PA09 - Heaps

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 insert(const DataType &newDataItem) 9

4.2.2.3 isEmpty() const 10

4.2.2.4 isFull() const 10

4.2.2.5 operator=(const Heap &other) 10

4.2.2.6 remove() 11

4.2.2.7 showStructure() const . . . 11
4.2.2.8 writeLevels() const . . . 12
4.2.3 Member Data Documentation . . . 12
4.2.3.1 DEFAULT_MAX_HEAP_SIZE . . . 12
4.3 Less< KeyType > Class Template Reference . . . 12
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 . . . 13
4.4.1.1 PriorityQueue(int maxNumber=defMaxQueueSize) . . . 13
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 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 . . . 15
4.6.1 Member Function Documentation . . . 16
4.6.1.1 getPriority() const . . . 16
4.6.1.2 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 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

5.3 Heap.h File Reference . . . 17

5.4 ossim.cpp File Reference . . . 18

5.4.1 Function Documentation . . . 18

5.4.1.1 main() . . . 18

5.5 PriorityQueue.cpp File Reference . . . 18

5.6 PriorityQueue.h File Reference . . . 18

5.6.1 Variable Documentation . . . 19

5.6.1.1 defMaxQueueSize . . . 19

5.7 test11.cpp File Reference . . . 19

5.7.1 Function Documentation . . . 19

5.7.1.1 main() . . . 19

5.7.1.2 printHelp() . . . 19

5.8 test11pq.cpp File Reference . . . 19

5.8.1 Function Documentation . . . 19

5.8.1.1 main() . . . 19

5.8.1.2 printHelp() . . . 19

Index **21**

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 > 12
Less< int > 12
TaskData 15
TestData 15
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 > 12
PriorityQueue< DataType, KeyType, Comparator > 13
TaskData 15
TestData 15
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 17
Heap.h 17
ossim.cpp 18
PriorityQueue.cpp 18
PriorityQueue.h 18
test11.cpp 19
test11pq.cpp 19

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)

  Default and Basic Constructor.
- Heap (const Heap &other)

  Copy constructor.
- Heap & operator= (const Heap &other)

  Overloaded Assignment Operator.
- ~Heap ()

  Destructor.
- void insert (const DataType &newDataItem) throw (logic_error)

  Insert.

- DataType remove () throw (logic_error)

  Remove.
- void clear ()

  Clear.
- bool isEmpty () const

  Empty Check.
- bool isFull () const

  Full Check.
- void showStructure () const

  Show Structure.
- void writeLevels () const

  Output in level order.

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)

Default and Basic Constructor.

Parameters

maxNumber	Maximum heap size

Postcondition

Creates an empty heap. Allocates enough memory for a heap containing maxNumber data items

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

Copy constructor.

Parameters

other	Address to the heap to be copied

Postcondition

Initializes the object to be an equivalent copy of other

See also

{references}

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

Destructor.

Postcondition

Deallocates the memory used to store the heap

See also

{References}

4.2.2 Member Function Documentation

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

Clear.

Postcondition

Removes all the data items in the heap

See also

{references}

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

Insert.

Parameters

newDataItem	Address of data item to be inserted

Precondition

Heap is not full

Postcondition

Inserts newDataItem into the heap.

Exceptions

<exception-object>	{exception description}

Note

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

See also

{references}

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

Empty Check.

Returns

True if heap is empty. Otherwise, returns false.

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

Full Check.

Returns

True if heap is full. Otherwise, returns false.

4.2.2.5 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.

Parameters

other	Address to the heap to be copied

Precondition

The address of this object cannot be the same as the address of other

Postcondition

Sets the heap to be equivalent to the other Heap

Returns

Reference to this object

See also

{references}

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

Remove.

Precondition

Heap is not empty

Postcondition

Removes the data item with the highest priority (the root) from the heap and returns it.

Exceptions

<exception-object>	{exception description}

Note

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

Returns

The root that is removed

See also

{references}

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

Show Structure.

Postcondition

Outputs the priorities of the data items in the heap in both array and tree form. The tree is output with its branches oriented from left (root) to right (leaves) - that is, the tree is output rotated counterclockwise ninety degrees from its conventional orientation. If the heap is empty, outputs "Empty Heap".

Note

Intended for testing/debugging purposes only

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

Output in level order.

Precondition

{description of the precondition}

Postcondition

{description of the postcondition}

Note

{text }

See also

{Reference}

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

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]

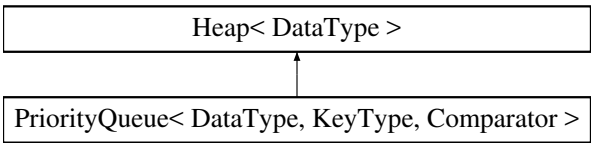
The documentation for this class was generated from the following file:

- 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)

  Constructor.
- void enqueue (const DataType &newDataItem) throw (logic_error)

  Enqueue.
- DataType dequeue () throw (logic_error)

  Dequeue.

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)

Constructor.

Parameters

maxNumber	Maximum queue size

Postcondition

Creates an empty priority queue. Allocates enough memory for a queue containing maxNumber data items

4.4.2 Member Function Documentation

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

Dequeue.

Precondition

Queue is not empty

Postcondition

Removes the highest priority (front) data item from the priority queue and returns it

Exceptions

<exception-object>	{exception description}

Returns

The front item that is removed from the queue

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

Enqueue.

Parameters

newDataItem	Address of item to be entered into the queue

Precondition

Queue is not full

Postcondition

Inserts newDataItem into the priority queue

Exceptions

<exception-object>	{exception description}

See also

{Reference}

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

Public Attributes

- int priority
- int arrived

4.5.1 Member Function Documentation

4.5.1.1 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

4.6.1 Member Function Documentation

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

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

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

- test11pq.cpp

4.7 TestDataItem< KeyType > Class Template Reference

Public Member Functions

- TestDataItem ()
- void setPriority (KeyType newPty)
- KeyType getPriority () const

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 > 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

```
#include <iostream>
#include <cmath>
#include "Heap.h"
```

5.3 Heap.h File Reference

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

Classes

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

5.4 ossim.cpp File Reference

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

Classes

- struct TaskData

Functions

- int main ()

5.4.1 Function Documentation

5.4.1.1 int main ( )

5.5 PriorityQueue.cpp File Reference

```
#include <iostream>
#include <stdexcept>
#include "PriorityQueue.h"
```

5.6 PriorityQueue.h File Reference

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

Classes

- class PriorityQueue< DataType, KeyType, Comparator >

Variables

- const int defMaxQueueSize = 10

5.6.1 Variable Documentation

5.6.1.1 const int defMaxQueueSize = 10

5.7 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 ()

5.7.1 Function Documentation

5.7.1.1 int main ( )

5.7.1.2 void printHelp ( )

5.8 test11pq.cpp File Reference

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

Classes

- class TestData

Functions

- void printHelp ()
- int main ()

5.8.1 Function Documentation

5.8.1.1 int main ( )

5.8.1.2 void printHelp ( )

Index

~Heap
    Heap, 9

arrived
    TaskData, 15

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

DEFAULT_MAX_HEAP_SIZE
    Heap, 12
defMaxQueueSize
    PriorityQueue.h, 19
dequeue
    PriorityQueue, 14

enqueue
    PriorityQueue, 14

getPriority
    TaskData, 15
    TestData, 16
    TestDataItem, 16
Greater
    operator(), 7
Greater< KeyType >, 7

Heap
    ~Heap, 9
    clear, 9
    DEFAULT_MAX_HEAP_SIZE, 12
    Heap, 8
    insert, 9
    isEmpty, 10
    isFull, 10
    operator=, 10
    remove, 11
    showStructure, 11
    writeLevels, 12
Heap< DataType, KeyType, Comparator >, 7
Heap.cpp, 17
Heap.h, 17

insert
    Heap, 9
isEmpty
    Heap, 10
isFull
    Heap, 10

LAB11_TEST1
    config.h, 17
Less
    operator(), 13
Less< KeyType >, 12

main
    ossim.cpp, 18
    test11.cpp, 19
    test11pq.cpp, 19

operator()
    Greater, 7
    Less, 13
operator=
    Heap, 10
ossim.cpp, 18
    main, 18

printHelp
    test11.cpp, 19
    test11pq.cpp, 19
priority
    TaskData, 15
PriorityQueue
    dequeue, 14
    enqueue, 14
    PriorityQueue, 13
PriorityQueue< DataType, KeyType, Comparator >, 13
PriorityQueue.cpp, 18
PriorityQueue.h, 18
    defMaxQueueSize, 19

remove
    Heap, 11

setPriority
    TestData, 16
    TestDataItem, 16
showStructure
    Heap, 11

TaskData, 15
    arrived, 15
    getPriority, 15
    priority, 15
test11.cpp, 19
    main, 19
    printHelp, 19
test11pq.cpp, 19
    main, 19

printHelp, 19
TestData, 15
getPriority, 16
setPriority, 16
TestDataItem
getPriority, 16
setPriority, 16
TestDataItem, 16
TestDataItem< KeyType >, 16

writeLevels
Heap, 12