[QUEUE AND STACK]

Marcus Francis TIPLER,
Computer Science at Cardiff University

Marcus Tipler

Cardiff University | ComSC – Abacws Building

Efficiency

circularly-queued data structure, specifically the `enqueue` and `dequeue` methods, demonstrates good practices in terms of efficiency, readability, and maintainability.

The ‘enqueue’ method efficiently handles queue overflow by doubling the size of the array when the current capacity is reached. This approach reduces the number of times the array needs to be resized, which improves performance.
The method also uses a ‘circular buffer’ or a ‘queue’ to store elements, allowing for efficient indexing and manipulation.

Good Practices

Good error handling because the ‘dequeue’ method properly checks if the queue is empty before attempting to remove an element. If the queue is empty, it throws a ‘IllegalStateException’, which helps prevent unexpected behaviour.
On top of that, the code is again readable as it is well organised and was supplied as commented with logical and understandable structure.
The code is also reusable as methods are used, meaning you can reimplement this code anywhere.
And finally: By setting the removed element's reference to null after dequeueing, the code ensures that memory is properly managed and prevents garbage collection issues (similar to Malloc in C).