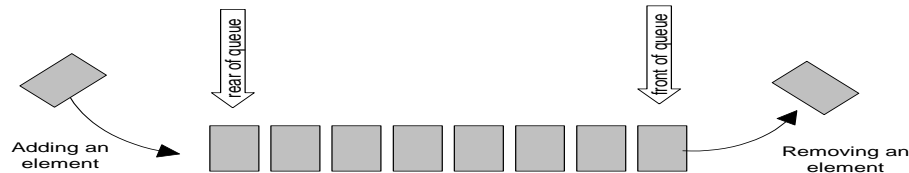

Data Structures

Lab7: Circular Queue

Queue is an ordered collection of items from which items may be deleted at one end (*front*) and into which items may be inserted at the other end (*rear*), [FIFO].



In this lab we going to implement a circular queue which is useful to reuses previously used array indexes.

CircularQueue
Int maxSize int front int rear int[] queArray
int size() boolean isEmpty() boolean isFull() enqueue (int ch) int dequeue () int front() display()

Variables:

Front: the first element in the sequence, where element access and deletion are restricted to it.

Rear: element insertion is restricted to the end of the sequence.

Methods:

enqueue(e): Insert element *e* at the rear .

dequeue(): Remove and return from the queue the object at the front; an error occurs if the queue is empty.

size(): Return the size of the queue.

isEmpty(): Return *true* or *false* indicating whether the queue is empty.

isFull(): Return *true* or *false* indicating whether the queue is empty.

front(): Return the front object in the queue; an error occurs if the queue is empty.