



链滴

jdk 源码: Integer.numberOfLeadingZero S

作者: [Maggie](#)

原文链接: <https://ld246.com/article/1537864275405>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. 源码

```
public static int numberOfLeadingZeros(int i) {
    if (i == 0)
        return 32;
    int n = 1;
    if (i >>> 16 == 0) { n += 16; i <<= 16; }
    if (i >>> 24 == 0) { n += 8; i <<= 8; }
    if (i >>> 28 == 0) { n += 4; i <<= 4; }
    if (i >>> 30 == 0) { n += 2; i <<= 2; }
    n -= i >>> 31;
    return n;
}
```

2. 方法功能

Returns the number of zero bits preceding the highest-order ("leftmost") one-bit in the two's complement binary representation of the specified {@code int} value. Returns 32 if the specified value has no one-bits in its two's complement representation, in other words if it is equal to zero.

翻译：返回int类型32位补码值最左边出现的1之前的0的个数，如果是0因为没有1，就返回32

举个栗子：10的补码为 0000 0000 0000 0000 0000 0000 0000 1010，则最左边1之前有28个0，以此方法返回28。

3. 源码分析

- 0的情况返回32
- 中间部分实际上是二分法的应用，比如第一个判断如果成立，表示第一个非零值位于低16位，然后将i值左移16位使得第一个非零值到达高16位进入第二个判断。依次类推进行二分。
- 还有一个巧妙的地方就是n初始化为1，因此到了最后一步*i*>>>31如果是1，那么n就减1，如果0就不变。

作者 @没有故事的老大爷

别想太多，想不过来

1