

stack<T>

Categorías: Contenedor

Tipo componente: tipo

Descripción

Una Pila (**stack**) es un contenedor que proporciona un subconjunto restringido de métodos: inserción, el borrado y la consulta del elemento que se sitúa en el tope de la pila. El **stack** es una estructura de datos que sigue la filosofía “último en entrar, primero en salir” (LIFO -- Last In First Out): el elemento en el tope del **stack** es aquel que fue insertado el último. El **stack** no permite la iteración a través de sus elementos.

Ejemplo

```
int main() {  
  
    stack<int> S;  
    S.push(8);  
    S.push(7);  
    S.push(4);  
    assert(S.size() == 3);  
  
    assert(S.top() == 4);  
    S.pop();  
  
    assert(S.top() == 7);  
    S.pop();  
  
    assert(S.top() == 8);  
    S.pop();  
  
    assert(S.empty());  
}
```

Definición

Definido en el fichero cabecera <stack,> y en <stack.h> para compatibilidad con versiones anteriores.

Parámetros de plantilla

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

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
<code>size_type</code>	Un <code>unsigned int</code>
<code>stack ()</code>	El constructor por defecto. Crea un <code>stack</code> vacío.
<code>stack (const stack &)</code>	El constructor de copia.
<code>Stack & operator=(const stack&)</code>	El operador de asignación.
<code>Bool empty () const</code>	Devuelve verdad si el <code>stack</code> no contiene ningún elemento, y falso en caso contrario. <code>S.empty ()</code> es equivalente <code>S.size() == 0</code>
<code>size_type size() const</code>	Devuelve el número de los elementos contenidos en el <code>stack</code> .
<code>T & top()</code>	Devuelve una referencia al elemento en el tope del <code>stack</code> . Pre-Condición: vacío () es falso.
<code>const T & top() const</code>	Devuelve una referencia constante al elemento en en tope del <code>stack</code> . Pre.-Condición: vacío () es falso.
<code>Void push(const T & x)</code>	Insertamos <code>X</code> en el tope del <code>stack</code> . Postcondiciones: el tamaño () será incrementado en 1, y el top () será igual al X.
<code>Void pop()</code>	Quita el elemento en el tope del <code>stack</code> . Pre-Condición: vacío () es falso. Post-condiciónon: el tamaño () decremento en 1.
<code>Bool operator==(const stack&, const stack &)</code>	Compara si dos <code>stacks</code> son iguales. Dos <code>stacks</code> son iguales si contienen el mismo número de elementos y si son iguales elemento-a-elemento. Es una función global, no una función miembro.
<code>Bool operator<(const stack&, const stack &)</code>	Compara utilizando un orden lexicográfica dos <code>stacks</code> . Es una función global, no una función del miembro.