fantasy19 / **Derp**

Code Issues Pull requests Actions Projects Security Insights

e681c03cbb

Derp / 280 / assignment04-BinaryTree / assignment04-BinaryTree / **AVLTree.h**

fantasy19 No commit message

1 contributor

Raw Blame

52 lines (43 sloc) | 1.8 KB

```cpp
/******************************************************************************/
/*!
\file   AVLTree.h
\author Ang Cheng Yong
\par    email: a.chengyong\@digipen.edu
\par    DigiPen login: a.chengyong
\par    Course: CS280
\par    Programming Assignment #2
\date   8/11/2016
\brief
This file contains the driver functions needed for AVL.
*/
/******************************************************************************/
//----------------------------------------------------------------------------
#ifndef AVLTREE_H
#define AVLTREE_H
//----------------------------------------------------------------------------
#include <stack>
#include "BSTree.h"

template <typename T>
class AVLTree : public BSTree<T>
{
  public:
          AVLTree(ObjectAllocator *OA = 0, bool ShareOA = false);
          virtual ~AVLTree();

          virtual void insert(const T& value) throw(BSTException);
          virtual void remove(const T& value);

      // Returns true if efficiency implemented
```

```
            static bool ImplementedBalanceFactor(void);

  private:
    // private stuff
            void insert_begin(typename BSTree<T>::BinTree &tree, const T& value);
            void insert_node(typename BSTree<T>::BinTree & node, const T& value, std::stack<typen
 
            void remove_begin(typename BSTree<T>::BinTree &tree, const T& value);
            void remove_node(typename BSTree<T>::BinTree &tree, const T& value, std::stack<typena

            void RotateLeft(typename BSTree<T>::BinTree &node);
            void RotateRight(typename BSTree<T>::BinTree &node);

            void BalanceAVLTree(std::stack<typename BSTree<T>::BinTree*> & nodes);
            unsigned int node_count(typename BSTree<T>::BinTree& tree) const;

};

#include "AVLTree.cpp"

#endif
//---------------------------------------------------------------------------
```