LinkedList<T>.AddAfter Method

Namespace: System.Collections.Generic

Assemblies: System.Collections.dll, System.dll, netstandard.dll

Adds a new node or value after an existing node in the LinkedList<T>.

In this article

Definition

Overloads

AddAfter(LinkedListNode<T>, LinkedListNode<T>)

AddAfter(LinkedListNode<T>, T)

Applies to

Overloads

AddAfter(LinkedListNode<T>, Linked ListNode<T>)	Adds the specified new node after the specified existing node in the LinkedList<T>.
AddAfter(LinkedListNode<T>, T)	Adds a new node containing the specified value after the specified existing node in the LinkedList<T>.

AddAfter(LinkedListNode<T>, LinkedListNode<T>)

Adds the specified new node after the specified existing node in the [LinkedList<T>](#).

C#	🗐 Copy

```csharp
public void AddAfter (System.Collections.Generic.LinkedListNode<T> node,
    System.Collections.Generic.LinkedListNode<T> newNode);
```

Parameters

node   LinkedListNode<T>

The LinkedListNode<T> after which to insert `newNode`.

newNode   LinkedListNode<T>

The new LinkedListNode<T> to add to the LinkedList<T>.

Exceptions

ArgumentNullException

`node` is null.

-or-

`newNode` is null.

InvalidOperationException

`node` is not in the current LinkedList<T>.

-or-

`newNode` belongs to another LinkedList<T>.

Examples

For an example that includes this method, see the [LinkedList<T>](#) class.

Remarks

[LinkedList<T>](#) accepts `null` as a valid [Value](#) for reference types and allows duplicate values.

This method is an O(1) operation.

See also

- [AddBefore](#)
- [AddFirst](#)
- [AddLast](#)
- [Remove](#)

AddAfter(LinkedListNode<T>, T)

Adds a new node containing the specified value after the specified existing node in the [LinkedList<T>](#).

C#	Copy

```csharp
public System.Collections.Generic.LinkedListNode<T> AddAfter (System.Collections.Generic.LinkedListNode<T>
node, T value);
```

Parameters

node   LinkedListNode<T>

The LinkedListNode<T> after which to insert a new LinkedListNode<T> containing `value`.

value   T

The value to add to the LinkedList<T>.

Returns

LinkedListNode<T>

The new LinkedListNode<T> containing `value`.

Exceptions

ArgumentNullException

`node` is `null`.

InvalidOperationException

`node` is not in the current LinkedList<T>.

Examples

For an example that includes this method, see the LinkedList<T> class.

Remarks

LinkedList<T> accepts `null` as a valid Value for reference types and allows duplicate values.

This method is an O(1) operation.

See also

- AddBefore
- AddFirst
- AddLast
- Remove

Applies to

.NET Core

3.0, 2.2, 2.1, 2.0, 1.1, 1.0

.NET Framework

4.8, 4.7.2, 4.7.1, 4.7, 4.6.2, 4.6.1, 4.6, 4.5.2, 4.5.1, 4.5, 4.0, 3.5, 3.0, 2.0

.NET Standard

2.1, 2.0, 1.6, 1.4, 1.3, 1.2, 1.1, 1.0

UWP

10.0

Xamarin.Android

7.1

Xamarin.iOS

10.8

Xamarin.Mac

3.0

Is this page helpful?

👍 Yes　👎 No