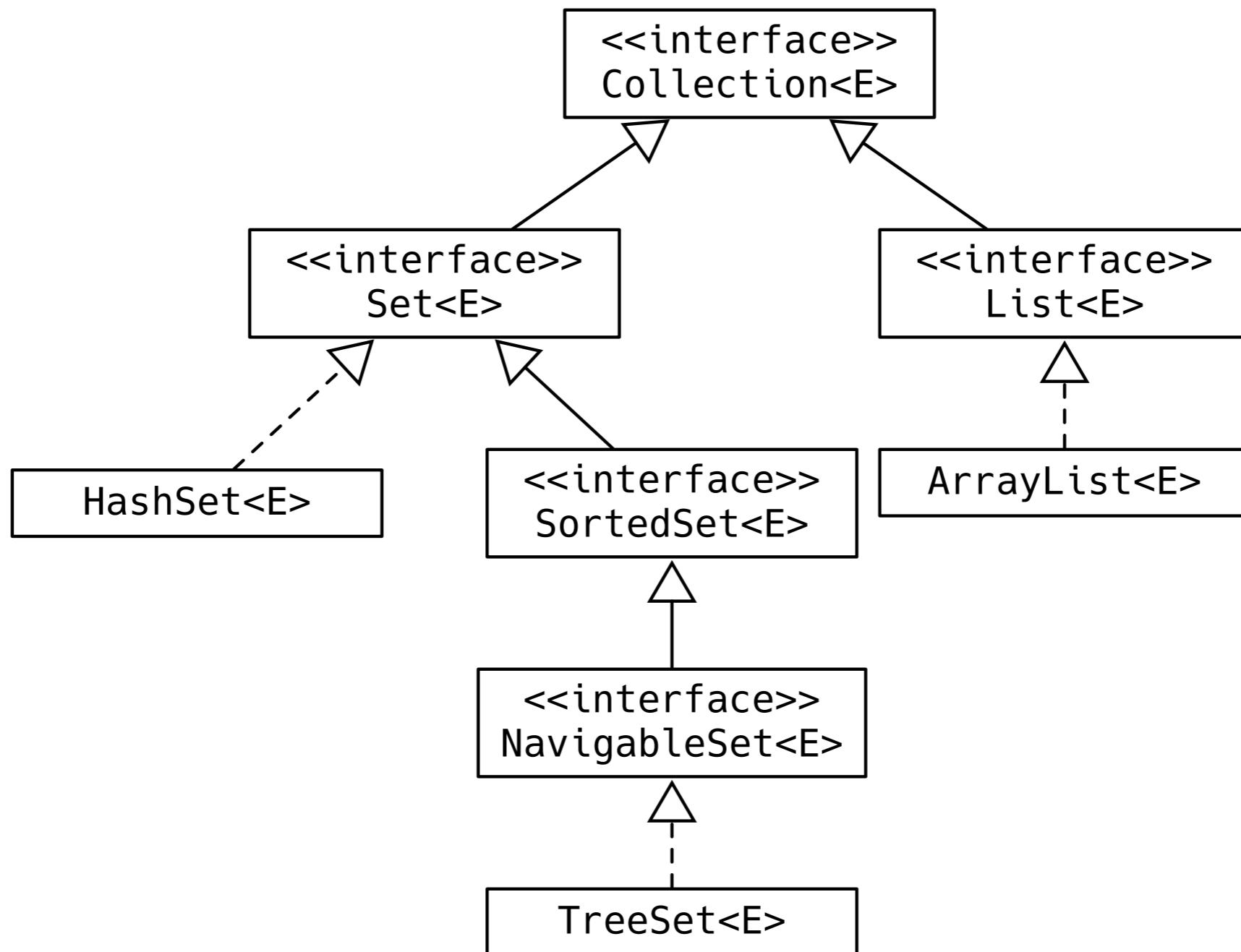
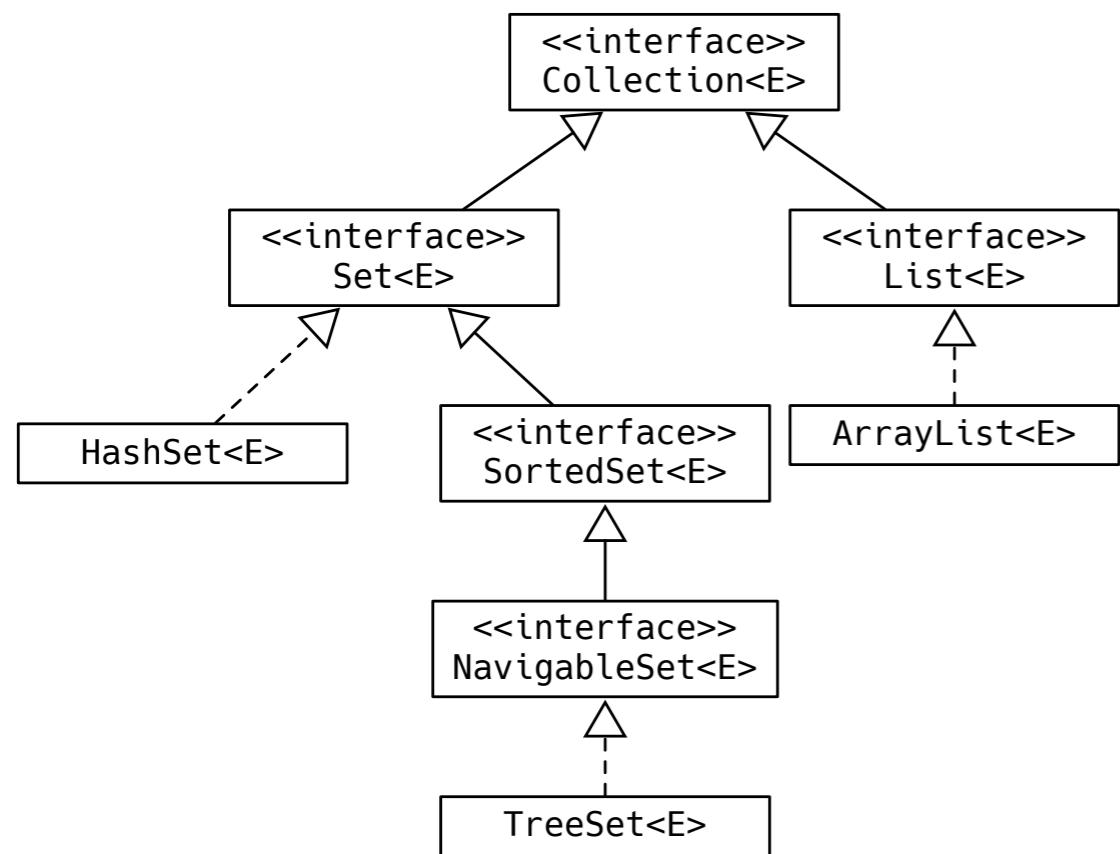


Collection Framework

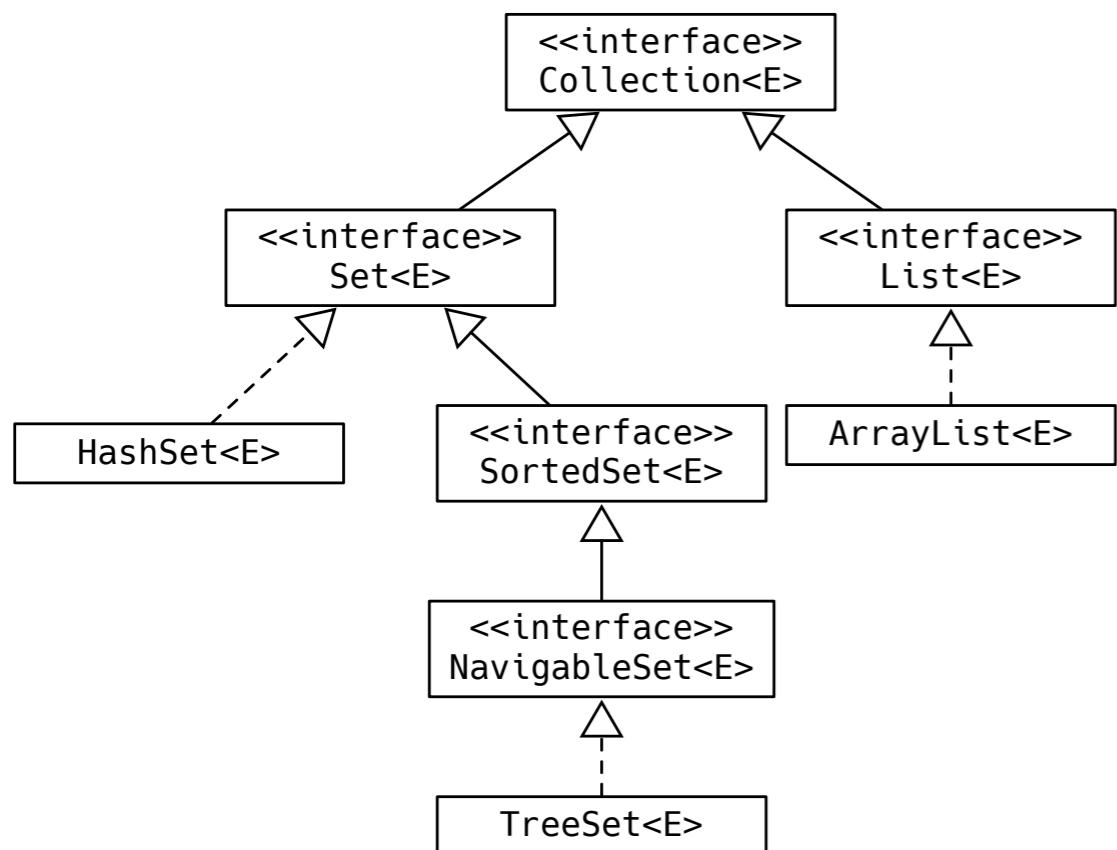


Collection



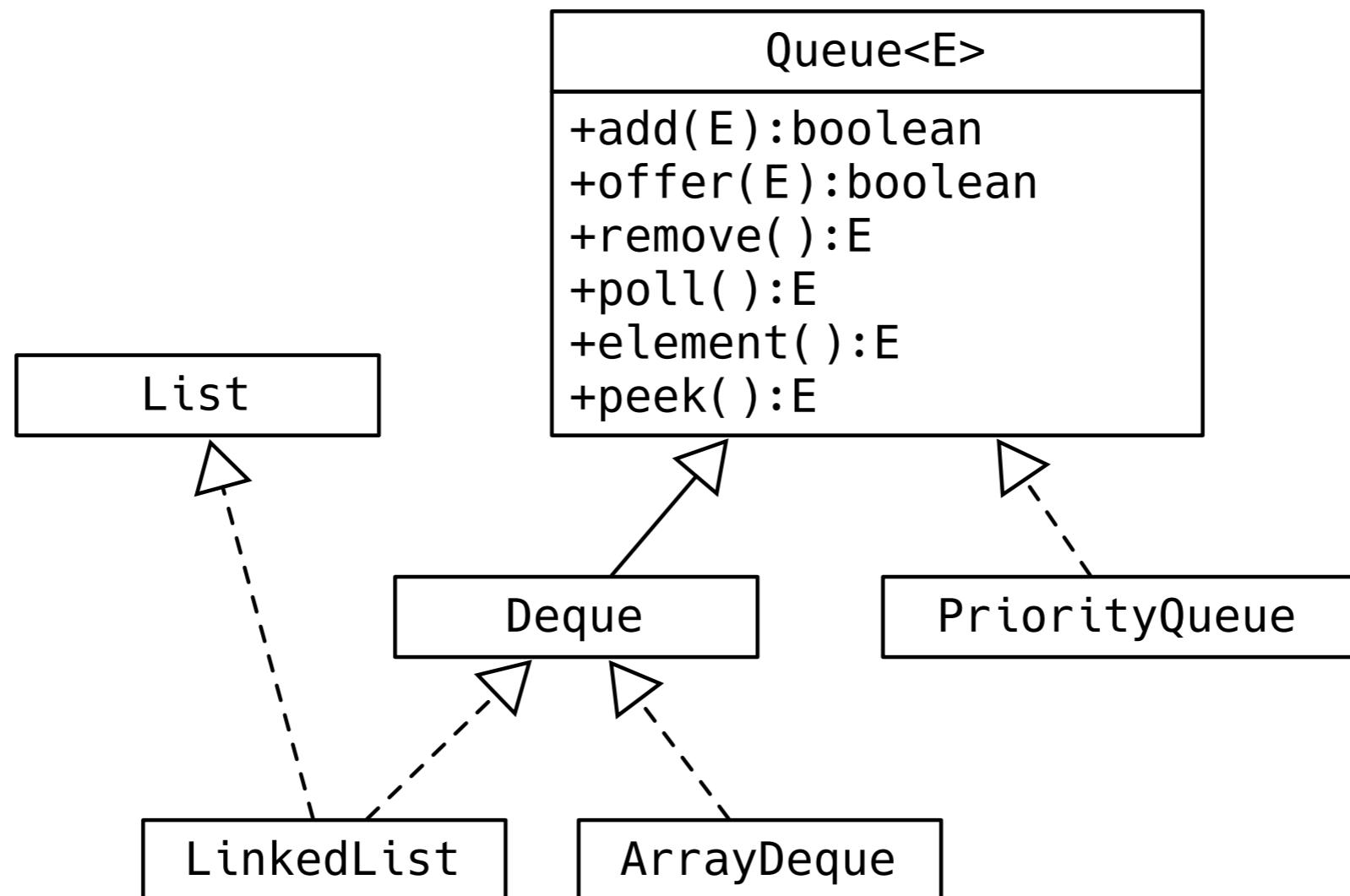
+add(E):boolean	新增元素
+clear():void	清除所有元素
+contains(E):boolean	是否包含指定元素
+isEmpty():boolean	是否沒有元素
+iterator():Iterator	轉換為Iterator
+remove(E):boolean	移除指定元素
+size():int	元素個數
+toArray():Object[]	轉換為陣列

List

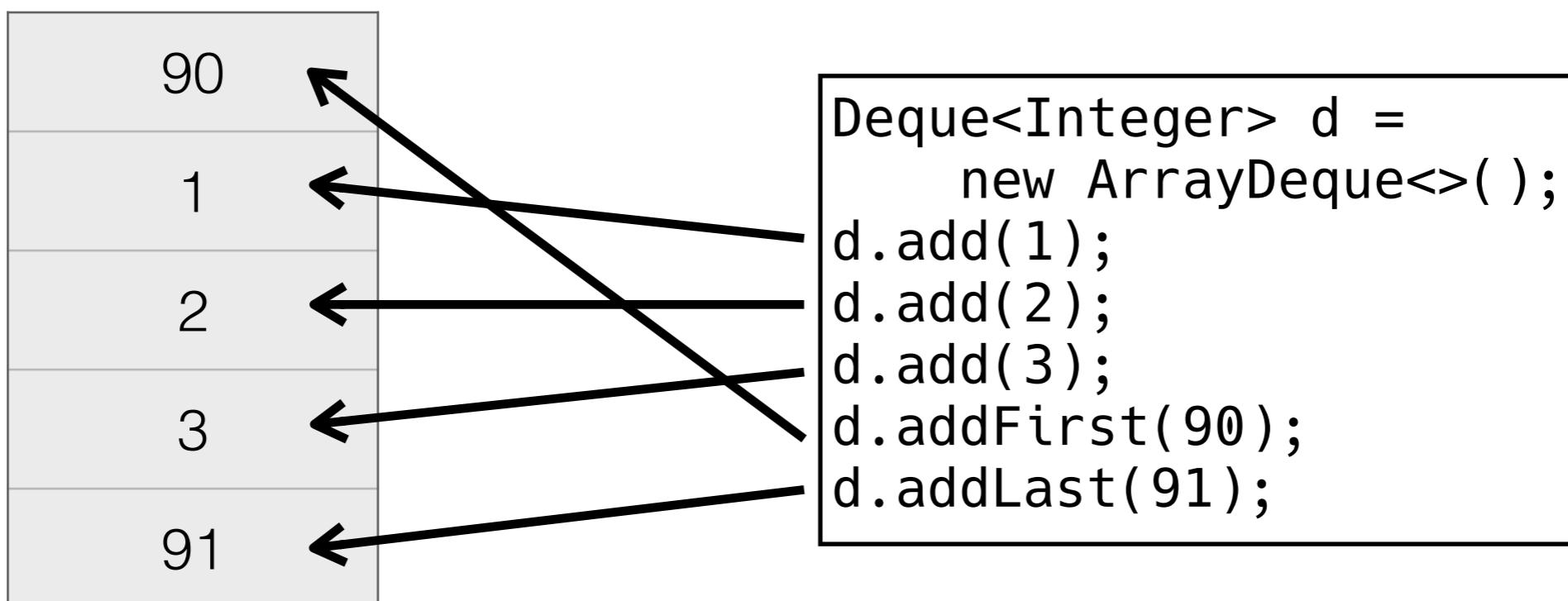


+add(int,E):boolean	新增元素到指定編號
+get(int):E	傳回指定編號元素
+indexOf(E):int	傳回元素編號
+lastIndexOf(E):int	傳回元素編號 (反向)
+remove(int):E	移除指定編號元素
+set(int,Object):E	設定指定編號元素

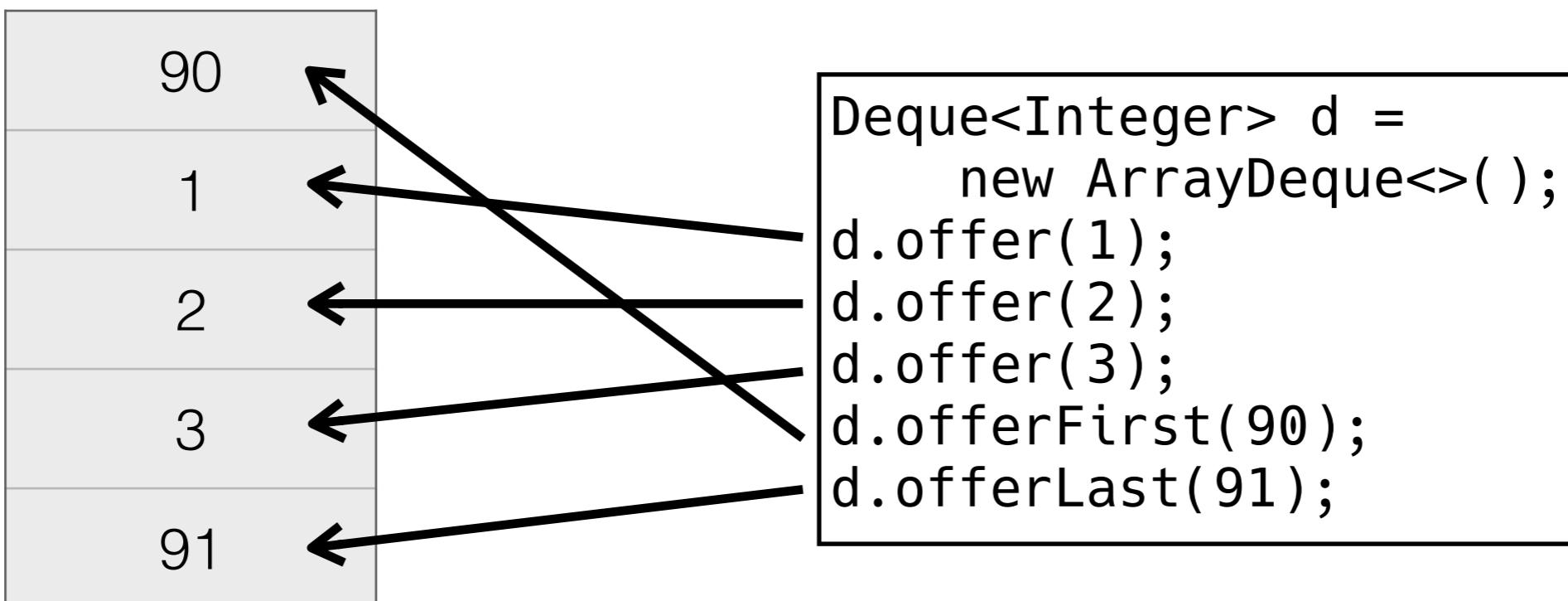
Queue



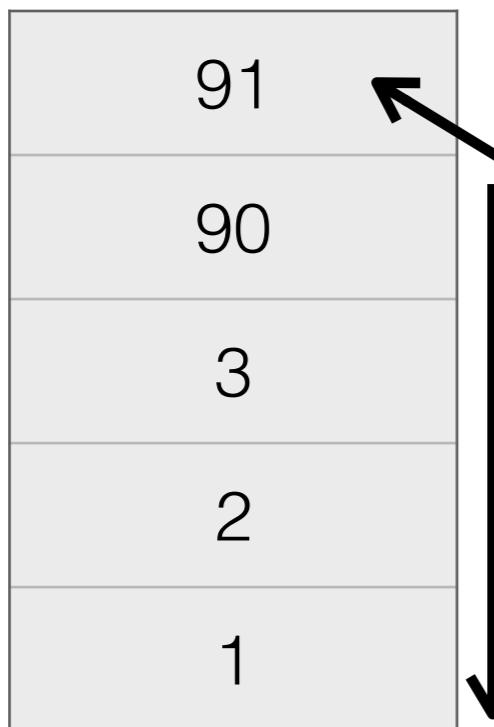
Deque add



Deque offer



Deque push



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

Deque remove & pop

removeFirst():E
pop():E



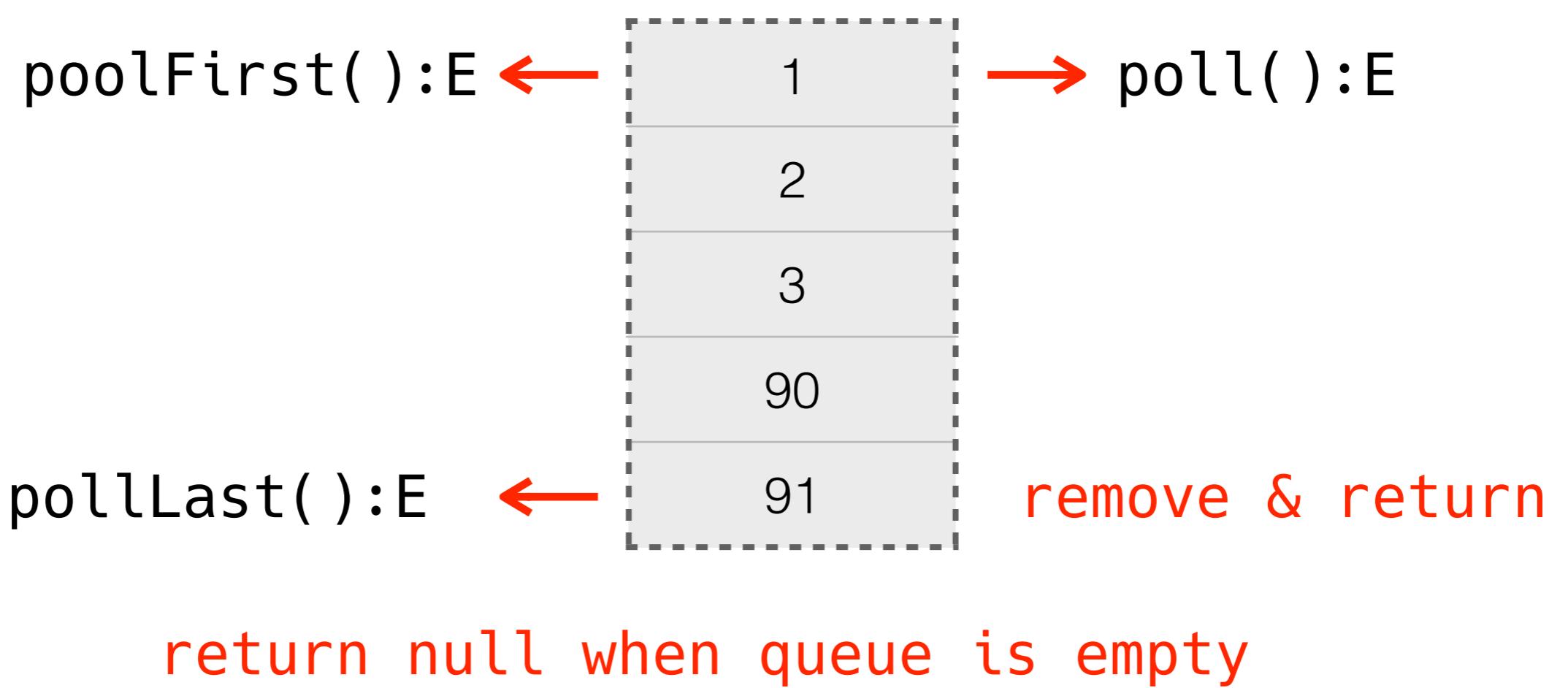
remove():E

removeLast():E

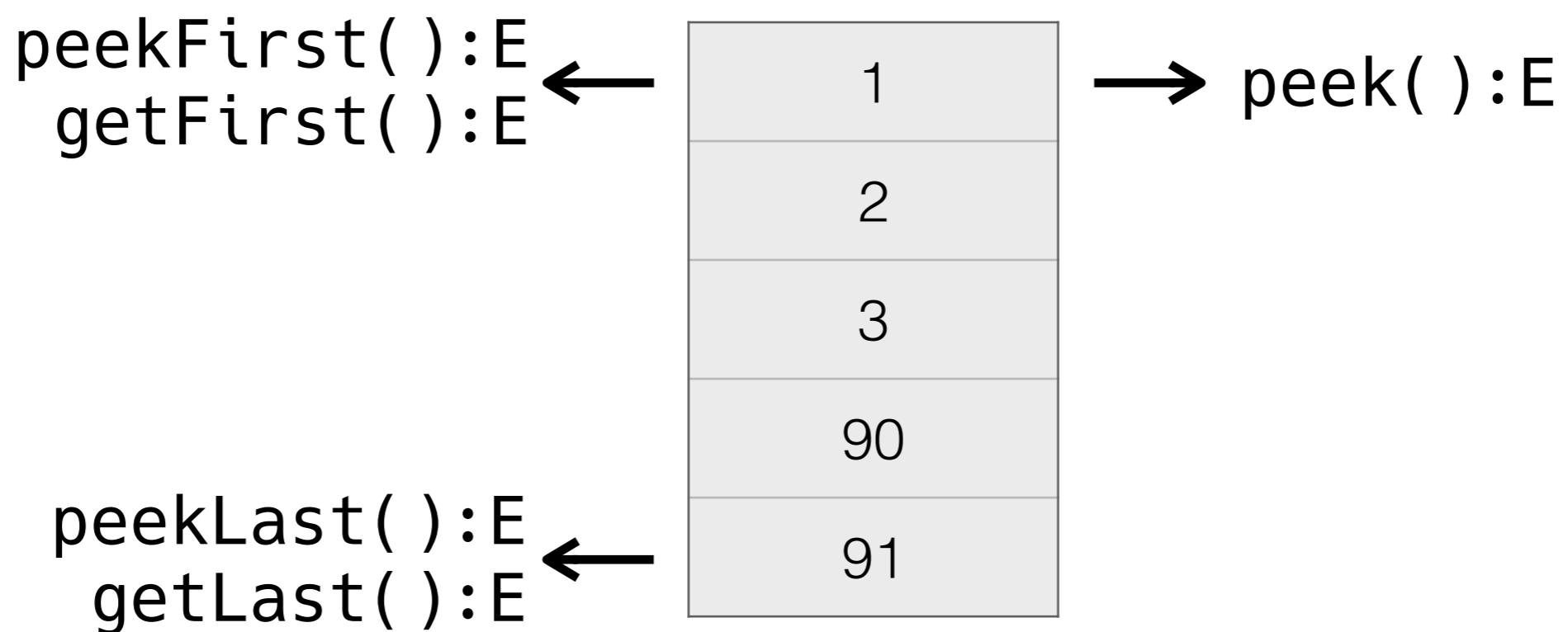
remove & return

throws NoSuchElementException when queue is empty

Deque poll



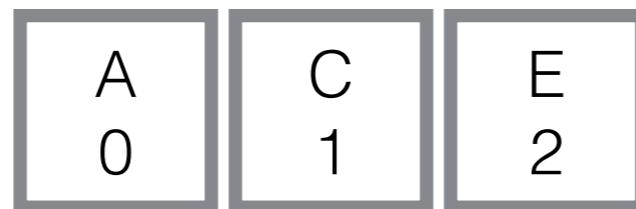
Deque peek & get



Arrays

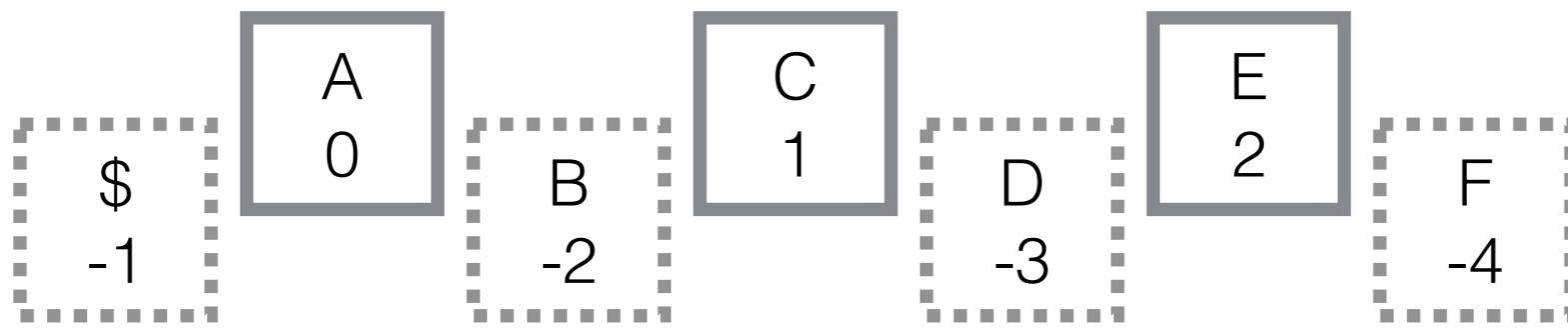
Method	Comment
asList(T...):List<T>	使用參數產生List
binarySearch(type[] a, type key):int	搜尋
binarySearch(T[] a, T key, Comparator<? super T> c)	搜尋
sort(type[] a)	排序
sort(T[] a, Comparator<? super T> c)	排序

Arrays.binarySearch



```
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
```