



链滴

# 用 Java 数组实现栈结构

作者: [wky181](#)

原文链接: <https://ld246.com/article/1571062871128>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 用Java数组实现栈结构

最近在学数据结构这门课，不过数据结构这门课是用的C语言，而我主要学的是Java，所以就用Java现一下。栈结构是先进后出的数据结构，在算法中非常有用，以后也会写一些关于栈的算法题解。

- 首先写出栈的接口，定义栈的基本操作。

```
public interface IStack<T> {  
    /**  
     * 元素入栈  
     * @param e  
     */  
    void push(T e);  
  
    /**  
     * 弹出栈顶  
     * @return  
     */  
    T pop();  
  
    /**  
     * 是否空栈  
     */  
    boolean empty();  
  
    /**  
     * 获得栈内元素个数  
     * @return  
     */  
    int getSize();  
  
    /**  
     * 弹出栈顶元素  
     * @return  
     */  
    T peek();  
}
```

- 然后定义一个栈类实现这个接口，实现原理很简单，首先定义栈顶下标、栈底下标、置零。入栈从数组后面添加元素，栈顶下标+1，出栈从数组后面取出元素栈顶下标-1，**栈顶下标永远是要插入元素的标**，当栈顶下标和栈底下标相等时表示空栈。

```
public class MyOrderStack<T> implements IStack {  
    private T[] elements;  
    private int capacity = 10;  
    //栈底下标  
    private int base = 0;  
    //栈顶下标  
    private int top = 0;
```

```

private int size = 0;
public MyOrderStack(int capacity){
    this.capacity = capacity;
    this.elements = (T[]) new Object [capacity];

}
public MyOrderStack(){
    this.elements = (T[]) new Object [capacity];
}
@Override
public void push(Object e) {
    //判断数组是否已满
    if (top==capacity){
        //建立一个是原来空间二倍的数组
        capacity = capacity *2;
        T[] temp = (T[]) new Object[capacity];
        for (int i = 0; i <elements.length ; i++) {
            //将原来数组元素挪进新数组
            temp [i] = elements [i];
        }
        elements = temp;
    }
    elements [top++] = (T) e;
    size++;
}

@Override
public Object pop() {
    if (top == base){
        throw new RuntimeException("栈空");
    }
    size--;
    return elements [--top];
}

@Override
public boolean empty() {
    return top == base;
}

@Override
public int getSize() {
    return size;
}

@Override
public Object peek() {
    return elements [top-1];
}
}

```