# 1.Math类

**Java.lang**

public static long round(double a) 四舍五入

范例:

|  |
| --- |
| System.out.println(Math.round(15.5)); //16  System.out.println(Math.round(-15.51)); //-16  System.out.println(Math.round(-15.5)); //-15 注意  System.out.println(Math.round(-14.5)); //-14 注意 |

如果对负数进行四舍五入,小数大于0.5才进位,小于等于0.5才进位

# 2.Random类

**Java.util**

public int nextInt(int bound) 返回 0<= x < bound

范例:

|  |
| --- |
| Random random = new Random();  System.out.print(random.nextInt(100)); |

# 3.BigInteger(大整型数)

**定义:**

public class BigInteger

extends Number

implements Comparable<BigInteger>

**方法:**

|  |  |
| --- | --- |
| 构造方法 | public BigInteger(String val) |
| 加法操作 | public BigInteger add(BigInteger val) |
| 减法操作 | public BigInteger subtract(BigInteger val) |
| 乘法操作 | public BigInteger multiply(BigInteger val) |
| 除法操作(返回的是一个大整型商值) | public BigInteger divide(BigInteger val) |
| 除法操作(返回一个含2个元素的数组,第一个为商值,第二个为余数 | Public BigInteger[] divideAndRemainder(BigInteger val) |

# 4.BigDecimal(大浮点数)

**定义:**

public class BigDecimal

extends Number

implements Comparable<BigDecimal>

**方法:**

|  |  |
| --- | --- |
| 构造函数接受字符串 | public BigDecimal(String val) |
| 构造函数接受double | public BigDecimal(double val) |
| 除法操作  参数1:被除数  参数2:保留的小数位  参数3:进位模式(public static final int ROUND\_HALF\_UP 四舍五入向上进位) | public BigDecimal divide(BigDecimal divisor,int scale,RoundingMode roundingMode) |

注意:BigDecimal和BigInteger基础计算方法都是一样的

**范例:四舍五入保留指定位数的小数**

|  |
| --- |
| public class test{  public static void main(String args[]) throws Exception{  System.out.println(MyMath.round(19.5,0));  System.out.println(MyMath.round(-19.5,0));  System.out.println(MyMath.round(-19.45433534534,3));  }  }  class MyMath{  /\*\*  \*  \* @param num 要进行四舍五入操作的数字  \* @param scale 要保留的小数位  \* @return 处理后的四舍五入数据  \*/  public static double round(double num,int scale){  BigDecimal bigA = new BigDecimal(num);  BigDecimal bigB = new BigDecimal(1);  return bigA.divide(bigB,scale,BigDecimal.ROUND\_HALF\_UP).doubleValue();  }  } |