如果基本的整数和浮点数精度不能够满足需求，那么可以使用java.math包中的两个很有平有用的类：BigInteger和BigDecimal。这两个类可以处理包含任意长度数字序列的数值。

BigInteger类实现了任意精度的整数运算

BigDecimal实现了任意精度的浮点数运算

使用静态的valueOf方法可以将普通的数值转换为大数值：

BigInteger a = BigInteger.valueOf(100);

遗憾的是，不能使用人们熟悉的算术运算符（+ \*）处理大数值。

而需要使用大数值类中的add和multiply方法

BigInteger c = a.add(b) //c = a + b

BigInteger d = c.multiply(b.add(BigInteger.valueOf(2))) // d = c \* ( b + 2 )

Java.math.BigInteger 1.1

BigInteger add(BigInteger other)

BigInteger subtract(BigInteger other)

BigInteger multiply(BigInteger other)

BigInteger divide(BigInteger other)

BigInteger mod(BigInteger other)

返回这个大整数和另一个大整数other的和、差、积、商以及余数

int compareTo(BigInteger other)

=other 返回 0 <other返回负数 否则返回正数

static BigInteger valueOf(long x)

返回值等于x的大整数

java.math. BigDecimal 1.1

BigDecimal add(BigDecimal other)

BigDecimal subtract(BigDecimal other)

BigDecimal multiply(BigDecimal other)

BigDecimal divide(BigDecimal other RoundingMode mode) 5.0

返回这个大实数和另一个大实数other的和、差、积、商

要想计算商，必须给出舍入方式。RoundingMode.HALF\_UP是四舍五入方式

int compareTo(BigDecimal other)

=other 返回0 <other 返回负数 否则返回正数

staitc BigDecimal valueOf(long x)

staitc BigDecimal valueOf(long x,int scale)

返回值为x或x/10scale的一个大实数