Java中的Math类和Random类

1. Math类是属于**java.lang包**中的，使用时不需要导入；而Random类是属于**java.util工具包**中的，使用时需要导入import java.util.Random;。
2. **Math类**是**final的**，不可以被继承，没有子类，直接继承于Object类，提供了**数学运算**所需要的方法。Math类提供的字段和方法都是**静态的**，可以直接利用**类名**进行调用。
3. **Math类提供的字段Fields：只有两个PI和E。**
4. PI ：是double类型的，相当于一个**常量**。获取pi值的方法就是**Math.PI**。



1. E：自然常数e，也是一个double类型的常量。获取e值的方法就是**Math.E**。



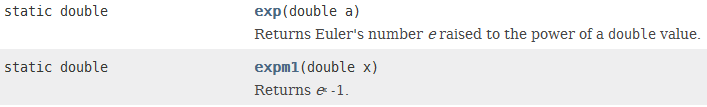
1. Math类提供的常用方法介绍：
2. **abs方法**：**获取绝对值**，可以接收**int、short、byte、float、double数据**。



1. **exp(double a)方法**：获取，即e的a次幂，返回double值。

**expm1（double x）：放回****。注意是m1数字1，不是l。**

如： System.out.println(Math.exp(3)== Math.expm1(3)+1);**//true**



1. **对数方法：log方法：**

以**10 为底数**的对数运算：返回**double值**。



**以自然常数e为底数的对数运算：**



**log1p（double x）方法是先把参数加一然后取其对数。**



**如：**System.out.println(Math.log10(1000));//3.0

System.out.println(Math.log(Math.exp(3)));//3.0

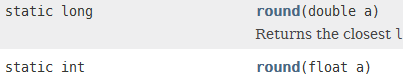
System.out.println(Math.log1p(Math.exp(3) -1));//3.0

1. **三个方法：floor、ceil、round、rint方法，参数都是double类型，floor和ceil、rint返回double类型，而round返回long类型。**

**floor方法**是返回**小于此数的最大整数（double类型）**，**ceil方法**是返回**大于此数的最小整数（double类型）**；rint、**round方法**是按照**四舍五入**返回最接近此数的**整数（long类型）**：**只有round返回的是long整型类型，而floor、ceil和rint返回的仍然是double类型。如例子所示。**









示例：

System.out.println(Math.ceil(56.3243));**//57.0 double类型**

System.out.println(Math.floor(56.3243));**//56.0 double类型**

System.out.println(Math.rint(56.3243));**//56.0 double类型**

System.out.println(Math.round(56.3243));/**/56 long类型**

1. **pow（double a, double b）方法**：返回的是**a的b次方**，返回。



1. **random() 方法** : 返回值为一个大于等于0小于1的double值(左闭右开区间)。注意是大于 0.0，小于1.0.且每次只可以产生一个，如果需要产生多个，利用for循环即可实现了。



1. **开根号运算：**
   1. **sqrt（double a）方法：求平方根。返回double值。square root 平方根sqrt**



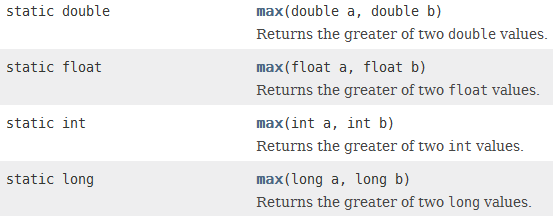
如： Math.sqrt(6.25)//2.5

* 1. **cbrt（double a）方法：求立方根：返回double值。cube root立方根**cbrt

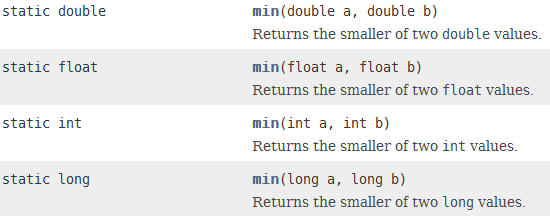
****

如:Math.cbrt(0.027)//0.3

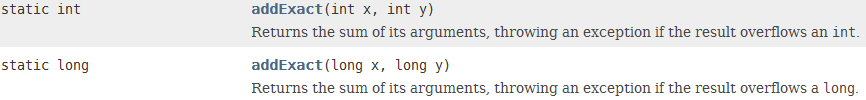
1. **max（）方法：返回两者中的较大者。**



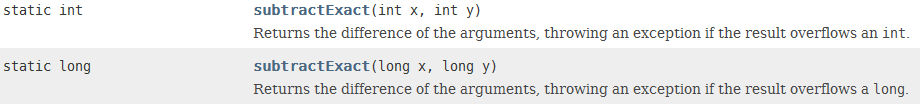
1. **min（） 方法：返回两者中的较小者。**



1. **加法运算：addExact** 
   1. 两个参数都是int类型，如果和超出了int类型的范围，就会抛出异常；
   2. 两个参数都是long类型，如果和超出了long类型的范围，就会抛出异常。



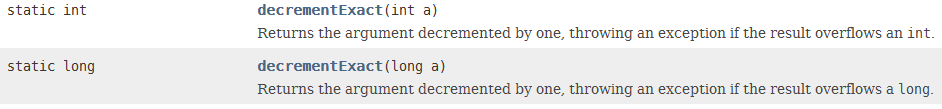
1. **减法运算：substractExact（）方法**



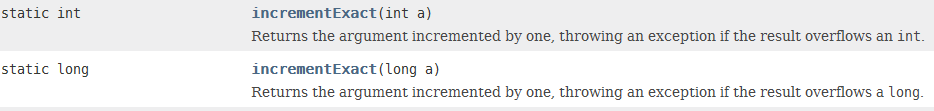
1. **减一运算：decrementExact（）**

参数是int或者long类型，调用一次，数值减一并返回，如果超出了int或者long的范围，就会抛出异常。

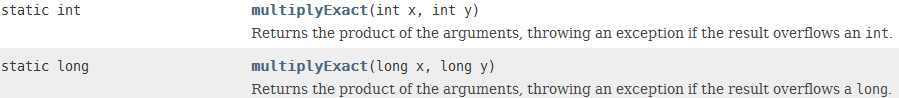
如：Math.decrementExact(Integer.MIN\_VALUE);**//抛出了异常**



1. **加一运算：incrementExact（）：与decrementExact（）方法相似。**



1. **乘法运算：multiplyExact（）方法**



1. **求商方法: floorDiv() 参数都是整型，返回值也是整型。**

**注意区别于操作符 / ：**

**/和floorDiv返回的都是整数，但是floorDiv是始终向负无穷取整数值，而 操作符/ 是始终向 0 靠拢。因此如果商是正的，则返回结果一样，但是如果商是负的，则返回结果不一样。即如果两个参数符号相同同正或同负，则floorDiv和 / 的结果一样，若符号相反，则 / 操作符的结果会比floorDiv的大1.**

**如：**System.out.println(Math.floorDiv(-5,3));//结果为-2

System.out.println(-5/3);//结果为-1

System.out.println(Math.floorDiv(5,-3));//-2

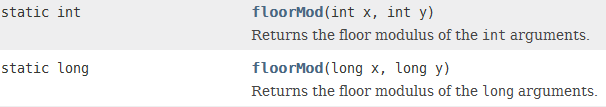
System.out.println(5/-3);//-1

而 System.out.println(Math.floorDiv(5,3));//结果为1

System.out.println(5/3);//结果为1

System.out.println(Math.floorDiv(-5,-3));//结果为1

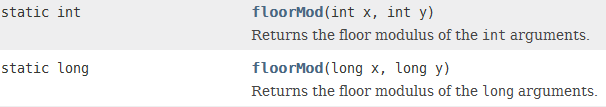
System.out.println(-5/-3); //结果为1



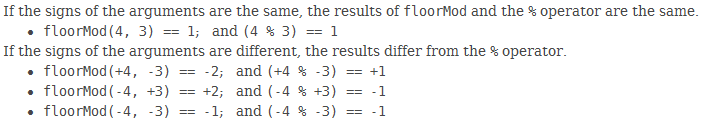
1. **求余方法：floorMod（）**

**floorMod和 % 的区别如同floorDiv 和 / 的区别差不多。**

**如果两个参数同号，则结果一样；若符号相反，则结果就会不同。（整数商一样，则余数一样，floorDiv和floorMod是一致的，%和/也是一致的）；**

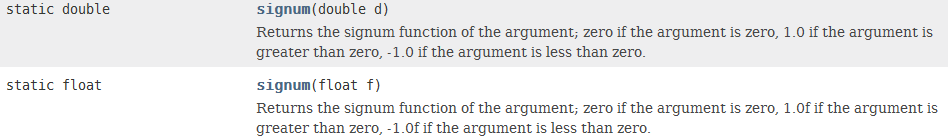


**如：**



1. **符号函数：**

**如果参数等于0，则返回0，如果参数大于0，则放回1.0；如果参数小于0，则返回-1.0。**



1. **三角函数集合：**
   1. 三角函数：
      1. sin



* + 1. cos



* + 1. tan
  1. 反三角函数
     1. acos：取值范围为0到PI;



* + 1. asin：取值范围为-pi/2到pi/2;



* + 1. atan: 取值范围为-pi/2到pi/2;



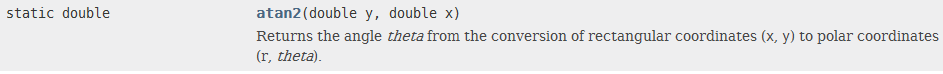
* 1. **atan2（double y,double x）: 功能是把直角坐标系的点坐标（x，y）传进去，然后返回极坐标中的theta值，极坐标中的点利用（r，theta）表示。**

**因为tan=sin/cos=y/x,故求反正切即可求得theta。**

**atan和atan2方法的区别：atan的参数就是一个double值，而atan2的参数是两个double值y和x，其实atan2比atan多了一步而已，atan2先求y/x;然后求反正切即可。如：**

System.out.println(Math.atan2(2,2)/Math.PI); // **返回0.25**

System.out.println(Math.atan(1)/Math.PI); // **返回0.25**



* 1. **双曲函数**：
     1. sinh



* + 1. cosh



* + 1. tanh



1. **弧度和角度之间的转换：**
2. **toDegrees()方法**：**把弧度转换成角度。**

如System.out.println(Math.toDegrees(Math.PI));//180.0



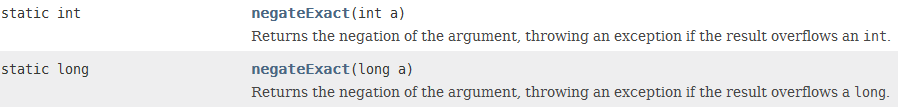
1. **toRadians()方法**：**把角度转换成弧度。**

如：System.out.println(Math.toRadians(180)/Math.PI);//1.0

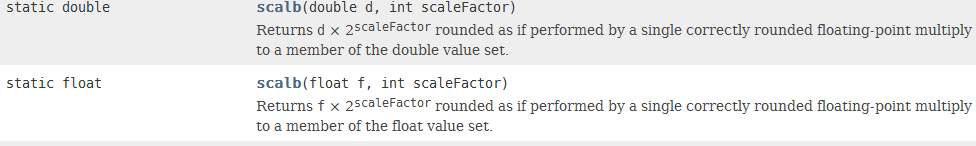


1. 取**相反数**的方法：**negateExact（int/long a）方法**：

**negate英 [nɪ'geɪt] n 对立面**

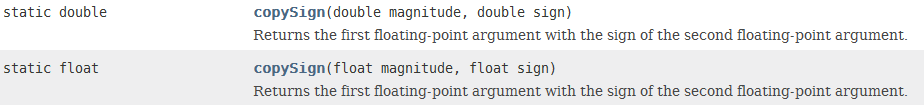


1. **scalb()方法**：返回值为。



1. **copySign方法**：JDK1.6才出现的新方法。

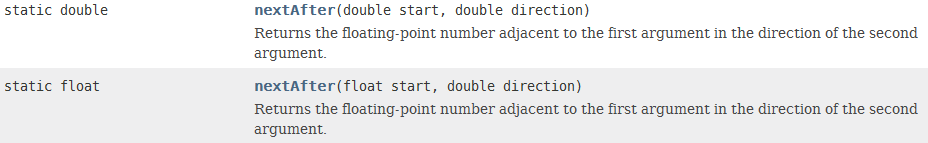
**两个double类型或float类型的参数**，第一个参数提供数值的大小，即提供其的绝对值，第二个参数提供数值的符号，返回**double类型或者float类型**的参数。



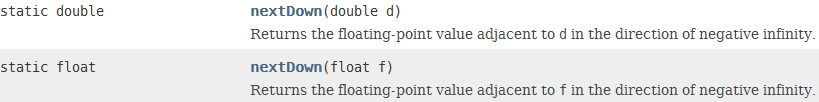
如：Math.copySign(3434.342,-2.32453)**//-3434.342**

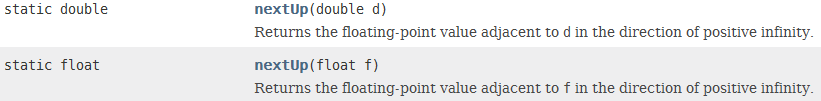
1. nextAfter(double start,double direction)方法：

direction指定向上或向下，返回一个接近start值的一个double值或float值。



nextAfter方法包括**了nextDown和nextUp方法**。





1. **hypot（double x，double y）方法**：返回**x的平方和y的平方**的**和的平方根**。

如System.out.println(Math.hypot(3,4));//5.0



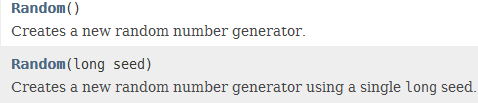
1. **toIntExact()方法**：转换过程中，如果long超出了int的范围，就会抛出异常。



1. Random如果R是大写的，则其是一个类，可以**实例化**。如果random的r是小写，则是Math类中的一个**静态方法**。
2. 因此如果需要产生随机数，可以直接调用**Math.random();方法**，不过**只可以产生一个大于等于0小于1的double值**；也可以生成一个Random的一个对象，通过调用此类对象的方法从而实现**产生随机数**，方法有很多可以查看文档。



1. **Random类：**
2. **构造方法：**



1. **方法：nextXxx方法**

**产生随机数的方法：利用nextXxx方法。**

**nextInt（） 空参数的方法:随机返回一个int值；**

**nextInt（int bound） 方法一个0到bound区间的一个整数值，左闭右开区间：包括0但不包括bound值。**

