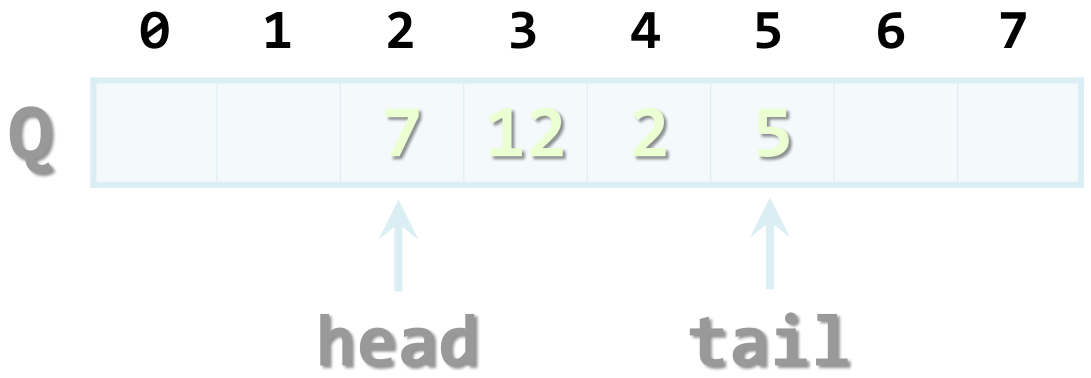
Опашка

Какво е опашка?

- **Опашката** е структура от данни, която има поведение от тип „първи влязъл, първи излиза“.
- Опашката може да се реализира:
  - **Статично**, чрез масив
  - **Динамично**, чрез възел със стойност и указател към следващ елемент

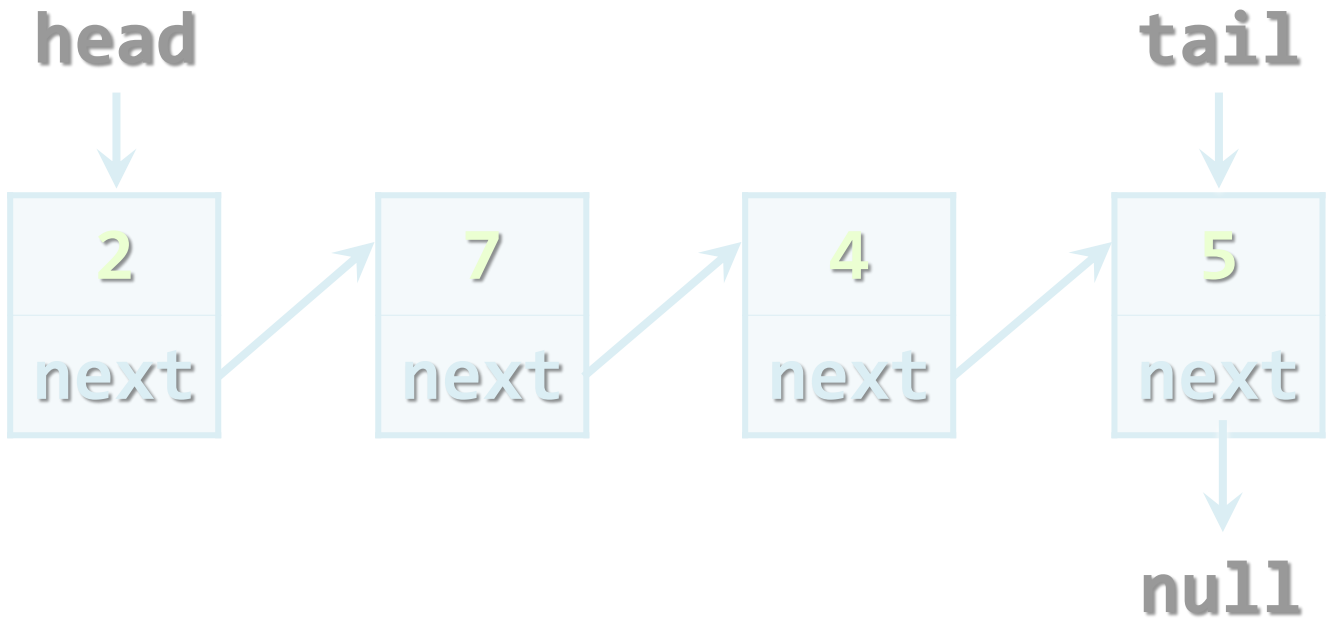
Статична (кръгова) опашка

- Статична (базирана на масив) имплементация
  - Имеплентира се като „кръгов масив“
  - Има ограничен капацитет (когато се запълни се заделя двойно място)
  - Има индекси за начало (**`head`**) и край(**`tail`**), сочещи към началото и края на кръговата опашка



Свързана Опашка

- Динамична имплементация
  - Всеки възел (node) има 2 полета: **`value`** и **`next`**
  - Позволява динамично създаване и изтриване



Queue<T> в .NET

- **`Queue<T>`** имплементира опашка чрез кръгов разтеглив масив
  - Елементите са от един и същ тип **`T`**
  - **`T`** може да бъде какъв да е тип, например **`int`** / **`Queue<int>`** / **`Queue<DateTime>`**
  - Размерът се увеличава динамично при нужда

Queue<T> Базова функционалност

- **`Enqueue(T)`** – добавя елемент в края на опашката

```
queue.Enqueue(5);
```

- **`Dequeue()`** – премахва и връща елемента от началото

```
int number = queue.Dequeue();
```

- **`Peek()`** – връща елемента от началото без триене

```
int number = queue.Peek();
```

- **`Count`** – връща броя елементи в

```
int elementCount = queue.Count;
```

Queue<T> Базова функционалност (2)

- **`Clear()`** – премахва всички елементи

```
queue.Clear();
```

- **`Contains(T)`** – проверява дали елемент се среща в опашка

```
bool isFound = queue.Contains(5);
```

- **`ToArray()`** – преобразува опашка в обикновен масив

```
int[] arr = queue.ToArray();
```

- **`TrimExcess()`** – изтрива допълнителното място

```
queue.TrimExcess();
```