* 学习目标

  * 能够掌握Map子类HashMap

  * 能够掌握Map子类LinkedHashMap

  * 能够掌握Map子类TreeMap

---

  * 回顾

   * Map

      * Key-Value，映射，集合

      * 身份证号和个人，题号和答案，用户与购物车，...

      * 双列的集合

   * 继承框架图

      * Map

         * HashMap

            * LinkedHashMap

         * TreeMap

         * ConcurrentHashMap

         * HashTable

            * Properties

   * Map常用方法

      * put，get，remove，clear，isEmpty，size，containsKey,containsValue,keySet,entrySet,values

   * HashMap

      * Hash,Hash算法（标志），MD5，Hash碰撞（冲突）

      * HashMap 底层数据结构（散列表，散列桶）

         * JDK1.8之前：数组+链表

           JDK1.8：数组+链表+红黑树

* 链表转换红黑树过程（树化）：当链表长度大于等于8并且数组长度大于64

* 红黑树转换链表：小于等于6

* Node（hash（key），key，value，next）

* 6621312%table.length：16：0-15

* HashMap的源码分析

* 常量

```
 * DEFAULT_INITIAL_CAPACITY: 1<<4,2的n次方
 * MAXIMUM_CAPACITY: 1 << 30
 * DEFAULT_LOAD_FACTOR:0.75f
 * TREEIFY_THRESHOLD:8
 * MIN_TREEIFY_CAPACITY:64
 * UNTREEIFY_THRESHOLD:6

```

* 节点构造

```
Node<K,V> implements Map.Entry<K,V>{
    int hash;
    K k;
    V v;
    Node<K,V> next;
    // 构造器
}

Map.Entry<K,V>{
    K getKey();
    V getValue();
}
```

* 变量

```
* table
* size
* Set<Map.Entry<K,V>>
* threshold
* loadFactor
```

    * put
        * hash
    * resize
    * get
    * 总结

* 能够掌握Map子类HashMap
* 能够掌握Map子类LinkedHashMap
* 能够掌握Map子类TreeMap