class DictionaryNode<K, V> implements Comparable<DictionaryNode<K, V>> {  
 K key;  
 V value;  
  
 public DictionaryNode(K key, V value) {  
 this.key = key;  
 this.value = value;  
 }  
  
 @Override  
 public int compareTo(DictionaryNode<K, V> node) {  
 return ((Comparable<K>) key).compareTo(node.key);  
 }  
}

public class DictionaryEntry {  
 String key;  
 String value;  
  
 public DictionaryEntry(String k, String v) {}  
 public String getKey() {}  
 public String getValue() {}  
}

public interface DictionaryADT<K,V> {  
 boolean contains(K key);  
 boolean add(K key, V value);  
 boolean delete(K key);  
 V getValue(K key);  
 K getKey(V value);  
 int size();  
 boolean isFull();  
 boolean isEmpty();  
 void clear();  
 Iterator<K> keys();  
 Iterator<V> values();  
}