

现代操作系统应用开发实验报告

学号： 15331248

班级： 上午班

姓名： 潘承远

实验名称： Todos(HomeWork4)

一. 参考资料

●《应用数据处理》：了解如何将数据写入设置（本地、漫游），如何从设置中读取数据（本地、漫游），根据 PPT 上面的代码以及微软官网对某些关键字的解释，使得自己熟悉了这些知识；

●《10-ApplicationLifecycle》：对生命周期加深理解；

●TA 给的 demo (End)：我仔细地研究了每个文件的代码，对 App.xaml.cs 和 Page2.xaml.cs 两个文件理解花的时间最多，App.xaml.cs 文件里面的代码让我更加了解生命周期的挂起与重启，以及如何恢复到挂起时的界面等知识。Page2.xaml.cs 文件让我了解如何在离开某界面时将设置写入设置，以及如何在调用” OnNavigatedTo” 方法回到这个界面时从设置中读取数据，Page2.xaml.cs 里面的代码对我在 MainPage.xaml.cs 写出将 IsChecked 保存在设置以及读取数据起了很大作用。

● [https://msdn.microsoft.com/en-us/library/system.windows.data.binding.updatesourcetrigger\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.data.binding.updatesourcetrigger(v=vs.90).aspx)

这个网站让我了解了 Binding.UpdateSourceTrigger Property，这是因为我对 Page2.xaml 里面的 TextBox 控件的绑定语句的一句代码——“UpdateSourceTrigger = PropertyChanged” 不够了解，于是自己在微软官网查了一下，明白这句代码就是能够让控件及时地被通知对象属性发生了变化，能让控件属性即时地在 UI 中改变并显示出来。我做了一个实验，就是将两个 TextBox 都与 Field1 绑定，这样当我在一个 TextBox 输入数据的时候，另一个 TextBox 就会实时地显示出来，这两个控件能产生这样的效果离不开“UpdateSourceTrigger=PropertyChanged” 这句代码，而这句代码的实现离不开 ViewModelBase 类使用了 INotifyPropertyChanged 接口

● <https://docs.microsoft.com/en-us/uwp/api/Windows.Storage.ApplicationDataCompositeValue>

这个网站让我了解了 ApplicationDataCompositeValue 类，加深了我对“组合设置”的理解。

● <https://social.msdn.microsoft.com/Forums/en-US/b1e9ffbd-450d-40c3-9c50-7ed354c27878/uwp-binding-radiobutton-ischecked-to-stackpanel-visibility?forum=wpdevelop>

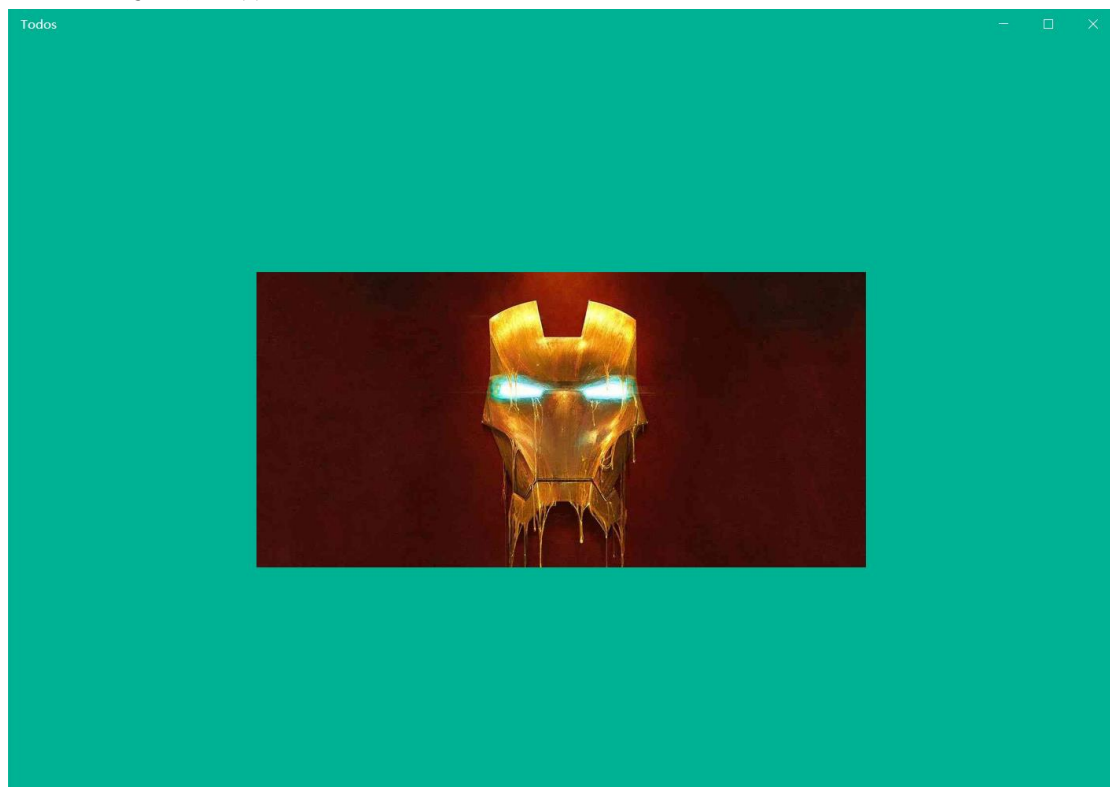
这是对我最有用的网站，它让我明白如何将 CheckBox 的 IsChecked 属性与 Line 的 Visibility 属性绑定起来，还提供了 BooleanToVisibilityConverter 转换器。

二．实验步骤

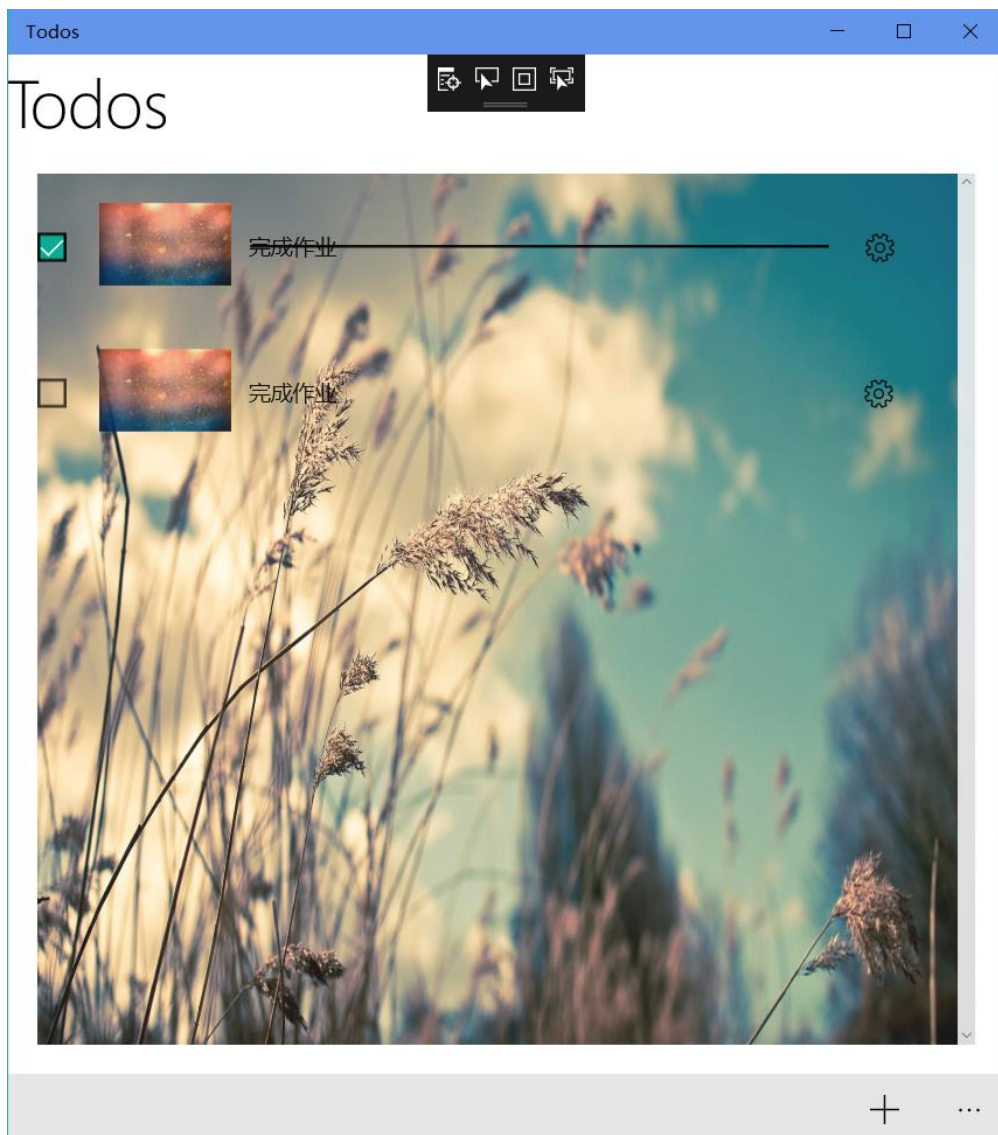
- 复习老师给的 PPT 资料，然后仔细研究 TA 给的 demo (End)，对每一个文件的每一个方法都想办法弄懂，最后综合 PPT 资料，使得自己对生命周期的理解更加深刻；
- 不清楚 Page2. xaml 中 TextBox 绑定语句中的“UpdateSourceTrigger=PropertyChanged”代码，找到了“参考资料”的第四条参考资料，学习并掌握。
- 实现“在 MainPage 中，点击 checkbox 出现横线，挂起并关闭程序，重新启动时，程序显示在 Mainpage 界面，并且点击的 checkbox 与对应横线都存在。”
- 在微软开发人员官网搜索 CheckBook 与 Line 绑定的相关信息，于是找到了“参考资料”中的最后一个网站。
- 实现“(NewPage. xaml. cs) 在 NewPage 中，输入数据，挂起并关闭程序，重新启动时，程序显示在 NewPage 页面，并且挂起前输入的数据还在。”
- 优化 UI 界面

三．实验结果截图

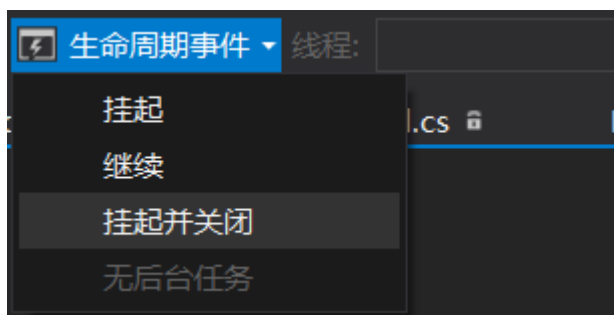
1. 程序的初始屏幕



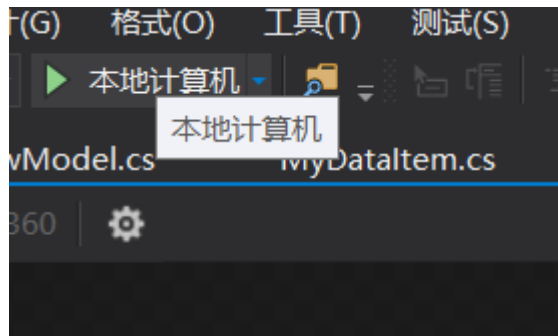
2. 进入 MainPage，并勾选第一项待办事项



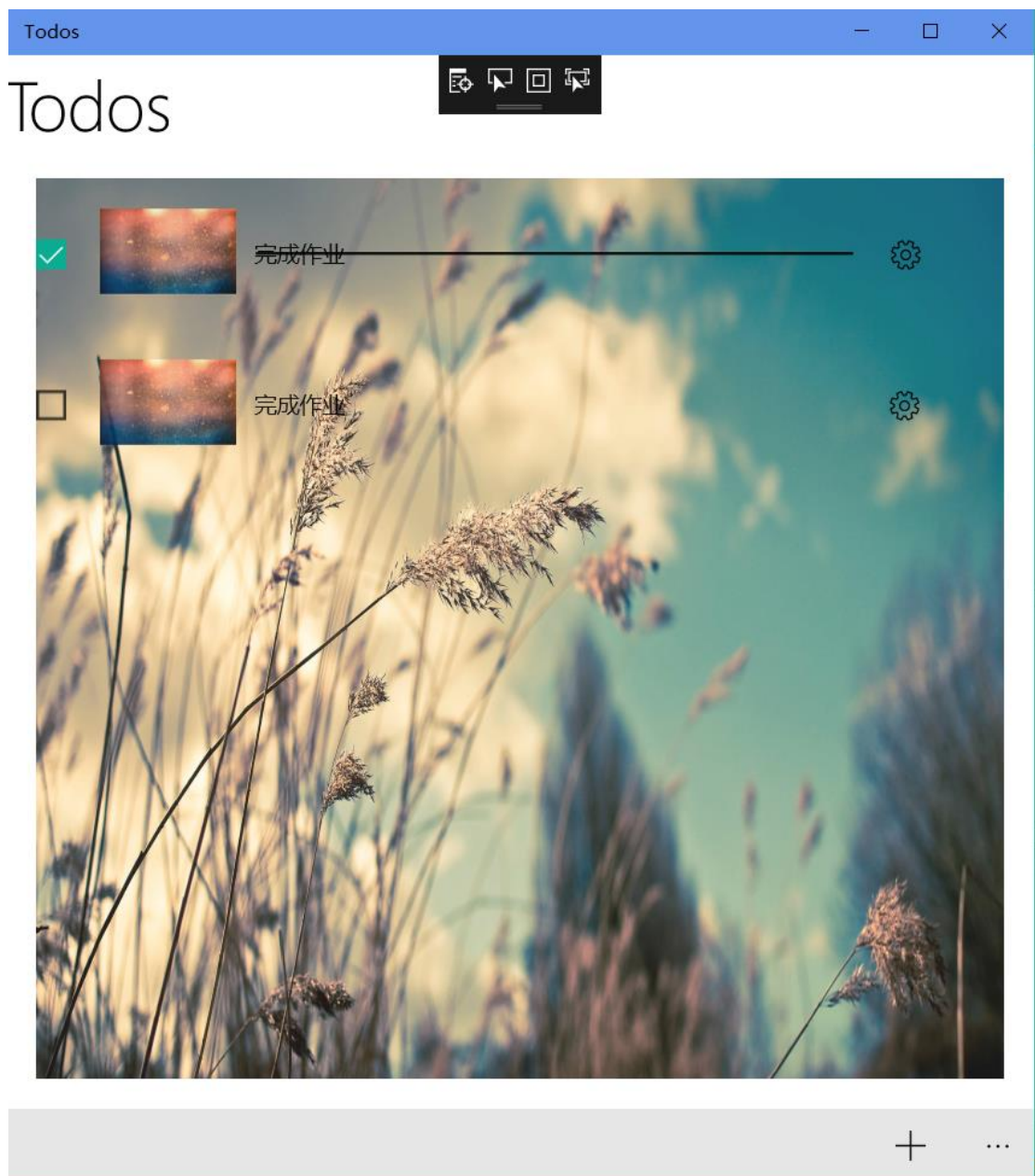
3. 点击“生命周期事件”的“挂起并关闭”



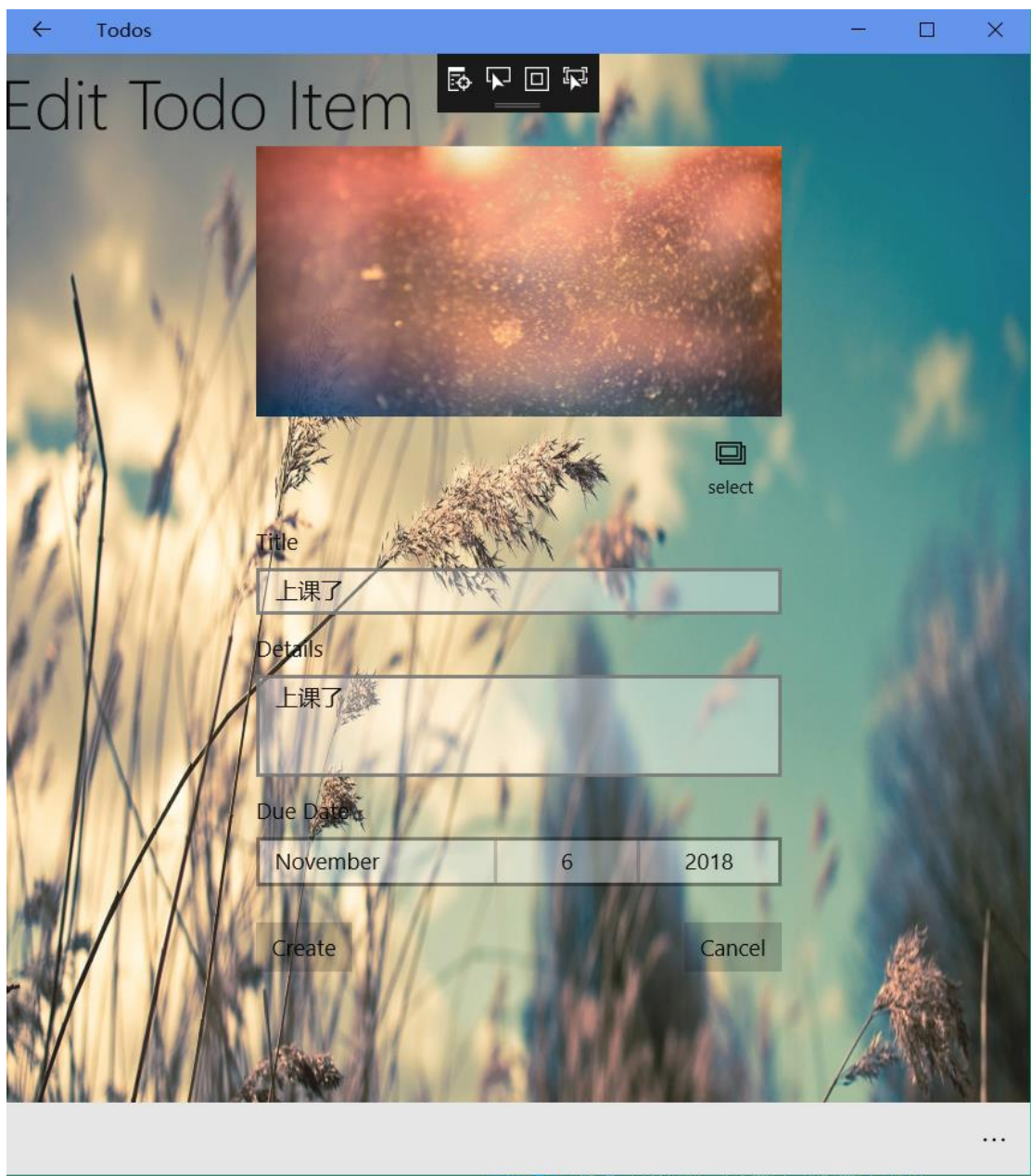
4. 重新启动



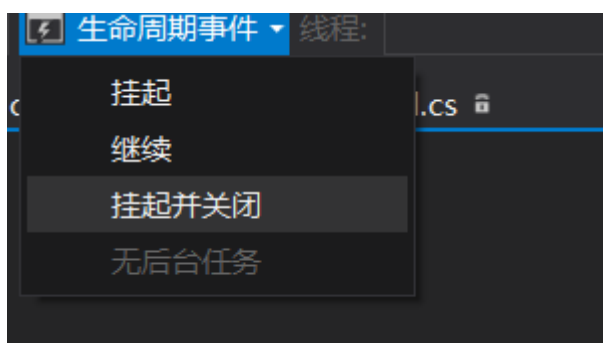
5. 可以看到程序显示在 Mainpage 界面，并且点击的 Checkbox 与对应横线都存在



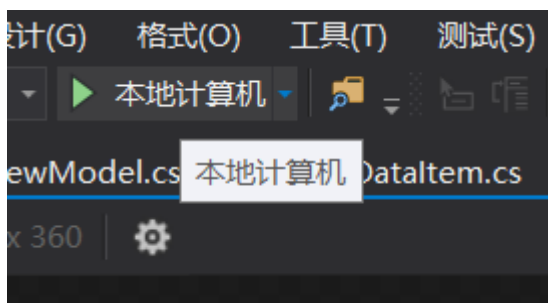
6. 进入 NewPage 界面，输入 Title、Details，选择日期。如下：



7. 点击“生命周期事件”的“挂起并关闭”



8. 重新启动



9. 程序重新启动时，程序显示在 NewPage 页面，并且挂起前输入的数据还在。



四 . 实验过程遇到的问题

- ```
Text="{Binding Field1, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />
```

对上面这条代码的“`UpdateSourceTrigger=PropertyChanged`”语句不够了解，于是自己在微软开发人员官网查了一下，明白这句代码就是能够让控件及时地被通知对象属性发生了变化，能让控件属性即时地在 UI 中改变并显示出来。我做了一个实验，就是将两个 TextBox 都与 Field1 绑定，这样当我在一个 TextBox 输入数据的时候，另一个 TextBox 就会实时地显示出来，这两个控件能产生这样的效果离不开“`UpdateSourceTrigger=PropertyChanged`”这句代码，而这句代码的实现离不开 ViewModelBase 类使用了 `INotifyPropertyChanged` 接口。

- ```
" IsChecked="{Binding Completed1, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>
Visibility="{Binding ElementName=CheckBox2, Path=IsChecked, Converter={StaticResource BooleanToVisibilityConverter}}"/>
```

上面这两条代码解决了困扰我很久的问题，这是我通过“参考资料的”最后一个网站学习到的知识。Line 的 Visibility 属性直接与 CheckBox 的 IsChecked 属性绑定在一起，这样只需要 CheckBox 与类方法绑定在一起就可以实现这周甚至上周的实验要求。上周我采用的是将 CheckBox 的 IsChecked 和 Line 的 Opacity 都绑定到 TodosItem 的“completed”属性，通过页面“刷新”将 Line 在 UI 中显示或隐藏。但是上面的代码启发我能够有更加优秀的做法-----可以将将 CheckBox 的 IsChecked 属性与 Line 的 Visibility 属性绑定起来，CheckBox 绑定“completed1”，同时使用 `INotifyPropertyChanged` 接口（在 CheckBox 绑定语句中添加“`UpdateSourceTrigger=PropertyChanged`”），这样就能更好地实现上周的要求，而不用刷新界面。

- 当我“挂起并关闭”后重启程序，如果是在 NewPage 上就不会显示返回键，所以我添加了如下代码，成功解决问题

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    SystemNavigationManager.GetForCurrentView().AppViewBackButtonVisibility = AppViewBackButtonVisibility.Visible;
```

五 . 思考与总结

我觉得我这周收获最大的是更加深入了解数据绑定，上周作业要求我们勾选 CheckBox 后 Line 就会显示出来，当时我的做法是将 CheckBox 的 IsChecked 和 Line 的 Opacity 都绑定到 TodosItem 的“completed”属性，通过页面“刷新”将 Line 在 UI 中显示或隐藏。但是通过这周的学习，让我明白可以将 CheckBox 的 IsChecked 属性与 Line 的 Visibility 属性绑定起来，CheckBox 绑定“completed”，同时使用 `INotifyPropertyChanged` 接口（需要在 CheckBox 绑定语句中添加“`UpdateSourceTrigger=PropertyChanged`”），这样就能更好地实现上周的要求，而不用刷新界面。这周作业也加深了我对 `INotifyPropertyChanged` 接口的理解，因为上周我没有用 `INotifyPropertyChanged` 接口，采用得方法是点击 CheckBox 后回到当前界面（“刷新”），用这样不是很正规的方法来实现题目要求。通过这周的学习，我可以将上周的代码修改得更加完善。

此外个人觉得相比上一周，这一周的作业会简单一些，重点在于如何将数据写入设置

以及如何从设置中读取数据，通过分析老师给的 PPT 就能够很好地理解。