**Java泛型中的标记符含义：**

**E - Element (在集合中使用，因为集合中存放的是元素)**

**T - Type（Java 类）**

**K - Key（键）**

**V - Value（值）**

**N - Number（数值类型）**

**？ -  表示不确定的java类型**

**S、U、V** - 2nd、3rd、4th types

遇到<A>,<B>,<K,V>等，是用到了java中的泛型。

**一般使用<T>来声明类型持有者名称**，自定义泛型类时，类持有者名称可以使用T(Type)   
如果是容器的元素可以使用E(Element)，若键值匹配可以用K(Key)和V(Value)等，   
若是<?>，则是默认是允许Object及其下的子类，也就是java的所有对象了。   
  
所以说，如果是字每A,B,C,D...定义的，就是泛型，这里T只是名字上的意义而已T---type,E----Element   
K----key, V----value   
如果是？定义的，就是普通的Object或者其子类   
  
举例说明：   
Set<T> 表示 集合里 是   T类的实例   
List<E> 表示  集合里 是  E类的实例   
List<?> 表示 集合里的对象类型不确定，未指定   
List 同 List<?> 是一样的。   
  
泛型的作用：   
1、用泛型：

**Java代码  收藏代码**

1. List<T> list=**new** ArrayList<T>();
2. T t=list.get(0);

2、不用泛型：

**Java代码  收藏代码**

1. List  list=**new** ArrayList();
2. T t=(T)list.get(0);

相信你已经看出：   
a、用泛型只是确定了集合内的元素的类型，但却是在**编译时确定了元素的类型再取出来时已经不再需要强转**，   
**增强程序可读性，稳定性和效率**  
b、不用泛型时，如果是装入集合操作，那么**元素都被当做Object对待**，失去自己的类型，那么从集合中取出来时， 

往往**需要转型，效率低，容易产生错误**

**【  不再需要强制转型; 】**