My Project

Generated by Doxygen 1.9.8

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 EnglishRussianDictionary Class Reference

Public Member Functions

- ∼**EnglishRussianDictionary** ()
- bool **IsContain** (std::string english_word)
- bool **AddNewPairOfWords** (std::string english_word, std::string russian_translation)
- int **CountNamberOfWords** ()
- bool **DeleteEnglishWord** (std::string english_word)
- bool **ChangeTranslation** (std::string english_word, std::string russian_translation)
- std::string **operator[ ]** (std::string english_word)
- bool **ReadDictionaryFromFile** (**EnglishRussianDictionary** &new_dictionary, std::string filename)

Protected Member Functions

- void **DeleteAll** (**PairOfWords** ∗root)
- **PairOfWords** ∗ **FindPerentForNew** (std::string english_word, **PairOfWords** ∗root)
- bool **DeleteEnglishWordNoChildren** (**PairOfWords** ∗root)
- bool **DeleteEnglishWordOneChild** (**PairOfWords** ∗root)
- bool **DeleteEnglishWordTwoChildren** (**PairOfWords** ∗root)
- **PairOfWords** ∗ **FindPerantElement** (std::string english_word, **PairOfWords** ∗root)
- **PairOfWords** ∗ **FindElement** (std::string english_word, **PairOfWords** ∗root)
- int **CountNamberOfWordsLogic** (**PairOfWords** ∗root)

Protected Attributes

- **PairOfWords** ∗ **english_russian_dictionary_tree_root_**

3.1.1 Constructor & Destructor Documentation

3.1.1.1 ∼EnglishRussianDictionary()

```
EnglishRussianDictionary::∼EnglishRussianDictionary ( )
```

Clears the memory

3.1.2 Member Function Documentation

3.1.2.1 AddNewPairOfWords()

```
bool EnglishRussianDictionary::AddNewPairOfWords (
             std::string english_word,
             std::string russian_translation )
```

Adds the pair of words. Also checs if the added element already in tree

3.1.2.2 ChangeTranslation()

```
bool EnglishRussianDictionary::ChangeTranslation (
             std::string english_word,
             std::string russian_translation )
```

Changes the translation

3.1.2.3 CountNamberOfWords()

```
int EnglishRussianDictionary::CountNamberOfWords ( )
```

Counts namber of words in dictinary

3.1.2.4 DeleteEnglishWord()

```
bool EnglishRussianDictionary::DeleteEnglishWord (
             std::string english_word )
```

Searches and deletes element

3.1.2.5 DeleteEnglishWordNoChildren()

```
bool EnglishRussianDictionary::DeleteEnglishWordNoChildren (
              PairOfWords * root )  [protected]
```

algorithm, that deletes element without children

3.1.2.6 DeleteEnglishWordOneChild()

```
bool EnglishRussianDictionary::DeleteEnglishWordOneChild (
              PairOfWords * root )  [protected]
```

algorithm, that deletes element with one child

3.1.2.7 DeleteEnglishWordTwoChildren()

```
bool EnglishRussianDictionary::DeleteEnglishWordTwoChildren (
              PairOfWords * root )  [protected]
```

algorithm, that deletes element with two children

3.1.2.8 FindPerantElement()

```
 PairOfWords * EnglishRussianDictionary::FindPerantElement (
             std::string english_word,
              PairOfWords * root )  [protected]
```

searches for elemen, that is perent

3.1.2.9 FindPerentForNew()

```
 PairOfWords * EnglishRussianDictionary::FindPerentForNew (
             std::string english_word,
              PairOfWords * root )  [protected]
```

Searches for plsce, were element coud be instolen

3.1.2.10 IsContain()

```
bool EnglishRussianDictionary::IsContain (
             std::string english_word )
```

Searches for an element

3.1.2.11 ReadDictionaryFromFile()

```
bool EnglishRussianDictionary::ReadDictionaryFromFile (
              EnglishRussianDictionary & new_dictionary,
             std::string filename )
```

Reads information from file

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

- Sample-Test2/ **EnglishRussianDictionary.h**
- Sample-Test2/ **EnglishRussianDictionary.cpp**

3.2 PairOfWords Struct Reference

Struct, thet contain information about nodes of the binar tree.

```
#include <EnglishRussianDictionary.h>
```

Public Attributes

- std::string **english_word_**
  It's english word, thet is used like key.
- std::string **russian_tranlation_**
  It's the translstion of the english word.
- **PairOfWords** ∗ **right_child_** = NULL
  It's the pointer to one of 2 children.
- **PairOfWords** ∗ **left_child_** = NULL
  It's the pointer to one of 2 children.

3.2.1 Detailed Description

Struct, thet contain information about nodes of the binar tree.

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

- Sample-Test2/ **EnglishRussianDictionary.h**

Chapter 4

File Documentation

4.1 Sample-Test2/EnglishRussianDictionary.cpp File Reference

```
#include "EnglishRussianDictionary.h"
#include <string>
```

4.1.1 Detailed Description

File, thet containe all informaition about realization of main class **EnglishRussianDictionary** (p. 5)

4.2 Sample-Test2/EnglishRussianDictionary.h File Reference

```
#include <string>
#include <fstream>
#include "gtest/gtest.h"
```

Classes

- struct **PairOfWords**

  Struct, thet contain information about nodes of the binar tree.
- class **EnglishRussianDictionary**

4.2.1 Detailed Description

This file containce prototipes of main Class

4.3 EnglishRussianDictionary.h

Go to the documentation of this file.

```

#pragma once
#include <string>
#include <fstream>
#include "gtest/gtest.h"
//#include "d:\LW1\packages\Microsoft.googletest.v140.windesktop.msvcstl.static.rt-dyn.1.8.1.6\build\native\include\gtest\gtest.h"
//#include "D:\LW1\packages\Microsoft.googletest.v140.windesktop.msvcstl.static.rt-dyn.1.8.1.6\build\native\include\gtest\gtest.h"

struct PairOfWords
{
    std::string english_word_;
    std::string russian_tranlation_;
    PairOfWords* right_child_ = NULL;
    PairOfWords* left_child_ = NULL;
};


class EnglishRussianDictionary
{
protected:
    PairOfWords* english_russian_dictionary_tree_root_;

    void DeleteAll(PairOfWords* root);//done

    PairOfWords* FindPerentForNew(std::string english_word, PairOfWords* root);//done

    bool DeleteEnglishWordNoChildren(PairOfWords* root);

    bool DeleteEnglishWordOneChild(PairOfWords* root);

    bool DeleteEnglishWordTwoChildren(PairOfWords* root);

    PairOfWords* FindPerantElement(std::string english_word, PairOfWords* root);

    PairOfWords* FindElement(std::string english_word, PairOfWords* root);//done

    int CountNamberOfWordsLogic(PairOfWords* root);//done

public:
    EnglishRussianDictionary();//done

    ~EnglishRussianDictionary();//done

    bool IsContain(std::string english_word);//done

    bool AddNewPairOfWords(std::string english_word, std::string russian_translation);//done

    int CountNamberOfWords();//done

    bool DeleteEnglishWord(std::string english_word);

    bool ChangeTranslation(std::string english_word, std::string russian_translation);//done

    std::string operator[](std::string english_word);//done

    bool ReadDictionaryFromFile(EnglishRussianDictionary& new_dictionary, std::string filename);//done
};


```

4.4 pch.h

```
//
// pch.h
//

#pragma once

#include "gtest/gtest.h"
```

4.5 Sample-Test2/test.cpp File Reference

```
#include "pch.h"
#include "EnglishRussianDictionary.h"
```

Functions

- **TEST** (TestCaseName, TestName)
- **TEST** (DictionaryTest, FindElementTest)
- **TEST** (DictionaryTest, DeleteElementTest)
- **TEST** (DictionaryTest, AddNewTest)
- **TEST** (DictionaryTest, CountNamberTest)
- **TEST** (DictionaryTest, ReadDictionaryFromFileTest)

Index