httpsmp.weixin.qq.comsBJuRwSYJTW1u5pd-Ti1drg

# BigDecimal一定不会丢失精度吗？

我们基本已经形成了常识，需要用到金钱的地方要用BigDecimal而不是其他，而我们也都知道浮点型变量在进行计算的时候会出现丢失精度的问题。

那么，你知道其实BigDecimal也会丢失精度吗？而使用BigDecimal的背后又有什么值得去探究的地方吗？今天，告诉你，知其然，也知其所以然。

如下一段代码：



输出：

0.060000000000000005  
0.5800000000000001  
401.49999999999994  
1.2329999999999999

可以看到在Java中进行浮点数运算的时候，会出现丢失精度的问题。那么我们如果在进行商品价格计算的时候，就会出现问题。

很有可能造成我们手中有0.06元，却无法购买一个0.05元和一个0.01元的商品。

因为如上所示，他们两个的总和为0.060000000000000005。

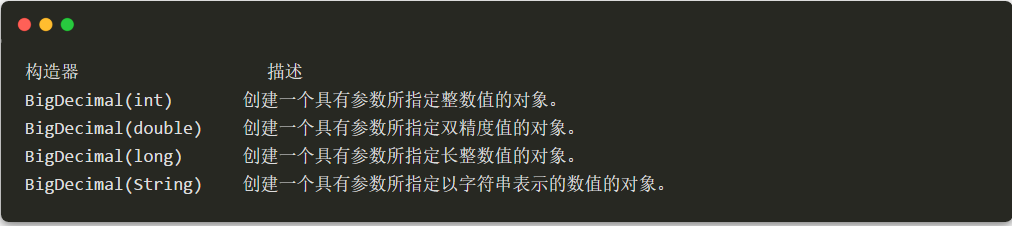
这无疑是一个很严重的问题，尤其是当电商网站的并发量上去的时候，出现的问题将是巨大的。可能会导致无法下单，或者对账出现问题。所以接下来我们就可以使用Java中的BigDecimal类来解决这类问题。

**普及一下：**

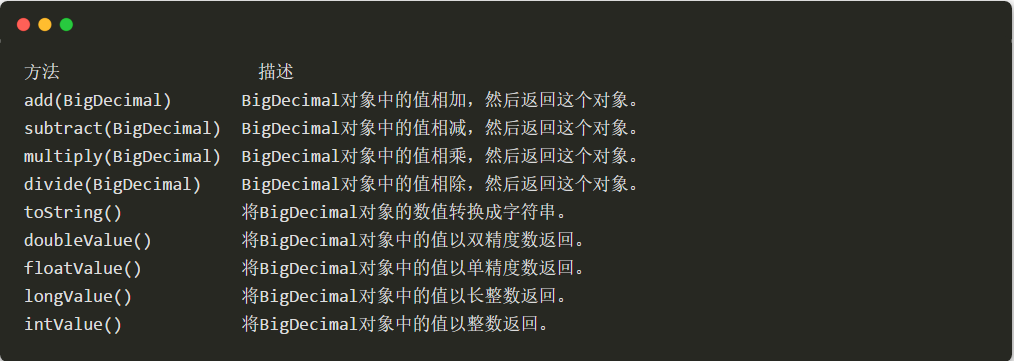
Java中float的精度为6-7位有效数字。double的精度为15-16位。

### API

构造器：



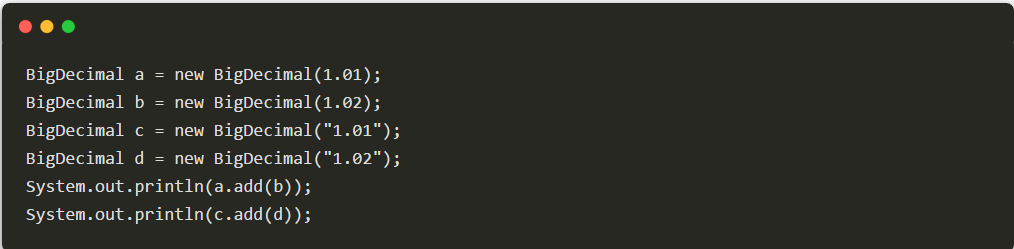
函数：



由于一般的数值类型，例如double不能准确的表示16位以上的数字。

### **BigDecimal精度也丢失**

我们在使用BigDecimal时，**使用它的BigDecimal(String)构造器创建对象才有意义。其他的如BigDecimal b = new BigDecimal(1)这种，还是会发生精度丢失的问题**。如下代码：



输出：  
2.0300000000000000266453525910037569701671600341796875  
2.03

可见论丢失精度BigDecimal显的更为过分。但是使用Bigdecimal的BigDecimal(String)构造器的变量在进行运算的时候却没有出现这种问题。

究其原因计算机组成原理里面都有，它们的编码决定了这样的结果。

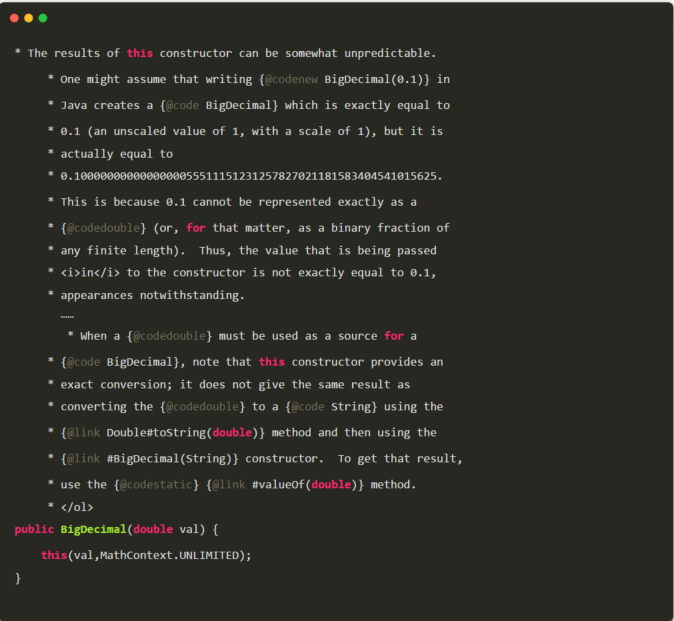
long可以准确存储19位数字，而double只能准备存储16位数字。

double由于有exp位，可以存16位以上的数字，但是需要以低位的不精确作为代价。如果需要高于19位数字的精确存储，则必须用BigInteger来保存，当然会牺牲一些性能。

所以我们一般使用BigDecimal来解决商业运算上丢失精度的问题的时候，声明BigDecimal对象的时候一定要使用它构造参数为String的类型的构造器。

同时这个原则Effective Java和MySQL 必知必会中也都有提及。float和double只能用来做科学计算和工程计算。商业运算中我们要使用BigDecimal。

而且我们从源码的注释中官方也给出了说明，如下是BigDecimal类的double类型参数的构造器上的一部分注释说明：



第一段也说的很清楚它只能计算的无限接近这个数，但是无法精确到这个数。

第二段则说，如果要想准确计算这个值，那么需要把double类型的参数转化为String类型的。并且使用BigDecimal(String)这个构造方法进行构造。去获取结果。

### **正确运用BigDecimal**

另外，BigDecimal所创建的是对象，我们不能使用传统的+、-、\*、/等算术运算符直接对其对象进行数学运算，而必须调用其相对应的方法。方法中的参数也必须是BigDecimal的对象，由刚才我们所罗列的API也可看出。

在一般开发过程中，我们数据库中存储的数据都是float和double类型的。在进行拿来拿去运算的时候还需要不断的转化，这样十分的不方便。这里我写了一个工具类：



该工具类提供了double类型的基本的加减乘除运算。直接调用即可。