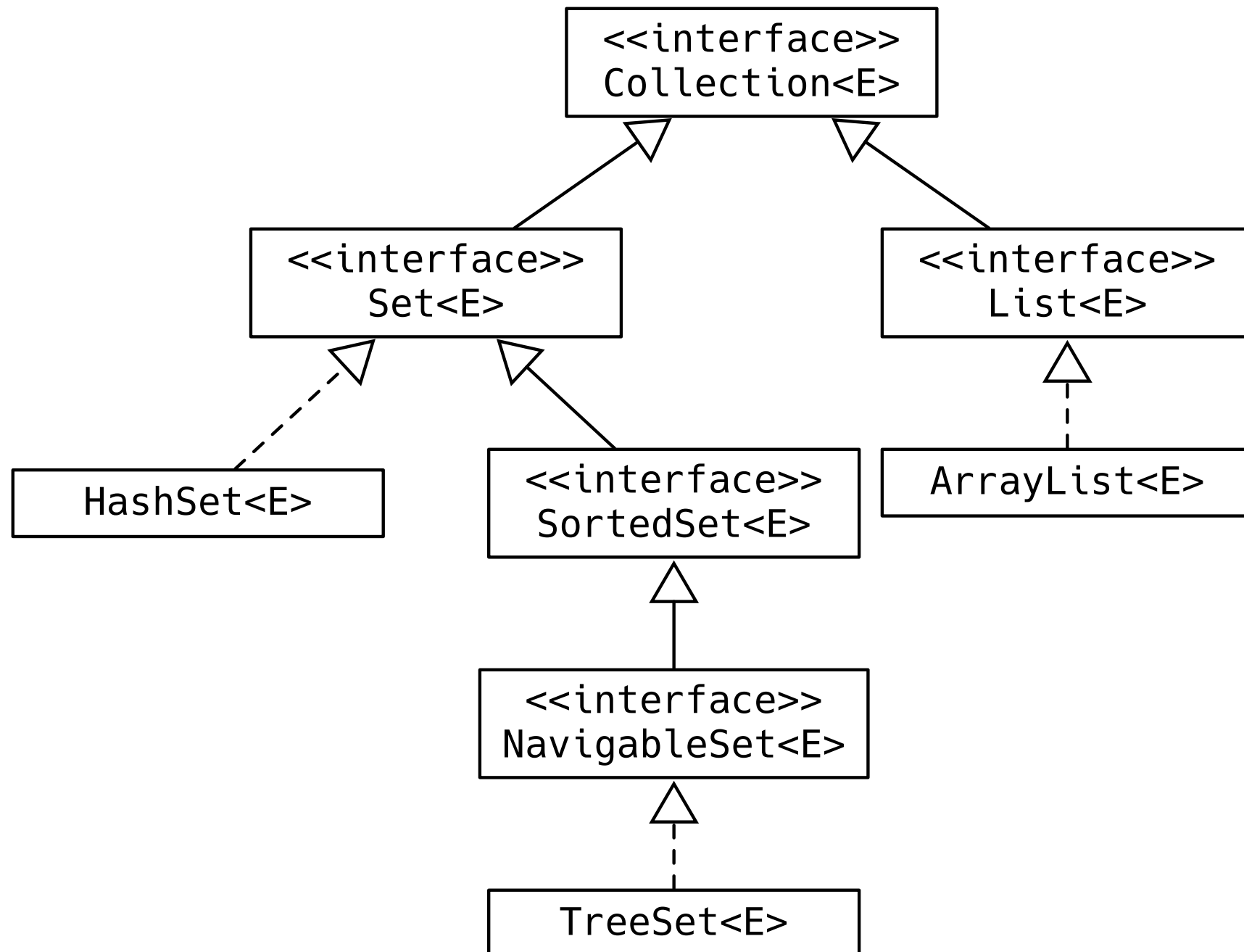
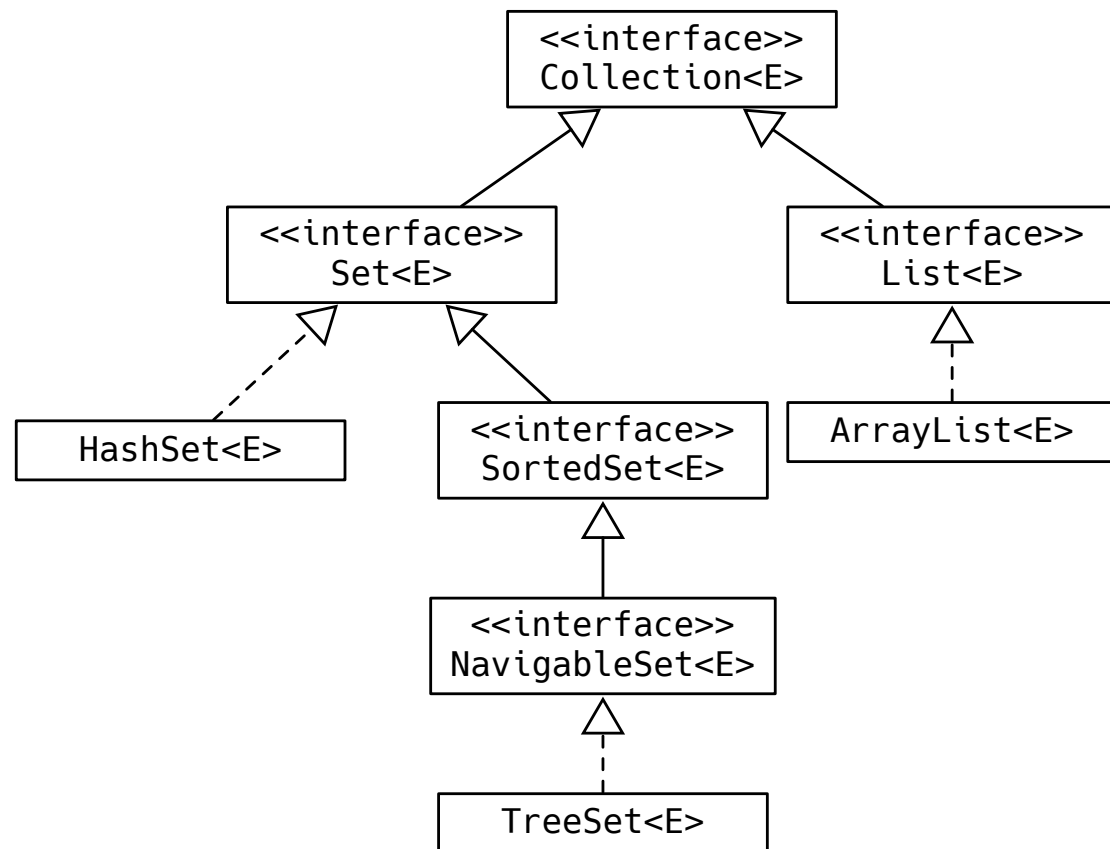


Collection Framework

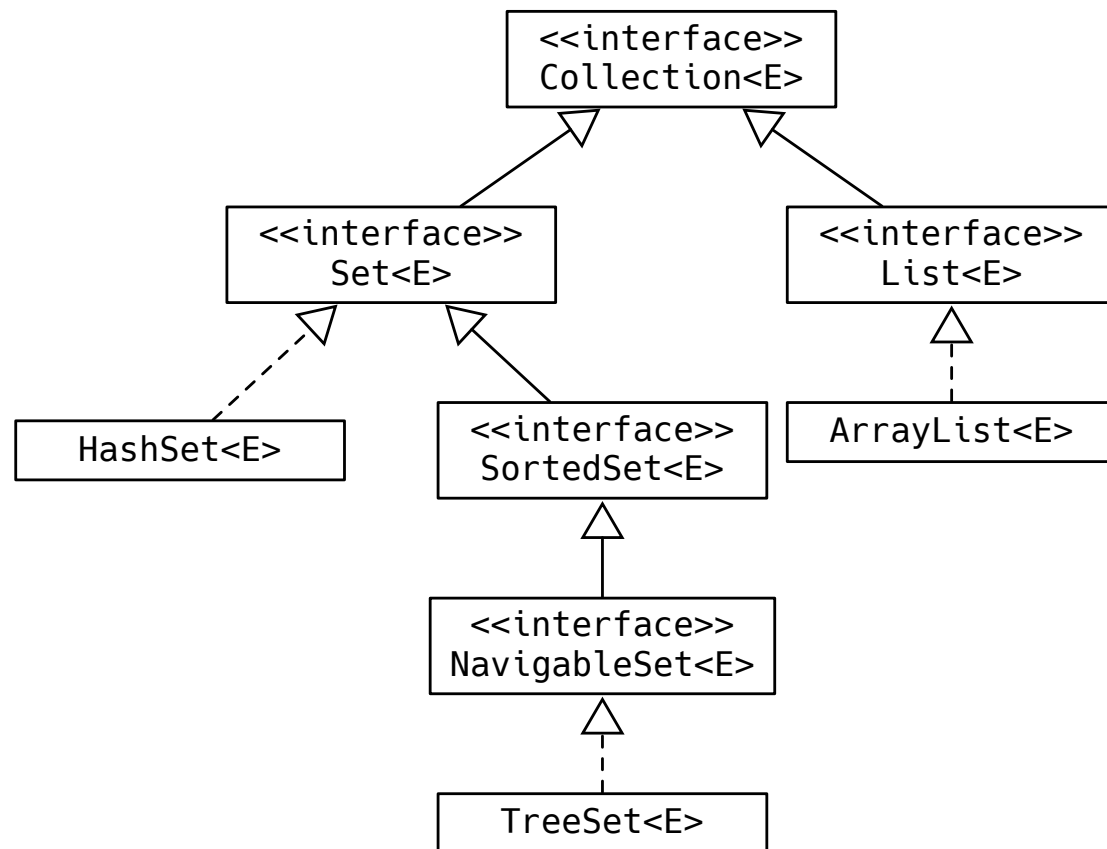


Collection



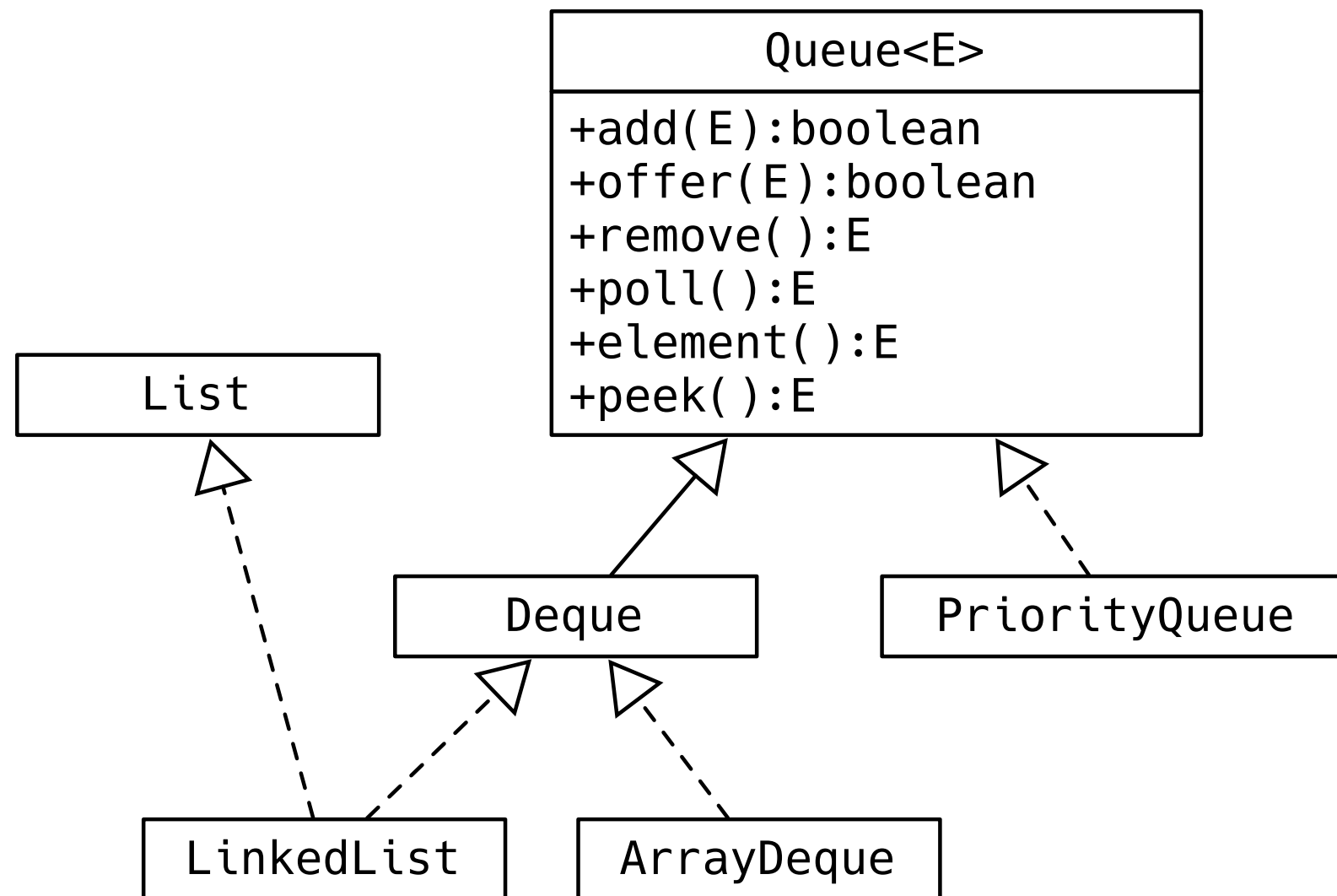
| | |
|----------------------|-------------|
| +add(E):boolean | 新增元素 |
| +clear():void | 清除所有元素 |
| +contains(E):boolean | 是否包含指定元素 |
| +isEmpty():boolean | 是否沒有元素 |
| +iterator():Iterator | 轉換為Iterator |
| +remove(E):boolean | 移除指定元素 |
| +size():int | 元素個數 |
| +toArray():Object[] | 轉換為陣列 |

List

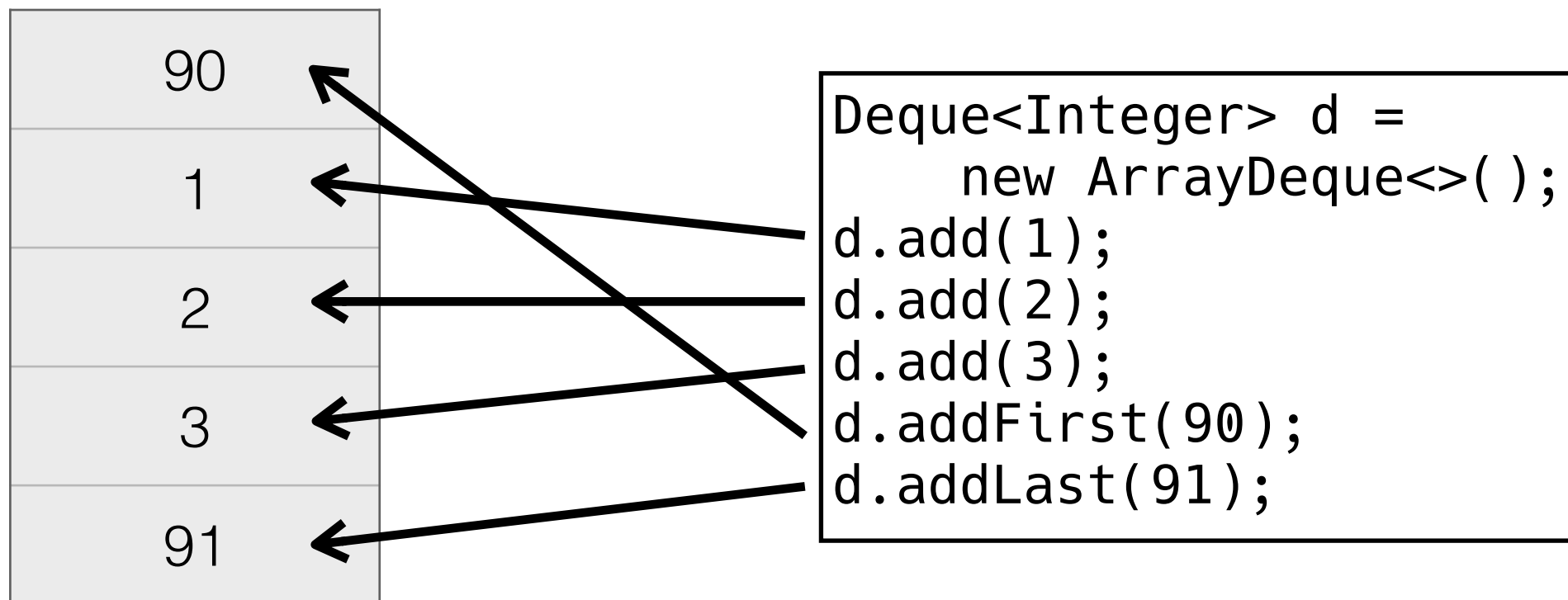


| | |
|---------------------|------------|
| +add(int,E):boolean | 新增元素到指定編號 |
| +get(int):E | 傳回指定編號元素 |
| +indexOf(E):int | 傳回元素編號 |
| +lastIndexOf(E):int | 傳回元素編號（反向） |
| +remove(int):E | 移除指定編號元素 |
| +set(int,Object):E | 設定指定編號元素 |

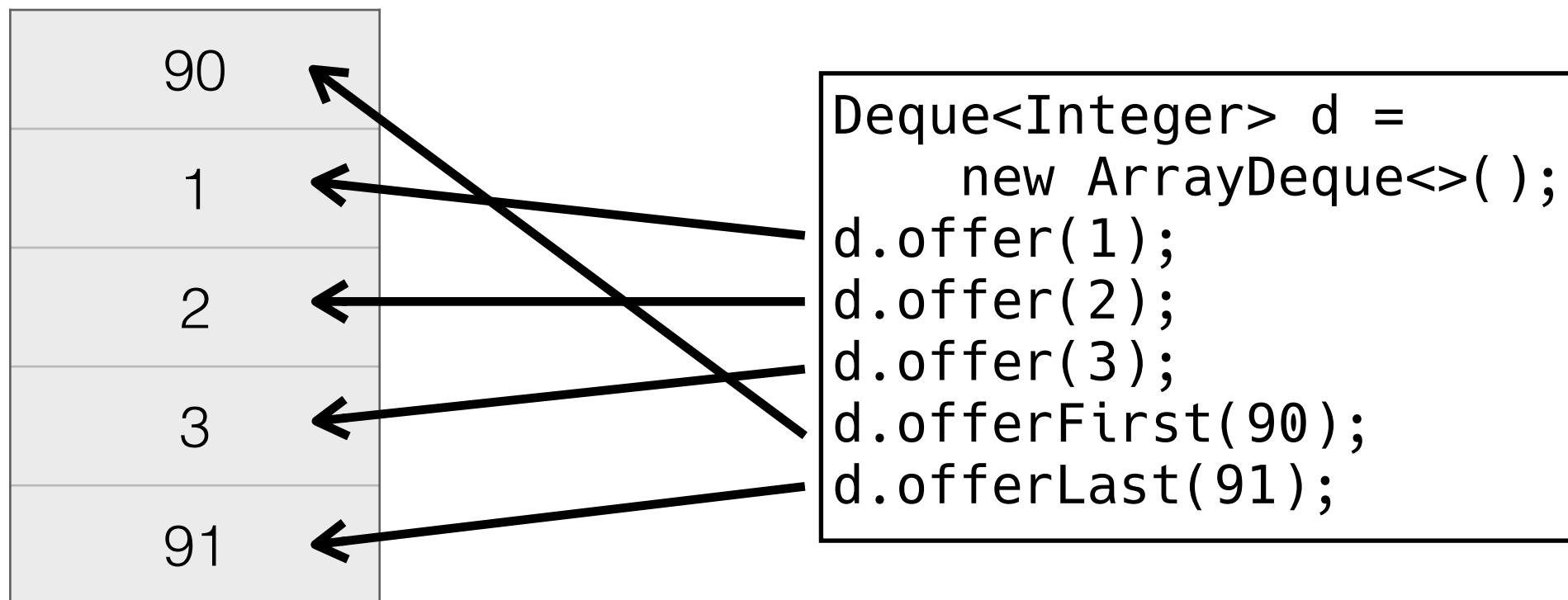
Queue



Deque add



Deque offer

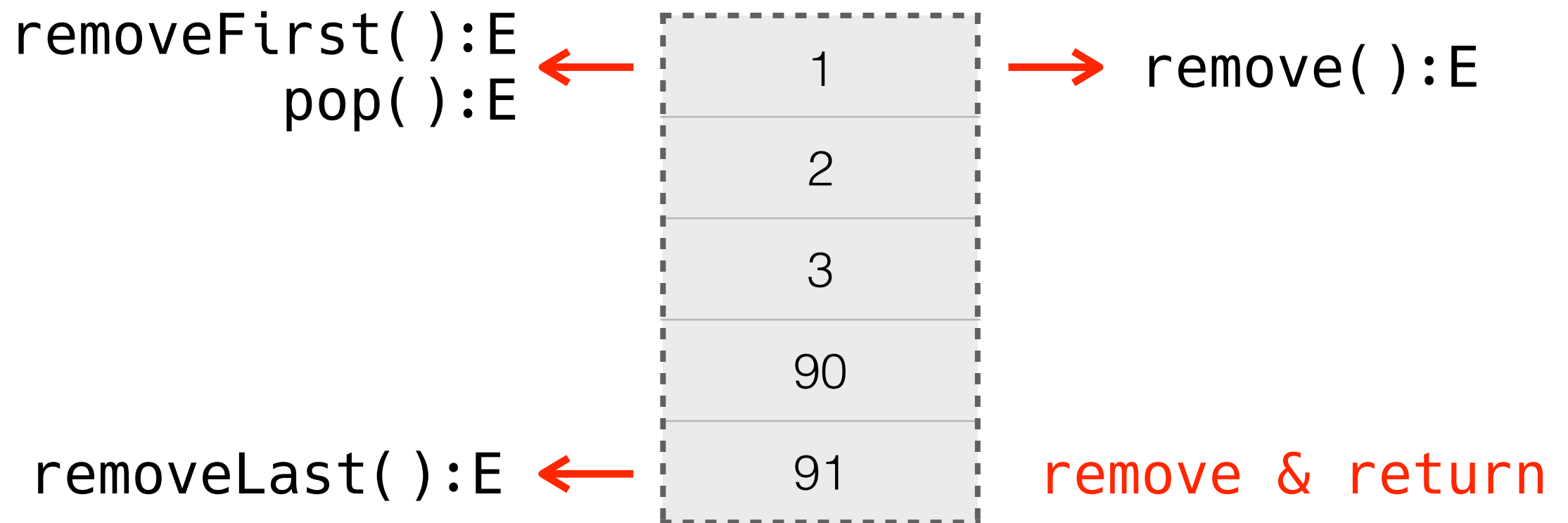


Deque push

| |
|----|
| 91 |
| 90 |
| 3 |
| 2 |
| 1 |

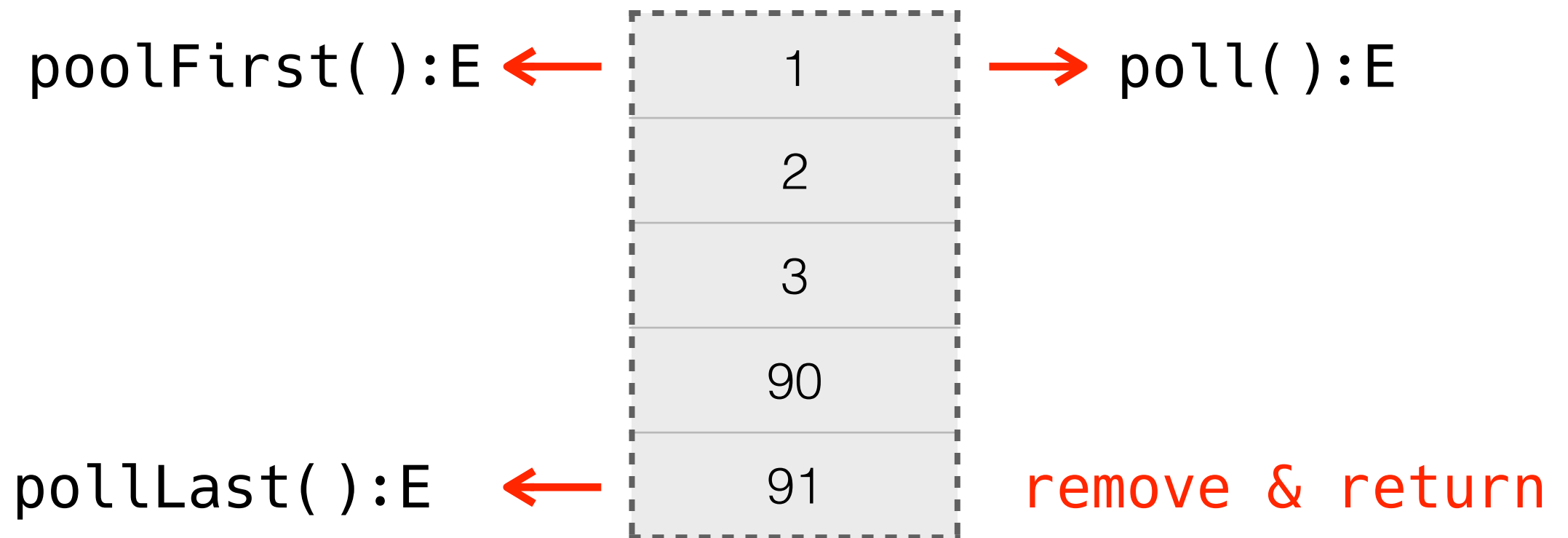
```
Deque<Integer> d =  
    new ArrayDeque<>( );  
d.push(1);  
d.push(2);  
d.push(3);  
d.push(90);  
d.push(91);
```

Deque remove & pop



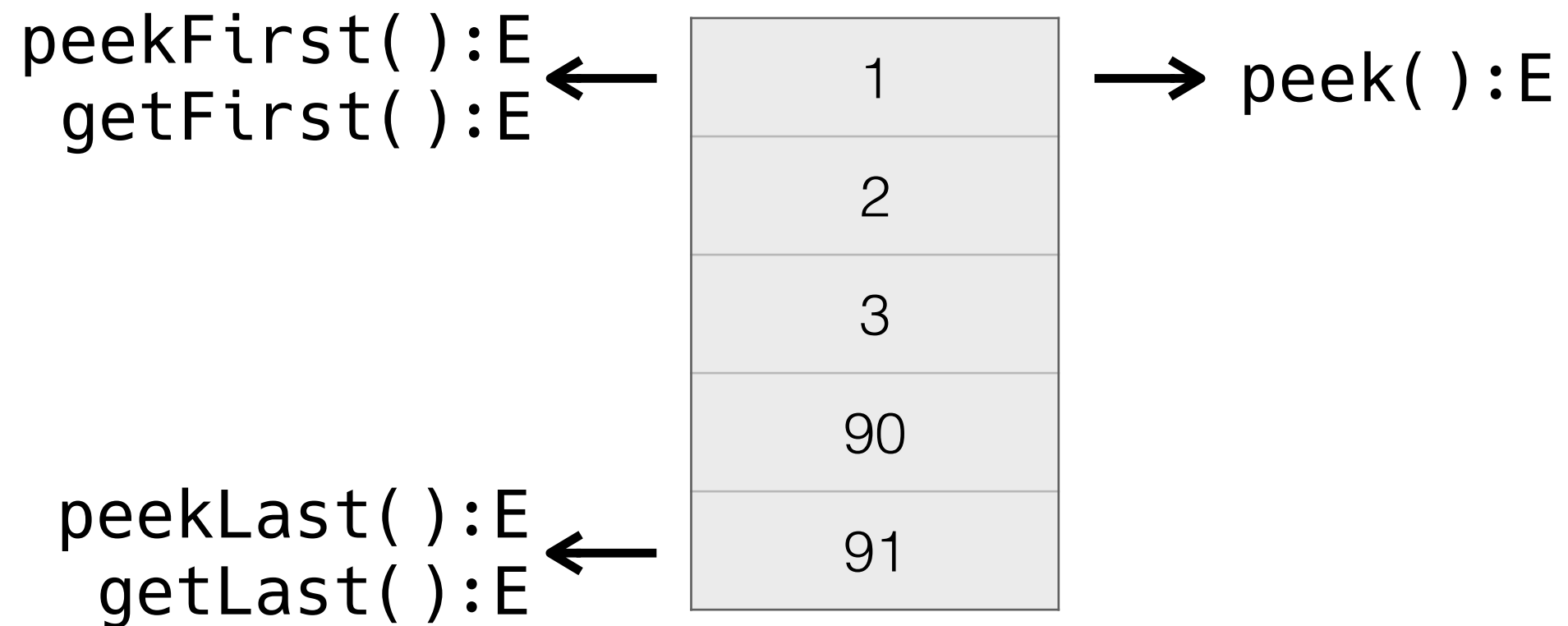
throws `NoSuchElementException` when queue is empty

Deque poll



return null when queue is empty

Deque peek & get



Arrays

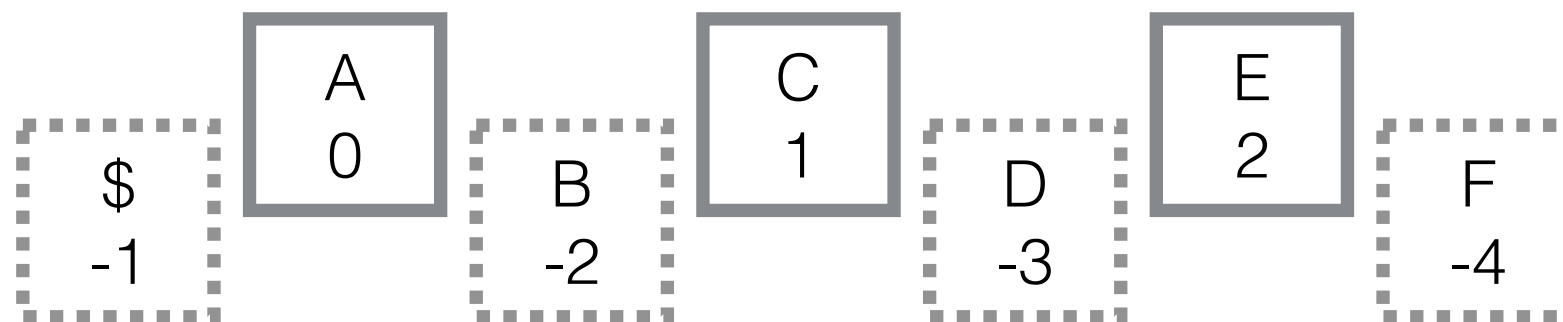
| Method | Comment |
|--|------------|
| <code>asList(T...):List<T></code> | 使用參數產生List |
| <code>binarySearch(type[] a, type key):int</code> | 搜尋 |
| <code>binarySearch(T[] a, T key, Comparator<? super T> c)</code> | 搜尋 |
| <code>sort(type[] a)</code> | 排序 |
| <code>sort(T[] a, Comparator<? super T> c)</code> | 排序 |

Arrays.binarySearch

| | | |
|---|---|---|
| A | C | E |
| 0 | 1 | 2 |

```
String[] names = {"A", "C", "E"};  
  
int indexA = Arrays.binarySearch(names, "A");  
int indexC = Arrays.binarySearch(names, "C");  
int indexE = Arrays.binarySearch(names, "E");
```

Arrays.binarySearch



```
String[] names = {"A", "C", "E"};

int index$ = Arrays.binarySearch(names, "$" );    // -1
int indexB = Arrays.binarySearch(names, "B" );    // -2
int indexD = Arrays.binarySearch(names, "D" );    // -3
int indexF = Arrays.binarySearch(names, "F" );    // -4
```