Given an array of size *n*, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

待测试输出版本

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** MajorityElement {

**public** **int** majorityElement(**int**[] nums) {

Map<Integer,Integer> m = **new** HashMap<Integer,Integer>();

**for**(**int** i=0; i<nums.length;i++) {

**if**(!m.containsKey(nums[i])) {

m.put(nums[i], 1);

}

**else** {

**int** temp = m.get(nums[i]);

m.put(nums[i], ++temp);

}

}

**for** (Map.Entry<Integer, Integer> entry : m.entrySet()) {

**if**(entry.getValue()>(nums.length)/2) **return** entry.getKey();

}

**return** 0;

}

**public** **static** **void** main(String args[]) {

MajorityElement m = **new** MajorityElement();

**int**[] nums = {1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,4,5,6,7};

**int** r = m.majorityElement(nums);

System.***out***.println(r);

}

}