

## Stacks and Queues

Stacks and queues are two commonly used abstract data types that allow insertion and removal of items at the ends only.

### Stacks

---

A stack is a collection of items that use the “last in, first out” (LIFO) method of retrieving objects. In order to visualize a stack, think of a stack of books. To add to the stack, you add books to the *top*. If you wanted to remove some books, you would remove the ones that were at the top of the stack. As you can see, the books are removed in the order that is opposite from the order in which they were added.

The Stack class uses an array list to implement a stack.

#### Basic Code for Stacks

- `push(Object obj)` – Adds the given object to the top of the stack.
- `pop()` – Removes and returns the object on the top of the stack.
- `peek()` – Returns the object on the top of the stack without removing it.
- `search()` – Returns the 1-based where an object is located within the stack.
- `empty()` – Tests to see if the stack is empty.

### Queues

---

A queue is similar to a stack except that it follows the “first in, first out” (FIFO) method of retrieving objects – objects are added at the *tail* and are removed at the *head*. A queue is like a line of people at a restaurant. When customers want to order something, they need to wait in line. The people who entered the line first will be the first to order and the people at the end of the line will order last.

Not all queues follow the “first in, first out” method. One example of this is a priority queue, which orders elements according to a given comparator.

The Queue interface uses a linked list to implement a queue.

#### Basic Code for Queues

- `add(Object obj)` - Adds the given object to the tail of the queue. Returns true if it was successful or throws an `IllegalStateException` if there is no more room in the queue.
- `element()` – Returns the object at the head of the queue without removing it.
- `offer(Object obj)` – Adds the given object to the tail of the queue. Returns true if it was successful or false if it wasn't.
- `peek()` – Returns the object at the head of the queue or returns null if the queue is empty.
- `poll()` – Removes and returns the object at the head of the queue or returns null if the queue is empty.
- `remove()` – Removes and returns the object at the head of the queue.