PA07

Generated by Doxygen 1.8.11

Contents

1 Class Index

1.1 Class List

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

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

3 Class Documentation

3.1 BinaryNode< ItemType > Class Template Reference

Public Member Functions

- **BinaryNode** (const ItemType &anItem)
- **BinaryNode** (const ItemType &anItem, std::shared_ptr< BinaryNode< ItemType >> leftPtr, std::shared_ptr< BinaryNode< ItemType >> rightPtr)
- void **setItem** (const ItemType &anItem)
- ItemType **getItem** () const
- bool **isLeaf** () const
- auto **getLeftChildPtr** () const
- auto **getRightChildPtr** () const
- void **setLeftChildPtr** (std::shared_ptr< BinaryNode< ItemType >> leftPtr)
- void **setRightChildPtr** (std::shared_ptr< BinaryNode< ItemType >> rightPtr)

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

- BinaryNode.h
- BinaryNode.cpp

3.2 RedBlackNode< ItemType > Class Template Reference

Public Member Functions

- **RedBlackNode** (const ItemType &anItem)
- void **setItem** (const ItemType &anItem)
- ItemType **getItem** () const
- bool **isLeaf** () const
- auto **getParentPtr** () const
- auto **getLeftChildPtr** () const
- auto **getRightChildPtr** () const
- void **setParentPtr** (std::shared_ptr< RedBlackNode< ItemType >> parent)
- void **setLeftChildPtr** (std::shared_ptr< RedBlackNode< ItemType >> leftPtr)
- void **setRightChildPtr** (std::shared_ptr< RedBlackNode< ItemType >> rightPtr)
- void **setColor** (const Color &color)
- void **setLeftColor** (const Color &color)
- void **setRightColor** (const Color &color)
- Color **getColor** () const
- Color **getLeftColor** () const
- Color **getRightColor** () const

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

- RedBlackNode.h
- RedBlackNode.cpp

3.3 RedBlackTree< ItemType > Class Template Reference

Public Member Functions

- bool isEmpty ()
- void add (const ItemType &item)
- int getHeight () const
- void inorderTraverse (void visit(ItemType &)) const
- void clear ()

3.3.1 Member Function Documentation

3.3.1.1 template<class ItemType > void RedBlackTree< ItemType >::add (const ItemType & *item*)

Adds the given data to this binary tree.

Parameters

newData	The data to add to the binary tree.

Postcondition

The binary tree contains the new data.

3.3.1.2 template<class ItemType > void RedBlackTree< ItemType >::clear ( )

Removes all data from this binary tree.

3.3.1.3 template<class ItemType > int RedBlackTree< ItemType >::getHeight ( ) const

Gets the height of this binary tree.

Returns

The height of the binary tree.

3.3.1.4 template<class ItemType > void RedBlackTree< ItemType >::inorderTraverse (void *visitItemType &*) const

Traverses this binary tree in inorder and calls the function visit once for each node.

Parameters

visit	A client-defined function that performs an operation on either each visited node or its data.

3.3.1.5 template<class ItemType > bool RedBlackTree< ItemType >::isEmpty ( )

Tests whether this binary tree is empty.

Returns

True if the binary tree is empty, or false if not.

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

- RedBlackTree.h
- RedBlackTree.cpp

4 File Documentation

4.1 BinaryNode.h File Reference

```
#include <memory>
#include "BinaryNode.cpp"
```

Classes

- class BinaryNode< ItemType >

4.1.1 Detailed Description

A class of nodes for a link-based binary tree.

4.2 RedBlackNode.h File Reference

```
#include <memory>
#include "RedBlackNode.cpp"
```

Classes

- class RedBlackNode< ItemType >

Enumerations

- enum **Color** { **RED**, **BLACK** }

4.2.1 Detailed Description

A class of nodes for a Red-Black tree.

4.3 RedBlackTree.h File Reference

```
#include <memory>
#include "RedBlackNode.h"
#include "RedBlackTree.cpp"
```

Classes

- class RedBlackTree< ItemType >

4.3.1 Detailed Description

Red-Black tree

Index