My Project

Generated by Doxygen 1.8.1.2

Wed Nov 27 2013 12:37:54

Contents

1 Todo List **1**

2 Class Index **3**

2.1 Class List . . . 3

3 Class Documentation **5**

3.1 diccionario< Key, Def >::const_iterator Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.2 diccionario< Key, Def > Class Template Reference . . . 5

3.2.1 Constructor & Destructor Documentation . . . 7

3.2.1.1 diccionario . . . 7

3.2.1.2 diccionario . . . 7

3.2.1.3 diccionario . . . 7

3.2.2 Member Function Documentation . . . 7

3.2.2.1 begin . . . 7

3.2.2.2 begin . . . 7

3.2.2.3 end . . . 8

3.2.2.4 end . . . 8

3.2.2.5 find . . . 8

3.2.2.6 operator= . . . 9

3.2.2.7 operator[] . . . 9

3.2.2.8 operator[] . . . 9

3.2.2.9 size . . . 9

3.2.3 Friends And Related Function Documentation . . . 10

3.2.3.1 operator<< . . . 10

3.3 diccionario< Key, Def >::iterator Class Reference . . . 10

3.3.1 Detailed Description . . . 10

3.3.2 Constructor & Destructor Documentation . . . 10

3.3.2.1 iterator . . . 10

3.3.3 Member Function Documentation . . . 11

3.3.3.1 operator!= . . . 11

3.3.3.2 operator* . . . 11

3.3.3.3 operator++ 11
3.3.3.4 operator++ 11
3.3.3.5 operator== 11

Chapter 1

Todo List

Member diccionario< Key, Def >::begin ()
implementa esta función

Member diccionario< Key, Def >::end ()
implementa esta función

Member diccionario< Key, Def >::find (const Key &s)
implementa esta función

Member diccionario< Key, Def >::iterator::iterator ()
implementa esta función

Member diccionario< Key, Def >::iterator::operator!= (const iterator &it)
implementa esta función

Member diccionario< Key, Def >::iterator::operator* () const
implementa esta función

Member diccionario< Key, Def >::iterator::operator++ ()
implementa esta función

Member diccionario< Key, Def >::iterator::operator++ (int)
implementa esta función

Member diccionario< Key, Def >::iterator::operator== (const iterator &it)
implementa esta función

Chapter 2

Class Index

2.1 Class List

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

diccionario< Key, Def >::const_iterator
Class const_iterator forward iterador constante sobre el diccionario, Lectura const_iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!= 5

diccionario< Key, Def > 5

diccionario< Key, Def >::iterator
Class iterator forward iterador sobre el diccionario, Lectura y Escritura iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!= 10

Chapter 3

Class Documentation

3.1 diccionario< Key, Def >::const_iterator Class Reference

class const_iterator forward iterador constante sobre el diccionario, Lectura const_iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!=

`#include <diccionario.h>`

Public Member Functions

- **const_iterator** (const const_iterator &it)
- **const_iterator** (const iterator &it)
- const entrada & **operator*** () const
- iterator **operator++** (int)
- iterator & **operator++** ()
- bool **operator==** (const iterator &it)
- bool **operator!=** (const iterator &it)
- const_iterator & **operator=** (const const_iterator &it)

Friends

- class **diccionario**

3.1.1 Detailed Description

template<typename Key = string, typename Def = int>class diccionario< Key, Def >::const_iterator

class const_iterator forward iterador constante sobre el diccionario, Lectura const_iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!=

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

- diccionario.h

3.2 diccionario< Key, Def > Class Template Reference

Classes

- class const_iterator

class const_iterator forward iterador constante sobre el diccionario, Lectura const_iterator ,operator, operator++, operator++(int) operator=, operator==, operator!=*

- class iterator

  class iterator forward iterador sobre el diccionario, Lectura y Escritura iterator ,operator, operator++, operator++(int) operator=, operator==, operator!=*

Public Types

- typedef pair< Key, Def > **entrada**
- typedef pair< const Key, Def > **value_type**
- typedef unsigned int **size_type**

Public Member Functions

- diccionario ()

  constructor primitivo.
- diccionario (const entrada &nula)

  constructor primitivo.
- diccionario (const diccionario &d)

  constructor de copia
- iterator begin ()

  entrada nula del diccionario
- iterator end ()

  entrada nula del diccionario
- const_iterator begin () const

  entrada nula del diccionario
- const_iterator end () const

  entrada nula del diccionario
- iterator find (const Key &s)

  busca una cadena en el diccionario
- Def & operator[] (const Key &s)

  busca una cadena en el diccionario
- const Def & operator[] (const Key &s) const

  Consulta una entrada en el diccionario.
- diccionario & operator= (const diccionario &org)

  operador de asignación
- size_type size () const

  numero de entradas en el diccionario
- bool empty () const

  vacia Chequea si el priority_queue esta vacio (size()==0)

Friends

- class **iterator**
- ostream & operator<< (ostream &sal, const diccionario< Key, Def > &D)

  imprime todas las entradas del diccionario

3.2.1 Constructor & Destructor Documentation

3.2.1.1 template<typename Key , typename Def > diccionario< Key, Def >::diccionario ( )

constructor primitivo.

Postcondition

3.2.1.2 template<typename Key = string, typename Def = int> diccionario< Key, Def >::diccionario (const entrada & *nula*)

constructor primitivo.

Parameters

in	*nula*	representa a la entrada nula para el diccionario

Postcondition

define la entrada nula

3.2.1.3 template<typename Key , typename Def > diccionario< Key, Def >::diccionario (const diccionario< Key, Def > & *d*)

constructor de copia

Parameters

in	*d*	diccionario a copiar

3.2.2 Member Function Documentation

3.2.2.1 template<typename Key , typename Def > diccionario< Key, Def >::iterator diccionario< Key, Def >::begin ( )

entrada nula del diccionario

Returns

Devuelve el iterador a la primera posición del diccionario.

Postcondition

no modifica el diccionario

Todo implementa esta función

3.2.2.2 template<typename Key = string, typename Def = int> const_iterator diccionario< Key, Def >::begin ( ) const

entrada nula del diccionario

Returns

Devuelve el iterador a la primera posición del diccionario.

Postcondition

no modifica el diccionario

3.2.2.3 template<typename Key , typename Def > diccionario< Key, Def >::iterator diccionario< Key, Def >::end ( )

entrada nula del diccionario

Returns

Devuelve el iterador a la ultima posición del diccionario.

Postcondition

no modifica el diccionario

Todo implementa esta función

3.2.2.4 template<typename Key = string, typename Def = int> const_iterator diccionario< Key, Def >::end ( ) const

entrada nula del diccionario

Returns

Devuelve el iterador a la ultima posición del diccionario.

Postcondition

no modifica el diccionario

3.2.2.5 template<typename Key, typename Def > diccionario< Key, Def >::iterator diccionario< Key, Def >::find (const Key & *s*)

busca una cadena en el diccionario

Parameters

s	cadena a buscar

Returns

un iterador que apunta a la entrada en el diccionario. Si la palabra s no se encuentra devuelve end()

Postcondition

no modifica el diccionario.

Todo implementa esta función

3.2.2.6 template<typename Key , typename Def > diccionario< Key, Def > & diccionario< Key, Def >::operator= (const diccionario< Key, Def > & *org*)

operador de asignación

Parameters

in	*org*	diccionario a copiar. Crea un diccionario duplicado exacto de org.

3.2.2.7 template<typename Key, typename Def > Def & diccionario< Key, Def >::operator[] (const Key & *s*)

busca una cadena en el diccionario

```
@param s cadena a buscar
@return un const_iterador que apunta a la entrada en el diccionario. Si la palabra s no se encuentra devue
@post no modifica el diccionario.
const_iterator  diccionario<K,D>::find( const Key & s) const;
```

/** Consulta/Inserta una entrada en el diccionario.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia al numero de ocurrencias de la misma en caso contrario la inserta, con frecuencia cero, devolviendo una referencia a este valor.

Parameters

in	*s*	cadena a insertar
out	*int*	& referencia a la definicion asociada a la entrada

Postcondition

Si s no esta en el diccionario, el size() sera incrementado en 1.

3.2.2.8 template<typename Key, typename Def > const Def & diccionario< Key, Def >::operator[] (const Key & *s*) const

Consulta una entrada en el diccionario.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia constante al numero de ocurrencias de la misma, si no la encuentra da un mensaje de error.

Parameters

in	*s*	cadena a insertar
out	*int*	& referencia constante a la definicion asociada a la entrada

Postcondition

No se modifica el diccionario.

3.2.2.9 template<typename Key , typename Def > diccionario< Key, Def >::size_type diccionario< Key, Def >::size ( ) const

numero de entradas en el diccionario

Postcondition

No se modifica el diccionario.

3.2.3 Friends And Related Function Documentation

3.2.3.1 template<typename Key = string, typename Def = int> ostream& operator<< (ostream & *sal,* const diccionario< Key, Def > & *D*) [friend]

imprime todas las entradas del diccionario

Postcondition

No se modifica el diccionario.

Todo implementar esta funcion

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

- diccionario.h
- diccionarioV1.hxx

3.3 diccionario< Key, Def >::iterator Class Reference

class iterator forward iterador sobre el diccionario, Lectura y Escritura iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!=

`#include <diccionario.h>`

Public Member Functions

- iterator ()
- **iterator** (const iterator &it)
- entrada & operator* () const
- iterator operator++ (int)
- iterator & operator++ ()
- bool operator== (const iterator &it)
- bool operator!= (const iterator &it)
- iterator & **operator=** (const iterator &it)

Friends

- class **diccionario**

3.3.1 Detailed Description

template<typename Key = string, typename Def = int>class diccionario< Key, Def >::iterator

class iterator forward iterador sobre el diccionario, Lectura y Escritura iterator ,operator*, operator++, operator++(int) operator=, operator==, operator!=

3.3.2 Constructor & Destructor Documentation

3.3.2.1 template<typename Key , typename Def > diccionario< Key, Def >::iterator::iterator ( )

Todo implementa esta función

3.3.3 Member Function Documentation

3.3.3.1 template<typename Key = string, typename Def = int> bool diccionario< Key, Def >::iterator::operator!= (const iterator & *it*)

Todo implementa esta función

3.3.3.2 template<typename Key , typename Def > diccionario< Key, Def >::entrada & diccionario< Key, Def >::iterator::operator* ( ) const

Todo implementa esta función

3.3.3.3 template<typename Key , typename Def > diccionario< Key, Def >::iterator diccionario< Key, Def >::iterator::operator++ (int *a*)

Todo implementa esta función

3.3.3.4 template<typename Key , typename Def > diccionario< Key, Def >::iterator & diccionario< Key, Def >::iterator::operator++ ( )

Todo implementa esta función

3.3.3.5 template<typename Key = string, typename Def = int> bool diccionario< Key, Def >::iterator::operator== (const iterator & *it*)

Todo implementa esta función

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

- diccionario.h
- diccionarioV1.hxx

Index

begin
    diccionario, 7

diccionario
    begin, 7
    diccionario, 7
    end, 8
    find, 8
    operator<<, 10
    operator=, 8
    size, 9
diccionario< Key, Def >, 5
diccionario< Key, Def >::const_iterator, 5
diccionario< Key, Def >::iterator, 10
diccionario::iterator
    iterator, 10
    operator*, 11
    operator++, 11
    operator==, 11

end
    diccionario, 8

find
    diccionario, 8

iterator
    diccionario::iterator, 10

operator<<
    diccionario, 10
operator*
    diccionario::iterator, 11
operator++
    diccionario::iterator, 11
operator=
    diccionario, 8
operator==
    diccionario::iterator, 11

size
    diccionario, 9