# Map容器

多例集合

Map是关联式容器，而不是集合容器

Map接口定义的集合称为查找表，用来存储key-value映射，key可以看成value的索引，key在集合中是不可重复的

Map接口定义的操作

int size();返回存储key-value的个数

boolean isEmpty();判断是否为空，为空的话返回true

boolean containsKey(Object key);判断映射表中是否存在key的映射关系，存在返回true

boolean containsValue(Object value);判断映射表中是否存在value的映射关系，存在返回true

V get(Object key);获取key对应的value

V put(K key, V value);添加key-value

V remove(Object key);删除key-value

void putAll(Map<? extends K, ? extends V> m);将当前指定的映射表复制到当前的映射表

void clear();清除映射表

Set<K> keySet();返回key集合

Collection<V> values();返回value的集合

Set<Map.Entry<K, V>> entrySet();返回内部节点集合

内部接口，实现类是用来存放key和value的节点

interface Entry<K,V> {

V getValue();

V setValue(V value);

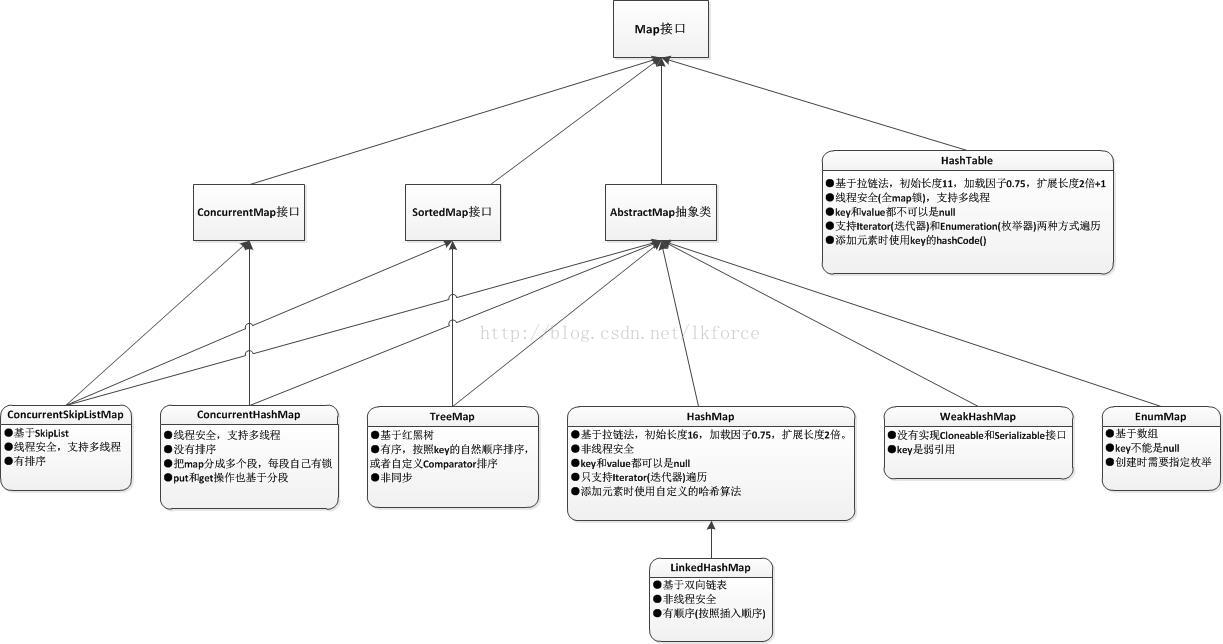
boolean equals(Object o);

int hashCode();

}

boolean equals(Object o);

int hashCode();



Hash介绍

hash:把输入值通过散列算法换算成对应的输出值，改输出就是散列值，这中转换就是映射

hash算法：将一个数据转换为一个标志，hash算法是一种广义的算法，也可以认为是一种思想，使用hash算法可以提高存储空间的利用率，

可以提供数据的查询效率；hash算法也被称为散列算法，虽然是一种算法，实际上更像是一种思想，hash算法没有固定的公式，只要满足散列思想的算法

都可以称为散列算法，如果2个散列值不同，那么输入key也不相同

存储输出值的数据结构称为hash表

**散列地址**：输出值在hash表的位置

常用的hash函数（hash算法）：

直接寻址发：取关键字或者关键字的某个线性函数的值作为散列地址

数字分析法：数字分析法就是找出数字的规律，尽可能利用这些数据来构造冲突几率较低的散列地址

平方取值法：取关键字平方后的中间几位作为散列地址

折叠法：将关键字分割成位数相同的几部分，最后一部分位数可以不同，然后取这几部分的叠加和（去除进位）作为散列地址。

随机数法：选择一随机函数，取关键字作为随机函数的种子生成随机值作为散列地址，通常用于关键字长度不同的场合。

除留余数法：取关键字被某个不大于[散列表](https://baike.baidu.com/item/%E6%95%A3%E5%88%97%E8%A1%A8)表长m的数p除后所得的余数为散列地址。

**Hash冲突**：2个不同的输入值（key）通过hash算法运算后得到相同的散列地址，这种现象就称为hash冲突；

当关键字值域（key的个数）远大于哈希表的长度，而且事先并不知道关键字的具体取值时。冲突就难免会发 生

解决hash冲突的方法

通过良好性能哈希函数可以减少冲突，但是不能完全避免hash冲突

链地址发：将所有hash地址相同的元素构建成一个链表，让链表的表头指向当前hash地址，隐藏查询删除添加主要是在链表上进行的，hashmap就是利用的这个思想来构建的

其他没有在实际应用的遇到过（忽略）（）