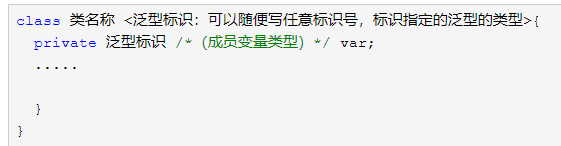
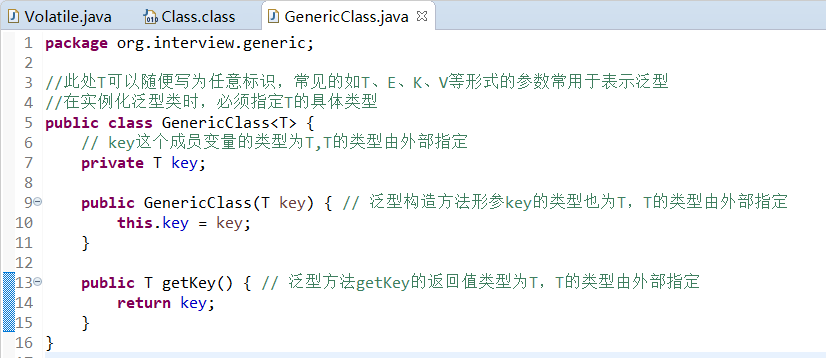
#### 泛型类

泛型类的最基本写法

泛型类主要是用来做容器



对应的示例

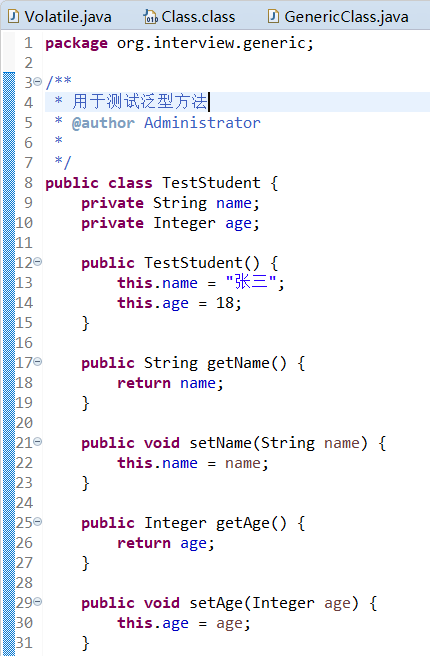


#### 泛型方法

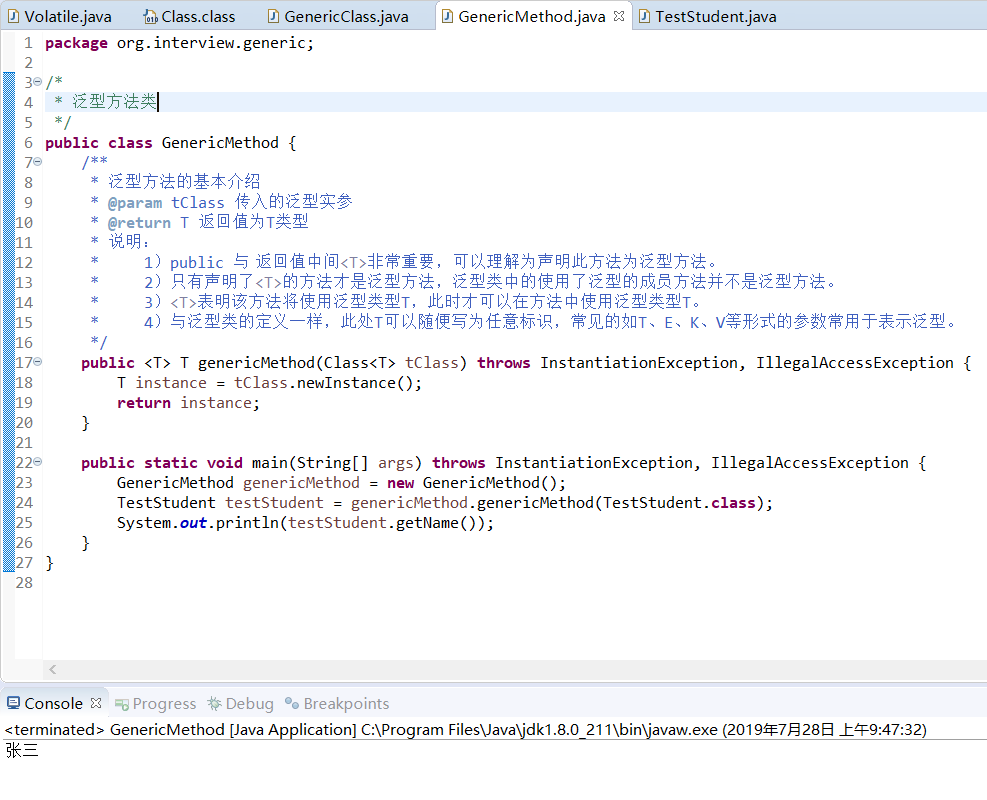
在正常的类里面可以指定一个带有泛型的方法

泛型方法主要是配合反射来做一些解耦的方法调用，如果不用反射的话，泛型方法能够做到的事情太少了，只能调用object的几个方法，失去了灵活性。

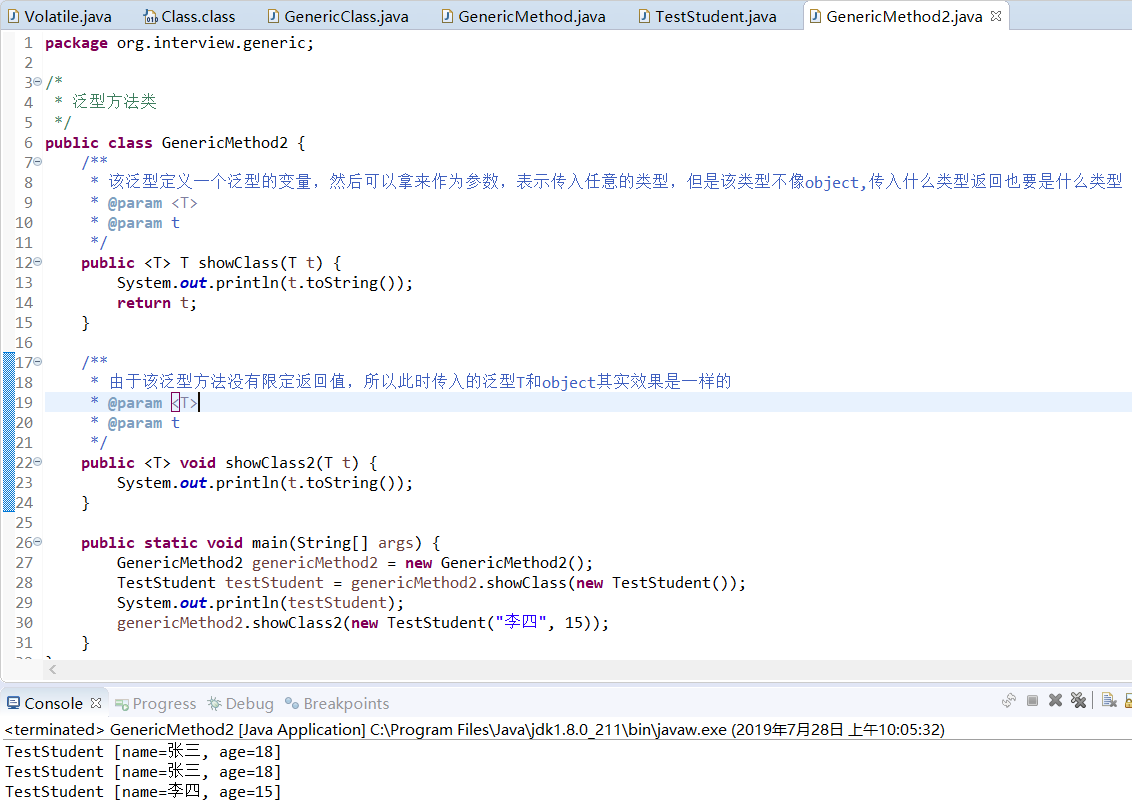
泛型方法的灵活性举例：以动物呼吸为例，牛呼吸空气，猪呼吸空气，鱼呼吸水。那么泛型可以实现，只需要指定呼吸的方法即可，也可以用实现接口的方式去做，但是泛型+反射的话更加的灵活通用



下图17行第一个<T>标识一个泛型，第一个T标识返回值。



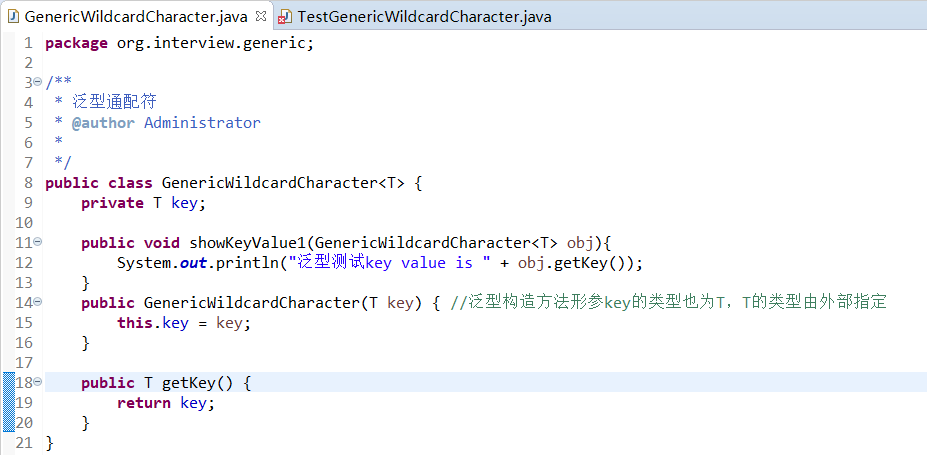
泛型方法的示例2



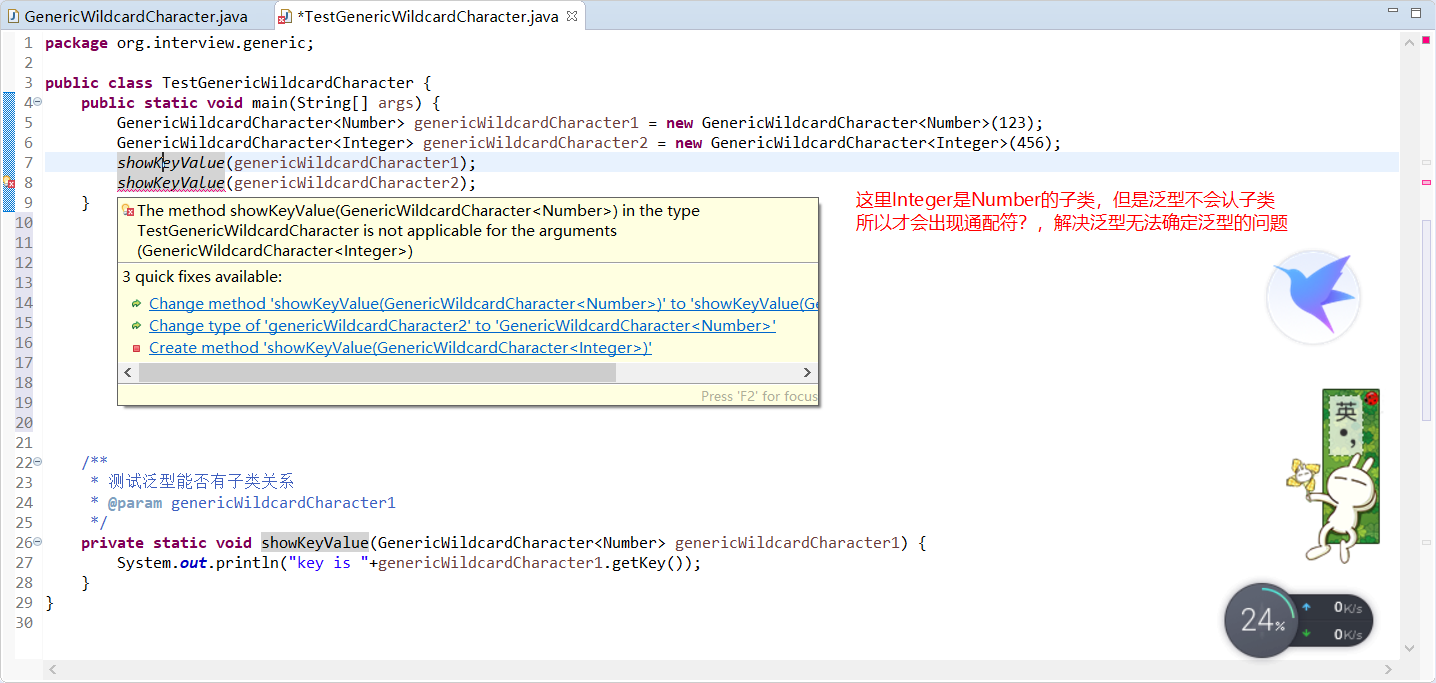
#### 泛型通配符和泛型上下边界

泛型中不存在子类的问题，如下

先做一个泛型类



然后去测试泛型的子类问题



类型通配符一般是使用？代替具体的类型实参，注意了，此处’？’是类型实参，而不是类型形参 。重要说三遍！此处’？’是类型实参，而不是类型形参 ！ 此处’？’是类型实参，而不是类型形参 ！再直白点的意思就是，此处的？和Number、String、Integer一样都是一种实际的类型，可以把？看成所有类型的父类。是一种真实的类型。

可以解决当具体类型不确定的时候，这个通配符就是 ? ；当操作类型时，不需要使用类型的具体功能时，只使用Object类中的功能。那么可以用 ? 通配符来表未知类型。

泛型的上下边界则是针对通配符做一个限定，限制泛型不能为某些类型。下图还证明了泛型通配符可以解决子类的问题

