TypeConverter

代码路径 `.net笔记/Code/TypeConverterDemo`

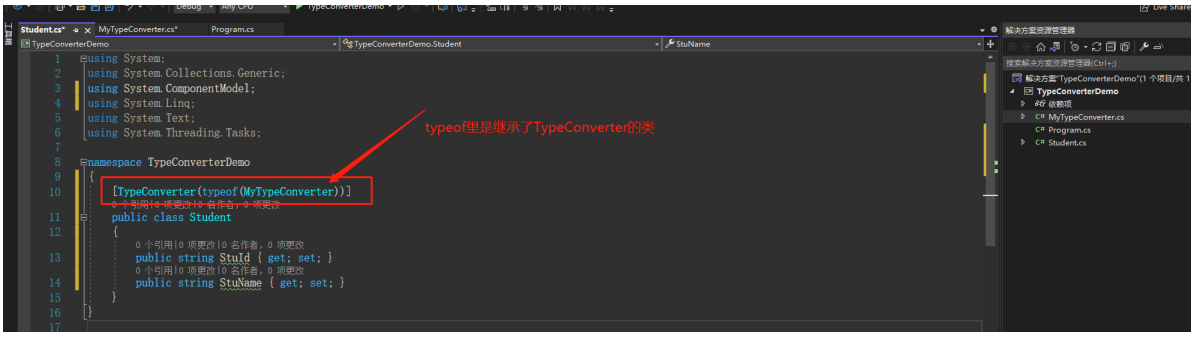
自定义一个类MyTypeConverter继承System.ComponentModel.TypeConverter并重写里面的类(需要用到哪个就重写哪个)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TypeConverterDemo
{
    public class MyTypeConverter:TypeConverter
    {
        public override bool CanConvertFrom(ITypeDescriptorContext? context,
Type sourceType)
        {
            return base.CanConvertFrom(context, sourceType);
        }

        public override bool CanConvertTo(ITypeDescriptorContext? context, Type?
destinationType)
        {
            return base.CanConvertTo(context, destinationType);
        }

        public override object? ConvertFrom(ITypeDescriptorContext? context,
CultureInfo? culture, object value)
        {
            return base.ConvertFrom(context, culture, value);
        }

        public override object? ConvertTo(ITypeDescriptorContext? context,
CultureInfo? culture, object? value, Type destinationType)
        {
            return base.ConvertTo(context, culture, value, destinationType);
        }
    }
}
```

自定义一个Student类,类名上标记TypeConverterAttribute特性

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

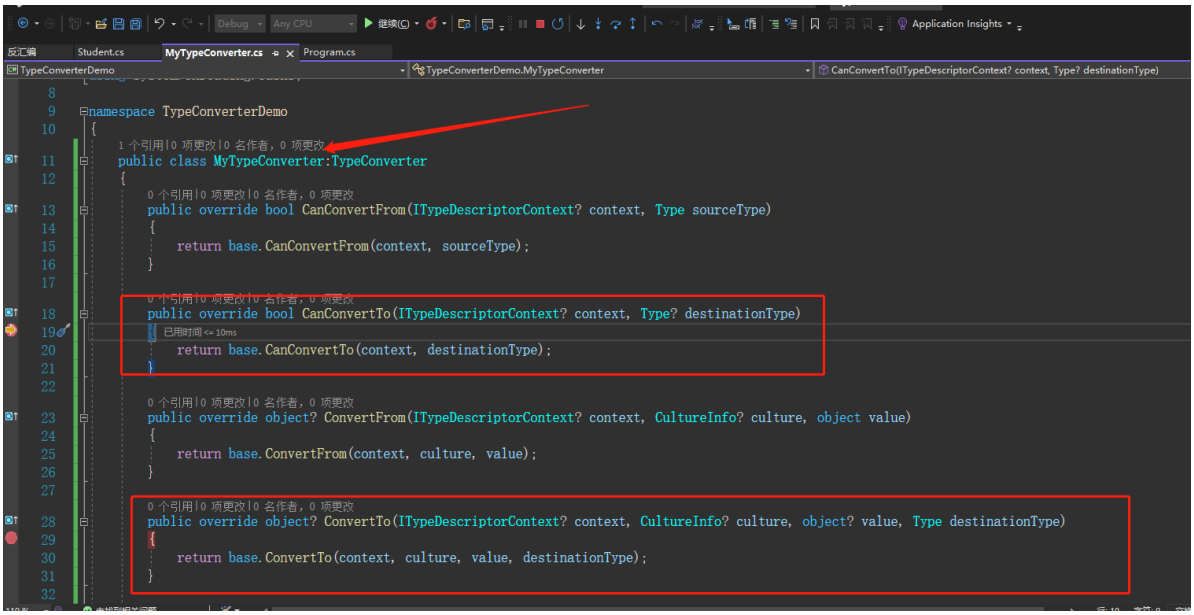
```
namespace TypeConverterDemo
{
    [TypeConverter(typeof(MyTypeConverter))]
    public class Student
    {
        public string StuId { get; set; }
        public string StuName { get; set; }
    }
}
```



使用方式

```
using System.ComponentModel;
using TypeConverterDemo;

Student student = new Student();
student.StuId = "125";
student.StuName = "he_zhw";
TypeConverter typeConverter = TypeDescriptor.GetConverter(typeof(Student));
bool v1 = typeConverter.CanConvertTo(typeof(string));//判断是否能将Student类型转换成string类型，如果CanConvertTo参数的为string类型的，则默认返回True
object? v = typeConverter.ConvertTo(student, typeof(string));//将Student类型转换成对应类型，如果子类未重写ConvertTo则默认转换成类型ToString后的字符串
```

类型上标记了`TypeConverterAttribute`调用了CanConvertTo与ConvertTo等以上方法后，会调用MyTypeConverter里重载后的方法



需特别注意的方法

- ## CanConvertFrom(Type)

返回该转换器是否可以将给定类型的对象转换为此转换器的类型

该方法默认返回false，需自己实现

- # TypeConverter.IsValid

返回给定的值对象是否对此类型和对指定的上下文有效。

从 .NET Framework 4 开始， IsValid 方法从和方法中捕获 CanConvertFrom 异常 ConvertFrom 。 如果输入值类型导致 CanConvertFrom 返回 `false` ，或者输入值导致 ConvertFrom 引发异常，则该 IsValid 方法将返回 `false`

由此可得结论:因CanConvertFrom默认返回false,所以在CanConvertFrom与ConvertFrom未被重写的情况下，IsValid默认也是返回false