KEY POINTS

Queue:

- arbitrary objects
- enqueue + dequeue
- returns null for any exceptions
- FIFO

Array-based

- N: size of array
- f: index of the front element
- sz: number of stored elements.

$r = (f + sz) \bmod N$

first empty slot past the rear of the queue

Java: java.util.Queue

EXCEPTIONS	!EXCEPTIONS
offer(e)	add(e)
poll()	remove()
peek()	element()

NAME/DATE/SUBJECT

ADT - Queue

NOTES

A DSA that conforms to first in first out (FIFO) principles.

- It also stores arbitrary objects
- Has two main functions:
  - enqueue (void)
  - dequeue - returns object at front of the queue and removes it.
- Attempting to dequeue an empty queue will return null.

Auxiliary operations:

object first() int size() bool isEmpty()

Applications:

Direct Applications:
- Waiting lists, bureaucracy
- Access to shared resources

Indirect Applications:
- Auxiliary data structure for algorithms.
- Component of other data structures.

SUMMARY

The queue ADT will store any arbitrary objects in accordance with the first-in first-out principle - otherwise known as FIFO.

Round robin scheduling will enqueue tasks, allot them a time slice, and then requeue the task if it is incomplete.