1. **Arrays**

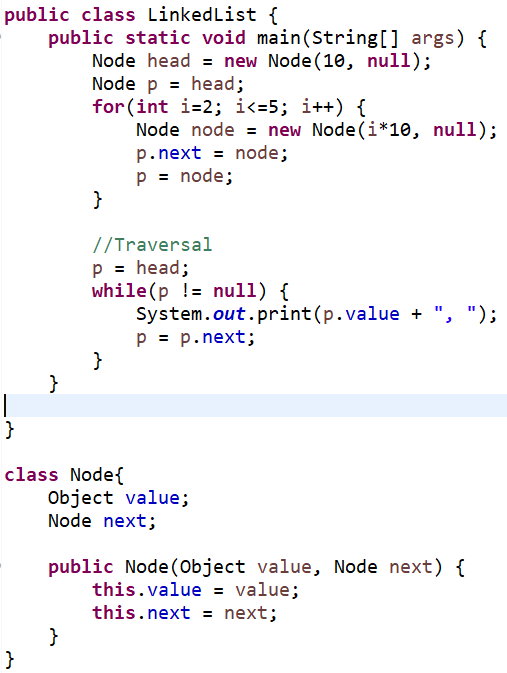
Static Array

* 1-D Array
* 2-D Array

1. **ArrayList**

Dynamic Array

1. **LinkedList**



1. **Stack**

Last In First Out

Basic Operations

* **push()**to insert an element into the stack
* **pop()**to remove an element from the stack
* **peek()**Returns the top element of the stack.
* **isEmpty()**returns true if stack is empty else false.
* **isFull()**returns true if the stack is full else false.

**Push Operation:** If (top == capacity-1) then throw Stack Overflow exception. Else increase top and add the element.

**Pop Operation:** If(top == -1) then throw Stack Underflow exception. Else return top element and reduce top by 1.

**Peek Operation:** Return top element if stack is not empty. Else return the top element.

**isEmpty Operation:** If (top == -1) then return true else false.

**isFull Operation:** If (top == capacity-1) then return true else false.

1. **Queue**

FIFO (First In First Out)

Basic Operations:

* **Enqueue:** Adds (or stores) an element to the end of the queue..
* **Dequeue:** Removal of elements from the queue.
* **Peek or front:** Acquires the data element available at the front node of the queue without deleting it.
* **rear:** This operation returns the element at the rear end without removing it.
* **isFull**: Validates if the queue is full.
* **isEmpty**: Checks if the queue is empty.

**Enqueue Operation**: If rear==capacity then throw Overflow exception. Else add element to the rear then increase rear by 1.

**Dequeue Operation**: If queue is empty then throw Underflow exception. Else move the front to the next available value.

**IsFull Operation:** If rear == capacity then return true else false.

**IsEmpty Operation:** If front == -1 then return true else false.