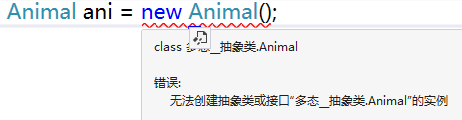
1. **为什么使用抽象类？**

当我们不知道父类的方法如何去实现的时候，并且关于里式转换第一条父类想调用子类中的方法的时候就可以使用抽象类

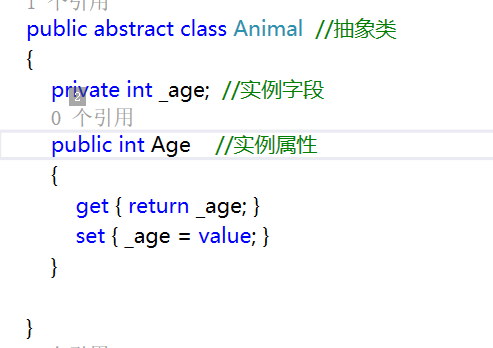
1. **使用抽象类条件：**

**1.** 必须具有继承关系，并且子类赋值了给父类；

**2.** 抽象类不能为自己的类创建对象；



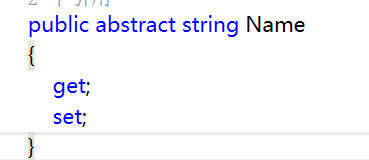
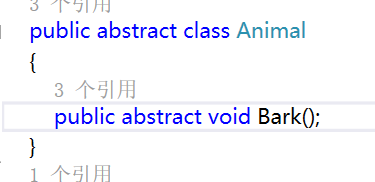
**3.** 抽象类能够包含实例成员（非抽象成员）；



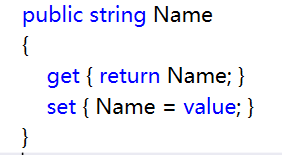
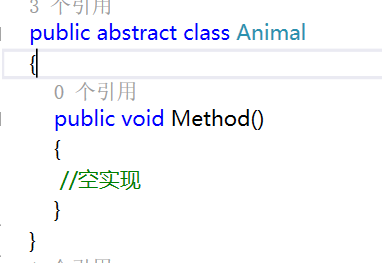
**4.** 当子类继承于抽象类父类的时，抽象类中的**所有抽象成员**都必须被子类重写！**除非子类也是个抽象类就不需要重写！**但是**实例字段、属性、方法可以不用重写！**



**5.** 抽象类中的抽象成员必须都标记为：**abstract**，**并且都必须无实现！**

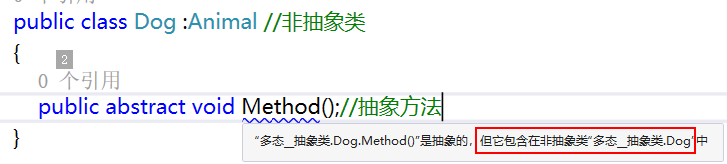


（这里的抽象方法和属性就没有方法体了，也叫做没实现）



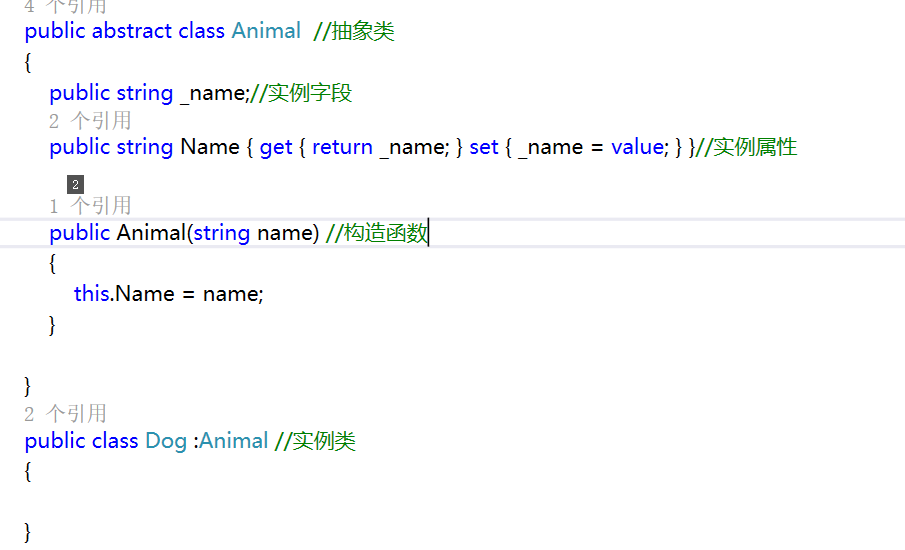
（这里的字段和属性都是有实现的）

**6.** 抽象成员只能允许存在于抽象类当中；

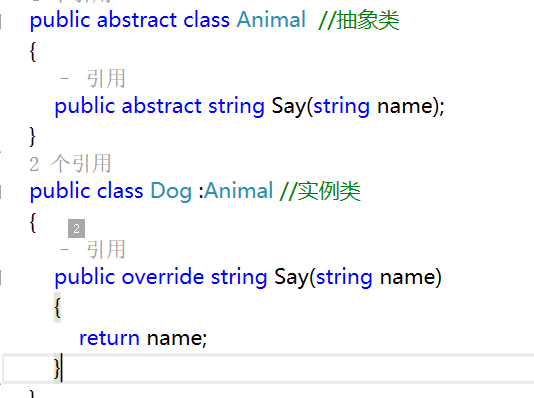


**7.** 抽象成员的访问修饰符不能是private；

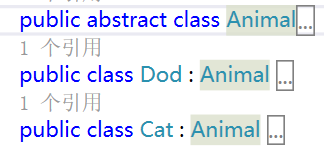
**8.** 抽象类中能够定义一个构造函数，并且正常通过子类来调用



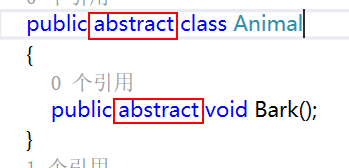
**9.** 如果抽象类父类中的抽象方法存在**参数和返回值**，那么子类重写该方法的时候必须要**有参数和返回值**

****

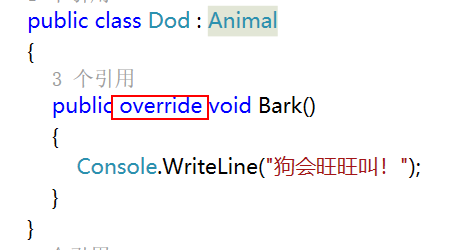
1. **抽象类的使用：**
2. 三个类需存在继承关系；



1. 通过关键字：abstract把父类定义成抽象类，把父类的方法定义成抽象方法，并且父类方法无实现；



1. 通过关键字：override 把子类的方法定义成重写方法



**如果父类中的方法有默认的实现，并且父类需要被实例化，这时可以考虑将父类定义成一个普通类，用虚方法来实现多态。**

**如果父类中的方法没有默认实现，父类也不需要被实例化，则可以将该类定义为抽象类。**