Урок 11. C++. STL Стеки и очереди (stack and queue)

**Стеки (stack)**

Стеки - это множество элементов, сложенных в стопку. Например, у нас есть коробка 3х5. Мы кладем в нее доски такого же размера с различными узорами. У нас получается стек. Достать из него мы можем только начинаяя с последнего элемента, поэтому первый положенные элемент вынут будет последним. В стеках реализуется принцип first in last out (FILO).  
Для создания стека нужно подключить <stack> и в коде программы его объявить:  
stack <type> name, где type - тип стека, а name - имя стека.  
У стека есть немного функций:

* push() - добавить элемент
* pop() - удалить верхний элемент
* top() - получить верхний элемент
* size() - размер стека
* empty() - true, если стек пуст

Пример:string s;

stack <string> st;

while (cin>>s);

st.push(s);

while (!(st.empty()))

{cout<<st.top();st.pop()}

В этом примере мы считываем слова с клавиатури и выводим их в обратном порядке

**Очереди (Queue)**

Очереди, как следует из название, используют принцип first in first out (FIFO). То есть, тот, кого мы первым запихнули в очередь, первым из нее и выйдет (хотя в реальной жизни не всегда так....)  
Реализуются очереди также просто.  
Подключаем <queue>  
И создаем очередь  
queue <type> name;  
Перечень функций почти тот-же:

* push() - добавить элемент
* pop() - удалить первый элемент
* size() - размер очереди
* empty() - true, если очередь пуста
* front() - получить первый элемент
* back() - получить последний элемент

Пример:

queue <int>  events;

int n;

while (cin>>n)

events.push(n);

while (!events.empty)

{make(events.front());events.pop()}

В данном примере make - какая-то функция, обрабатывающая события.

# Класс deque

Упорядочивает элементы данного типа в линейном размещения и, как и вектор, обеспечивает быстрый прямой доступ к любому элементу, и эффективные вставки и удаления в обратно контейнера. Однако в отличие от вектор, класс **deque** также поддерживает эффективную вставки и удаления в начале контейнера.

template <

class Type,

class Allocator=allocator<Type>

>

class deque

## [Параметры](javascript:void(0))

*Type*

Тип данных элементов, сохраняемых в deque.

*Allocator*

Тип, представляющий сохраненный объект распределителя, инкапсулирующий информацию о выделении deque и освобождение памяти. Этот аргумент является необязательным и значение по умолчанию**allocator<Type>**.

## [Заметки](javascript:void(0))

Выбор типа контейнера должен быть в целом основан на типе поиска и вставки требуемом в приложении. [Столбцы](https://msdn.microsoft.com/ru-ru/library/9xd04bzs.aspx) должен быть предпочтительным контейнер для управления последовательность при прямой доступ к любому элементу в награде и вставках или удалениях элементов необходим только в конце последовательности. Производительность контейнера [list](https://msdn.microsoft.com/ru-ru/library/802d66bt.aspx) при главна эффективные вставки и удаления (при расчете времени) в любом месте внутри последовательности в награде. Такие операции в середине последовательности для копирования и назначения элемента пропорциональных на число элементов в последовательности (линейном времени).

Распространение Deque возникает, когда функцию-член должен вставка или привести к удалению элементов последовательности.

* Если элемент вставляется в пустую последовательность, или если элемент стерт, чтобы оставить пустую последовательность, то предыдущие итераторов возвращенные [begin](https://msdn.microsoft.com/ru-ru/library/z50625f7.aspx) и [end](https://msdn.microsoft.com/ru-ru/library/zz7bd328.aspx) являются недопустимыми.
* Если элемент вставляется в позиции первой deque, все итераторы, но отсутствуют ссылки, которые обозначают существующие элементы не будут недопустимыми.
* Если элемент вставляется в конце deque, затем [end](https://msdn.microsoft.com/ru-ru/library/zz7bd328.aspx) и всех итераторов, но нет ссылок, которые обозначают существующие элементы ознакомьтесь недопустимым.
* Если элемент стерт в начале deque, только такой итератор и ссылки на стертому элемент становятся недопустимыми.
* Если последний элемент стерт от конца deque, только такой итератор в последний элемент и ссылки на стертому элемент становятся недопустимыми.

В противном случае вставка или стирающ элемент и что все итераторы ссылки.

### 22a9t119.collapse_all(ru-ru,VS.120).gifконструкторов;

|  |  |
| --- | --- |
| [deque](https://msdn.microsoft.com/ru-ru/library/xexht24s.aspx) | Создает **deque.** предоставляют несколько конструкторов, чтобы настраивать содержимое нового **deque** различными  способами: пустой; загружается с указанным количеством пустых элементов; содержимое перемещены или копироваться  из другого **deque**; содержимое, скопированных или перемещенные с использованием итератора; скопированный элемент  и один раз в **deque***\_Count*.Некоторые конструкторы позволяют использование пользовательского **allocator** создание  элементов. |

### 22a9t119.collapse_all(ru-ru,VS.120).gifОпределения типов

|  |  |
| --- | --- |
| [allocator\_type](https://msdn.microsoft.com/ru-ru/library/f4df60zz.aspx) | Тип, представляющий класс **allocator** для объекта **deque**. |
| [const\_iterator](https://msdn.microsoft.com/ru-ru/library/c7t62atk.aspx) | Тип, который предоставляет произвольно-доступный итератор, может получать и чтение элементов в  **deque** как **const** |
| [const\_pointer](https://msdn.microsoft.com/ru-ru/library/e7dw47e0.aspx) | Тип, который содержит указатель элемент в **deque** как **const.** |
| [const\_reference](https://msdn.microsoft.com/ru-ru/library/yfydsbxz.aspx) | Тип, который предоставляет ссылку на элемент в **deque** для чтения и других операций в качестве **const.** |
| [const\_reverse\_iterator](https://msdn.microsoft.com/ru-ru/library/1e1ha123.aspx) | Тип, который предоставляет произвольно-доступный итератор, может получать и чтение элементов в  **deque** как **const**. Deque представляет в обратном порядке. Дополнительные сведения см. в разделе  [Класс reverse\_iterator](https://msdn.microsoft.com/ru-ru/library/6x256702.aspx). |
| [difference\_type](https://msdn.microsoft.com/ru-ru/library/fwc9ya3f.aspx) | Тип, который содержит различие между 2 произвольно-доступными итераторами, относящимися  к элементам в том же **deque**. |
| [iterator](https://msdn.microsoft.com/ru-ru/library/52wz3bxc.aspx) | Тип, который предоставляет произвольно-доступный итератор, который может считывать и изменять  любой элемент в **deque**. |
| [pointer](https://msdn.microsoft.com/ru-ru/library/9z5h6tc6.aspx) | Тип, который содержит указатель элемент в **deque**. |
| [Reference](https://msdn.microsoft.com/ru-ru/library/x8staf81.aspx) | Тип, который предоставляет ссылку на элемент хранящихся в **deque**. |
| [reverse\_iterator](https://msdn.microsoft.com/ru-ru/library/xhswkwy7.aspx) | Тип, который предоставляет произвольно-доступный итератор,  который может считывать и изменять элемент в **deque**. Deque просматриваемый в обратном порядке порядок. |
| [size\_type](https://msdn.microsoft.com/ru-ru/library/6xzd9atc.aspx) | Тип, который подсчитывает число элементов в **deque**. |
| [value\_type](https://msdn.microsoft.com/ru-ru/library/aw0z2z80.aspx) | Тип, представляющий тип данных, хранящихся в **deque**. |

### 22a9t119.collapse_all(ru-ru,VS.120).gifФункции-члены

|  |  |
| --- | --- |
| [assign](https://msdn.microsoft.com/ru-ru/library/0492k3bb.aspx) | Удаляет элементы из **deque** и копирует новую последовательность элементов к целевому объекту **deque**. |
| [at](https://msdn.microsoft.com/ru-ru/library/ez2f78e4.aspx) | Возвращает ссылку на элемент в указанном расположении в **deque**. |
| [back](https://msdn.microsoft.com/ru-ru/library/7a0asdxc.aspx) | Возвращает ссылку на последнему элементу **deque**. |
| [begin](https://msdn.microsoft.com/ru-ru/library/z50625f7.aspx) | Возвращает произвольно-доступный итератор слишком первый элемент в **deque**. |
| [deque::cbegin](https://msdn.microsoft.com/ru-ru/library/dd998429.aspx) | Возвращает итератор const на первый элемент в **deque**. |
| [deque::cend](https://msdn.microsoft.com/ru-ru/library/dd998373.aspx) | Возвращает произвольно-доступный итератор **const**, указывающий только за пределы **deque**. |
| [clear](https://msdn.microsoft.com/ru-ru/library/99ah58wb.aspx) | Удаляет все элементы **deque**. |
| [deque::crbegin](https://msdn.microsoft.com/ru-ru/library/dd984377.aspx) | Возвращает произвольно-доступный итератор const на первый элемент в порядке  просмотретьом **deque** в обратном порядке. |
| [deque::crend](https://msdn.microsoft.com/ru-ru/library/dd779310.aspx) | Возвращает произвольно-доступный итератор const на первый элемент в порядке  просмотретьом **deque** в обратном порядке. |
| [deque::emplace](https://msdn.microsoft.com/ru-ru/library/dd984342.aspx) | Вставляет элемент построен на месте в **deque** в указанной позиции. |
| [deque::emplace\_back](https://msdn.microsoft.com/ru-ru/library/dd984403.aspx) | Добавляет элемент, созданный на месте в конец **deque**. |
| [deque::emplace\_front](https://msdn.microsoft.com/ru-ru/library/dd984235.aspx) | Добавляет элемент, созданный на месте, чтобы **deque**. |
| [Empty](https://msdn.microsoft.com/ru-ru/library/f75d5w2z.aspx) | Возвращает **true** при **deque** содержит ноль элементов и **false**, если оно содержит один  или несколько элементов. |
| [end](https://msdn.microsoft.com/ru-ru/library/zz7bd328.aspx) | Возвращает произвольно-доступный итератор, указывающий только за пределы **deque**. |
| [erase](https://msdn.microsoft.com/ru-ru/library/c4x516tc.aspx) | Удаляет элемент или набор элементов в **deque** из заданных позиций. |
| [front](https://msdn.microsoft.com/ru-ru/library/s78e3as4.aspx) | Возвращает ссылку на первый элемент в **deque**. |
| [get\_allocator](https://msdn.microsoft.com/ru-ru/library/673dbata.aspx) | Возвращает копию объекта **allocator**, который используется для построения **deque**. |
| [insert](https://msdn.microsoft.com/ru-ru/library/zcww84w5.aspx) | Вставляет элемент, несколько элементов или диапазон элементов в **deque** в указанной позиции. |
| [max\_size](https://msdn.microsoft.com/ru-ru/library/17y0taey.aspx) | Возвращает максимально возможная длина **deque**. |
| [pop\_back](https://msdn.microsoft.com/ru-ru/library/0x0e3528.aspx) | Удаляет элемент в конце **deque**. |
| [pop\_front](https://msdn.microsoft.com/ru-ru/library/533z94y0.aspx) | Удаляет элемент в начале **deque**. |
| [push\_back](https://msdn.microsoft.com/ru-ru/library/7sf31cdc.aspx) | Добавляет элемент в конец **deque**. |
| [push\_front](https://msdn.microsoft.com/ru-ru/library/52wh801h.aspx) | Добавляет элемент к началу **deque**. |
| [rbegin](https://msdn.microsoft.com/ru-ru/library/y5awb5d9.aspx) | Возвращает произвольно-доступный итератор на первый элемент в обращенном **deque**. |
| [rend](https://msdn.microsoft.com/ru-ru/library/deczs23e.aspx) | Возвращает произвольно-доступный итератор, указывающий только за  последним элементом в обращенном **deque**. |
| [resize](https://msdn.microsoft.com/ru-ru/library/zxad4y7c.aspx) | Определяет новый размер для **deque**. |
| [deque::shrink\_to\_fit](https://msdn.microsoft.com/ru-ru/library/dd998399.aspx) | Отменяет резерв рабочей мощности. |
| [size](https://msdn.microsoft.com/ru-ru/library/t4x090h4.aspx) | Возвращает количество элементов в **deque**. |
| [swap](https://msdn.microsoft.com/ru-ru/library/b75t5aw0.aspx) | Меняет местами элементы 2 **deque** s. |

### 22a9t119.collapse_all(ru-ru,VS.120).gifОператоры

|  |  |
| --- | --- |
| [operator[]](https://msdn.microsoft.com/ru-ru/library/8azczwby.aspx) | Возвращает ссылку на элемент **deque** в указанной позиции. |
| [deque::operator=](https://msdn.microsoft.com/ru-ru/library/dd779343.aspx) | Заменяет элементы **deque** копией другого **deque**. |

## [Требования](javascript:void(0))

**Заголовок**: <deque>

# Класс list

**Visual Studio 2013**

[Другие версии](javascript:;)

Список классов STL является классом шаблона последовательностей контейнеров, которые поддерживают свои элементы в линейном упорядочение и позволяют эффективно вставлять и удалять элементы в любом месте в пределах последовательности. Последовательность сохраняется в виде двунаправленного связанного списка элементов, каждый из которых содержит член какого-либо типа Type.

C++

template < class Type, class Allocator=allocator<Type> > class list

## [Параметры](javascript:void(0))

Type

Тип данных элементов, сохраняемых в списке.

*Allocator*

Тип, представляющий сохраненный объект распределителя, содержащий сведения о распределении списка и отмене распределения памяти. Этот аргумент необязателен. Значение по умолчанию —**allocator**<Type>.

## [Заметки](javascript:void(0))

Выбор типа контейнера должен быть в целом основан на типе поиска и вставки требуемом в приложении. Векторы должны быть предпочитаемыми контейнерами для управления последовательностями, когда важен произвольный доступ к любому элементу, а вставка и удаление элементов требуется лишь в конце последовательности. Производительность контейнера класса двусторонней очереди выше, когда требуется произвольный доступ, а также вставка и удаление элементов и в начале последовательности, и в ее конце.

Функции-члены списка [merge](https://msdn.microsoft.com/ru-ru/library/ec0yw4kc.aspx), [reverse](https://msdn.microsoft.com/ru-ru/library/eteb1fdc.aspx), [unique](https://msdn.microsoft.com/ru-ru/library/thhhbacy.aspx), [remove](https://msdn.microsoft.com/ru-ru/library/kw666720.aspx) и [remove\_if](https://msdn.microsoft.com/ru-ru/library/9fyx5kx8.aspx) оптимизированы для объектов-списков и представляют собой быстродействующие альтернативы своим универсальным аналогам.

Перераспределение списка происходит, когда функция-член должна вставить или удалить элементы списка. Во всех таких случаях только итераторы или ссылки, указывающие на удаленные части последовательности, становятся недействительными .

Добавьте стандартный заголовок STL <list>, чтобы определить список классов шаблонов [контейнеров](https://msdn.microsoft.com/ru-ru/library/1fe2x6kt.aspx) и некоторые вспомогательные шаблоны.

### 802d66bt.collapse_all(ru-ru,VS.120).gifКонструкторы

|  |  |
| --- | --- |
| [list](https://msdn.microsoft.com/ru-ru/library/ee9w41zy.aspx) | Создает список определенного размера или с элементами конкретного значения или с определенным  **allocator** или как копия другого списка. |

### 802d66bt.collapse_all(ru-ru,VS.120).gifОпределения типов

|  |  |
| --- | --- |
| [allocator\_type](https://msdn.microsoft.com/ru-ru/library/ftdks0kx.aspx) | Тип, представляющий класс **allocator** для объекта списка. |
| [const\_iterator](https://msdn.microsoft.com/ru-ru/library/sff55syh.aspx) | Тип, предоставляющий двунаправленный итератор, который может читать элемент **const** в списке. |
| [const\_pointer](https://msdn.microsoft.com/ru-ru/library/swc4esxk.aspx) | Тип, предоставляющий указатель на элемент **const** в списке. |
| [const\_reference](https://msdn.microsoft.com/ru-ru/library/d68e4atb.aspx) | Тип, предоставляющий ссылку на элемент **const**, хранящийся в списке, для чтения и  выполнения операций **const**. |
| [const\_reverse\_iterator](https://msdn.microsoft.com/ru-ru/library/w4813e72.aspx) | Тип, предоставляющий двунаправленный итератор, который может читать любой элемент **const** в списке. |
| [difference\_type](https://msdn.microsoft.com/ru-ru/library/t849xtae.aspx) | Тип, предоставляющий разницу между двумя итераторами, ссылающимися на элементы в  одном и том же списке. |
| [iterator](https://msdn.microsoft.com/ru-ru/library/3yh6ac95.aspx) | Тип, предоставляющий двунаправленный итератор, который может читать или изменять  любой элемент в списке. |
| [pointer](https://msdn.microsoft.com/ru-ru/library/2sdc3t9h.aspx) | Тип, предоставляющий указатель на элемент в списке. |
| [Reference](https://msdn.microsoft.com/ru-ru/library/5akzk53z.aspx) | Тип, предоставляющий ссылку на элемент **const**, хранящийся в списке, для чтения и  выполнения операций **const**. |
| [reverse\_iterator](https://msdn.microsoft.com/ru-ru/library/e9c0wxsf.aspx) | Тип, предоставляющий двунаправленный итератор, который может читать или  изменять любой элемент в обратном списке. |
| [size\_type](https://msdn.microsoft.com/ru-ru/library/2579931d.aspx) | Тип, считающий количество элементов в списке. |
| [value\_type](https://msdn.microsoft.com/ru-ru/library/z5hzs9hc.aspx) | Тип, представляющий тип данных, хранящихся в списке. |

### 802d66bt.collapse_all(ru-ru,VS.120).gifФункции-члены

|  |  |
| --- | --- |
| [assign](https://msdn.microsoft.com/ru-ru/library/w3eba3dd.aspx) | Удаляет элементы из списка и копирует новый набор элементов в список назначения. |
| [back](https://msdn.microsoft.com/ru-ru/library/y4d8h412.aspx) | Возвращает ссылку на последний элемент в списке. |
| [begin](https://msdn.microsoft.com/ru-ru/library/eheeheb8.aspx) | Возвращает итератор, указывающий на первый элемент в списке. |
| [list::cbegin](https://msdn.microsoft.com/ru-ru/library/dd984436.aspx) | Возвращает итератор const, ссылающийся на первый элемент в списке. |
| [list::cend](https://msdn.microsoft.com/ru-ru/library/dd984297.aspx) | Возвращает константный итератор, адресующий расположение после последнего элемента в списке. |
| [list::clear](https://msdn.microsoft.com/ru-ru/library/xche46w9.aspx) | Удаляет все элементы списка. |
| [list::crbegin](https://msdn.microsoft.com/ru-ru/library/dd984300.aspx) | Возвращает константный итератор, адресующий первый элемент в обратном списке. |
| [list::crend](https://msdn.microsoft.com/ru-ru/library/dd779239.aspx) | Возвращает константный итератор, адресующий расположение после последнего элемента в обратном списке. |
| [list::emplace](https://msdn.microsoft.com/ru-ru/library/dd998367.aspx) | Вставляет элемент, созданный на месте, в указанное положение в списке. |
| [list::emplace\_back](https://msdn.microsoft.com/ru-ru/library/dd779438.aspx) | Добавляет элемент, созданный на месте, в конец списка. |
| [list::emplace\_front](https://msdn.microsoft.com/ru-ru/library/dd984295.aspx) | Добавляет элемент, созданный на месте, в начало списка. |
| [Empty](https://msdn.microsoft.com/ru-ru/library/a4swty34.aspx) | Проверяет, пустой ли список. |
| [end](https://msdn.microsoft.com/ru-ru/library/17465c5a.aspx) | Возвращает итератор, адресующий расположение после последнего элемента в списке. |
| [erase](https://msdn.microsoft.com/ru-ru/library/1fef72t6.aspx) | Удаляет элемент или диапазон элементов с указанных положений в списке. |
| [front](https://msdn.microsoft.com/ru-ru/library/a5e17kyc.aspx) | Возвращает ссылку на первый элемент в списке. |
| [get\_allocator](https://msdn.microsoft.com/ru-ru/library/99yb3sbe.aspx) | Возвращает копию объекта объекта **allocator**, использованного для создания списка. |
| [insert](https://msdn.microsoft.com/ru-ru/library/xt5w65ek.aspx) | Вставляет элемент или количество элементов или диапазон элементов в указанное положение в списке. |
| [max\_size](https://msdn.microsoft.com/ru-ru/library/wed3bxx3.aspx) | Возвращает максимальную длину списка. |
| [merge](https://msdn.microsoft.com/ru-ru/library/ec0yw4kc.aspx) | Удаляет элементы из списка аргументов, вставляет их в список назначения и сортирует новый о  бъединенный набор элементов по возрастанию или в ином указанном порядке. |
| [pop\_back](https://msdn.microsoft.com/ru-ru/library/37e8bh1e.aspx) | Удаляет элемент в конце списка. |
| [pop\_front](https://msdn.microsoft.com/ru-ru/library/s0xe8ba1.aspx) | Удаляет элемент в начале списка. |
| [push\_back](https://msdn.microsoft.com/ru-ru/library/tysfte5b.aspx) | Добавляет элемент в конец списка. |
| [push\_front](https://msdn.microsoft.com/ru-ru/library/2a84tfbt.aspx) | Добавляет элемент в начало списка. |
| [rbegin](https://msdn.microsoft.com/ru-ru/library/zs0x11et.aspx) | Возвращает итератор, адресующий первый элемент в обратном списке. |
| [remove](https://msdn.microsoft.com/ru-ru/library/kw666720.aspx) | Удаляет из списка элементы, совпадающие с заданным значением. |
| [remove\_if](https://msdn.microsoft.com/ru-ru/library/9fyx5kx8.aspx) | Удаляет из списка элементы, для которых выполняется заданное условие. |
| [rend](https://msdn.microsoft.com/ru-ru/library/yd0bb29w.aspx) | Возвращает итератор, адресующий расположение после последнего элемента в обратном списке. |
| [resize](https://msdn.microsoft.com/ru-ru/library/9fwb523y.aspx) | Задаёт новый размер для списка. |
| [Reverse](https://msdn.microsoft.com/ru-ru/library/eteb1fdc.aspx) | Изменяет порядок следования элементов в списке на обратный. |
| [size](https://msdn.microsoft.com/ru-ru/library/5a8xb6w6.aspx) | Возвращает количество элементов в списке. |
| [sort](https://msdn.microsoft.com/ru-ru/library/kz841ss7.aspx) | Упорядочивает элементы списка по возрастанию или в другом порядке. |
| [splice](https://msdn.microsoft.com/ru-ru/library/72fb8wzd.aspx) | Удаляет элементы из списка аргументов и вставляет их в список назначения. |
| [swap](https://msdn.microsoft.com/ru-ru/library/7x4139as.aspx) | Меняет местами элементы двух списков. |
| [unique](https://msdn.microsoft.com/ru-ru/library/thhhbacy.aspx) | Удаляет из списка повторяющиеся соседние элементы или соседние элементы,  удовлетворяющие условию какого-либо другого двоичного предиката. |

### 802d66bt.collapse_all(ru-ru,VS.120).gifОператоры

|  |  |
| --- | --- |
| [list::operator=](https://msdn.microsoft.com/ru-ru/library/dd984375.aspx) | Заменяет элементы списка копией другого списка. |

## [Требования](javascript:void(0))

**Заголовок**: <list>