java和scala之间的基本数据类型，基本都可以互转——————当然scala自己定义了一些类，语法，scala应该能使用所有的java类。

但有的时候，为什么总是发生scala类无法转成java类呢?

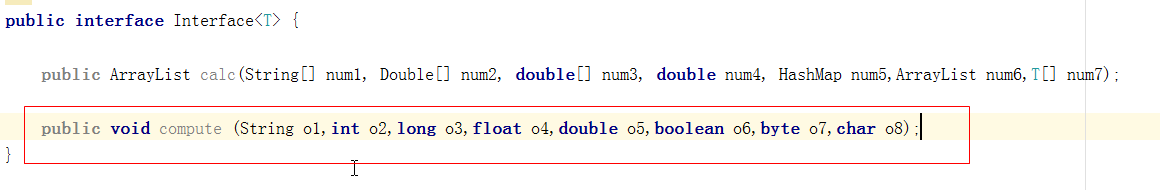
原因： 你的scala在调用java类的时候，没有调用对。

分为两种数据类型的对应：基本类型，引用类型（还有一个特殊的数组，这里属于容器的范畴了）

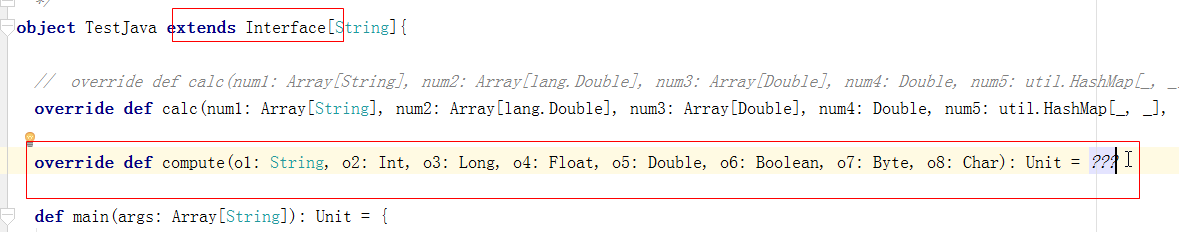
java和scala的基本数据类型的对应，如下，是可以互相转换的（其实就是scala引用了java的基本数据类型）：

我用java的接口测试，然后用scala继承该接口，发现是这种对应关系：

java接口：



scala实现类：



因此对应关系如下：

java————>scala：

String————>String

int——————>Int

long—————>Long

float—————>Float

double————>Double

boolean————>Boolean

byte——————>Byte

char——————>Char

另外： java中的数组[]，对应的就是scala中的Array

其实是“引用类型”————即自定义的（不管是官方java还是用户自定义的）：

像java.lang.Double 、 java.lang.Integer 、java.util.Arrays

这些都是“引用类型”， 是为了方便，对基本数据类型进行操作，从而进行包装。

分别对应double,int, 数组[]

对于这种“引用类型”， scala在调用时，直接调用这些类的java包名.类名（或者先import java包再直接调用类名）

比如： scala如何调用java的数组呢，具体的比如java的 Double[]呢？

方式如下：

1. val arr = Array[java.lang.Double]

千万不要写成Array(Double),因为其就是Array[scala.Double]

而java.lang.Double != scala.Double

scala.Double只是等于java基本类型，double

另一个问题，scala的类型和java类型如何转换呢？

比如： scala 的Array[Double]转成 java的 Double[ ]， 如下方式：

java的Double[ ] 就等于——————scala的Array[ lang.Double]

scala的Array[Double]就等于——————java的 double[ ]

那么问题就是: java的double[ ] 如何转成 java的Double[ ]

貌似只能用for循环转了。 double转Double，用Double.parse(double)