Queue

1.0

Generated by Doxygen 1.8.8

Sun Nov 23 2014 23:20:27

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

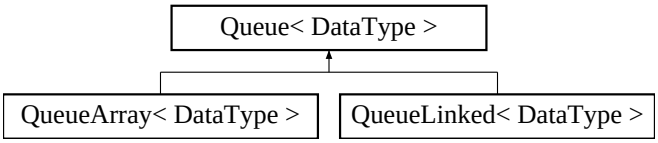
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 Queue< DataType > Class Template Reference

Inheritance diagram for Queue< DataType >:



Public Member Functions

- virtual void **enqueue** (const DataType &newDataItem)=0 throw (logic_error)
- virtual DataType **dequeue** ()=0 throw (logic_error)
- virtual void **clear** ()=0
- virtual bool **isEmpty** () const =0
- virtual bool **isFull** () const =0
- virtual void **showStructure** () const =0

Static Public Attributes

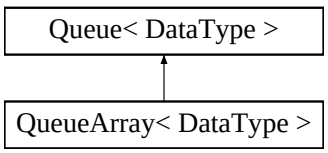
- static const int **MAX_QUEUE_SIZE** = 8

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

- Queue.h

3.2 QueueArray< DataType > Class Template Reference

Inheritance diagram for QueueArray< DataType >:



Public Member Functions

- **QueueArray** (int maxNumber=Queue< DataType >::MAX_QUEUE_SIZE)
- **QueueArray** (const QueueArray &other)
- QueueArray & **operator=** (const QueueArray &other)
- void **enqueue** (const DataType &newDataItem) throw (logic_error)
- DataType **dequeue** () throw (logic_error)
- void **clear** ()
- bool **isEmpty** () const
- bool **isFull** () const
- void **putFront** (const DataType &newDataItem) throw (logic_error)
- DataType **getRear** () throw (logic_error)
- int **getLength** () const
- void **showStructure** () const

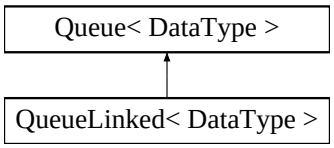
Additional Inherited Members

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

- QueueArray.h

3.3 QueueLinked< DataType > Class Template Reference

Inheritance diagram for QueueLinked< DataType >:



Public Member Functions

- QueueLinked (int maxNumber=Queue< DataType >::MAX_QUEUE_SIZE)
- QueueLinked (const QueueLinked &other)
- QueueLinked & operator= (const QueueLinked &other)

  Assignment operator overloaded.
- ~QueueLinked ()
- void enqueue (const DataType &newDataItem) throw (logic_error)
- DataType dequeue () throw (logic_error)
- void clear ()
- bool isEmpty () const
- bool isFull () const
- void putFront (const DataType &newDataItem) throw (logic_error)
- DataType getRear () throw (logic_error)
- int getLength () const
- void **showStructure** () const

Additional Inherited Members

3.3.1 Constructor & Destructor Documentation

3.3.1.1 template<typename DataType > QueueLinked< DataType >::QueueLinked (int *maxNumber* = Queue<DataType>::MAX_QUEUE_SIZE)

Default Constructor, sets pointers to NULL input parameter is irrelevant since we have no max size

3.3.1.2 template<typename DataType > QueueLinked< DataType >::QueueLinked (const QueueLinked< DataType > & *other*)

Copy Constructor copies the input parameter of another queue if it is empty just set pointers to NULL otherwise copy

3.3.1.3 template<typename DataType > QueueLinked< DataType >::~QueueLinked ( )

Deconstructor clears all memory

3.3.2 Member Function Documentation

3.3.2.1 template<typename DataType > void QueueLinked< DataType >::clear ( ) [virtual]

clear deletes all allocated memory for the queue nodes

Implements Queue< DataType >.

3.3.2.2 template<typename DataType > DataType QueueLinked< DataType >::dequeue ( ) throw logic_error) [virtual]

dequeue takes the front data

Implements Queue< DataType >.

3.3.2.3 template<typename DataType > void QueueLinked< DataType >::enqueue (const DataType & *newDataItem*) throw logic_error) [virtual]

Enqueue adds new data to queue create new node

if the current is empty make first (just realized this could have been coded better, too late to fix/test)

Implements Queue< DataType >.

3.3.2.4 template<typename DataType > int QueueLinked< DataType >::getLength ( ) const

getlength returns the length of the queue

3.3.2.5 template<typename DataType > DataType QueueLinked< DataType >::getRear ( ) throw logic_error)

getrear grabs the last data item in the queue check for empty

if there is only one item

save the back

go to prior

delete the last

set the next of the new back to null

3.3.2.6 template<typename DataType > bool QueueLinked< DataType >::isEmpty ( ) const [virtual]

isEmpty checks to see if the queue has any data

Implements Queue< DataType >.

3.3.2.7 template<typename DataType > bool QueueLinked< DataType >::isFull ( ) const [virtual]

ifFull irrelevant because always false

Implements Queue< DataType >.

3.3.2.8 template<typename DataType > void QueueLinked< DataType >::putFront (const DataType & *newDataItem*) throw logic_error)

putfront will add new data to the queue but differently than enqueue, will put the newest data up front

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

- QueueLinked.h
- QueueLinked.cpp