

## My Project

Generated by Doxygen 1.8.17



---

<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 LinkedList< T > Class Template Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.1.2 Member Function Documentation . . . . .	5
3.1.2.1 insertFront() . . . . .	5
3.1.2.2 insertRear() . . . . .	6
3.1.2.3 main() . . . . .	6
3.1.2.4 removeFront() . . . . .	6
3.1.2.5 removeRear() . . . . .	6
3.1.3 Member Data Documentation . . . . .	6
3.1.3.1 front . . . . .	7
3.1.3.2 rear . . . . .	7
<b>4 File Documentation</b>	<b>9</b>
4.1 LinkedList.java File Reference . . . . .	9
<b>Index</b>	<b>11</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">LinkedList&lt; T &gt;</a> . . . . .	5
---	---



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">LinkedList.java</a> . . . . .	9
---	---





## Chapter 3

# Class Documentation

### 3.1 `LinkedList< T >` Class Template Reference

#### Public Member Functions

- void `insertFront` (T item)
- void `insertRear` (T item)
- void `removeFront` ()
- void `removeRear` ()

#### Static Public Member Functions

- static void `main` (String a[])

#### Private Attributes

- `Node< T >` `front`
- `Node< T >` `rear`

#### 3.1.1 Detailed Description

This class implements the data structure `LinkedList`, which consists of sequential nodes containing information, and links to the previous and the next nodes

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 `insertFront()`

```
void LinkedList< T >.insertFront (  
    T item ) [inline]
```

**Parameters**

<i>item</i>	The element to be inserted at the front of the <a href="#">LinkedList</a> Inserts the element to the front of the <a href="#">LinkedList</a> , links the old front node to the new one and updates the front node of <a href="#">LinkedList</a> Prints the element being added to the front
-------------	---

add element at the beginning of the queue

**3.1.2.2 insertRear()**

```
void LinkedList< T >.insertRear (
    T item ) [inline]
```

**Parameters**

<i>item</i>	The element to be inserted at the rear of the <a href="#">LinkedList</a> Inserts the element to the rear of the <a href="#">LinkedList</a> , links the old rear node to the new one and updates the rear node of <a href="#">LinkedList</a> Prints the element being added to the rear
-------------	--

add element at the end of the queue

**3.1.2.3 main()**

```
static void LinkedList< T >.main (
    String a[] ) [inline], [static]
```

The main driver code for building the [LinkedList](#)

**3.1.2.4 removeFront()**

```
void LinkedList< T >.removeFront ( ) [inline]
```

Removes the element at the front of the [LinkedList](#), links the new front node to null and updates the front node of [LinkedList](#) to the second node in the old List Prints the element being removed from the front remove an item from the beginning of the queue

**3.1.2.5 removeRear()**

```
void LinkedList< T >.removeRear ( ) [inline]
```

Removes the element at the rear of the [LinkedList](#), links the new rear node to null and updates the rear node of [LinkedList](#) to the second last node in the old List Prints the element being removed from the rear remove an item from the beginning of the queue

**3.1.3 Member Data Documentation**

### 3.1.3.1 front

```
Node<T> LinkedList< T >.front [private]
```

### 3.1.3.2 rear

```
Node<T> LinkedList< T >.rear [private]
```

The documentation for this class was generated from the following file:

- [LinkedList.java](#)



## Chapter 4

# File Documentation

### 4.1 LinkedList.java File Reference

#### Classes

- class [LinkedList< T >](#)
- class **Node< T >**



# Index

front  
    LinkedList< T >, [6](#)

insertFront  
    LinkedList< T >, [5](#)

insertRear  
    LinkedList< T >, [6](#)

LinkedList< T >, [5](#)  
    front, [6](#)  
    insertFront, [5](#)  
    insertRear, [6](#)  
    main, [6](#)  
    rear, [7](#)  
    removeFront, [6](#)  
    removeRear, [6](#)  
LinkedList.java, [9](#)

main  
    LinkedList< T >, [6](#)

rear  
    LinkedList< T >, [7](#)

removeFront  
    LinkedList< T >, [6](#)

removeRear  
    LinkedList< T >, [6](#)