Práctica 2 Estructuras de Datos

Generated by Doxygen 1.8.8

Sun Nov 2 2014 21:01:30

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 defM Class Reference

Public Member Functions

- **defM** ()

  constructor primitivo.
- **defM** (const **defM** &x)

  constructor de copia
- void **setCode** (const string &s)

  Guarda el código de un meteorito.
- void **setMas** (const double &m)

  Guarda la masa del meteorito.
- void **setFall** (bool f)

  Guarda si el meteorito ha caído o no.
- void **setYear** (const string &y)

  Guarda el año del meteorito.
- void **setLat** (double &lat)

  Guarda la latitud del meteorito.
- void **setLong** (double &lon)

  Guarda la longitud del meteorito.
- vector< string > **getCodes** ()

  Proporciona los códigos de todos los meteoritos meteorito.
- double **getMas** ()

  Proporciona la masa del meteorito.
- bool **getFall** ()

  Proporciona si ha caído o no el meteorito.
- string **getYear** ()

  Proporciona el año del meteorito.
- double **getLat** ()

  Proporciona la latitud del meteorito.
- double **getLong** ()

  Proporciona la longitud del meteorito.

Friends

- ostream & **operator**<< (ostream &, const **defM** &)

2.1.1 Constructor & Destructor Documentation

2.1.1.1 defM::defM ( )

constructor primitivo.

Postcondition

definición por defecto.

2.1.1.2 defM::defM (const defM & *x*)

constructor de copia

Parameters

in	*x*	definición a copiar.

2.1.2 Member Function Documentation

2.1.2.1 vector< string > defM::getCodes ( )

Proporciona los códigos de todos los meteoritos meteorito.

Parameters

out	*vector<string>*	codigos

2.1.2.2 bool defM::getFall ( )

Proporciona si ha caído o no el meteorito.

Parameters

out	*fall*	bool

2.1.2.3 double defM::getLat ( )

Proporciona la latitud del meteorito.

Parameters

out	*latitud*	double

2.1.2.4 double defM::getLong ( )

Proporciona la longitud del meteorito.

Parameters

out	*longitud*	double

2.1.2.5 double defM::getMas ( )

Proporciona la masa del meteorito.

Parameters

out	*mas*	double

2.1.2.6 string defM::getYear ( )

Proporciona el año del meteorito.

Parameters

out	*year*	string

2.1.2.7 void defM::setCode (const string & *s*)

Guarda el código de un meteorito.

Parameters

in	*s*	cadena de caracteres

2.1.2.8 void defM::setFall (bool *f*)

Guarda si el meteorito ha caído o no.

Parameters

in	*f*	bool

2.1.2.9 void defM::setLat (double & *lat*)

Guarda la latitud del meteorito.

Parameters

in	*lat*	double

2.1.2.10 void defM::setLong (double & *lon*)

Guarda la longitud del meteorito.

Parameters

in	*long*	double

2.1.2.11 void defM::setMas (const double & *m*)

Guarda la masa del meteorito.

Parameters

in	*m*	double

2.1.2.12 void defM::setYear (const string & *y*)

Guarda el año del meteorito.

Parameters

in	*y*	string

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

- meteorito.h
- meteorito.hxx

2.2 diccionario Class Reference

Public Types

- typedef pair< nombreM, **defM** > **entrada**
- typedef unsigned int **size_type**

Public Member Functions

- **diccionario** ()
  constructor primitivo.
- **diccionario** (const **diccionario** &d)
  constructor de copia
- pair< entrada, bool > **find** (const nombreM &s) const
  busca una meteorito en el diccionario
- bool **insert** (const entrada &e)
  Inserta una entrada en el diccionario.
- **defM** & **operator[]** (const nombreM &s)
  Consulta/Inserta una entrada en el diccionario.
- const **defM** & **operator[]** (const nombreM &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 diccionario esta vacio (**size()** (p. 9)==0)*

Friends

- ostream & **operator**<< (ostream &, const **diccionario** &)

2.2.1 Constructor & Destructor Documentation

2.2.1.1 diccionario::diccionario ( )

constructor primitivo.

Método constructor por defecto Se crea un diccionario vacío.

Postcondition

define la entrada nula como el par ("",-1)
Se modifica el diccionario

2.2.1.2 diccionario::diccionario (const diccionario & *d*)

constructor de copia

Método constructor.

Parameters

in	*d*	diccionario a copiar

Crea una copia del diccionario que nos facilitan sin ordenar

Parameters

in	*diccionario*	& d referencia a un diccionario

Postcondition

No se modifica el diccionario

Crea una copia del diccionario que nos facilitan ordenándolo de menor a mayor mediante el método de ordenación por burbuja.

Parameters

in	*diccionario*	& d referencia a un diccionario

Postcondition

Se modifica el diccionario

2.2.2 Member Function Documentation

2.2.2.1 bool diccionario::empty ( ) const

vacia Chequea si el diccionario esta vacio (**size()** (p. 9)==0)

2.2.2.2 pair< diccionario::entrada, bool > diccionario::find (const nombreM & *s*) const

busca una meteorito en el diccionario

Método buscar Busca un meteorito en el diccionario.

Parameters

s	cadena a buscar

Returns

una copia de la entrada en el diccionario. Si no se encuentra devuelve la entrada con la definicion por defecto

Postcondition

no modifica el diccionario.

Uso

```
if (D.find("aaa").second ==true) cout << "Esta" ;
else cout << "No esta";
```

Parameters

in	*const*	nombreM & s referencia a una cadena constante a buscar

Returns

una copia de la entrada en el diccionario en caso de encontrarla. En caso de no encontrarla, devuelve la entrada con la definición por defecto.

Postcondition

no modifica el diccionario

2.2.2.3 bool diccionario::insert (const entrada & *e*)

Inserta una entrada en el diccionario.

Inserta una entrada en el diccionario Para insertar una entrada en nuestro diccionario de meteoritos primero llama a la función buscar para que, en caso de que ya se encuentre un meteorito con la misma clave, no lo inserte sino que lo descarte.

Parameters

e	entrada a insertar

Returns

true si la entrada se ha podido insertar con éxito, esto es, no existe un meteorito con igual nombre en el diccionario. False en caso contrario

Postcondition

Si e no esta en el diccionario, el **size()** (p. 9) sera incrementado en 1.

2.2.2.4 diccionario & diccionario::operator= (const diccionario & *org*)

operador de asignación

Parameters

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

2.2.2.5 defM & diccionario::operator[] (const nombreM & *s*)

Consulta/Inserta una entrada en el diccionario.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia a la definición de la misma en caso contrario la inserta, con una definición por defecto, devolviendo una referencia a este valor.

Parameters

in	*s*	cadena a insertar
out	***defM*** *(p. 3)*	& referencia a la definicion asociada a la entrada, nos permite modificar la definición

Postcondition

Si s no esta en el diccionario, el **size()** (p. 9) sera incrementado en 1.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia a la definición de la misma. En caso contrario, la inserta, con una definición por defecto, devolviendo la referencia a este valor.

Parameters

in	*const*	nombreM & s referencia a una cadena constante a consultar
out	***defM*** *(p. 3)*	& referencia a la definición asociada a la entrada, nos permite modificar la definición.

Postcondition

si s no está en el diccionario el **size()** (p. 9) se incrementa en 1.

2.2.2.6 const defM & diccionario::operator[ ] (const nombreM & *s*) const

Consulta una entrada en el diccionario.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia constante aa la definición 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.

Busca la cadena s en el diccionario, si la encuentra devuelve una referencia constante a la definición de la misma. En caso contrario, lanza un mensaje de error.

Parameters

in	*const*	nombreM & s referencia a una cadena constante a consultar
out	*int*	& referencia constante a la definición asociada a la entrada.

Postcondition

no se modifica el diccionario

2.2.2.7 diccionario::size_type diccionario::size ( ) const

numero de entradas en el diccionario

Postcondition

No se modifica el diccionario.

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

- diccionario.h
- diccionarioV1.hxx
- diccionarioV2.hxx