

Масиви

Емилиян Кадийски

<https://digitalrazgrad.org>
<https://digitaltargovishte.org>

Планът за днес

1. Какво са масивите?
2. Декларация
3. Как ги използваме?
4. Особености
5. Упражнение

A solid dark blue circle is centered on the left side of the image. Inside the circle, the word "МАСИВИ" is written in white, bold, uppercase letters.

МАСИВИ

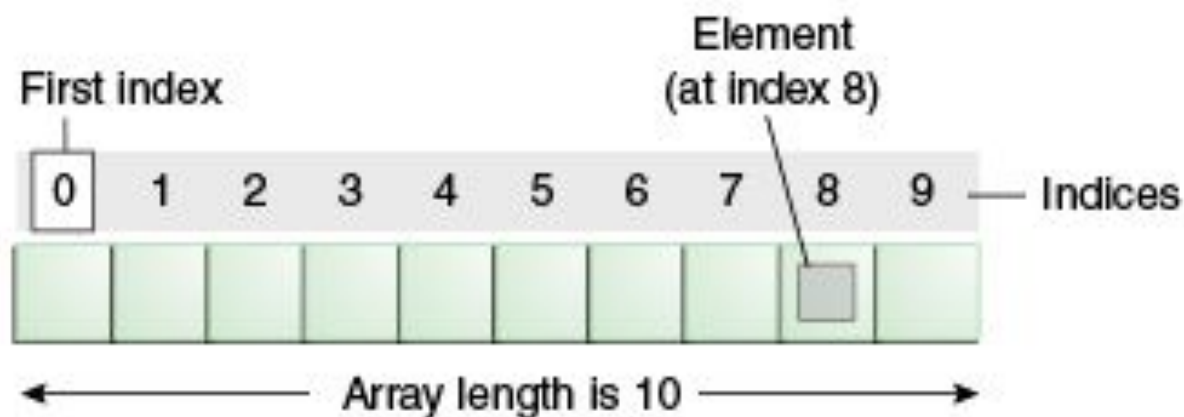
Определение за масив

Масивите са структура, представляваща наредена последователност от елементи от един и същ тип, към които можем да се обръщаме посредством общо име



Защо за ни нужни масивите?

Вместо да декларираме много променливи от един тип поотделно, понякога ни се налага да декларираме много променливи наведнъж и да ги групираме, за да имаме по-лесен достъп до тях.



A large, solid dark blue circle is centered on the page. Inside the circle, the word "ДЕКЛАРАЦИ" is written in white, bold, uppercase letters. Below it, the letter "Я" is also written in white, bold, uppercase letters.

ДЕКЛАРАЦИ
Я

Декларация

От всеки един тип може да се направи масив. Той се използва, когато имаме много променливи от един и същ тип и ни се налага да ги обхождаме машинно с помощта на цикли например.

масив от
променливи
от тип int



```
int[] numbers = {1, 2, 3, 4};
```

масив от
boolean



```
boolean[] isRaining = {true, true, false, false};
```

масив от char



```
char[] letters = {'a', 'b', 'c', 'd'}
```

Декларация

```
String[] cars = new String[2];
```

- знаем само броя на елементите
- декларираме масив от тип String, който искаме да съдържа два елемента
- операторът new заделя памет за масива
- масивът в момента не съдържа конкретни стойности, които сме задали ние, а елементите му са тези по подразбиране (за String е null, за int е 0 и т.н.)

```
String[] cars = {"Volvo", "BMW"};
```

- знаем стойностите на елементите
- декларираме масив от тип String, който искаме да съдържа низовете "Volvo" и "BMW"
- така с декларацията директно се случва и въвеждането на стойност
- масивът сам определя с каква дължина да е - броят на елементите в къдравите скоби, не е нужно да го посочваме ние изрично



**КАК ГИ
ИСПОЛЗВАМ
Е?**

Пример

Да се напише програма, която от подаден масив от дробни числа отпечатва всичките му елементи.

б) генерира нов масив със същия брой елементи като всеки елемент на първия масив е умножен по 2.

Примерен вход: `double [] nums = {2.5, 3.1, 6, 7.4, 4}`

Примерен изход: `numsDoubled: 5.0, 6.2, 12.0, 14.8, 8`

Достъп до елементите

Достъпът до елементите става с квадратни скоби []:

```
String[] bones = new String[206];  
bones[0] = "ulna";
```

Първия елемент на масива има стойност, всички останали са стойностите по подразбиране. Важно уточнение е, че **първият елемент е на позиция 0 в масива, а на последния - дължината на масива -1**

Индекси

Елементите в масива са **индексирани** и можем да достъпваме произволен елемент спрямо позицията му в масива. **Индексирането започва от 0, а последният елемент се намира на позиция `length - 1`.**

Взимаме
дължината
на масива

```
int[] numbers = {2, 6, 10, 30, 8, 5};  
System.out.println(numbers[0]);  
int length = numbers.length;  
System.out.println(numbers[length - 1]);
```

Принтира 2
на конзолата

Принтира 5
на конзолата

Обхождане

Често ни се налага да минем през всички елементи на масив и да направим нещо с тях. Този процес се казва **обхождане**. Възможен е чрез цикли.

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {
    System.out.println(cars[i]);
}
```

Можем да използваме дължината на масива, за да знаем колко пъти да завъртим цикъла

A solid dark blue circle is centered on the page. Inside the circle, the Russian words "ОСОБЕННОСТ" and "И" are written in white, bold, sans-serif capital letters, stacked vertically.

**ОСОБЕННОСТ
И**

Фиксирана дължина

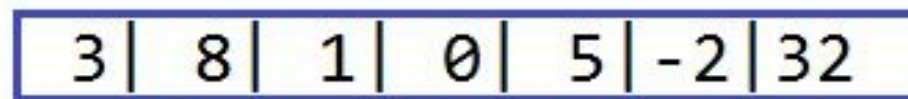
Масивите имат **фиксирана дължина**. След като веднъж заделим памет за нашия масив, не можем да добавяме нови елементи към него. Можем да **достъпваме дължината** на нашия масив с помощта на **length**

```
int[] numbers = {2, 6, 10, 30, 8, 5};  
System.out.println(numbers.length);
```

ArrayIndexOutOfBoundsException

Доста често срещано изключение е `ArrayIndexOutOfBoundsException`. Той се хвърля, за да покаже, че се опитваме да получим достъп до елемент на масив с нелегален индекс. Това изключение се изхвърля, когато индексът е или отрицателен, или по-голям или равен на размера на масива.

Array :



Indices:

0 1 2 3 4 5 6

Сравняване на масиви

Масивите не могат да се сравняват, както примитивните типове. Ако се опитахме да ги сравним с помощта на `==`, ние ще проверим дали адресите им в паметта съответстват, а не дали елементите им са еднакви. За да ги сравним, трябва да ги обходим и поелементно да направим сравнение.

Именуване

Когато създаваме масиви, трябва да **следваме конвенциите за именуване на променливи** - имената трябва да са достатъчно **описателни**, за да можем да разберем какво съдържа масивът. Не е добра практика да си кръщавате масивите само `array` например, тъй като това име не носи достатъчно информация.

Примерни задачи

1. Имате масива `int[] numbers = {1, 2, 3, 4, 5};`
Обърнете елементите му, така че новия масив да е 5, 4, 3, 2, 1.
2. Даден ви е следният масив:
`int[] numbers = {43, 56, 4, 3, 6, 8, 43, 5, 7, 87, 4, 3, 5, 6}.`
Намерете най-големия елемент в него и го изведете. Програмата ви трябва да работи за всички масиви от тип `int`.

Самостоятелни задачи

задачи за упражнение

Въпроси?



ДИГИТАЛНО
ОБЩЕСТВО
ТЪРГОВИЩЕ



ДИГИТАЛНО
ОБЩЕСТВО
РАЗГРАД

ДА
ОТВОРИМ
КРЪГА



КЛУБ НА НСО

Trainings @ Digital Razgrad & Digital Targovishte

- Digital Razgrad
 - <https://digitalrazgrad.org>
 - <https://facebook.com/digitalrazgrad.org>
 - digitalrazgrad.slack.com
- Digital Targovishte
 - <https://digitaltargovishte.org>
 - <https://facebook.com/digitaltargovishte.org>
 - digitaltargovishte.slack.com

