

## java int的取值范围为什么负数比正数多1

### 1.int

int的位数在不同的编译器中是不同的，但不会少于16位。

在16位编译器中，一个int是16位的，在32位的编译器中，一个int是32位的。

如果是16位的int，表示数的范围是-32768~+32767。

如果是32位的int，表示数的范围是-2147483648~+2147483647。

### 2. 解释

以16位的int举例

有符号类型中：

int型 16位，15个数值位 1个符号位

正数：0000 0000 0000 0000 至 0111 1111 1111 1111

负数：1000 0000 0000 0000 至 1111 1111 1111 1111

这里有一个0值的差别。

0000 0000 0000 0000 (正零)

1000 0000 0000 0000 (负零)

从数学角度上，是没区别的，可是用两种形式表示一个数，明显是浪费了。

于是计算机存储就约定，当符号位为0，即正零时才是0。

符号位为1时，让它去表示另外一个数好了。

(上面这句话是重点,因为它表示另外一个数,符号位就不少纯粹的符号位了，就代表符号位也会参与取反)

原码:1000 0000 0000 0000首先符号位为1，是个负数，

反码，0111 1111 1111 1111

补码，1000 0000 0000 0000

这时补码中的1代表的就是值了，而不是符号位了。

计算其值就是32768。

于是 1000 0000 0000 0000 就表示成了-32768

---

正数有一个数用来表示+0了，所以少了一个数。当符号位为1，其余位为0时，符号位参与运算，用来表示最小的负整数。