**HashMap的原理**

参考： <https://blog.csdn.net/vking_wang/article/details/14166593>

<https://blog.csdn.net/tuke_tuke/article/details/51588156>

<https://www.cnblogs.com/williamjie/p/9358291.html>

<https://blog.csdn.net/qq_26440803/article/details/83795122>

1. 数组的特点是：寻址容易，插入和删除困难；数组存储区间是连续的空间，复杂度很大，但时间复杂度小，为O(1)。
2. 链表的特点是：寻址困难，插入和删除容易；链表存储区间离散，空间复杂度很小，但时间复杂度很大，达到O（N）。
3. Hashmap
4. Hashmap不是线程同步的，允许为null的键和为null的值。
5. 影响hashmap性能的因素：初始容量和负载因子；
6. 每个桶里的链表长度大于8时，将该桶的链表转换成树；
7. 若key.equals(k)为true，则key.hashCode == k.hashCode；若key.hashCode == k.hashCode，key.equals(k)不一定为true；（不同的值可能会存在hash冲突）
8. HashMap死锁：扩容rehash时，多线程导致循环链，从而造成死锁

参考：<https://www.cnblogs.com/kxdblog/p/4323892.html>

1. LinkedHashMap
2. ConcurrentHashMap

参考：<https://www.jianshu.com/p/cf5e024d9432>

Jdk1.7之前：分段锁

Jdk1.8之后：读volatile；写CAS和synchronized