

## Глава 1

# Устройство элементов

Элементом двунаправленной очереди структура `node`, хранящий данные и ссылки на предыдущий и следующий узлы.

## Глава 2

# Устройство push/pop

### 2.1 Устройство push

`push_front(int k)/push_back(int k)` добавит новый узел в начало/конец, присвоит переменной `data` значение `k`, предыдущий `first/last` станет для нового узла `next/prev`, новый `node` станет `first/last`.

### 2.2 Устройство pop

`pop_front()/pop_back()` вернёт `data` первого/последнего элемента и, в отличие от функций `head()/tail()`, удалит этот узел. `node`, являющийся `next/prev` для удаленного, станет новым `first/last`.

## Глава 3

# Оценки сложности работы

- `head()` -  $O(1)$ , т.к. хранится в качестве переменной;
- `tail()` -  $O(1)$ , т.к. хранится в качестве переменной;
- `push_front(int k)` -  $O(1)$ , т.к. можно добавлять только с одной стороны;
- `push_back(int k)` -  $O(1)$ , т.к. можно добавлять только с одной стороны;
- `pop_front()` -  $O(1)$ , т.к. можно удалять только с одной стороны;
- `pop_back()` -  $O(1)$ , т.к. можно удалять только с одной стороны;
- `size()` -  $O(1)$ , т.к. хранится в качестве переменной;
- `print()` -  $O(n)$  - т.к. идём с первого элемента к последнему и выписываем;
- `clear()` -  $O(n)$  - т.к. идём с первого элемента к последнему и удаляем;