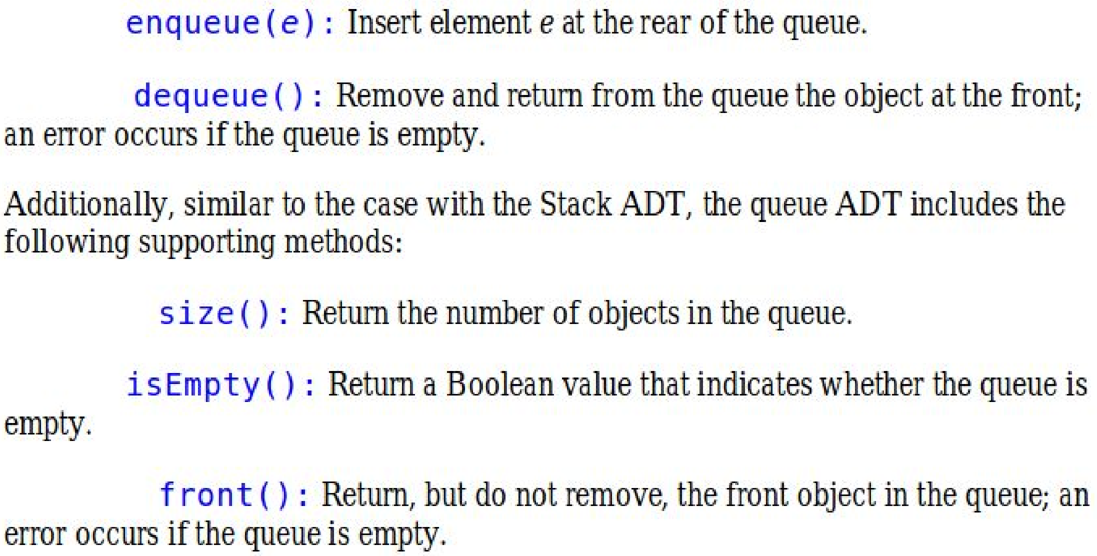
Renteria Magana Rayni Damian
17212178
Tarea 9. Colas

Another fundamental data structure is the queue. It is a close "cousin" of the stack, as a queue is a collection of objects that are inserted and removed according to the first-in first-out(FIFO) principle. That is, elements can be inserted at any time, but only the element that has been in the queue the longest can be re moved at any time.

Formally, the queue abstract data type defines a collection that keeps objects in a sequence, where element access and deletion are restricted to the first element in the sequence, which is called the front of the queue, and element insertion is restricted to the end of the sequence, which is called the rear of the queue. This restriction enforces the rule that items are inserted and deleted in a queue according to the first-in first-out (FIFO) principle.

The queue abstract data type (ADT) supports the following two fundamental methods:



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

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

Additionally, similar to the case with the Stack ADT, the queue ADT includes the following supporting methods:

`size()`: Return the number of objects in the queue.

`isEmpty()`: Return a Boolean value that indicates whether the queue is empty.

`front()`: Return, but do not remove, the front object in the queue; an error occurs if the queue is empty.

Works Cited

Goodrich, Michael T., et al. *Data Structures and Algortihms in Python*. Wiley, 2013.

Goodrich, Michael T., et al. *Data Structures and Algorithms in Java*. John Wiley &amp; Sons, 2015.