

# HashMap的继承关系

| 丕/

```
public class HashMap<K,V> extends AbstractMap<K,V>
    implements Map<K,V>, Cloneable, Serializable {
```

HashMap继承自父类AbstractMap, 实现了Map、Cloneable、Serializable接口。其中, Map接口定义了一组通用的操作, Cloneable接口可以进行拷贝, 在HashMap中, 实现的是浅拷贝; Serializable接口用于HashMap实现了序列化。

# HashMap的数据结构

HashMap底层以数组+链表存储, 在1.8之后引入了红黑树, 引入红黑树是为了提高效率。在链表数据个数达到8即转换成红黑树, 在个数减少至6又转换为链表。

# HashMap的put方法

```
/**
 * Associates the specified value with the specified key in this map.
 * If the map previously contained a mapping for the key, the old
 * value is replaced.
 *
 * @param key key with which the specified value is to be associated
 * @param value value to be associated with the specified key
 * @return the previous value associated with <tt>key</tt>, or
 *         <tt>null</tt> if there was no mapping for <tt>key</tt>.
 *         (A <tt>null</tt> return can also indicate that the map
 *         previously associated <tt>null</tt> with <tt>key</tt>.)
 */
public V put(K key, V value) { return putVal(hash(key), key, value, onlyIfAbsent: false, evict: true); }
```

HashMap以key, value的形式存储数据, key, value都允许为null, 在执行put方法时, 首先获取key的HashCode值在确定在数组中的位置, 假设数据该位置没有数据, 则直接存入, 若有数据, 则调用Equals方法比较key是否相同, 相同则替换, 不同则组成链表;

# HashMap1.7与1.8的区别 :

1.8主要优化减少了Hash冲突, 提高哈希表的存、取效率; 1.7底层结构是数组+链接, 1.8是数组+链表+红黑树, 当链表数据大于8时, 转换为红黑树, 链表小于6时转换为链表; 往链表添加数据时, 1.7添加在链表头部, 1.8添加在尾部, 不容易出现环形链表; 扩容时, 1.7会打乱原链表顺序, 1.8会保证原链表顺序。