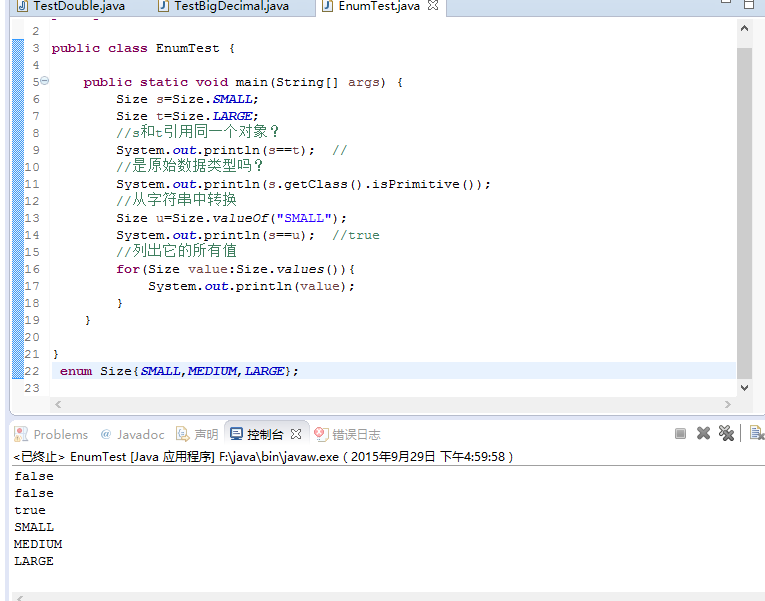
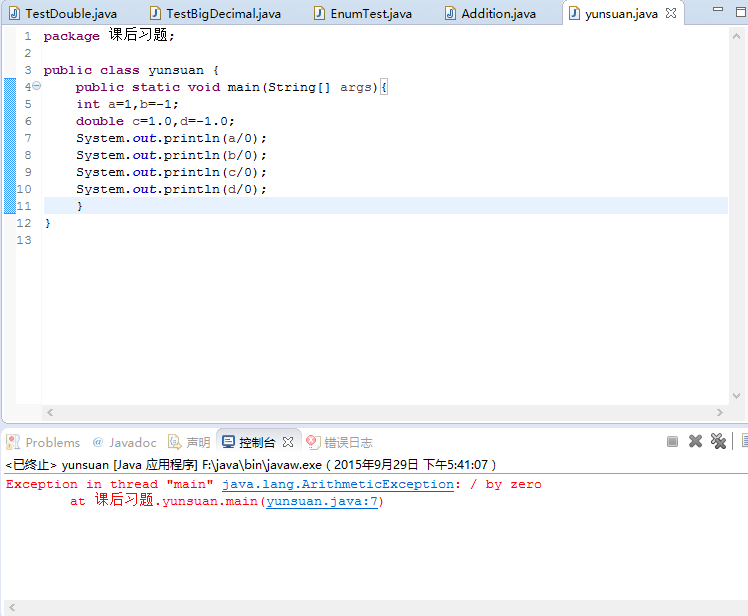
1. 标题： java
2. 时间： 2015-09-29 18:52
3. 博客地址：<http://www.cnblogs.com/java-meng/p/4846875.html>
4. 内容：

1.java中枚举类型的基本用法



java中的枚举类型可以看做是一个类，其中存放着常量，可以看做对象，不能对枚举中的常量赋值，枚举成员就的属性可以不光是字符串，可以是任意对象，因为枚举对象默认是static final 其成员对象都是stastic 属性，直接调用。枚举类型可以先声明后定义，也可直接定义。枚举类型是引用类型！枚举不属于原始数据类型，它的每个具体值都引用一个特定的对象。相同的值则引用同一个对象。可以使用“==”和equals()方法直接比对枚举变量的值，换句话说，对于枚举类型的变量，“==”和equals()方法执行的结果是等价的。

2.java算数运算符

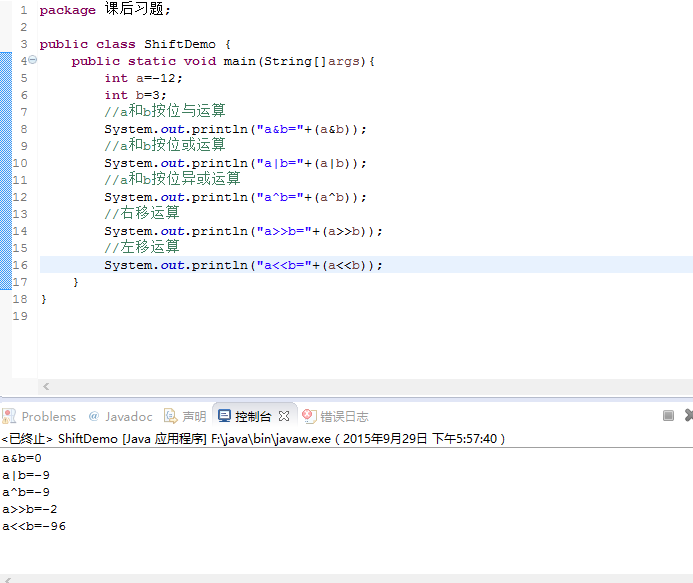


3.位运算符

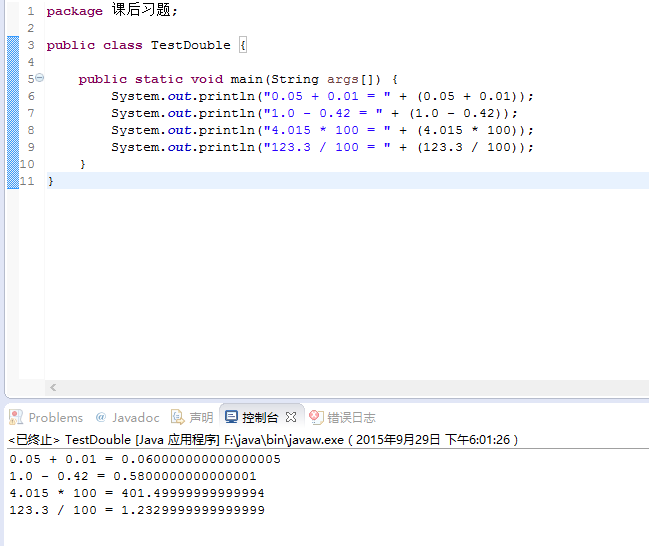
所谓原码就是二进制定点表示法，即最高位为符号位，“0”表示正，“1”表示负，其余位表示数值的大小。

反码表示法规定：正数的反码与其原码相同；负数的反码是对其原码逐位取反，但符号位除外。

补码表示法规定：正数的补码与其原码相同；负数的补码是在其反码的末位加1。



4.double精度测试

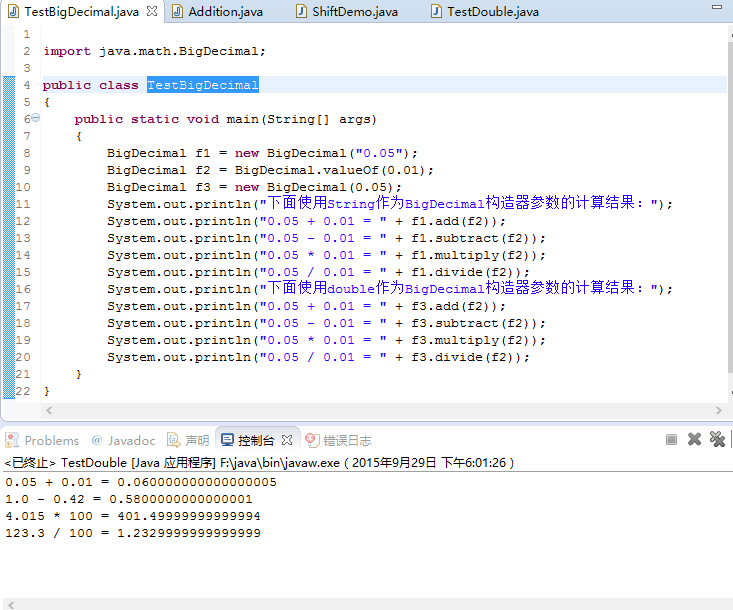


可以看出使用double类型的数值进行计算，  其结果是不精确的。

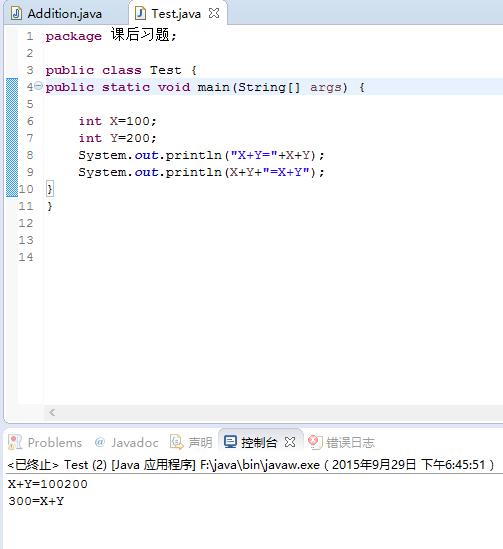
为什么double类型的数值进行运算得不到“数学上精确”的结果？

事实上浮点运算很少是精确的，只要是超过精度能表示的范围就会产生误差。往往产生误差不是 因为数的大小，而是因为数的精度。因此，产生的结果接近但不等于想要的结果。尤其在使用 float 和 double 作精确运 算的时候要特别小心。  
可以考虑采用一些替代方案来实现。如通过 String 结合 BigDecimal 或 者通过使用 long 类型来转换。

还可以用使用BigDecimal类来实现。



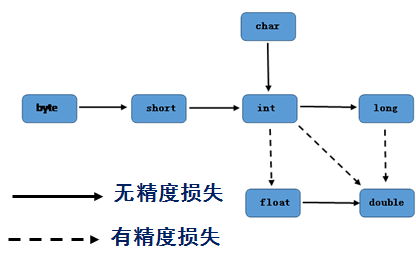
5.输出连接符（+）



有以上结果可知：

如果在加号前面先输出了一串字符串，那么加号后面的数据一律按照字符串格式输出，此时的加号代表着两个字符串相连；反之，如果在加号前面并没有先输出字符串，那么就按照正常的加法运算。

6.Java中的类型转换



由此可以看出数据在内存由大到小的转换中有精度损失，反之没有精度损失。