**Section 5: Stacks**

**Stack**

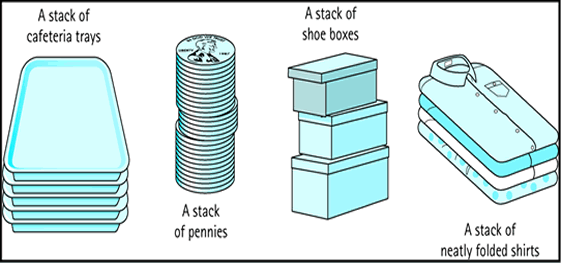
* ADT
* **LIFO** – Last In, First Out
* Ideal backing data structure is a **Linked List**

[**https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Stack.html**](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Stack.html)

**Operations**

* **push** – adds an item to the top of the stack
* **pop** – removes the top item on the stack
* **peek** – gets the top item on the stack without popping it
* **empty** – tests if the stack is empty
* **search** – returns the 1-based position where an object is on this stack

**Time Complexity**

* **O(1)** for push, pop and peek, when using a **Linked List**
* If you use an array, then push is **O(n)**, because the array may have to be resized
* If you know the maximum number of items, that will ever be on the stack, an array can be a good choice
* If memory is tight, an array might be a good choice
* **Linked List** is ideal