



# Массивы

[Что такое массив](#)

[Индексы и элементы массива](#)

[Длина массива](#)

[Работа с массивами с помощью циклов](#)

## Что такое массив

Для хранения однотипных элементов в Java используется специальная структура — **массив**

Объявление массива:

```
String[] currencies = {"USD", "EUR", "JPY", "RUB"};
```

`String` — значения какого типа будут храниться в массиве. Могут быть `int`, `double`, `boolean` и т.п.

`[]` — указатель массива, отличает его от других типов данных

`currencies` имя переменной типа массив

`{"USD", "EUR", "JPY", "RUB"}` заполнение массива в соответствии с указанным типом данных



Чтобы указать значения массива после его объявления, необходимо перечислить их в фигурных скобках `{ }`. Заполнение массива значениями называется его **инициализацией**.

## Индексы и элементы массива

**Индекс** — номер элемента в массиве. Чтобы извлечь из массива определённый элемент, надо указать его индекс: `currencies[0]`. Индексация массива начинается не с единицы, а с **нуля**. Таким образом, в массиве из четырёх элементов первый индекс всегда будет 0, а последний — 3.

```
String[] currencies = {"USD", "EUR", "JPY", "RUB"}; //Массив со строковыми значениями.  
System.out.println("В Америке все цены в: " + currencies[0]); //Выведен фразу: "В Америке все цены в: USD"
```

Тип значения индекса элемента в массиве всегда — `int`, а тип самого элемента массива может быть любым. Для выполнения арифметических операций типы данных обоих операндов должны быть одинаковыми

```
double[] expenses = {1.5, 25, 11.25, 6};  
expenses[2] = expenses[2] + 115.0; // Увеличить в массиве значение expenses под индексом 2 на 115.0
```

## Длина массива

При инициализации массива с помощью фигурных скобок длина массива задаётся автоматически и равняется количеству элементов.

```
String[] currencies = {"USD", "EUR", "JPY", "RUB"};
```

Размер массива невозможно изменить после того, как массив создан: нельзя добавить в массив больше значений, чем определено при его создании, он задаётся явно, в момент создания.

Создание массива можно разделить на несколько этапов:

- объявление (указание имени переменной и типа данных)
- инициализация (заполнение массива значениями).

```
Тип_данных[] название_массива = {инициализация}
```

Используя синтаксис с ключевым словом `new`, можно явно указать длину массива и разделить объявление и инициализацию.

```
Тип_данных[] название_массива = new Тип_данных[количество_элементов]
```

```
название_массива [индекс элемента] = значение элемента ;
```

```
String[] currencies; // Объявляем массив, содержащий строки — названия валют  
currencies = new String[4]; // Создаем массив — его элементы пока пустые  
// Инициализируем массив  
currencies[0] = "USD";  
currencies[1] = "EUR";  
currencies[2] = "JPY";  
currencies[3] = "RUB";  
// Теперь массив готов к использованию
```



В качестве индекса можно использовать только те числа, которые больше или равны нулю и строго меньше длины массива. Это называется границы массива.

Если обратиться к несуществующему элементу массива (за его границы) появится исключение `ArrayIndexOutOfBoundsException` переводится как «ошибка выхода за границы массива». Чтобы избежать ошибок при вводе массива пользователем, воспользуйтесь ветвлением и проверяйте, входит ли введенный индекс в границы массива.

## Работа с массивами с помощью циклов

Для работы с массивами используют цикл `for`. Длина массива потребуется для того, чтобы задать условие выхода из цикла — она всегда известна и может быть получена с помощью свойства `length` `currencies.length`

```
for (int i = 0; i < currencies.length; i++) {  
    System.out.println("Поддерживаемая валюта: " + currencies[i]);  
}
```



Обратите внимание, что индекс последнего элемента в массиве всегда на единицу меньше, чем длина массива. При длине массива 4 индекс последнего элемента будет равен 3. Поэтому в цикле `for` обычно используют знак «строго меньше» — `i < currencies.length`