maps:  
一个map 模型是 一组可以搜索的key-value entry组成的collection

一个map允许我们以利用key来加速定位element的方式来存储element

map的主要操作是searching,inserting,deleting

不允许多个entry有同一个key，换句话来说map里的key是独立的

1. HashMap<Integer, Integer> hashMap = new HashMap<>();
3. hashMap.put(5, 2);
4. hashMap.put(9, 2);
5. hashMap.put(8, 1);
6. hashMap.put(7, 3);

遍历

（1）利用haspmap.entrySet().iterator()：利用迭代器，从Entry中取出键、取出值，推荐使用这种方式进行遍历，效率较高

Iterator<Entry<Integer, Integer>> iterator = hashMap.entrySet().iterator();

while (iterator.hasNext()) {

Entry<Integer, Integer> entry = iterator.next();

Integer key = entry.getKey();

Integer value = entry.getValue();

System.out.print(key + "--->" + value);

System.out.println();

A:Map集合的功能概述

* a:添加功能
  + V put(K key,V value):添加元素。
    - 如果键是第一次存储，就直接存储元素，返回null
    - 如果键不是第一次存在，就用值把以前的值替换掉，返回以前的值
* b:删除功能
  + void clear():移除所有的键值对元素
  + V remove(Object key)：根据键删除键值对元素，并把值返回
* c:判断功能
  + boolean containsKey(Object key)：判断集合是否包含指定的键
  + boolean containsValue(Object value):判断集合是否包含指定的值
  + boolean isEmpty()：判断集合是否为空
* d:获取功能
  + Set<Map.Entry<K,V>> entrySet():
  + V get(Object key):根据键获取值
  + Set<K> keySet():获取集合中所有键的集合
  + Collection<V> values():获取集合中所有值的集合
* e:长度功能
  + int size()：返回集合中的键值对的个数