30/30

Printout for cs320-09-A17-BiTreeCSV.hpp

```
// File: DictionaryBST/DictTcontainsVis.hpp
// Olivia Lara
// November 3, 2017

#ifndef DICTTCONTAINSVIS_HPP_
#define DICTTCONTAINSVIS_HPP_

#include <iostream>
#include "ADictionary.hpp"
#include "BiTreeCSV.hpp"
using namespace std;

// ========= DictTcontainsVis =========
template <class K, class V>
class DictTcontainsVis : public ABiTreeCSVVis< DictPair<K, V> > {
private:
    K const &_key; // Input parameter.
    V const *_val; // Output result.
    bool _found; // Output result.

public:
    // ========= Constructor =========
    DictTcontainsVis(K const &key) :
        _key(key) {
    }

    // ========= visit =========
    void emptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        _found = false;
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        if(_key < host.root().key()) {
            host.left().accept(*this);
        } else if(_key > host.root().key()) {
            host.right().accept(*this);
        } else {
            _val = &(host.root().val());
            _found = true;
        }
    }

    // ========= visit const =========
    void emptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        _found = false;
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        if(_key < host.root().key()) {
            host.left().accept(*this);
        } else if(_key > host.root().key()) {
            host.right().accept(*this);
        } else {
            _val = &(host.root().val());
            _found = true;
        }
    }

    // ========= result =========
    // Pre: This visitor has been accepted by a host tree.
    // Post: If key is found, then val is the associated value, and true is returned;
    // otherwise false is returned.
    bool result(V &val) const {
        if (_found) {
            val = *_val;
        }
        return _found;
    }
};

#endif
// new page
// File: DictionaryBST/DictTinsertVis.hpp
// Olivia Lara
// November 3, 2017

#ifndef DICTTINSERTVIS_HPP_
#define DICTTINSERTVIS_HPP_

#include <iostream>
#include "ADictionary.hpp"
#include "BiTreeCSV.hpp"
using namespace std;

// ========= DictTinsertVis =========
template < class K, class V>
class DictTinsertVis: public ABiTreeCSVVis< DictPair<K, V> > {
private:
    K const &_key; // Input parameter.
    V const &_val; // Input parameter.

public:
    // ========= Constructor =========
    DictTinsertVis(K const &key, V const &val):
        _key(key),
        _val(val) {
    }

    // ========= visit =========
    // Pre: This visitor has been accepted by a host tree.
    // Post: The host dictionary contains key and its associated value, val.
    void emptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        host.insertRoot(DictPair<K, V> (_key, _val));
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        if(_key < host.root().key()){
            host.left().accept(*this);
        } else if (_key > host.root().key()){
            host.right().accept(*this);
        } else if (_key == host.root().key()){
            host.root() = DictPair<K, V> (_key, _val);
        }
    }

    // ========= visit const =========
    void emptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        cerr << "DictTinsertVis precondition violated: "
             << "Cannot insert into a const tree." << endl;
        throw -1;
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        cerr << "DictTinsertVis precondition violated: "
             << "Cannot insert into a const tree." << endl;
        throw -1;
    }
};

#endif

// new page
// File: DictionaryBST/DictTremoveVis.hpp
// Olivia Lara
// November 3, 2017

#ifndef DICTTREMOVEVIS_HPP_
#define DICTTREMOVEVIS_HPP_

#include <iostream>
#include "ADictionary.hpp"
#include "BiTreeCSV.hpp"
#include "BiTCSVisEmptyVis.hpp"
using namespace std;

// ========= DictTremoveVis =========

template <class K, class V>
class DictTremoveVis : public ABiTreeCSVVis< DictPair<K, V> > {
private:
    K const &_key; // Input parameter.
    V _val; // Output result.
    bool _found; // Output result.

public:
    // ========= Constructor =========

    DictTremoveVis(K const &key) :
        _key(key) {
    }

    // ========= visit =========

    void emptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        _found = false;
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > &host) override {
        if(_key < host.root().key()) {
            host.left().accept(*this);
        } else if(_key > host.root().key()) {
            host.right().accept(*this);
        } else {
            _val = host.root().val();
            if(isEmpty(host.left())) {
                host.remRoot();
            } else {
                BiTreeCSV< DictPair<K, V> > *maxLeft = &host.left();
                while(!(isEmpty(maxLeft->right()))) {
                    maxLeft = &maxLeft->right();
                }
                host.root() = maxLeft->root();
                maxLeft->remRoot();
            }
          _found = true;
        }
    }

    // ========= visit const =========

    void emptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        cerr << "DictTremoveVis precondition violated: "
             << "Cannot remove from a const tree." << endl;
        throw -1;
    }

    void nonEmptyCase(BiTreeCSV< DictPair<K, V> > const &host) override {
        cerr << "DictTremoveVis precondition violated: "
             << "Cannot remove from a const tree." << endl;
        throw -1;
    }

    // ========= result =========
    // Pre: This visitor has been accepted by a host tree.
    // Post: If key is found, then val is the associated value, and true is returned;
    // otherwise false is returned.
    // The host dictionary does not contain key and its associated value.

    bool result(V &val) const {
        if (_found) {
            val = _val;
        }
        return _found;
    }
};

#endif
// new page
// new page
```

```
clang++ -stdlib=libc++ -std=c++11 -c DictionaryBSTMain.cpp \
        -I ../dp4dsDistribution/Utilities \
        -I ../dp4dsDistribution/ADictionary \
        -I ../dp4dsDistribution/ArrayT 2>&1
clang++ -stdlib=libc++ -std=c++11 -o DictionaryBSTMain DictionaryBSTMain.o \
        ../dp4dsDistribution/Utilities/Utilities.o
strip DictionaryBSTMain

// new page
```

```
Testing cs320-09

Testing cs320-09 unit-contains
==============================

<40, cat>
|_ <20, rat>
|   |_ ()
|   |_ <30, giraffe>
|       |_ ()
|       |_ ()
|_ <50, dog>
    |_ ()
    |_ ()

Key 15 is not in the dictionary.

Key 25 is not in the dictionary.

Key 35 is not in the dictionary.

Key 45 is not in the dictionary.

Key 55 is not in the dictionary.

The value for key 20 is rat.

The value for key 30 is giraffe.

The value for key 40 is cat.

The value for key 50 is dog.

// new page
```

```
Testing cs320-09 unit-insert
============================

<40, cat>
|_ <20, rat>
|   |_ <10, pig>
|   |   |_ ()
|   |   |_ ()
|   |_ <30, giraffe>
|       |_ ()
|       |_ ()
|_ <50, dog>
    |_ ()
    |_ ()

<40, cat>
|_ <20, rat>
|   |_ <10, pig>
|   |   |_ ()
|   |   |_ ()
|   |_ <30, hippopotamus>
|       |_ ()
|       |_ ()
|_ <50, dog>
    |_ ()
    |_ ()

// new page
```

```
Testing cs320-09 unit-remove
============================

<70, cat>
|_ <60, dog>
|   |_ <10, rat>
|   |   |_ ()
|   |   |_ <30, giraffe>
|   |       |_ <20, lion>
|   |       |   |_ ()
|   |       |   |_ ()
|   |       |_ <50, aardvark>
|   |           |_ <40, zebra>
|   |           |   |_ ()
|   |           |   |_ ()
|   |           |_ ()
|   |_ ()
|_ <80, snake>
    |_ ()
    |_ <100, pig>
        |_ <90, fish>
        |   |_ ()
        |   |_ ()
        |_ <110, bird>
            |_ ()
            |_ ()

Key 5 is not in the dictionary.

Key 15 is not in the dictionary.

Key 25 is not in the dictionary.

Key 35 is not in the dictionary.

Key 115 is not in the dictionary.

Key 80 removed with a value of snake.

<70, cat>
|_ <60, dog>
|   |_ <10, rat>
|   |   |_ ()
|   |   |_ <30, giraffe>
|   |       |_ <20, lion>
|   |       |   |_ ()
|   |       |   |_ ()
|   |       |_ <50, aardvark>
|   |           |_ <40, zebra>
|   |           |   |_ ()
|   |           |   |_ ()
|   |           |_ ()
|   |_ ()
|_ <100, pig>
    |_ <90, fish>
    |   |_ ()
    |   |_ ()
    |_ <110, bird>
        |_ ()
        |_ ()

Key 90 removed with a value of fish.

<70, cat>
|_ <60, dog>
```

```
|  |_ <10, rat>
|  |  |_ ()
|  |  |_ <30, giraffe>
|  |     |_ <20, lion>
|  |     |  |_ ()
|  |     |  |_ ()
|  |     |_ <50, aardvark>
|  |        |_ <40, zebra>
|  |        |  |_ ()
|  |        |  |_ ()
|  |        |_ ()
|  |_ ()
|_ <100, pig>
   |_ ()
   |_ <110, bird>
      |_ ()
      |_ ()

Key 60 removed with a value of dog.

<70, cat>
|_ <50, aardvark>
|  |_ <10, rat>
|  |  |_ ()
|  |  |_ <30, giraffe>
|  |     |_ <20, lion>
|  |     |  |_ ()
|  |     |  |_ ()
|  |     |_ <40, zebra>
|  |        |_ ()
|  |        |_ ()
|  |_ ()
|_ <100, pig>
   |_ ()
   |_ <110, bird>
      |_ ()
      |_ ()

Key 70 removed with a value of cat.

<50, aardvark>
|_ <10, rat>
|  |_ ()
|  |_ <30, giraffe>
|     |_ <20, lion>
|     |  |_ ()
|     |  |_ ()
|     |_ <40, zebra>
|        |_ ()
|        |_ ()
|_ <100, pig>
   |_ ()
   |_ <110, bird>
      |_ ()
      |_ ()

// new page
```