

HashMap

HashMap.....	1
1. 简介.....	2
1.1. HashMap 主要用来存放键值对，它基于哈希表的 Map 接口实现，是常用的 Java 集合之一，是非线程安全的。.....	2
1.2. HashMap 可以存储 null 的 key 和 value，但 null 作为键只能有一个，null 作为值可以有多个.....	2
1.3. HashMap 默认的初始化大小为 16。之后每次扩充，容量变为原来的 2 倍。并且，HashMap 总是使用 2 的幂作为哈希表的大小。.....	2
2. 工作原理.....	2
2.1. 存储机制.....	2
2.1.1. 在 Java 1.8 中，如果链表的长度超过了 8 且数组长度最小要达到 64，那么链表将转化为红黑树；链表长度低于 6，就把红黑树转回链表.....	2
2.2. Java 1.8 中 HashMap 的不同.....	2
2.2.1. 在 Java 1.8 中，如果链表的长度超过了 8，那么链表将转化为红黑树；链表长度低于 6，就把红黑树转回链表；.....	2
2.2.2. 发生 hash 碰撞时，Java 1.7 会在链表头部插入，而 Java 1.8 会在链表尾部插入；.....	2
2.2.3. 在 Java 1.8 中，Entry 被 Node 代替（换了一个马甲）.....	2
2.3. put 过程.....	2
2.3.1. a. 对 Key 求 Hash 值，然后再计算下标.....	2
2.3.2. b. 如果没有碰撞，直接放入桶中（碰撞的意思是计算得到的 Hash 值相同，需要放到同一个 bucket 中）.....	3
2.3.3. c. 如果碰撞了，以链表的方式链接到后面.....	3
2.3.4. d. 如果链表长度超过阈值(TREEIFY THRESHOLD==8)，就把链表转成红黑树，链表长度低于 6，就把红黑树转回链表.....	3
2.3.5. e. 如果节点已经存在就替换旧值.....	3
2.3.6. f. 如果桶满了(容量 16*加载因子 0.75)，就需要 resize（扩容 2 倍后重排）.....	3
2.4. get 过程.....	3
2.4.1. a. 当我们调用 get() 方法，HashMap 会使用键对象的 hashCode 找到 bucket 位置.....	3
2.4.2. b. 找到 bucket 位置之后，会调用 keys.equals() 方法去找到链表中正确的节点.....	3
2.4.3. c. 最终找到要找的值对象.....	3

1. 简介

1.1. HashMap 主要用来存放键值对，它基于哈希表的 **Map** 接口实现，是常用的 **Java** 集合之一，是非线程安全的。

1.2. HashMap 可以存储 **null** 的 **key** 和 **value**，但 **null** 作为键只能有一个，**null** 作为值可以有多个

1.3. HashMap 默认的初始化大小为 **16**。之后每次扩充，容量变为原来的 **2** 倍。并且，**HashMap** 总是使用 **2** 的幂作为哈希表的大小。

2. 工作原理

2.1. 存储机制

2.1.1. 在 **Java 1.8** 中，如果链表的长度超过了 **8**且数组长度最小要达到**64**，那么链表将转化为红黑树；链表长度低于**6**，就把红黑树转回链表

2.2. Java 1.8 中 HashMap 的不同

2.2.1. 在 **Java 1.8** 中，如果链表的长度超过了 **8**，那么链表将转化为红黑树；链表长度低于**6**，就把红黑树转回链表；

2.2.2. 发生 **hash** 碰撞时，**Java 1.7** 会在链表头部插入，而 **Java 1.8** 会在链表尾部插入；

2.2.3. 在 **Java 1.8** 中，**Entry** 被 **Node** 代替（换了一个马甲）

2.3. put过程

2.3.1. a. 对**Key**求**Hash**值，然后再计算下标

2.3.2. b.

如果没有碰撞，直接放入桶中（碰撞的意思是计算得到的Hash值相同，需要放到同一个bucket中）

2.3.3. c. 如果碰撞了，以链表的方式链接到后面

2.3.4. d. 如果链表长度超过阈值(TREEIFY

THRESHOLD==8)，就把链表转成红黑树，链表长度低于6，就把红黑树转回链表

2.3.5. e. 如果节点已经存在就替换旧值

2.3.6. f. 如果桶满了(容量16*加载因子0.75)，就需要 resize（扩容2倍后重排）

2.4. get过程

2.4.1. a.

当我们调用get()方法，HashMap会使用键对象的hashcode找到bucket位置

2.4.2. b.

找到bucket位置之后，会调用keys.equals()方法去找到链表中正确的节点

2.4.3. c. 最终找到要找的值对象