**Java 泛型方法**

@author ixenos

* **泛型方法可以定义在普通类中，也可以定义在泛型类中**
  + 类型变量放在修饰符（如public static）后面，返回类型的前面
  + 定义在泛型类中时，禁止使用带有类型变量的静态方法如
    - public static <T> T getMiddle(T...a) {...}
    - 因为类型擦除，泛型类只有一个模板，也就只有一种静态域和静态方法，无法满足泛型的要求，所以设计者直接不给你编译

* **泛型方法的调用**
  + 调用泛型方法时，在方法名前的尖括号中放入具体的类型
    - String middle = ArrayAlg.<String>getMiddle("I'm","ixenos");
  + 大多数情况下，方法调用可以省略<T>，编译器有足够信息推断出调用的方法，这里利用形参的类型String[]与泛型类型T[]匹配，推断出T是String，故可以调用：
    - String middle = ArrayAlg.getMiddle("I'm","ixenos");

* **有时候形参会判断失误**
  + double middle = ArrayAlg.getMiddle(3.14 , 1234 , 0);
  + 编译器会先自动打包一个Double对象和两个Integer对象，然后寻找这些类的共同超类型，找到Number和Comparable接口，这两个也是泛型类型（如Number<T>），故无法确定，这种情况下补救措施是将所有参数写为double值
  + 如果想知道编译器对一个泛型方法调用最终判断出哪些类型，可以有目的地引入一个错误参数，根据错误消息得到，，如
    - found:
    - java.lang.**Object**&java.io.**Serializable**&java.lang.**Comparable**<? extends java.lang.Object&java.io.Serializable&java.lang.Comparable<?>>
      * 大致意思：可以将结果赋给Object、Serializable或Comparable