

栈的实现

```
Interface Stack<E>    ◀----- ArrayStack<E>
• void push(E)        implement
• E pop()
• E peek()
• int getSize()
• boolean isEmpty()
```

@9138126

慕课网

栈的复杂度分析

```
ArrayStack<E>
• void push(E)          0(1) 均摊
• E pop()               0(1) 均摊
• E peek()              0(1)
• int getSize()         0(1)
• boolean isEmpty()     0(1)
```

慕课网

队列的实现

```
Queue<E>
• void enqueue(E)
• E dequeue()
• E getFront()
• int getSize()
• boolean isEmpty()
```

慕课网

数组队列的复杂度分析

ArrayQueue<E>

- void enqueue(E) $O(1)$ 均摊
- E dequeue() $O(n)$
- E front() $O(1)$
- int getSize() $O(1)$
- boolean isEmpty() $O(1)$

4 腾讯网

为什么要使用循环队列；以及什么时候循环队列为空或者为满？

循环队列

front == tail 队列为空

$(tail + 1) \% c == front$ 队列满

data

front



capacity

tail

capacity中，浪费一个空间

4 腾讯网