PA09

Generated by Doxygen 1.8.11

Contents

1 Hierarchical Index **1**

1.1 Class Hierarchy 1

2 Class Index **3**

2.1 Class List 3

3 File Index **5**

3.1 File List 5

4 Class Documentation **7**

4.1 Greater< KeyType > Class Template Reference 7

4.1.1 Member Function Documentation 7

4.1.1.1 operator()(const KeyType &a, const KeyType &b) const 7

4.2 Heap< DataType, KeyType, Comparator > Class Template Reference 7

4.2.1 Constructor & Destructor Documentation 8

4.2.1.1 Heap(int maxNumber=DEFAULT_MAX_HEAP_SIZE) 8

4.2.1.2 Heap(const Heap &other) 8

4.2.1.3 ~Heap() 9

4.2.2 Member Function Documentation 9

4.2.2.1 clear() 9

4.2.2.2 getLeftChild(const int nodeIndex) const 9

4.2.2.3 getParent(const int nodeIndex) const 10

4.2.2.4 getRightChild(const int nodeIndex) const 10

4.2.2.5 insert(const DataType &newDataItem) 10

4.2.2.6 isEmpty() const 11

4.2.2.7 isFull() const . . . 11
4.2.2.8 operator=(const Heap &other) . . . 11
4.2.2.9 remove() . . . 11
4.2.2.10 showStructure() const . . . 12
4.2.2.11 sortDown(const int nodeIndex) . . . 12
4.2.2.12 sortUp(const int nodeIndex) . . . 12
4.2.2.13 writeLevels() const . . . 12
4.2.3 Member Data Documentation . . . 13
4.2.3.1 DEFAULT_MAX_HEAP_SIZE . . . 13
4.3 Less< KeyType > Class Template Reference . . . 13
4.3.1 Member Function Documentation . . . 13
4.3.1.1 operator()(const KeyType &a, const KeyType &b) const . . . 13
4.4 PriorityQueue< DataType, KeyType, Comparator > Class Template Reference . . . 13
4.4.1 Constructor & Destructor Documentation . . . 14
4.4.1.1 PriorityQueue(int maxNumber=defMaxQueueSize) . . . 14
4.4.2 Member Function Documentation . . . 14
4.4.2.1 dequeue() . . . 14
4.4.2.2 enqueue(const DataType &newDataItem) . . . 14
4.5 TaskData Struct Reference . . . 15
4.5.1 Member Function Documentation . . . 15
4.5.1.1 getArrival() const . . . 15
4.5.1.2 getPriority() const . . . 15
4.5.2 Member Data Documentation . . . 15
4.5.2.1 arrived . . . 15
4.5.2.2 priority . . . 15
4.6 TestData Class Reference . . . 16
4.6.1 Member Function Documentation . . . 16
4.6.1.1 getPriority() const . . . 16
4.6.1.2 getPriority() const . . . 16
4.6.1.3 setPriority(int newPriority) . . . 16
4.6.1.4 setPriority(int newPriority) . . . 16
4.7 TestDataItem< KeyType > Class Template Reference . . . 16
4.7.1 Constructor & Destructor Documentation . . . 16
4.7.1.1 TestDataItem() . . . 16
4.7.2 Member Function Documentation . . . 16
4.7.2.1 getPriority() const . . . 16
4.7.2.2 operator=(const TestDataItem &orig) . . . 16
4.7.2.3 setPriority(KeyType newPty) . . . 16

5 File Documentation **17**

5.1 config.h File Reference . . . 17

5.1.1 Macro Definition Documentation . . . 17

5.1.1.1 LAB11_TEST1 . . . 17

5.2 Heap.cpp File Reference . . . 17

5.2.1 Detailed Description . . . 17

5.3 Heap.h File Reference . . . 18

5.4 heapsort.cs File Reference . . . 18

5.4.1 Function Documentation . . . 18

5.4.1.1 heapSort(DataType dataItems[], int size) . . . 18

5.4.1.2 moveDown(DataType dataItems[], int root, int size) . . . 18

5.5 ossim.cpp File Reference . . . 18

5.5.1 Function Documentation . . . 19

5.5.1.1 main() . . . 19

5.6 PriorityQueue.cpp File Reference . . . 19

5.6.1 Detailed Description . . . 19

5.7 PriorityQueue.h File Reference . . . 19

5.7.1 Variable Documentation . . . 19

5.7.1.1 defMaxQueueSize . . . 19

5.8 show11.cpp File Reference . . . 19

5.9 test11.cpp File Reference . . . 19

5.9.1 Function Documentation . . . 20

5.9.1.1 main() . . . 20

5.9.1.2 printHelp() . . . 20

5.10 test11hs.cpp File Reference . . . 20

5.10.1 Function Documentation . . . 20

5.10.1.1 main() . . . 20

5.10.2 Variable Documentation . . . 20

5.10.2.1 MAX_NUM_DATA_ITEMS . . . 20

5.11 test11pq.cpp File Reference . . . 20

5.11.1 Function Documentation . . . 21

5.11.1.1 main() . . . 21

5.11.1.2 printHelp() . . . 21

Index **23**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Greater< KeyType > 7
Heap< DataType, KeyType, Comparator > 7
Heap< DataType > 7
    PriorityQueue< DataType, KeyType, Comparator > 13
Less< KeyType > 13
Less< int > 13
TaskData 15
TestData 16
TestDataItem< KeyType > 16

Chapter 2

Class Index

2.1 Class List

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

Greater< KeyType > 7
Heap< DataType, KeyType, Comparator > 7
Less< KeyType > 13
PriorityQueue< DataType, KeyType, Comparator > 13
TaskData 15
TestData 16
TestDataItem< KeyType > 16

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

config.h . . . 17
Heap.cpp
An implementation file for a Heap . . . 17
Heap.h . . . 18
heapsort.cs . . . 18
ossim.cpp . . . 18
PriorityQueue.cpp
An implementation file for a Priorty Queue . . . 19
PriorityQueue.h . . . 19
show11.cpp . . . 19
test11.cpp . . . 19
test11hs.cpp . . . 20
test11pq.cpp . . . 20

Chapter 4

Class Documentation

4.1 Greater< KeyType > Class Template Reference

Public Member Functions

- bool operator() (const KeyType &a, const KeyType &b) const

4.1.1 Member Function Documentation

4.1.1.1 template<typename KeyType = int> bool Greater< KeyType >::operator() (const KeyType & *a*, const KeyType & *b*) const [inline]

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

- test11.cpp

4.2 Heap< DataType, KeyType, Comparator > Class Template Reference

```
#include <Heap.h>
```

Public Member Functions

- Heap (int maxNumber=DEFAULT_MAX_HEAP_SIZE)

  "Constructor. Creates an empty heap. Allocates enoughy memory for a heap containing maxNumber data items."
- Heap (const Heap &other)

  "Copy constructor. Initializes the object to be an equivalent copy of other."
- Heap & operator= (const Heap &other)

  "Overloaded assignment operator. Sets the heap to be equivalent to the other Heap and returns a reference to this object."
- ~Heap ()

  "Destructor. Deallocates (free) the memory used to store the heap."
- void insert (const DataType &newDataItem) throw (logic_error)

"Inserts newDataItem into the heap. Inserts this data item as the bottom rightmost data item in the heap and moves it upward until the properties that define a heap are restored."

- DataType remove () throw (logic_error)

  "Removes the data item with the highest priority (the root) from the heap and returns it. Replaces the root data item with the bottom rightmost data item and moves this data item downward until the properties that define a heap are restored."
- void clear ()

  "Removes all the data items in the heap."
- bool isEmpty () const

  "Returns true if the heap is empty. Otherwise, returns false."
- bool isFull () const

  "Returns true if the heap is full. Otherwise, returns false."
- void showStructure () const
- void writeLevels () const

  "Outputs the data items in a heap in level order, one level per line. Only outputs each data item's priority. If the heap is empty, then outputs 'Empty heap'."
- int getLeftChild (const int nodeIndex) const

  returns the left child
- int getRightChild (const int nodeIndex) const

  returns the right child
- int getParent (const int nodeIndex) const

  returns the parent
- void sortUp (const int nodeIndex)

  The sortUp function that recursively sorts for insert()
- void sortDown (const int nodeIndex)

  The sortDown function that recursively sorts for remove()

Static Public Attributes

- static const int DEFAULT_MAX_HEAP_SIZE = 10

4.2.1 Constructor & Destructor Documentation

4.2.1.1 template<typename DataType , typename KeyType , typename Comparator > Heap< DataType, KeyType, Comparator >::Heap (int *maxNumber* = DEFAULT_MAX_HEAP_SIZE)

"Constructor. Creates an empty heap. Allocates enoughy memory for a heap containing maxNumber data items."

Parameters

int	maxNumber

Returns

none

4.2.1.2 template<typename DataType , typename KeyType , typename Comparator > Heap< DataType, KeyType, Comparator >::Heap (const Heap< DataType, KeyType, Comparator > & *other*)

"Copy constructor. Initializes the object to be an equivalent copy of other."

Parameters

const	Heap& other

Returns

none

4.2.1.3 template<typename DataType , typename KeyType , typename Comparator > Heap< DataType, KeyType, Comparator >::~Heap ( )

"Destructor. Deallocates (free) the memory used to store the heap."

Parameters

none	

Returns

none

4.2.2 Member Function Documentation

4.2.2.1 template<typename DataType , typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::clear ( )

"Removes all the data items in the heap."

Parameters

none	

Returns

none

4.2.2.2 template<typename DataType , typename KeyType , typename Comparator > int Heap< DataType, KeyType, Comparator >::getLeftChild (const int *nodeIndex*) const

returns the left child

Parameters

const	int nodeIndex

Returns

(2 ∗ nodeIndex + 1)

4.2.2.3 template<typename DataType , typename KeyType , typename Comparator > int Heap< DataType, KeyType, Comparator >::getParent (const int *nodeIndex*) const

returns the parent

Parameters

const	int nodeIndex

Returns

(nodeIndex / 2) - 1 or (nodeIndex / 2)

4.2.2.4 template<typename DataType , typename KeyType , typename Comparator > int Heap< DataType, KeyType, Comparator >::getRightChild (const int *nodeIndex*) const

returns the right child

Parameters

const	int nodeIndex

Returns

(2 ∗ nodeIndex + 2)

4.2.2.5 template<typename DataType, typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::insert (const DataType & *newDataItem*) throw logic_error)

"Inserts newDataItem into the heap. Inserts this data item as the bottom rightmost data item in the heap and moves it upward until the properties that define a heap are restored."

Parameters

const	DataType& newDataItem

Returns

none

4.2.2.6 template<typename DataType , typename KeyType , typename Comparator > bool Heap< DataType, KeyType, Comparator >::isEmpty ( ) const

"Returns true if the heap is empty. Otherwise, returns false."

Parameters

none	

Returns

true or false

4.2.2.7 template<typename DataType , typename KeyType , typename Comparator > bool Heap< DataType, KeyType, Comparator >::isFull ( ) const

"Returns true if the heap is full. Otherwise, returns false."

Parameters

none	

Returns

true or false

4.2.2.8 template<typename DataType , typename KeyType , typename Comparator > Heap< DataType, KeyType, Comparator > & Heap< DataType, KeyType, Comparator >::operator= (const Heap< DataType, KeyType, Comparator > & *other*)

"Overloaded assignment operator. Sets the heap to be equivalent to the other Heap and returns a reference to this object."

Parameters

const	Heap& other

Returns

∗this

4.2.2.9 template<typename DataType , typename KeyType , typename Comparator > DataType Heap< DataType, KeyType, Comparator >::remove ( ) throw logic_error)

"Removes the data item with the highest priority (the root) from the heap and returns it. Replaces the root data item with the bottom rightmost data item and moves this data item downward until the properties that define a heap are restored."

Parameters

none	

Returns

Datatype temp

4.2.2.10 template<typename DataType , typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::showStructure ( ) const

4.2.2.11 template<typename DataType , typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::sortDown (const int *nodeIndex*)

The sortDown function that recursively sorts for remove()

Parameters

const	int nodeIndex

Returns

none

4.2.2.12 template<typename DataType , typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::sortUp (const int *nodeIndex*)

The sortUp function that recursively sorts for insert()

Parameters

const	int nodeIndex

Returns

none

4.2.2.13 template<typename DataType , typename KeyType , typename Comparator > void Heap< DataType, KeyType, Comparator >::writeLevels ( ) const

"Outputs the data items in a heap in level order, one level per line. Only outputs each data item's priority. If the heap is empty, then outputs 'Empty heap'."

Parameters

none	

Returns

none

4.2.3 Member Data Documentation

4.2.3.1 template<typename DataType, typename KeyType = int, typename Comparator = Less<KeyType>> const int Heap< DataType, KeyType, Comparator >::DEFAULT_MAX_HEAP_SIZE = 10 [static]

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

- Heap.h
- Heap.cpp
- show11.cpp

4.3 Less< KeyType > Class Template Reference

`#include <Heap.h>`

Public Member Functions

- bool operator() (const KeyType &a, const KeyType &b) const

4.3.1 Member Function Documentation

4.3.1.1 template<typename KeyType = int> bool Less< KeyType >::operator() (const KeyType & *a*, const KeyType & *b*) const [inline]

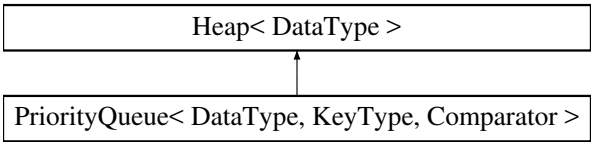
The documentation for this class was generated from the following file:

- Heap.h

4.4 PriorityQueue< DataType, KeyType, Comparator > Class Template Reference

`#include <PriorityQueue.h>`

Inheritance diagram for PriorityQueue< DataType, KeyType, Comparator >:



Public Member Functions

- PriorityQueue (int maxNumber=defMaxQueueSize)

  The default constructor for the PriorityQueue.
- void enqueue (const DataType &newDataItem)

  enqueue function for the PriortyQueue, it calls the insert function within the Heap ADT
- DataType dequeue ()

  dequeue function for the PriortyQueue, it calls the remove function within the Heap ADT

Additional Inherited Members

4.4.1 Constructor & Destructor Documentation

4.4.1.1 template<typename DataType , typename KeyType , typename Comparator > PriorityQueue< DataType, KeyType, Comparator >::PriorityQueue (int *maxNumber* = defMaxQueueSize)

The default constructor for the PriorityQueue.

Parameters

int	maxNumber

Returns

none

4.4.2 Member Function Documentation

4.4.2.1 template<typename DataType , typename KeyType , typename Comparator > DataType PriorityQueue< DataType, KeyType, Comparator >::dequeue ( )

dequeue function for the PriortyQueue, it calls the remove function within the Heap ADT

Parameters

none	

Returns

none

4.4.2.2 template<typename DataType , typename KeyType , typename Comparator > void PriorityQueue< DataType, KeyType, Comparator >::enqueue (const DataType & *newDataItem*)

enqueue function for the PriortyQueue, it calls the insert function within the Heap ADT

Parameters

const	DataType& newDataItem

Returns

none

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

- PriorityQueue.h
- PriorityQueue.cpp

4.5 TaskData Struct Reference

Public Member Functions

- int getPriority () const
- int getArrival () const

Public Attributes

- int priority
- int arrived

4.5.1 Member Function Documentation

4.5.1.1 int TaskData::getArrival ( ) const [inline]

4.5.1.2 int TaskData::getPriority ( ) const [inline]

4.5.2 Member Data Documentation

4.5.2.1 int TaskData::arrived

4.5.2.2 int TaskData::priority

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

- ossim.cpp

4.6 TestData Class Reference

Public Member Functions

- void setPriority (int newPriority)
- int getPriority () const
- void setPriority (int newPriority)
- int getPriority () const

4.6.1 Member Function Documentation

4.6.1.1 int TestData::getPriority ( ) const [inline]

4.6.1.2 int TestData::getPriority ( ) const [inline]

4.6.1.3 void TestData::setPriority (int *newPriority*) [inline]

4.6.1.4 void TestData::setPriority (int *newPriority*) [inline]

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

- test11hs.cpp
- test11pq.cpp

4.7 TestDataItem< KeyType > Class Template Reference

Public Member Functions

- TestDataItem ()
- void setPriority (KeyType newPty)
- KeyType getPriority () const
- TestDataItem & operator= (const TestDataItem &orig)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 template<typename KeyType > TestDataItem< KeyType >::TestDataItem ( ) [inline]

4.7.2 Member Function Documentation

4.7.2.1 template<typename KeyType > KeyType TestDataItem< KeyType >::getPriority ( ) const [inline]

4.7.2.2 template<typename KeyType > TestDataItem& TestDataItem< KeyType >::operator= (const TestDataItem< KeyType > & *orig*) [inline]

4.7.2.3 template<typename KeyType > void TestDataItem< KeyType >::setPriority (KeyType *newPty*) [inline]

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

- test11.cpp

Chapter 5

File Documentation

5.1 config.h File Reference

Macros

- #define LAB11_TEST1 1

5.1.1 Macro Definition Documentation

5.1.1.1 #define LAB11_TEST1 1

Heap class configuration file. Activate test #N by defining the corresponding LAB11_TESTN to have the value 1.

5.2 Heap.cpp File Reference

An implementation file for a Heap.

```
#include "Heap.h"
#include "show11.cpp"
```

5.2.1 Detailed Description

An implementation file for a Heap.

Author

Christopher Eichstedt

5.3 Heap.h File Reference

```
#include <stdexcept>
#include <iostream>
```

Classes

- class Less< KeyType >
- class Heap< DataType, KeyType, Comparator >

5.4 heapsort.cs File Reference

Functions

- void moveDown (DataType dataItems[ ], int root, int size)
- void heapSort (DataType dataItems[ ], int size)

5.4.1 Function Documentation

5.4.1.1 void heapSort (DataType *dataItems[ ],* int *size*)

5.4.1.2 void moveDown (DataType *dataItems[ ],* int *root,* int *size*)

5.5 ossim.cpp File Reference

```
#include <iostream>
#include <cstdlib>
#include "PriorityQueue.cpp"
```

Classes

- struct TaskData

Functions

- int main ()

5.5.1 Function Documentation

5.5.1.1 int main ( )

5.6 PriorityQueue.cpp File Reference

An implementation file for a Priorty Queue.

```
#include "PriorityQueue.h"
```

5.6.1 Detailed Description

An implementation file for a Priorty Queue.

Author

Christopher Eichstedt

5.7 PriorityQueue.h File Reference

```
#include <stdexcept>
#include <iostream>
#include "Heap.cpp"
```

Classes

- class PriorityQueue< DataType, KeyType, Comparator >

Variables

- const int defMaxQueueSize = 10

5.7.1 Variable Documentation

5.7.1.1 const int defMaxQueueSize = 10

5.8 show11.cpp File Reference

5.9 test11.cpp File Reference

```
#include <iostream>
#include <string>
#include <cctype>
#include "Heap.cpp"
#include "config.h"
```

Classes

- class TestDataItem< KeyType >
- class Greater< KeyType >

Functions

- void printHelp ()
- int main ()

5.9.1 Function Documentation

5.9.1.1 int main ( )

5.9.1.2 void printHelp ( )

5.10 test11hs.cpp File Reference

```
#include <iostream>
#include "heapsort.cpp"
```

Classes

- class TestData

Functions

- int main ()

Variables

- const int MAX_NUM_DATA_ITEMS = 10

5.10.1 Function Documentation

5.10.1.1 int main ( )

5.10.2 Variable Documentation

5.10.2.1 const int MAX_NUM_DATA_ITEMS = 10

5.11 test11pq.cpp File Reference

```
#include <iostream>
#include <cctype>
#include "PriorityQueue.cpp"
```

Classes

- class TestData

Functions

- void printHelp ()
- int main ()

5.11.1 Function Documentation

5.11.1.1 int main ( )

5.11.1.2 void printHelp ( )

Index

~Heap
    Heap, 9

arrived
    TaskData, 15

clear
    Heap, 9
config.h, 17
    LAB11_TEST1, 17

DEFAULT_MAX_HEAP_SIZE
    Heap, 13
defMaxQueueSize
    PriorityQueue.h, 19
dequeue
    PriorityQueue, 14

enqueue
    PriorityQueue, 14

getArrival
    TaskData, 15
getLeftChild
    Heap, 9
getParent
    Heap, 10
getPriority
    TaskData, 15
    TestData, 16
    TestDataItem, 16
getRightChild
    Heap, 10
Greater
    operator(), 7
Greater< KeyType >, 7

Heap
    ~Heap, 9
    clear, 9
    DEFAULT_MAX_HEAP_SIZE, 13
    getLeftChild, 9
    getParent, 10
    getRightChild, 10
    Heap, 8
    insert, 10
    isEmpty, 10
    isFull, 11
    operator=, 11
    remove, 11
    showStructure, 12
    sortDown, 12
    sortUp, 12
    writeLevels, 12
Heap< DataType, KeyType, Comparator >, 7
Heap.cpp, 17
Heap.h, 18
heapSort
    heapsort.cs, 18
heapsort.cs, 18
    heapSort, 18
    moveDown, 18

insert
    Heap, 10
isEmpty
    Heap, 10
isFull
    Heap, 11

LAB11_TEST1
    config.h, 17
Less
    operator(), 13
Less< KeyType >, 13

MAX_NUM_DATA_ITEMS
    test11hs.cpp, 20
main
    ossim.cpp, 19
    test11.cpp, 20
    test11hs.cpp, 20
    test11pq.cpp, 21
moveDown
    heapsort.cs, 18

operator()
    Greater, 7
    Less, 13
operator=
    Heap, 11
    TestDataItem, 16
ossim.cpp, 18
    main, 19

printHelp
    test11.cpp, 20
    test11pq.cpp, 21
priority
    TaskData, 15
PriorityQueue
    dequeue, 14

enqueue, 14
PriorityQueue, 14
PriorityQueue< DataType, KeyType, Comparator >, 13
PriorityQueue.cpp, 19
PriorityQueue.h, 19
defMaxQueueSize, 19

remove
Heap, 11

setPriority
TestData, 16
TestDataItem, 16
show11.cpp, 19
showStructure
Heap, 12
sortDown
Heap, 12
sortUp
Heap, 12

TaskData, 15
arrived, 15
getArrival, 15
getPriority, 15
priority, 15
test11.cpp, 19
main, 20
printHelp, 20
test11hs.cpp, 20
MAX_NUM_DATA_ITEMS, 20
main, 20
test11pq.cpp, 20
main, 21
printHelp, 21
TestData, 16
getPriority, 16
setPriority, 16
TestDataItem
getPriority, 16
operator=, 16
setPriority, 16
TestDataItem, 16
TestDataItem< KeyType >, 16

writeLevels
Heap, 12