链滴

动手写个可变长顺序表

作者：JackHoo

原文链接：https://ld246.com/article/1510057936758

来源网站：链滴

一.线性表的描述

线性表是其组成元素之间具有线性关系的一种线性结构，比如java集合框架里面的ArrayList和LinkedList等都是典型的线性表。对线性表的基本操作主要有**插入**，**删除**，**查找**，**替换**等。这些操作可以在线表的任何位置进行。线性表可以采用顺序存储结构和链式存储结构来表示，采用顺序存储结构表示就是**序表**，比如ArrayList，采用链式结构表示就是**链表**，比如LinkedList。

二.顺序表代码实现

```
package List;

import java.util.Arrays;

/**
 * Created by hushangjie on 2017/11/7.
 */
 public class MyArrayList<T> {
    private static final int INITIAL_CAPACITY = 10;
    private int size = 0;
    private Object[] array;

    public MyArrayList(int initial) {
        if (initial <= 0) {
            throw new RuntimeException("init error!");
        }
        array = new Object[initial];
    }

    public MyArrayList() {
        this(INITIAL_CAPACITY);
    }
    public int size(){
        return size;
    }

    //添加元素
  public void add(Object o) {
        if ((size + 1) > array.length) {
            //增加容量
  int oldCapacity = array.length;
            int newCapacity = oldCapacity * 2;
            array = Arrays.copyOf(array, newCapacity);
        }
        array[size++] = o;
    }

    //获取指定位置元素
  public T get(int index) {
        rangeCheck(index);
        return (T) array[index];
    }
```

原文链接：动手写个可变长顺序表

```
    //给指定位置赋值，返回原来的值
    public T set(int index, T t) {
        rangeCheck(index);
        T old = (T) array[index];
        array[index] = t;
        return old;
    }

    //删除指定下标元素
    public T remove(int index) {
        rangeCheck(index);
        T old = (T) array[index];
        int numMoved = size - index - 1;
        if (numMoved > 0) {
            System.arraycopy(array, index + 1, array, index, numMoved);
        }
        array[--size]= null;
        return old;
    }

    //获取指定元素下标
    public int indexOf(Object o) {
        if (o == null) {
            for (int i = 0; i < size; i++) {
                if (array[i] == null) {
                    return i;
                }
            }
        }else {
            for (int i = 0; i < size; i++) {
                if (o.equals(array[i])) {
                    return i;
                }
            }
        }
        return -1;

    }

    //范围检测
    public void rangeCheck(int index) {
        if (index < 0 || index > array.length) {
            throw new IndexOutOfBoundsException("out of size");
        }
    }
}
```