



链滴

HashMap 源码解读

作者: [acer](#)

原文链接: <https://ld246.com/article/1534990409827>

来源网站: [链滴](#)

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

数据结构

- HashMap整体结构为数组+链表(解决hash冲突)

length

- 数组默认的 **length**为16，填充因子为0.75，即当数组填充了 $\text{length} \times 0.75$ 个后数组开始扩容为 $\text{length} \times 2$

put

- 根据新计算的 **hash**值取模数组长度，得到散列的**index**，若当前**index**为**null**，直接在**index**处添数据，为**Node**类型，链表的数据结构，若当前**index**不为**null**，循环判断当前的**index.next**是否为**null**添加到当前链表的末尾。当链表长度大于特定的阈值时(默认为8)，将链表结构调整红黑树的结构，**ode**类型更改为**TreeNode**

resize

- 数组扩容时，old数组的元素要重新计算 **index**放入到扩容后的新数组，**next**为空时仍然用**hash**和容后的数组长度去模得到新的**index**，**next**不为空的链表会将**next**的值放到当前**index**，原来该**index**元素放到 $\text{oldIndex} + \text{oldLength}$ (调整过程略复杂，感觉类似于平衡二叉树的自调整)

index

- 计算散列的 **index**时使用的方法是 $\text{hash} \& (\text{length} - 1)$ 。这是一种较为优雅的取模算法，效果等同于 $\text{ash} \% \text{length}$ 。前提是数组长度为2的整数幂
- 验证方法

1010010 % 10000

1010010
& 1111
0000010

高位都是0，只有低位的1才能决定最终的结果，等同于取模运算