

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

### Детали реализации

Вам необходимо реализовать шаблонный класс `Array` - упрощенный аналог `std::array`. Ваш класс должен быть параметризован типом хранимых элементов, а также размером массива. Как было сказано ранее, этот класс должен иметь ровно одно поле - C-style массив требуемого типа и размера. Важным моментом является то, что это поле должно быть публичным! В этом случае, как известно, становится доступна агрегатная инициализация вашего массива:

```
int arr[4]{1, 2, 3}; // массив {1, 2, 3, 0}
```

```
Array<int, 4> my_arr{1, 2, 3}; // если поле публично, то {1, 2, 3, 0}, иначе - СЕ
```

Для корректного прохождения тестов ваш класс должен реализовывать следующие методы (подумайте, какие методы должны быть константными, какие - неконстантными, а какие должны иметь обе версии):

- Операция `[]` для доступа к элементу массива по индексу;
- Метод `At(size_t idx)`, который обеспечивает безопасный (с проверкой границ) доступ к элементу по индексу. В случае выхода за границы необходимо бросать исключение типа `ArrayOutOfRangeException`, которое уже написано за вас (см. замечания);
- Методы `Front()` и `Back()` для доступа к первому и последнему элементу соответственно;
- Метод `Data()`, возвращающий указатель на начало массива;
- Методы `Size()` и `Empty()`;
- Метод `Fill(const T& value)`, который заполняет массив значениями `value`;
- Метод `Swap(Array<T, N>& other)`, обменивающий содержимое массивов одинакового размера.