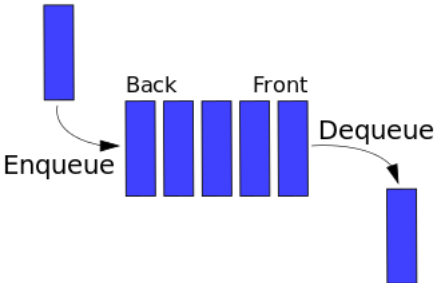


# Очередь (программирование)

Материал из Википедии — свободной энциклопедии

**О́чередь** — абстрактный тип данных с дисциплиной доступа к элементам «первый пришёл — первый вышел» (**FIFO**, англ. *first in, first out*). Добавление элемента (принято обозначать словом enqueue — поставить в очередь) возможно лишь в конец очереди, выборка — только из начала очереди (что принято называть словом dequeue — убрать из очереди), при этом выбранный элемент из очереди удаляется.



## Содержание

### Способы реализации очереди

- Массив
- Связный список
- Реализация на двух стеках

### Очереди в различных языках программирования

### Применение очередей

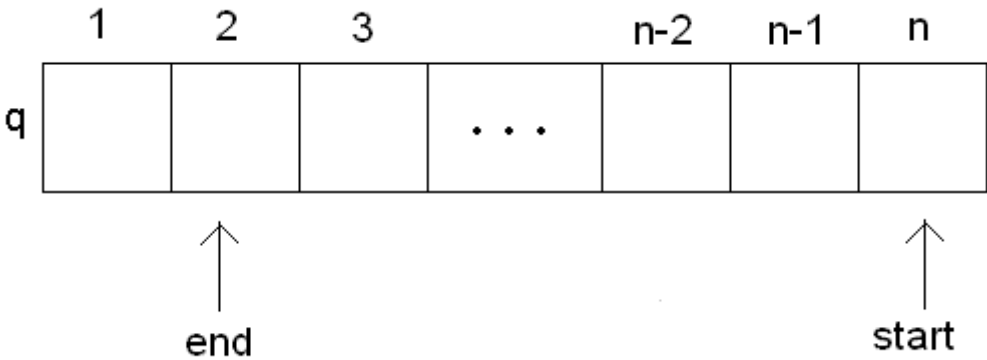
- См. также
- Примечания
- Ссылки

## Способы реализации очереди

Существует несколько способов реализации очереди в языках программирования.

### Массив

Первый способ представляет очередь в виде массива и двух целочисленных переменных start и end.



Обычно `start` указывает на голову очереди, `end` — на элемент, который заполнится, когда в очередь войдёт новый элемент. При добавлении элемента в очередь в `q[end]` записывается новый элемент очереди, а `end` уменьшается на единицу. Если значение `end` становится меньше 1, то мы как бы циклически обходим массив, и значение переменной становится равным `n`. Извлечение элемента из очереди производится аналогично: после извлечения элемента `q[start]` из очереди переменная `start` уменьшается на 1. С такими алгоритмами одна ячейка из `n` всегда будет незанятой (так как очередь с `n` элементами невозможно отличить от пустой), что компенсируется простотой алгоритмов.

Преимущества данного метода: возможна незначительная экономия памяти по сравнению со вторым способом; проще в разработке.

Недостатки: максимальное количество элементов в очереди ограничено размером массива. При его переполнении требуется перевыделение памяти и копирование всех элементов в новый массив.

## Связный список

Второй способ основан на работе с динамической памятью. Очередь представляется в качестве линейного списка, в котором добавление/удаление элементов идет строго с соответствующих его концов.

Преимущества данного метода: размер очереди ограничен лишь объёмом памяти.

Недостатки: сложнее в разработке; требуется больше памяти; при работе с такой очередью память сильнее фрагментируется; работа с очередью несколько медленнее.

## Реализация на двух стеках

Методы очереди могут быть реализованы на основе двух стеков `S1` и `S2`, как показано ниже:

```
Процедура enqueue(x):  
    S1.push(x)
```

```
Функция dequeue():  
    если S2 пуст:  
        если S1 пуст:  
            сообщить об ошибке: очередь пуста
```

```
    пока S1 не пуст:  
        S2.push(S1.pop())
```

```
    вернуть S2.pop()
```

Такой способ реализации наиболее удобен в качестве основы для построения персистентной очереди.

## Очереди в различных языках программирования

Практически во всех развитых языках программирования реализованы очереди. В CLI для этого предусмотрен класс `System.Collections.Queue` с методами `Enqueue` и `Dequeue`. В STL также присутствует класс `queue<T>`, определённый в заголовочном файле `queue`. В нём используется та же терминология (`push` и `pop`), что и в стеках.

## Применение очередей

Очередь в программировании используется, как и в реальной жизни, когда нужно совершить какие-то действия в порядке их поступления, выполнив их последовательно. Примером может служить организация событий в Windows. Когда пользователь оказывает какое-то действие на приложение, то в приложении не вызывается соответствующая процедура (ведь в этот момент приложение может совершать другие действия), а ему присылается сообщение, содержащее информацию о совершенном действии, это сообщение ставится в очередь, и только когда будут обработаны сообщения, пришедшие ранее, приложение выполнит необходимое действие.

Клавиатурный буфер BIOS организован в виде кольцевого массива, обычно длиной в 16 машинных слов, и двух указателей: на следующий элемент в нём и на первый незанятый элемент.

## См. также

---

- Коллекция
- Массив
- Список
- Стек
- Двухсторонняя очередь
- Очередь с приоритетом
- Именованный канал
- FIFO (информатика)
- LIFO (информатика)

## Примечания

---

## Ссылки

---

- Определение с NIST (http://www.nist.gov/dads/HTML/queue.html)

---

Источник — https://ru.wikipedia.org/w/index.php?title=Очередь\_(программирование)&oldid=92849237

---

**Эта страница последний раз была отредактирована 24 мая 2018 в 09:55.**

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.