



Ву! ИИ готовит к ЕГЭ по информатике

[Попробовать](#)

## Урок Спринт 1.6

## Ёмкость, переполнение и реаллокация

Ёмкость слайса (capacity) определяет, сколько элементов может содержать в себе исходный массив. Если ёмкость переполнится, Go найдет новое место в памяти компьютера (примерно в два раза большее, чем занимал исходный слайс) и скопирует туда все элементы старого массива — реаллоцирует их.

Рассмотрим код:

```
a := make([]int, 2, 3)
fmt.Println(len(a), cap(a)) // 2, 3

a[0], a[1] = 0, 1

b := append(a, 1) // ёмкость a больше длины, копирования не происходит
b[0] = 21

c := append(b, 2) // ёмкость b равна длине, для увеличения массива создаётся копия
c[0] = 42

fmt.Println(a) // [21 1]
fmt.Println(b) // [21 1 1]
fmt.Println(c) // [42 1 1 2]
```

Мы создали слайс с длиной 2 и ёмкостью 3. Это значит, что исходный массив может содержать в себе максимум 3 элемента.

Первый вызов `append` увеличил длину слайса на один элемент, то есть длина слайса стала равна его ёмкости. Следующий вызов `append` вышел за рамки ёмкости слайса, поэтому был создан новый массив, который включил в себя элементы старого массива.

Мы поменяли нулевой элемент слайса `c`, но слайсы `c` и `b` ссылаются на разные массивы.

## Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»