Dynamic Array Implementation

1

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 DynamicArray< T > Class Template Reference

Public Member Functions

- DynamicArray ()

  Default Constructor.
- DynamicArray (int size)

  Constructor with an initial size.
- DynamicArray (int size, T initialData)

  Constructor with and initial size and initializing value.
- ∼DynamicArray ()

  Destructor.
- T pop ()

  This will remove the last element and will return it.
- T & back ()

  This function will return the last element of the DynamicArray and can be also used to set the last element.
- T & front ()

  This function will return the first element of the DynamicArray and can be also used to set the first element.
- T & at (int index)

  This function will return the element stored at given index and can be also used to set the value at given index.
- T & operator[ ] (int index)

  This is another way to access and modify the value at given index. This is same as at() function.
- T removeAt (int index)

  This function will remove the element stored at the given index.
- void push_back (T key)

  This fucntion will add an element at last of the DynamicArray.
- void insertAt (int index, T key)

  This function will insert a value at given index.
- int size ()

  This function will return the size of the DynamicArray.
- int indexOf (T key)

  This function will return the index of an element in the DynamicArray, if found otherwise it will return -1.
- bool is_empty ()

  This function will return true if DynamicArray is empty else false.
- bool remove (T key)

  This function will remove an element from the DynamicArray if found and will return true else false.
- bool contains (T key)

  This will check whether given element is present in the Dynamic Array and will return true and false accordingly.

2.1.1 Constructor & Destructor Documentation

2.1.1.1 DynamicArray() [1/2]

```
template<class T >
DynamicArray< T >::DynamicArray (
             int size )
```

Constructor with an initial size.

Parameters

size	The initial size of the dynamic array

2.1.1.2 DynamicArray() [2/2]

```
template<class T >
DynamicArray< T >::DynamicArray (
             int size,
             T initialData )
```

Constructor with and initial size and initializing value.

Parameters

size	The initial size of the dynamic array
initialData	Initial value to initialize the array with.

2.1.2 Member Function Documentation

2.1.2.1 at()

```
template<class T >
T & DynamicArray< T >::at (
             int index )
```

This function will return the element stored at given index and can be also used to set the value at given index.

Parameters

index	Index where we need to access the element.

2.1.2.2 contains()

```
template<class T >
bool DynamicArray< T >::contains (
             T key )
```

This will check whether given element is present in the Dynamic Array and will return true and false accordingly.

Parameters

key	Element which has to be check whether it is present in the dynamicArray or not.

2.1.2.3 indexOf()

```
template<class T >
int DynamicArray< T >::indexOf (
             T key )
```

This function will return the index of an element in the DynamicArray, if found otherwise it will return -1.

Parameters

key	Element whose index has to be find

2.1.2.4 insertAt()

```
template<class T >
void DynamicArray< T >::insertAt (
             int index,
             T key )
```

This function will insert a value at given index.

Parameters

index	Index where we need to insert the element
key	Value which has to be inserted

2.1.2.5 operator[]()

```
template<class T >
T & DynamicArray< T >::operator[] (
             int index )
```

This is another way to access and modify the value at given index. This is same as at() function.

Parameters

index	Index where we need to access the element.

2.1.2.6 push_back()

```
template<class T >
void DynamicArray< T >::push_back (
             T key )
```

This fucntion will add an element at last of the DynamicArray.

Parameters

key	The value which has to be added at the last of the DynamicArray.

2.1.2.7 remove()

```
template<class T >
bool DynamicArray< T >::remove (
             T key )
```

This function will remove an element from the DynamicArray if found and will return true else false.

Parameters

key	Element which has to be removed

2.1.2.8 removeAt()

```
template<class T >
T DynamicArray< T >::removeAt (
             int index )
```

This function will remove the element stored at the given index.

Parameters

index	From where we need to remove the element

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

- include/dynamicArray.h
- src/dynamicArray.cpp

Index