

释放内存

}

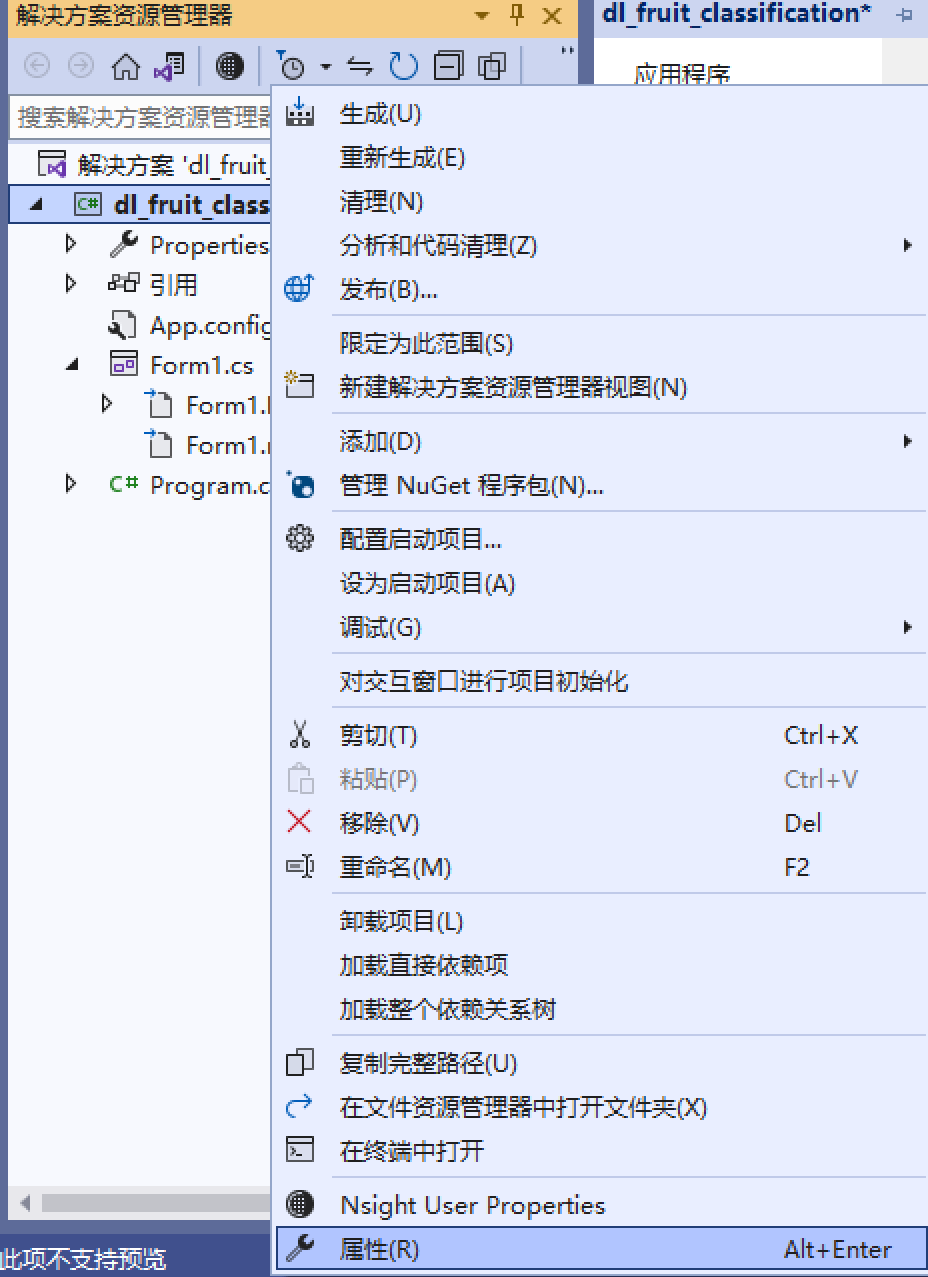
#endif

后面那段是main函数，用不上

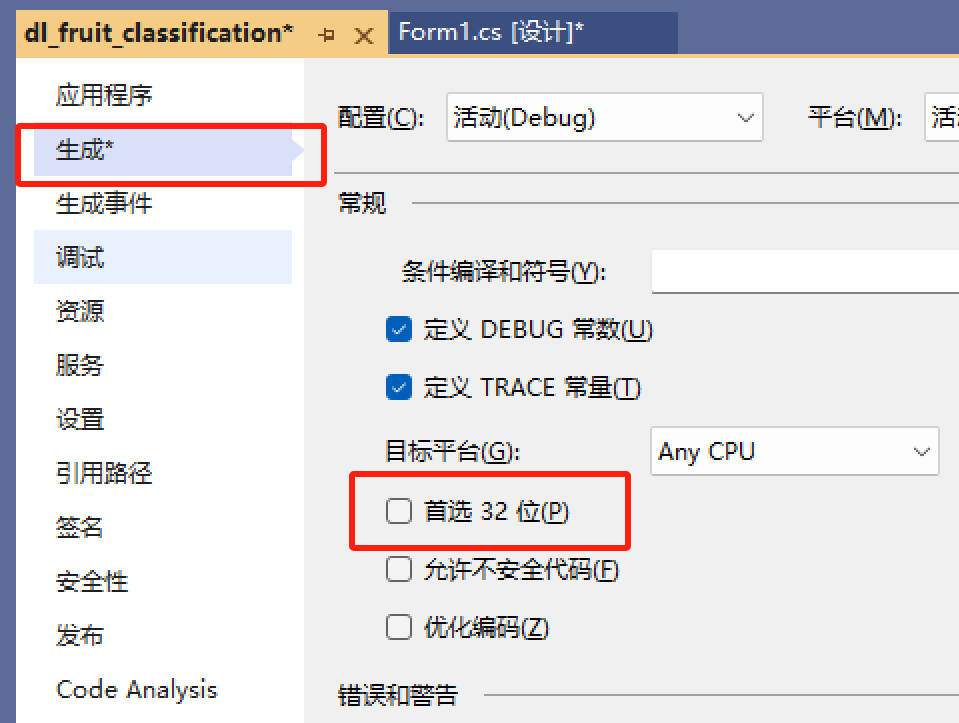
### 11.2环境配置

接下来进行项目环境配置。

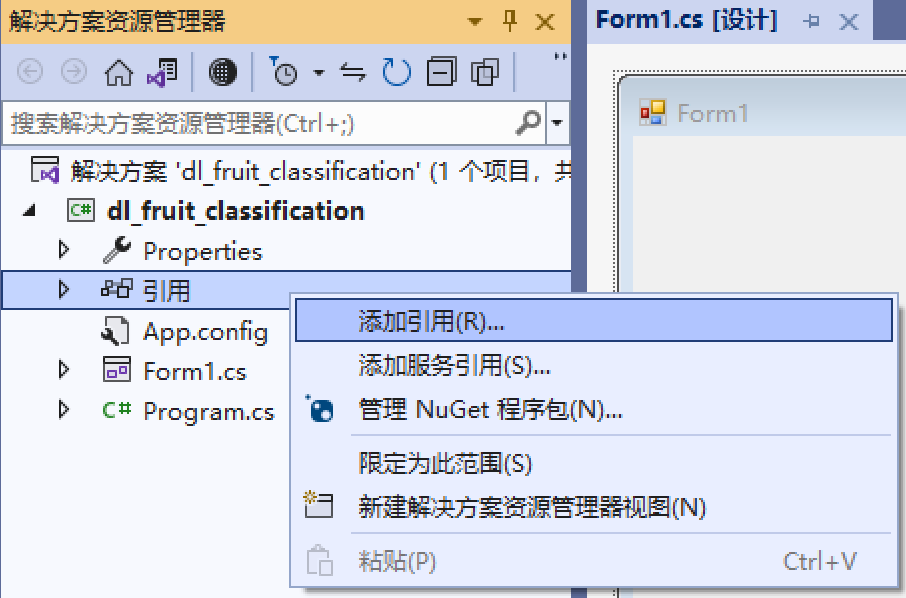
1. 新建一个winform的新项目，在“解决方案资源管理器”中右键本项目打开选项卡，找到【属性】。



打开之后找到【生成】选项中的“首选32位”，将其取消勾选。



在“解决方案资源管理器“中找到“引用”，右键选择【添加引用】



点击【浏览】



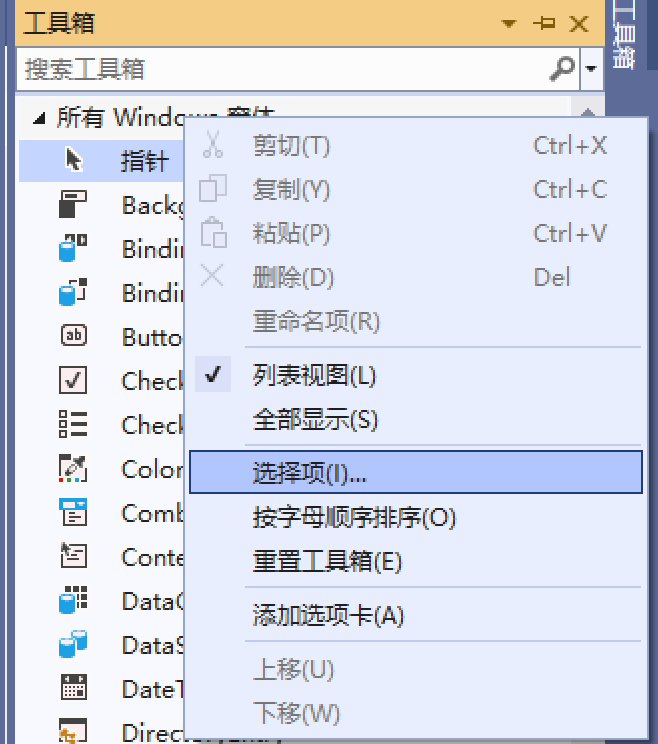
在halcon的安装目录下找到halcondotnet.dll和hdevenginedotnet.dll两个文件，选择后点击【添加】。



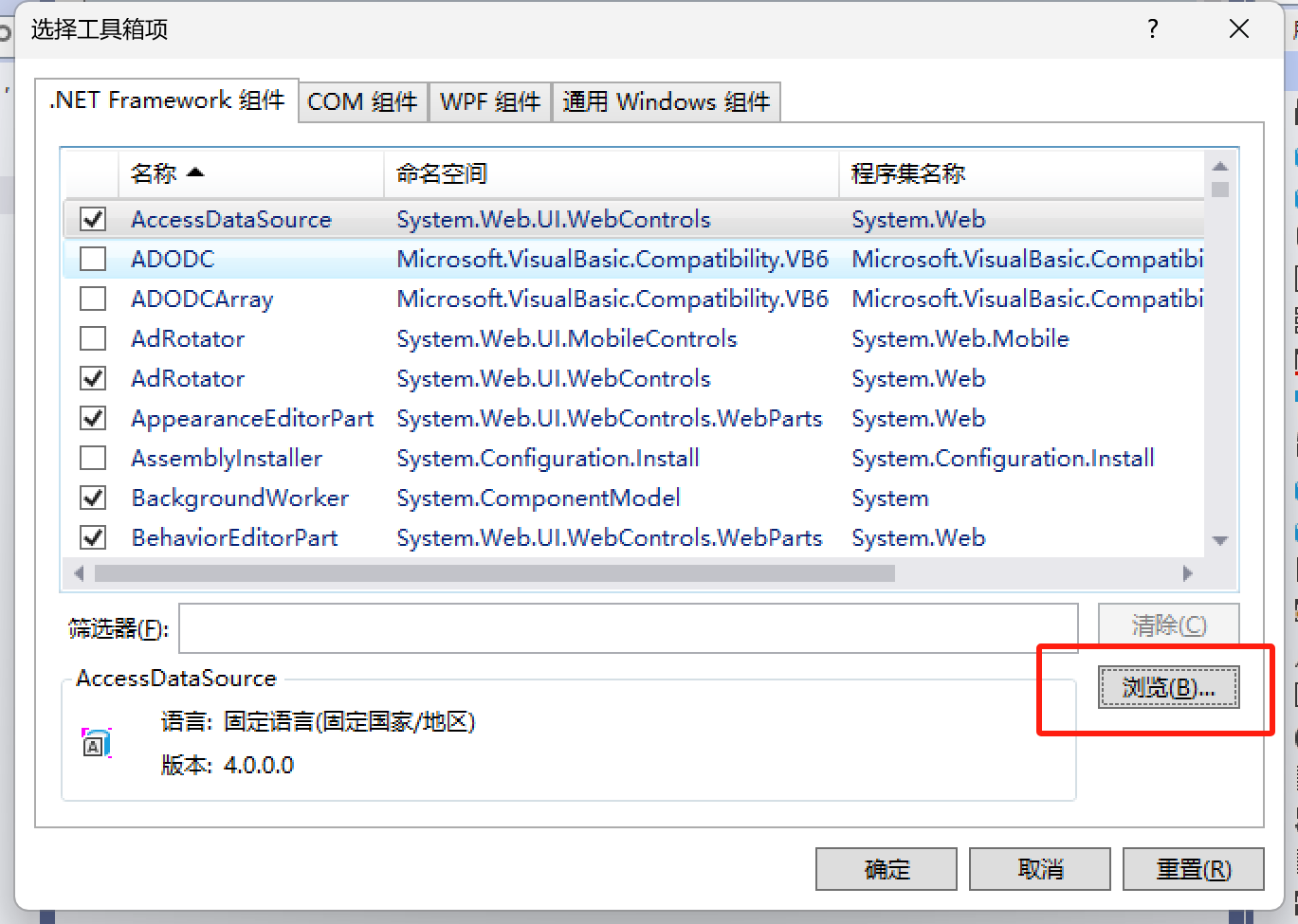
然后点击【确认】按钮。



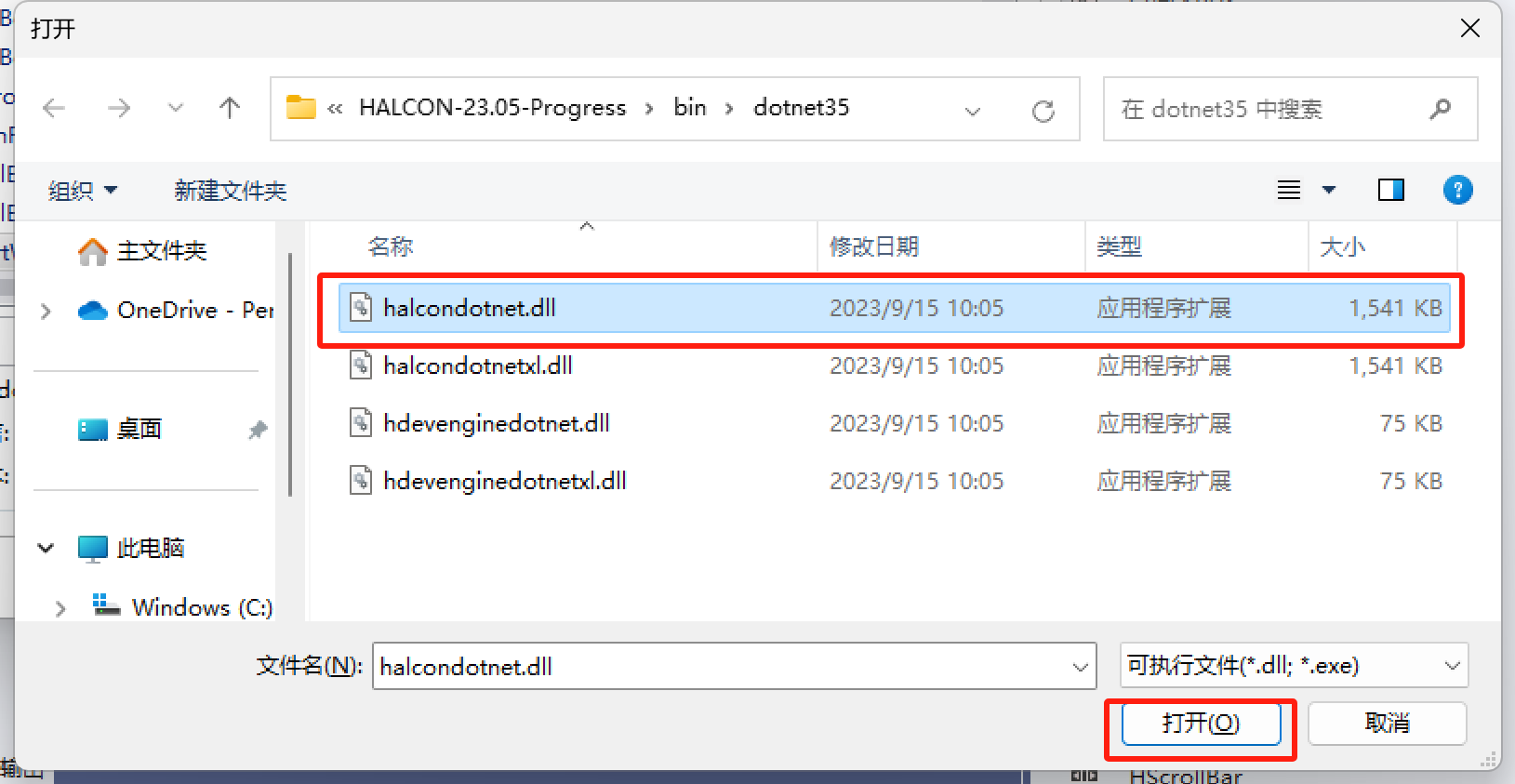
到窗口设计器中的工具箱中右键打开选项卡，从中找到【选择项】。



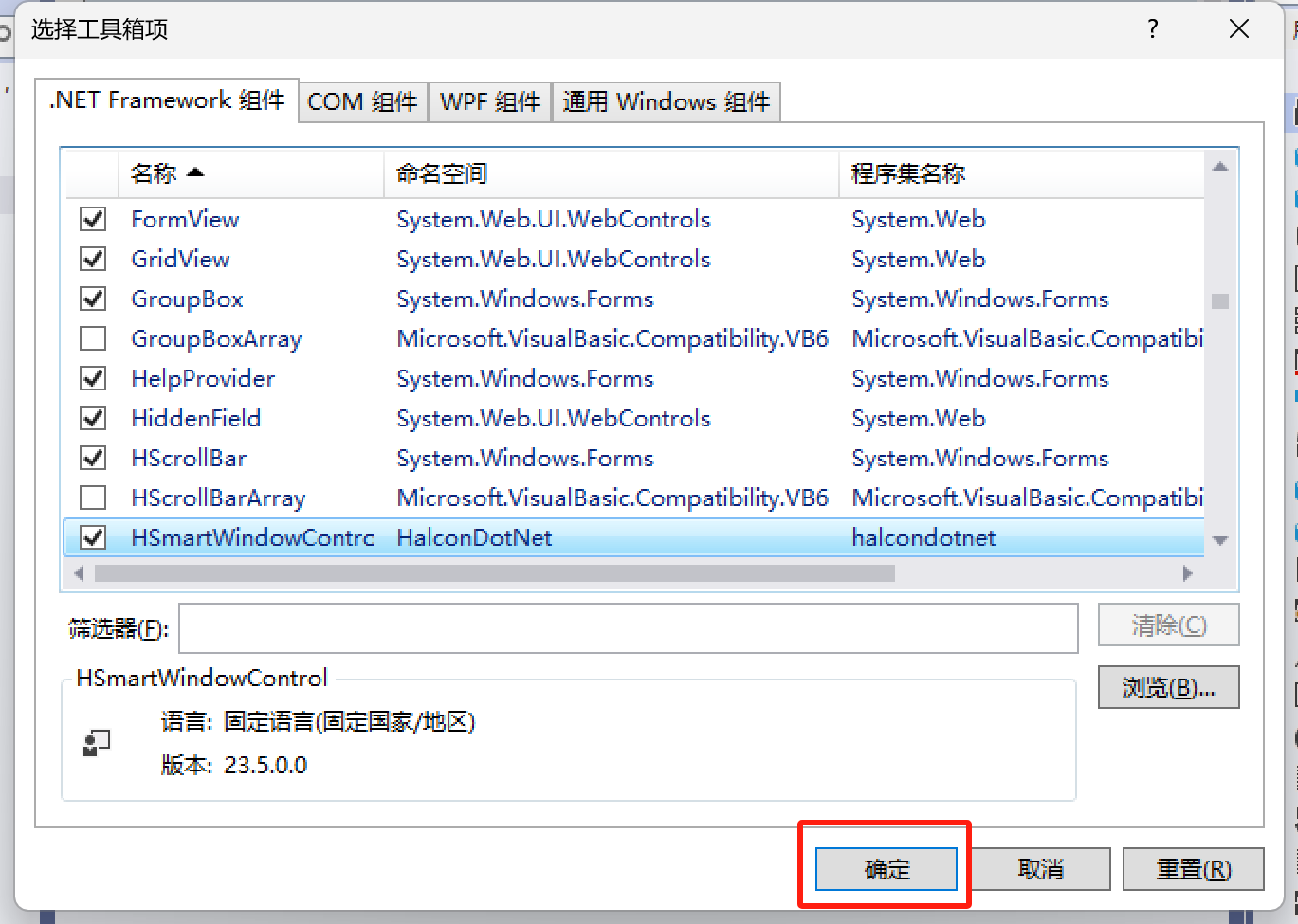
点击【浏览】按钮



找到刚刚添加文件的路径，选择halcondotnet.dll文件，点击【打开】按钮。

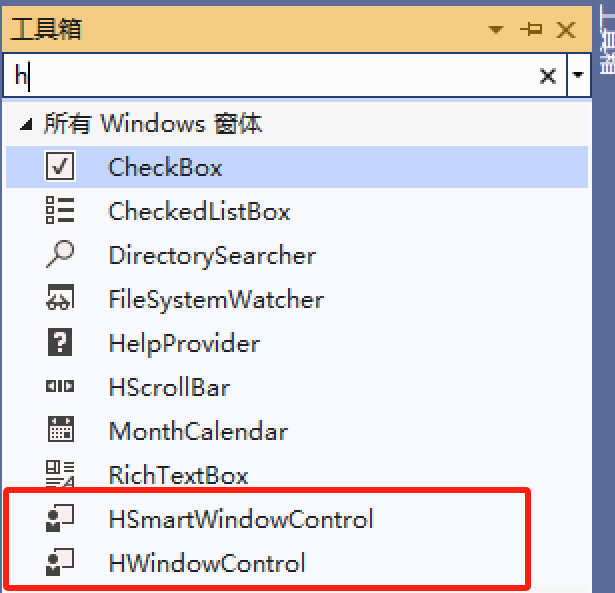


点击【确认】按钮。



这样就可以在窗口设计器的工具箱中找到HWindowControl和HSmartWindowControl这两个halcon窗体控件了。HSmartWindowControl可以很方便的实现图像的自适应显示以及响应鼠标事件实现图像的缩放、平移，但是不支持不支持传统HWindowControl的Draw\_\*函数。HWindowControl虽然支持传统的Draw\_\*函数，但是想要实现图像的自适应显示需要通过计算图像及窗口大小比例来实现，而处理窗口交互则需要依靠winform本身的鼠标事件来实现。

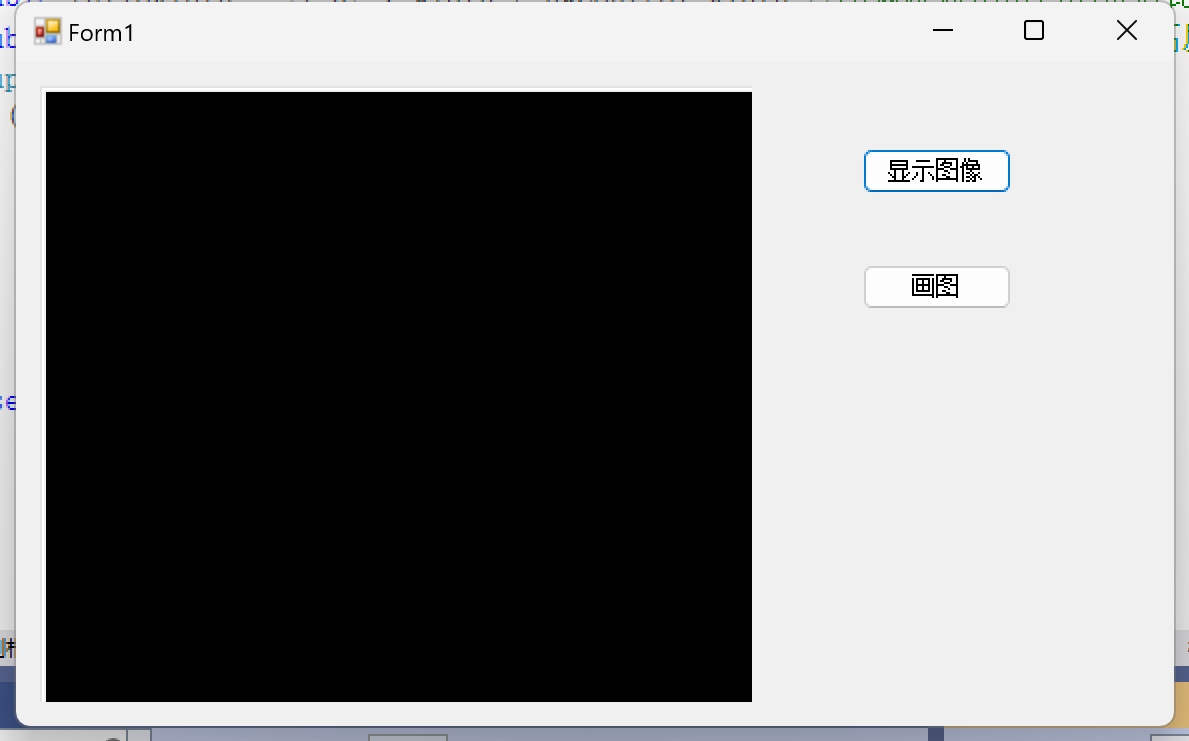
两种窗体控件各有优劣，但是总体上来说HSmartWindowControl会比较好一些，但是在工程应用中，并不是很需要频繁的平移缩放操作，反而是需要进行roi区域的划定，所以本书只讲解HWindowControl控件，对HSmartWindowControl有需要的请自行查阅资料学习。



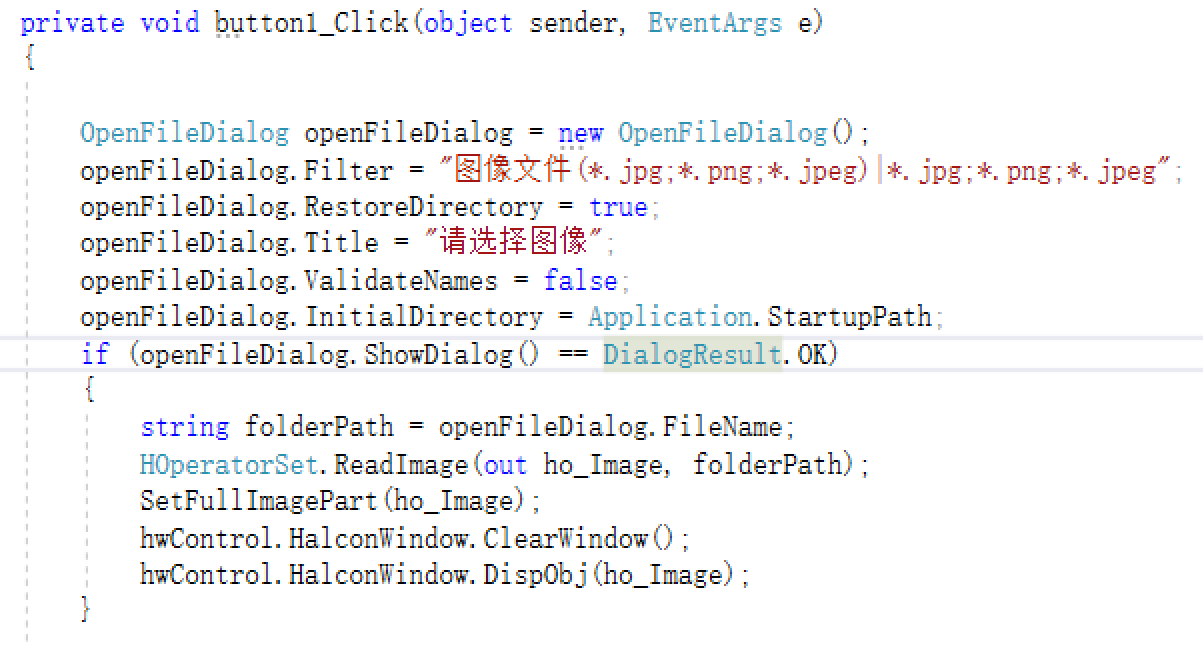
### 11.3 HWindowControl控件常用用法

接下来演示一下用HWindowControl如何显示图片，并让其按原比例缩放以适应窗口大小。如图所示，在窗口设计器中拖入一个HWindowControl控件和一个button控件。这里各个控件的（name）属性如下,

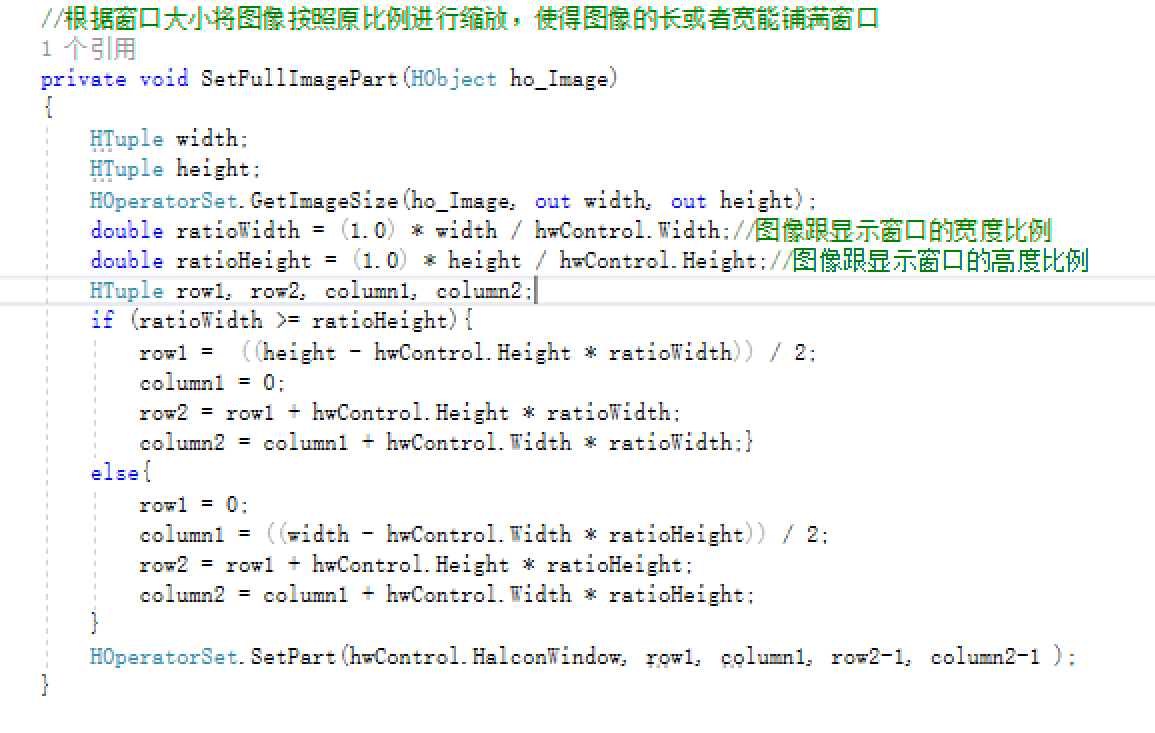
|  |  |
| --- | --- |
| 控件 | （name） |
| HWindowControl | hwControl |
| 显示图像按钮 | bnt\_read\_img |
| 画图按钮 | Btn\_draw |



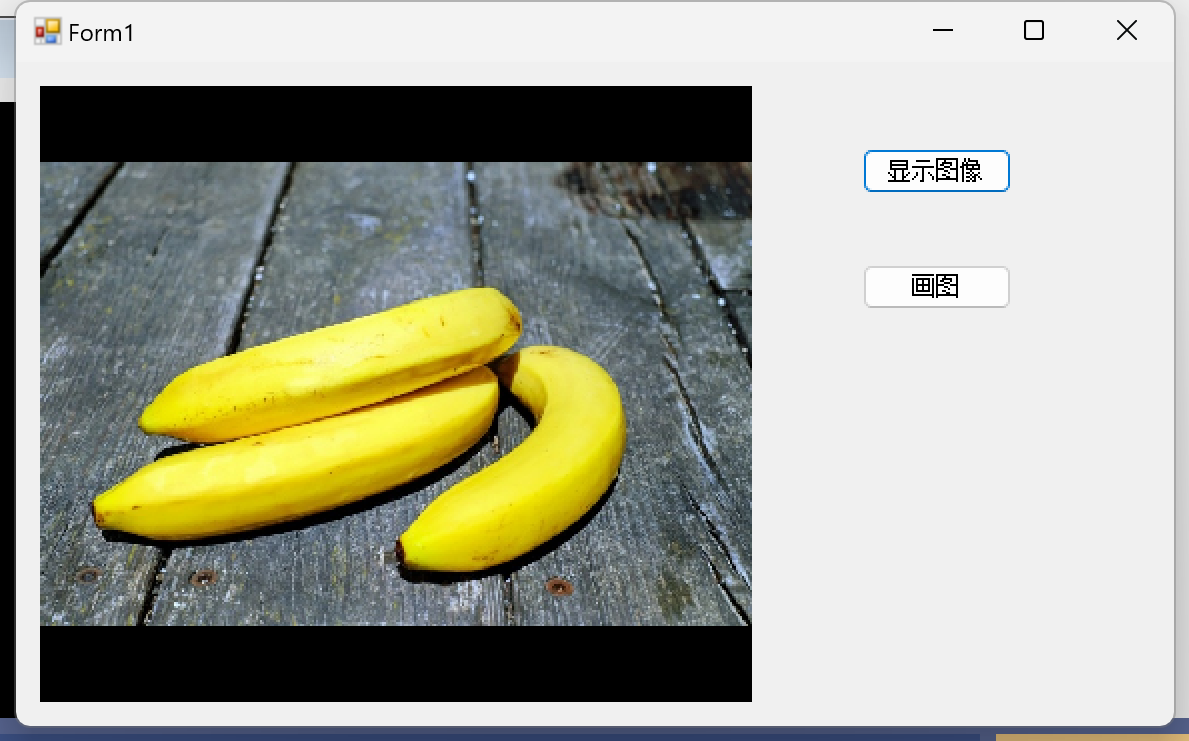
双击“显示图像”按钮，添加一个响应按钮点击事件的函数，利用OpenFileDialog类弹出选择文件对话框，用以获取图片路径，在获取到图片路径后读取图片，并显示到窗口中。



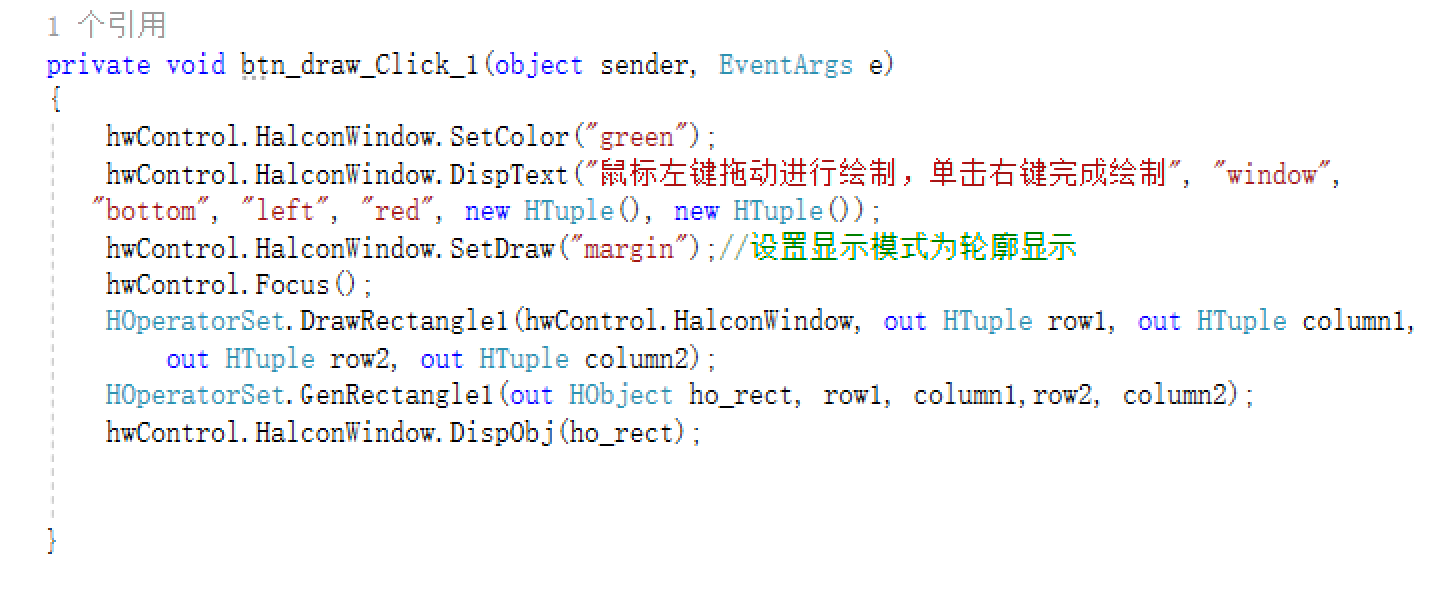
其中的SetFullImagePart（）函数是自定义函数，需要进行定义，在代码中添加函数的定义，并写入以下代码。这部分代码的作用将图像按照原比例铺满整个窗口并居中显示，基本思路是：计算传入图像与窗体控件的宽和高各自比值，按照较大的值来进行计算坐标。主要是利用halcon的setpart算子来实现，这个算子作用是修改图像的显示部分。就是显示传入的两个坐标组成的矩形区域的图像，如果矩形区域尺寸超过图像原本区域，则用黑色像素来替代。



完成后就可以执行程序了，在运行的界面点击“显示图像”按钮，选择一个图像文件后可以看到如图所示的画面，选择的图像居中并铺满了整个窗口。

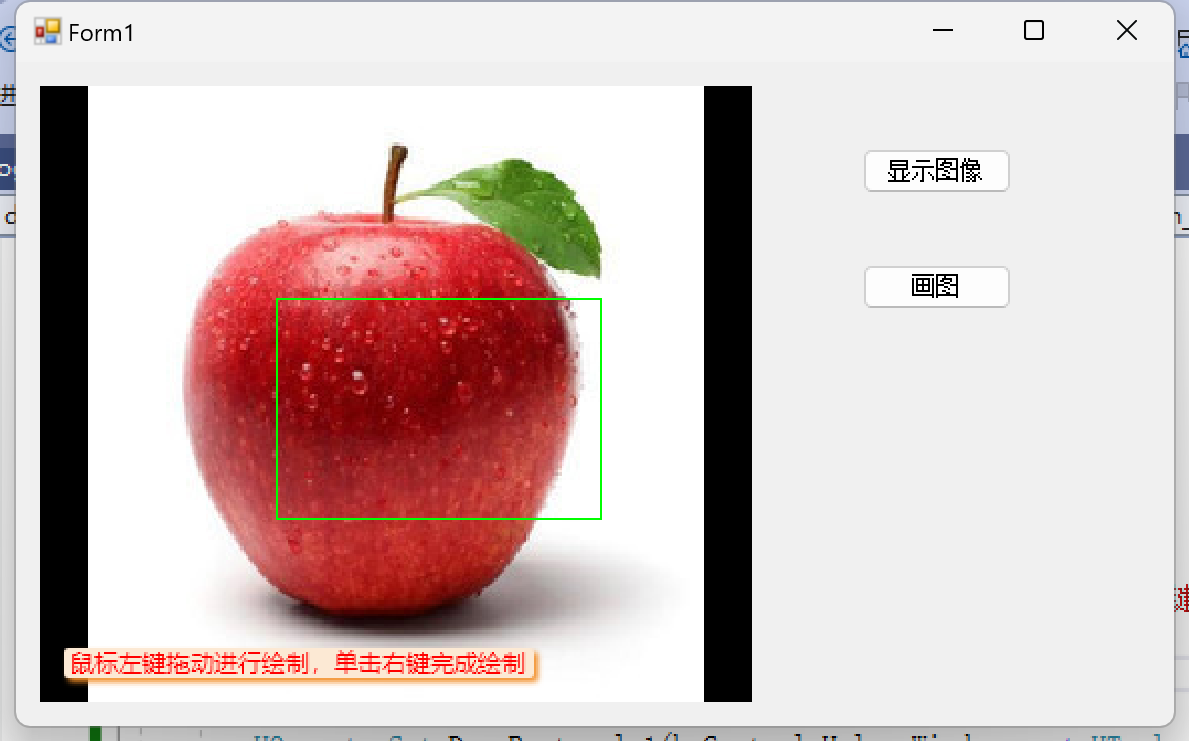


有时我们需要在窗口上进行绘制roi的操作，接下来介绍如何实现。双击“画图”按钮添加一个按钮点击事件的响应函数，在函数内编写以下代码，实现的功能是在窗口进行矩形roi的绘制。



其中dispText算子是在窗口上显示文本信息，setdraw是设置窗口的显示模式为轮廓显示，draw\*是在窗口上进行形状绘制，gen\*算子是根据窗口绘制时输出的坐标进行roi区域的创建，不同的形状对应不同的算子，比如绘制圆形roi用到的就是DrawCircle和GenCircle算子，依次类推。

来看一下实现的效果，点击画图按钮，在窗口进行绘制一个矩形区域，鼠标右键结束绘制。可以用disptext来在窗口上显示信息，用来提示用户。

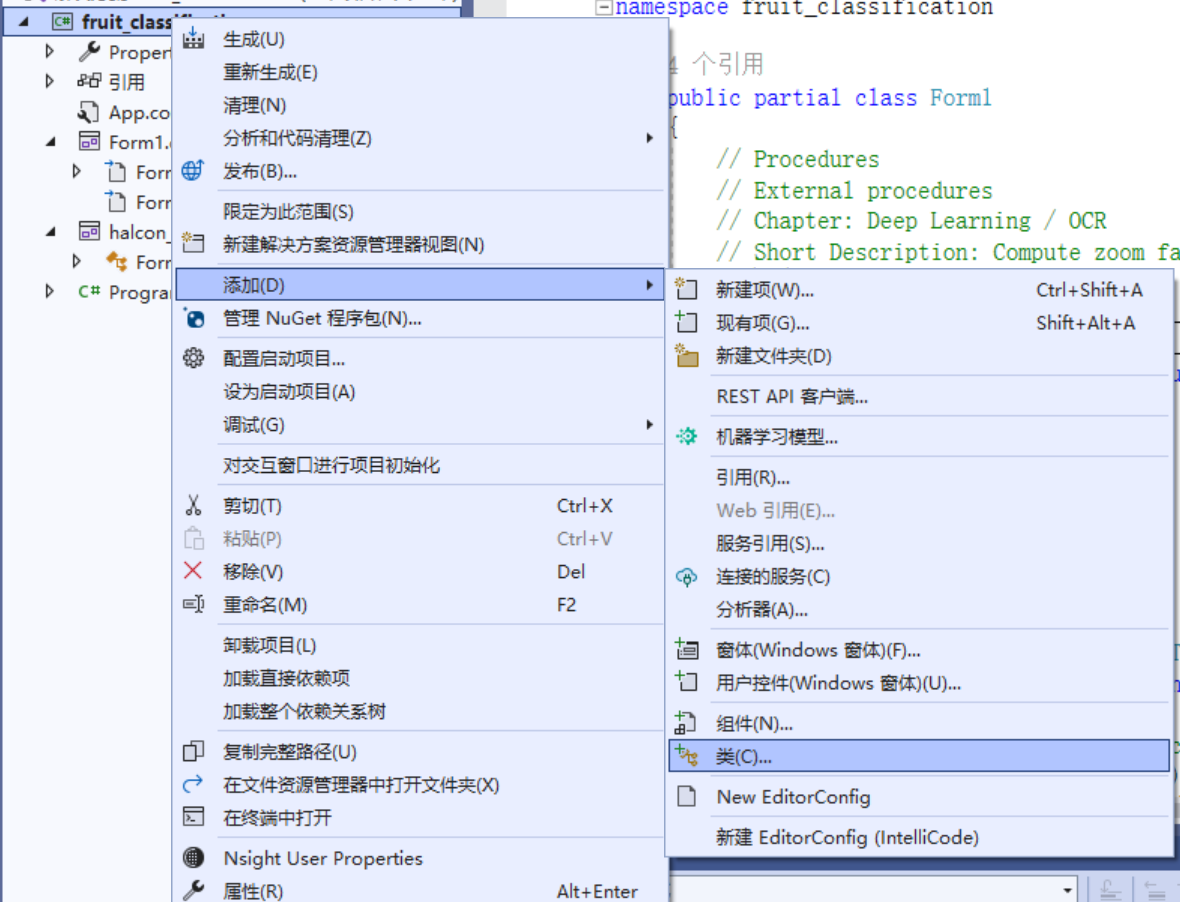


关于HWindowControl这个控件就介绍到这里，还有一些像鼠标平移、缩放的交互操作这里就不介绍了，因为其实在在封装界面中用的比较少。到此关于halcon联合winform的环境配置以及常用的控件的用法就已经介绍完了。

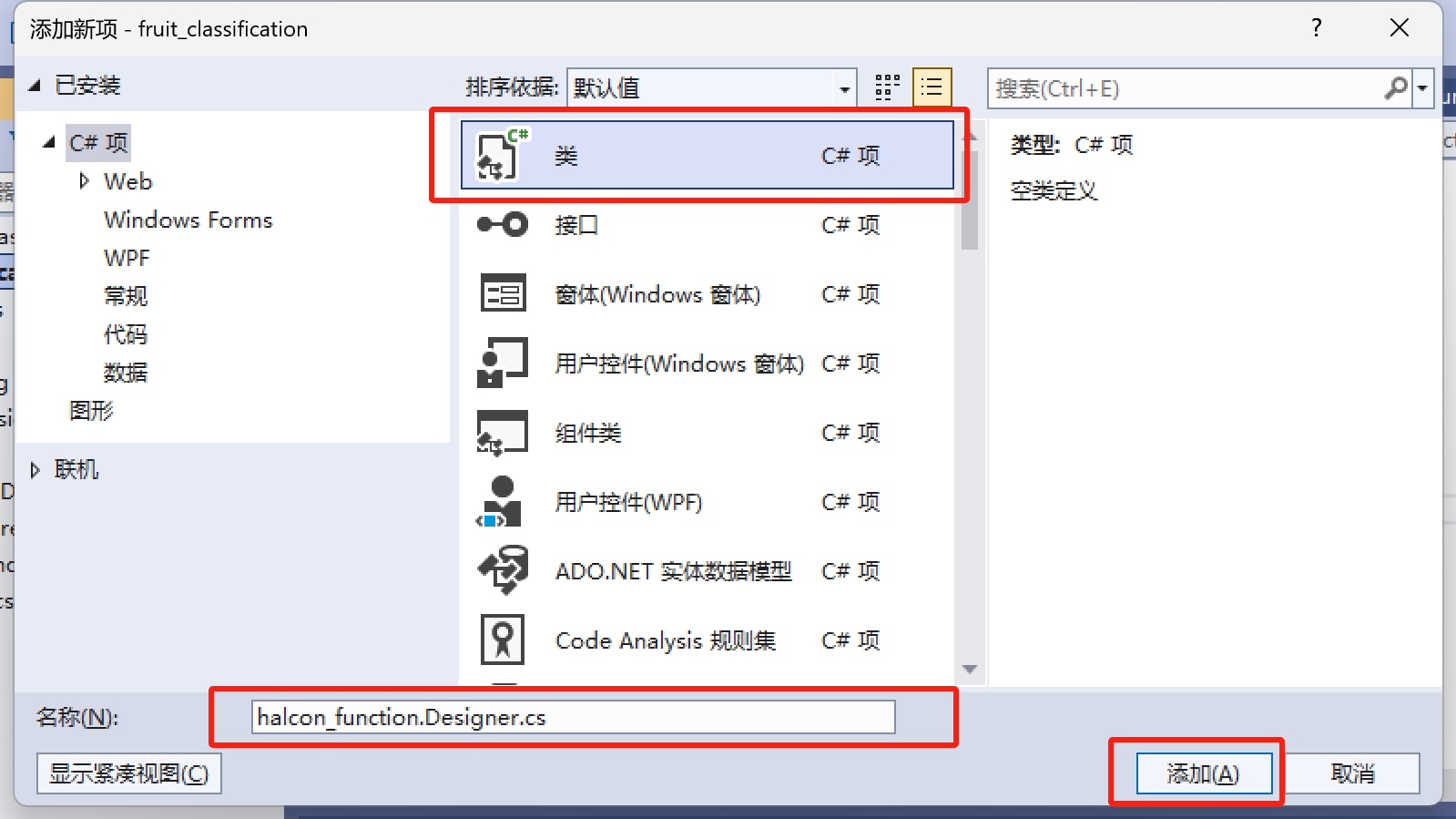
### 小技巧

这里介绍一个小技巧，在halcon代码中，经常能看到有一些淡绿色代码，这些属于外部函数或者是库函数，并不包含在halcon的算子库中，在将haclon导出为C#代码时，这些函数也会导出到文件中，可以看到这些代码非常长，如果全部放在一起会很影响编写代码，所以一般将这部分代码拆分到单独的文件中，下面演示如何处理这部分代码：

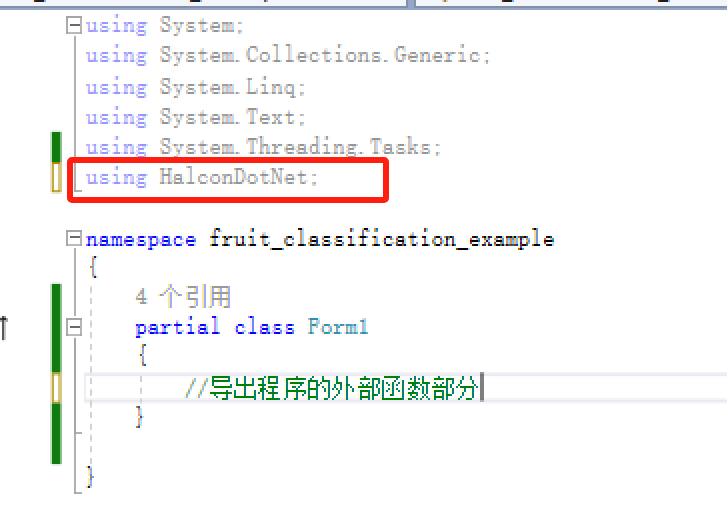
首先在项目中添加一个类，右键项目，选择【添加】→【新建项】



然后选择一个类的模板，文件名可随意，但是文件后缀要修改为“.Designer.cs”。



然后修改类名，并用partial关键字修饰，表示与主窗口类是同一个类，编译的时候合并在一起，如下图所示。然后是添加引用和将从halcon导出的外部函数部分复制到类中就可以在窗体的各种响应函数中调用这些外部函数了。



### 案例：水果分类界面封装

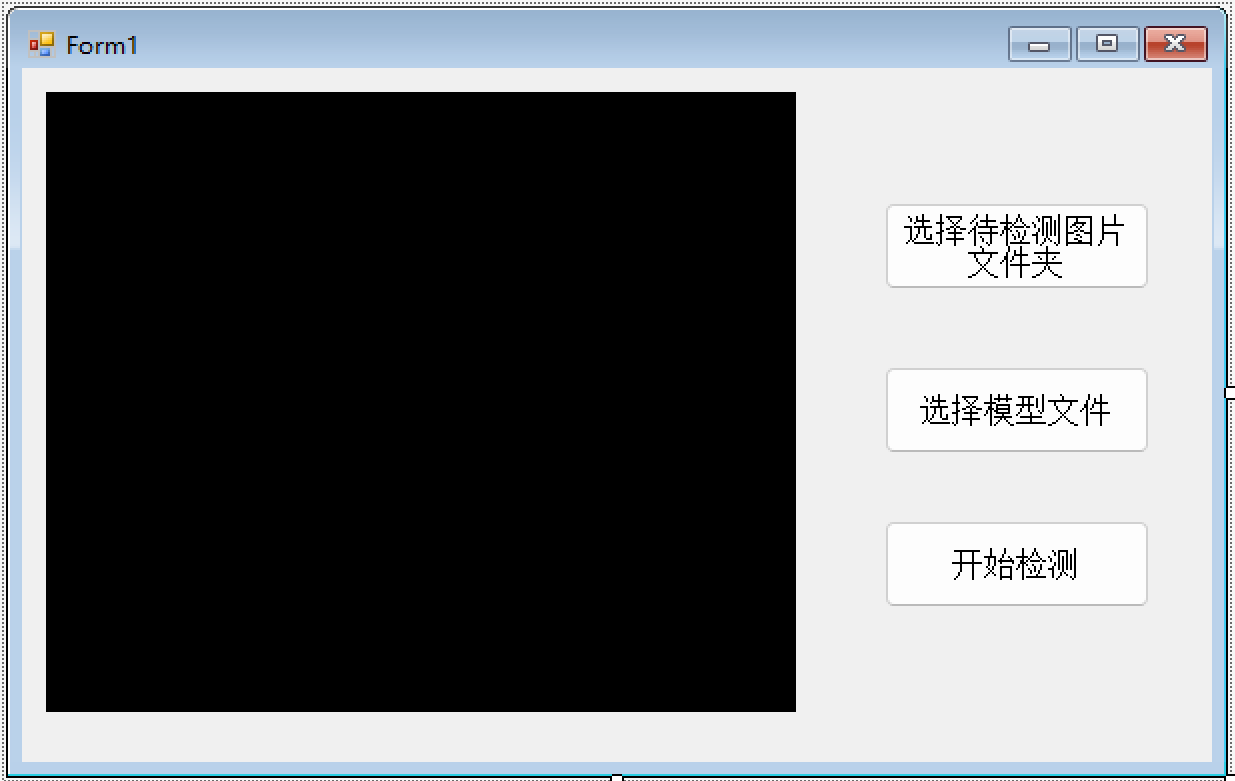
接下来通过将一个简单的例子进行界面封装来加强学习一下，就以深度学习水果分类的模型推理代码为例进行讲解。

1. 在完成在halcon中的代码编写后将代码导出为C#版本。Halcon代码如图。



1. 新建一个winform项目，配置环境，记得将首选32位取消勾选。
2. 如下图所示，完成界面布局，以及修改控件的属性，主要修改（name）属性,可参考下表。

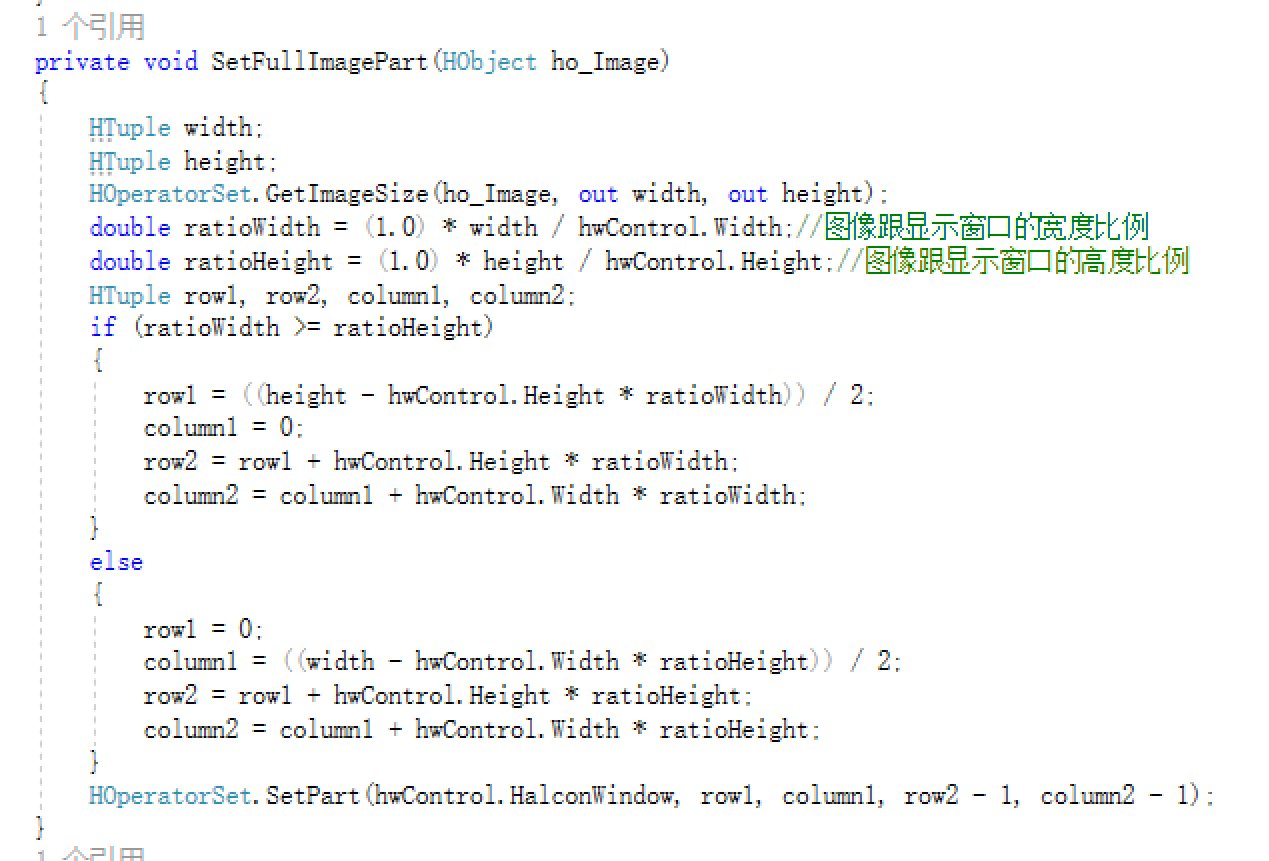
|  |  |
| --- | --- |
| 控件 | （name） |
| “选择待检测图片文件夹”按钮 | btn\_read\_img |
| “选择模型文件”按钮 | btn\_read\_model |
| “开始检测”按钮 | btn\_detection |
| HWindowControl窗口 | hwControl |



1. 按照之前介绍方法将导出程序的外部函数部分拆分到另一个文件中。
2. 将导出代码中申请变量部分代码复制到窗口类中，如图所示。

由于例子较小，不需要考虑内存问题，所以将所有的变量都当作成员变量。再定义一个int类型变量，作为文件索引。

1. 添加SetFullImagePart（）函数，这个函数内容在之前提到过，是为了让图像适应窗口显示。



1. 在窗口设计器中双击“选择待检测图片文件夹”按钮，添加一个按钮点击事件的响应函数，添加以下代码。



利用OpenFileDialog类来获取一个图片文件夹的路径。然后将获取的文件夹路径给list\_image\_files（）函数，获取文件夹里的图片路径，这个函数是halcon代码导出来的。然后是让索引i=0。

1. 双击“选择模型”按钮，添加按钮的点击事件响应函数，代码如下。



这里要实现的功能与选择待检测图片文件夹”按钮类似，获取文件路径后读入一个模型，根据这个模型创建字典，然后获取电脑设备参数并设置到模型中。

1. 双击“开始检测”按钮，添加按钮点击响应函数，添加以下代码



原来的代码是用for循环进行遍历检测的，这里就不能用了，因为halcon中的stop（）函数导出C#并不可以用，所以改成每点击一次按钮，检测一张图片，这就是为什么要定义一个索引成员变量了。

Halcon中为了方便编程，每执行一行代码都会自动显示输出的图像，但是导出来的代码可没有带这个功能，所以我们还需要手动添加显示图像的代码。还有需要把显示文本信息DispText（）前的判断语句if (HDevWindowStack.IsOpen())删掉，因为作者习惯问题，不喜欢用openwindow（）函数来创建halcon窗口，所以那段判断语句永远不为真导致DispText（）执行不到，删掉就可以了。再把DispText（）的第一个参数改为hwControl.HalconWindow。