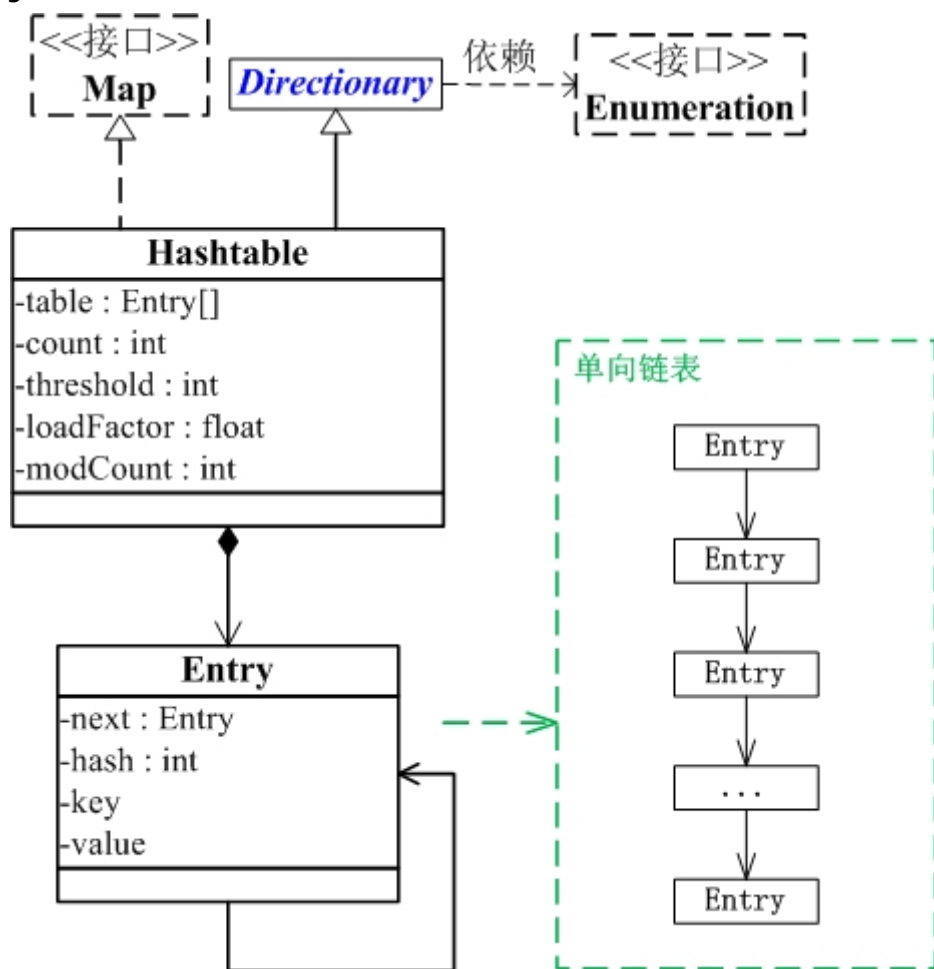


散列表，存储键值对映射。Hashtable和HashMap用法和底层实现几乎一样，key、value都不可以为null。Hashtable中的映射不是有序的。不过Hashtable的方法添加了synchronized关键字确保线程同步检查，效率较低。

Hashtable **继承于Dictionary**，实现Map、Cloneable、java.io.Serializable接口。也是通过拉链法解决哈希冲突的。



主要API

clear()

contains() 和 **containsValue()**

containsKey()

elements()

elements() 的作用是**返回“所有value”的枚举对象**

(01) 若Hashtable的实际大小为0,则返回“空枚举类”对象

emptyEnumerator ;

(02) 否则，返回正常的**Enumerator对象**。(Enumerator实现了迭代器和枚举两个接口)

Enumerator的作用是**提供了“通过elements()遍历Hashtable的接口”和“通过entrySet()遍历Hashtable的接口”**。因为，它同时实现了“Enumerator接口”和“Iterator接口”。

get()

获取key对应的value，没有的话返回null

put()

putAll()

remove()

序号	方法描述
1	void clear() 将此哈希表清空，使其不包含任何键。
2	Object clone() 创建此哈希表的浅表副本。
3	boolean contains(Object value) 测试此映射表中是否存在与指定值关联的键。
4	boolean containsKey(Object key) 测试指定对象是否为此哈希表中的键。
5	boolean containsValue(Object value) 如果此 Hashtable 将一个或多个键映射到此值，则返回 true。
6	Enumeration elements() 返回此哈希表中的值的枚举。
7	Object get(Object key) 返回指定键所映射到的值，如果此映射不包含此键的映射，则足 (key.equals(k)) 的从键 k 到值 v 的映射，则此方法返回 v
8	boolean isEmpty() 测试此哈希表是否没有键映射到值。
9	Enumeration keys() 返回此哈希表中的键的枚举。
10	Object put(Object key, Object value) 将指定 key 映射到此哈希表中的指定 value。
11	void rehash() 增加此哈希表的容量并在内部对其进行重组，以便更有效地容
12	Object remove(Object key) 从哈希表中移除该键及其相应的值。
13	int size()

	返回此哈希表中的键的数量。
14	String toString() 返回此 Hashtable 对象的字符串表示形式，其形式为 ASCII 5 条目。

遍历

遍历Hashtable的键值对

第一步：根据entrySet()获取Hashtable的“键值对”的Set集合。

第二步：通过Iterator迭代器遍历“第一步”得到的集合。

通过Iterator遍历Hashtable的值

第一步：根据value()获取Hashtable的“值”的集合。

第二步：通过Iterator迭代器遍历“第一步”得到的集合。

通过Enumeration遍历Hashtable的键

第一步：根据keys()获取Hashtable的集合。

第二步：通过Enumeration遍历“第一步”得到的集合。

通过Enumeration遍历Hashtable的值

第一步：根据elements()获取Hashtable的集合。

第二步：通过Enumeration遍历“第一步”得到的集合。