**LList<T> Class**

**Definition:**

Namespace: ARLinkedList

Summary: LList is a generic class that allows you to make a list that has links to both previous and next nodes on the list. This is called a doubly linked list

*Public class LList<T>*

**Type Parameters**:

*T* Represents the element type of the list

**Nested Class:**

[Public class LListNode](#NodeClass)

**Remarks:**

LList<T> is a generic doubly linked list that is made up of LListNode elements.

LList<T> is not a circular linked list, therefore, the first node has a null in the previous node reference and the last node as a null value in the next node reference

**Constructors:**

|  |  |
| --- | --- |
| LList() | Default constructor that creates an empty list |

**Private Variables:**

|  |  |
| --- | --- |
| first | Holds the first node of the list |
| last | Holds the last node of the list |
| current | Holds the current node when iterating through the list |
| count | Holds the number of elements in the list |

**Properties:**

|  |  |
| --- | --- |
| Count | Gets the number of elements in the list |

**Methods:**

|  |  |
| --- | --- |
| Find(T findItem): | Searches the list linearly starting with the first node to find the specified data listed as parameter. If the data is not found, a default value of parameter datatype is returned. |
| AddFirst(T firstItem) | Adds firstItem parameter data to the list as the first element |
| AddBack(T lastItem) | Adds lastItem parameter data to the list as the last element |
| Delete (T delItem) | Searches the list starting with the first node and deletes that element if found |
| GetEnumerator() | Allows for iteration through the loop with a foreach loop |
| Min() | Finds and returns the Minimum value contained in the list |
| Max() | Finds and returns the Maximum value contained in the list |

See Also:

[LListNode class](#NodeClass)

**LListNode class**

**Definition**:

*Public class LListNode: Comparer<T>*

**Type Parameters:**

*T* Represents the type parameter of the LList

*Implements: Comparer<T>*

**Remarks:**

LListNode class is

**Constructors:**

|  |  |
| --- | --- |
| LListNode() | Default constructor that creates an empty node |
| LListNoce(T datatype) | Constructor that creates an node with the date specified |

**Properties:**

|  |  |
| --- | --- |
| Item | Gets the element data |
| Next | Gets the next node element in the list |
| Prev | Gets the previous node element in the list |

**Methods:**

|  |  |
| --- | --- |
| Compare(T x, T y) | Compares one node to another. |
| ToString() | Allows fo the node element to be written in string format |

See Also:

[LList<T> class](#_top)