**Класс-шаблон list**

Класс-шаблон описывает объект, который контролирует изменение длины последовательности элементов, имеющей двунаправленный доступ. Используется контейнер **list** для управления последовательность элементов в качестве двунаправленного связанного списка узлов, каждый расположении один элемент.

|  |  |
| --- | --- |
| **Функция Member** | **Описание** |
| [list::assign (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398048.aspx) | Заменяет все элементы. |
| [list::back (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385219.aspx) | Получает последний элемент. |
| [list::begin (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385234.aspx) | Задает начало управляемой последовательности. |
| [list::clear (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385478.aspx) | Удаляет все элементы. |
| [list::empty (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386256.aspx) | Тесты отсутствуют ли какие-либо элементы. |
| [list::end (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398036.aspx) | Задает конец управляемой последовательности. |
| [list::erase (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb157756.aspx) | Удаляет элементы в указанных позициях. |
| [list::front (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386132.aspx) | Получение первого элемента. |
| [list::insert (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385250.aspx) | Добавляет элементы в указанной позиции. |
| [list::list (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385353.aspx) | Создает объект контейнера. |
| [list::merge (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386283.aspx) | Объединяет две упорядоченные управляемые последовательности. |
| [list::pop\_back (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385001.aspx) | Удаляет последний элемент. |
| [list::pop\_front (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb157669.aspx) | Удаляет первый элемент. |
| [list::push\_back (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386250.aspx) | Добавляет новый последний элемент. |
| [list::push\_front (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385303.aspx) | Добавляет новый первый элемент. |
| [list::rbegin (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385813.aspx) | Задает начало обратной управляемой последовательности. |
| [list::remove (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386133.aspx) | Удаляет элемент с указанным значением. |
| [list::remove\_if (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398083.aspx) | Удаляет элементы, прошедшие определенного теста. |
| [list::rend (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386051.aspx) | Задает конец обратной управляемой последовательности. |
| [list::resize (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398042.aspx) | Изменяет количество элементов. |
| [list::reverse (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398191.aspx) | Возвращает контролируемую последовательность. |
| [list::size (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385284.aspx) | Подсчитывает количество элементов. |
| [list::sort (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386274.aspx) | Упорядочивает управляемую последовательность. |
| [list::splice (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386161.aspx) | Ссылки Restitches между узлами. |
| [list::swap (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385085.aspx) | Меняет местами содержимое двух контейнеров. |
| [list::to\_array (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385273.aspx) | Копирует управляемую последовательность в новый массив. |
| [list::unique (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398038.aspx) | Удаляет смежные элементы, которые прошли заданный тест. |

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| [list::back\_item (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb157615.aspx) | Получает последний элемент. |
| [list::front\_item (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398075.aspx) | Получение первого элемента. |
| **operator** | **Описание** |
| [list::operator= (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385042.aspx) | Заменяет контролируемая последовательность. |
| [operator!= (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385959.aspx) | Определяет, является ли объект **list** не равен другому объекту **list**. |
| [operator< (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb157667.aspx) | Определяет, является ли объект **list** меньше другой объект **list**. |
| [operator<= (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386011.aspx) | Определяет, является ли объект **list** меньше или равно другому объекту **list**. |
| [operator== (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb386028.aspx) | Определяет, является ли объект **list** равен другому объекту **list**. |
| [operator> (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385461.aspx) | Определяет, является ли объект **list** больше другого объекта **list**. |
| [operator>= (list) (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398069.aspx) | Определяет, является ли объект **list** больше или равно другому объекту **list**. |

[Заметки](javascript:void(0))

Объект выделяет и освобождает хранилище для последовательности его элементы управления, как отдельные узлы в двунаправленной связи ".Он выполняет переупорядочивание элементов, изменив ссылки между узлами, не скопировать содержимое одного узла на другой. Это означает, что можно вставлять и удалять элементы свободно, не нарушая остальные элементы. Таким образом, список хорошим кандидатом для основного контейнера для шаблона класса шаблона класса [queue (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385855.aspx) или [stack (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb157681.aspx).

Объект list поддерживает двунаправленные итераторы. это означает, что можно выполнить с соседним элементами. Специальный головной узел соответствует итератору [list::begin (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb385234.aspx). Используется для получения итератора, который задает начало управляемой последовательности, однако в случае изменения длины управляемой последовательности его состояние может измениться.

Можно увеличивать такой итератор для достижения последнего элемента в контролируемой последовательности.

Итератор [list::end (STL/CLR)](https://msdn.microsoft.com/ru-ru/library/bb398036.aspx)() используется для получения итератора, обозначающего конец контролируемой последовательности. Можно уменьшать этот итератор для достижения головного узла.

Пример использования

#include <iostream>

#include <list> // подключаем заголовок списка

//#include <iterator> // заголовок итераторов

using namespace std;

int main()

{

list<int> myList; // объявляем пустой список

for(int i = 0; i < 11; i++) {

myList.push\_back(i); // добавляем в список новые элементы

}

cout << "Список: ";

int k=myList.size();

for (int i=1;i<=k;i++)

{

int n= myList.front(); //получаем первый элемент

myList.pop\_front(); // удаляем первый элемент

myList.push\_back(n); // добавляем в конец первый элемент

cout<<n<<endl;

}

myList.insert(myList.begin(),100); // добавляем в начало элемент 100

myList.insert(myList.end(),999); // добавляем в конец элемент 999

myList.remove(5); // удаляем 5

cout << " Измененный список: ";

k=myList.size();

for (int i=1;i<=k;i++)

{

int n= myList.front(); //получаем первый элемент

myList.pop\_front(); // удаляем первый элемент

myList.push\_back(n); // добавляем в конец первый элемент

cout<<n<<endl;

}

// вывод на экран элементов списка с помощью итератора вывода

// copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout," "));

cout << "\nПервый элемент списка: " << myList.front();

cout << "\nПоследний элемент списка: " << myList.back();

return 0;

}