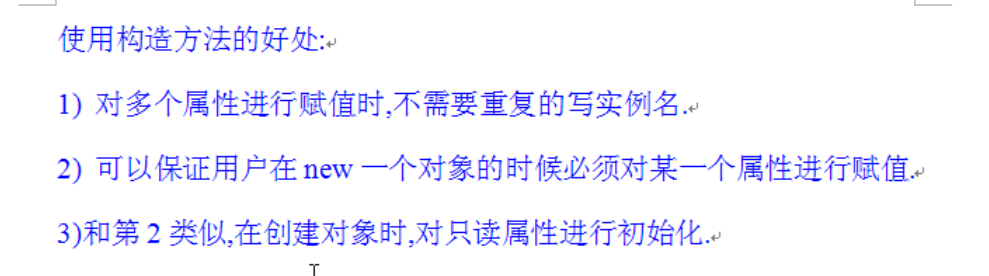
**我们在什么情况下需要使用构造方法？**

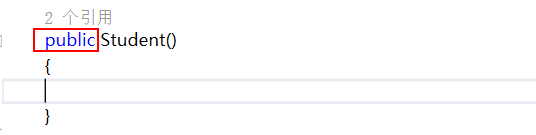
**如图：**

****

**关于第三点的注释：例如我们想给一个名字赋值的时候，但是名字为了安全考虑写成了一个只读的属性，不能够写入，这时候我们就能利用构造方法传参的方式对只读的属性进行赋值**

**构造方法的使用方法**

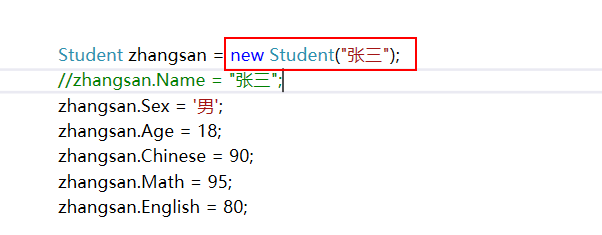
**如图：**

****

**注意红色框，如果不加访问修饰符的话系统默认为 private私有的，那么在其他类中就不能够被访问到  
并且没有返回值 void可以省略不写，并且构造方法名与类名相同**

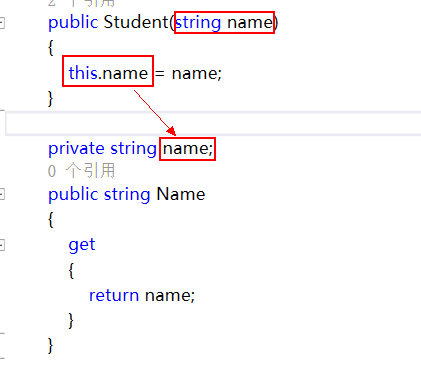
**构造方法的具体使用方法和详解：**

**图1：**

****

**（这里是Program类中的截图，注意，这里我们已经定义好了构造方法，并且开始使用了，通过传参的方式把“张三”的值传给Student类中的构造方法）**

**图2：**

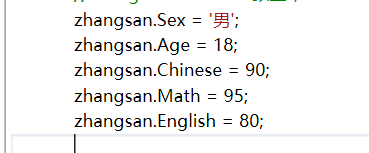
****

**(this指的是 这个类中的字段或者属性 所以这里的this.name 是指Student这个类中，private string name 的这个字段)**

**（这个是Student类中的截图，如图1所示，我们把“张三”传参过来，并且通过了string name来接受，需要注意的是，这里的this.name值的name不是方法中的string name，而是箭头所指的private string name）**

1. **通过构造方法传参，我们就避免了，像名字 身份证号 性别等等不可改变的值写成只读属性的时候不可被赋值的情况**
2. **当需要重复重复赋值多个属性或者字段的时候**

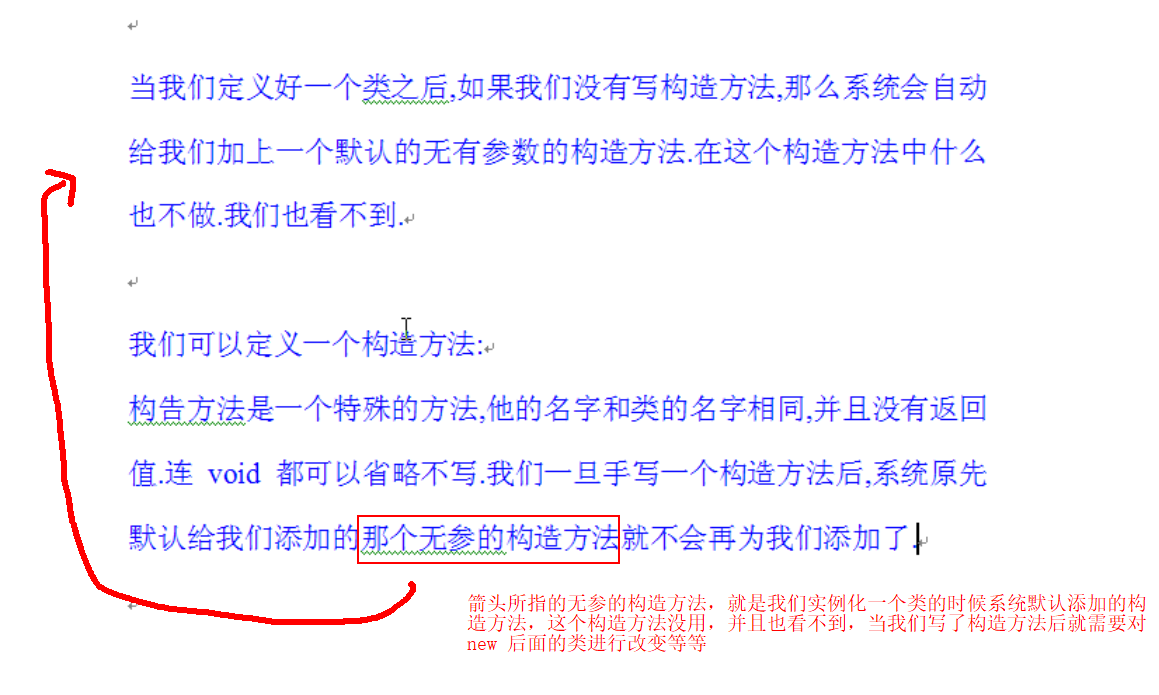
**如图：**

****

**这样我们通过传多个参数也可以避免这种问题**

**3）通过构造方法我们也可以限定用户在传参的时候一定要先赋值，因为你写了构造方法后如果没有进行传参的步骤是连实例化都进入不了**

**关于构造方法的文字详解：**

****

**注意：当我们没有写构造方法的时候 系统也有构造方法的**

**例如：**

**微信截图_20171205122930**

**这里的Student()其是也是构造方法，但是这个构造方法没有参数，也没有用**

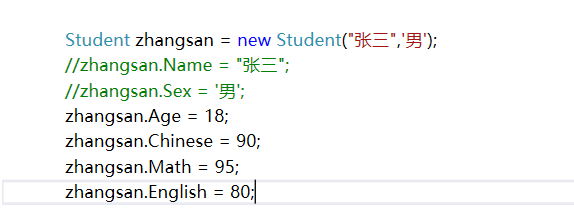
**当我们在Studen类中定义了一个构造方法并且写了参数后 才有用**

**我们还可以构造方法的重载**

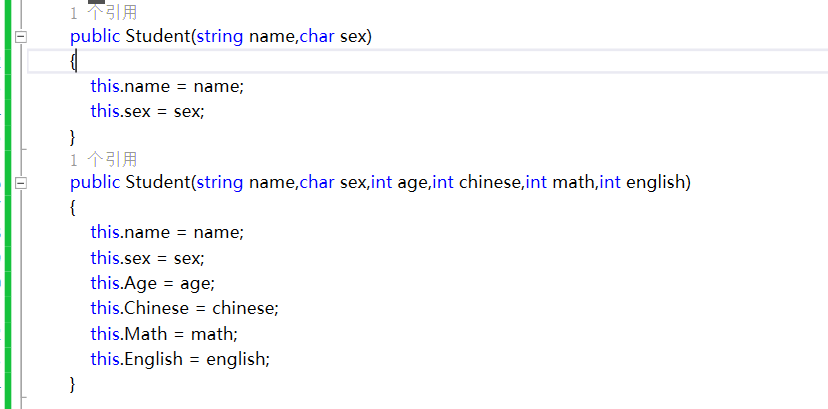
**方法与面向过程中写方法的重载类似**

**=>=>注意事项**

**那么在面向对象的类中创建构造方法具体方法如图所示：**

****

**（这里是Program类中的代码）**

****

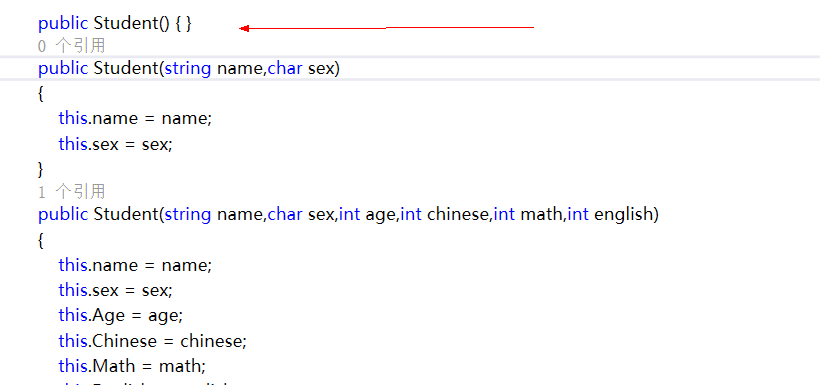
**（这是Student类中的代码，如图所示，我们就成功的定义了个构造方法的重载！）**

**当我们写了构造方法后系统会默认把原先无参的构造方法取消，而自定你所写的构造方法，这时候如果还需要使用一个无参的构造方法是不行的**

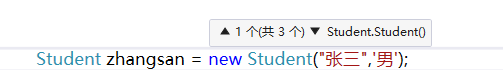
**如图：**

**微信截图_20171205152056（因为我们已经在Student类中定义了一个构造方法，所以这时候就不行）**

**除非 自己写一个无参的构造方法:**

****

**注意箭头，其实这个无参的构造方法与下面的两个构造方法都是组成了一个构造方法的重载**

****