**Summary**

这页不是实验报告，实验报告从下一页开始。

由于开始的时候就实现了动态绑定，实验二和实验三的提交内容均是本份报告和同样的代码

实现的内容

1. 通过Grid、StackPanel实现可伸缩的布局。
2. 通过Entity Framework + LINQ实现对Microsoft SQL Server 2016 LocalDB的数据查询。
3. 将数据库中获得的数据动态绑定到界面上。（所有内容，包括TreeView均为动态生成）



.NET技术实验报告



实验三 WPF动态生成

学生姓名 杨振华

学生学号 201626811225

学生班级 软件工程 1603

任课教师 简琤峰

提交日期 2019年5月2日

**计算机科学与技术学院**

**目录**

[一. 引言 4](#_Toc7729957)

[1.1 实验要求 4](#_Toc7729958)

[1.2 具体要求 4](#_Toc7729959)

[二. 程序设计说明 5](#_Toc7729960)

[2.1 程序实现 5](#_Toc7729961)

[2.2 测试中的问题与解决实现 8](#_Toc7729962)

[2.3 测试样例运行结果 8](#_Toc7729963)

[2.4 尚未解决的问题 9](#_Toc7729964)

[三． 实验体会 9](#_Toc7729965)

[四． 源程序 9](#_Toc7729966)

# 引言

## 实验要求

设计WPF界面，实现窗口

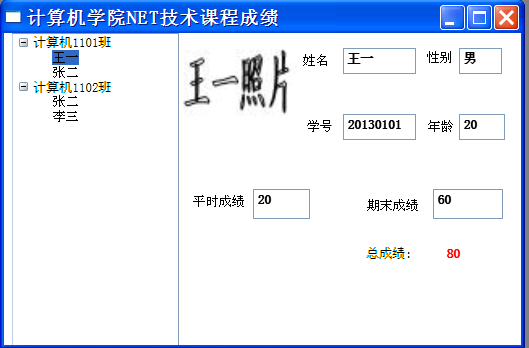
## 具体要求

1. 如图1所示，设计WPF窗体界面，其中左边是一个treeview控件，右边为label和text以及image控件。左边treeview控件显示表1中的班级和姓名二层数据。当鼠标击左侧的姓名如“王一”时候，右边显示表1中对于“王一”的行数据，并显示到相应的控件中，其中总成绩的80为红色显示的标签label（包括具体的照片显示）。
2. 代码实现对表1数据的动态读取显示，例如当鼠标击左侧的姓名如“A”时候，右边显示表1中对于“A”的行数据，并显示到相应的控件中，其中总成绩=平时成绩+期末成绩。

表1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 班级 | 学号 | 姓名 | 照片 | 年龄 | 平时成绩 | 期末成绩 | 总成绩 |
| 计算机1101班 | 20130101 | 王一 | 王一照片.jpg | 20 | 20 | 60 | 80 |
| 计算机1101班 | 20130102 | 张二 | 张二照片1.jpg | 21 | 15 | 50 | 65 |
| 计算机1102班 | 20130201 | 张二 | 张二照片2.jpg | 19 | 25 | 70 | 95 |
| 计算机1102班 | 20130202 | 李三 | 李三照片.jpg | 20 | 25 | 65 | 90 |

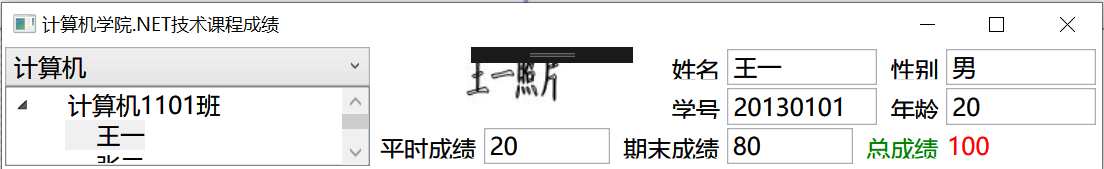
图1

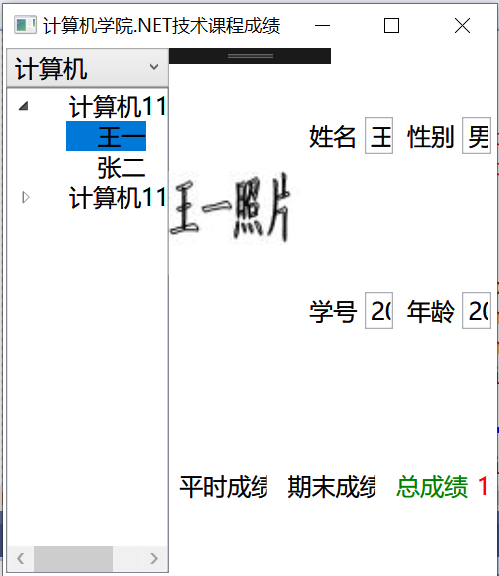


# 程序设计说明

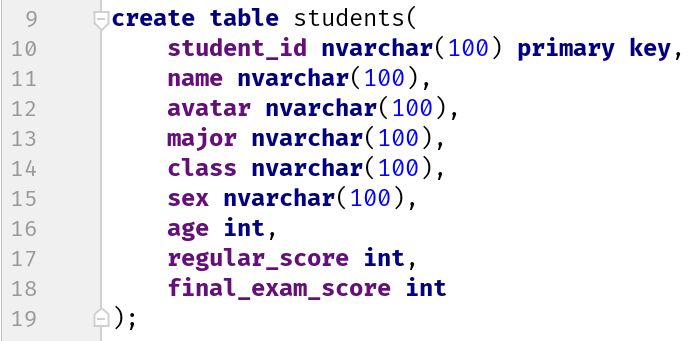
## 2.1 程序实现

首先实现界面，为了程序可以适应窗口的拉伸、收缩，这里采用了Grid、DockPanel等布局器，具体实现效果如下图。

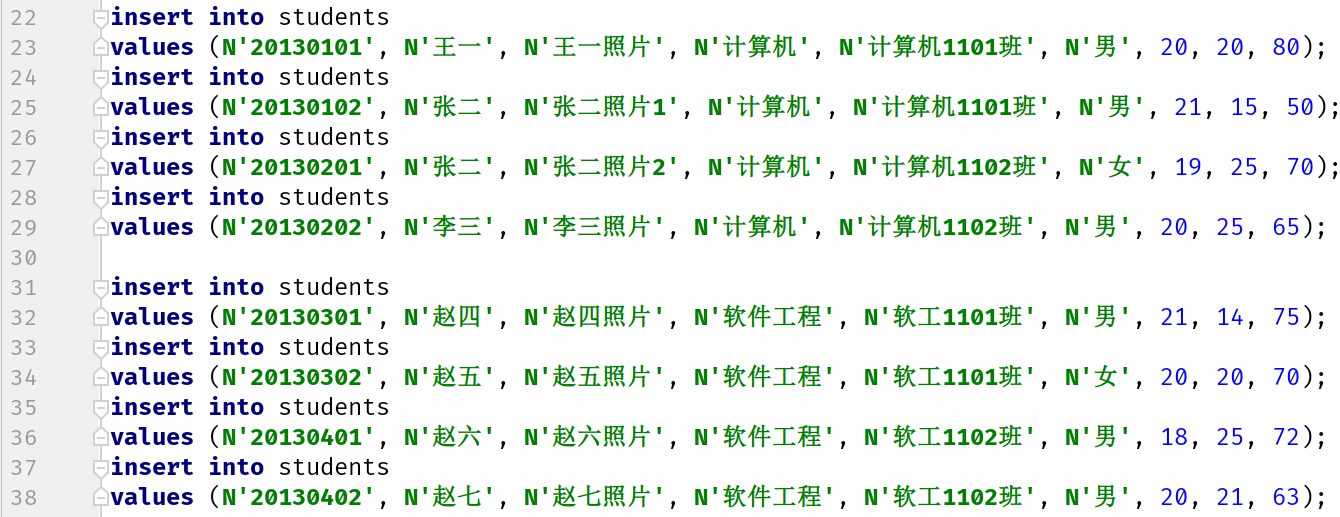




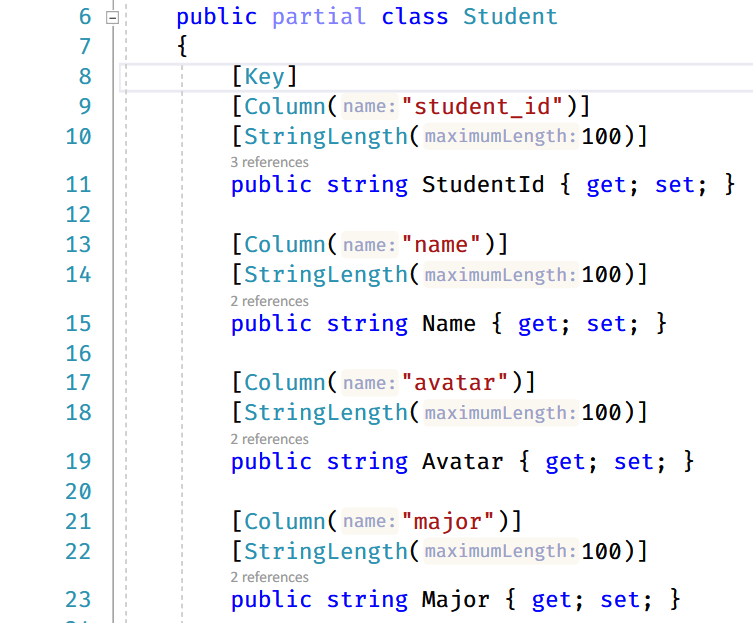
接下来实现数据库查询，为了简化编程，这里使用了SQL Server 2016 LocalDB以及Entity Framework ORM框架。首先建立数据库表结构。



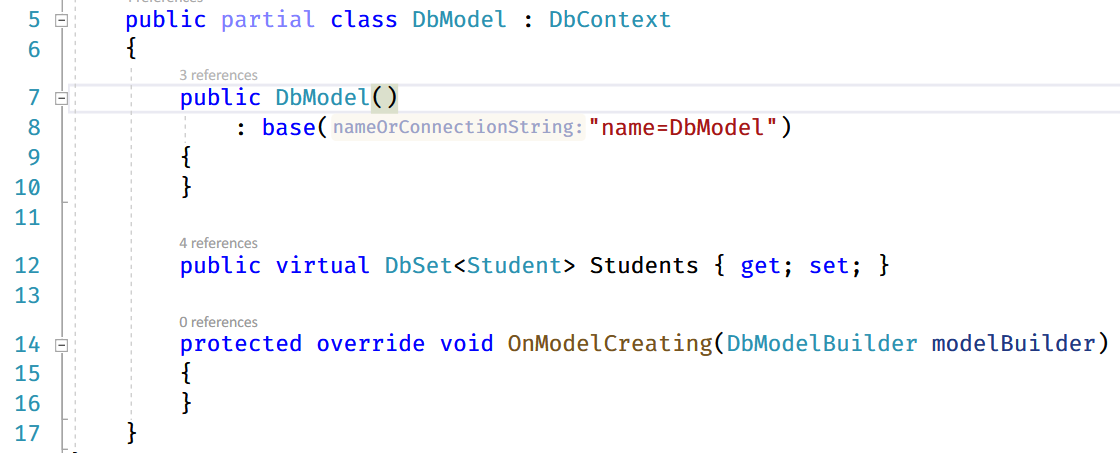
然后插入数据。



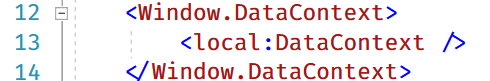
接着建立ORM的对应这个表的Model，下图中给出Model类局部。



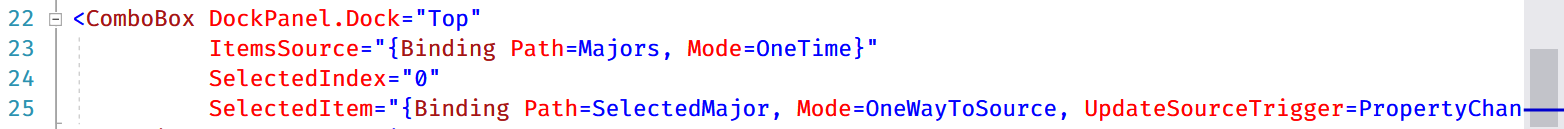
然后编写数据库类。



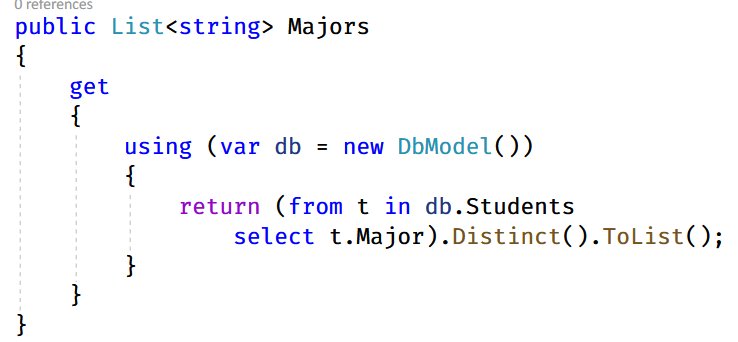
接下来就可以实现数据绑定了，首先在XAML中设置DataContext。



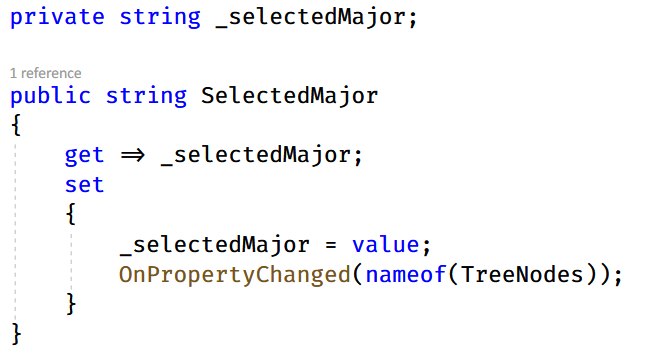
这里首先以选择专业的ComboBox为例。



它绑定了Majors和SelectedMajor两个属性，我们来看一下Major的实现。



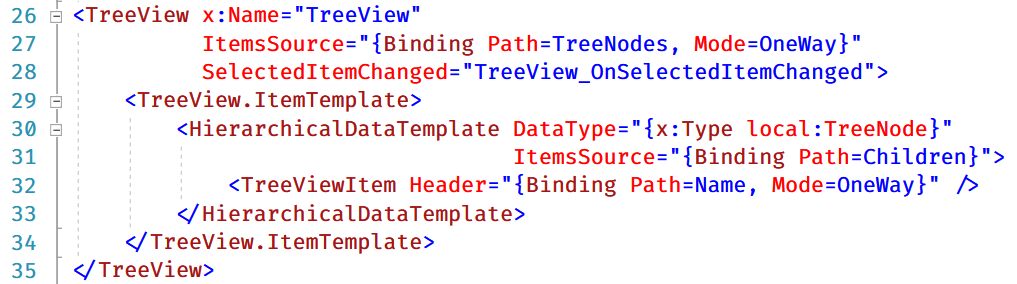
Majors通过LINQ查询数据库获得专业。一旦用户选择专业，View便会设置SelectedMajor的值。



之后SelectedMajor通过OnPropertyChanged通知ViewModel更新TreeView的数据。



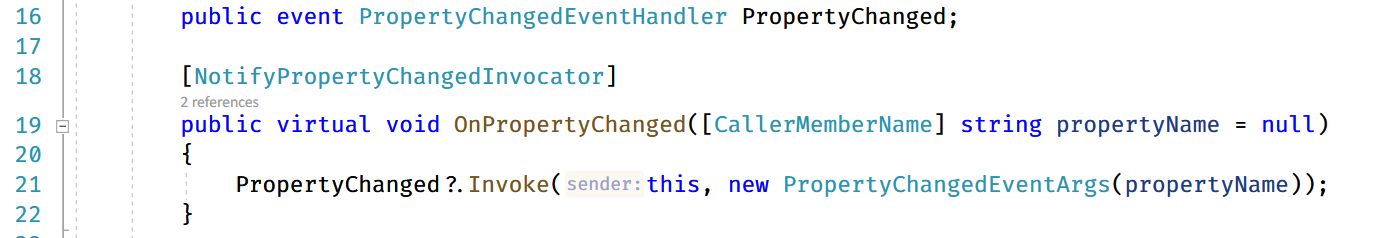
TreeNodes绑定到TreeView，通过更加复杂的一段LINQ实现数据的查询和绑定，这里再看一下TreeView处的XAML代码。



同样，更新主界面信息的过程类似，这里不再赘述。

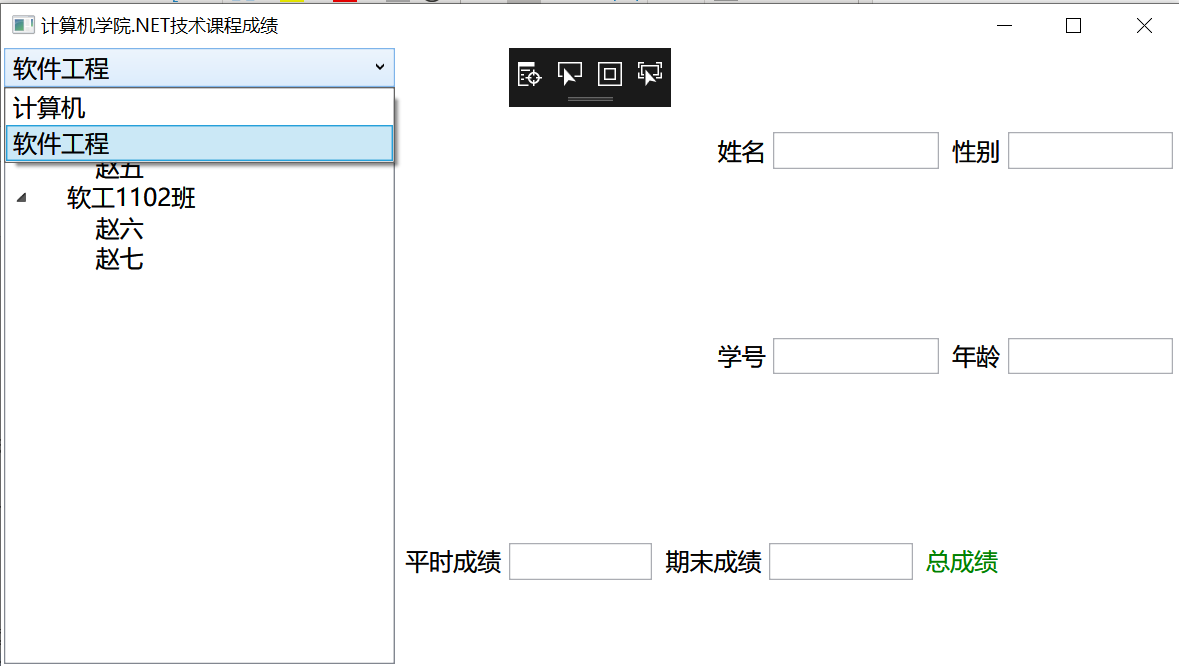
## 2.2 测试中的问题与解决实现

1. 双向绑定更新选择学生的ViewModel后未动态更新界面上的学生信息。主要原因是未调用如下图所示的OnPropertyChanged，传入需要更新的属性名。解决方案是合理调用OnPropertyChanged。



## 2.3 测试样例运行结果

1. 根据专业ComboBox在TreeView中动态加载班级和学生信息。



1. 根据在TreeView中的选择动态在主界面中加载学生信息。



## 2.4 尚未解决的问题

1. 无法更新数据库中

# 实验体会

通过本次实验，我熟悉了通过StackPanel、Grid等布局器和TextBlock、TreeView等控件进行布局、熟悉了通过Entity Framework、LINQ进行数据库查询、学习了WPF的动态绑定和MVVM设计模式等内容。这加深了我对C#和.NET技术的理解和运用。

# 源程序

详见附件源码。