Deque, или двусторонняя очередь, является структурой данных, которая позволяет добавлять и удалять элементы как в начале, так и в конце очереди. Это означает, что вы можете использовать Deque для решения широкого спектра задач, где вам нужно быстро добавлять и удалять элементы в очереди с обоих концов.

Класс Deque проще всего реализовать с помощью двусвязного списка. Он позволяет просматривать, удалять и добавлять элементы в начало и в конец списка.

**Абстрактный тип данных “дек”**

Deque()

* создаёт новый пустой дек.
* Не нуждается в параметрах
* возвращает пустой дек.

addFront(item)

* добавляет новый элемент в голову дека.
* Параметр (элемент) необходим;
* ничего не возвращает.

addRear(item)

* добавляет новый элемент в хвост дека.
* Параметр (элемент) необходим;
* ничего не возвращает.

removeFront()

* удаляет первый элемент из дека.
* Не нуждается в параметрах и возвращает элемент.
* Дек модифицируется.

removeRear()

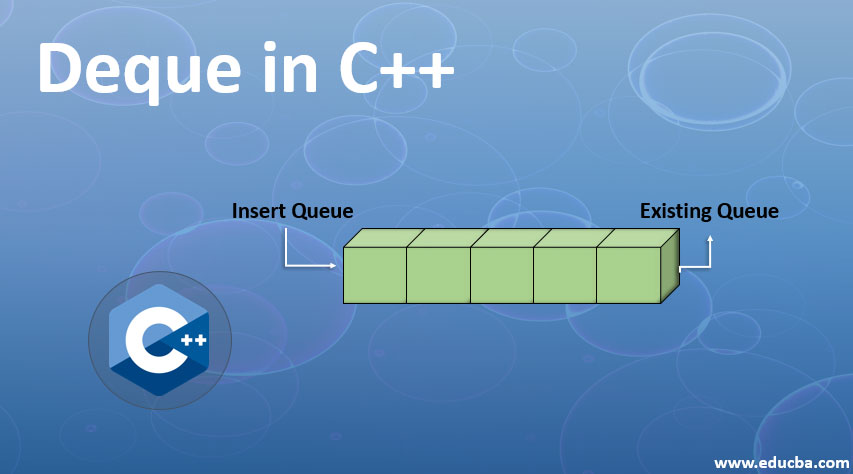
* удаляет последний элемент из дека.
* Не нуждается в параметрах и возвращает элемент
* Дек модифицируется.

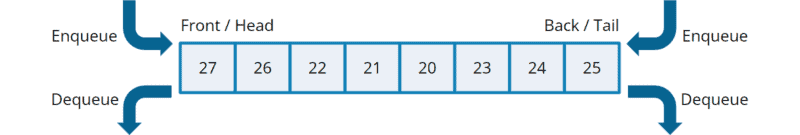
isEmpty()

* проверяет дек на пустоту.
* Не нуждается в параметрах
* возвращает булево значение.

size()

* возвращает количество элементов в деке.
* Не нуждается в параметрах
* возвращает целое число.





Пример на C++:

#include <iostream> // подключение библиотеки для работы с вводом-выводом

#include <deque> // подключение библиотеки для работы с деком

using namespace std;

int main() {

deque<int> myDeque; // объявление пустого дека с элементами типа int

myDeque.push\_front(10); // добавляем элемент в начало дека

myDeque.push\_back(20); // добавляем элемент в конец дека

myDeque.push\_front(5); // добавляем элемент в начало дека

cout << "Дек содержит " << myDeque.size() << " элементов: ";

for (int i = 0; i < myDeque.size(); i++) {

cout << myDeque[i] << " "; // выводим все элементы дека

}

cout << endl;

myDeque.pop\_front(); // удаляем первый элемент дека

myDeque.pop\_back(); // удаляем последний элемент дека

cout << "Теперь дек содержит " << myDeque.size() << " элементов: ";

for (auto it = myDeque.begin(); it != myDeque.end(); ++it) {

cout << \*it << " "; // выводим все элементы дека с помощью итератора

}

cout << endl;

return 0;

}

источники: <https://tproger.ru/translations/stacks-and-queues-for-beginners/>

https://www.happycoders.eu/algorithms/java-deque-vs-stack/