

# Семинар #3: Массивы.

## Массивы

Массивы - это объекты, которые могут хранить внутри себя большое количество других объектов одного типа. Например, мы можем создать массив, который будет хранить 6 чисел типа `int` вот так:

```
int a[6] = {4, 8, 15, 16, 23, 42};
```

После того, как мы создали массив, мы можем получать доступ к каждому элементу массива по номеру. Номер элемента массива также называется его индексом. При этом нумерация в массиве начинается с нуля.

Массив a:	4	8	15	16	23	42
Индексы:	0	1	2	3	4	5

Доступ к элементу по индексу осуществляется через квадратные скобки. Например, если мы хотим поменять в массиве, определенном выше, число 15 на 20 нужно написать:

```
a[2] = 20;
```

## Подмассивы

Подмассив - это некоторая последовательная часть массива. В языке C нет никаких специальных средств для работы с подмассивами. Мы будем задавать подмассив в коде как два числа – индексы граничных элементов. Будем обозначать подмассивом `a[l, r]` такую часть массива, элементы которого имеют индекс `i` в диапазоне  $l \leq i < r$ . Обратите внимание, что мы договорились, что элемент `a[r]` не входит в подмассив `a[l, r]`.

Например, в подмассив `[1, 4]` массива `a` входят элементы 8, 15, 16, а элемент 23 не входит.

		l			r	
		↓			↓	
Массив a:	4	8	15	16	23	42
Индексы:	0	1	2	3	4	5

## Сортировка

Сортировка – это упорядочение элементов по возрастанию, убыванию или по какому-то другому критерию.

### Сортировка выбором

Сортировка выбором – это простейший алгоритм сортировки, который заключается в следующем:

Для каждого подмассива `[j, n]` (где `j` последовательно меняется от 0 до `n - 1`) поменять местами первый и минимальный элементы этого подмассива.

### Сортировка пузырьком

Сортировка пузырьком – это простейший алгоритм сортировки, который заключается в следующем:

Для каждого подмассива `[0, n - j]` (где `j` последовательно меняется от 0 до `n - 1`) мы делаем следующую операцию: пробегаем по этому подмассиву и, если соседние элементы стоят неправильно, то меняем их местами.