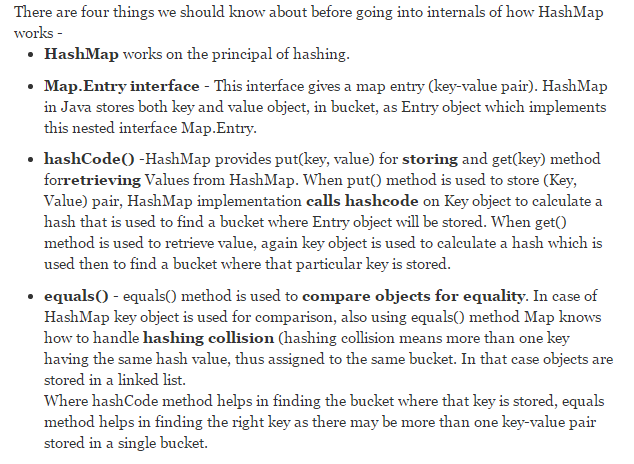
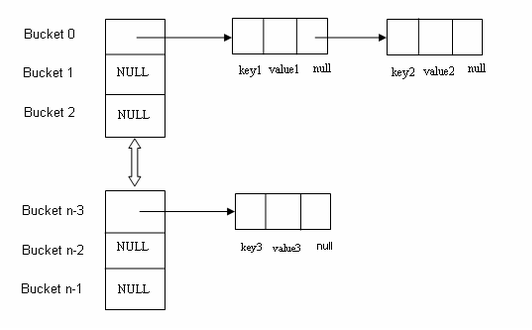
以下是HashMap的四个预备知识：



在java中HashMap的实现其实就是：数组(Array) + 链表(Linked List)

下图就是HashMap的模型，一维数组中的每个格子叫Bucket, 数组的index就是经过HsahCode(Hash Function)算出来的哈希码，每个Bucket中就是存放链表，链表中的每个节点(Node)存放的就是Entry(key-value对)，哈希码是有可能相同的，所以存放的Entry的key需要用Equals方法来判定。

在每次需要用get方法取出值的时候，就会根据key同样经过HsahCode(Hash Function)算出index,就能定位到存放链表的Bucket中，然后再根据key来取出对应的value.



从上文中可以了解到，每次取出时，如果链表的长度太大，就需要循环链表来查找想要的值，这样会大大降低效率，这就是HashMap在Java7和之前版本中的短板。在Java8中，使用balanced tree来解决了一些效率上的问题，原文如下：

