queue<T>

Categorías: contenedor **Tipo componente**: tipo

Descripción

`Una cola (queue)` es un contenedor que proporciona un subconjunto restringido de métodos con la funcionalidad de "primero en entrar, primero en salir" (FIFO – First In First Out). Es decir, los elementos se agregan a la parte posterior de la `queue` y se pueden quitar del frente; el frente (`front`) es el elemento que fue insertado en el `queue` lo menos recientemente. `La queue` no permite la iteración a través de sus elementos.

Ejemplo

```
int main() {

  queue<int> Q;
  Q.push(8);
  Q.push(7);
  Q.push(6);
  Q.push(2);

  assert(Q.size() == 4);
  assert(Q.back() == 2);

  assert(Q.front() == 8);
  Q.pop();

  assert(Q.front() == 7);
  Q.pop();

  assert(Q.front() == 6);
  Q.pop();

  assert(Q.front() == 2);
  Q.pop();

  assert(Q.empty());
}
```

Definición

Definido en la queue estándar del jefe, y en el jefe anormal stack.h. de la al revés-compatibilidad.

Parámetros de la plantilla

Parámetro	Descripción	Defecto
T	El tipo de objeto almacenado en el queue.	

Requerimientos del Tipo

- T es un tipo que tiene el operador de asignación
- Si se utiliza el `operator==`, entonces `T` debe tener definido el operador de igualdad
- Si se utiliza el `operator<`, entonces `T` debe tener definido el operador menor_que

Miembros

Miembro	Descripción
`size_type`	Ver abajo.
`queue ()`	El constructor por defecto. Crea una `queue` vacía.
`queue (const queue& )`	El constructor de copia.
`Queue & operator=(const queue& )`	El operador de asignación.
`Bool empty () const`	Devuelve `verdad` si el queue no contiene ningún elemento, y `falso` en caso contrario. `Q.empty ()` es equivalente Q.size() `==` `0`
`size_type size ()const`	Devuelve el número de los elementos contenidos en el `queue`.
`T & front ()`	Devuelve una referencia al elemento en el frente de la cola, el más antiguo Pre-Condición: `vacío ()` es `falso`.
`const T & front() const`	Devuelve una referencia constante al elemento en el frente de la cola, el más antiguo Pre.-Condición: `vacío ()` es `falso`.
`T & back()`	Devuelve una referencia al elemento en la parte trasera de la cola, el último insertado Pre-Condición: `vacío ()` es `falso`.
`const T & back() const`	Devuelve una referencia constante al elemento en la parte trasera de la cola, el último insertado Pre.-Condición: `vacío ()` es `falso`.
`Void push (const T & x)`	Insertamos `x` en la parte trasera de la queue. Postcondiciones: `el tamaño ()` será incrementado en `1`, y el back `()` será igual al `X`.

`Void pop()`	Quita el elemento en el frente de la cola. Pre-Condición: `vacío ()` es `falso`. Post-condiciónon: `el tamaño ()` decremento en `1`.
`Bool operator==(const queue&, const queue &)`	Compara si dos queue son iguales. Dos queue son iguales si contienen el mismo número de elementos y si son iguales elemento-a-elemento. Es una función global, no una función miembro.
`Boool operator<(const queue&, const queue &)`	Compara utilizando un orden lexicográfica dos queue. Es una función global, no una función del miembro.