链滴

手写 HashMap

作者：vitalQ

原文链接：https://ld246.com/article/1536504941455

来源网站：链滴

接口:

```
public interface VMap<K, V> {

  V put(K key, V value);

  V get(K key);

  Integer size();

 interface Entry<K, V> {

   K getKey();

    V getValue();
  }
}
```

实现:

```
import java.util.Arrays;

public class VHashMap<K, V> implements VMap<K, V> {

    //大小
  private Integer size = 0;
  //默认大小
  private Integer defaultLength = 16;
  //负载因子
  private double loadFactor = 0.75D;
  //存储数据的数组
  private Entry<K, V>[] table;

  VHashMap(Integer _defaultLength, double _loadFactor) {
    this.defaultLength = _defaultLength;
     this.loadFactor = _loadFactor;
     this.table = new Entry[defaultLength];
  }

    VHashMap() {
      this.table = new Entry[defaultLength];
  }

    Integer getIndex(K key) {
      return Math.abs((key.hashCode() % (defaultLength - 1)));
  }

  @Override
  public V put(K key, V value) {
        //获取下标
    Integer index = getIndex(key);
    //判断是否足够存储
    if (table.length <= index + 1) {
```

```
    Integer len = table.length + (int) (table.length * loadFactor);
    Entry<K, V>[] newTable = Arrays.copyOf(table, len);
   this.table = newTable;
 }
 //判断下标是否被占用
 Entry<K, V> kvEntry = table[index];
 //没有被占用
 if (kvEntry == null) {
   table[index] = new Entry(key, value, null, index);
   size++;
 } else {
   //判断是否相同的key
   if (kvEntry.key.equals(key)) {
     //覆盖
     table[index] = new Entry(key, value, kvEntry.next, index);
   } else {
     //把新的值放进去
     table[index] = new Entry(key, value, kvEntry, index);
     size++;
   }
}

     return table[index].getValue();
 }

  @Override
 public V get(K key) {
 Integer index = getIndex(key);
   Entry<K, V> kvEntry = table[index];
   do {
     if (kvEntry.key.equals(key)) {
       return table[index].getValue();
     }
     Entry<K, V> next = kvEntry.next;
     if (next == null) {
       return null;
     }
     kvEntry = kvEntry.next;
   } while (!key.equals(kvEntry.key));

    return kvEntry.getValue();
 }

   @Override
   public Integer size() {
        return size;
   }

 class Entry<K, V> implements VMap.Entry<K, V> {

  private Entry<K, V> next;
  private Integer index;
  private K key;
  private V value;
```

```
        Entry(K _key, V _value, Entry<K, V> _next, Integer _index) {
          this.key = _key;
         this.value = _value;
         this.next = _next;
         this.index = _index;
        }

          Entry() {
          }

        @Override
        public K getKey() {
            return this.key;
        }

        @Override
        public V getValue() {
              return this.value;
        }
      }
}
```

原文链接：手写 HashMap