

Collections with Modification Order: Queues, Deques and Stacks



Richard Warburton

@richardwarburto | www.insightfullogic.com



Outline

First In, First Out

Highest Priority Out

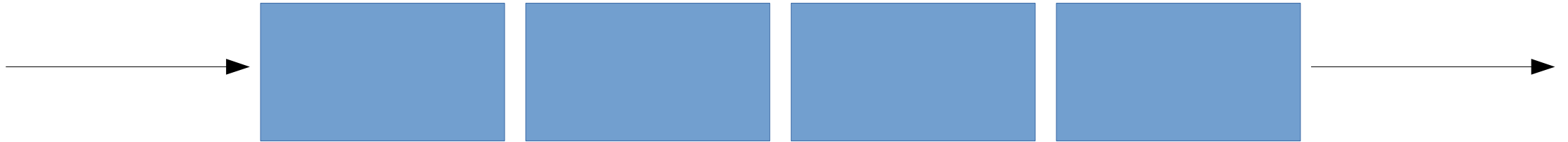
Last In, First Out

Implementations

First In, First Out

Queues

First In, First Out



```
boolean offer(E e)
```

```
boolean add(E e)
```

Adding elements

`offer` returns false if the queue is full

`add` exceptions if the queue is full.

`Collection` API requires that it only returns false if element is already present.

`E element()`

`E peek()`

Read without removing

`element` throws `Exception` when empty, `peek` returns `null`.

E remove()

E poll()

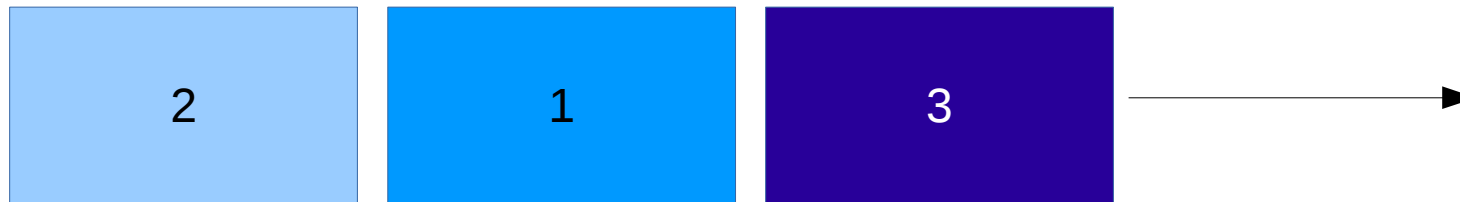
Removing and return the element

`remove` throws Exception when empty, `poll` returns null.

Highest Priority Out

Priority Queues

Highest Priority Out

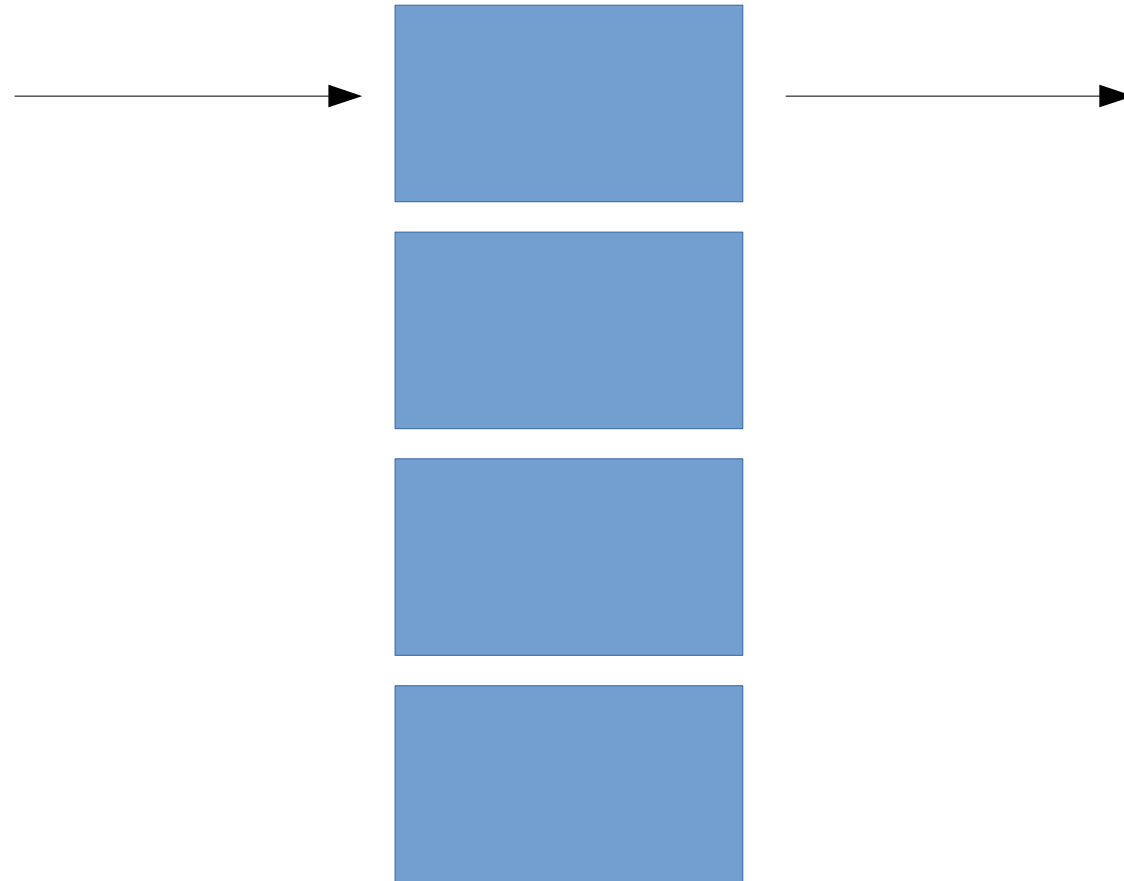


Priority really just defines
ordering

Last In, First Out

Stacks

Last In, First Out



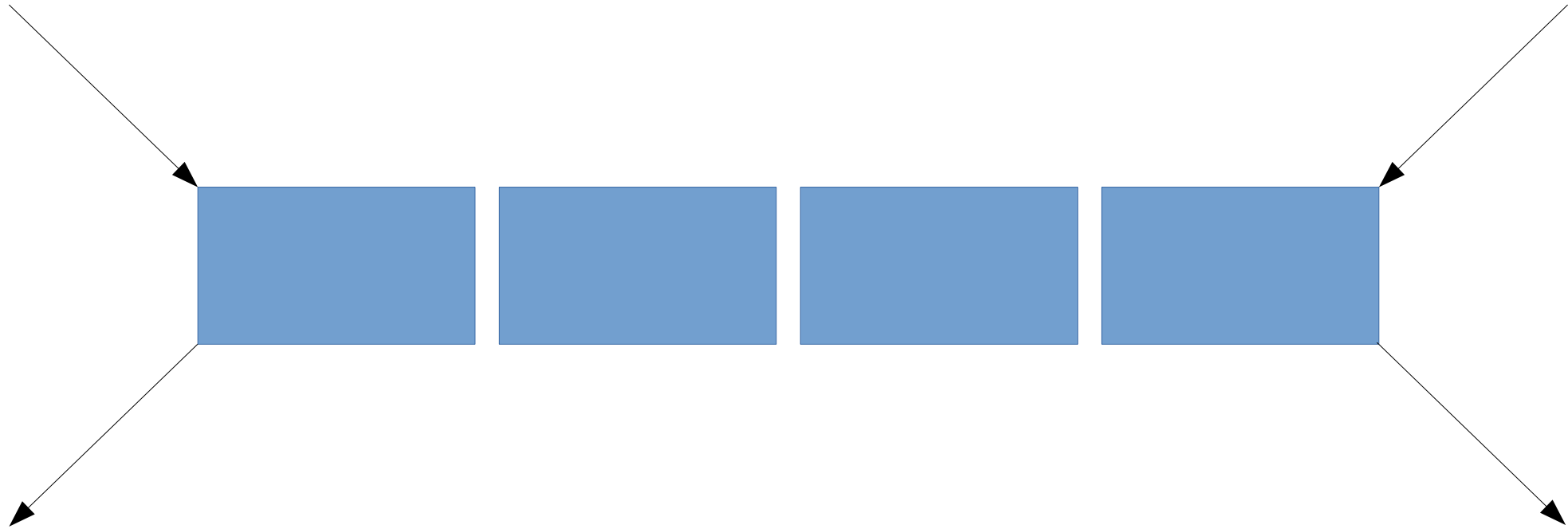


Double Ended Queues

Dequeues

Deque

Double Ended Queue



```
boolean offerFirst(E e)
```

```
boolean offerLast(E e)
```

```
void addFirst(E e)
```

```
void addLast(E e)
```

Adding elements

`offer*` returns false if the queue is full

`add*` inherited from `Collection`, throws an exception if the queue is full

E removeFirst()

E removeLast()

E pollFirst()

E pollLast()

Removing and return the element

`remove*` throws Exception when empty, `poll*` returns null.

E `getFirst()`

E `getLast()`

E `peekFirst()`

E `peekLast()`

Read without removing

`get*` throws Exception when empty, `peek*` returns null.

```
void push(E e)
```

```
E pop()
```

Semantic naming for Stacks/LIFO

Implementations

Performance Trade offs



There are multiple concurrent implementations of Queue that we won't be evaluating

Comparison Layout

ArrayDeque

- RingBuffer based implementation
- Constant time addition/removal
- Less Memory, Faster
- No random access

LinkedList

- Previously discussed in Lists
- Very seldom used as a Queue
- Has random access!
 - But its $O(N)$
- Allows `null` elements

Summary

Summary



Queue, Stack, Deque, Priority Queue

Try to avoid Stack and LinkedList

Queues are really useful!

MAPA-MUNDI HISTÓRICO.

