

Práctica 2. Pilas y Colas

Generated by Doxygen 1.9.8

1 Pilas como colas y colas como pilas.	1
1.1 Introducción	1
1.2 Ejecutables	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 elemento Struct Reference	7
4.1.1 Member Function Documentation	7
4.1.1.1 toString()	7
4.2 MaxQueue Class Reference	8
4.2.1 Member Function Documentation	8
4.2.1.1 empty()	8
4.2.1.2 front()	8
4.2.1.3 push()	8
4.2.1.4 size()	9
4.2.1.5 swap()	9
4.3 MaxStack Class Reference	9
4.3.1 Member Function Documentation	10
4.3.1.1 empty()	10
4.3.1.2 size()	10
4.3.1.3 top()	10
5 File Documentation	11
5.1 include/elemento.h File Reference	11
5.1.1 Detailed Description	11
5.1.2 Function Documentation	11
5.1.2.1 operator<<() [1/2]	11
5.1.2.2 operator<<() [2/2]	12
5.2 elemento.h	12
5.3 include/maxqueue.h File Reference	13
5.3.1 Detailed Description	13
5.4 maxqueue.h	13
5.5 include/maxstack.h File Reference	14
5.5.1 Detailed Description	14
5.6 maxstack.h	14
5.7 src/elemento.cpp File Reference	14
5.7.1 Detailed Description	15
5.8 src/maxqueue.cpp File Reference	15
5.8.1 Detailed Description	15

5.9 src/maxstack.cpp File Reference	15
5.9.1 Detailed Description	15
Index	17

Chapter 1

Pilas como colas y colas como pilas.

1.1 Introducción

En este proyecto se ha propuesto realizar la estructura de las pilas usando colas y al contrario. Además en cada posición de la cola o de la pila debe aparecer su valor junto a el máximo.

Como estas estructuras de datos permiten almacenar cualquier tipo de dato que se le indique, hemos creado un objeto llamado *elemento* que almacena dos enteros, el primero el valor de esa posición de la pila o cola y el segundo el máximo valor.

1.2 Ejecutables

cola_max <Valor>

Parameters

< Valor>	Aceptará dos opciones: Número entero que se insertará en la cola o "." que mostrará el elemento del frente de la cola y lo eliminará.
----------	--

pila_max <Valor>

Parameters

< Valor>	Aceptará dos opciones: Número entero que se insertará en la pila o "." que mostrará el elemento de lo mas alto de la pila y lo eliminará.
----------	--

Chapter 2

Class Index

2.1 Class List

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

elemento	7
MaxQueue	8
MaxStack	9

Chapter 3

File Index

3.1 File List

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

include/ elemento.h	Cabecera del struct elemento usado para organizar el valor y el máximo en las pilas y colas	11
include/ maxqueue.h	Archivo de especificación del TDA MaxQueue	13
include/ maxstack.h	Archivo de especificación del TDA MaxStack	14
src/ elemento.cpp	Este fichero tiene la implementación de los métodos del struct elemento	14
src/ maxqueue.cpp	Archivo de implementación del TDA MaxQueue	15
src/ maxstack.cpp	Archivo de implementación del TDA MaxStack	15

Chapter 4

Class Documentation

4.1 elemento Struct Reference

Public Member Functions

- **elemento** (int valor, int maximo)
Constructor de la estructura.
- string [toString](#) () const
este método pasa a una cadena los dos valores de la estructura.

Public Attributes

- int **valor**
- int **maximo**

4.1.1 Member Function Documentation

4.1.1.1 toString()

```
string elemento::toString ( ) const
```

este método pasa a una cadena los dos valores de la estructura.

Returns

La cadena con el valor y el máximo.

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

- include/[elemento.h](#)
- src/[elemento.cpp](#)

4.2 MaxQueue Class Reference

Public Member Functions

- const [elemento](#) & [front](#) () const
Este método devuelve la referencia al primer elemento de la cola.
- void [pop](#) ()
Este método borra el primer elemento de la cola.
- void [push](#) (int val)
Este método inserta un nuevo valor en la última posición.
- bool [empty](#) () const
Método que verifica si la cola está vacía.
- std::stack< [elemento](#) >::size_type [size](#) () const
Este método nos indica el tamaño de la cola.
- void [swap](#) (MaxQueue &queue)
Este método intercambia dos [MaxQueue](#).

4.2.1 Member Function Documentation

4.2.1.1 empty()

```
bool MaxQueue::empty ( ) const
```

Método que verifica si la cola está vacía.

Returns

true si está vacía false en caso contrario.

4.2.1.2 front()

```
const elemento & MaxQueue::front ( ) const
```

Este método devuelve la referencia al primer elemento de la cola.

Returns

referencia constante al elemento en la primera posición.

4.2.1.3 push()

```
void MaxQueue::push (  
    int val )
```

Este método inserta un nuevo valor en la última posición.

Parameters

<i>val</i>	Valor a insertar.
------------	-------------------

4.2.1.4 size()

```
std::stack< elemento >::size_type MaxQueue::size ( ) const
```

Este método nos indica el tamaño de la cola.

Returns

El tamaño de la cola, respetando los tipos de variables de la stl.

4.2.1.5 swap()

```
void MaxQueue::swap (
    MaxQueue & queue )
```

Este método intercambia dos [MaxQueue](#).

Parameters

<i>queue</i>	Con esta cola hace el intercambio la cola que llama al método
--------------	---

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

- [include/maxqueue.h](#)
- [src/maxqueue.cpp](#)

4.3 MaxStack Class Reference

Public Member Functions

- const [elemento](#) & [top](#) ()
Este metodo devuelve el elemento en lo alto de la pila.
- void **pop** ()
Método que elimina el elemento de lo alto de la pila.
- void **push** (int val)
Método para añadir un elemento a la pila @elem val Elemento a añadir.
- bool [empty](#) () const
Método que verifica si la cola está vacía.
- std::queue< [elemento](#) >::size_type [size](#) ()
Este método nos indica el tamaño de la pila.

4.3.1 Member Function Documentation

4.3.1.1 empty()

```
bool MaxStack::empty ( ) const
```

Método que verifica si la cola está vacía.

Returns

true si está vacía false en caso contrario.

4.3.1.2 size()

```
std::queue< elemento >::size_type MaxStack::size ( )
```

Este método nos indica el tamaño de la pila.

Returns

El tamaño de la pila, respetando los tipos de variables de la stl.

4.3.1.3 top()

```
const elemento & MaxStack::top ( )
```

Este metodo devuelve el elemento en lo alto de la pila.

Returns

Referencia constante al elemento de lo alto de la pila

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

- include/[maxstack.h](#)
- src/[maxstack.cpp](#)

Chapter 5

File Documentation

5.1 include/elemento.h File Reference

Cabecera del struct elemento usado para organizar el valor y el máximo en las pilas y colas.

```
#include <iostream>
```

Classes

- struct [elemento](#)

Functions

- ostream & [operator<<](#) (ostream &os, const [elemento](#) &elem)
Esta función sobrecarga el operador << para aceptar un elemento en cualquier salida de flujo.
- ostream & [operator<<](#) (ostream &os, [elemento](#) &elem)
Esta función sobrecarga el operador << para aceptar un elemento en cualquier salida de flujo.

5.1.1 Detailed Description

Cabecera del struct elemento usado para organizar el valor y el máximo en las pilas y colas.

5.1.2 Function Documentation

5.1.2.1 operator<<() [1/2]

```
ostream & operator<< (  
    ostream & os,  
    const elemento & elem )
```

Esta función sobrecarga el operador << para aceptar un elemento en cualquier salida de flujo.

Parameters

<i>os</i>	Representa la salida de los datos.
<i>elem</i>	Referencia constante del elemento a introducir en el flujo.

Returns

Se devuelve

Parameters

<i>os</i>	para poder concatenar entradas de flujo.
-----------	--

5.1.2.2 operator<<() [2/2]

```
ostream & operator<< (
    ostream & os,
    elemento & elem )
```

Esta función sobrecarga el operador << para aceptar un elemento en cualquier salida de flujo.

Parameters

<i>os</i>	Representa la salida de los datos.
<i>elem</i>	Referencia del elemento a introducir en el flujo.

Returns

Se devuelve

Parameters

<i>os</i>	para poder concatenar entradas de flujo.
-----------	--

5.2 elemento.h

[Go to the documentation of this file.](#)

```
00001
00006 #ifndef P03_PILAS_COLAS_ELEMENTO_H
00007 #define P03_PILAS_COLAS_ELEMENTO_H
00008
00009 #include <iostream>
00010
00011 using namespace std;
00012
00013 struct elemento{
00014     int valor;
00015     int maximo;
00016
00020     elemento(int valor, int maximo){
00021         this->valor = valor;
```



```

00022         this->maximo = maximo;
00023     }
00028     string toString() const;
00029 };
00030
00031
00038 ostream& operator<<(ostream &os, const elemento &elem);
00039
00046 ostream& operator<<(ostream &os, elemento &elem);
00047 #endif //P03_PILAS_COLAS_ELEMENTO_H

```

5.3 include/maxqueue.h File Reference

Archivo de especificación del TDA [MaxQueue](#).

```

#include <stack>
#include "elemento.h"

```

Classes

- class [MaxQueue](#)

5.3.1 Detailed Description

Archivo de especificación del TDA [MaxQueue](#).

Author

Iván Rodríguez Chacón

5.4 maxqueue.h

[Go to the documentation of this file.](#)

```

00001
00006 #include <stack>
00007 #include "elemento.h"
00008
00009 class MaxQueue{
00010 private:
00011     std::stack<elemento> pila;
00012 public:
00017     const elemento& front() const;
00018
00022     void pop();
00023
00028     void push(int val);
00029
00034     bool empty() const;
00035
00040     std::stack<elemento>::size_type size() const;
00041
00046     void swap(MaxQueue& queue);
00047 };

```

5.5 include/maxstack.h File Reference

Archivo de especificación del TDA [MaxStack](#).

```
#include <queue>
#include "elemento.h"
```

Classes

- class [MaxStack](#)

5.5.1 Detailed Description

Archivo de especificación del TDA [MaxStack](#).

Author

Javier Lama Jiménez

5.6 maxstack.h

[Go to the documentation of this file.](#)

```
00001
00007 #include <queue>
00008 #include "elemento.h"
00009
00010 class MaxStack{
00011 private:
00012     std::queue<elemento> queue;
00013
00014 public:
00019     const elemento& top();
00020
00024     void pop();
00025
00030     void push(int val);
00031
00036     bool empty() const;
00037
00042     std::queue<elemento>::size_type size() ;
00043
00044 };
```

5.7 src/elemento.cpp File Reference

Este fichero tiene la implementación de los métodos del struct elemento.

```
#include "elemento.h"
```

Functions

- ostream & **operator**<< (std::ostream &os, const [elemento](#) &elem)
- ostream & **operator**<< (std::ostream &os, [elemento](#) &elem)

5.7.1 Detailed Description

Este fichero tiene la implementación de los métodos del struct elemento.

5.8 src/maxqueue.cpp File Reference

Archivo de implementación del TDA [MaxQueue](#).

```
#include "maxqueue.h"
```

5.8.1 Detailed Description

Archivo de implementación del TDA [MaxQueue](#).

Author

Iván Rodríguez Chacón

5.9 src/maxstack.cpp File Reference

Archivo de implementación del TDA [MaxStack](#).

```
#include "maxstack.h"
```

5.9.1 Detailed Description

Archivo de implementación del TDA [MaxStack](#).

Author

Javier Lama Jiménez

Index

- elemento, [7](#)
 - toString, [7](#)
- elemento.h
 - operator<<, [11](#), [12](#)
- empty
 - MaxQueue, [8](#)
 - MaxStack, [10](#)
- front
 - MaxQueue, [8](#)
- include/elemento.h, [11](#), [12](#)
- include/maxqueue.h, [13](#)
- include/maxstack.h, [14](#)
- MaxQueue, [8](#)
 - empty, [8](#)
 - front, [8](#)
 - push, [8](#)
 - size, [9](#)
 - swap, [9](#)
- MaxStack, [9](#)
 - empty, [10](#)
 - size, [10](#)
 - top, [10](#)
- operator<<
 - elemento.h, [11](#), [12](#)
- Pilas como colas y colas como pilas., [1](#)
- push
 - MaxQueue, [8](#)
- size
 - MaxQueue, [9](#)
 - MaxStack, [10](#)
- src/elemento.cpp, [14](#)
- src/maxqueue.cpp, [15](#)
- src/maxstack.cpp, [15](#)
- swap
 - MaxQueue, [9](#)
- top
 - MaxStack, [10](#)
- toString
 - elemento, [7](#)